Edge Face Recognition System Based on One-Shot Augmented Learning

Diego M. Jiménez-Bravo[1,2]*, Álvaro Lozano Murciego[1], André Sales Mendes[1], Luis Augusto Silva[1], Daniel H. De La Iglesia[1,3]

[1] Expert Systems and Applications Lab, Faculty of Science, University of Salamanca, Plaza de los Caídos s/n, 37008 Salamanca (Spain)
[2] Ontology Engineering Group, Departamento de Inteligencia Artificial, ETSI Informáticos, Universidad Politécnica de Madrid, 28660 Madrid (Spain)
[3] Faculty of Informatics, Pontifical University of Salamanca, 37002 Salamanca (Spain)

Received 6 April 2022 | Accepted 4 July 2022 | Early Access 12 September 2022

Abstract

There is growing concern among users of computer systems about how their data is handled. In this sense, IT (Information Technology) professionals are not unaware of this problem and are looking for solutions to meet the requirements and concerns of their users. During the last few years, various techniques and technologies have emerged that allow us to answer to the problem posed by users. Technologies such as edge computing and techniques such as one-shot learning and data augmentation enable progress in this regard. Thus, in this article, we propose the creation of a system that makes use of these techniques and technologies to solve the problem of face recognition and form a low-cost security system. The results obtained show that the combination of these techniques is effective in most of the face detection algorithms and allows an effective solution to the problem raised.

Keywords

Data Augmentation, Edge Artificial Intelligence, Edge Computing, Face Recognition, One-Shot Learning, Security System.

DOI: 10.9781/ijimai.2022.09.001

I. Introduction

Nowadays, we live in a society governed by technology and its advances, which generally aim to help society and make its day-to-day life easier and simpler. Large technology companies and in many cases public institutions are participants in these advances and put them into practice to obtain a business benefit in many cases. Society is aware of this benefit that companies and institutions can obtain and sometimes show rejection or mistrust of these innovations. Such systems can often reuse the information obtained from the user for other purposes, although this is not always the case. Given these circumstances, it is important to inform and educate users about these new technologies and how their information can be processed. Hence, this is an agreement in which companies and institutions have to participate with their commitment to being more transparent and, at the same time, users with the pro-activity to learn and understand the new systems.

With this in mind, one of the fields that most concerns society is the field of Artificial Intelligence (AI). In recent years, the advances developed in this field have been tremendous and its applications extend to practically all fields of society and in almost all areas of knowledge. Therefore, it is understandable that these systems, and especially the information they use and collect to provide results and benefits, can be a source of concern to users. In particular, when the data collected is related to them and can be used for other purposes or even sold to third parties. This collected data is commonly processed and stored in cloud systems. This is also where there is fear on behalf of users as they lose the perception of where their data is located.

Progress in computer science has led to the emergence of new computing techniques which are closer to the user, such as Fog and Edge computing. These techniques significantly reduce the amount of data sent over the Internet. However, the complexity of many AI systems and the number of resources they require make it impossible for such systems to run on devices with these computing architectures. Consequently, at present, it appears necessary to send data over the Internet to cloud services to obtain accurate and fast results in AI models with high computational needs.

However, currently, some techniques allow to reduce the amount of data sent through the Internet and they may reduce the amount of information obtained from the user or his environment (however, they still can collect huge amount of data from the users and theirs environments). However, these techniques, in which less data is used to train AI models, often have certain disadvantages associated with them, such as a decrease in the efficiency of the algorithms. Indeed, this makes perfect sense since complex models need large amounts of data to get good results, and if this data is reduced, the efficiency of the systems is reduced accordingly. However, it is important to note that despite this lack of data, many systems have been able to achieve high efficiency.

Unfortunately, this is not always the case; in fact, the vast majority

* Corresponding author.

E-mail address: dmjimenez@usal.es

of systems reduce their effectiveness. As a result, it is necessary to implement techniques that increase effectiveness despite the lack of data or that increase effectiveness even more despite a large amount of data. These techniques are known as data augmentation, which generates new data by applying slight modifications to the original data. In this way, the combination of data augmentation with the limited amount of user data can allow systems to achieve the minimum required performance and at the same time minimise the use of user data and/or information.

In this way, this paper presents the research developed by combining one-shot learning and data augmentation techniques to evaluate their effectiveness in systems that use fewer user data and also avoid sending data over the Internet. In this regard, the aim is to develop an edge computing system that makes use of an AI model developed through a combination of the two previous techniques for the intelligent recognition of different people.

A. Background

In this subsection, we analyze and explain the main concepts and techniques related to the areas of our research. Therefore, brief descriptions of each of the areas are included along with some of the most recent studies.

1. One-Shot Learning

The one-shot learning technique was first introduced in 2000 by Miller et al. [1] It is a subtype of supervised learning; as is well known, supervised learning uses labelled data or examples to learn and obtain knowledge and generate a model. Generally speaking, these types of training and models require a large amount of data to obtain good results. Nevertheless, the one-shot learning technique aims to do the opposite, with only one or a few samples it can train and generate an AI model [2]. However, this lack of data often has consequences in terms of model and results' efficiency.

An area where this technique has particular application is in the field of computer vision, where datasets with a large variety of samples are often not available. On the other hand, this area is one of the major applications of deep learning models; as is well known, these models require huge amounts of data to obtain highly efficient models. However, advances in science and research have led to the emergence of techniques such as transfer learning that allow the use of one-shot learning and obtain outstanding results.

The one-shot learning approach has been used in recent studies. Vinyals et al. [3] use it to increase the efficiency of other one-shot approaches on Omniglot and ImageNet datasets; specifically, they use matching nets to facilitate fast learning from a few labeled samples and classify images into the corresponding class. Another application is the use of one-shot learning for image segmentation [4]; they use a few labelled samples to extract the area of the image where the desired objects are located. This solution increases the efficiency of other similar investigations. Others, Woodward and Finn [5], have combined reinforcement learning and one-shot learning to increase the efficiency of pure supervised systems. They allow the systems to determine when is worth doing the classification or pay a penalty to receive the correct label. Another interesting approach is developed by Wang et al. [6] they combine the features extracted from the image along with the features located in the embedding of the class name. They mix these two pieces of information to map images and labels to predict unlabelled images. This approach obtains better performance than the baselines used in the research.

Moreover, one-shot learning techniques can also be applied to other areas. For example, [3] applies one-shot learning to text processing tasks, although not very successfully. In contrast, [7] applies this technique to drug discovery using a combination of LSTM (Long Short-Term Memory) and graph convolutional neural networks and a small number of samples.

2. Data Augmentation

As mentioned above, many models do not have the appropriate amount of data to train and implement a quality model. Furthermore, given the characteristics of these models, they are not able to obtain the insights needed from techniques such as one-shot learning. Thus, it is necessary to look for a solution capable of handling these two challenges. This solution is called data augmentation.

The main objective of data augmentation techniques is to increase the size and quality of the datasets. Like the previous technique, it is extremely related to the field of computer vision and imaging. Thus, image augmentation techniques [8] are the following: geometric transformations, color space augmentations, kernel filters, mixing images, random erasing, feature space augmentation, adversarial training, Generative Adversarial Networks (GAN), neural style transfer, and meta-learning. Generally speaking, these techniques can be classified into two main groups, (I) the most basic techniques, and (II) techniques based on deep learning solutions, especially those based on GAN architectures.

Besides, other more recent data augmentation techniques have emerged recently. A study developed by Zhong et al. [9] proposed to select a rectangle area of an image and erase its pixels with random values. This technique is called Random Erasing. On the other hand, Perez and Wang [10] proposed neural augmentation; this technique is based on a neural network that learns which data augmentation technique to use to maximise the performance of the system.

It can be observed that new data augmentation techniques are being proposed; in fact, new techniques are being proposed even outside the area of computer vision. This is the case of the study developed by Park et al. [11]; the researchers have developed SpecAugment, an augmentation method for automatic speech recognition.

3. Face Recognition

One of the most active areas of research in recent years in computer science is object recognition; one of its subfields is face recognition. This is a set of techniques that aim to identify faces within a set of images. The task of face recognition involves the development of other subtasks in a pipeline necessary for the correct performance of face recognition. These subtasks are listed and explained below:

1. **Face detection**: it determines the position and size of a human face in a digital image [12]. During the years several approaches have been presented, like Viola and Jones' Haar cascade method [13] to find face features with Haar-like features. A similar approach is Dlib HOG [14] which uses Histogram of Oriented Gradients (HOG) in the combination of Support Vector Machines (SVM) to detect faces. Dlib CNN [14] uses the power of Convolutional Neural Networks (CNN) to extract the features in faces; this is combined with Maximum-Margin Object Detector to maximize the results. Also, another well-known method is the SSD-Resnet [15], another CNN method to detect objects in digital images. More recent approaches like Multi-task Cascaded Convolutional Networks (MTCNN) [16] are also based on deep learning. MTCNN uses three subnetworks, P-Net, R-Net, and O-Net. These subnets are responsible for producing proposed regions, refining the proposals, and do face landmarking. Similarly, FaceNet [17] follows a process and uses CNN to detect a face on images.
2. **Face alignment**: it is the process in which the face landmark is usually rotated to obtain a face perfect alignment in case the face was originally rotated. Through the years several methods have been used to resolve this problem, ASMs [18], AAMs [19], CLMs [20], and cascade regression models [21]. Nonetheless, more recent

methods based on CNNs have been presented. These methods are divided into two main categories, coordinate regression models and heatmap regression models. A few examples of the first category are [22], [23]. On the other hand, examples of the second category are [24]–[26].

3. **Face encoder**: it is the process in which a face is transformed into an array representation. This array contains the features of the face and it is a numeric representation of it. This array of features is what usually is stored in a system that makes use of face recognition with its users. There are several techniques normally used to obtain the face features, such as LBPH (Local Binary Patterns Histogram) [27]; this technique divides the face into different blocks and obtains a histogram for each block, after that, it combines the histograms in only one. Another solution is OpenFace [28], a toolkit for face recognition. Also, ResNet [29] improves the encoding process as well as the whole process. Moreover, FaceNet [30] can also accomplish the encoding process.
4. **Face classifier**: this is the last stage of the process and consists of classifying the encoders from the previous phase into one of the classes provided during the classifier training or as an unknown person. Supervised learning algorithms are generally used. Thus, at this last stage, we obtain the results of the process, a label, and the confidence of the label.

This process is followed in recent studies using face recognition for different purposes. Such is the case of the [31]–[34] studies, among others.

