RESEARCH ARTICLE OPEN ACCESS

Raspberry Pi for Biometrics and Security Applications

Misael Fernando Perilla Benítez

Universidad de la Rioja
Corresponding author: Misael Fernando Perilla Benítez

ABSTRACT
This document summarizes the work and the results obtained for the master's final investigation (TFM) in which there appears the study of a variety of algorithms that are frequently used for face detection and / or facial recognition checking its performance on a computer device with limited resources and small size as is the Raspberry Pi, all with the intention of determining the results of each one, under a series of pre-established operating circumstances and the performance that each of these algorithms presents when they are executed, effectiveness, consumption, and other relevant factors to finally determine which of these is the most efficient, in front of IT solutions, which cover the needs that they present with the rise of modern society, on surveillance and security systems indoor such as homes, offices, research and development centers, just refering some. For this project were implemented the different algorithms in Phyton 3,5 language, specially for the advantages that this one offers on having been executed in this type of board. And the abundant number of aviable libraries, run on the operating system Raspbian Jessie version (4.4 with Pixel).
Index Terms: facial recognition, algorithms, biometry, raspberry, security.

Date of Submission: 01-07-2017 Date of acceptance: 14-08-2017

I. INTRODUCTON

Today, the use of electronic systems to support security and monitoring such as surveillance cameras, access controls that use biometric readers, or the use of technologies based on the reading cards or other equivalent identification devices, , just refering some, impacting on government organizations and entities, increasing the budget that these should use in the installation and maintenance of these systems, the quantity and quality of the security staff who must be trained in the effective use of this technology, as well as handling the data and information supplied by the above mentioned devices and systems, which in multiple cases, they may not meet all the needs that final user wishes, along with technological limitations they may arise.

There is a trend in information systems that is based on process automation, likewise these advances are correlated by the use of the artificial intelligence and its different ramifications, working in turn to cover the need on having used more robust devices and with a lower consumption of resources, one of the devices that currently has more implementations to cover the cited needs, is the motherboard Raspberry Pi; Thanks to its compact size and great compatibility with many other hardware devices (biometric readers between them), internet connectivity and the use of a stable and easily accessible operating system, does of this one, a suitable platform for the development of safety software, thus offering the power of a pc in the palm of the hand, at a reduced cost. One of the biggest trends in security automation, it is spearheaded by biometric personal recognition software, such as face recognition which is fed by a database, and runs on a system core based on a specific algorithm, depending on the type of data that has this system, which can be in real time (taking photographs or video), or static type (using previously captured images), there develops a software that is capable of recognizing different characteristics of a human face, and associate these traits with those previously contained in the database, thus determining the identity of a person, this without human intervention (or at least with a direct intervention on the identification process), many of these systems are used for the identification of the staff at the entrance of a building, searching for criminals by law enforcement agencies, when it is sought to verify the innocence, guilt and / or participation of the investigated in a criminal fact. There have been a number of developments and researches related to biometrics, but there is still no research that explores which of the many existing algorithms and technologies offer greater and / or better performance in this class of electronic platforms, with limited operation and reduced size, which have clear differences in their processing and storage power when compared to current computing equipment on having been compared by current computing equipment, with results that will be used for future developments of software and hardware, with results that will be used for future develop-

ments of software and hardware guided to these sections of the security based on biometric sensors assembled in a Raspberry Pi.

II. THEORETICAL FRAMEWORK

2.1 State of the art

There are multiple projects related to the implementation of Raspberry Pi, and the use of algorithms that measure different patterns and / or personal characteristics (biometric data), to identify the persons' identity, this with different intentions not exclusively tied to safety, on Internet can be found various projects and results of these, where facial recognition, voice, and fingerprints are used, nevertheless, many of these projects in many cases were developed without wishing of getting scientific information, for which the documentation is limited, the following ones are projects that account for an advance in the development of security applications using one or several aspects covered by this research.

