Analysis of video surveillance images using computer vision in a controlled security environment

[1]Guillermo Casanova[0000-0001-5262-2459], [1]Daniel Yandún [0000-0002-2149-8886] and [1,2]Graciela Guerrero[0000-0002-0903-734X]
[1] Departamento de Ciencias de la Computación,
Universidad de las Fuerzas Armadas ESPE, Sangolquí, Ecuador.
[2] Departamento de Sistemas Informáticos
Universidad de Castilla - La Mancha, Albacete, Spain.
[1]{gjcasanova, deyandun2, rgguerrero}@espe.edu.ec, [2]rosagraciela.guerrero@alu.uclm.es

Abstract — **Facial analysis using video camera images proves to be a useful tool in the field of both personal and industrial security. By means of the facial analysis it is possible to develop applications based on the use of algorithms developed in Python programming languages and even with the support of Haar Cascade from the OpenCV library it is possible to solve problems caused by unexpected factors when analyzing an image. These factors can greatly influence image quality, whether it be poor lighting, shadows or even hardware problems. This document details the development of a facial recognition application for images from surveillance cameras, with the aim of providing security in a controlled environment.**

Keywords - computer vision, artificial intelligence, object detection, video surveillance, image processing.

I. INTRODUCTION

Video surveillance has had an important boom in both small and large organizations because they see it as a way of protecting themselves from criminal acts [1]. The number of videos captured by security cameras is increasing every day, which is why new techniques are required to process these data quickly and efficiently [2]. Surveillance systems analyze images in real time faster than a human. Due to this situation it is necessary to obtain help from technology that allows the processing of these images [3]. A commonly used tool is computer vision (CV) which currently has a wide application in different areas such as: video surveillance, traffic control, human-machine interaction [4].

Computer vision is a combination of search processes, detection, extraction, labeling, classification and grouping of facts, which allows to analyze moving images in real time by creating software tools based on mathematical processes. These software tools allow to process, analyze and interpret the data that are of interest in a planar image. Additionally, computer vision allows the process of inverse analysis of an image, contributing to the generation of new methods used in graphic visualization.

A video surveillance camera can capture images at speeds faster than the human eye [5]. Each image is composed of frames, which are captured every second depending on the coding standard of the device. For example, a Phase Alternating Line (PAL) uses 25 frames per second (fps) and the National Television Standard Committee (NTSC) 29.97 fps. While it is possible for the human eye to recognize only primary colors in the range of 0 to 239, based on an 8-bit image [6]. For this reason and others such as low cost, high availability and collection of evidence, it has become common to use video cameras applied to security as they replace physical actors within an organization.

Image processing using computer tools such as VC is not an easy task, because there are several factors that affect image quality such as: occlusion, low quality, lighting, shadows among others [1]. Therefore, through the use of algorithms, techniques and methods used in computer vision, important results have been obtained when analyzing objects captured by cameras in controlled environments [3]. OpenCV must be used for facial recognition because it is a well-known, precise and fast tool for any type of image. Compared to other similar ones, such as Dlib, which turns out to be much more precise than OpenCV when it has a threshold of 0.6 with an accuracy of 99.38%, but which, failing that, fails to collate images with a size smaller than 70x70.

This document aims to analyze a prototype facial recognition system using video cameras and software tools necessary for the analysis of images stored in a database through computer vision in a controlled security environment.

The work presented below is structured as follows: Section 2 establishes the work related to the analysis of facial images obtained from a digital camera. Section 3 establishes the system architecture. Section 4 establishes the design and implementation methodology for the facial image analysis prototype. Section 5 shows the findings of the researchers.

II. RELATED WORKS

Facial reconstruction by using front side images plays an important role in the area of video surveillance, this reconstruction is done through mathematical algorithms and software needed to process each frame that is captured by the security cameras. The biggest problem is that lateral facial images may exist, making it impossible to clearly observe the facial characteristics of a certain person [6]. Another way to address the drawbacks presented when analyzing images is the use of convolutional neural networks (CNN) for to reconstruct a low resolution (LR) face image before analyzing it [7].

