Robust identification of users by convolutional neural network in MATLAB and Raspberry Pi

Paula Useche Murillo[1], Robinson Jiménez-Moreno[1], Javier Eduardo Martínez Baquero[2]
[1]Department of Mechatronic Engineering, Universidad Militar Nueva Granada, Bogotá, Colombia
[2]Engineering School, Faculty of Basic Sciences and Engineering, Universidad de los Llanos, Villavicencio, Colombia

Article Info

Article history:

Received Feb 14, 2024
Revised Apr 26, 2024
Accepted May 12, 2024

Keywords:

Convolutional network
Directed acyclic graph-convolutional neural network
Face recognition
MATLAB
Raspberry pi

ABSTRACT

The following article presents the development of an algorithm embedded in a Raspberry Pi 3B board, where a user identification was made, using the convolutional neural network (CNN) for 5 predefined users, with the option of loading remotely a new network for a new user. Comparatively, the same application was programmed in MATLAB programming software to evaluate the results and identify the advantages between them. Networks were trained for 5 different users, using the Caffe library on the Raspberry Pi, and the MATLAB neural network package on the computer. Where it was found that the training made by Caffe on an embedded system is much slower and less efficient than the ones performed in MATLAB, obtaining less than 55% accuracy with Caffe networks and more than 90% with MATLAB networks, training with the same number of samples, the same architecture, and the same database. Finally, the accuracy obtained through confusion matrix is over 88% in each case of users identification.

Corresponding Author:

Robinson Jiménez-Moreno
Department of Mechatronic Engineering, Universidad Militar Nueva Granada
Carrera 11 # 101-80, Bogotá, Colombia
Email: robinson.jimenez@unimilitar.edu.co

1. INTRODUCTION

Facial recognition systems have had a great importance today due to their wide field of action, which has led them to be used in different applications and requirements of society, such as solving the problem of student attendance at the university, demonstrated in [1], or its implementation in security office environments, as discussed in [2]. Additionally, the development and use of new electronic devices such as dedicated programmable embedded boards, as the Raspberry Pi type, have allowed facial recognition algorithms to be used in all types of environments. Such as a system's capability to not only detect unauthorized access but also to identify individuals through facial recognition [3] and with a wide programming capacity of any type of application, such as those developed in [4], where was designed an access secure control system by user facial detection to replace automated teller machine (ATM) cards and personal identification number (PIN).

Some applications established their approach to facial recognition embedded in Raspberry Pi, to assess the ability of this board to perform face detection and classification tasks in real time, such as the projects developed in [5] and [6]. Where Haar classifiers and local binary pattern algorithm (LBP) for facial recognition are used, and they are compared in [7], concluding that the accuracy of Haar cascade is more than the local binary pattern but the execution time in Haar cascade is more than local binary pattern. On the other hand, other embedded applications have been developed in the Raspberry Pi, which focused on the generation of security algorithms by user identification, applicable to real life. In study [8] Raspberry Pi platform was used to create a smart door lock system based on facial recognition. In study [9] the LBP algorithm was used along with principal component analysis (PCA) and fisher faces, which allow to increase the speed of facial recognition with respect to the other mentioned methods. Beyond the programmable electronic boards and their capacity and limited processing speed, multiple projects have been developed to generate more robust facial recognition systems, such as those presented in [10], [11].

Furthermore, artificial intelligence techniques have been implemented that use convolutional filters for patterns' recognition and image classification, such as the convolutional neural networks (CNN) [12], for user's identification and faces recognition. In study [13] shows some examples of the CNN used for facial recognition where the labeled faces in the wild (LFW) database was implemented in many of the training and tests. In study [14], the FaceNet was developed, which consists of an optimized deep CNN (DCNN) that allows facial recognition with only 128 bytes of information per face. Finally, in [15], a CNN was used for the recognition of smiles on faces, with an accuracy of 97.6%, where tests were performed with more than 10 different subjects, and it was possible to differentiate a smile from different facial expressions.

In study [16], the results of a previous work are presented for the detection and recognition of faces by CNN, without embedding it in a programmable board, where Haar AdaBoost classifiers were used for detection, and a shallow network, with 3 layers of convolution, for the identification of three users (Paula, Javier and others), where 99.3% accuracy was achieved. In this article, the development of an application focused on the robust identification of users is presented, programmed in both Raspberry Pi and MATLAB, developed with artificial intelligence techniques, such as CNN. This application presents an innovative method of user detection, where it is possible to access the program remotely (in the case of the Raspberry Pi) and select a specific network for the classification of a particular user or load a new network for identification of a new user, which compares the quality of face recognition executed in two different programming systems and its scope.