4. Edge Computing

According to the European Edge Computing Consortium (EECC) edge computing is a computing paradigm in which certain services run near or on the very devices that request them [35]. Among other advantages, this type of computing increases privacy and decreases network latency. However, let us briefly describe some of the main features of edge computing according to the EECC [35].

- **Security**: security is an important aspect of all computing systems and it is important to ensure security between communications in an edge network. In terms of privacy, edge computing-based systems increase the privacy of data as all data is processed on these devices and the ownership of the data is kept between the owners of the edge devices.
- **Real-time**: the responses of services hosted on edge systems generally offer a much faster response and decrease the waiting time for users. This is mainly caused by lower data traffic between devices.
- **Acceleration**: resource-intensive processes that require large amounts of resources to respond to users, such as AI processes, can speed up their responses by finding computing centers closer to the user.
- **Management**: another important feature is the management that can be performed over edge computing networks. Firstly, its architecture allows it to be fail-stable and new devices can be added easily and with little or no change to the network topology. And another fundamental advantage is that by not depending on external services, it is possible to have a great deal of reliability in its operation.

The edge computing paradigm is widely used, proof of which are the studies carried out in recent years in areas such as autonomous vehicles [36], smart cities [37], smart homes [38], and even security systems such as the one proposed by Dirgantoro et al. [39] which proposes a security system based on recognition through Generative Adversarial Networks.

Taking all this into account, this article proposes the use of one-shot learning, data paradigm, face recognition, and edge computing paradigm to create a security system that is as transparent as possible for the end-user and based on edge devices.

The rest of the article is structured as follows: section II describes the proposed system, materials and methods of this study; section III shows the results obtained in the research; section IV explains the main ideas obtained from the results analysis; and finally, section V talks about the study conclusions and future lines of research.

II. Methods

In this section, we will describe the main features of the system proposed in this article as well as its software and hardware characteristics. The section is divided into three subsections.

A. Proposed System

The proposed system is based on the four concepts explained in the subsection A. To explain in a more detailed and precise way the proposed system in the following subsections, each of its modules is illustrated in Fig. 1 providing an overall view of the system.

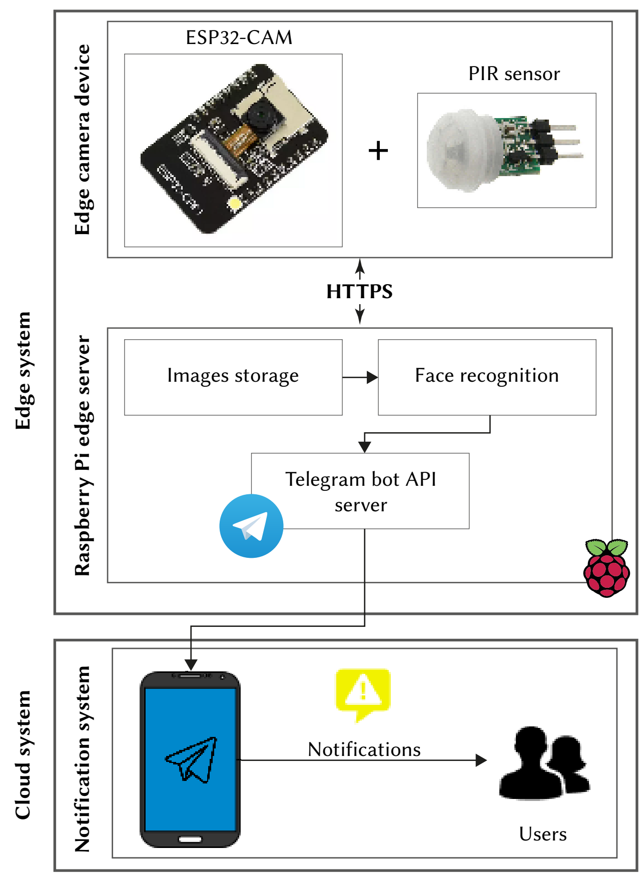

Fig. 1. Proposed system.

1. Edge Camera Device

As shown in Fig. 1, the system is composed of three main modules. The first one is the "Edge camera device", which is a hardware edge device that will take images whenever it detects movement in its proximity. Details of the specific hardware that composes this device will be explained in detail in 2. In addition, while the device detects movement, the system will take pictures every second to be able to detect in some of these images the faces of the people who have caused the camera to wake up. In this way, this device will act as a mini edge security camera that will send the images obtained to another edge device.

2. Raspberry Pi Edge Server

The second module is the "Raspberry Pi edge server" which is formed by a Raspberry Pi (we will describe its features in the next section) that will act as a mini server to store both the images received by the edge camera device, the facial recognition model, and the Telegram bot in charge of notifying the user. The images are sent through the HTTPS protocol within a local network, so the images and the system aim to work as close as possible to the user and on devices that the user can have at home. On the other hand, the facial recognition model is trained from a single image of the users of the system using the one-shot augmented learning technique. In this way, the proposed system not only reduces the transit of user images over the Internet by using a local network but also reduces the number of user images required for the system to work. Furthermore, the models used to implement this model are proposed based on the study we implemented in this work where all possible combinations with the most relevant models for face recognition are studied (this part of the study is detailed in more detail in the next paragraph). Finally, this second module will also contain a space to store the API (Application Programming Interface) of the Telegram bot configured for our system. The bot will be in charge of using the facial recognition model to identify the faces in the images and program notifications to the Telegram account of the users when unknown people are identified in the images.

Face Recognition Study The facial recognition study was based on the techniques explained in A. With this study, we intend to test the effectiveness of the various existing models for face recognition with one-shot learning and one-shot augmented learning techniques. To carry out this study we propose different stages that allow us to obtain comparative results between the different methods. In Fig. 2 we can see a scheme of each of these stages and therefore of the general behaviour of this study. Likewise, this scheme is explained in detail below.

- **Data augmentation**: one of the objectives of the research is to compare the efficiency of one-shot learning and
- one-shot augmented learning techniques. To do so, we need to build two datasets for each of these techniques. The datasets are obtained from an original dataset (1) that is formed by folders associated with each of them to a person. To generate the one-shot augmented dataset, an image is selected and 100 new images are randomly generated from the selected one.
- The new images are generated by randomly modifying a series of variables: width and height shift range, brightness, zoom range, and rotation range.
- **Training datasets**: after the data augmentation process two datasets are obtained, the one-shot dataset obtained
- with the selected image and the one-shot augmented dataset obtained from the images generated from the image selected for the one-shot dataset.
- **Testing dataset**: the test set has the same structure as the previous datasets and contains the rest of the images not
- selected for the training sets. This set is used to test the resulting models after the training phase.
- **Face recognition**: the next step is to evaluate the techniques with each of the generated datasets. Therefore, a process is generated for the one-shot learning technique and another one for the one-shot augmented learning technique. For each, different combinations of models are tested; in particular, face detector models, face encoder models, and different classification algorithms are combined. In this way, it is possible to evaluate

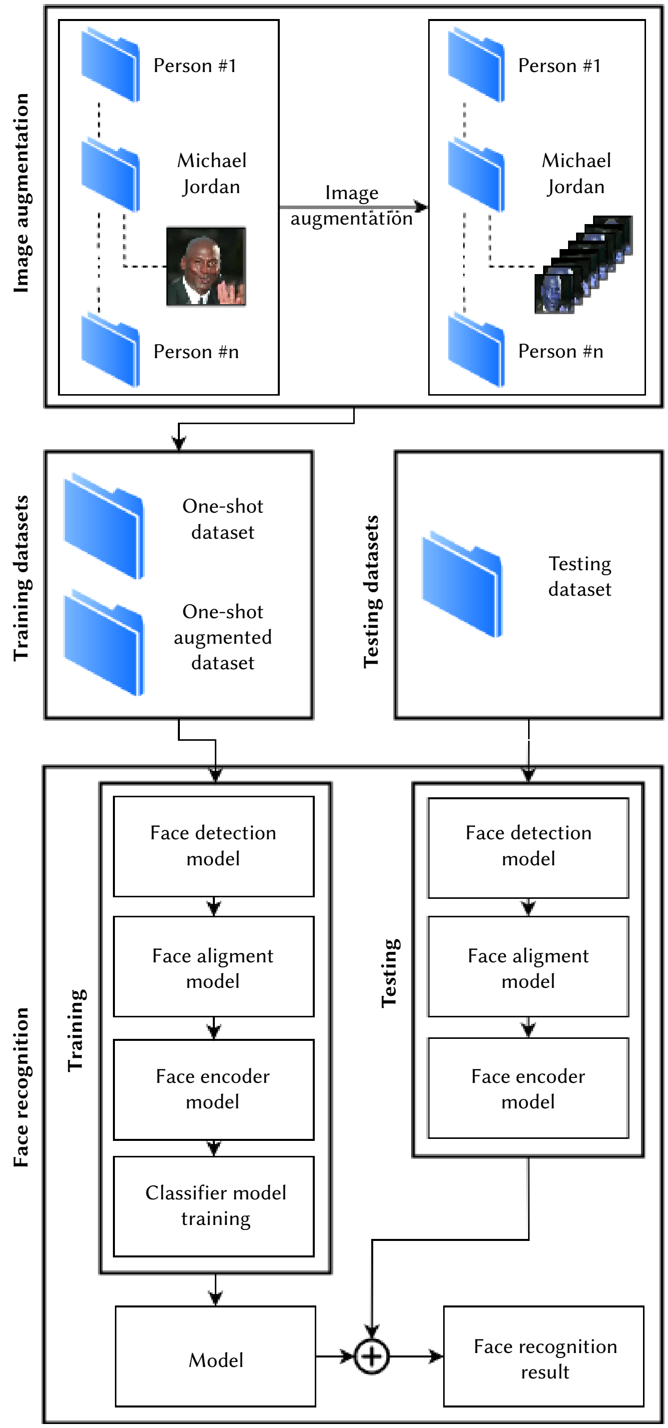

Fig. 2. Face recognition study proposal.

which combination of models works best with which dataset. The face detection models used are Haar Cascade (HC), Dlib HOG, Dlib CNN, SSD-Resnet, MTCNN, and FaceNet. On the other hand, the encoder models used are OpenFace, Dlib ResNet, and FaceNet. Finally, the classification models are as follows: Naive Bayes (NB), Linear Kernel SVM (LKSVM), Radial Basis Function (RBF) kernel SVM, k-Nearest Neighbours (k-NN), Decision Tree (DT), Random Forest (RF), Neural Network (NN), AdaBoost (AB), and QuaDrAtic (QDA) classifier.

