Dissertação de Mestrado

Inatel

An IoT-based Face Recognition Solution Using a Residual Network Model for Deep Metric Learning

Eduardo Henrique Teixeira

Janeiro - 2021

AN IOT-BASED FACE RECOGNITION SOLUTION USING A RESIDUAL NETWORK MODEL FOR DEEP METRIC LEARNING

Eduardo Henrique Teixeira

Dissertação apresentada ao Instituto Nacional de Telecomunicações, como parte dos requisitos para obtenção do Título de Mestre em Telecomunicações.

Orientador: Prof. Dr. Samuel Baraldi Mafra
Coorientador: Prof. Dr. Joel José Puga Coelho Rodrigues

Santa Rita do Sapucaí
2021

Teixeira, Eduardo Henrique
T235a

An IoT-based Face Recognition Solution Using a Residual Network Model for Deep Metric Learning. / Eduardo Henrique Teixeira. – Santa Rita do Sapucaí, 2021.

110 p.

Orientadores: Prof. Dr. Samuel Baraldi Mafra e Prof. Dr. Joel José Puga Coelho Rodrigues.

Dissertação de Mestrado em Telecomunicações – Instituto Nacional de Telecomunicações – INATEL.

Inclui bibliografia e anexo.

1. Internet of Things 2. Face Recognition 3. ResNet 4. Deep Metric Learning 5. Accuracy 6. Mestrado em Telecomunicações. I. Mafra, Samuel Baraldi. II. Rodrigues, Joel José Puga Coelho III. Instituto Nacional de Telecomunicações – INATEL. IV. Título.

CDU 621.39

Ficha Catalográfica elaborada pelo Bibliotecário da Instituição
CRB6-2718 – Paulo Eduardo de Almeida

An IoT-based Face Recognition Solution, using a Residual Network model for Deep Metric Learning

Dissertação apresentada ao Instituto Nacional de Telecomunicações – Inatel, como parte dos requisitos para obtenção do Título de Mestre em Telecomunicações.

Trabalho aprovado em______/______/______ pela comissão julgadora:

____

Prof. Dr. Samuel Baraldi Mafra
Orientador – Inatel

____

Prof. Dr. Joel José Puga Coelho Rodrigues
Coorientador – UFPI

____

Prof. Dr. Antônio Oseas de Carvalho Filho
UFPI

____

Prof. Dr. Felipe Augusto Pereira de Figueiredo
Inatel

____

Prof. Dr. José Marcos Câmara Brito
Coordenador do Curso de Mestrado

Santa Rita do Sapucaí-MG – Brasil

2021

“If I have seen further it is by standing on the
shoulders of Giants”

Isaac Newton

Dedication

To my parents, Maria Izabel de Freitas Teixeira, and Nelson Lazaro Teixeira, who have always supported me in every way so I could walk my way here.

Acknowledgments

I thank the National Telecommunications Institute (Inatel) for the opportunity to course this master's degree. I thank the professors and staff of the university, especially my advisors Prof. Dr. Samuel Baraldi Mafra and Prof. Dr. Joel J. P. C. Rodrigues for devoting their time and knowledge in the construction of this work.

I would like to thank my undergraduate professors for transmitting to me the necessary experience and encouragement to continue my studies. I am grateful for my friends and colleagues for their good times together and for all the help during my training. In particular, Mariana A. Ferreira, with whom I have had a long time of dating and friendship. And to all my family members who have somehow been able to contribute to this long journey.

I acknowledge the Coordination for the Improvement of Higher Education Personnel (CAPES), which provided me with the financial support necessary to carry out this project. And finally. I thank Inatel again, for the scholarship and the opportunity to work in the Teaching Internship Program.

Abstract

Biometric identification has been widely used in recent years, mainly because they represent more secure authentication systems than conventional ones. In this context, facial recognition is highlighted since it allows detecting and recognizing a person in real-time for their facial characteristics. This technology is particularly important and used in many applications such as smart surveillance. The evolution in surveillance technologies, thanks to Internet of Things (IoT), allows greater automation of this process since many monitoring functions performed by people can be replaced by real-time recognition techniques, turning the system even smarter, giving more information to the user, or increasing security in monitoring environments. It is noted that society is at a point where different types of technologies are converging and adding up. It is known that computer vision techniques are being incorporated into surveillance systems and deep learning models have proven innovative in solving various visual recognition problems. In this sense, this dissertation proposes a surveillance system, which uses these techniques to identify the individuals present in the vision field of a camera through a combination including Histogram of Oriented Gradient (HOG), Support Vector Machine (SVM), and a deep learning model, called ResNet (Residual Network). The set of detection and recognition techniques was deployed in a hardware with limited processing power, quite common in IoT devices. The idea is to demonstrate that even under these conditions, the proposed architecture still manages to work with high precision and in real-time. To achieve the proposed objective, experiments were carried out in different scenarios to verify the accuracy and robustness of the techniques adopted under different conditions. Two techniques were used in the detection scenario, but only one was carried out in the experiments since it consumes 20 times less processing time when compared to the second. The accuracy of the ResNet model used reached about 99.38% in the Labeled Faces in the Wild (LFW) Benchmark while it manages to deliver a rate of 1-3 fps (frames per second), showing excellent results, especially considering an embedded system. The performance evaluation of the system against different types of noise showed high invariability with darkening of the images and high precision and robustness against blur type interference.

Keywords

Internet of Things; Face Recognition; ResNet; Deep Metric Learning; Accuracy; Edge Computing.

Resumo

A identificação biométrica tem sido amplamente utilizada nos últimos anos, principalmente por representar sistemas de autenticação mais seguros que os convencionais. Neste contexto, destaca-se o reconhecimento facial que permite detectar e reconhecer uma pessoa, em tempo real, pelas suas características faciais. Essa tecnologia é particularmente importante e usada em muitas aplicações, como vigilância inteligente. A evolução das tecnologias de vigilância, graças à Internet das coisas (do Inglês, *Internet of Things* – IoT), permite maior automação desse processo, já que grande parte das funções de monitoramento desempenhadas por um ser humano podem ser substituídas por técnicas de reconhecimento, tornando o sistema ainda mais inteligente, dando mais informações ao usuário ou aumentando a segurança em ambientes de monitoramento. Nota-se que nossa sociedade está em um ponto em que diferentes tipos de tecnologias estão convergindo, as técnicas de visão computacional estão sendo incorporadas aos sistemas de vigilância e que os modelos de aprendizado profundo têm se mostrado inovadores na solução de diversos problemas de reconhecimento visual. Nesse sentido, esta dissertação propõe a construção de um sistema de vigilância que utiliza essas técnicas para identificar os indivíduos presentes no campo de visão da câmera por meio de uma combinação de Histograma de Gradiente Orientado, Máquina de Vetores de Suporte e o modelo de aprendizagem profunda, ResNet. O conjunto de técnicas de detecção e reconhecimento foi implementado em um hardware com poder de processamento limitado, muito comum em dispositivos IoT. A ideia é demonstrar que mesmo nessas condições, a arquitetura proposta ainda consegue trabalhar com alta precisão e em tempo real. Para avaliar o desempenho da solução proposta para atingir o objetivo deste estudo, foram realizados experimentos em diferentes cenários para verificar a precisão e robustez das técnicas adotadas nas diferentes condições. Duas técnicas foram empregadas no cenário de detecção, porém apenas uma foi levada adiante nos experimentos devido ao fato de consumir 20 vezes menos tempo de processamento em comparação com a segunda. A precisão do modelo ResNet utilizado alcançou 99,38% no LFW (do inglês, *Labeled Faces in the Wild*) Benchmark, enquanto consegue entregar uma taxa de 1-3 fps (do inglês, *frames per second*), apresentando ótimos resultados, principalmente levando em consideração um sistema embarcado. A avaliação do sistema contra diferentes tipos de ruído demostrou alta invariabilidade com escurecimento das imagens e alta precisão e robustez contra interferência do tipo “*blur*”.

Palavras-chave

Internet das Coisas; Reconhecimento Facial; ResNet; Aprendizado Métrico Profundo; Precisão; Computação de Borda.

Table of Contents

List of Figures

List of Tables

List of Abbreviation and Acronyms

AI	– Artificial Intelligence
ANN	– Artificial Neural Network
CCTV	– Closed-Circuit Television
CNN	– Convolutional Neural Network
CV	– Computer Vision
DBMS	– Data Base Management System
DCNN	– Deep Convolutional Neural Network
DML	– Deep Metric Learning
FR	– Facial Recognition
FPS	– Frames Per Second
GPU	– Graphics Processing Unit
HOG	– Histogram of Oriented Gradients
ICA	– Independent Component Analysis
IEEE	– Institute of Electrical and Electronics Engineers
ILSVRC	– ImageNet Large Scale Visual Recognition Challenge
IoT	– Internet of Things
IoMT	– Internet of Multimedia Things

IP	– Internet Protocol
KNN	– K Nearest Neighbors
LBP	– Local Binary Pattern
LBPH	– Local Binary Pattern Histogram
LDA	– Linear Discriminant Analysis
LFW	– Labeled Faces in the Wild
MLP	– Multilayer Perceptron
MMOD	– Max Margin Object Detector
ReLU	– Rectified Linear Unit
ResNet	– Residual Network
PCA	– Principal Component Analysis
SGD	– Stochastic Gradient Descent
SIFT	– Scale Invariant Feature Transform
SURF	– Speeded-Up Robust Features
SVM	– Support Vector Machine

1. Introduction

1.1 Motivation

Internet of Things (IoT) has been widely applied in facial recognition through unmanned aerial vehicle, intelligent classroom, home security system, smart home, intelligent surveillance, among others [1]. It is known that visual analysis applications will show enormous growth in the next 5 years - more than 50% - which will generate a positive economic impact from US $ 3.9 trillion to US $ 11.1 trillion annually until the end of the year 2025 for the IoT [2]. In addition, new methods of facial recognition are object of study by several authors around the world, being a topic of great importance for society [3]. This becomes evident when searching in the Institute of Electrical and Electronics Engineers (IEEE) database, the IEEExplorer, with the keywords "face recognition". This research results in 28,095 papers published in Conferences, Journals, Books, Magazines, Courses, and Early Access Articles, of which most (18,935) were published in the last decade (2010-2021) as it may be seen in Figure 1 [4].

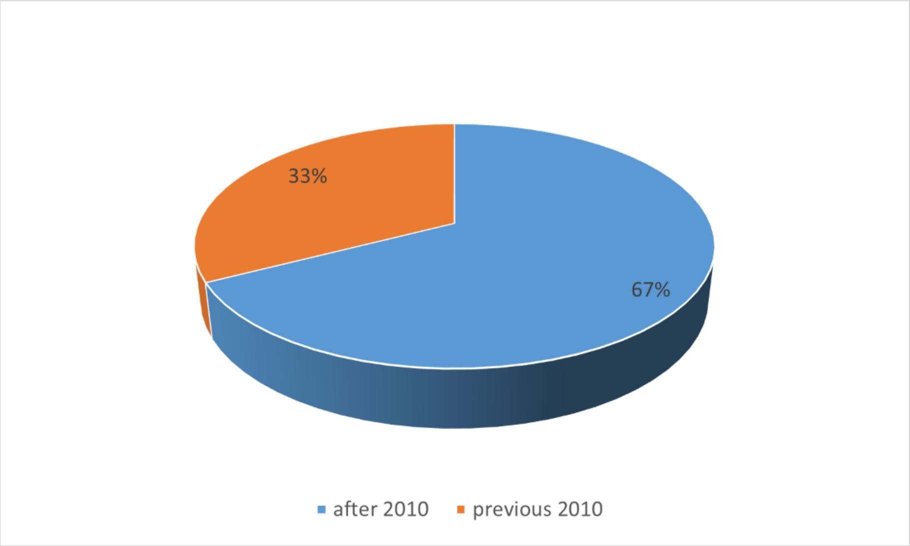


Figure 1 - Percentage of scientific work about face regonition on IEEE.

At the same time, searching in the same database with the keywords ‘"face recognition" AND "IoT"’ about 220 papers are found, all published after 2010. It confirms that research that integrates these two technologies are very recent. Furthermore, it is noted that this is a topic that has been increasingly explored, since 115 are from the pass year (2019-2020), corroborating the exponential interest in this topic [4].

1.2 Problem Definition

Human beings are able to identify people through their faces, and they do this very well, even in places with variations of lighting, pose and facial expressions [3]. Since the dawn of our civilization, characteristics such as the face and the voice have been essential for mutual recognition. Such characteristics, which are unique in each person, have provided what we know today by biometric identification.

Evidence shows that since centuries ago, biometrics has been used to identify people. As an example we have the head of the criminal identification division of the Paris police, Alphonse Bertillon, who in the 19th century already used body measures to identify criminals [5]. Since then, with the creation and evolution of computational techniques, it is possible to perform demographic estimations automatically [6], to obtain approximate data of age, sex or race through images, and even to create systems that operate in real-time, capable of identifying people with a high degree of accuracy [7].

Biometric identification has, fundamentally, the objective of recognizing a person through technologies by a single and quantifiable parameter, e.g. physical or behavioral characteristics [8]. Its methods can be classified as intrusive, e.g., the use of iris and fingerprints, or non-intrusive, such as facial recognition. The difference between the two methods is due to the need or not of the consent of the person to be analyzed [9]. Therefore, biometric identification offers a much more promising security system than traditional authentication methods, such as passwords and identity cards, which can be easily forgotten, cloned, stolen or lost [1], [10]. This, in addition to recent advances in technology, has fostered diverse research in the field of biometrics, resulting in several initiatives focused on researching innovative and efficient systems for biometric identification [9].

In the context of biometric identification methods, the use of facial recognition and fingerprints stands out as the most common mainly because of their uniqueness and consistency over time [11]. The use of both technologies is seen in an increasingly common way in banks, cell phone applications, monitoring systems, and access control, i.e., it is already part of the daily life of today's society.

With the growth in the flow of multimedia on the Internet through real-time applications, videoconferences, videos on demand, telepresence and content delivery in real-time [12], it is notable that, today, society is in a point where different types of technologies are converging and adding up, which facilitates the use of data collected from various sources [2]. Along with this, exists the Internet of Things (IoT), which acts significantly in everyday life and in the way we interact with all the physical objects present around us. This network of new devices capable of exchanging information is enabled by new communication technologies and internet protocols [13]. As a result, we have sensors, including biometric ones, which can collaborate autonomously within a system, without the need of human involvement.