The present article is divided into four main sections: in the first section, a brief introduction is made about facial recognition systems and the different user identification techniques developed in the state of the art. In the second section, the functionality of the algorithm is explained. In the third section, the results of the algorithm application with different users are shown and an analysis is made about them. The conclusions reached are presented in the fourth section.

2. MATERIALS AND METHODS

For user identification by face, the present development required the design of a shallow custom convolutional network architecture, capable of being embedded in the Raspberry Pi, which has limited storage capacity. Next, the basic operation of the program is presented, where the graphic interface and the networks designed, its implementation, and the logic of the developed algorithm are divided and explained into 6 subsections, for user identification and subsequently comparing the performance of the networks in MATLAB and Raspberry Pi. The first subsection explains the behavior of the algorithm and its flow of operation. The second one raises the working conditions necessary for the correct execution of the application. The third subsection describes the initialization process of variables prior to the beginning of the application. The fourth one explains the process of facial detection by Haar classifiers. The fifth subsection describes the user classification process by CNN architectures, and finally, the sixth subsection details the architecture of the trained networks and their accuracy percentages, both for Raspberry Pi and MATLAB software.

2.1. Algorithm operation

The general algorithm developed requires receiving as input a previously trained CNN for the detection of a specific user. Where only a face that matches the user entered is marked with a detection frame, if the classification percentage is greater than one limit previously established. The program allows selecting the user to recognize between a pre-loaded database or entering a new network for a new user.

The sequence of operation of the algorithm, for both Raspberry Pi and MATLAB, is presented in the diagram of Figure 1, where each step was marked with numbers from 0 to 2. Figure 1 is explained in this section with the results of research and at the same time is given the comprehensive discussion. Results can be presented in figures, graphs, tables, and others that make the reader understand easily. In step 0, it is expected that the user selects some option of the graphical interface, and the variables required by the program are initialized.

In step 1, the detection of all the faces present in the video capture is performed, using Haar classifiers [7]. In step 2, each face of the video is trimmed and evaluated with the CNN [17] loaded in the interface, selecting as user that face that is classified in the category of the expected individual with a percentage of confidence higher than that established. Steps 1 and 2 are repeated until the user presses the letter “q” (in the case of the Raspberry Pi) or press START again (in the case of MATLAB). Then the While stops and the program ends.

Next, the working conditions required for the execution of the algorithm are presented and each of the steps shown in the flow diagram of Figure 1 are explained. A Raspberry Pi 3 model B and a Webcam were used to capture the work environment and the input image was re-sized into a scale of 1.5 from 480×640 pixels to 320×427 pixels, to reduce memory requirements and accelerate the face detection process by reducing the number of pixels to be evaluated. In the case of the MATLAB application, a Webcam was also used, but the input image was maintained at 480×640 pixels, given the bigger computer’s processing capacity without considerably affecting the speed of operation of the application.

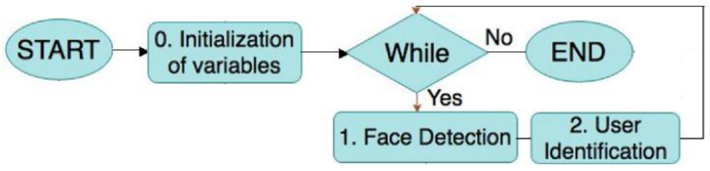

Figure 1. Algorithm flow diagram

Five networks were trained to identify 5 different users (Cesar, Julian, Paula, Javier, and Robinson) whose names and faces are shown in Figure 2 as preloaded users. To perform the robust identification process, a CNN was trained for everyone, where the network determined if the face entered corresponds to the expected and in what percentage, or if does not match and belongs to the “Others” category. An initial database of 400 photos per person was taken, and Haar classifiers were used for the detection being the face images trimmed the training and test database. All the extracted faces were resized to 70×70 pixels, and each image was checked to ensure that the entire face was visible. The database was increased using the data augmentation program of [18], where the brightness was varied and noise was added to the photos, until obtaining a total of 2,400 training images, and 420 of test images, by category.