Winkler et al. [8] propose to use local binary convolutional networks (LBCNN) in order to avoid having a very complex infrastructure and to reduce the processing time, as well as to

2020 15th Iberian Conference on Information Systems and Technologies (CISTI)
24 – 27 June 2020, Seville, Spain
ISBN: 978-989-54659-0-3

perform image pre-processing stages and image insertion in a noise environment. For the pre-processing stage, mathematical tools such as Sigmoid functions are used to increase the contrast of the image and Laplace allows the edges of the image to be delimited. Winkler et.al [8] point out that LBCNN is very susceptible to Gaussian noise, but the most robust when introducing noise into a training set.

Cárdenas et al. [1] propose a four-stage model of segmentation, identification, detection and reconstruction of objects regardless of their size. The Gunnar Farnebäck optical flow algorithm was used and to minimize errors a VIP descriptor was used to separate shadow images in conjunction with morphological operations. The Haar Cascade upperbody and Haar Cascade LRF-LBP algorithm based on the OpenCV library was chosen for the treatment of faces.

According to Winkler et al. [8], facial recognition based on video surveillance cameras has currently experienced a significant boom due to the large number of devices installed in public and private places. In view of this situation, several methods have been designed for facial recognition such as: local descriptors, local binary pattern (LBP) supported by CNN for better results. Wang et al. [2] propose an end-to-end video analysis network called MPNET. Based on convolutional algorithms by region (RCNN), this proposal allows to fully contemplate the general framework captured by the camera lens, opening the possibility of analyzing a wide range of images. A common problem with video surveillance cameras is lens curvature, which causes image distortion at these edges. In view of this situation, Ferraz et.al [9] propose a rotary cutting method to analyze images from different angles. The method proposed by Ferraz et al. [9] allowed to compare the HOG+SVM technique and RCNN, reaching the conclusion that RCNN is the best alternative when analyzing images from cameras with curved lenses. One method of analyzing still images to be considered is that proposed by Lin et al [10], which analyzes the difference between each pixel and its neighboring pixels to determine whether it is in the foreground or background. If the image is in the foreground, a shadow analysis is performed using the difference between the input image and the shade of the background image. Finally, impulse noise is eliminated by applying filters. The video image detection proposed by Hwang et al. [11], is based on the creation of a triple channel capable of modelling the spatial, temporal and spatial-temporal aspect by means of a 2D clustering method using convolutional neural networks.

III. Architecture and Methodology

This section describes the proposed architecture for the development of the video surveillance system using artificial vision. It also describes the methodology proposed to meet the objective of the subject which is to perform the analysis where the requirements of the proposal are determined, design of the solution that consists of the definition of the scenarios, the structure of the software and the resources used, development and evaluation of the functionality of the video surveillance system.

A. Architecture

The application architecture is divided into four modules (Figure 1): database, controllers, command line interface (CLI) and hardware, where:

- Database: the application requires a database in which all the face images that will be compared with each user are saved, as well as the configurations, contacts and events captured by the video cameras. The database to be used is MongoDB, which allows data to be stored in a non-relational way, thus speeding up data integration.
- Controllers: base their operation on the OpenCV library for face detection, Microsoft's Face API for image matching, and Haar Casacade that is used during the training of the algorithm for data collection. As a whole, they allow the handling of images in an effective way, especially the facial recognition that is the topic of interest of the application.
- CLI: through the command interface it is possible to make the entrance and exit of users, configuration of tools. All this is done through processes that have been previously established in the programming of software tools.
- Hardware: represents the physical aspect of the application, here are especially the video surveillance cameras. These are responsible for supplying the images to be processed by the software tools established in the control and database stage.

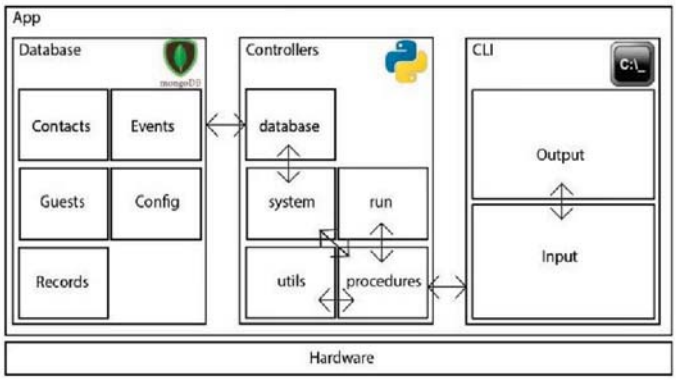

Figure 1. Architecture.