With the exponential growth of IoT observed in recent years, the interaction of multiple devices is imperative [14]. In an IoT-based system, intelligence and drive capabilities are incorporated into devices with the help of sensors and actuators. At the same time, a cloud allows developing, maintaining, and executing different services, providing scalable computing and storage resources. Such system allows users the ability to monitor and control devices at anytime from anywhere [12]. However, IoT devices has limited memory and processing capabilities, and their systems cannot successfully connect everything if they are not capable to include all the ‘multimedia things’. Today's world and the rapid growth of multimedia traffic in IoT is leading to new techniques to satisfy this constraint [12], [14]. Therefore, the Internet of Multimedia Things (IoMT) represents a specialized subset of IoT, that enables the integration and cooperation among heterogeneous multimedia devices with distinct capabilities, computational characteristics, and resources. It is a technology that requires bigger memory, higher computational power, and consumes a lot of energy with higher bandwidth, and its main quality is the timely and reliable delivery of the data, which implies strict quality of service requirements, and demands efficient network architecture [12], [14]. Also, according to [14], the IoMT has contributed to areas such as industry, smart home and health, traffic, real-time multimedia and smart surveillance, the latter being the scope of this work. For example, in a potential robbery on any street, a smart surveillance system makes it possible to observe in which direction the suspect went and his person identification by facial information sent to an identity database, to access the suspect's personal data, as well as his criminal record.

A facial recognition system allows to detect and recognize in real-time a person by their facial characteristics. This technology is very important and is used in many applications for various purposes. Thus, a facial recognition system can be installed to monitor and identify people in public or restricted areas for providing security control, photos matching, user verification, user access control, etc. For example, facial recognition can be used to validate attendance during classes in a university [15]. Another applicability of this system is to locate suspects, in real-time, using tracking and identification techniques [16].

All the facial recognition technologies can be included in the field of computer vision. A science, with a set of tools, which allows a device to process and analyze real-world images. A process similar to what is done by the human eye in conjunction with the brain. Currently, several computer vision techniques are incorporated into surveillance systems. One of its advantages is to avoid continuous monitoring of images from being performed by a human. Since it is possible to program the algorithm to display only the relevant information. Also, it is known that deep learning models, based on artificial intelligence, innovated in a singular way in the solution of several problems of visual recognition. Among the models of deep learning, the deep neural networks, more specifically the Convolutional Neural Networks (CNN) have been extensively studied [17]. It is a supervised learning model that, in recent years, has become the state of the art in several applications in the field of computer vision, such as object detection, image classification and people recognition.

Several studies have demonstrated the many advantages of using CNN for facial recognition, due to the perception of patterns with high variability and robustness to distortions, and the perception of simple geometric transformations such as scaling, rotation and noise [1], [18]. Among the different CNN models, we can mention ResNet, which has shown excellent results for facial recognition and wide use in object classification [19]–[21]. Therefore, the present proposal for a master thesis is to develop a real-time facial recognition system that is connected to an IoT environment.

1.3 Research Objectives

The main objective of this dissertation is to deploy and analyze a recognition system, based on IoT. Thus, different techniques of facial detection and recognition are explored as well as an analysis of several important aspects for their choice and deployment. To achieve this main objective, the following partial objectives were defined:

- Review the related literature on IoT-based Face Recognition Systems and Embedded Facial Recognition Systems;
- Proposal of a real-time detection and recognition solution;
- Prototype construction and integration with a Database Management System and an IoT middleware platform;
- Performance evaluation, demonstration, and validation of the complete solution.

1.4 Main Contributions and Publications

The main contribution of the research presented in this master's thesis is to advance the state of the art in evaluating the benefits of integrating a facial recognition system with an IoT middleware. In addition, an evaluation of the types of databases that can be used with the proposed IoT system, demonstrates the advantages obtained in this combination of technologies. Plus, produce new insights and knowledge by analyzing the processing time for the different stages of the system and the robustness of detection and recognition techniques in different types of noise introduced in the photos, such as low lighting and distortions.

During this research, one scientific paper has been published in a National Conference:

- E. H. Teixeira, S. B. Mafra, J. J. P. C. Rodrigues, A. A. Werner, N. Silveira, and O. Diallo. “A Review and Construction of a Real-time Facial Recognition System,” in *XII Simpósio Brasileiro de Computação Ubíqua e Pervasiva* (SBCUP 2020), *LX Congresso da Sociedade Brasileira de Computação* (CSBC 2020), Cuiabá, MT, Brasil, 16-20 de Novembro, 2020, pp. 191–200.

This paper presented contributions when comparing the processing time of different techniques under the same conditions and in the matter of choosing the detection and recognition architecture that allows a real-time operation.

1.5 Thesis Statement

Smart surveillance systems are a key topic in the society. Several have adopted this technology to improve security systems. An important part of any security system is the person identification mostly done by facial recognition. Therefore, this study aims to investigate a surveillance system based on IoT system, exploring facial detection and recognition techniques embedded in a device that performs them in real-time. Important aspects for the choice and implementation of the proposed architecture were analyzed, such as processing time, accuracy, and robustness to demonstrate the effectiveness of the system in the real-time recognition task and show that even under limited process conditions, it still manages to work with high precision.

1.6 Document Organization

The dissertation is organized in six chapters. Chapter 1 contains the motivation for the study and the problem definition, main objectives, research objectives, main scientific contributions and publications, and conclude with the document organization.

Chapter 2 presents a deep review of the state of the art in the topic and the most relevant concepts for the study, such as the IoT, computer vison for processing facial images, their contributions, applications and evolution in relation to more recent techniques.

Chapter 3 shows the detection and recognition techniques for the proposed system. Going through the face detection methods, the align and normalization factor construction, the recognition stage based on a residual network, the training stage procedure and the database registration steps.

Chapter 4 covers the IoT infrastructure for this proposal, which enables connections and storage, presenting sections of tools used in cooperation with FR systems, such as the software middleware and the database management system.

Chapter 5 exposes the experiments and results found in several stages of the proposed architecture. Thus, presenting the six different scenarios used to evaluate the system performance and the variables analyzed in each one (detection efficiency, average detection time, recognition accuracy, average recognition time, assessment of blur interference and of light interference).

To finish, Chapter 6 presents the main conclusions of this study, along with the lessons learned, and insights for future work.

2. Related Work

This chapter presents the main concepts necessary for the development of this research. Some related works are compared to the proposed solution and an analysis of some relevant points of these works are addressed.

2.1 Background on Internet of Things

Internet of Things (IoT) applications require a connection for data exchange and are usually present in devices at the edge of the network. As a promising edge technology, it can be applied in several segments (e.g., smart cities, rural environments, residential applications, and security and monitoring scenarios), offering greater interaction and connectivity between different devices and applications [22].

The use of an IoT infrastructure allows that through these connections, devices can have greater intelligence and autonomy in their decision making. One of the systems that has undergone considerable evolution in recent years, has been video analysis. In this application, a traditional architecture may not be able to support the exchange of video from many cameras. As a result, several IoT applications have been proposed in which system intelligence is applied directly to the final device [23]. In these cases, the traditional system has been replaced by a system that uses computer vision techniques to implement automatic video analysis. This processing is performed on the local device itself and with that, it sends only the relevant data instead of the frames [24]. Then, the result of the video analysis can be stored for use in other applications. The accumulation of this data from IoT devices can also be used for big data analysis [22].

The selection of computer vision techniques that will be used depends on the specific requirements of each system and its adequate choice is of the utmost importance. In some cases, such as surveillance, requirements such as accuracy, processing time and robustness to variations in image quality, must be met for the system to respond efficiently. Thus, the correct choice of technique, together with the optimization of resources, can be the key point in being able to identify people in uncontrolled images [25].

2.2 Computer Vision

Computer vision (CV) is a recent field of computer science that seeks to find useful information from real objects and scenes detected in images. It is an area that has presented significant growth in the last 25 years and that has been the scope of several researches [26].

The goal of this emerging field is to build artificial systems that can extract information from images, that is, to make computers understand images and videos, through techniques of acquisition of scenes (sequence of videos, depth images, visualizations of multiple cameras and even multidimensional data of image sensors), processing, analysis and, finally, interpretation of captured images or videos [27].

CV seeks to mimic the human visual system, causing computers to process and identify images and videos in the same way. Thus, the structures of computer vision and human vision have similarities, that is, both have sensors that convert the light transmitted in the scene into an image, in addition to a mechanism of processing and interpretation [26], [27]. Briefly, in CV, the input is an image, and the output is the interpretation of a scene This way, computer vision allows to describe a real-world scene in one or more images, identify and reconstruct its properties from quantitative measurements, and produce an interpretation that will be used for a final decision making.

In [26], Bhuyan discusses some essential techniques and applications of computer vision, such as: machine learning algorithms and their applications in medical image segmentation; motion estimation and object tracking, – widely used in video surveillance systems – face and facial expression recognition; gesture recognition and image fusion, the latter, widely used in surveillance. This is due to the incessant pursuit of security systems in reducing vulnerability, dealing with risks, and avoiding accidents. CV, through sensor-based devices, facilitates the monitoring of any suspicious activity in industrial sites or other restricted locations [28].

Another way to create a more secure environment is to use machine learning technologies with network technologies. By creating different scenarios, the algorithm or technique can differentiate between safe and dangerous scenarios and detect moving objects in real-time, through devices embedded in sensors that can easily detect light waves across multiple spectrum ranges [29].

There are sensor devices, which capture as much detail as possible from an image, including infrared data and more distant images with greater accuracy, and so-called interpretation device, which process image information and extract senses from it [2]. Thus, the behavior of an object is detected automatically and an intelligent surveillance system emits an alarm in case of any suspicious, threatening or illegal activity. Intelligent visual surveillance systems, offer safe monitoring of human activities, forestry, natural environments, human-machine interactions (HMI), content-based video encoding and many other areas [26].

Many of the visual security systems still rely on a human operator to detect and monitor illegal and suspicious activity. This conventional technology works with traditional video motion detection through background changes and alerts security officials to suspicious activity [30]. However, this approach is subject to many errors, as natural elements such as dynamic backgrounds, slow leaf movements, and cloudy environments alter video characteristics, resulting in false alarms. Another major disadvantage is that this conventional technology cannot be used for large amounts of data, especially in metropolitan cities where the number of objects is larger, making it very costly and unsustainable [30].

With improved processing capacity, sensor-based devices and independent transport systems, there are now fully automatic security routines – data sent to the cloud and visual analysis are applied to track breaches – that are shown to be better than their human counterparts as the number of false alarms becomes smaller [31].

It is known that in the past, computer vision applications were limited only to selected platforms. Today, there are many types of new technologies converging to help individuals or interested people use the video data that has already been collected, as well as data collected from new sources [26].

When combined with devices connected to the Internet Protocol (IP), it allows you to create a set of real-time based applications. Such techniques added to advanced data analysis and Artificial Intelligence (AI) have led to a revolutionary journey into innovations. Video-transmitted data is an increasingly rapid sphere of big data or cloud computing along with IoT projects , and its importance is increasing to correspond to its profusion, being considered the great engines of this new technological world [32].

Such as intelligent surveillance systems that require the use of control systems and IoT technologies that allow users to monitor their private domains from anywhere with the help of Internet resources and sensor-based mobile devices. Still, these technologies help smart cities, driverless cars, and emergency departments. And even other lesser-known examples including Closed-Circuit Television (CCTV) footage and sensor-based devices within the IoT widely used at airport check-in, a reader examines a passenger's boarding pass and a camera examines that person's face, without the video data that the camera captures, the system would be less useful [33].

The potential for visual IoT is enormous: identifying suspicious people in public domains or for access control, and effectively monitoring object movements to ensure that no suspicious activity is occurring [34].

2.3 Facial Recognition and Detection

According to [1], Facial Recognition (FR) technologies currently involve countless researchers and is increasingly surpassing other biometric identification tools. This is because when using the human face, FR becomes a stable technology, which provides a high precision and a lower error rate than the others, since the human face changes little throughout life.

FR can be divided into steps such as: identifying, distinguishing and processing the face [35]. Still, this can be done in two ways: the first is by checking, where the current image is compared to a specific face requested earlier. And the second, by identification, where the current image is compared with multiple images within a database in order to determine some similarity[7].

The facial detection method should locate points of interest such as the eye’s contour. These points are used for an algorithm to make facial alignment, thus allowing a more precise clipping of the region of interest, which will later be classified [3].

Among the best known techniques of facial detection and recognition (and CV) [36], it is worth highlighting the Viola-Jones algorithm, widely used for its efficiency in computing time, something that allowed its application in real-time systems [37].

2.3.1 Classical Approaches

Several approaches can be used for the detection and recognition of human faces. Among the most well-known techniques, it can be mentioned the Haar-cascade algorithm, developed by Viola and Jones [36]. This algorithm is based on Haar-type features that use cascading classifiers and provide human face detection regardless of background conditions, size, shape, and color of the image [38]. It is widely used for its efficiency in computing time, which allowed its application in real-time systems [36].

In [36], the authors bring three main contributions: a new image representation that allows detector resources to be calculated quickly; a simple and efficient classifier to select a small number of critical visuals together very large of potential features, and finally a method for combining classifiers into a "cascade" that allows background regions of the image to be quickly dropped and focuses sorting on face-like parts. Such work has been so successful that it is used to this day because it does not require very sophisticated hardware and has excellent results [37], [39]–[43].

There are also appearance-based techniques such as Principal Component Analysis (PCA), Linear Discriminatory Analysis (LDA), Independent Component

Analysis (ICA), and Support Vector Machine (SVM). These techniques use *machine learning* to learn the characteristics of the model [44]. Although these techniques have good results, they are relatively difficult to implement, mainly due to the diversity of samples needed in the training phase.

Feature descriptors are also applied as methods for FR. In a comparison between three descriptors of facial features: Histograms of Oriented Gradients (HOG), Gabor and Local Binary Pattern (LBP), [35] presents the advantages and disadvantages of using each method.

Considering embedded implementations, different techniques applied during the identification phase can also be noted. The Table 1 shows some systems implemented in a Raspberry Pi, where the authors use different detection and recognition techniques and the proposed solution in this work. In addition, the last column explains the destination of the recognition data applied for each one.

Table 1 – Summary of the most relevant related works.

Contributions	Detection Technique	Recognition Technique	Applications
Patil et al. 2014, [46].	Haar Cascade	PCA + LDA	DBMS
Yadav et al. 2017, [47].	Haar Cascade	PCA	Display LCD
Sajjad et al. 2017, [48].	Haar Cascade	SVM	Cloud Framework
Wazwaz et al. 2018, [49].	Haar Cascade	LBP	Server App
Nagpal et al. 2018, [50].	Haar Cascade	LBPH	Display OLED
Espinosa et al. 2019, [51].	Haar Cascade	LBP	Smartphone App
Purohit et al. 2019, [52].	Haar Cascade	CNN (ResNet)	Web Application
Proposed Solution	HOG + SVM	CNN (ResNet)	Middleware / DBMS

As may be seen in [46], the authors propose a face detection and face recognition system to marking the student's attendance in a classroom, to do that, they use DBMS with name and number of the students, in addition to facial images. In [47], the authors seek to verify the technique performance in gender identification, and the information is presented on a Display LCD. To assist law enforcement agencies, [48] propose a face recognition framework where the classifier is stored and trained on the cloud. The [49] research's offer a security system to public places (Malls, Universities, and airports) applied in a server app.