After the training process, the resulting models are evaluated using the test set. The metrics used to evaluate these methods are explained in 2.

This process is repeated for different configurations of the datasets in which the number of classes and therefore of people vary. The purpose of this is to see how the effectiveness of both techniques also varies according to the number of classes.

3. Notification System

The last module, "Notification system", is responsible for receiving notifications received by the Telegram bot on a device compatible with the Telegram application. In this way, the user can consult them whenever he/she wishes, together with the image in which an unknown individual/s has/have been detected by the edge security system proposed in this article.

B. Materials

In this subsection, we will explain the dataset and hardware used during the course of this research.

1. Datasets

For the comparative study of one-shot learning and one-shot augmented learning techniques, a well-known dataset prepared for facial recognition models has been used. This dataset is called The labelled Faces in the Wild (LFW) dataset [40]. The dataset contains an average of 2.03 images per person, which is a total of 13,233 images for a total of 5,749 people. The internal structure of the dataset is divided into folders, one for each of the persons included in the dataset. Within these folders, we will find at least one image of the person to whom the folder refers. These images have a series of characteristics in common, all of them have a size of 250 x 250 pixels and are in JPG digital format.

According to [40] each of the images, belonging to the LFW dataset, are obtained employing the Haar cascade method proposed by [13] and increasing the detection area by a factor of 2.2 in each of its dimensions. This is done to obtain a larger viewing area than that provided by the Haar cascade. In this way, each image contains only one face per image allowing us to better evaluate one-shot learning and one-shot augmented learning techniques.

2. Hardware

The hardware used for the development of the prototype used in this article is detailed below.

- **ESP32-CAM**: this is one of the main components of the proposed system. The ESP32-CAM (embedded microcontroller, Espressif Systems, Shanghai, People's Republic of China) is an embedded microcontroller that operates independently. It has WiFi and Bluetooth connectivity. It also has a small integrated video camera and a MicroSD slot. Some other features of this microcontroller are as follows:
  - Connectivity: WiFi 802.11b/g/n, and Bluetooth 4.2 with BLE. Supports image upload over WiFi.
  - Connections: UART, SPI, I2C, and PWM. It has 9 GPIO pins.
  - Clock frequency: up to 160Mhz.
  - Microcontroller computational power: up to 600 DMIPS.
  - Memory: 520KB SRAM, 4MB PSRAM, and SD card slot.
  - Extras: has multiple sleep modes, firmware upgradeable by OTA, and LED for flash memory built-in.
  - Camera: supports OV2640 cameras that can be bundled or purchased separately. This type of camera has:
    - 2 MP sensor.
    - UXGA array size of 1622×1200 px.
    - Output format YUV422, YUV420, RGB565, RGB555 and 8-bit data compression.
    - It can transfer images between 15 and 60 FPS.
- **PIR sensor AM312**: it is a Passive Infrared Sensor (PIR sensor, ARCELI) that can detect the presence of moving objects in its area of action. Some of its characteristics are the following:
  - Working voltage: DC 2.7-12V.
  - Static power consumption: <0.1mA.
  - Delay time: 2 seconds.
  - Blocking time: 2 seconds.
  - Trigger: repeatable.
  - Detection range: cone angle of ď100 degrees, 3-5 m (required depending on the lens).
  - Working temperature: -20 to + 60.
  - Size PCB: 10 mm x 8 mm.
  - Overall size: Approx. 12 mm x 25 mm.
  - Lens Module: Small lens.
- **10k ohm electrical resistor**: 10k ohm electrical resistor (electrical resistor, AZ-Delivery Vertriebs GmbH, Deggendorf, Germany).
- **1k ohm resistor**: a 1k ohm electrical resistor (electrical resistor, AZ-Delivery Vertriebs GmbH, Deggendorf, Germany).
- **Transistor 2N3904**: a transistor (transistor, BOJACK, Guangdong, People's Republic of China) in charge of amplifying the signal coming from the PIR sensor AM312 and transmitting it to the ESP32-CAM.
- **Raspberry Pi 4 Model B**: the Raspberry Pi (computer board, Raspberry Pi, Cambridge, United Kingdom) will serve as a local server to host both the images transmitted by the ESP32-CAM and the Telegram bot that will be in charge of notifying the user. The technical characteristics of the model used are as follows:
  - RAM: 8GB.
  - Type of RAM: DDR3 SDRAM.
  - Operating system: Raspbian OS.
  - Processor: A-Series Dual-Core A4-3305.
  - Hardware interface: Bluetooth 5.0, WiFi 802.11b/g/n, Ethernet, Micro-HDMI, USB-C, USB 3.0, and USB 2.0.
  - Graphics card: Radeon Vega 8.
  - Graphics memory type: DDR4 SDRAM.
  - Graphics card interface: PCI-Express x4.
  - Voltage: 5 Volts.

C. Methods

In this subsection, we will explain all the methods and processes that have been followed during this research in such a way that the whole process is reproducible.

1. Materials Manipulation

Generation of Datasets and Subsets Generation of datasets and subsets: The dataset used for this research has been explained above (1). However, this dataset is modified to obtain two different datasets, one for the evaluation of the one-shot augmented dataset and the other for the evaluation of the one-shot dataset. Thus, the process to obtain these two datasets is explained below.

Most of the folders (people) of the LFW dataset contain only one image; in our experiment, these folders are not taken into account for the evaluation of the methods, since no different image would be available for the testing process. Therefore only folders containing more than one image will be taken into account in our experiments. From these folders we select a random image that will form the training subset for the one-shot dataset; on the other hand, the rest of the images in this folder will form the test subset for the one-shot dataset. Once the one-shot learning dataset is formed, the process explained in 2 can be followed to generate the one-shot augmented learning dataset. To obtain the augmented images of the one-shot augmented learning dataset, the parameters and intervals described in Table I have been used.

TABLE I. Parameters Used for Data Augmentation

Parameter	Interval/value
Width shift range	[-1, 1]
Height shift range	[-1, 1]
Brightness range	[0.7, 1.3]
Zoom range	[-9, 11]
Rotation range	0.4

This methodology allows us to obtain two different datasets originating from the same input, the one-shot dataset and the one-shot augmented dataset. However, to evaluate the effectiveness of both techniques, several datasets have been configured for one-shot learning and one-shot augmented learning. The datasets are obtained in the same way but differ in the number of classes/people. As the number of classes decreases, the number of images available for each person is considered; in such a way that priority is always given to those that have more images that can be used for the test stage. As a result, pairs of datasets (one-shot and one-shot augmented) with 143, 85, 57, 41, and 19 classes are formed.

Edge Hardware Configuration The hardware components described in 2 constitute the different subsystems of the proposed system in A. In this way, we are going to explain how to configure each of these subsystems.

- **Edge camera device**: it consists of an ESP32-CAM, an AM312 PIR sensor, a 10k ohm electrical resistor, a 1k ohm electrical resistor, and a 2N3904 transistor. The electronic schematic of this subsystem is shown in Fig. 3. Also, keep in mind that the ESP32-CAM will remain asleep until the PIR sensor detects motion, and then the ESP32-CAM will wake up and take the necessary pictures that will be sent to the Raspberry Pi edge server.
- **Raspberry Pi edge server**: in this case, the system consists of a Raspberry Pi 4 Model B with Raspbian OS. In this device, an HTTP server has been configured to receive and store the images sent by the edge camera device. In addition, this device is responsible for processing these images with a facial recognition model, which we will specify in later sections, and will send the results through a Telegram bot to warn the user of possible intrusions.

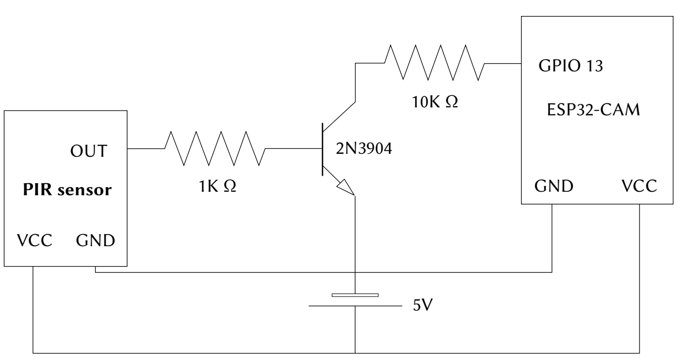

Fig. 3. Electronic scheme of the edge camera device.

2. Metrics

As explained above, the comparative study between one-shot learning and one-shot augmented learning techniques aims to obtain the best combination of models and therefore to establish the models to be used in the face recognition stage of the proposed system. However, the evaluation and comparison must be carried out taking into account some quality metrics. In this study, the accuracy metric described in (1) has been used for the evaluation:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

where TP (True Positive) are the instances correctly classified as a positive class, TN (True Negative) are the instances correctly classified as negative instances, FP (False Positive) are the instances incorrectly classified as positive, and FN (False Negative) are the instances incorrectly classified as negative class.

III. Results

This section will explain the results of the different sub- systems or studies of the proposed system. This section will analyze and explain these results in an objective way.

A. Face Recognition Study Results

When analyzing the results of the face recognition study, different situations are observed, which we will show in this subsection. The totality of the results of each of the experiments can be seen in Table III which is included in the annex A.

Analyzing the results obtained and shown in Table III it can be observed that the one-shot augmented learning technique increases the efficiency of the combinations of the different algorithms concerning the one-shot learning. In the experiment with 143 classes, the one-shot augmented learning increases the results in 99 out of 162 algorithm combinations; in the experiment with 85 classes there are also 99 out of 162 algorithm combinations that improve their accuracy with the one-shot augmented learning technique; the experiment with 57 classes obtains better results in 95 algorithm combinations using the one-shot augmented learning; with 41 classes it improves in 94 algorithm combinations; while in the case study with 19 classes 110 combinations out of 162 increase their accuracy with the one-shot augmented learning technique.

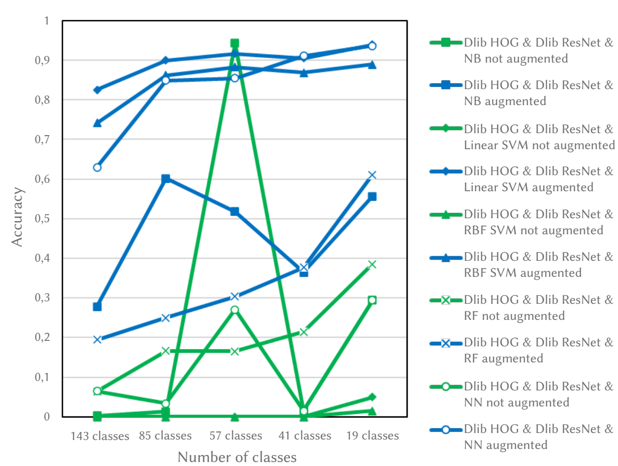

Fig. 4. One-shot augmented learning vs. one-shot learning.