B. Methodology

The methodology proposed for the development of a facial recognition application for images obtained by means of video cameras is based on the experiences of the researchers and on the bibliographic analysis previously carried out by them.

The methodology is divided into four phases, as shown in Figure 2: analysis, design, development, testing and implementation. Each of the phases of the implemented methodology is described below.

2020 15th Iberian Conference on Information Systems and Technologies (CISTI)
24 – 27 June 2020, Seville, Spain
ISBN: 978-989-54659-0-3

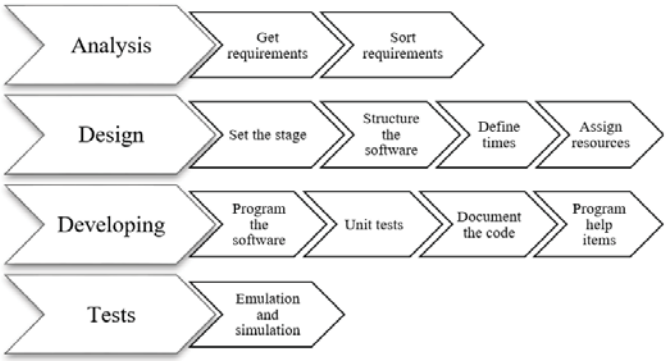

Figure 2. Methodology.

C. Analysis

At this stage all the requirements that the application needs are determined, for which tasks of obtaining and classifying requirements must be carried out (see table 1). With regard to the environment, the place of implementation of the application is determined. In our case, it will be developed in a domestic environment.

TABLE I. REQUIREMENTS

Requirements	Description
User	Command line interface Help
Environment	Home environment
Functional	Software (Python interpreter, libraries, email account and a Microsoft cognitive services subscription) Hardware (cameras)
Non-functional	Economic resources Portability Migration

D. Design

In this phase, the solution for the implementation of the facial recognition application must be put into practice. To do this, four tasks are carried out: definition of the scenario, software structure, delivery times and resources.

1) Definition of the scenario:

The place where the application will be implemented is defined, and it can be for domestic or industrial use. Three scenarios have been defined for the evaluation of the system:

- Movement detection in isolated environments, consists of checking if the system can detect any event in an environment without participants.
- Monitoring with mixed surveillance, consists of detecting the presence of authorized and unauthorized individuals in a common environment, before which the system will be able to analyze and compare the face of each individual to later send a confirmation or denial signal to the person in charge of the monitoring.
- Monitoring based on known individuals; this scenario presents authorized individuals without the presence of intruders. The system will only send confirmation signals to the person in charge of monitoring when entering or leaving the controlled environment. In general, the implementation of the facial monitoring system will depend directly on existing economic resources.

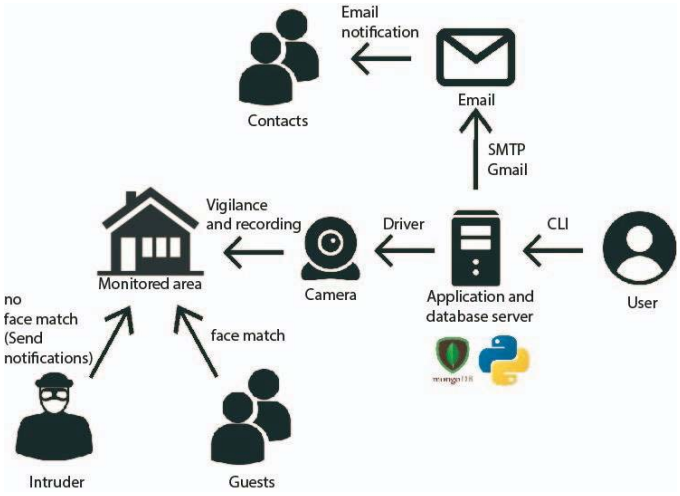

Figure 3. Application structure.