The contribution of [50] is to develop a system to identify suspicious individuals using a Display OLED to show the data output. Also [51] presents a design of a vehicle security and alert system based on facial recognition to identify the driver, the system interacts with the car´s owner by a smartphone app. The last contribution is by [52], were they developed a system using a smart speaker and mirror that can also display the information on the mirror itself, in order to improve usability of the home automation system among the smart home system in a web application.

When considering an IoT scenario, the use of specific applications - developed with the intention of serving only a single device - can bring particularities that hinder interoperability with other equipment. This incompatibility occurs in some cases where there is the use of communication protocols or technologies that are not common to all platforms [34]. To solve this problem, the proposed system concentrates the data in a database and in an IoT middleware. Thus, the development of applications becomes more malleable in relation to the use of different protocols since access to this information is facilitated by the two technologies in which the data is concentrated [53].

2.3.2 Artificial Neural Network

An Artificial Neural Networks (ANN) works analogously to biological neural structures, but are mathematical models with computational capacity acquired through hundreds or thousands of processing units c. The classic models of these networks are biologically inspired, being the isolated processing units, neurons, which when connected are capable of a data processing structure. They are organized into connected layers, which allow an input to pass through the network to the output layer, providing the response to that input.

ANN has been applied in the resolution of various tasks within the scope of computer vision. The basic principles of operation of these networks have been known for a long time, but in recent years, the interest in using these networks in a practical way has been increasing considerably [26], c. One of the main reasons for this growing attention by these networks is due to the great accuracy they present in classification challenges, such as the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [55], [56]. Another reason is that the greater processing power of the devices nowadays that facilitates the feasibility of applying this network. In addition, ANN has a variety of applications, such as image classification, character recognition, object detection, speech recognition, noise removal, image processing, instance segmentation, among others [57].

As mentioned earlier, ANNs replicate the biological form of the human neuron, and therefore it is common to group these artificial neurons to perform more complex operations [54], [58]. Figure 2 represents an ANN, where we can observe the entries in blue, the green circles are the hidden artificial neurons, and the red ones symbolize the output neurons.

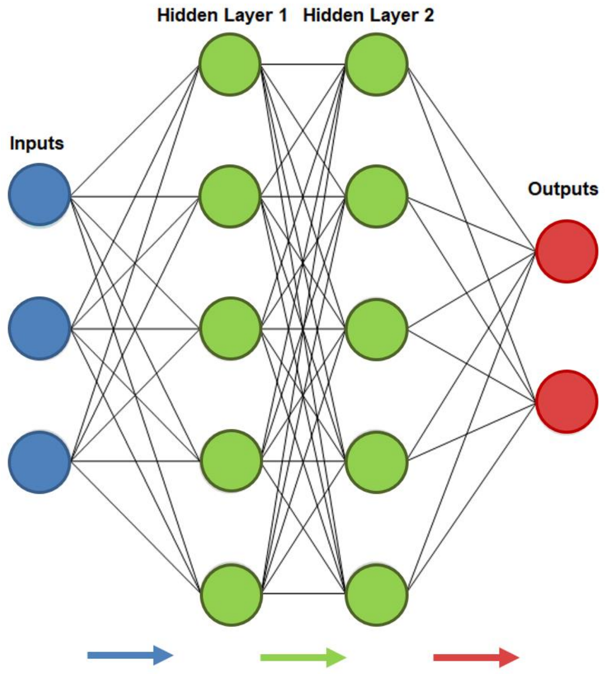


Figure 2 - Exemple of Artificial Neural Network.

In the simplified way, a neural network is formed by a sequence of fully connected layers, and each neuron receives its own weight configuration during the training stage. In this procedure, the parameters of each neuron are learned automatically through machine learning [54].

Considering supervised learning, the first step is to propagate multiple entries across the network and compare the responses obtained with the right response. For example, images labeled on the input are propagated over the network so that a weight adjustment occurs, capable of making any generic image applied to the network input able to be classified as to its type [58].

This weight update technique is called backpropagation [58], in it, as its name says, a backpropagation of the error derivative occurs, that is, the values of the network parameters are adjusted by means of an optimization method, such as stochastic descending gradient. This process is iterated until the error is decreased to the smallest possible value. Lastly, we have the last layer, in which the circles in red can match the classes and it is through it that the network expresses the result of its operations.

Another important point for correct training is the size of the labeled input dataset since the training is done when you know the correct answer to each input possibility and the network uses this same information for weight adjustment. Training an ANN with partially labeled or unlabeled outputs is also possible in some applications, but the focus of this work will be on supervised learning [57].

Analyzing the application of ANNs in images, we can perceive a high number of weights. To exemplify, considering an image of size 150x150x3 (150 pixels wide, 150 high, and 3 color channels), we already have $150 * 150 * 3 = 67500$ weights only considering neurons in the input layer, without considering the other layers [59]. Thus, it is possible to note that for larger image sizes, the number of weights tends to increase rapidly. Considering the weights of the posterior layers, we can say that the number of neurons present in each layer and the depth of the network are also strongly linked to the computational cost of the network during the training stage, as they directly affect the amount of weights to be adjusted.

The evolution of computational resources in recent years has facilitated the practical application of ANN in real problems. The favoring use of this network is due to recent advances in hardware, such as the emergence of general-purpose computing in GPUs, which has provided a major reduction in the training time of ANNs with complex architectures.

The "neurons", basic units of an ANN, are called Perceptron. In this module, each input connection has an associated weight, so the more important the input is, the greater the weight tends to be. Conclusively, the perceptron output is the weighted sum of the input data, following a nonlinearity function to avoid the singularity problem that the sum generates, improving the generalization capacity of the model [60].

Perceptron contains only one neuron, being quite common in linearly separable system applications such as showed in Figure 3. The neuron is able to create a separation hyperplane between two classes, while the output is responsible for determining which of the two regions each data entered in the input belongs, so that the function performs well in returning the desired outputs as specified in the task [60].

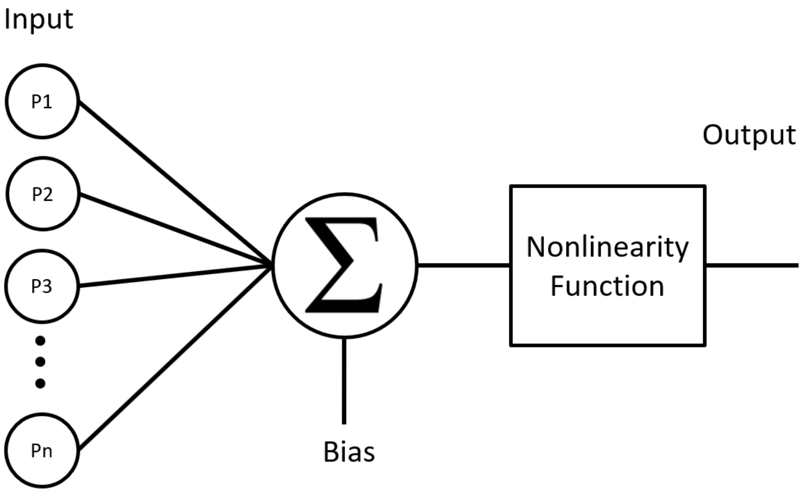


Figure 3 - *Perceptron with one neuron.*

2.3.3 Convolutional Neural Network

Convolutional Neural Network (CNN) can be defined as a type of neural network that has the convolution operation in one or more layers [58], being a specialized network type for processing data that has a grid-shaped layout.

CNN architectures have achieved promising results in applications in computer vision, since this field of study is based on images, or in a simplified way, on a grid of pixels. This makes the architecture more efficient since certain specific properties of image processing are added to the network structure. In this context, in terms of architecture, a conventional ANN performs less than CNN, mainly because they do not scale well to images [61]. CNN are specially designed for visual signal processing and feature specific image processing properties in your network structure, making architecture more efficient.

CNN has deep learning architectures that are like Multilayer Perceptron (MPL) networks, where neurons with weights that adjust during the training stage. The neurons of a CNN make the analysis of three dimensions: height, width, and depth, having connections only with neurons of the anterior layer that are in a nearby spatial position, called region of interest or receptive field. This visual signal processing property favors CNN compared to ANN, since traditional networks have their basic processing units fully connected [62].

ANN does not consider the spatial structure of the input pixels, considering them all as isolated units, something unfavorable in image analysis, where the distance of the pixels and their spatial structure can provide relevant information during the analysis. This property is based on the biological process contained in the visual cortex of animals, when a receptive field is observed [63].

As may be seen in Figure 4, neurons in a layer do not connect to every input, but to just one region of it. Still, both networks of each neuron perform a weighted sum of the inputs with their respective weights and then use an activation function to define the output value.

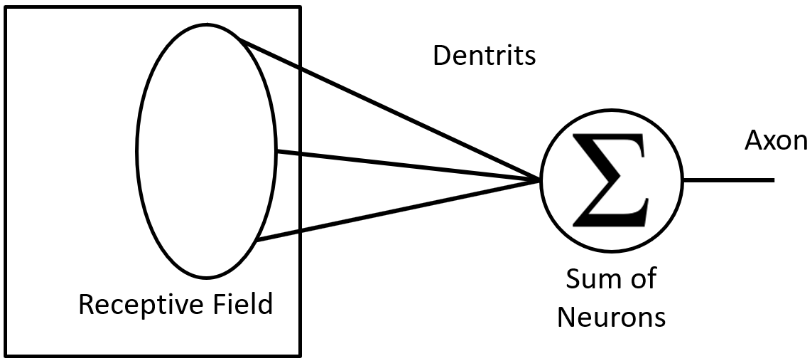


Figure 4 - *Receptive Field of a neuron.*

In addition to performing better than an ANN in the image processing task, CNN eliminates the need for manual extraction of characteristics because the network itself is able to decide what are the best parameters to consider during training, and is a powerful machine learning algorithm that delivers state-of-the-art recognition results and is able to be retrained to meet other demands, allowing them to be built from pre-trained networks machine learning [56].

CNN has a structure composed of several layers, where the convolutional layer is the key point and is present in all current models, something that differentiates them from the models of classical artificial neural networks. Figure 5 presents the architecture of CNN with its different layers. However, the model is simplified and presents only one layer of each type, and in fact the CNN models are composed of several layers distributed interleaved [59].

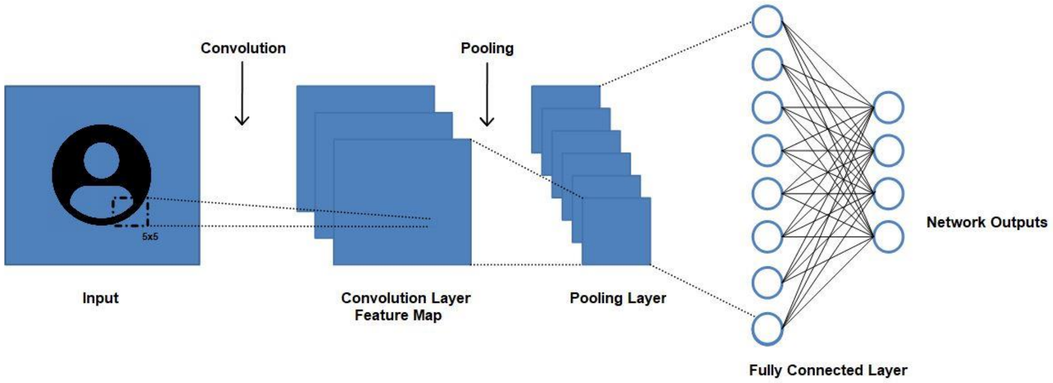


Figure 5 - *Basic blocks of a CNN architecture.*

Following this context, [64], [65] built the first CNN architecture applied to a practical operation, the recognition of numerical numerals, reaching at the time, the state of the art in solving this problem. Table 2 shows the architecture of this CNN, LeNet-5, where it is possible to observe, as mentioned earlier, the repetition of some types of layers and the alternation between them.

Table 2 - *LeNet-5 architecture.*

LeNet	Layer	Feature map	Size	Kernel Size	Stride	Activation
Input	Image	1	32x32	-	-	-
1	Convolution	6	28x28	5x5	1	tanh
2	Average pooling	6	14x14	2x2	2	tanh
3	Convolution	16	10x10	5x5	1	tanh
4	Average pooling	16	5x5	2x2	2	tanh
5	Convolution	120	1x1	5x5	1	tanh
6	Fully Connected	-	84	-	-	tanh
Output	Fully Connected	-	10	-	-	softmax

After the emergence of LeNet, there has been a great evolution that has enabled the creation of deeper and more complex architectures, with different layers and operations organizations [17]. Thus, the following sections present their respective structures, to explain in more detail, the functioning and basic components of a CNN, being them: convolution layer, pooling layer and activation function, and fully connected layer.

Convolution layer

Convolutional layers consist of several neurons responsible for applying filters on specific parts of the input image. They are composed of a set of neurons responsible for calculating the output values of each neuron connected to a certain region of the input volume through convolution with the filters contained in this structure, that is, it aims to extract a set of characteristics of the layer immediately preceding it. Thus, each neuron is connected to a set of neurons of the previous layer, and for each connection is assigned a weight, called a synaptic weight. The input weights of each neuron are combined with each other, producing an output that is passed to the next layer. The weights attributed to the connections between neurons play the role of a convolutional filter applied in the spatial domain [54].

The sharing of weights is the property responsible for enabling the application of neural networks in spatially dense data, such as images, by the significant reduction of parameters in the network. This reduction happens in comparison to the standard model of feedforward networks, since instead of each neuron receiving as input all elements produced in the previous layer, in CNN to produce each map of characteristics the number of weights is limited to the size of the filter applied (filter height x filter width x depth of the previous layer). Over the depth of filters, typically the first convolution layer processes a small amount of input channels (one for grayscale images, three for color channels, among others), while the other layers start processing the number of slices generated as feature maps. In this sense, the shallower layers of the network tend to learn lower-level characteristics, and as deeper layer convolutions are performed, the filters come around more complex characteristics, such as textures and parts of an object [59], [66]. They have a set of parameters that are adjusted during the network training process.

These filters are adjusted to be activated in the presence of relevant characteristics identified in the input, in a network training step. Convolutional layer neurons use local connectivity, see Figure 6, each neuron in the N layer is connected to only a few neurons of the N+1 layer, rather than connecting to all neurons in the layer. Neurons of the same layer are grouped together, and their outputs form characteristic maps.