One of the personal recognition projects with important results to this project is the development of a system of pattern recognition of the subcutaneous veins of the back – upper part – in the palm of a person's hand, with works in real time, using a classification based on collaborative representation, developed by Joardar, Chatterjee, & Rakshit [1], the purpose of this project was to develop a software system using Palm Dorsa Subcutaneous Vein Pattern (PDSVP), and verifying the effectiveness of this pattern as a reliable and measurable biometric physical feature, the system has been developed on a computer board considered as low cost, specifically the Raspberry Pi Model B +, in conjunction with an infrared sensitive camera (Raspberry Pi NoIR camera), and other electronic components, it can identifies the infrared nearby radiation (Near Infrared Radiations - NIR), And using these data appropriately, it is possible to obtain the pattern of the vascular structure present in the subcutaneous layer of the skin in the palm of the human hand, to later use this information for purposes of personal identification.

The number of projects directly related to recognition of people by means of biometrics using almost exclusively Raspberry Pi is more unusual, one of the projects that stands out is the development of a system for recognizing patterns present in fingerprints and foot in newborns, using Raspberry Pi, developed by Sivaranjani & Sumathi [2], which demonstrates the implementation of a trace extraction system using Raspberry Pi, controlled by a modified version of Debian Li-nux optimized for ARM (Raspbian) architecture, and a series of algorithms needed for the recognition of images for the measurement of biometric values, using OpenCV-2.4.9 (a series of Open Source libraries) implemented on CMake, g ++, Makefile and the extraction algorithm "Minutiae" for fingerprints, Many of the algorithms in this project are based on PCA, specially used for the plant of the feet (using OpenCV's own methods), this project has a lot to do with what was proposed in the research, Nevertheless it does not approach wear and tear of the system, as well as the existence of other algorithms that could perform better.

In general it is possible to demonstrate that there is no study exposing and comparing the consumption of computing resources that an algorithm can cover when executed directly on a Raspberry Pi 2 Model B + (or other related models), to biometric recognition through as well as subject matters are not also approached as the performance of several algorithms in this process, its utility and / or efficiency to be executed directly on ARM type hardware like the Raspberry, and the comparative one between them, which is the final aim of this work.

2.2 Objetives

General Objetive

To carry out a research on the different facial recognition algorithms and their performance on the Raspberry Pi 2 B + board.

Specífic Objetives

- To check the state of the art on previous developments of recognition projects using Raspberry Pi 2 B +.
- To investigate the different algorithms for facial recognition.
- To codify applications compatible with Raspberry Pi, based on the recognition algorithms previously collected.
- To run previously developed applications on a Raspberry Pi 2 B + and check its correct functioning.
- To perform comparative tests between the different facial recognition applications in search of relevant data to determine their performance, efficiency, effectiveness and utility.
- To evaluate the tests results, from which to generate conclusions and / or recommendations.

III. METODOLOGY

The basic methodology of this project is applied research, also known as Research and Development (R & D), which involves the creation of technological solutions that deal with obtaining, developing knowledge and capabilities whose goal is the solution of practical problems with help of the technology, it uses the results obtained from basic sciences, action – oriented skills and knowledge with practical experiences.

The aim to be achieved, through the implementation of this methodology, is the creation and / or care of potential technological benefits, or of central technological competence, that allow direct practical applications on the subject to investigate, in order to obtain more specific information and data, on which to generate hypotheses and solid results. The term of technological development is similar to the term applied sciences in the natural sciences and engineering, where it is sought to find a solution to practical problems with the help of the technique, and it is here that the methodology was developed as shown in figure 1.

Using this context, it is intended to initiate the research with a digital media search and reliable prints, on the different algorithms that are used in the development of facial recognition applications, consulting different means of investigative character Such as IEEE Xplore, Google Scholar, among others that are available on the Internet, To obtain the algorithms and, as far as possible, functional code of each of these algorithms, in the case that codes can not be executed due to language or use of libraries that are no longer supported, the corrections and / or necessary adjustments will be realized to make it functional.