However, it is important to mention that this increase in accuracy in many of the combinations is not significant. Nevertheless, there are certain combinations of algorithms especially those involving NB, Linear SVM, RBF SVM, RF, and NN classifiers. This can be seen in Fig. 4 where the evolution of these algorithms with the two techniques and in combination with the Dlib HOG and Dlib ResNet algorithms is illustrated. Thus, it can be seen that in general, these classifiers perform much better as they have more information to train their

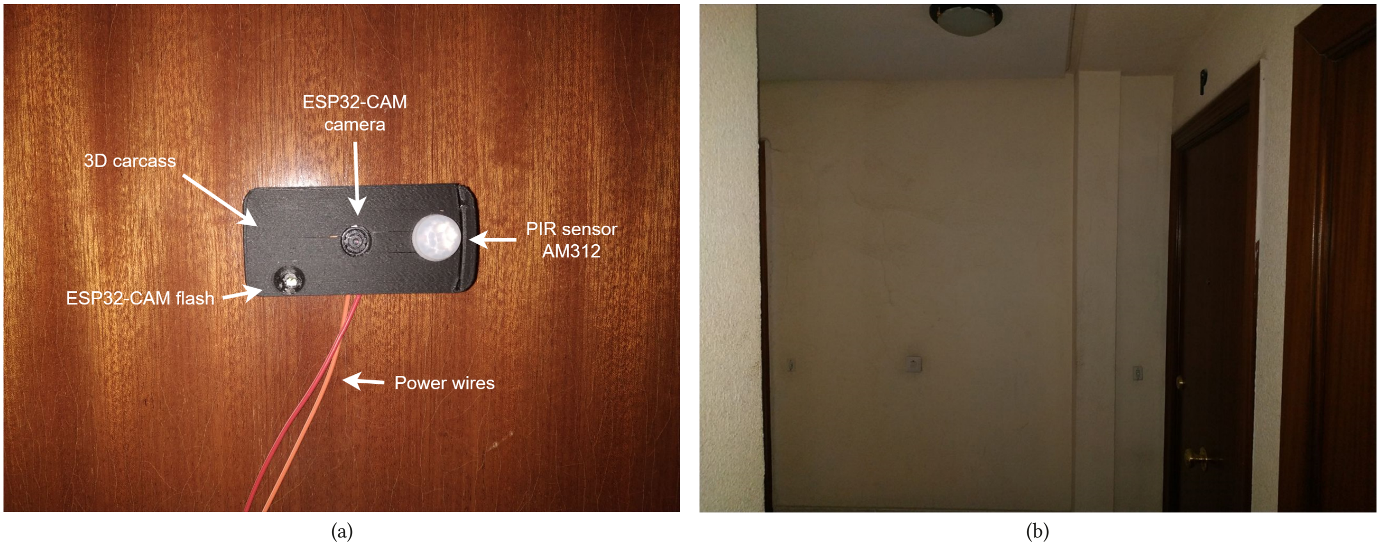

Fig. 5. Edge camera device location and view.

models. Similarly, and in general, for all the combinations studied, it can be observed that as the number of classes decreases, the accuracy of the algorithms increases.

Similarly, it is important to analyze which combination of algorithms is most effective in each of the experiments that have been developed. These can be found in Table II. It is interesting to note that the best combination of algorithms is the one formed by the Dlib HOG algorithm for face detection, the Dlib ResNet for the face encoder, and the k-NN algorithm as the face classification method. This combination of algorithms and methods obtains the best results in every experiment carried out in this study.

TABLE II. Best Algorithms Combinations

Experiment case	Algorithms combination	Accuracy value
143 classes not augmented	Dlib HOG & Dlib ResNet & k-NN	0.9096
143 classes augmented	Dlib HOG & Dlib ResNet & k-NN	0.8720
85 classes not augmented	Dlib HOG & Dlib ResNet & k-NN	0.9326
85 classes augmented	Dlib HOG & Dlib ResNet & k-NN	0.9201
57 classes not augmented	Dlib HOG & Dlib ResNet & k-NN and Dlib HOG & Dlib ResNet & NB	0.9427
57 classes augmented	Dlib HOG & Dlib ResNet & k-NN	0.9254
41 classes not augmented	Dlib HOG & Dlib ResNet & k-NN	0.9429
41 classes augmented	Dlib HOG & Dlib ResNet & k-NN	0.9102
19 classes not augmented	Dlib HOG & Dlib ResNet & k-NN	0.9477
19 classes augmented	Dlib HOG & Dlib ResNet & k-NN	0.9400

B. Hardware Systems Results

Following the design model of the hardware components shown in Fig. 3, an edge camera device that sends photos to the Raspberry Pi edge server every time it detects movement in the viewing area of its PIR sensor has been formed. This hardware is complemented with the software loaded in the ESP32-CAM microcontroller that is in charge of reading the signal received from the PIR sensor and waking up the ESP32-CAM module to capture and send the images to the edge server.

To protect the hardware, it is housed in a 3D printed casing so that its components are protected and less accessible to users. This housing will be located at the entrance door of a house as shown in Fig. 5a. In addition, a view of what the ESP32-CAM's optical sensor observes can be seen at Fig. 5b.

C. Notification Systems Results

The notification system consists of a Telegram bot that the user has to start in his Telegram application and from that moment on he will receive notifications when an unknown person approaches the edge device. This notification will provide a picture of the unknown person along with the exact time the image was taken. An example of the bot chat can be seen at Fig. 6.

D. System Recognition Results

The resulting system has been tested in a real environment to prove its effectiveness with images captured by the hardware developed during the research. As explained in B the hardware has been placed at the front door of a house. This way when a person walks into the hardware the PIR sensor wakes up the ESP32-CAM module and takes pictures which are sent to the Raspberry Pi edge server for analysis.

In this sense, we have performed tests to capture images that are analyzed by the Telegram bot and generated models based on the Dlib HOG detector, the Dlib ResNet encoder, and the k-NN classifier. Thus, the system has been tested with different lighting conditions and by placing the face in different positions.

During these experiments, we have trained a model based on a photograph of any of the people that the model will be able to identify (in our case a single person) that has been augmented following the data augmentation principles already followed previously. This model will be in charge of analyzing the images captured by the hardware.

After analyzing the results, it is observed that in the real environment the model obtains an accuracy value of 100%.

IV. Discussion

After observing the results shown in III, certain conclusions can be drawn that are interesting to discuss in this section.

In general, it can be observed that the one-shot augmented learning approach increases the results concerning the one-shot learning approach. However, in certain combinations of algorithms, the

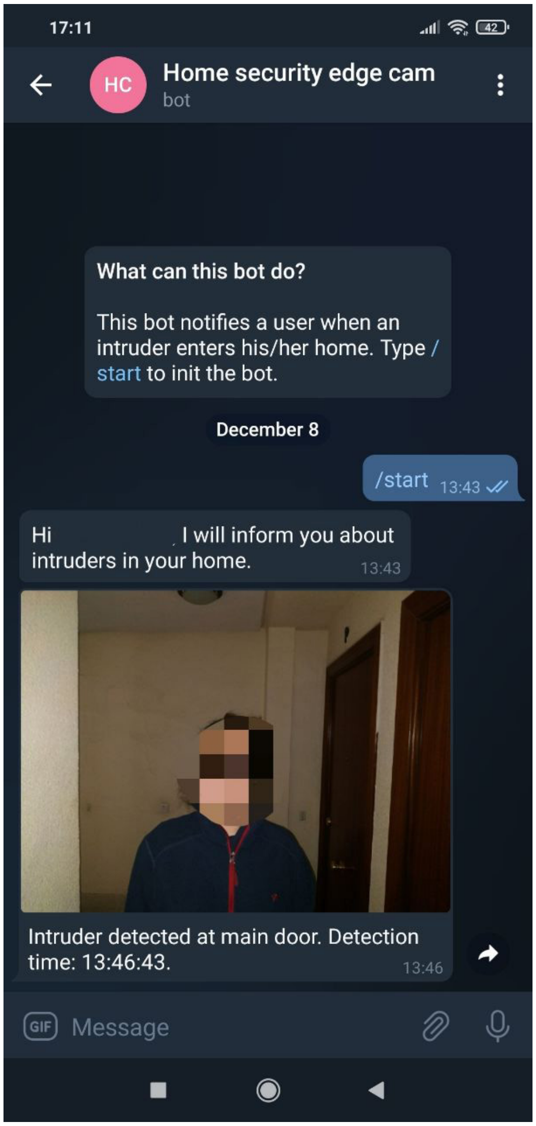

Fig. 6. Telegram bot chat screenshot.

difference is practically negligible, or in others, the results are lower. But in the same way, certain combinations of algorithms such as those shown in Fig. 4 greatly increase the results concerning one-shot learning. This makes us think that specifically these combinations of algorithms, especially the classifier, work better the more data variety they have.

Similarly, it can be seen that the best performing combination of algorithms is the combination of the Dlib HOG, Dlib ResNet, and k-NN algorithms. It is this combination of algorithms that has been used to identify the faces captured by the edge camera device. And as mentioned in D, the accuracy of the model is 100%. In principle, the expected efficiency of the model was high, although such a high efficiency was not expected. The efficiency is likely to be increased as the model has fewer classes and can discern better between users and because the position of the sensor allows good quality images to be obtained. The fact that the accuracy of the algorithm increases as the number of classes decreases is also supported by the results shown in Table 3; it can be observed that the efficiency and the number of classes are inversely proportional to each other.

Furthermore, in this study, it has to be taken into account that the use of edge and one-shot learning technology reduces one of the concerns of computer system users, the use of their data. The proposed system makes use of easily and cheaply available hardware that is always within the user's reach and connected to a local Internet network. On the one hand, the edge solution allows the user's data to be processed on devices close to the user and reduces the transit of information through cloud systems, with the user being the owner of his or her information. On the other hand, the use of one-shot learning reduces the amount of user information needed to train AI models.