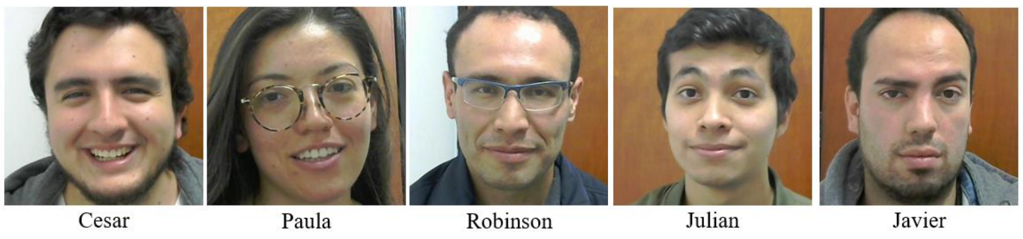

Figure 2. Pre-loaded users in the application

To generate the remote connection with the Raspberry Pi, VNC was used, an application that allows you to control the board through the internet and share files with it. In this way, it is possible to send a new network, already trained, and upload it to the program through the interface. To load a new network to the application, it must consist of a *.caffemodel* file and a *.protxt* file, since the application works with the Caffe artificial intelligence package [19]. In the case of MATLAB, the networks are loaded directly from the file browser, as shown in subsection 2.5, where it is only necessary to load a “*.mat*” file, which contains both the trained network and the user’s name to identify.

2.2. Step 0: initialization of variables

The first step of the algorithm is to initialize the variables that will be used throughout the program, which are found in Table 1, where its function, name and initialization value are mentioned, both for the application developed in Raspberry Pi, and for the MATLAB. The first contact that the user has with the Raspberry Pi program is with the graphical interface shown in Figure 3. Where the five top buttons have a CNN network preloaded for the user written on the button, and the three lower elements consist of a button to cancel the program (“Cancel”). A box to enter the name of a new user to be identified, and an “Open” button with which a new trained network is loaded, corresponding to the user entered in the text box.

Table 1. Variable initialization

Variable	Function	Value
dimCNN	Image dimensions of the CNN input in pixels	70×70
net	Network	*.caffemodel* *.protxt* (Raspberry) *.mat* (MATLAB)
porc	Minimum percentage of confidence to identify different users	95

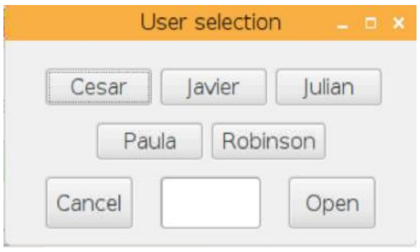

Figure 3. Graphical user interface in the Raspberry Pi

In case the user selects any of the buttons with pre-loaded CNN, the program automatically starts, executing steps 1 and 2 of Figure 1. Otherwise, fill in the name of the new user and then press "Open" to select a caffe model file and another *.protxt*, which correspond to the new network to be loaded. After selecting the files, the program automatically follows steps 1 and 2 into the algorithm.

2.3. Step 1: face detection

In step 1, face detection is generated by Haar classifiers [7], to extract all the faces, present in the work environment and then enter them to CNN in step 2. To do this, a while cycle was designed, as it is observed in Figure 1, that in each iteration evaluates a frame, and extracts all the users whose faces have been detected. Basically, the program takes a photo with the camera and enters it into the Haar classifier, which delivers a "Faces" variable that contains the coordinates [x y width height] of all the detected faces (one row per face), with [x y] being the coordinates from the upper left corner of the cropped image, taken with respect to the upper left corner of the global image, and [width height] the width and height, respectively, of that frame (all in pixels).

2.4. Step 2: user identification

In step 2, the detected faces (Faces) are entered to CNN for classification, and then they are passed to a saturator, in charge of filtering all those individuals who did not qualify as the expected user, or who did recognize themselves as the user. but with a percentage of confidence less than *porc*, showing on the screen, only, the detection of those faces belonging to the category of the expected user, such as the example shown in Figure 4. In case the user is categorized as "Others", no detection frame is generated.