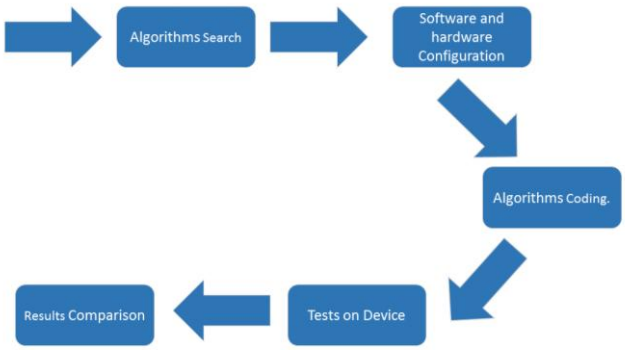

Figure 1: *Methodology Flowchart*

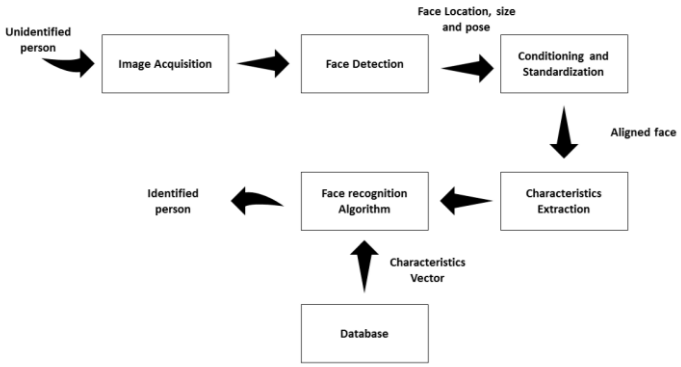

Figure 2: *Process of Faces Recognition*

Later, a Raspberry Pi 2 B + micro computer will be configured, with all the hardware and needed software, in order that the algorithms could be executed and work in a correct way, performing in this device the whole process of facial recognition:

As soon as the information and the algorithms are obtained (code or pseudo code), they will be adapted to the Python language, so they can be deployed in the hardware of the Raspberry. On this basis, we proceed to perform the corresponding tests of performance and results obtained from each algorithm, to be compared to each other and to validate this information to be presented, and it could serve as a feasibility study for future developments.

Finally, by having the performance and outcome data, a comparison of the obtained data is made, for example, the rate of false positives of identification (False Positive Identification Rate - FPIR), average execution time, among other data, to arrive at a series of conclusions allowing and / or serve as support when making decisions on the use of one or several algorithms (depending on the objectives set and / or expected results), in facial recognition projects on Raspberry Pi or other devices with similar abilities

This work seeks to create an updated state of the art to trends in computer science, technologies of information and communication, where algorithms developed for the recognition of people through face patterns, which are executed on devic-

es of limited performance or also called microcomputers, where already have relevant information And that helps to determine which are the most recommended algorithms according to the needs of the development, which would allow later projects to save time and effort in the search of an algorithm that fulfills all its objectives.

IV. IMPLEMENTED ALGORITHMS

4.1 PCA (Principal Components Analysis)

The PCA recognition algorithm is based on the technique of main component analysis, which is a process by means of which amount of data is entered there are reduced to a least amount, which are called main components, in the case of character recognition, this technique is used to identify patterns within the data obtained by the images and then to pose them in such a way that their similarities and differences can be evidenced and compared to determine the identity of a person [3].

4.2 LDA (Linear Discriminant Analysis)

Linear Discriminant Analysis is one of the most used as a data dimension reduction technique, in the pre-process for the classification of patrons and machine learning, the objective of this technique is to project a database into a dimensional space, with much less space with a good class of separability in order to avoid overfitting and overtraining in a learning algorithm, as well as to reduce computational costs [4]. The operation of LDA is very similar to PCA, but contrary to PCA which is an "unsupervised" algorithm, LDA if it is "monitored" and calculates directions (linear discriminations), which can represent the axes that maximize the separation between multiple classes.

4.3 ICA (Independent Component Analysis)

The independent component analysis is a statistical technique very similar to PCA, used to decompose a complex set of data into independent sub parts [5]. The image is transformed into a new image by the transformation of traces, which is a 2-D function dependent on the parameters (ρ, Φ), where different transformations can be produced from an image if different functions of traces are used, one of the advantages that is found in this algorithm is that it possesses a key property, and it can be used to build invariant characteristics of rotation, translation and scaling.

4.4 EBGM (Elastic Bunch Graph Matching)

EBGM is an algorithm used for the recognition of objects or classes of objects in an image, based on a representation of graphs extracted from other images. This algorithm has been especially used in the recognition and analysis of faces, but also to recognize gestures and other kinds of objects [6].