On the other hand, the alert system tries to be as unobtrusive as possible for the user. First of all, using an instant messaging application such as Telegram allows the user to easily use the notification system. Secondly, the configuration of the Telegram bot itself prevents it from sending constant notifications to the user; only when a user unknown to the system approaches the edge camera device. Thus, it can be seen that the proposed system offers a comprehensive system for home security control using little data and powerful and accurate AI algorithms for the correct functioning of the alert system.

V. Conclusions

In this paper, we have built a face recognition system based on edge computing technologies and the combination of one-shot learning and data augmentation. The recognition system also has a notification system based on a bot of the Telegram application capable of notifying the user of intruders or unknown persons in the vicinity of their home.

In addition, the facial recognition system is efficient and capable of running on systems with limited computing capabilities. On the other hand, the effectiveness of the one-shot augmented learning technique has been demonstrated in certain algorithms focused on facial recognition. Moreover, we have discovered the perfect combination of detection, encoder, and classification algorithms for tasks in this domain.

Also, the fact that the system is focused on the edge computing paradigm allows users to be more aware of the information that is used about them in a local network environment. In this way, users are less reticent to use these technologies, and the security and speed in the delivery of results are improved concerning computing systems such as the cloud.

Thus, this article also demonstrates that complex artificial intelligence models can now be run on edge devices with certain computational capabilities without time and response delays. In the case of the proposed system, this is an important feature because the speed of intrusion detection and notification is essential for the physical security of the system's users.

On the other hand, following the conclusion of this study, new lines of research have emerged that may be interesting to address in the future. For example, it would be interesting to implement a face detection model for microcontrollers in such a way as to reduce the transit of images between the ESP32-CAM microcontroller and the Raspberry Pi edge server; in this way, the risk of the photos taken by the edge camera device being leaked to third parties can be minimized as much as possible, as only the identity of the person/s detected would be transmitted. Another interesting area of research is to analyze whether the one-shot augmented learning technique is equally effective in other areas of object detection, such as traffic analysis in urban environments.

Appendix

A. Face Recognition Study Results

We show in Table III a complete table with the results of all the experiments carried out in the facial recognition study.

TABLE III. Face Recognition Study Results

Models	143 classes		85 classes		57 classes		41 classes		19 classes	
	Not augmented	Augmented	Not augmented	Augmented	Not augmented	Augmented	Not augmented	Augmented	Not augmented	Augmented
Haar Cascade & OpenFace & NB	0.0034	0.0680	0.0105	0.1034	0.0086	0.1286	0.0138	0.1083	0.0329	0.2123
Haar Cascade & OpenFace & Linear SVM	0.0014	0.0737	0.0036	0.1125	0.0010	0.1385	0.0022	0.1337	0.0144	0.2327
Haar Cascade & OpenFace & RBF SVM	0.0030	0.0701	0.0094	0.0926	0.0090	0.1263	0.0080	0.1297	0.0284	0.2133
Haar Cascade & Open Face & k-NN	0.0996	0.0744	0.1368	0.1037	0.1651	0.1318	0.1472	0.1508	0.2197	0.2043
Haar Cascade & OpenFace & DT	0.0050	0.0107	0.0138	0.1205	0.0138	0.0205	0.0262	0.0523	0.0543	0.1091
Haar Cascade & OpenFace & RF	0.0261	0.0430	0.0312	0.0675	0.0473	0.0720	0.0494	0.0912	0.1405	0.1699
Haar Cascade & OpenFace & NN	0.0522	0.0614	0.0626	0.1114	0.0525	0.1219	0.0556	0.1312	0.0384	0.2063
Haar Cascade & OpenFace & AB	0.0055	0.0076	0.0176	0.1186	0.0230	0.0189	0.0229	0.0222	0.0653	0.0448
Haar Cascade & OpenFace & QDA	0.0055	0.0034	0.0176	0.0105	0.0230	0.0106	0.0229	0.0163	0.0653	0.0309
Haar Cascade & Dlib ResNet & NB	0.0062	0.3899	0.0105	0.6261	0.0080	0.6673	0.0313	0.6185	0.0404	0.7613
Haar Cascade & Dlib ResNet & Linear SVM	0.0002	0.7429	0.0011	0.8258	0.0000	0.8439	0.0004	0.8201	0.0015	0.8989
Haar Cascade & Dlib ResNet & RBF SVM	0.0005	0.7008	0.0011	0.7827	0.0006	0.8228	0.0004	0.7820	0.0055	0.8825
Haar Cascade & Dlib ResNet & k-NN	0.8679	0.8162	0.8889	0.8729	0.8861	0.8759	0.8870	0.8674	0.9008	0.9038
Haar Cascade & Dlib ResNet & DT	0.0098	0.0108	0.0179	0.0447	0.0534	0.0390	0.0560	0.0930	0.0597	0.1280
Haar Cascade & Dlib ResNet & RF	0.0994	0.1401	0.1238	0.2294	0.1398	0.1663	0.2013	0.4001	0.3478	0.5850
Haar Cascade & Dlib ResNet & NN	0.0469	0.4064	0.0163	0.7717	0.0077	0.7780	0.0403	0.8216	0.0344	0.9033
Haar Cascade & Dlib ResNet & AB	0.0085	0.0151	0.0565	0.0342	0.0528	0.0198	0.0465	0.0443	0.1151	0.0772
Haar Cascade & Dlib ResNet & QDA	0.0085	0.0041	0.0565	0.0119	0.0528	0.0150	0.0465	0.0185	0.1151	0.0274
Haar Cascade & FaceNet & NB	0.0034	0.2775	0.0105	0.2934	0.0102	0.3474	0.0149	0.3521	0.0229	0.4668
Haar Cascade & FaceNet & Linear SVM	0.0002	0.4519	0.0008	0.4921	0.0010	0.5691	0.0029	0.3899	0.0065	0.6517
Haar Cascade & FaceNet & RBF SVM	0.0046	0.2580	0.0025	0.3300	0.0016	0.2889	0.0022	0.2544	0.0060	0.5047
Haar Cascade & FaceNet & k-NN	0.3988	0.5501	0.5230	0.5966	0.5390	0.6372	0.5214	0.5385	0.6572	0.6841
Haar Cascade & FaceNet & DT	0.0112	0.0080	0.0099	0.0179	0.0198	0.0266	0.0291	0.0400	0.0862	0.0693
Haar Cascade & FaceNet & RF	0.0327	0.0888	0.0585	0.1977	0.0972	0.2505	0.0763	0.1867	0.2013	0.3672
Haar Cascade & FaceNet & NN	0.2772	0.4464	0.3303	0.5751	0.2879	0.5861	0.1439	0.4931	0.1061	0.7145
Haar Cascade & FaceNet & AB	0.0060	0.0050	0.0265	0.0168	0.0377	0.0157	0.0578	0.0164	0.1216	0.0304
Haar Cascade & FaceNet & QDA	0.0060	0.0451	0.0265	0.0540	0.0377	0.0499	0.0578	0.0730	0.1216	0.1420
Dlib HOG & OpenFace & NB	0.0083	0.0033	0.0233	0.0108	0.2370	0.0099	0.0409	0.0169	0.1096	0.0353
Dlib HOG & OpenFace & Linear SVM	0.0082	0.0033	0.0233	0.0108	0.0003	0.0099	0.0409	0.0169	0.1096	0.0353
Dlib HOG & OpenFace & RBF SVM	0.0083	0.0033	0.0233	0.0108	0.0003	0.0099	0.0409	0.0169	0.1096	0.0353
Dlib HOG & Open Face & k-NN	0.0083	0.0033	0.0233	0.0108	0.2370	0.0099	0.0409	0.0169	0.1096	0.0353
Dlib HOG & OpenFace & DT	0.0083	0.0033	0.0233	0.0108	0.0186	0.0099	0.0409	0.0169	0.1096	0.0353
Dlib HOG & OpenFace & RF	0.0083	0.0033	0.0233	0.0108	0.0620	0.0099	0.0409	0.0169	0.1096	0.0353
Dlib HOG & OpenFace & NN	0.0083	0.0033	0.0233	0.0108	0.1574	0.0099	0.0409	0.0169	0.1096	0.0353
Dlib HOG & OpenFace & AB	0.0083	0.0033	0.0233	0.0108	0.0334	0.0099	0.0409	0.0169	0.1096	0.0353
Dlib HOG & OpenFace & QDA	0.0083	0.0033	0.0233	0.0108	0.0334	0.0099	0.0409	0.0169	0.1096	0.0353
Dlib HOG & Dlib ResNet & NB	0.0035	0.2779	0.0134	0.6016	0.9427	0.5174	0.0154	0.3651	0.2946	0.5564