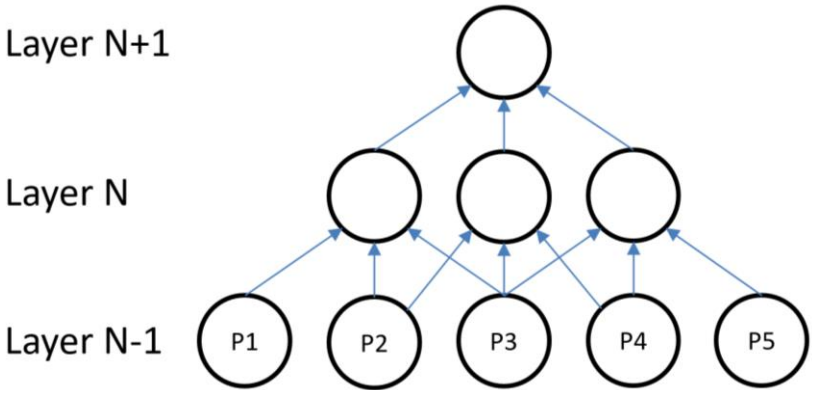


Figure 6 - *The blue silhouette in the N+1 layer represents a map of characteristics, defined as the grouping of neurons of the same layer.*

The ability to find a hierarchical structure of characteristics is the main reason why CNN works so well for pattern recognition in images. Hierarchical learning, exemplified by the adapted abstraction of [66] presented in Figure 7 demonstrates as the kernels (or convolutional filter) that are closer to the network input have more primitive characteristics, and as the network deepens, the later layers begin to display more complex representations.

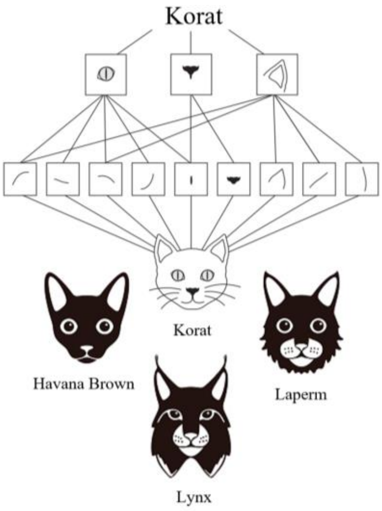


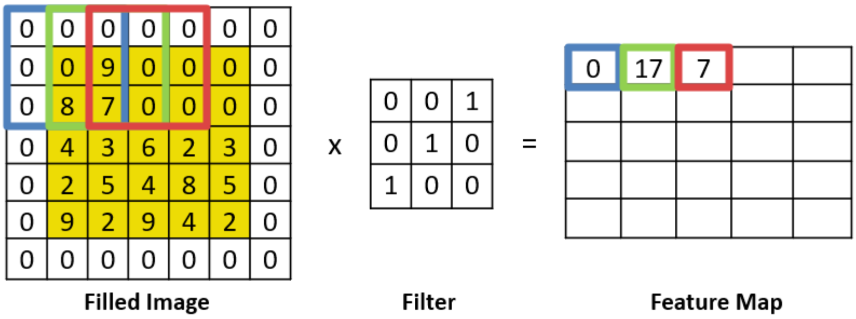
Figure 7 - *Representation of convolutional layers in a spatial hierarchy. Adapted from:* [66]

The kernel allows you to extract increasingly complex, abstract visual concepts from an image, creating representations of the image as highlighting edges, shape, texture, or some other characteristic that is pertinent [66], [67]. This is how the network learns to identify these specific structures, such as eyebrow, eye, nose, and mouth, because they represent useful partial information in identifying within an image. Therefore, applying filters on the input image is the most important part of a CNN, since it is in these layers that much of the processing is performed. The filters typically employed in these layers have reduced spatial dimension (height and width), but extend throughout the depth of the input volume, as an example, a kernel applied to the input layer can be 7x7x3 in size, i.e., 7 pixels wide and in height, and 3 pixels deep, which is the same number of channels present in an RGB digital image.

There are two parameters that can influence the size of the map characteristics, the padding and stride [67]. Stride (also used in pooling layers) in a simplified way, represents how fast filters will move through the image. The filter size will define the size of the neighborhood that each neuron of the layer will process. Therefore, this value defines the size of the jump in pixels between each fragment of the image, in other words, indicates the way the convolution will be performed. For example, when the stride value is equal to 1, the filter is moved one pixel at a time, when the value is equal to 2, the filter jumps two pixels at a time, and so on. This may produce smaller output volumes. The higher the stride value, the lower the output volume. By default, in convolutions and applications involving image recognition, the stride is set to l x l, moving 1 pixel at a time in one of the directions.

Depending on the size of the filter and step size there may be a need to evaluate the filter value in a region outside the image, when it is necessary to fill the edges of an image, we use the fill parameter to define the size of that border. For this process to occur, you must fill the region of the edges of the image with zeros. The number of rows and columns that will be populated are given by this parameter.

When using padding, that is, when elements are added at the edge of the input array with zero value, this allows to apply the filter to all elements of the input array.. All these parameters are important in the training stage. Figure 8 shows how the convolution operation is performed, where a filter with these parameters is applied over a given region of the image matrix.



0	0	0	0	0	0	0
0	0	9	0	0	0	0
0	8	7	0	0	0	0
0	4	3	6	2	3	0
0	2	5	4	8	5	0
0	9	2	9	4	2	0
0	0	0	0	0	0	0

0	0	1
0	1	0
1	0	0

0	17	7		

Figure 8 - *Partial convolutional layer, using a 3x3 filter*.

It is important to note that not all combinations of these factors are possible since the size of the output volume must be an integer. From the behavior of convolutional layers, it is possible to conclude that they work as extractors of characteristics, not as classifiers, and so there is a need to add layers fully connected to the end of the architecture. Also, it is worth highlighting certain characteristics of CNNs that contrast with certain paradigms of the models of shallow architectures. The first one would-be sparse connectivity. This type of connectivity happens since the kernel has much smaller dimensions than the input matrix, generating a local spatial correlation by applying a local connectivity pattern between neurons of adjacent layers. Unlike what we see in global MLPs architectures. In Figure 9 it is possible to verify that when the model makes use of Matrix Multiplication (MLP), in this, all output units are affected by input, and in the use of the convolutional layer, using a size 3 kernel, only three units of the output layer are affected by the input.

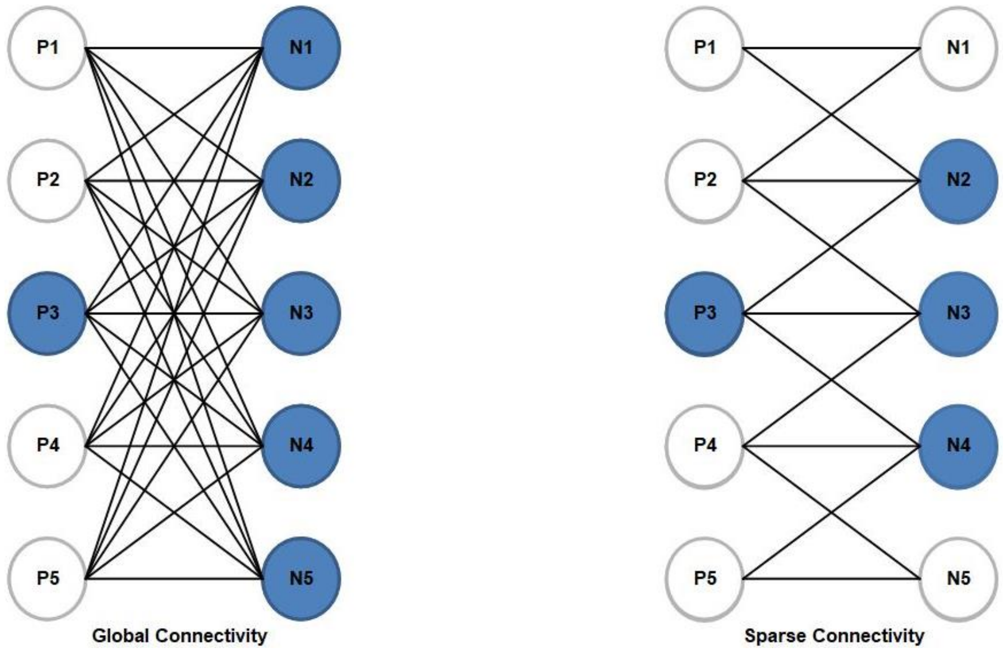


Figure 9 - *Global connectivity (MLP) and Sparse connectivity (CNN).*

The second characteristic is the local receptive fields: they are modeled by limiting the input connections in each neuron only to elements within a given neighborhood established on the signs of previous layers, as shown in Figure 10.

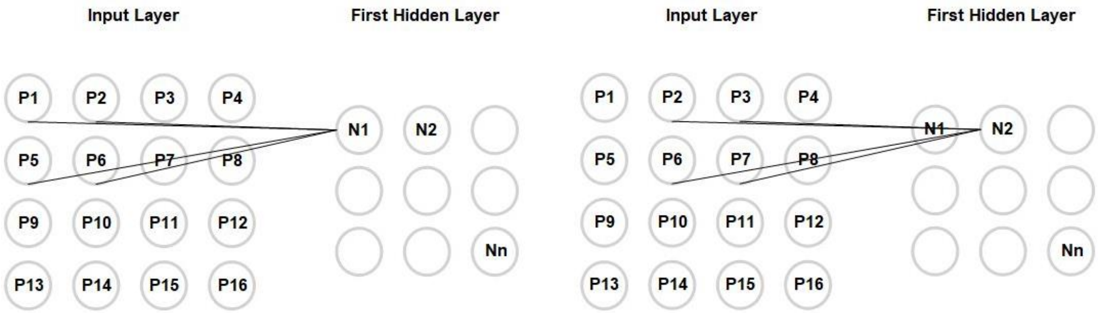


Figure 10 - *CNN receptive local field example.*

When analyzing the Figure 10, it is noted that only the connected region in the figure affects the output neuron N1. This highlighted region is the name of the local receptive field of the N1 output neuron. The same goes for the N2 output neuron.

Finally, we have parameter sharing, which occurs because the same filters are applied at different locations in the input array, causing patterns that occur frequently in the input array and that are located anywhere in the input can be learned. As we can see in Figure 11.

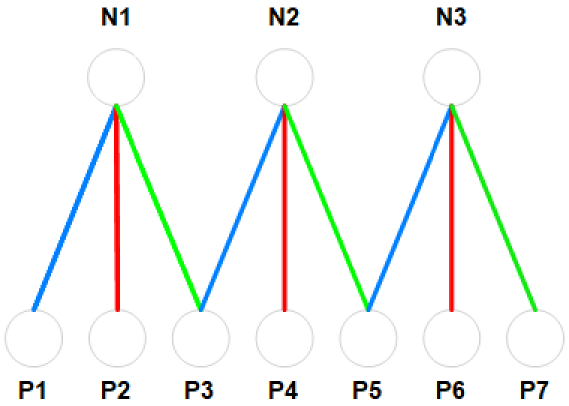


Figure 11 - *Parameter connections. Connections in the same color represent weights or kernel elements being shared.*

For the authors in [44], the use of the characteristics mentioned above allow CNNs to store fewer parameters, reducing memory requirements of the model and improving its statistical efficiency. In addition, the generation of the results obtained requires fewer operations, which allows a great improvement in the efficiency of these network models.

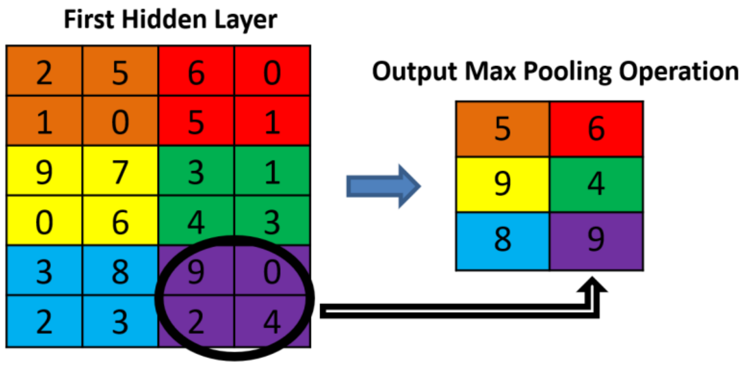
Pooling layers

Pooling layers are typically used immediately after convolutional layers, and are responsible for trying to find the most important and meaningful information [44]. Also, according to the authors, the pooling layers replace the output of the convolutional layers in each region with a summarized statistic of the nearest outputs, which generates a reduction in the number of neurons in the previous layer.

The function that reports this value from a given region is called the pooling function. In addition to this, a lot of functions are used such as: the mean, the L2 standard or the weighted average of the local region analyzed [54].

An important property of pooling is that it provides a fixed-size output array c. For example, an application that makes use of 100 filters in the convolution layers, in pooling each filter will result in a 100-dimensional output vector, indifferent to the filter size, or even the size of the input. Therefore, in applications involving classification of texts, where input vectors must have the same size, sentences and filters of different sizes will always result in an output vector of the same dimension, thus enabling the use of the classifier [54].

When applied to image recognition, the grouping provides a value for the analyzed region. The max pooling operator will always choose the maximum value [59]. Figure 12 shows the maximum operator for a 4x6 matrix with a 2x2 size filter.



2	5	6	0
1	0	5	1
9	7	3	1
0	6	4	3
3	8	9	0
2	3	2	4

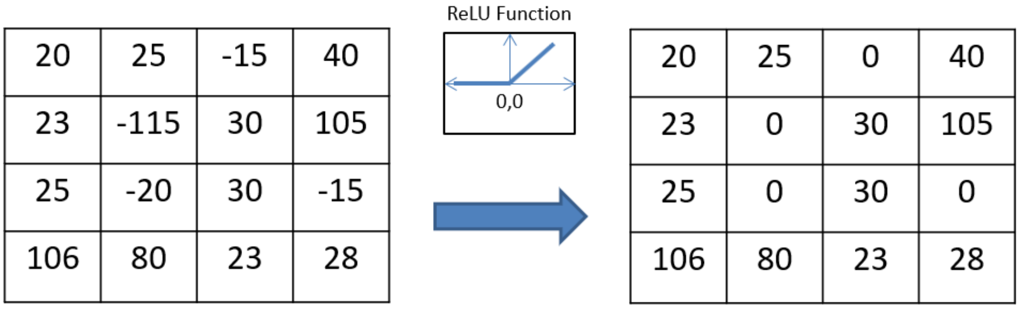
5	6
9	4
8	9

Figure 12 - *Pooling using Max operator, with 2x2 filter and stride equal to 2.*

Activation Function

On CNN, activation functions are intended to determine the activation or not of a neuron. An activated neuron implies propagation of its signal to the next layer of the network. If the neuron is an output neuron, then its activation function determines the output of the network [59]. Mathematically, this occurs through the computation of a neuron given by the sum of the product of the input values and their corresponding weights, in this result an activation function is applied that determines the output value of the neuron [54].

With the appropriate activation function, a neural network model can solve nonlinear problems such as those involving image manipulation, video, audio, and speech recognition. The convolution of a CNN is a linear system, and thus multiple convolutions, cascading, would also form a linear system. Therefore, in so that a CNN can solve nonlinearly separable problems, it is necessary to use nonlinear activation functions between each convolution layer [58]. As shown in the Figure 13.