The operation of this algorithm consists of two ways to be implemented, the first way is for faces in identical poses (as similar as possible) and other one of different poses.

4.5 AAM (Active Appearance Models)

This algorithm allows to locate points in the images, to then make use of the restrictions of the models of figures, to treat interpretation as an optimization problem in which the differences between a new image and a synthesized by the appearance model [7] can be minimized.

4.6 Kernel Methods

This type or category of algorithms (since there are several This class or category of algorithms (since there are several that are considered as Kernel) are based on Mercer's theorem, which is a representation of a symmetric function defined as positive in a square, as the sum of a convergent sequence of product functions, which is a representation of a symmetric function defined as positive in a square, as the sum of a convergent sequence of product functions, when implemented, allow to obtain nonlinear versions of linear algorithms such as PCA, LDA and ICA.

For this project, the Kernel PCA algorithm was used, which is described as a method to develop a nonlinear form of the Principal Component Analysis, through the use of kernel operator functions, that make the computation process for PCA more efficient in large spaces [8], in which it is presented when calculating a point product of ($\Phi(x)$. $\Phi(y)$) it Is used a Kernel representation of the form $k(x, y)$=($\Phi(x)$.$\Phi(y)$), which allows to calculate the value of the point product in F (which is a set of characteristics of a large dimensional space) without having to use the map Φ, which reduces execution and processing times. The purpose of these nonlinear algorithms is to reduce the drawbacks that traditional algorithms have, reducing the computational cost and making them less heuristic, in addition to establishing an improved control for overtraining, which in many cases represents higher resource consumption and longer algorithm training times.

4.7 Trace Transform

The Trace Transform (transformation of traces), is a generalization of the transformation of Radon, is usually considered as a new tool for the processing of images, which can be used for object recognition by means of transformations such as rotation, translation and scaling, to produce the transformation a trace of functional lines is calculated through the lines of an image [9].

The image is transformed into a new image with the transformation of traces, which is a function in 2-D dependent parameters (ρ, Φ), different transformations can be produced from an image if

different functions of traces are used, one of the advantages that is found in this algo-tempo is that it possesses a key property, and that it can be used to construct invariant features of rotation, translation and scaling.

4.8 Viola-Jones

Viola Jones' algorithm is an algorithm that works in parallel, using the CPU and GPU of a device, allowing a faster and more efficient way to detect a face in an image. this is based on the Haar-like features and an AdaBoost cascade classifier, and is considered the first framework with face detection capability in real time, for this reason many of the applications of image processing that implement facial recognition functions that require as a data entry a face, use this algorithm and it is considered as the most common method that is used in face detection [10].

This algorithm consists of three basic steps, the first step is to use a Haar classifier to use it as a model when it comes to detecting faces, then the classifier is trained using thousands of positive and negative images that allow a much more efficient detection of faces during its execution, finally a window is displayed on the image, which shows a scaled version of it to demonstrate the different faces detected, their location and sizes.

4.9 Eigenfaces

Eigen's algorithm is based on the PCA for the reduction of the quantity of space used, in order to have a smaller amount of data easier to process, Eigenfaces' method develops the recognition of faces of the following way:

- Projecting all the samples in the PCA sub-space.
- To Project the query of the image in the PCA sub-space.
- To find the nearest neighbor between the training images and the search performed.

As soon as the reduction of the space is effected, the next step involves the generation of eigenfaces, the covariance matrix is evaluated from the image matrix by the concatenation of rows and columns, This covariance matrix evaluates the quantity with respect to its neighboring signals, there is assumed the average is zero (indicating that the image is focused) and the correlation is equal to the covariance [11].

The next step is to solve the covariance matrix, whose solution is reached, through the use of operations of basic linear matrices, the obtained results are indexed as the own values (eigenvalues). For every own value, there are an own vector (eigen-vector). These eigenvectors have been standardised, and each eigenvalue generates a face, the eigenvectors are sorted from the one containing the highest values to minors, the eigenvector associated with the highest value is which reflects the highest variance in the image.