Models	143 classes		85 classes		57 classes		41 classes		19 classes	
	Not augmented	Augmented	Not augmented	Augmented	Not augmented	Augmented	Not augmented	Augmented	Not augmented	Augmented
Dlib HOG & Dlib ResNet & Linear SVM	0.0007	0.8246	0.0000	0.8993	0.0000	0.9165	0.0000	0.9046	0.0497	0.9390
Dlib HOG & Dlib ResNet & RBF SVM	0.0005	0.7410	0.0000	0.8615	0.0000	0.8820	0.0000	0.8681	0.0154	0.8888
Dlib HOG & Dlib ResNet & k-NN	0.9096	0.8720	0.9326	0.9201	0.9427	0.9254	0.9429	0.9098	0.9477	0.9400
Dlib HOG & Dlib ResNet & DT	0.0106	0.0097	0.0162	0.0466	0.0242	0.0484	0.0451	0.0793	0.1522	0.3509
Dlib HOG & Dlib ResNet & RF	0.0646	0.1943	0.1667	0.2502	0.1654	0.3033	0.2145	0.3768	0.3852	0.6112
Dlib HOG & Dlib ResNet & NN	0.0654	0.6298	0.0344	0.8487	0.2711	0.8542	0.0154	0.9102	0.2940	0.9355
Dlib HOG & Dlib ResNet & AB	0.0170	0.0168	0.0421	0.0358	0.1385	0.0282	0.0796	0.0237	0.1619	0.3279
Dlib HOG & Dlib ResNet & QDA	0.0170	0.0035	0.0421	0.0114	0.1385	0.0259	0.0796	0.0154	0.1619	0.2941
Dlib HOG & FaceNet & NB	0.0035	0.0378	0.0199	0.0478	0.4332	0.0447	0.0616	0.0616	0.1086	0.1296
Dlib HOG & FaceNet & Linear SVM	0.0035	0.0378	0.0199	0.0478	0.0007	0.0447	0.0616	0.0616	0.1086	0.1296
Dlib HOG & FaceNet & RBF SVM	0.0035	0.0378	0.0199	0.0478	0.0007	0.0447	0.0058	0.0616	0.1086	0.1296
Dlib HOG & FaceNet & k-NN	0.0035	0.0378	0.0199	0.0478	0.4332	0.0447	0.0616	0.0616	0.1086	0.1296
Dlib HOG & FaceNet & DT	0.0035	0.0378	0.0199	0.0478	0.0166	0.0447	0.0616	0.0616	0.1086	0.1296
Dlib HOG & FaceNet & RF	0.0035	0.0378	0.0199	0.0478	0.0620	0.0447	0.0616	0.0616	0.1086	0.1296
Dlib HOG & FaceNet & NN	0.0035	0.0378	0.0199	0.0478	0.3099	0.0447	0.0616	0.0616	0.1086	0.1296
Dlib HOG & FaceNet & AB	0.0035	0.0378	0.0199	0.0478	0.0308	0.0447	0.0616	0.0616	0.1086	0.1296
Dlib HOG & FaceNet & QDA	0.0035	0.0378	0.0199	0.0478	0.0308	0.0447	0.0616	0.0616	0.1086	0.1296
Dlib CNN & OpenFace & NB	0.0064	0.0032	0.0168	0.0108	0.0235	0.0116	0.0368	0.0149	0.0882	0.0304
Dlib CNN & OpenFace & Linear SVM	0.0064	0.0032	0.0168	0.0108	0.0234	0.0116	0.0368	0.0149	0.0882	0.0304
Dlib CNN & OpenFace & RBF SVM	0.0064	0.0032	0.0168	0.0108	0.0235	0.0116	0.0368	0.0149	0.0882	0.0304
Dlib CNN & Open Face & k-NN	0.0064	0.0032	0.0168	0.0108	0.0235	0.0116	0.0368	0.0149	0.0882	0.0304
Dlib CNN & OpenFace & DT	0.0064	0.0032	0.0168	0.0108	0.0234	0.0116	0.0368	0.0149	0.0882	0.0304
Dlib CNN & OpenFace & RF	0.0064	0.0032	0.0168	0.0108	0.0235	0.0116	0.0368	0.0149	0.0882	0.0304
Dlib CNN & OpenFace & NN	0.0064	0.0032	0.0168	0.0108	0.0235	0.0116	0.0368	0.0149	0.0882	0.0304
Dlib CNN & OpenFace & AB	0.0064	0.0032	0.0168	0.0108	0.0235	0.0116	0.0368	0.0149	0.0882	0.0304
Dlib CNN & OpenFace & QDA	0.0064	0.0032	0.0168	0.0108	0.0235	0.0116	0.0368	0.0149	0.0882	0.0304
Dlib CNN & Dlib ResNet & NB	0.0071	0.4392	0.0119	0.7280	0.0090	0.0116	0.0186	0.4529	0.3034	0.5989
Dlib CNN & Dlib ResNet & Linear SVM	0.0002	0.8041	0.0003	0.8695	0.0045	0.0116	0.0004	0.8892	0.0010	0.9108
Dlib CNN & Dlib ResNet & RBF SVM	0.0011	0.7619	0.0003	0.8381	0.0045	0.8525	0.0004	0.8240	0.0010	0.8804
Dlib CNN & Dlib ResNet & k-NN	0.8848	0.8429	0.9092	0.8891	0.9148	0.8994	0.9122	0.8899	0.9183	0.9118
Dlib CNN & Dlib ResNet & DT	0.0135	0.0112	0.0317	0.0345	0.0312	0.0437	0.0652	0.0612	0.1380	0.3473
Dlib CNN & Dlib ResNet & RF	0.0516	0.1489	0.1799	0.2720	0.1630	0.2713	0.2077	0.3407	0.3827	0.6846
Dlib CNN & Dlib ResNet & NN	0.1172	0.5890	0.0248	0.8378	0.0283	0.8271	0.0292	0.8750	0.3338	0.8954
Dlib CNN & Dlib ResNet & AB	0.0289	0.0048	0.0394	0.0270	0.0543	0.1356	0.0893	0.0361	0.3627	0.2890
Dlib CNN & Dlib ResNet & QDA	0.0289	0.0037	0.0394	0.0113	0.0543	0.0084	0.0893	0.0149	0.3627	0.2985
Dlib CNN & FaceNet & NB	0.0046	0.0328	0.0171	0.0436	0.0289	0.0437	0.0517	0.0714	0.0977	0.1305
Dlib CNN & FaceNet & Linear SVM	0.0046	0.0328	0.0171	0.0436	0.0289	0.0437	0.0517	0.0714	0.0977	0.1305
Dlib CNN & FaceNet & RBF SVM	0.0046	0.0328	0.0171	0.0436	0.0289	0.0437	0.0517	0.0714	0.0977	0.1305
Dlib CNN & FaceNet & k-NN	0.0046	0.0328	0.0171	0.0436	0.0289	0.0437	0.0517	0.0714	0.0977	0.1305

Models	143 classes		85 classes		57 classes		41 classes		19 classes	
	Not augmented	Augmented	Not augmented	Augmented	Not augmented	Augmented	Not augmented	Augmented	Not augmented	Augmented
Dlib CNN & FaceNet & DT	0.0046	0.0328	0.0171	0.0436	0.0289	0.0437	0.0517	0.0714	0.0977	0.1305
Dlib CNN & FaceNet & RF	0.0046	0.0328	0.0171	0.0436	0.0289	0.0437	0.0517	0.0714	0.0977	0.1305
Dlib CNN & FaceNet & NN	0.0046	0.0328	0.0171	0.0436	0.0289	0.0437	0.0517	0.0714	0.0977	0.1305
Dlib CNN & FaceNet & AB	0.0046	0.0328	0.0171	0.0436	0.0289	0.0437	0.0517	0.0714	0.0977	0.1305
Dlib CNN & FaceNet & QDA	0.0046	0.0328	0.0171	0.0436	0.0289	0.0437	0.0517	0.0714	0.0977	0.1305
SSD-Resnet & OpenFace & NB	0.0032	0.0659	0.0185	0.0107	0.0334	0.0105	0.0228	0.0186	0.0815	0.0349
SSD-Resnet & OpenFace & Linear SVM	0.0038	0.0032	0.0185	0.0107	0.0335	0.0096	0.0022	0.0186	0.0815	0.0349
SSD-Resnet & OpenFace & RBF SVM	0.0038	0.0032	0.0185	0.0107	0.0335	0.0105	0.0112	0.0186	0.0815	0.0349
SSD-Resnet & Open Face & k-NN	0.0038	0.0032	0.0185	0.0107	0.0335	0.0105	0.1317	0.0186	0.0815	0.0349
SSD-Resnet & OpenFace & DT	0.0038	0.0032	0.0185	0.0107	0.0335	0.0105	0.0231	0.0186	0.0815	0.0349
SSD-Resnet & OpenFace & RF	0.0038	0.0032	0.0185	0.0107	0.0335	0.0105	0.0465	0.0186	0.0815	0.0349
SSD-Resnet & OpenFace & NN	0.0038	0.0032	0.0185	0.0107	0.0335	0.0105	0.0519	0.0186	0.0815	0.0349
SSD-Resnet & OpenFace & AB	0.0038	0.0032	0.0185	0.0107	0.0335	0.0105	0.0167	0.0186	0.0815	0.0349
SSD-Resnet & OpenFace & QDA	0.0038	0.0032	0.0185	0.0107	0.0335	0.0105	0.0167	0.0186	0.0815	0.0349
SSD-Resnet & Dlib ResNet & NB	0.0078	0.5956	0.0182	0.7255	0.0105	0.7009	0.0246	0.6904	0.2934	0.7367
SSD-Resnet & Dlib ResNet & Linear SVM	0.0070	0.3656	0.0075	0.4888	0.0017	0.5798	0.0016	0.5904	0.0379	0.7079
SSD-Resnet & Dlib ResNet & RBF SVM	0.0030	0.3365	0.0044	0.4762	0.0019	0.5492	0.0016	0.5561	0.0301	0.6770
SSD-Resnet & Dlib ResNet & k-NN	0.7746	0.7242	0.7930	0.7787	0.8086	0.7870	0.8003	0.7500	0.8147	0.8047
SSD-Resnet & Dlib ResNet & DT	0.0082	0.0100	0.0320	0.0270	0.0496	0.0227	0.0740	0.0596	0.1369	0.3461
SSD-Resnet & Dlib ResNet & RF	0.0697	0.1069	0.0845	0.2354	0.0981	0.3073	0.1429	0.2497	0.3247	0.4826
SSD-Resnet & Dlib ResNet & NN	0.0377	0.1886	0.0296	0.5148	0.0300	0.7080	0.0221	0.7538	0.2934	0.7926
SSD-Resnet & Dlib ResNet & AB	0.0199	0.0633	0.0304	0.1645	0.0508	0.1885	0.1006	0.1875	0.2424	0.3548
SSD-Resnet & Dlib ResNet & QDA	0.0199	0.0173	0.0304	0.0175	0.0508	0.0366	0.1006	0.0356	0.2424	0.3034
SSD-Resnet & FaceNet & NB	0.0052	0.0362	0.0245	0.0469	0.0303	0.0468	0.0397	0.0737	0.0990	0.1129
SSD-Resnet & FaceNet & Linear SVM	0.0052	0.0362	0.0245	0.0469	0.0303	0.0468	0.0010	0.0737	0.0990	0.1129
SSD-Resnet & FaceNet & RBF SVM	0.0052	0.0362	0.0245	0.0469	0.0303	0.0468	0.0026	0.0737	0.0990	0.1129
SSD-Resnet & FaceNet & k-NN	0.0052	0.0362	0.0245	0.0469	0.0303	0.0468	0.4577	0.0737	0.0990	0.1129
SSD-Resnet & FaceNet & DT	0.0052	0.0362	0.0245	0.0469	0.0303	0.0468	0.0372	0.0737	0.0990	0.1129
SSD-Resnet & FaceNet & RF	0.0052	0.0362	0.0245	0.0469	0.0303	0.0468	0.0942	0.0737	0.0990	0.1129
SSD-Resnet & FaceNet & NN	0.0052	0.0362	0.0245	0.0469	0.0303	0.0468	0.1135	0.0737	0.0990	0.1129
SSD-Resnet & FaceNet & AB	0.0052	0.0362	0.0245	0.0469	0.0303	0.0468	0.0679	0.0737	0.0990	0.1129
SSD-Resnet & FaceNet & QDA	0.0052	0.0362	0.0245	0.0469	0.0303	0.0468	0.0679	0.0737	0.0990	0.1129
MTCNN & OpenFace & NB	0.0050	0.0028	0.0142	0.0108	0.0076	0.0116	0.0142	0.0184	0.0271	0.0302
MTCNN & OpenFace & Linear SVM	0.0000	0.0028	0.0023	0.0108	0.0007	0.0116	0.0142	0.0184	0.0041	0.0302
MTCNN & OpenFace & RBF SVM	0.0005	0.0028	0.0043	0.0108	0.0033	0.0116	0.0142	0.0184	0.0072	0.0302
MTCNN & Open Face & k-NN	0.1143	0.0028	0.1670	0.0108	0.1700	0.0116	0.0142	0.0184	0.3183	0.0302
MTCNN & OpenFace & DT	0.0064	0.0028	0.0117	0.0108	0.0189	0.0116	0.0142	0.0184	0.0450	0.0302
MTCNN & OpenFace & RF	0.0434	0.0028	0.0478	0.0108	0.0832	0.0116	0.0142	0.0184	0.1372	0.0302
MTCNN & OpenFace & NN	0.0703	0.0028	0.0774	0.0108	0.1107	0.0116	0.0142	0.0184	0.0415	0.0302