2) Software structure:

The operating structure of the application is defined, which can be seen in Figure 3. It shows the video surveillance system capturing the face of the people who wish to enter the installation, previously each user is registered by the system administrator by capturing their face and storing it in the MongoDB database and API Face. The process of creating, deleting and transferring information to the user, is done by Haar Cascade algorithms from the OpenCV library through the Python programming language. These self-learning meta-algorithms allow for the classification of positive images (with a face) and negative images (without a face) through Adaptive Boosting (Adaboost), helping to improve the performance of image classifiers. When a person wants to enter the installation, the video camera captures his face and checks the API face records for a match. If the match exists, the ID of the registered person is obtained and the information is retrieved from the MongoDB server, else an alert notification is sent through Gmail's SMTP protocol, although it can also work with other email providers such as Exchange or Yahoo to the registered users.

3) Resources:

The financial, human and technological resources necessary to carry out each planned activity are allocated.

4) System requirements:

For the system to operate correctly, the Python interpreter 3.6, the opencv-python 4.2, cognitive_face 1.5 and pymongo 3.9 libraries with their respective dependencies are required. An email account is also required, from which notifications are sent, a subscription to Microsoft cognitive services for image processing in the cloud, and an instance of Mongo DB.
As for hardware, the system requires a camera, and a computer with an Internet connection.

ISBN: 978-989-54659-0-3

E. Development

In this phase the design of the application is implemented, for which the following tasks are performed:

- Coding: with the help of the Python programming language, API Face (that are used for matching of facial images and detection of people during monitoring), and the Haar Cascade meta algorithms from the OpenCV library (that are used during training of the algorithm for data collection but do not directly intervene in image monitoring during surveillance), the respective programming for the communication between the MongoDB database and the application drivers in Python is performed. In this way, the face match can be performed for the entry of authorized users.
- Unit tests: the correct operation of the application in the selected environment is verified, observing if the results obtained are the expected ones, otherwise the necessary corrections must be made.
- Documenting the code: each change or observation made in the code is documented to have support in case of future inconveniences or updates.
- Coding help: a didactic manual of activities and installation of the application must be elaborated and if possible, it will be incorporated in the graphic interface that the user uses.

F. Functionality Tests

The purpose of the performance tests is to test the application in the previously established scenario. To do so, the following tasks are performed: Emulation and simulation: the necessary tests are performed in the established scenario, in order to register possible failures in the application. If errors are recorded, it is essential to return to the development stage and code based on the documentation established in that stage. If there are no failures, the implementation in the selected environment proceeds.

In Figure 4, the face capture is observed by means of a digital camera. Where the image box indicates the area to be captured, everything outside of it is irrelevant to the system.

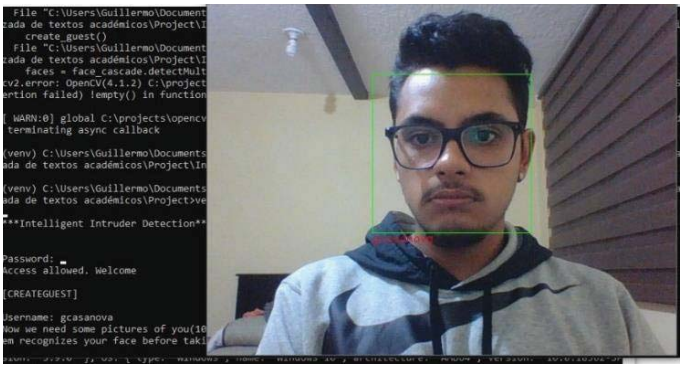

Figure 4. Facial Capture.

IV. Evaluation and Analysis of Results

This section evaluates three proposed scenarios, from each of which results are obtained and analyzed in relation to the percentage of error obtained in the evaluation of the three scenarios.

A. Evaluation

The functionality tests were performed in three previously established scenarios:

- Motion detection in isolated environments, the functionality test consisted of entering a person within the controlled environment and observing if the system could recognize the individual. In this scenario the presence of a person without restriction was detected and the conformation sent by means of an email to the person in charge of the monitoring, see figure 5.
- Monitoring with mixed surveillance, for the functionality test a population of three individuals was used, of which two were recognized persons and one intruder, obtaining as a result that there were two identified persons who entered the environment, in addition to an identified person. Subsequently, it was recorded that one identified person left the environment.
- Monitoring based on known individuals, for the test three previously registered persons were taken as reference, who entered the environment. The system verified the entry of two persons, but was not able to detect the third individual, in response to which the system sent an error message of recognition. The possible cause of the error is since the person captured by the camera was in an area with low image quality, which will be considered for future functionality tests.