V. TESTS

Stress tests were planned on the different algorithms to be executed on the hardware of the Raspberry Pi 2 B +, with which it is tried to verify the following aspects:

- Positive identification rate.
- Negative identification rate.
- False positive identification rate (FPIR)
- False negative identification rate (FNIR)
- Average runtime
- Algorithm performance on the hardware at Raspberry Pi 2
- Number of Kilopixels analyzed per second (Kpps).

The evaluation process for the different algorithms and their operation is the following one:

The Yale's faces database A was taken (The Yale Face Database A) [12], downloaded from the Yale University website (http://vision.ucsd.edu/content/yale-face-database), which contains 165 GIF images of 15 people, of which there are 11 images. Each of these images are arranged in the following configurations or facial expressions: centered, with glasses, happy, left oriented, no glasses, normal, right-facing, drowsy, surprised and winking. There are other databases of both Yale (Extended, B and Extended B), As of other institutions, another one of the most used is AT & T, but these surpass in size the needs of this project, since they are composed of a much larger number of images and with a higher resolution, which makes the required sub-space very wide, to this it is added the size to take up in storage device because could cause a reduction in the speed of operation at storage device.

As soon as the set of images is obtained it takes each of the applications that work based on one of the algorithm and are executed, by validating a random series of 15 images (1 per person), randomly, the different aspects mentioned above are verified and stored on the performance of the algorithm.

Once all algorithms have been tested, the process is repeated with another set of images from the Yale database, until the tests have been carried out with the whole database, it should be emphasise that the time of image capture and training of each algorithm will not be evaluated in this project, since this depends on factors such as the environment where the images were taken, as the natural lighting, time of capture and the presence of artificial lighting sources, finally it procedes to verify and compare the results obtained from each algorithm in the different aspects that were evaluated, generating comparative tables.

VI. RESULTS

6.1 Hit rate

The results obtained on having completed the tests on every algorithm are the following ones:

ALGO-RITHM	TIME (SEC)	PERFORMANCE	POSITIVES ID /TOTAL	FALSE POSITIVES	FALSE NEGATIVES
PCA	2,34	30%	165/165	0	0
LDA	2,56	35%	165/165	0	0
ICA	3,43	35%	164/165	0	1
EGBM	4	95%	163/165	0	0
AAM*	10	90%	165/165	0	0
KERNEL PCA	2	40%	165/165	0	0
TRACE TRANS-FORM**	4,67	55%	165/165	0	0
VIOLA JONES	2,74	30%	165/165	0	0
EIGEN FACES***	1	30%	154/165	2	0

Table 1: *Test Results on Algorithms*

The performance of the Raspberry exposed in the table, it is the one that the device shows in its graphical environment (taskbar), where the same operating system indicates the processes it is running and the consumption of each one of them, and the cumulative total, in this case, is used as a reference the accumulated (as seen in Illustration 3), To have clarity on the use being executed the algorithm on the processing capacity of the Raspberry, among higher value (between 0 and 100), greater is the energy consumption and the use of the processor, a performance of 36% means that the processor can still execute more processes parallel to the recognition algorithm, therefore, it would be possible to have other active services without affecting the progress and speed of calculation in the facial recognition system, moreover, a nearby or equal consumption to 100 %, is an overspending to the capacities of the Raspberry Pi plate, which could even compromise the functioning of the entire system, and in addition to this, the facial recognition process can be affected, which could even reject the use of an algorithm it will consume so much physical and energetic resources.

As can be seen in Table 1, the algorithms with the best performance in execution are PCA and deriving from it as Eigenfaces and Kernel (which for this project was used kPCA), with times that are below 2.5 seconds, while algorithms working with elastic identification of graphs and / or reference points on the face as AAM and EGBM not only consume more execution time, but also they demand much more than the device feautures, however it should take into account that these algorithms use complex mathematics designed better for video capture than for still images, and there generate a number N of reference points and graphs, which demand more calculation and processing power.