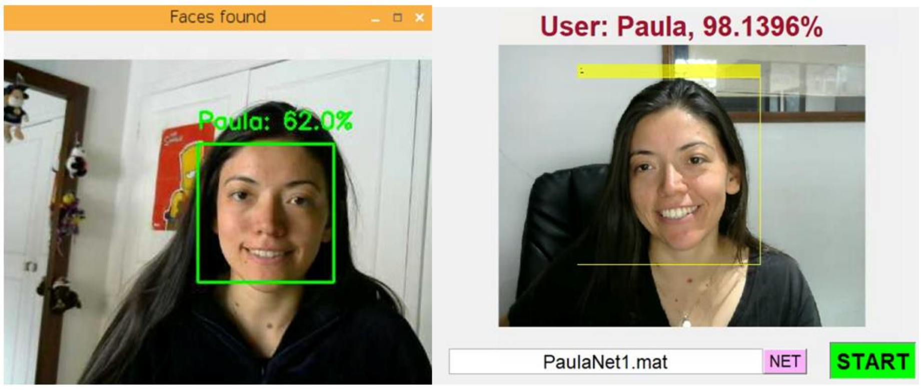

Figure 4. User identification "Paula", Raspberry Pi (Left), MATLAB (Right)

2.5. Convolutional neural networks

Convolutional neural network (CNN), as mentioned in [20], is a type of neural network designed for the classification of images and patterns, by convolutional filters that allow the extraction of relevant characteristics of an image. These filters go through the image, and during the training, they vary the value of their weights to learn the necessary characteristics to achieve the classification. Finally, the extracted characteristics are entered into a softmax layer that selects the correspondence category according to a probability bell.

Five CNNs were trained in MATLAB and 5 in Caffe, one for each user, all with the same architecture, trained during the same number of epochs and with the same database. Table 2 shows the CNN architecture used for the classification of preloaded users, where input images of 70×70 pixels were used, and square filters for the equivalent extraction of both horizontal and vertical information. As shown in Table 1 and as indicated in [12], CNN has convolutional type layers (CONV), rectified linear units (ReLU), MaxPool (POOL), average pool, batch normalization (BATCH), fully connected (FC), DropOut (DROP), softmax (SOFT), among others.

Table 2. CNN Architecture to Identify Users

LAYERS	Filter Size H×W	Stride	Number of Filters
CONV+BATCH+RELU	8×8	1	16
CONV+BATCH+RELU	7×7	1	64
MAXPOOL	3×3	2	--
CONV+BATCH+RELU	6×6	1	128
CONV+BATCH+RELU	5×5	1	256
MAXPOOL	3×3	2	--
CONV+BATCH+RELU	3×3	1	512
FC1+RELU+DROP	--	--	--
FC2+RELU+DROP	--	--	--
FC3+SOFT	--	--	--

3. RESULTS AND DISCUSION

The networks for each of the users were trained for 17 epochs, with a MiniBatchSize of 48, 2,400 training images and 420 test images per category, graphics processing unit (GPU) NVIDIA GTX 1060 6 GB for training in MATLAB. Central processing unit (CPU) was used for training with Caffe in the Raspberry Pi with Cortex-A53 of 64 bits to 1.4 GHz [21]. The low computational profile of the Raspberry limits the development characteristics of the computing equipment to parameterize the comparison between the networks.

The number of training epochs and the MiniBatchSize were selected according to the capacity and the processing time of the Raspberry Pi, where it was observed that to achieve the 17 epochs, it was necessary to train the networks for 5 consecutive days, and that, additionally, the board did not have enough capacity to handle a MiniBatchSize greater than 50, and using a smaller one would imply a greater number of epochs so that the network could learn all the images. Figure 5 shows the confusion matrix [22], [23] of each of the networks trained in MATLAB where the user Javier had the lowest percentage of accuracy with 87.14%, and Julian and Paula had the highest percentages of 98.93%. For users Cesar, Javier and Julian, 1 corresponds to their names, and 2 to others, while for the other two users, 1 is from others and 2 corresponds to their names.