Afterwards, all the participants left the environment and the event was registered by the system.

Figure 5. Motion Detection.

B. Analysis of results

A general population of 7 people was taken for the performance tests and distributed in each scenario, as shown in Table II. In the first scenario, the system responded affirmatively when a participant walked in front of the camera to demonstrate

ISBN: 978-989-54659-0-3

that the system can detect movement. For the second scenario three participants were used, two of them were individuals previously registered in the database system and the third person was considered an intruder, obtaining as a system response the satisfactory detection of both registered individuals and intruders. Finally, for the third scenario, three authorized individuals were used, but the system could only capture two of them, obtaining a failure rate of 33.33%.

TABLE II. RESULTS OF THE PROPOSED SCENARIOS

N.-	Stage	Action	Result	Users
1	Motion detection in isolated environments	Motion detection of a person in an isolated environment	The system detects the presence of one or more individuals and sends an alert	1
2	Surveillance with mixed monitoring	Intruder detection and authorized individuals in an environment with people known and unknown to the system	The system recognizes authorized individuals and detects intruders	3
3	Surveillance with monitoring based on known individuals	Intruder detection and authorized individuals in an environment with people known to the system	The system does not detect all individuals	3

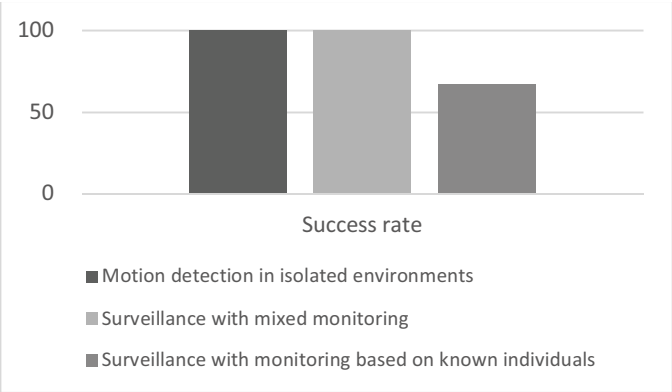

Figure 6. Motion Detection.

The failure was later controlled by sampling the environment every five seconds, which allowed the system to accurately record the entry or exit of each actor within the selected environment. Figure 6 shows the percentage of error committed in each scenario.

V. CONCLUSIONS

By completing the development of the facial image processing system, it was possible to determine the number of factors that must be considered to carry out an image analysis that is reliable and efficient. Although the OpenCV library for Python represents an aid in the development life cycle, there are still several problems to overcome. Among the problems mentioned is the high processing capacity required for the continuous analysis of images in videos according to the scale, which must be very small to give greater accuracy in the results. Also, it was determined that the lack of light and the presence of shadows strongly affect the results obtained giving false positives and negatives. In addition, the free classifiers offered by OpenCV fail if the camera angle is not correct, which is why the pre-processing of each frame is necessary before being analyzed, and it may even be necessary to experiment with classifiers specifically trained for the cases that the proposed software seeks to meet.

The use of cloud services such as Azure Cognitive Services can be a solution to the problem of real-time facial recognition, since they are highly trained and therefore offer greater accuracy compared to local models and classifiers. However, it is not advisable to do all the work in the cloud, but only the operations that require greater accuracy, due to latency and network traffic, in addition to the cost of the service.

VI. FUTURE WORKS

As a follow-up to the present work related to facial recognition, some lines of research have been left open that may be resolved in the future. Throughout the development of the facial recognition system, some issues arose that are directly related to the development of the application, such as the type of algorithm responsible for making the comparison between faces captured in real time and those previously registered in the MongoDB database. But additionally, there were factors that influenced indirectly in the development of the application and that can be used by other researchers in the approach of new research ideas. Here are performance factors of the selected algorithm for facial recognition.