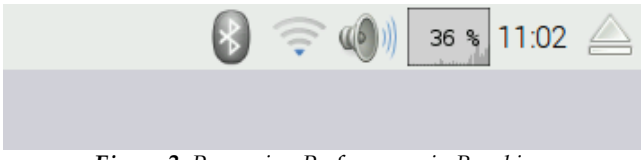

Figure 3: *Processing Performance in Raspbian*

The tests were applied on a Raspberry Pi Model B +, with only 2 devices connected, the first was a Genius USB webcam, the second was a WiFi connector, the operating system (Raspbian) was updated to its latest version (Jessie with kernel 4.4), in addition to this, the following libraries had to be installed and updated:

- NumPy (python-numpy).
- SciPy (python-scipy).
- Sphinx (python sphinx).
- OpenCV 2.4.x (Python version 2.7).
- MatPlotLib (python-matplotlib).
- SymPy (python-sympy).
- Pandas (python-pandas).
- OpenGL Support (libgl1-mesa-dri).

Something important to keep in mind is that the Rasbpian operating system, installed in the Raspberry Pi 2 B+ used for this project, does not have direct support for the generation of graphics based on OpenGL, which are widely used by many of the

algorithms in the generation of images, frames and many other used libraries (including OpenCV), and there does not exist an update in the system that enables this support, it is for this reason that the *libgl1-mesa-dri* library should be installed manually every time the operating system is updated either automatically or by command *sudo apt-get upgrade*.

6.2 Performance

The data show that the algorithm whose computer charge over run time turns out to be much more optimal (efficient) is the algorithm Eigenfaces, the most effective algorithm is PCA, which not only detected effectively all the faces, but also did in a relatively short time, but due to the existence of libraries like OpenCV.

The algorithm that is easier to implement is Viola-Jones, which is slightly slower than those mentioned above, but still their results are very positive, likewise these data are relative to the images used in the development of this project, this does not imply that other algorithms can not be used in projects, such as AAM, which has a considerably high execution time, but this one is more focused on capturing real-time video data, which would allow its use to be more related in the field of staff analysis working in an organization, and to have an input / output control, to have an example.

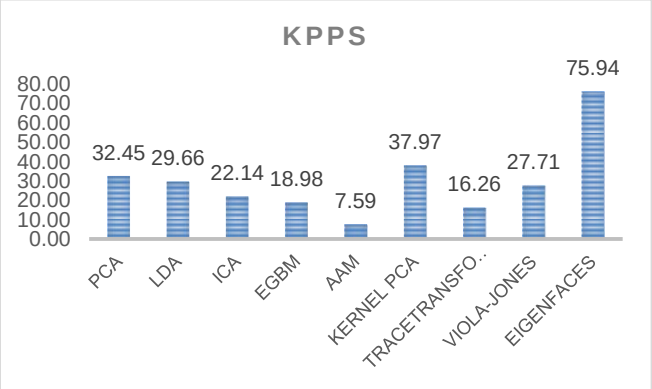

Figure 4: *Number of Kpixels / second processed by each algorithm*

In the previous table, it finds the amount of K Pixels per second (Kpps) that are processed by each of the algorithms on the images of the database, which have 75.9375 Kp (K Pixels), where as a reference it has 1 Kp corresponding to 1024 pixels, or equivalent in size to an image used in application icons installed on a PC (32x32).
Other tests performed randomly with images that do not correspond to those of the Yale Database showed the different delays that exist when performing the analysis process with higher resolution images, which implies an upper subspace, giving for example the following results using the Viola-Jones algorithm, that was one it has the best time and resource consumption:

KPPS	Time (Sec)	Size	Number of faces	Sub-space
146,16	8,21	Grande (1280x960)	3	1.228.800
159,95	4,22	Mediana (720x960)	2	691.200
195,65	4,6	Mediana (960x960)	1	921.600
27,71	2,74	Pequeña (320x243)	1	77.760

Table 2: *Results for reduction of subspaces*

As you can see in table 2, to reduce the Subspace of an image and applying the algorithm above mentioned, it drastically reduces the time required for processing, this is very important if you must work with images in real time. Finally, to the check the values obtained in the amount of Kilopixels processed by second, is can define a metric of effectiveness for each an of them implementations displayed above. It is metric will be given by the kpps product and the number of positive identifications of each algorithm. For this last parameter, the value higher retrieved is the corresponding to the algorithm Eigenface, followed of them algorithms based on PCA, however, is must have in has that the algorithm of Eigen is which greater index of false negative obtained facing its greater speed of processing, which it makes ideal for projects of recognition facial dynamic.