Models	143 classes		85 classes		57 classes		41 classes		19 classes	
	Not augmented	Augmented	Not augmented	Augmented	Not augmented	Augmented	Not augmented	Augmented	Not augmented	Augmented
MTCNN & OpenFace & AB	0.0127	0.0028	0.0489	0.0108	0.0315	0.0116	0.0142	0.0184	0.1008	0.0302
MTCNN & OpenFace & QDA	0.0127	0.0028	0.0489	0.0108	0.0315	0.0116	0.0142	0.0184	0.1008	0.0302
MTCNN & Dlib ResNet & NB	0.0059	0.3817	0.0188	0.4835	0.0080	0.5502	0.0169	0.4462	0.0271	0.6095
MTCNN & Dlib ResNet & Linear SVM	0.0000	0.7512	0.0009	0.8595	0.0007	0.8581	0.0000	0.8639	0.0041	0.9012
MTCNN & Dlib ResNet & RBF SVM	0.0002	0.6832	0.0003	0.7924	0.0007	0.8081	0.0000	0.8185	0.0020	0.8654
MTCNN & Dlib ResNet & k-NN	0.8123	0.8239	0.8857	0.8885	0.8910	0.8943	0.8886	0.8598	0.9217	0.9094
MTCNN & Dlib ResNet & DT	0.0193	0.0083	0.0739	0.0580	0.0365	0.0404	0.0408	0.0600	0.0619	0.2462
MTCNN & Dlib ResNet & RF	0.0772	0.1499	0.1487	0.2506	0.2310	0.3338	0.1856	0.3930	0.2917	0.5896
MTCNN & Dlib ResNet & NN	0.0482	0.4455	0.0256	0.5762	0.0172	0.8101	0.0169	0.8766	0.0271	0.8884
MTCNN & Dlib ResNet & AB	0.0198	0.0080	0.0538	0.0535	0.1028	0.0345	0.0964	0.1057	0.1029	0.3306
MTCNN & Dlib ResNet & QDA	0.0198	0.0080	0.0538	0.0100	0.1028	0.0361	0.0964	0.0405	0.1029	0.3076
MTCNN & FaceNet & NB	0.0042	0.0372	0.0111	0.0404	0.0076	0.0431	0.0664	0.0589	0.0271	0.1269
FaceNet & OpenFace & Linear SVM	0.0000	0.0630	0.0023	0.1263	0.0007	0.1299	0.0041	0.1320	0.0031	0.1270
FaceNet & OpenFace & RBF SVM	0.0005	0.0574	0.0043	0.0904	0.0017	0.1385	0.0124	0.1207	0.0077	0.3367
FaceNet & Open Face & k-NN	0.1164	0.0659	0.1638	0.1297	0.1720	0.1177	0.1729	0.1466	0.3321	0.2124
FaceNet & OpenFace & DT	0.0054	0.0106	0.0225	0.0102	0.0199	0.0182	0.0326	0.0442	0.0461	0.0885
FaceNet & OpenFace & RF	0.0165	0.0305	0.0407	0.0475	0.0673	0.0726	0.0645	0.0716	0.1080	0.1837
FaceNet & OpenFace & NN	0.0710	0.0531	0.0757	0.1024	0.1223	0.1127	0.0960	0.1125	0.0328	0.2646
FaceNet & OpenFace & AB	0.0127	0.0090	0.0344	0.0097	0.0421	0.0133	0.0585	0.0337	0.1039	0.0415
FaceNet & OpenFace & QDA	0.0127	0.0068	0.0344	0.0108	0.0421	0.0149	0.0585	0.0277	0.1039	0.2984
FaceNet & Dlib ResNet & NB	0.0084	0.3834	0.0108	0.4849	0.0080	0.5406	0.0169	0.4421	0.0271	0.6024
FaceNet & Dlib ResNet & Linear SVM	0.0002	0.7476	0.0006	0.8544	0.0007	0.8598	0.0000	0.8616	0.0036	0.9017
FaceNet & Dlib ResNet & RBF SVM	0.0002	0.6759	0.0003	0.7958	0.0007	0.8127	0.0000	0.7945	0.0020	0.8634
FaceNet & Dlib ResNet & k-NN	0.8135	0.8239	0.8842	0.8874	0.8919	0.8943	0.8886	0.8594	0.9232	0.9084
FaceNet & Dlib ResNet & DT	0.0189	0.0116	0.0304	0.0284	0.0779	0.0222	0.0255	0.0664	0.0686	0.2344
FaceNet & Dlib ResNet & RF	0.0725	0.1551	0.1493	0.3049	0.1511	0.2883	0.1995	0.3067	0.4734	0.5328
FaceNet & Dlib ResNet & NN	0.0541	0.4585	0.0262	0.5552	0.0080	0.8121	0.0169	0.8733	0.0271	0.8884
FaceNet & Dlib ResNet & AB	0.0118	0.0102	0.0401	0.0427	0.0776	0.0663	0.0855	0.1219	0.1039	0.3347
FaceNet & Dlib ResNet & QDA	0.0118	0.0102	0.0401	0.0100	0.0776	0.0365	0.0855	0.0397	0.1039	0.3086
FaceNet & FaceNet & NB	0.0047	0.0626	0.0105	0.0392	0.0080	0.0722	0.0169	0.0769	0.0271	0.1807
FaceNet & FaceNet & Linear SVM	0.0005	0.2875	0.0014	0.2844	0.0013	0.3271	0.0015	0.2767	0.0031	0.3536
FaceNet & FaceNet & RBF SVM	0.0032	0.0918	0.0017	0.0518	0.0000	0.0679	0.0049	0.1053	0.0241	0.1162
FaceNet & FaceNet & k-NN	0.3021	0.3762	0.3842	0.4337	0.4461	0.4713	0.4942	0.3547	0.5415	0.5061
FaceNet & FaceNet & DT	0.0123	0.0094	0.0176	0.0162	0.0182	0.0169	0.0176	0.0199	0.1325	0.0425
FaceNet & FaceNet & RF	0.0375	0.0737	0.0566	0.0904	0.0965	0.1173	0.0874	0.1324	0.1100	0.2958
FaceNet & FaceNet & NN	0.1803	0.3019	0.2392	0.3706	0.1936	0.4309	0.1616	0.3135	0.1566	0.4749
FaceNet & FaceNet & AB	0.0177	0.0113	0.0185	0.0108	0.0633	0.0129	0.0536	0.0229	0.0502	0.3117
FaceNet & FaceNet & QDA	0.0177	0.0380	0.0185	0.0364	0.0633	0.0663	0.0536	0.0739	0.0502	0.1274

Acknowledgment

This work was supported by the Spanish Agencia Estatal de Investigación. Project Monitoring and tracking systems for the improvement of intelligent mobility and behavior analysis (SiMoMIAC). PID2019-108883RB-C21 / AEI /

10.13039/501100011033. The research of Diego M. Jiménez-Bravo has been co-financed by the European NextGenrationEU Fund, Spanish "Plan de Recuperación, Transformación y Resilencia" Fund, Spaninsh Ministry of Universities and University of Salamanca ("Ayudas para la recualificación del sistema universitario español para 2021-2023"). André Filipe Sales Mendes's research was co-financed by the European Social Fund and Junta de Castilla y León (Operational Programme 2014-2020 for Castilla y León, EDU/556/2019 BOCYL).

References

[1] E. G. Miller, N. E. Matsakis, P. A. Viola, "Learning from one example through shared densities on transforms," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1, pp. 464– 471, 2000, doi: 10.1109/CVPR.2000.855856.

[2] L. Fei-Fei, R. Fergus, P. Perona, "One-shot learning of object categories," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, pp. 594–611, 4 2006, doi: 10.1109/TPAMI.2006.79.

[3] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, D. Wierstra, "Matching networks for one shot learning," in *Advances in Neural Information Processing Systems*, 6 2016, pp. 3637–3645, Neural information processing systems foundation.

[4] A. Shaban, S. Bansal, Z. Liu, I. Essa, B. Boots, "One-shot learning for semantic segmentation," *British Machine Vision Conference 2017, BMVC 2017*, 9 2017, doi: 10.5244/c.31.167.

[5] M. Woodward, C. Finn, "Active One-shot Learning," *arXiv*, 2 2017.

[6] P. Wang, L. Liu, C. Shen, Z. Huang, A. Van Den Hengel, H. T. Shen, "Multi-attention network for one shot learning," 2017. doi: 10.1109/CVPR.2017.658.

[7] H. Altae-Tran, B. Ramsundar, A. S. Pappu, V. Pande, "Low Data Drug Discovery with One-Shot Learning," *ACS Central Science*, vol. 3, pp. 283–293, 4 2017, doi: 10.1021/acscentsci.6b00367.

[8] C. Shorten, T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *Journal of Big Data*, vol. 6, pp. 1–48, 12 2019, doi: 10.1186/s40537-019-0197-0.

[9] Z. Zhong, L. Zheng, G. Kang, S. Li, Y. Yang, "Random erasing data augmentation," 8 2020. [Online]. Available: https://github.com/zhunzhong07/Random-Erasing., doi: 10.1609/aaai.v34i07.7000.

[10] L. Perez, J. Wang, "The Effectiveness of Data Augmentation in Image Classification using Deep Learning," *arXiv*, 12 2017.