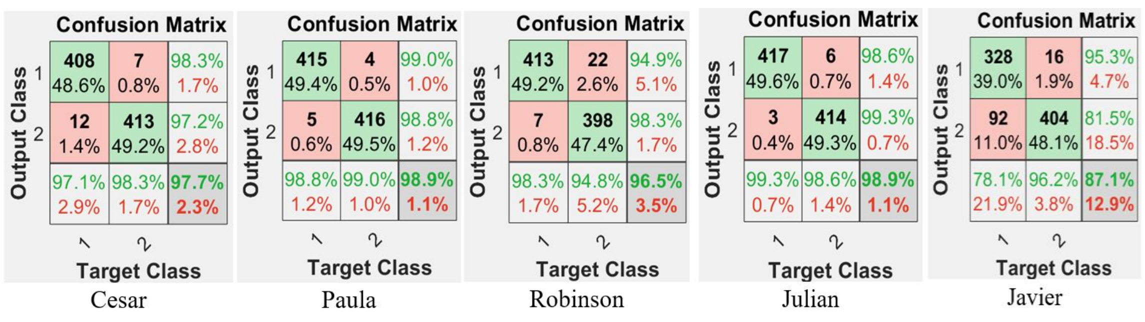

Figure 5. Confusion matrix of each user using the CNN

In the case of the networks trained in the Raspberry Pi, the percentages of accuracy ranged between 50% and 60%. However, when using the networks in the real application, the same classification was always generated regardless of the user, i.e., all the faces were classified in the same category, as shown in the example of Figure 6(a). All users were classified as Julian independent of the resemblance to the real user. Figure 6(b) shows an example of the robust identification for the user Paula in the case of the application in MATLAB, where more than one individual is found in the work environment, but only Paula's face is detected, with a high percentage of confidence.

The training of the CNNs of user identification in both systems differed in numerous aspects, despite having the same architecture, and the same training characteristics. First, because the Raspberry Pi does not have a GPU that can be implemented with the CNNs, the training time of each network was considerably higher than the training times in MATLAB, where a GPU was used. While MATLAB trained 17 epochs in less than 10 minutes, the Raspberry Pi took around 5 consecutive days.

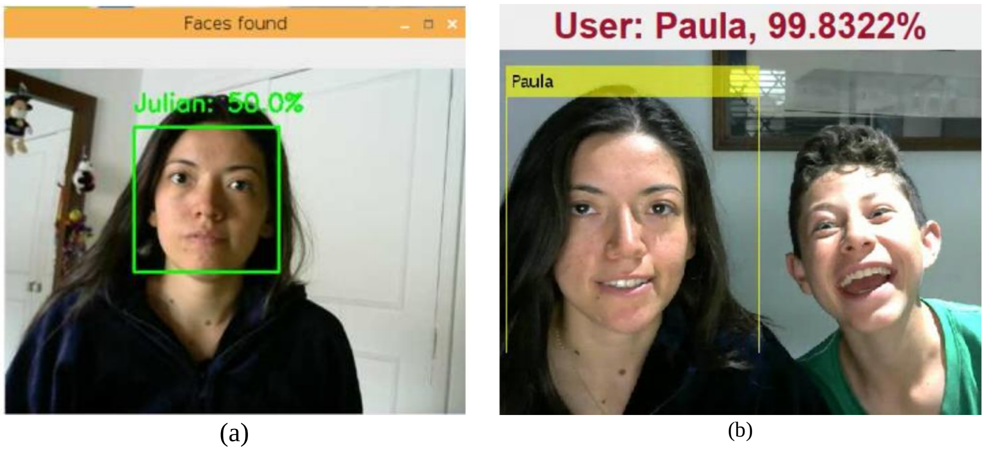

Figure 6. User identification (a) Raspberri-pi and (b) MATLAB

Additionally, MATLAB trainings are better optimized than in Caffe's bookstores, since, despite having trained each network for such a short period of time (with training of 300 epochs or more being usual), the accuracy percentages were higher than 85%, most of them very close to 95%. While the networks trained with Caffe did not exceed 60%, and in practice all the faces were classified in the same category, leading to suppose that a classification is not really generated. As it could be observed in Figure 6, it was necessary to reduce the value of *porc* to 60% for the case of the Raspberry Pi, while for MATLAB, it was possible to recognize the user with more than 98%.

The reason for training CNN-type networks in the Raspberry Pi, was to generate an embedded user identification system that could be implemented in some specific environment. However, as was observed when comparing Caffe training with MATLAB training, acceptable results we not achieved to achieve the objective. So it was decided to deepen the application developed in MATLAB, and seek an improvement in the quality of face recognition, using networks type directed acyclic graph-CNN (DAG-CNN) [24], [25] (the which cannot be programmed in Caffe because it does not support that type of architectures). DAG-CNN allow to extract characteristics of different shapes and dimensions by CNN branches, from the same input image, thanks to the fact that it has more than one convolution line with different hyperparameters per branch. For the DAG-CNN [26] an architecture was used as shown in Table 3, where both branches receive the input image, and join after the last FC (FC2+ReLU+DROP) in a single SOFT layer, to then generate the classification. The filters were selected in such a way that overall image information could be captured (large filters with stride greater than one) and detailed information (small filters and single stride filters).