20	25	-15	40
23	-115	30	105
25	-20	30	-15
106	80	23	28

20	25	0	40
23	0	30	105
25	0	30	0
106	80	23	28

Figure 13 - *ReLu function applied in Feature Map example.*

It is also important to consider the choice of activation function. This characteristic allows the use of the backpropagation training method, which allows the propagation of the error and, consequently, the adjustment of weights. The backpropagation algorithm was proposed by [53] in the 1980s, and to this day it is relevant when it comes to neural network training. In this context, the functions of logistic activation (sigmoid) and hyperbolic tangent (tanh) have become extremely popular because they are smoother approximations of the step function and allowed the application backpropagation. The sigmoid function is an asymmetric function that produces only non-negative real values in the range between 0 and 1, so it is recommended to use it in binary classification problem [54]. However, some for some applications, the sigmoid function can be a problem, precisely because it produces only non-negative actual values, in the interval between zero and one, thus the hyperbolic tangent function becomes useful, because it is only a staggered version of the sigmoid function [69]. Because the hyperbolic tangent function is symmetric relative to the origin, it produces real output values in the range between minus one and one.

There are also activation functions considered modern as rectified linear (ReLU) and maxout, which are linear by parts, computationally inexpensive and work well in practice. In [70], the most commonly used activation function in convolutional neural networks is the Rectified Linear Unit (ReLU) in which negative values are replaced by zero and positive values are maintained.

For several authors, ReLU computationally has some advantages, taking into account the other activation functions mentioned, such as: do not saturate, do not have explosion problems or gradient disappearance, simplify gradient propagation and retropropagation steps to decrease the complexity of the calculations involved, efficiency and sparse representation of the model [69].

It is noteworthy that, the last one allows CNNs not to need pre-training, making the process more efficient. These specialties show the relevance of the ReLU function and its variations in the context of CNNs. In addition to computational efficiency, the use of ReLU functions is justified, since their composition allows the approximation of more complex functions [71]. Furthermore, it is because of these reasons that the use of ReLU accelerates the learning of deep CNNs and has allowed the training of increasingly deep topologies [71].

Fully Connected Layer

CNNs networks have one or more fully connected layers. The purpose of this layer is to use the highest-level feature maps generated in the previous layers - convolutional and pooling - to perform classification of input data into multiple classes based on a training set [59]. Unlike the previous layers, where weights are connected to only a certain region, fully connected layers are formed by a neural network fully connected with all neurons of the previous layer (usually pooling), and it is necessary to convert the map of characteristics into only one vector [39].

It is also composed of a last layer, called output layer, where each neuron represents a certain class (target) of the model, so the number of neurons in the output layer corresponds to the number of classes present in the model. It is noteworthy that if only the fully connected layer is analyzed, it is possible to identify it as a traditional MLP neural network [57]. This is responsible for classifying the received characteristics, generally using a SoftMax function in the output units, which obtains the probability of an input image belonging to a particular class [17].

2.3.4 CNN Models

In the following, the main CNN models found in the literature and that were chosen to perform this work are discussed.

CNN Architectures

Much of CNN's current popularity comes from the results obtained from the 2012 Edition of ILSVRC, reported in [72]. The authors used a technique proposed by Yan Lecun [65] in the 1990s.

As mentioned earlier, another important factor for studies on CNN is the increase in computational power that has been going on since the beginning of the century.

There are several CNN architectures with a huge range of applications. Next, we will discuss architectures selected for their historical importance or relevance in the face of the challenges of classification of images on large bases, specifically those winners of the 2012 to 2015 editions of the ILSVRC classification challenge [55].

AlexNet developed by the authors of [72], is a CNN architecture with more than 60 million parameters winning ILSVRC large-scale visual recognition challenge in 2012. It surpassed the performance of state-of-the-art methods of that time in the competition, it scored 16% error compared to second place with 26% error in the Top-5 ranking, drawing attention to the potential of deep CNNs. Basically, this network consists of eight layers, the first five consisting of convolutional layers (some of them followed by *max pooling layers)* and the last three formed by fully connected layers.

According to [73], AlexNet uses the rectified linear unit (ReLU) as an activation function. In addition, it employs the *use of the dropout technique* to avoid over-fitting the network, in which it consists of zeroing the output values of a neuron from a layer with a 50% chance. In this way, the network learns more robust descriptors, since it eliminates the contribution of some neurons and takes away the assurance that one neuron can depend on another.

Compared to LeNet, AlexNet is larger, deeper, works with color images at higher resolution and has a greater number of feature maps stacked per layer

ZefNet [74], ILSVRC 2013's winning architecture, has made important improvements over the hyperparameters that define AlexNet, expanding the size of intermediate convolution layers and reducing the size of the filters of the first convolution layer. This work also proposes network reversal layers in the structure called deconvnet, allowing the visualization of the activations produced by the characteristic maps along the CNN layer hierarchy.

The GoogLeNet [75], which was developed by Google researchers, is a 22-layer architecture and the winner of the 2014 edition of ILSVRC. It brings a module called Inception that considerably reduces the number of network parameters compared to AlexNet. The Inception module combines filters of different sizes creating mixed and wider layers. Added to this, it eliminates the fully connected layers typically used on top of CNNs, and puts in place an average pooling layer, thereby eliminating more training parameters. Improvements to the original model were proposed by [76], [77].

A VGGNet [78] was the winning architecture of the localization challenge and second place in the ILSVRC 2014 edition ranking challenge. It showed that the depth of the network is a crucial component for performance improvement. VGGNet is an extremely homogeneous architecture in two models, with 16 or 19 convolution layers alternating with grouping, followed by 3 layers fully connected at the end of the network. It uses 3 x 3 filters across the network with clusters in 2 x 2 windows. Compared to GoogLeNet, it has higher demand for memory and parameters.

A ResNet developed by [79] was the winning architecture of the ILSVRC 2015 edition and enable the development of neural networks deeper than traditional CNNs. Its main contributions are the inclusion of special connections called *Skip connections*, in addition to the use of batch normalization. Such innovations have allowed the training of considerably deeper networks reaching 1000 layers, but with less complexity than the shallower networks proposed so far.

Its architecture consists of multiple convolution layers combined with *Skip connections*. As shown in [80], the increase of layers can generate a degradation in the accuracy of the training. In their experiments, the authors of [61] demonstrate that the use of deep residual neural networks converge faster than flat (conventional) neural networks and exhibit lower errors during training and validation.

3. Proposal of Image Processing Architecture

A surveillance structure is usually composed of cameras with good resolution, connected to processing and storage equipment. The proposal intended here is to use this structure to provide an automatic analysis of the images coming from the camera with the application of detection techniques and FR in real-time. In addition, the proposed system includes a comparative analysis of processing time in some operations. Therefore, the real-time FR system works in Raspberry Pi 3 Model B hardware and the operational system of a Raspbian. Basically, surveillance camera images are sent to the Raspberry via the standard 802.11n wireless. During transmission, these images are received and processed, creating a record with information about the detection and identification of all individuals who were within the range of the camera.

Due to the ease of connection to the hardware, many works in the literature use the standard Raspberry camera. In this document, a standard surveillance camera model is used, which guarantees the prototype shown in Figure 14, greater fidelity in relation to real surveillance systems. Moreover, the use of this wireless standard for data exchange, allows the hardware and the camera to be installed at different locations. Furthermore, it is worth noting that the designed system has two relevant characteristics: the first is the operation in real-time and the second is the use of a model that has high precision of FR. The recognition technique used in this work, has an accuracy of 99.38%, shown by the authors in [81], using the Labeled Faces in the Wild (LFW), one of the most widely used databases for reference in FR systems [82]. In the performance evaluation section of this work, this result is proven.

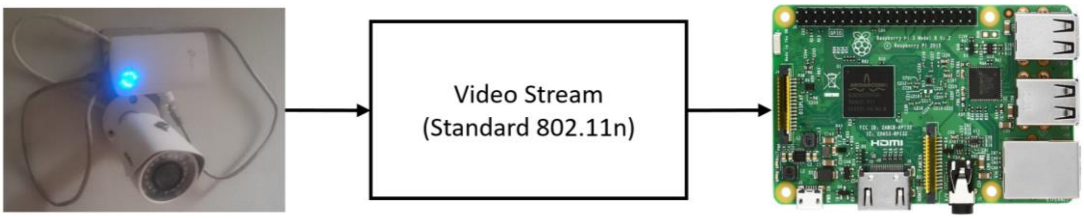


Figure 14 - *Standart surveillance camera model and propotype.*

3.1 Detection

Detection systems are widely used as a solution to various computer vision problems. In facial recognition systems, one of the first steps is the detection of the faces [83]. Detection makes up part of a preprocessing, required to find one or more faces in an image. Facial localization consists of the use of one or more techniques capable of estimating the position of faces within an image, or in a video frame.

3.1.1 Detection Techniques

Among the various techniques present in the literature, the best-known detector is the Viola-Jones, which was widely used due to its efficiency in computing time. Even today, it is the main search algorithm in many jobs, despite presenting itself as a good detector only for faces in front position, and a certain compromise of results when the faces are in a different direction [84]. Another very popular method is the HOG, presented in [85], which aims to extract specific characteristics in the images. Its choice as a detection technique in this architecture is due to its good results, combined with its processing speed, which will be presented in later experiments in this dissertation. The algorithm is based on the idea that the contours of an object can be identified many times by the intensity of the gradients of the edges, without a prior knowledge of the position of such edges.

HOG extracts the characteristics based on the image gradients, by manipulating the frames to generate gradient vectors of intensity, with directions that depend on the variation between pixels. It does not examine the characteristics of the full image, but rather of subdivisions of the image, called cells [86]. It then calculates the gradient of each cell and the extracted gradient vectors providing a way of visualizing the faces within an image, similar to what is shown in the Figure 15.

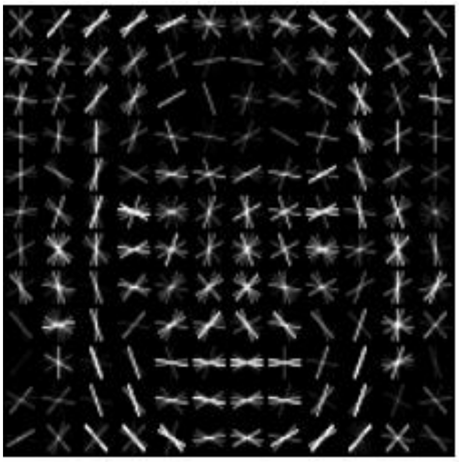


Figure 15 - *Exemple of HOG filter learned via MMOD.*

The HOG has extraction capabilities based on geometric characteristics, so it is possible to maintain a good invariance to the optical changes of a local image, which is that increases the robustness of the technique against lighting problems.

The Support Vector Machines (SVM) algorithms establish a learning technique used in the task of detecting and classifying training data. The SVM algorithm works with the idea of finding a hyperplane in an N-dimensional space, which classifies the input data distributed over points in that space.

The objective of the technique is to find a hyperplane that separates the data and, at the same time, stay as far away from both sides ensuring a high generalization. In this context, the hyperplane that maximizes the distance from both sides and simultaneously manages to divide the two sets, becomes a result of an ideal training. Through the example presented in the Figure 16, we can see that A and B do not correspond to ideal values of separation.

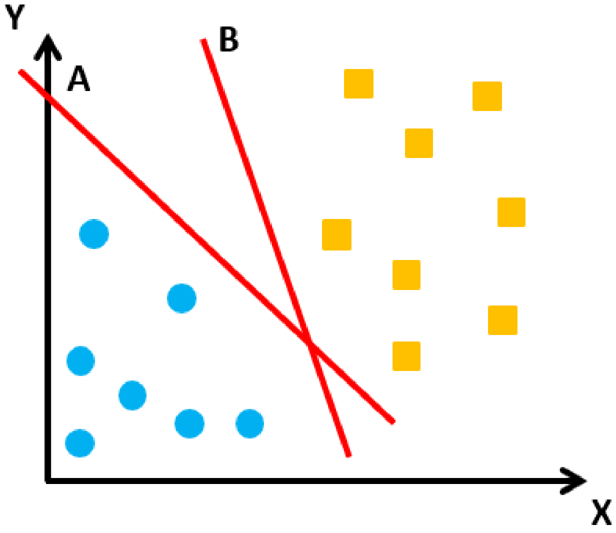


Figure 16 - *Example of class separation lines.*

As demonstrated in Figure 17, there are many hyperplanes that can be traced to separate the data points. In this case, the goal of the SVM algorithm will be to find a plane that has the maximum distance between data points of different classes. Maximizing margin distance enables future data to be classified with greater confidence. Thus, the points used to determine the maximum distance between data points of distinct classes are referred to as support vectors (see Figure 17). Therefore, the essence of the SVM algorithm lies in finding the support vectors that determine the maximum distance between data points of distinct classes.

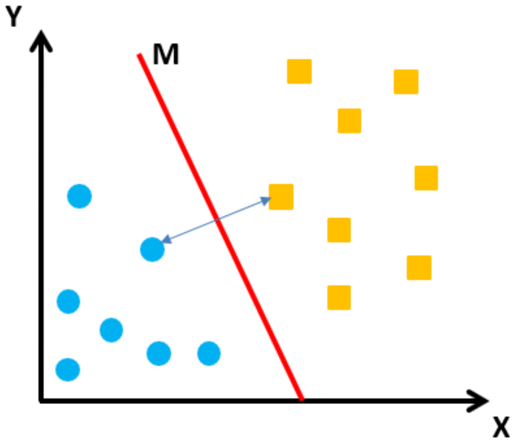


Figure 17 - *Exemple of an support vector machine (SVM)***.**

Separation hyperplanes depend solely on support vectors, which are found during training. However, because most real problems are not linearly separable and SVMs are binary classifiers, it is necessary to use multiple binary SVMs to build a multiclass classifier, that is, the larger classification problem is subdivided into several binary problems. Furthermore, increasing dimensions can also cause an optimization problem to fail for a Linear SVM.

The HOG used in this structure was trained by a Max-Margin Object Detection (MMOD) method, which plays a similar role to SVM - maximizing margins and sections. MMOD training makes system performance improved by attenuating missed detections and false alarms. It can be used to improve any method of detecting objects, whose scoring functions are linear in the learned parameters [87], allowing the identification of faces within the frames in which the technique is applied, and the transmission to the recognition step.