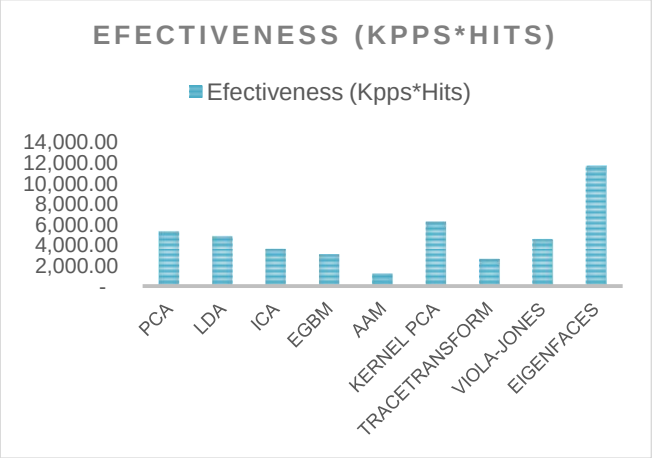

Figure 5: *Effectiveness of each algorithm*

6.3 Effectiveness

On having checked all the previous information, it can be determined that a good quality image will subsequently carry a larger size of subspace in spite of which working. Therefore, if one wishes to implement one of the algorithms with better results, It is good practice to scale the image down (apply a reduction) by adjusting it on a corrected scale to maintain a value in the sub-space that can be handled in the shortest possible time, this fact was evidenced using the images of Table 2, the Large image (of 1280x960 pixels, with a subspace of 1,228,800), was reduced to one of 960x720 pixels, decreasing its subspace to 691,200, and processing time with Viola-Jones was reduced from 8.21 seconds to 2.92 seconds, medium image was also resized from 720x960 pixels to 540x720, reducing the subspace to 388,800, and the time was reduced from 4.22 seconds to 2.02 seconds.

6.4 Access

To be able to execute the different algorithms on the Raspberry in the graphical mode of Raspbian, Without the need to have connected a screen, or in a TV with HDMI input, the use of two mass media with different intranet route, and to have access to a remote visual environment, in order to reduce energy consumption and a decrease in the amount of processing power of the board in the generation and sending of images by cable. One of the methods used was the VNC, and the other was RDP, VNC (Virtual Network Computing), it is an open source protocol, which allows to create a shared desktop system that enables to control remotely the desktop interface of one computer from another, conveying mouse and keyboard events from the controller computer, And these are received on the screen of the computer accessed remotely, while RDP (Remote Desktop Protocol) is owned by Microsoft, and originally it is intended to be implemented on Windows applications running on networked servers, however RDP was designed to support different types of network topologies and LAN protocols, When performing the tests on these two protocols, it could be noticed a better performance using RDP, as well as the installation of this protocol to be served in the Raspberry also proved to be simpler and faster.

VII. CONCLUSIONS

The results and conclusions on the performance of a variety of facial recognition algorithms obtained in this work can be used as a presider, since to date it was not possible to determine what kind of algorithms are the most efficient. For it, evaluation metrics based on the accuracy of the algorithm have been used in the performance of the implementation on the hardware and a commitment metric has been proposed that we have called effectiveness (Kpps * hits).

According to the results found, the best algorithms for facial recognition executed with Python directly on a Raspberry Pi 2 B + board are PCA and its optimized derivatives (Kernel PCA and Eigenfaces), which have short run times and a fairly high effectiveness in facial recognition on the Raspberry Pi 2 B + Hadrware, using the Raspbian operating system.

It is strongly recommended to use OpenCV in version 2.4.11 to be implemented, along with applications encoded in Python 2.7 language, as a library for support to facial recognition projects, this library has versions available for other languages, since it can be found several algorithms already present that can be used (these include PCA and Eigenfaces), without the need to develop functions from zero, which saves execution and production times of a Project.

For remote connections on the graphical desktop Raspberry device, a better connection was achieved by using the RDP protocol using connections through VCN, especially when accessing the Raspberry from a computer with a Windows operat-

ing system, although from Ubuntu also the performance of the remote desktop in RDP was better, faster and additionally, it handles better security policies with respect to other protocols.