By using the DAG-CNN it was possible to obtain percentages of accuracy of 99% and 100% in facial recognition, as shown in the confusion matrices of Figure 7, with training times of around 14 minutes, 17 epochs and MiniBatchSize of 48. Although the DAG-CNN for Javier also had a percentage lower than 90% (as happened with CNN), there was a lower number of oscillations during training and testing, and better accuracy, so it is considered that this user requires a greater number of training epochs to achieve a better percentage. However, the training and its accuracy in all cases, per user, is high.

Table 3. Architecture of each DAG-CNN branch

LAYERS BRANCH 1	Filter size H×W	Stride	Number of filters	LAYERS BRANCH 2	Filter size H×W	Stride	Number of filters
CONV+BATCH+RELU	8×8	2	16	CONV+BATCH+RELU	6×6	1	16
CONV+BATCH+RELU	7×7	1	64	CONV+BATCH+RELU	4×4	1	32
MAXPOOL	2×2	2	--	MAXPOOL	2×2	2	--
CONV+BATCH+RELU	6×6	1	128	CONV+BATCH+RELU	3×3	1	64
CONV+BATCH+RELU	5×5	1	512	CONV+BATCH+RELU	3×3	1	128
FC1+RELU+DROP	--	--	--	MAXPOOL	3×3	2	--
FC2+RELU+DROP	--	--	--	CONV+BATCH+RELU	3×3	1	256
				CONV+BATCH+RELU	3×3	1	512
				FC1+RELU+DROP	--	--	--
				FC2+RELU+DROP	--	--	--

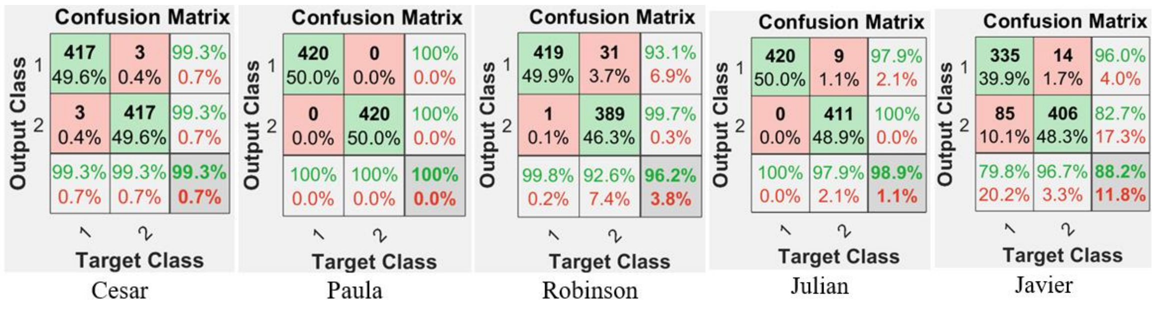

Figure 7. Confusion matrix of each user using the DAG-CNN

4. CONCLUSION

It was possible to generate a robust user identification system, based on low-cost tools, with a high percentage of accuracy, over greater than 95% in most cases. MATLAB tools allow achieve the target, while with Caffe’s bookstores it was not possible to generate a successfully CNN that had the capacity to differentiate one face from another, due to the large number of parameters that must be identified to achieve this. With a CNN it is possible to generate robust facial recognition systems, however, with the DAG-CNN, better results are obtained, using twice the training time, but with a more complex feature extraction that allows to evaluate the input image with filters, both large and small, simultaneously, allowing to obtain accuracies of up to 100% in only 17 epochs or less.

The Raspberry Pi embedded systems used allow a robust facial identification system, able to be used for several applications by biometric information. So, for this case the obtained results with specific learning parameters and few users, without requiring a high-cost computer, is a good solution, however, big data base and better facial recognition results would be obtained on a computer with GPU. Future work use vision processing units (VPU) like “Intel Neural Compute Stick 2” to improve the performance of the raspberry pi and its use with convolutional networks, which allow deeper architectures, better accuracy and processing time.

ACKNOWLEDGEMENTS

The authors are grateful to Universidad Militar Nueva Granada, which through its Vice chancellor for research finances the present project with code IMP-ING-2290 and titled "Prototype of robot assistance for surgery", from which the present work is derived and Universidad de los Llanos for its participation.