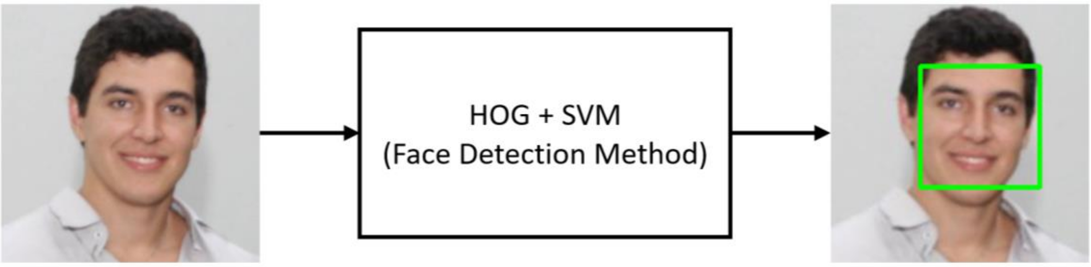


Figure 18 - *Face Dectection Method HOG+SVM*

$$\min_{w,\xi} \frac{1}{2}\left|\left|w\right|\right|^2 + \frac{C}{n}\sum_{i=1}^{n} \xi i \quad (1)$$

MMOD optimization is used to improve the classification of the algorithm, its objective function is presented in the equation (1), which is intended to minimize the detection error. In this equation, C is the variable that allows defining the relationship of commitment between minimizing errors in the training set in relation to the complexity of the system. This variable is usually configured in the application of the SVM technique, since MMOD and SVM operate in analogous way, addressing the concept of maximum margin and optimization by cutting plane. *w* is defined as the parameter of a vector that leads to fewer detection errors and ξ is an upper limit of the loss incurred by the training example.

The facial model used was defined with parameters C = 50 and ξ = 0.01C, according to [87]. In practical terms, this step defines the set of coordinates where the face clipping will be. In this way, the clipped face is sent to the alignment step, reducing the complexity of the problem. All frames go through clippings until all present faces are found, if no images are found within a given frame, the process terminates, and a new frame is read.

Lin *et al.* [88] suggest the use of more advanced features, such as the use of CNN for facial detection. Neural network-based detection methods have been notorious since the last century. They started with face detection only for the front position and then started detecting rotated faces [88]. CNN has mastered many computer vision tasks, which justifies the growth of this type of technique in the detection stage.

The advantage of applying this type of net is the ease of adaptation when the face is in position or angle variations. However, some authors claim that CNN has a high computational cost and presents better performance when running on Graphics Processing Units (GPU's) [88].

3.2 Align and Normalization

Facial alignment also attracts great research interest, as it is a challenging factor when the face is under partial occlusion, low illumination or in a very different angulation from the frontal position [89]. A face aligner is usually constructed offline through manual sample mining operations.

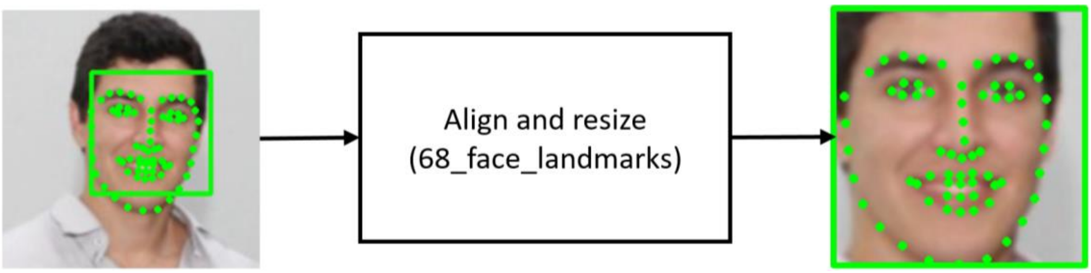


Figure 19 - *Align and resize of the main landmarks on the face*.

To perform the alignment, a 68 point landmarks pose predictor is used, based on regression trees [90]. It is used in the image to estimate the positions of the main landmarks on the face, such as eyes, nose, mouth, and face contour, as shown in Figure 19. These geometric representations provide metrics and positions that can be used to aid in detection techniques and FR [91]. After identifying where the face is and what its position, the face is normalized and aligned using a simple 2D transformation, making the images sent to the network have eyes and nose in similar places in all the photos. Projects like OpenFace [92], also work with this type of normalization. Herewith, the images received by the Raspberry go through the HOG application, and when the gradient vectors, in part of the image, correspond the facial model, a new face is detected and sent to the Residual Network (ResNet).

3.3 Recognition

FR can be used in two ways, the first is for verification, comparing the current image with a specific face previously requested. The second way is for identification, in which the current image is compared with a database, to determine the similarity between the current image and any of the images that are registered in the database. The application developed in this work uses the second case, in which the image is tested alongside all the users registered in the system. The embedded code was written in python and uses some specific libraries for image treatment, such as OpenCV [50].

3.3.1 Residual Network

The structure used in recognition is a CNN, more specifically, a ResNet. Residual Networks make use of previous layer information more efficiently compared to a flat CNN structure [93]. This makes convolutional filters carry more information to detect patterns. This feature allowed the creation of a 152-layer ResNet, that is, 8 times deeper than one of its main predecessors: the VGG [94].

Deep networks naturally integrate more levels of resources due to the number of stacked layers, which can greatly increase their accuracy. The problem is that deeper networks are usually harder to train. The use of a residual learning structure facilitates network training since the layers are reformulated and learn from residual functions. In this case, the lower layers reference the residual block input data, rather than learning only from unreferenced functions, optimizing the process.

The combination of depth and the use of residual blocks has made the ResNet-152 [93] achieve greater accuracy due to its depth gain, and reach a lower complexity, even with the increase in the number of layers.

The degradation problem suggests that solvers may have difficulty approaching identity mappings by multiple nonlinear layers. With this, even after increasing the depth of the network, the accuracy can remain constant or even decrease by the effect of saturation. The central idea of ResNet to reduce this problem, is to introduce the "identity shortcut connection". Identity shortcuts are particularly important not to increase the complexity of architectures. Despite this, the entire network can still be trained end to end by SGD [95] with backpropagation [96].

In the comparison between simple and residual networks, we have the addition of an identity shortcut. In this case, shortcut connections simply perform identity mapping, and their outputs are added to the outputs of the stacked layers. As we can see in the Figure 20, the flat architecture is presented first. Soon after, the equivalent residual network is presented, with the shortcut connections, both with the same number of parameters, depth, width and computational cost.

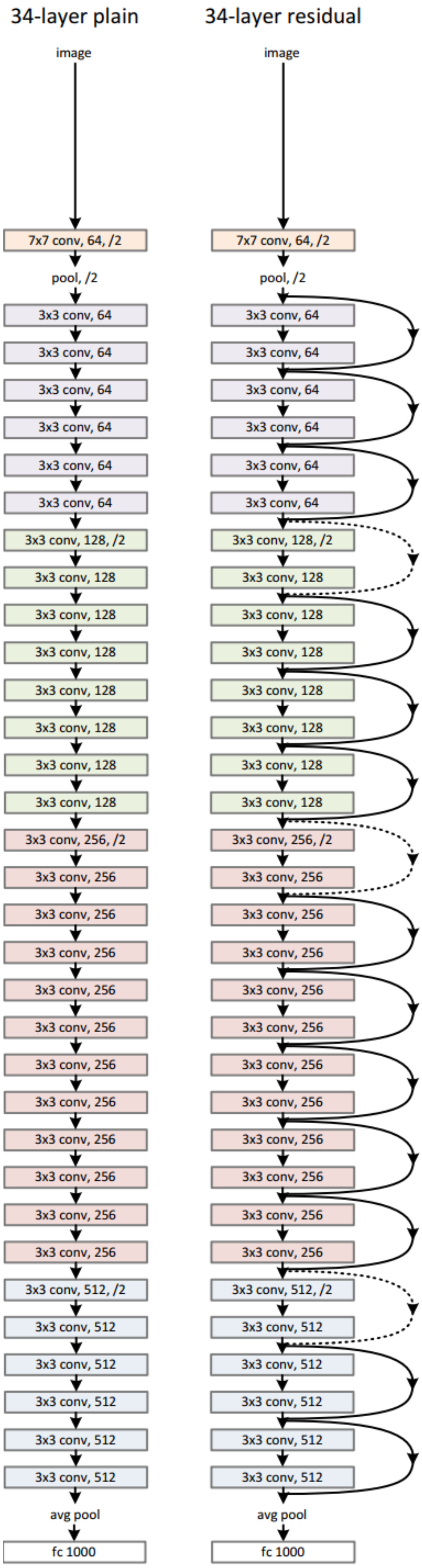


Figure 20 - *Example of a CNN 34-layers in a plain and residual variants. From:* [93].

The process of building this network is like that of a conventional CNN, but as stated earlier, incorporates residual layers along with convolutions, which act as nonlinear functions of convolutive layers. The intention of adding this new layer is to prevent accuracy from derailing in cases where deeper networks are being used.

A residual block from ResNet has two convolutive layers in its infrastructure and two Rectified Linear Unit (ReLU) activation functions, as shown in the Figure 21. The residual layer differs from the other convolutive layers, in the sense that its output is the sum of the output of the second convolutive layer with the input of the residual layer. This structure allows deeper layers to directly receive data from the most superficial layers.

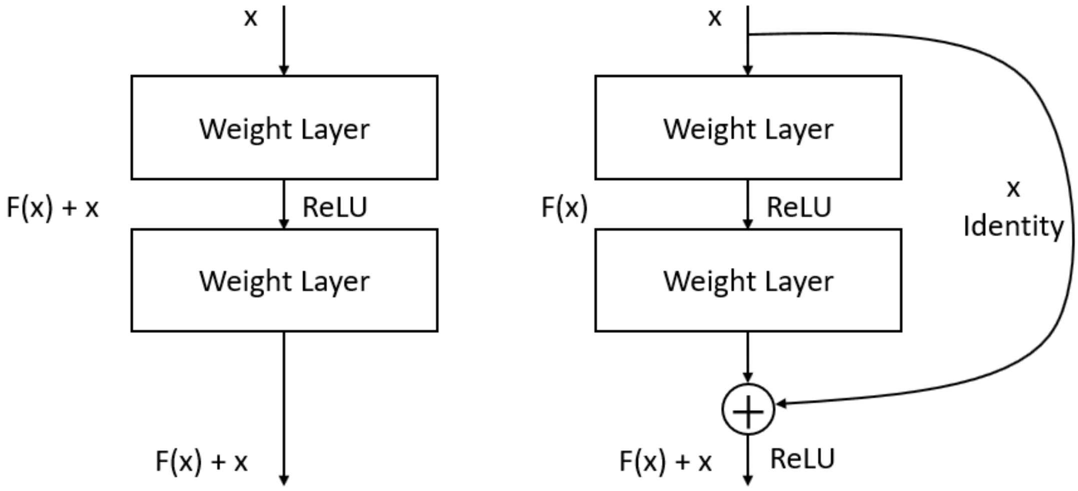


Figure 21 - *Conventional learning and Residual learning.*

In a non-residual CNN, the network is trained to adapt its parameters to all content F(x) + x. In the residual architecture the value of x is added directly to the output through identity operations, so the network must adjust only to the content F(x). As a result, the residual network becomes simpler to optimize and achieves high accuracy.

The advantage of the ReLU activation function is to avoid vanishing or exploding gradient problems in positive values, while reducing negative values to zero [71]. This activation function can be used to improve the learning speed of deep neural networks, and for this reason has become one of the most widely used in deep learning problems.

ResNet was evaluated in the ImageNet classification dataset [56], more specifically in the ImageNet Large Scale Visual Recognition Challenge, which consists of 1000 classes. The results were considerably low training and classification errors. In addition to presenting data generalization. This indicates that the degradation problem is well solved and it was possible to obtain precision gains with increasing depth.

3.3.2 Network Training Stage

As said before, the proposed FR works with deep metric learning and has been trained using the machine learning toolkit, Dlib [97]. This network has the capacity to perform FR through deep metric learning, and this architecture is based on ResNet-34 of [93], but with a reduced number of filters and presents 29 layers, as show in Figure 22.

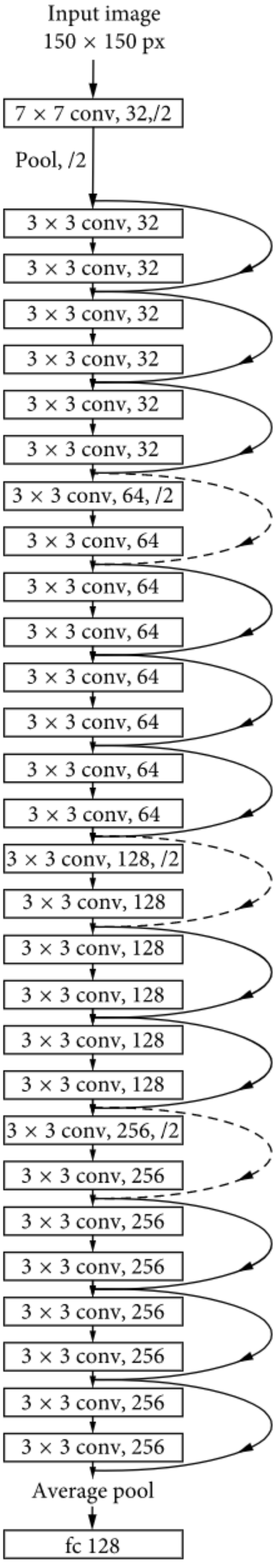


Figure 22 - *Network architeture of the ResNet 29.*

The ResNet used was trained with an input set of about 3 million faces, derived primarily from two large data sets, the face scrub dataset [98] and VGG dataset [99]. The collected images presented different resolutions, so a pre-processing was done to leave all faces aligned, normalized and with a fixed size of 150x150 pixels before it could be used for network training.

The training process works by analyzing three face images at a time, two from the same person (Face 1-A and 1-B) and one from a different person (Face 2). Then, the algorithm analyzes the measurements it is currently generating for each of these three images. Weights are adjusted so that the neural network always generates vectors of 128 similar dimensions for images of the same person and distant values when working with the image of a different person. This process is shown in Figure 23.

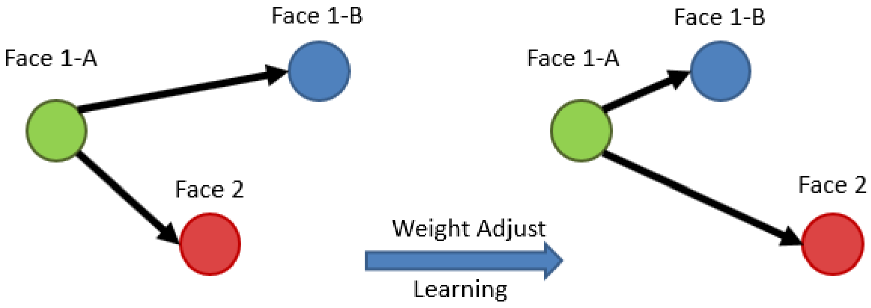


Figure 23 - *Learning triplet process.*

After repeating this step millions of times, for millions of images of many different individuals, the neural network learns to generate 128 measurements reliably for each person. Being that photos of the same person, they should always provide approximately the same measurements. The advantage is that the training is done prior to its application in a real system. Therefore, even if the training takes a long time, the network can still deliver quick results when it operates in the stage of executing the recognition algorithm [54].