[11] D. S. Park, W. Chan, Y. Zhang, C. C. Chiu, B. Zoph, E. D. Cubuk, Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 2019-Septe, pp. 2613–2617, 4 2019, doi: 10.21437/Interspeech.2019-2680.

[12] A. Kumar, A. Kaur, M. Kumar, "Face detection techniques: a review," *Artificial Intelligence Review*, vol. 52, pp. 927–948, 8 2019, doi: 10.1007/s10462-018- 9650-2.

[13] P. Viola, M. J. Jones, "Robust Real-Time Face Detection," *International Journal of Computer Vision*, vol. 57, pp. 137–154, 5 2004, doi: 10.1023/B:VISI.0000013087.49260.fb.

[14] A. Jadhav, S. Lone, S. Matey, T. Madamwar, S. Jakhete, "Survey on Face Detection Algorithms," 2021. [Online]. Available: www.ijisrt.com.

[15] X. Lu, X. Kang, S. Nishide, F. Ren, "Object detection based on SSD-ResNet," in *Proceedings of 2019 6th IEEE International Conference on Cloud Computing and Intelligence Systems, CCIS 2019*, 12 2019, pp. 89–92, Institute of Electrical and Electronics Engineers Inc.

[16] K. Zhang, Z. Zhang, Z. Li, Y. Qiao, "Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks," *IEEE Signal Processing Letters*, vol. 23, pp. 1499–1503, 4 2016, doi: 10.1109/LSP.2016.2603342.

[17] Chunming Wu, Ying Zhang, "MTCNN and FACENET Based Access Control System for Face Detection and Recognition," *Automatic Control and Computer Sciences*, vol. 55, pp. 102–112, 1 2021, doi: 10.3103/S0146411621010090.

[18] S. Milborrow, F. Nicolls, "Locating facial features with an extended active shape model," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5305 LNCS, 10 2008, pp. 504–513, Springer Verlag.

[19] J. Saragih, R. Goecke, "A nonlinear discriminative approach to AAM fitting," in *Proceedings of the IEEE International Conference on Computer Vision*, 2007.

[20] N. Kumar, P. Belhumeur, S. Nayar, "FaceTracer: A search engine for large collections of images with faces," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5305 LNCS, 10 2008, pp. 340–353, Springer Verlag.

[21] X. Cao, Y. Wei, F. Wen, J. Sun, "Face alignment by explicit shape regression," *International Journal of Computer Vision*, vol. 107, no. 2, pp. 177–190, 2014, doi: 10.1007/s11263-013-0667-3.

[22] G. Trigeorgis, P. Snape, M. A. Nicolaou, E. Antonakos, S. Zafeiriou, "Mnemonic Descent Method: A Recurrent Process Applied for End-to-End Face Alignment," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016- Decem, 12 2016, pp. 4177–4187, IEEE Computer Society.

[23] J. Lv, X. Shao, J. Xing, C. Cheng, X. Zhou, "A deep regression architecture with two-stage re-initialization for high performance facial landmark detection," 2017. doi: 10.1109/CVPR.2017.393.

[24] J. Yang, Q. Liu, K. Zhang, "Stacked Hourglass Network for Robust Facial Landmark Localisation," 2017. [Online]. Available: https://www.umdfaces.io., doi: 10.1109/CVPRW.2017.253.

[25] A. Newell, K. Yang, J. Deng, "Stacked hourglass networks for human pose estimation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9912 LNCS, pp. 483–499, 3 2016, doi: 10.1007/978- 3-319-46484-8_29.

[26] J. Deng, G. Trigeorgis, Y. Zhou, S. Zafeiriou, "Joint Multi-View Face Alignment in the Wild," *IEEE Transactions on Image Processing*, vol. 28, pp. 3636–3648, 8 2019, doi: 10.1109/TIP.2019.2899267.

[27] A. Ahmed, J. Guo, F. Ali, F. Deeba, A. Ahmed, "LBPH based improved face recognition at low resolution," in *2018 International Conference on Artificial Intelligence and Big Data, ICAIBD 2018*, 6 2018, pp. 144–147, Institute of Electrical and Electronics Engineers Inc.

[28] T. Baltrusaitis, A. Zadeh, Y. C. Lim, L. P. Morency, "OpenFace 2.0: Facial behavior analysis toolkit," in *Proceedings - 13th IEEE International Conference on Automatic Face and Gesture Recognition, FG 2018*, 6 2018, pp. 59–66, Institute of Electrical and Electronics Engineers Inc.

[29] K. He, X. Zhang, S. Ren, J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, 12 2016, pp. 770– 778, IEEE Computer Society.

[30] F. Schroff, D. Kalenichenko, J. Philbin, "FaceNet: A Unified Embedding for Face Recognition and Clustering," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June-2015, pp. 815–823, 3 2015, doi: 10.1109/CVPR.2015.7298682.

[31] Z. Wang, G. Wang, B. Huang, Z. Xiong, Q. Hong, H. Wu, P. Yi, K. Jiang, N. Wang, Y. Pei, H. Chen, Y. Miao, Z. Huang, J. Liang, "Masked Face Recognition Dataset and Application," 3 2020.

[32] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, W. Liu, "CosFace: Large Margin Cosine Loss for Deep Face Recognition," 2018. doi: 10.1109/CVPR.2018.00552.

[33] W. Liu, Y. Wen, Z. Yu, M. Li, B. Raj, L. Song, "SphereFace: Deep hypersphere embedding for face recognition," 2017. doi: 10.1109/CVPR.2017.713.

[34] A. Suruliandi, A. Kasthuri, S. P. Raja, "Deep Feature Representation and Similarity Matrix based Noise Label Refinement Method for Efficient Face Annotation," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 7, pp. 2–66, doi: 10.9781/ijimai.2021.05.001.

[35] "European Edge Computing Consortium(EECC)." [Online]. Available: https://ecconsortium.eu/.

[36] X. Jiang, F. R. Yu, T. Song, V. C. Leung, "Intelligent Resource Allocation for Video Analytics in Blockchain- Enabled Internet of Autonomous Vehicles with Edge Computing," *IEEE Internet of Things Journal*, pp. 1–1, 9 2020, doi: 10.1109/jiot.2020.3026354.

[37] M. Gheisari, G. Wang, S. Chen, "An Edge Computing- enhanced Internet of Things Framework for Privacy-preserving in Smart City," *Computers and Electrical Engineering*, vol. 81, p. 106504, 1 2020, doi: 10.1016/j.compeleceng.2019.106504.

[38] H. Albataineh, M. Nijim, D. Bollampall, "The Design of a Novel Smart Home Control System using Smart Grid Based on Edge and Cloud Computing," *2020 8th International Conference on Smart Energy Grid Engineering, SEGE 2020*, pp. 88–91, 8 2020, doi: 10.1109/SEGE49949.2020.9181961.

[39] K. P. Dirgantoro, J. M. Lee, D. S. Kim, "Generative Adversarial Networks Based on Edge Computing with Blockchain Architecture for Security System," *2020 International Conference on Artificial Intelligence in Information and Communication, ICAIIC 2020*, pp. 039– 042, 2 2020, doi: 10.1109/ICAIIC48513.2020.9065231.

[40] L. J. Karam, T. Zhu, "Quality labeled faces in the wild (QLFW): a database for studying face recognition in real-world environments," 2015. [Online]. Available: http://vis-www.cs.umass.edu/lfw/., doi: 10.1117/12.2080393.

Diego M. Jiménez-Bravo

He studied a degree in Computer Engineering Management and Information Systems at the University of the Basque Country (UPV/EHU) (2016). Subsequently, he obtained a Master's degree in Intelligent Systems from the University of Salamanca (USAL) (2017). He concluded his formative stage by obtaining a Ph.D. in Computer Engineering from the USAL (2020). He is part of the research group ESALab, of the same entity, where he carries out research work under a postdoctoral contract funded by the European NextGenrationEU Fund, Spanish "Plan de Recuperación, Transformación y Resilencia" Fund, Spanish Ministry of Universities, and the University of Salamanca. His main research interests focus on the field of artificial intelligence, IoT (Internet of Things), smart-homes, energy optimisation and social networks, among others. He also collaborates in several research projects within the research group.

Álvaro Lozano Murciego

He received the master's degree in intelligent systems and the Ph.D. degree in computer engineering from the University of Salamanca, in 2015 and 2019, respectively. He is currently working on the Expert Systems and Applications Laboratory Research Group, Computer Science Department, University of Salamanca, as an Assistant Professor. Throughout his training, he has followed a well-defined line of research, focused on machine learning, route optimization, IoT sensors, and edge computing.

André Sales Mendes

He is a Ph.D. student in computer engineering. He has completed a computer engineering degree and a master's degree in intelligent systems at the University of Salamanca. He is currently focusing his doctoral studies on artificial intelligence and expert robots. He is a member of the Expert Systems And Applications Laboratory research group at the University of Salamanca. In addition, he has several publications in JCR indexed journals.

Luis Augusto Silva

He received the degree in Internet systems from the Federal Institute of Santa Catarina (IFC), Camboriú, Brazil, in February 2017, and the master's degree in applied computing from the University of Itajaí Valley, Brazil, in 2019. He is currently pursuing the Ph.D. degree incomputer engineering with the University of Salamanca, Spain. His research during his master's degree included the field of notification systems,the IoT, and data privacy. During the master's degree, he was a collaborating Researcher with the Laboratory of Embedded and Distributed Systems (LEDS), UNIVALI, collaborating with research projects related to the Internet of Things. Since August 2020, he has been a Researcher with the Expert Systems and Applications Laboratory (ESALab), Universidad de Salamanca. His research interests include the Internet of Things, embedded drone systems, and data privacy applied to smart environments.

Daniel H. De La Iglesia

He is a Ph.D. in Computer Engineering, a Technical Engineer in Computer Systems and a Degree in Computer Engineering from the University of Salamanca. He has an official master's degree in Intelligent Systems and in recent years has been linked to different research groups in the field of artificial intelligence where he has participated in dozens of national and international research projects. He is the author of different book chapters and has presented more than twenty research papers in different international congresses. In addition, it has numerous scientific publications in international impact journals indexed in the JCR reference ranking. He has been awarded the first prize of the Open Data Contest organized by the Junta de Castilla y León (Spain) in 2013, and with the first innovation award for the best research project awarded by the Junta de Castilla y León (Spain) in 2016. He is currently a Professor at the Pontifical University of Salamanca.