REFERENCES

[1] A. Budiman, Fabian, R. A. Yaputera, S. Achmad, and A. Kurniawan, “Student attendance with face recognition (LBPH or CNN): Systematic literature review,” *Procedia Computer Science*, vol. 216, pp. 31–38, 2023, doi: 10.1016/j.procs.2022.12.108.
[2] G. Rajeshkumar *et al.*, “Smart office automation via faster R-CNN based face recognition and internet of things,” *Measurement: Sensors*, vol. 27, p. 100719, Jun. 2023, doi: 10.1016/j.measen.2023.100719.
[3] A. K. Singh, M. Dhawan, E. D. Sharma, and S. Kumar, “Advance anti-theft flooring security system using Raspberry PI,” in *2023 International Conference on Recent Advances in Science and Engineering Technology (ICRASET)*, IEEE, Nov. 2023. doi: 10.1109/icraset59632.2023.10420254.
[4] C. K. M. J, G. V, and K. Arunkumar, “Bio-Pi cash: next gen cardless ATM with facial recognition and fingerprint security using Raspberry Pi,” in *2023 2nd International Conference on Automation, Computing and Renewable Systems (ICACRS)*, IEEE, Dec.

2023. doi: 10.1109/icacrs58579.2023.10404290.
[5] H. Meddeb, Z. Abdellaoui, and F. Houaidi, "Development of surveillance robot based on face recognition using Raspberry-PI and IoT," *Microprocessors and Microsystems*, vol. 96, p. 104728, Feb. 2023, doi: 10.1016/j.micpro.2022.104728.
[6] M. Chavhan, N. Vyavhare, S. Rampelliwar, M. Kamble, and N. Totala, "Surveillance and image processing with Off-Terrain vehicle," *Materials Today: Proceedings*, Jun. 2023, doi: 10.1016/j.matpr.2023.05.611.
[7] A. B. Shetty, Bhoomika, Deeksha, J. Rebeiro, and Ramyashree, "Facial recognition using Haar cascade and LBP classifiers," *Global Transitions Proceedings*, vol. 2, no. 2, pp. 330–335, Nov. 2021, doi: 10.1016/j.gltp.2021.08.044.
[8] M. Yashashwini, K. S. Kumar, R. Pitchai, K. S. Sai Sankeerth, G. Arun Prasath, and D. Trinath, "Face recognition based smart door lock system using convolution neural network," in *2023 International Conference on Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering (RMKMATE)*, IEEE, Nov. 2023. doi: 10.1109/rmkmate59243.2023.10368950.
[9] Umm-e-Laila, M. A. Khan, M. K. Shaikh, S. A. bin Mazhar, and K. Mehboob, "Comparative analysis for a real time face recognition system using raspberry Pi," in *2017 IEEE 4th International Conference on Smart Instrumentation, Measurement and Application (ICSIMA)*, IEEE, Nov. 2017. doi: 10.1109/icsima.2017.8311984.
[10] J. Guo, Y. Fu, and T. Liu, "Automatic face recognition of target images based on deep learning algorithms," *Microprocessors and Microsystems*, p. 104936, Sep. 2023, doi: 10.1016/j.micpro.2023.104936.
[11] H.-K. Jee, S. Jung, and J.-H. Yoo, "Liveness detection for embedded face recognition system," *International Journal of Biological and Medical Sciences*, vol. 1, pp. 235–238, Dec. 2006.
[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Neural Information Processing Systems*, 2012, pp. 1106–1114.
[13] C. R. Kumar, S. N, M. Priyadharshini, D. G. E, and K. R. M, "Face recognition using CNN and Siamese network," *Measurement: Sensors*, vol. 27, p. 100800, Jun. 2023, doi: 10.1016/j.measen.2023.100800.
[14] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2015. doi: 10.1109/cvpr.2015.7298682.
[15] M. Matsugu, K. Mori, Y. Mitari, and Y. Kaneda, "Subject independent facial expression recognition with robust face detection using a convolutional neural network," *Neural Networks*, vol. 16, no. 5–6, pp. 555–559, Jun. 2003, doi: 10.1016/s0893-6080(03)00115-1.
[16] P. Catalina Useche M., J. O. Pinzo Arenas, and R. Jimeez Moreno, "Face recognition access control system using convolutional neural networks," *Research Journal of Applied Sciences*, vol. 13, no. 1, pp. 47–53, Nov. 2019, doi: 10.36478/rjasci.2018.47.53.
[17] J. O. Pinzón-Arenas, R. Jimenez-Moreno, and J. E. Martinez Baquero, "Comparison of convolutional neural network models for user's facial recognition," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 14, no. 1, p. 192, Feb. 2024, doi: 10.11591/ijece.v14i1.pp192-198.
[18] P. Useche Murillo, J. Pinzón-Arenas, and R. Moreno, "Implementation of a data augmentation algorithm validated by means of the accuracy of a convolutional neural network," *Journal of Engineering and Applied Sciences*, vol. 12, pp. 5323–5331, Nov. 2017, doi: 10.36478/jeasci.2017.5323.5331.
[19] Y. Jia *et al.*, "Caffe: convolutional architecture for fast feature embedding," in *Proceedings of the 22nd ACM international conference on Multimedia*, in MM '14. ACM, Nov. 2014. doi: 10.1145/2647868.2654889.
[20] E. Bulus, "Gender determination from pictures with CNN models," in *2021 6th International Conference on Computer Science and Engineering (UBMK)*, IEEE, Sep. 2021. doi: 10.1109/ubmk52708.2021.9558915.
[21] F. Duque, F. Rivera, and R. Velásquez, "Optimizing convolutional neural networks for efficient weapon detection on edge devices," in *2023 IEEE CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON)*, IEEE, Dec. 2023. doi: 10.1109/chilecon60335.2023.10418646.
[22] H. Yun, "Prediction model of algal blooms using logistic regression and confusion matrix," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 3, pp. 2407-2413, Jun. 2021, doi: 10.11591/ijece.v11i3.pp2407-2413.
[23] L.-E. Pomme, R. Bourqui, R. Giot, and D. Auber, "Relative confusion matrix: efficient comparison of decision models," in *2022 26th International Conference Information Visualisation (IV)*, IEEE, Jul. 2022. doi: 10.1109/iv56949.2022.00025.
[24] L. Wang, H. Wang, T. Lu, and C. Wang, "Synchronous condenser reactive power output model based on DAG-CNN," in *2021 International Conference on Communications, Information System and Computer Engineering (CISCE)*, IEEE, May 2021. doi: 10.1109/cisce52179.2021.9445967.
[25] L.-H. Li and R. Tanone, "Ensemble learning based on CNN and transformer models for leaf diseases classification," in *2024 18th International Conference on Ubiquitous Information Management and Communication (IMCOM)*, IEEE, Jan. 2024. doi: 10.1109/imcom60618.2024.10418393.
[26] M. Poyser and T. P. Breckon, "Neural architecture search: A contemporary literature review for computer vision applications," *Pattern Recognition*, vol. 147, p. 110052, Mar. 2024, doi: 10.1016/j.patcog.2023.110052.