The images collected for construction of the bank were under different levels of brightness, color, intensity, and angle when they were used for training purposes. The use of images on different variations, increases the robustness of the system and the chances that new images in different conditions may have better results when the recognition system is applied.

3.3.3 Database Registration Stage

In this step, a folder containing photos of people with their names is indicated. With that, the algorithm goes through the folder and analyzes the faces present inside each one. Figure 24 shows the complete flow, which is repeated until all users are registered. This analysis is done by converting an image into a matrix and applying a classifier to find the face within the image. The classifier used was a combination of HOG and SVM, due to the good results obtained in previous works already mentioned. The result of the detection performed by this technique can be seen in Figure 24.

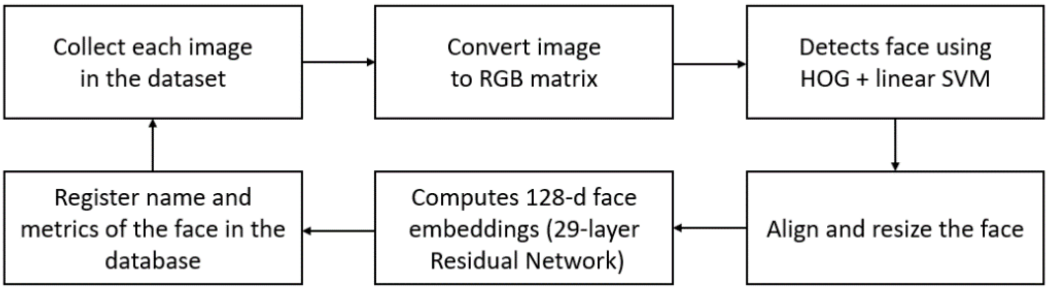

Collect each image in the dataset
Convert image to RGB matrix
Detects face using HOG + linear SVM
Align and resize the face
Computes 128-d face embeddings (29-layer Residual Network)
Register name and metrics of the face in the database

Figure 24 - *New images registration sequence*.

When the facial images are sent to the metric network, it generates a metric vector of 128 dimensions of the faces of each person. The network has 29 layers of convolution which is a simplified version of ResNet-34 with some layers removed and a smaller number of filters per layer.

3.3.4 Recognition Flow

The facial recognition process is done constantly, through a processing flow of facial images cut in size 150x150 during the detection process. The predictor is used in the image to estimate the positions of the main landmarks on the face. After identifying where the face is and what is its position, the face is normalized and aligned. As a result, the images sent to the network have eyes and nose in similar places in all the photos.

Next step is to compare the faces found with the list of people registered in the system. This search is made through correspondences between the vectors of 128 dimensions that have the shortest Euclidean distance, that is, the person in the database who presents the measurements closest to each of found faces. If no image in the database has measurements close, the result returned will be “unknown”. A K-Nearest Neighbors (KNN) algorithm was used, which implements the Euclidean distance for decision. The expression that describes the implemented rule is given by equation (2), where the closest vectors are recognized via distance metric, in which p and q represents the generic points of two different vectors.

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \cdots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \quad (2)$$

The complete and detailed flow of the entire system proposed during the operation stage is shown in Figure 25. This flow covers from the moment of capturing the images and applying the sequence of techniques, until the display of the system's output results.

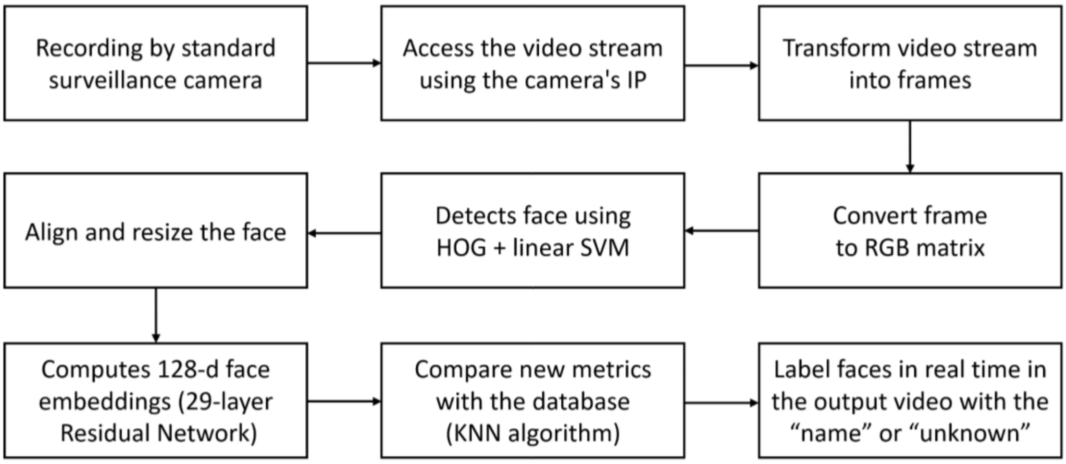

Recording by standard surveillance camera
Access the video stream using the camera's IP
Transform video stream into frames
Convert frame to RGB matrix
Detects face using HOG + linear SVM
Align and resize the face
Computes 128-d face embeddings (29-layer Residual Network)
Compare new metrics with the database (KNN algorithm)
Label faces in real time in the output video with the “name” or “unknown”

Figure 25 - *Full flow of system operating mode.*

When you go through the recognition step, all detected faces are rated. This process returns the username of the database or the "unknown" code. The Figure 26 shows two of these cases. The first, in which the detected face is registered in the system and the second, in which the user is not yet registered. If a user's face is detected as unknown in the video stream, there is a command capable of adding that face to the existing set. This allows the registration of any user, even with the system running.

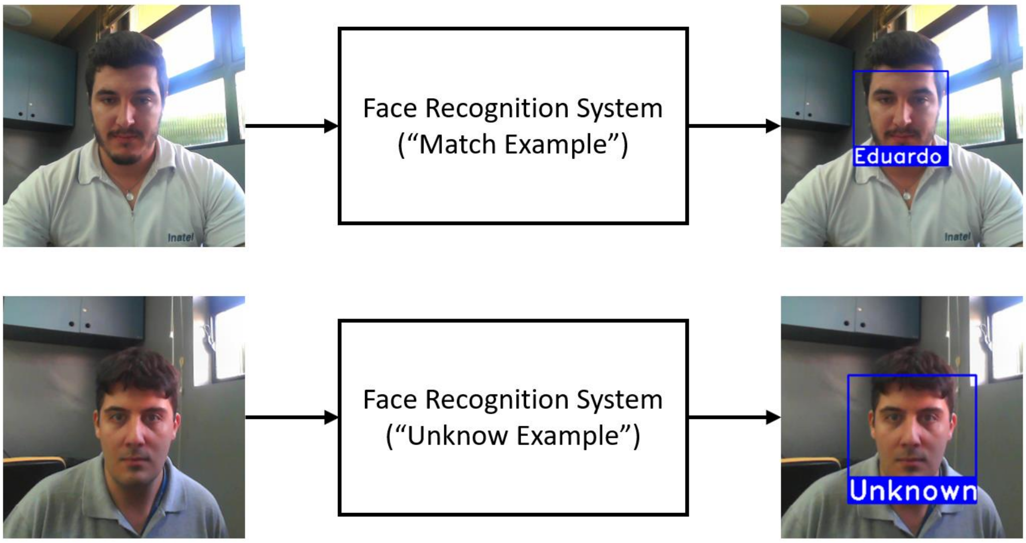


Figure 26 - *System output examples.*

4. Support Structure for Internet of Things

The Internet was initially designed for connections between computers, but with the miniaturization of electronic components and technologies, smaller and smaller devices could communicate using this data network.

The increasingly intense development of objects enabled to connect to the Internet brings a new concept of network use: the so-called Internet of Things (IoT). The first applications of IoT consisted of controlling remote objects and collecting data from them through a mobile app. This perspective has been evolving and today changes even the way data is generated.

The term IoT refers to a system of smart devices or that has the ability to connect and exchange data, promoting increasingly intelligent and connected scenarios. In other words, it is a network of sensory devices with IP addresses capable of generating, transmitting, storing, and receiving millions of data daily without any human assistance.

IoT predicts a world in which physical, digital, and virtual objects are interconnected on a network that supports higher order applications and comes from automated data processing from an existing state or the environmental state in which it is immersed. This data is then passed to a processing node where it is parsed, and then is passed back to the smart object.

Cisco estimates that IoT emerged between 2008 and 2009, when the number of internet-connected devices exceeded the number of people on earth [100]. Currently, IoT has gained strength with the growing number of interconnected physical objects providing interactions and is considered the most promising technology today.

IoT can include many applications designed to help many sectors such as industry, transportation, markets, education, agriculture, health, environment and smart cities [101], and it is estimated that by the end of 2020, there will be about 30 billion IoT devices or devices worldwide [2].

This is due to the increased exchange of data on increasingly simple devices, so the trend is that in some years, billions of objects are connected and communicating (Evolution of the Normal Internet to IoT). Something that is inserted in the IoT paradigm, which has as main role to be a fundamental facilitator of the integration of various application solutions and communication technologies [101].

As the increase in the number of connected devices grows exponentially, the number of data generated, therefore, in addition to the intelligence to enable large-scale communication, the storage of this data also becomes complex requiring an adequate tool, since in IoT scenarios the data generated are responsible for making system decision-making.

The growth of multimedia traffic in IoT has led to discoveries of new techniques, such as IoMT devices. Although they need greater demands for their operation, IoMT is very versatile, and it has been used in several areas such as emergency response systems, traffic monitoring, criminal inspection, smart cities, smart homes, smart hospitals, smart agriculture, surveillance systems and Industrial IoT (IIoT) [14]. Nevertheless, IoMT, still according to [14], need to overcome some challenges for multimedia communication such as, heterogeneous devices and data, strict QoS, dynamic networks, rand delay sensitivity and reliability requirements.

An example of IoMT use is the very way surveillance systems have been evolving, as shown in Figure 27. In classic surveillance systems the camera is able to do the recording and storage on disk, already if we consider more current systems, in addition to storage, the technology present in the system is able to process the images of the camera and use them as a means to generate data and take actions, such as recognizing a person in an image and making records related to this information, either on the disk itself or in the cloud.

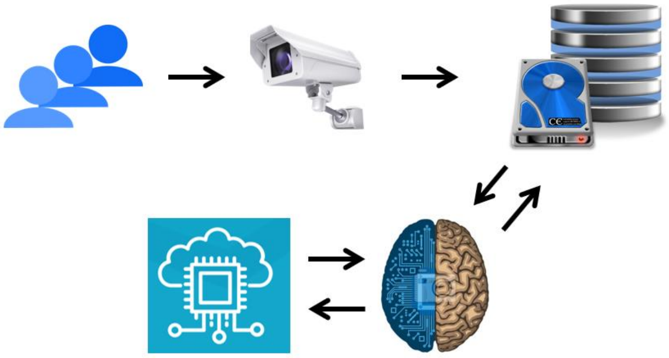


Figure 27 - *Representation of a Smart Surveilance System.*

In both cases, different applications can take advantage of the collected data and make the system even smarter, giving more information to the user or increasing security in monitoring environments. Some interesting tools are known to enable this type of application: an information concentrator called Middleware can act as a gateway to these devices, also acting as a form of intelligence and connection, since these scenarios can present heterogeneous and complex environments, such as IoT networks in smart cities. In this way, Middleware and Databases are tools that are complementary to provide a quality infrastructure to IoT scenarios.

In next section we will explore how these tools can be used in cooperation with FR systems to contribute to system intelligence, search automation, storage, and connection ability with other applications.

4.1 Middleware for Internet of Things

Middleware is a software capable of connecting basic systems to each other or to third-party applications. IoT devices are a quite common example of the application of this technology. It plays a crucial role as it allows all data received from the devices to be stored and made available to the user quickly and consistently.

Such software operates as a translation layer that enables communication and data management for applications [102]. There are many IoT middleware solutions available in the literature, as well as on the market. Some of these solutions are open source and free to download and test. As most are open source, the code can be changed according to the needs of the application.

Other solutions are closed source which are available in the cloud in the form of PaaS (Platform as a Service). The advantage of PaaS solutions is that they are located in the cloud and authenticated users can access data located on the server from anywhere in the world without having to worry about deploying or managing the infrastructure [53].

Camera surveillance systems have been around since the last millennium, but analog technology cameras today give way to IP-based cameras, because the use of this protocol is one of the main concepts behind the implementation of IoT environments. In addition, the advent of smart cities has brought with it not only smarter surveillance systems, but also a multitude of other sensors and / or actuators which have a strong characteristic of heterogeneity between devices. Much of this justification is due to the diversity of software and hardware used in those environments, in addition to the different standards adopted by manufacturers. In this scenario, Middleware IoT emerges as a possible solution to enable cooperative operation of these different technologies [103]. The role of Middleware in this type of application is fundamental, given the growing demand for connection between devices and applications, as it aims to solve a large part of the heterogeneity problem present in IoT environments.

Middleware allows objects with different technologies to communicate with different cloud services, solving compatibility problems in the IoT scenario. The toolkit found in Middleware can provide interoperability between devices and applications.

The architecture existing in this scenario of objects connected to the internet is able to abstract technology restrictions and protocols which are used in the extremity layers, and uses the IoT platform as a kind of “bridge” between these layers, as shown in Figure 28 The platform is a software package that integrates devices, networks and applications, preventing API developers from needing to understand the complexity of technologies used in physical infrastructure devices.

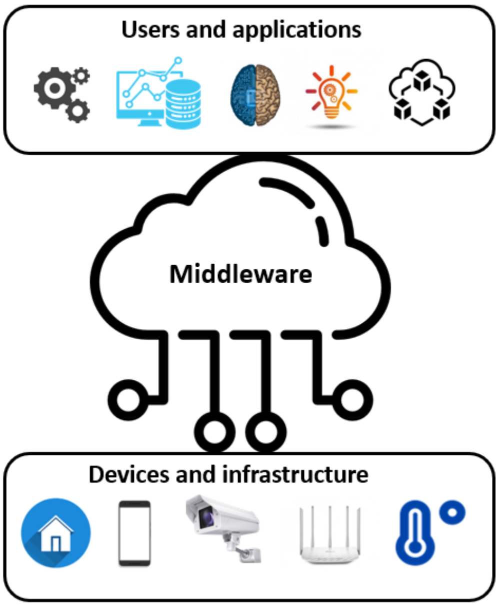


Figure 28 - *Simplified IoT-layered architecture.*

For different applications and devices to be able to interact through Middleware, it is necessary that the system meets some operational and security requirements. The platform chosen for this dissertation work can provide the necessary tools so that the FR system is able to operate correctly and safely. Listed below are some of these characteristics and their purpose for the system in question:

- Data and event management to record all information regarding the readings performed by the recognition system, in addition to allowing the consultation of this data.
- Real-Time Operation: once the system has this as one of its main characteristics.
- Scalability - in case the system needs to be expanded and a multi-camera needs to be implanted.
- Security and privacy tools that protect and guarantee access to data only by applications with the appropriate credentials.
- Easy configuration by users who do not have advanced technical knowledge, allowing the platform to be widely disseminated.
- Ability to adapt and interoperate with different devices, applications, and technologies, allowing the system to easily adapt to new needs.