Below are some future works that can be developed by the researchers that were not covered in this document because they are not considered part of the research topic.

Among the possible future works are: i) Developing a facial recognition algorithm that allows minimizing the effects of shadows caused by the lack of illumination at the time of capturing the image to be analyzed, in the same way the algorithm should be able to minimize as much as possible the errors due to false positives. ii) Implementation of facial recognition of emotions of people entering a controlled environment, in order to know their reaction to various events. Making it possible for the system to be integrated into a service store, in order to know the trends of its customers based on a facial expression of the face. iii) Improve the usability of the system by implementing a more user-friendly interface. iv) Enable the customization of certain software settings (for example: change the default action when detecting an intruder or modify the number of verifications prior to sending an alert) so that the system can adapt to different scenarios in a better way. v) More rigorous testing and more participants in each of the scenarios, to test the operation of the system in environments that have larger group of people.

REFERENCES

[1] Cárdenas T, R. J., Castañón, C. A. B., & Cáceres, J. C. G. (2018, March). Face Detection on real Low Resolution Surveillance Videos. In *Proceedings of the 2nd International Conference on Compute and Data Analysis* (pp. 52-59). ACM.

ISBN: 978-989-54659-0-3

[2] Wang, H., Wang, P., & Qian, X. (2018). MPNET: An End-to-End Deep Neural Network for Object Detection in Surveillance Video. *IEEE Access*, *6*, 30296-30308.

[3] Moctezuma, D., Téllez, E. S., Miranda-Jiménez, S., & Graff, M. (2019). Appearance model update based on online learning and soft-biometrics traits for people re-identification in multi-camera environments. *IET Image Processing*, *13*(12), 2162-2168.

[4] Olague, G., Hernández, D. E., Clemente, E., & Chan-Ley, M. (2018). Evolving head tracking routines with brain programming. *IEEE Access*, *6*, 26254-26270.

[5] Ruiz, D. G., Térmens, M., & Ribera, M. (2012). Modelo de indicadores para evaluar los formatos digitales para la preservación de vídeo. *Revista española de documentación científica*, *35*(2), 281-297.

[6] Llanga-Vargas, A., Santillán-Lima, J., Rocha-Jacome, C., & GuerreroMorejón, K. (2018). Visión Artificial en la detección de la pupila del ojo humano para el control motriz de una silla de ruedas. *Revista de Investigación Talentos*, *2018*, 679-685.

[7] Nozawa, N., Kuwahara, D., & Morishima, S. (2015, July). 3D face reconstruction from a single non-frontal face image. In *ACM SIGGRAPH 2015 Posters* (p. 57). ACM.

[8] Winkler, R., Qu, C., Voth, S., & Beyerer, J. (2018, December). 3D Face Reconstruction from Low-Resolution Images with Convolutional Neural Networks. In *Proceedings of the 2018 the 2nd International Conference on Video and Image Processing* (pp. 83-88). ACM.

[9] Ferraz, C. T., & Saito, J. H. (2018, October). A comprehensive analysis of Local Binary Convolutional Neural Network for fast face recognition in surveillance video. In *Proceedings of the 24th Brazilian Symposium on Multimedia and the Web* (pp. 265-268). ACM.

[10] Lin, H., Kong, Z., Wang, W., Liang, K., & Chen, J. (2018, November). Pedestrian Detection in Fish-eye Images using Deep Learning: Combine Faster R-CNN with an effective Cutting Method. In *Proceedings of the 2018 International Conference on Signal Processing and Machine Learning* (pp. 55-59). ACM.

[11] Hwang, S., Uh, Y., Ki, M., Lim, K., Park, D., & Byun, H. (2017, January). Real-time background subtraction based on GPGPU for high-resolution video surveillance. In *Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication* (p. 109). ACM.

[12] Li, Y., Ge, R., Ji, Y., Gong, S., & Liu, C. (2017). Trajectory-pooled spatial-temporal architecture of deep convolutional neural networks for video event detection. *IEEE Transactions on Circuits and Systems for Video Technology*.

2020 15th Iberian Conference on Information Systems and Technologies (CISTI)
24 – 27 June 2020, Seville, Spain
ISBN: 978-989-54659-0-3