BIOGRAPHIES OF AUTHORS

Paula Useche Murillo is a mechatronics engineer graduated with honors in 2017 from Universidad Militar Nueva Granada Military in Bogotá, Colombia. She is a M.Sc. in mechatronics engineering and works as a research assistant in the mechatronics engineering program. She can be contacted at email: est.paula.useche@unimilitar.edu.co.

Robinson Jiménez-Moreno is an electronic engineer graduated from Universidad Distrital Francisco José de Caldas in 2002. He received a M.Sc. in engineering from Universidad Nacional de Colombia in 2012 and Ph.D. in Engineering at Universidad Distrital Francisco José de Caldas in 2018. His current working as assistant professor of Universidad Militar Nueva Granada and research focuses on the use of convolutional neural networks for object recognition and image processing for robotic applications such as human-machine interaction. He can be contacted at email: robinson.jimenez@unimilitar.edu.co.

Javier Eduardo Martínez Baquero is an electronic engineer graduated from Universidad de los Llanos in 2002. Posgraduated in electronic instrumentation from Universidad Santo Tomas in 2004, posgraduated in instrumentation and industrial control at Universidad de los Llanos in 2020 and M.Sc. in educative technology and innovative media for education at Universidad Autonoma de Bucaramanga in 2013. His current working as associated professor of Universidad de los Llanos and research focuses on instrumentation, automation, control and renewable energies. He can be contacted at email: jmartinez@unillanos.edu.co.