REFERENCIAS

[1]. S. Joardar, A. Chatterjee y R. Anjan, «A Real-Time Palm Dorsa Sub-cutaneous Vein Pattern Recognition System Using Collaborative Representation-Based Classification,» April 4 2015. [Online].

[2]. S. Sivaranjani y S. Sumathi, «Implementation of Fingerprint and Newborn Footprint Feature Extraction on Raspberry Pi.,» IEEE Sponsored 2nd International Conference on Innovations in Infor-mation Embedded and Communication Systems ICIIECS' 15, 2015.

[3]. A. S. Navaz, T. D. Sri y P. Mazumder, «FACE RECOGNITION USING PRINCIPAL COMPONENT ANALYSIS AND NEURAL NETWORKS,» International Journal of Computer Networking, Wireless and Mobile Communications, pp. 245-256, 2013.

[4]. S. Raschka, «Linear Discriminant Analysis - Bit by Bit,» 3 Agosto 2014. [Online]. Available: http://sebastianraschka.com/Articles/2014_python_lda.html#lda-in-5-steps. [Accessed: July 1 2016].

[5]. A. Hyvärinen, «What is Independent Component Analysis: A Demo,» Aalto University, 5 Marzo 2013. [Online]. Available: http://research.ics.aalto.fi/ica/icademo/. [Accessed: July 3 2016].

[6]. L. Wiskott, R. P. Würtz y G. Westphal, «Elastic Bunch Graph Match-ing,» Scholarpedia, 10 Febrero 2014. [Online]. Available: http://www.scholarpedia.org/article/Elastic_Bunch_Graph_Matching. [Accessed: July 5 2016].

[7]. T. F. Cootes y C. J. Taylor, «Statistical Models of Appearance for Computer Vision,» University of Manchester, Mancheter, 2004.

[8]. B. Schölkopf, A. Smola y K. R. Müller, «Nonlinear Component Analysis as a Kernel Eigenvalue Problem,» Max-Planck-Institut für biologische Kybernetik Arbeitsgruppe Bülthoff, pp. 1-18, 1996.

[9]. S. Srisuk, K. Ratanarangsank, K. Kurutach y W. Sahatsawat, «Face Recognition using a New Texture Representation of Face Images,» [Online]. Available: http://www.face-rec.org/algorithms/Trace/eecon03.pdf. [Accessed: Juny 7 2016].

[10]. H. Sharma, S. Saurav, S. Singh, A. K. Saini y R. Sa, «Analyzing Impact of Image Scaling Algorithms on Viola-Jones Face Detection Framework,» 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1715-1718, 2015.

[11]. S. K. Shiji, «Biometric Prediction on Face Images using Eigenface Approach,» IEEE Conference on Information and Communication Technologies (ICT 2013), pp. 104-109, 2013.

[12]. P. N. Belhumeur, J. P. Hespanha y D. Kriegman, «Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection.,» IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 711-720, 1997.

[13]. D. K. Shah, V. A. Bharadi y V. J. Kau, «End-to-end Encryption based Biometric SaaS: Using Raspberry Pi as a Remote Authentication Node,» 2015 International Conference on Computing Communica-tion Control and Automation, 2015.

[14]. S. Joardar, A. Chatterjee y R. Anjan, «A Real-Time Palm Dorsa Sub-cutaneous Vein Pattern Recognition System Using Collaborative Representation-Based Classification,» April 4 2015. [Online].

Misael Fernando Perilla Benítez. Software Engineer - Universidad de Cundinamarca (2009), Specialist in university teaching - Universidad Cooperativa de Colombia (2012), Master in Computer Security - Universidad de la Rioja (2016). He worked in the virtual office of the capital district región of SENA (Servicio Nacional de Aprendizaje - National Learning Service), he has developed software for private companies, he is currently a research professor at Universidad de Cundinarmca, he was scholarship between 2003 and 2008 by the social corporation of Cundinamarca for his academic excellence.

Misael Fernando Perilla Benítez. "Raspberry Pi for Biometrics and Security Applications ." International Journal of Engineering Research and Applications (IJERA), vol. 7, no. 8, 2017, pp. 04–12.