In this case, as in [101] and [104], the used Middleware is In.IoT, developed during the work [53] and available in [105]. It is a state-of-the-art Middleware that includes the necessary security requirements, in addition to allowing users to track data in real-time. The development interface can be seen in Figure 29.

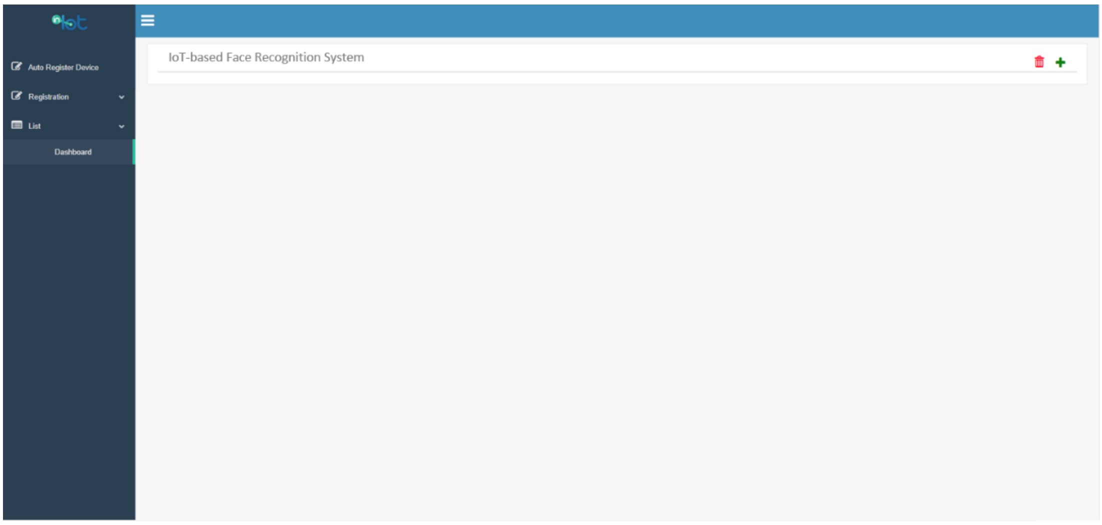
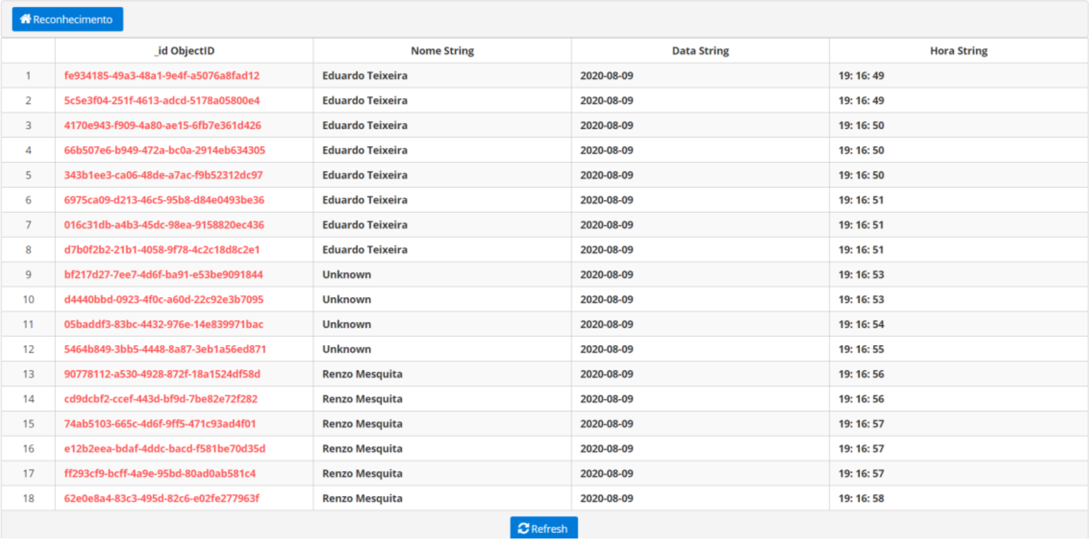


Figure 29 - *Development Interface.*

The use of these tools to manage activities also allows relieving the processing on the final devices which is something crucial, since the computational power of these devices is generally much lower, when compared to the processing power of the platform.

The application shown in Figure 30, was built to view the detection data. It collects data from the platform and represents it visually. The purpose of building the application is that any user may be able to identify the access control data, even if they do not technically know how the solution works.



	_id ObjectID	Nome String	Data String	Hora String
1	fe934185-49a3-48a1-9e4f-a5076a8fad12	Eduardo Teixeira	2020-08-09	19: 16: 49
2	5c5e3f04-251f-4613-adcd-5178a05800e4	Eduardo Teixeira	2020-08-09	19: 16: 49
3	4170e943-f909-4a80-ae15-6fb7e361d426	Eduardo Teixeira	2020-08-09	19: 16: 50
4	66b507e6-b949-472a-bc0a-2914eb634305	Eduardo Teixeira	2020-08-09	19: 16: 50
5	343b1ee3-ca06-48de-a7ac-f9b52312dc97	Eduardo Teixeira	2020-08-09	19: 16: 50
6	6975ca09-d213-46c5-95b8-d84e0493be36	Eduardo Teixeira	2020-08-09	19: 16: 51
7	016c31db-a4b3-45dc-98ea-9158820ec436	Eduardo Teixeira	2020-08-09	19: 16: 51
8	d7b0f2b2-21b1-4058-9f78-4c2c18d8c2e1	Eduardo Teixeira	2020-08-09	19: 16: 51
9	bf217d27-7ee7-4d6f-ba91-e53be9091844	Unknown	2020-08-09	19: 16: 53
10	d4440bbd-0923-4f0c-a60d-22c92e3b7095	Unknown	2020-08-09	19: 16: 53
11	05baddf3-83bc-4432-976e-14e839971bac	Unknown	2020-08-09	19: 16: 54
12	5464b849-3bb5-4448-8a87-3eb1a56ed871	Unknown	2020-08-09	19: 16: 55
13	90778112-a530-4928-872f-18a1524df58d	Renzo Mesquita	2020-08-09	19: 16: 56
14	cd9dcbf2-ccef-443d-bf9d-7be82e72f282	Renzo Mesquita	2020-08-09	19: 16: 56
15	74ab5103-665c-4d6f-9ff5-471c93ad4f01	Renzo Mesquita	2020-08-09	19: 16: 57
16	e12b2eea-bdaf-4ddc-bacd-f581be70d35d	Renzo Mesquita	2020-08-09	19: 16: 57
17	ff293cf9-bcff-4a9e-95bd-80ad0ab581c4	Renzo Mesquita	2020-08-09	19: 16: 57
18	62e0e8a4-83c3-495d-82c6-e02fe277963f	Renzo Mesquita	2020-08-09	19: 16: 58

Figure 30 - *Web Application to data view.*

4.2 Database Management System

The impact of IoT growth presents a new challenge for DBMS. IoT-based devices produce a large amount of data that can be stored and converted into knowledge. For this purpose, databases are extremely relevant, as they provide information that can be submitted to various techniques for knowledge extraction, such as deep learning and machine learning algorithms [67].

For this to be done, the data must be stored continuously, so that they can be accessed later and converted into information through processing techniques [106]. At the time of storage, we have two categorizations for DBMS's, which are divided into SQL banks that are relational and NoSQL, which are non-relational.

Some comparisons are made between the SQL and NoSQL databases for IoT Applications [107], Big Data [108], Web Applications [109], and Cloud Environments [110] scenarios, all of these works cited used PostgreSQL and MySQL for relational database, for the non-relational model MongoDB was selected. These banks are among the first five database management systems, classified according to their popularity as shown in Table 3. The reason the other two banks are not considered during the surveys is that they are not free or open source, which somewhat restrict their use in academic studies.

Table 3 - Data management systems. Adapted from [111].

RANK (2019-2020)	DBMS	Database Model
1°	Oracle	Relational, multi-model
2°	MySQL	Relational, multi-model
3°	Microsoft SQL Server	Relational, multi-model
4°	PostgreSQL	Relational, multi-model
5°	MongoDB	Document, multi-model

The two banks can serve IoT applications, however, during the comparisons, the authors point out several advantages of MongoDB in relation to PostgreSQL and MySQL. The authors point out the higher processing speed for basic operations, such as Insert, Select, Update and Delete, in the processing of large volumes of data, even the possibility of storage without a schema and easy support for unstructured data. cases that can be common in IoT scenarios.

NoSQL databases sacrifice ACID transactional properties in order to achieve greater availability and scalability, being designed to achieve greater performance in data processing [112]. This characteristic is remarkably interesting from the IoT point of view, since the data generation trend is exponential.

MongoDB is a type of NoSQL database, based on documents and open source. Table 3 presents a comparison of terminology used in SQL banks compared to document-oriented NoSQL banks, a sub-class to which MongoDB fits. In document-oriented banks, data communication can be carried out via JSON or XML. This feature facilitates the integration with the middleware, since there is middleware that work with at least one or more languages. However, when comparing these two formats, JSON has a simpler structure, which does not generate unnecessary communication overhead and ends up becoming a factor that influences the choice when we approach the IoT scenario [113].

5. Performance Evaluation of the Proposed IoT Surveillance System

The experimental results were obtained with the intention of evaluating various parameters of the system. Thus, the parameters investigated were processing time, accuracy, and robustness of the system in view of the different types of noise in the images. To perform the experiments, an open-access image database is used, created to assist researchers in their technical analysis and standardize benchmarking. The LFW bank is one of the most used in the evaluation of facial recognition and detection systems. It is a public reference for face checking and has 13,233 images of 5,749 different people, of which 1,680 have more than one image [114].

During the analyses, different experimental scenarios were defined. Each scenario presents the techniques used, the LFW sampling scenario, the processing device, and whether there is any type of modifier in the image. The goal is to demonstrate how the system behaves under each test condition presented.

When evaluating different models, it is essential that the same test conditions are applied. With that, all tests were done with images taken from the LFW database. Except for the ResNet accuracy test, which used the entire database, all other scenarios were designed with LFW sampling. All images in this database are 250 x 250 pixels, and are always cropped to the appropriate size for the architecture.

The other experiments had different images of 20 individuals, in total, 6 images of each one was used, resulting in a subset of 120 images. The subset was divided into two parts, the 20 images that make up Figure 31 – one for each individual – were used as a record in the recognition system, while the remaining 100 images were used for testing in both stages: recognition and detection.

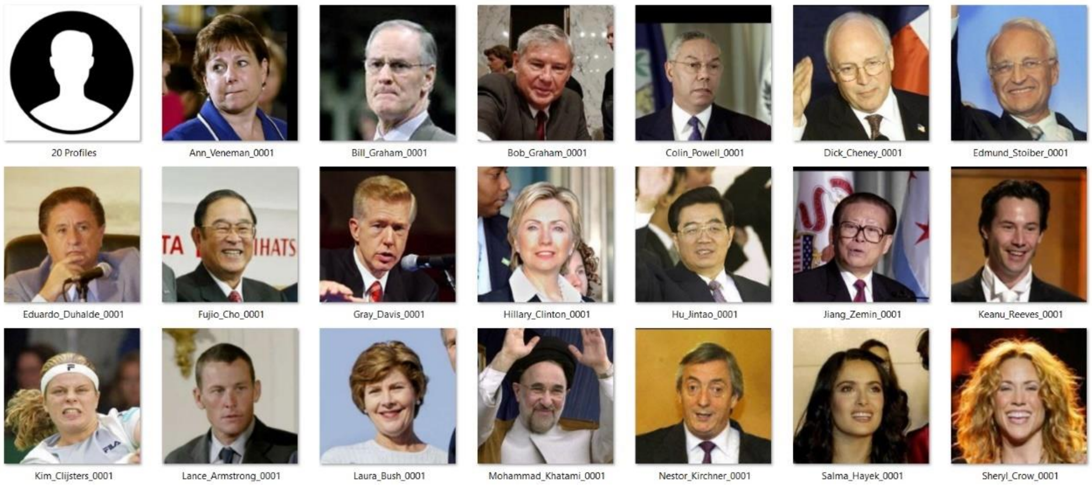


Figure 31 - *Samples of LWF.*

Analyzing the database images, it is possible to notice that most of them are almost in ideal processing conditions, that is, few images have low resolution and low luminosity. However, as the IoT-based recognition system can be used "in the wild", in some experiments will be added noise in the system to verify the robustness of the techniques adopted under various conditions.

To perform the experiments, different processing units were used, the first is the Raspberry itself that represents the IoT device of the proposed architecture. The second device is a notebook, used for comparisons where you want to evaluate scenarios with different processing capabilities. Table 4 presents the comparison between the characteristics of the two devices.

Table 4 - *Comparison of the processing variables of the two devices.*

Model	**Raspberry Pi 3 Model B**	**Notebook Dell Vostro 3560**
Processor	Broadcom BCM2837 SoC @ 1.2GHz	Intel Core i7-3632QM CPU @ 2.2GHz
RAM	1GB LPDDR2	8GB DDR3
GPU	VideoCore IV 400 MHz	Radeon HD 7670M 900 MHz

5.1 System Evaluation on the original samples

5.1.1 Scenario 1 - Detection Efficiency

The first scenario aims to analyze the efficiency of both detectors tested. In this scenario, the following conditions were adopted:

- 100 samples for testing (20 individuals with 5 samples each)
- Different techniques for image processing
- Devices with different computing capabilities

Both techniques adopted achieved detection in 100% of the photos. Thus, the criterion to define which technique will be applied will be the detection time, since it needs a real-time operated system.

5.1.2 Scenario 2 - Average Detection Time

The objective of this scenario is to evaluate the processing time between the two techniques used to perform the face detection. In this scenario, the following conditions were adopted:

- 100 samples for testing (20 individuals with 5 samples each)
- Different techniques for image processing
- Devices with different computing capabilities

The processing time for each technique is obtained during the facial recognition process, this is done by using the two techniques in the same 100 samples. Then, a simple average is performed between the values obtained. The results found are presented in Figure 32. These results suggest what was commented by the authors in [88], about the CNN method requiring greater computational power to be used in real-time systems. Therefore, the choice of the HOG+SVM detection technique instead of CNN is justified by the difference in processing time within the analyzed set. As both techniques obtained a detection rate of 100%, the choice of the technique was defined by analyzing the processing time. As the model chosen was the HOG with training via

MMOD, all the next steps involving a recognition scenario will have this model implemented in the detection stage.

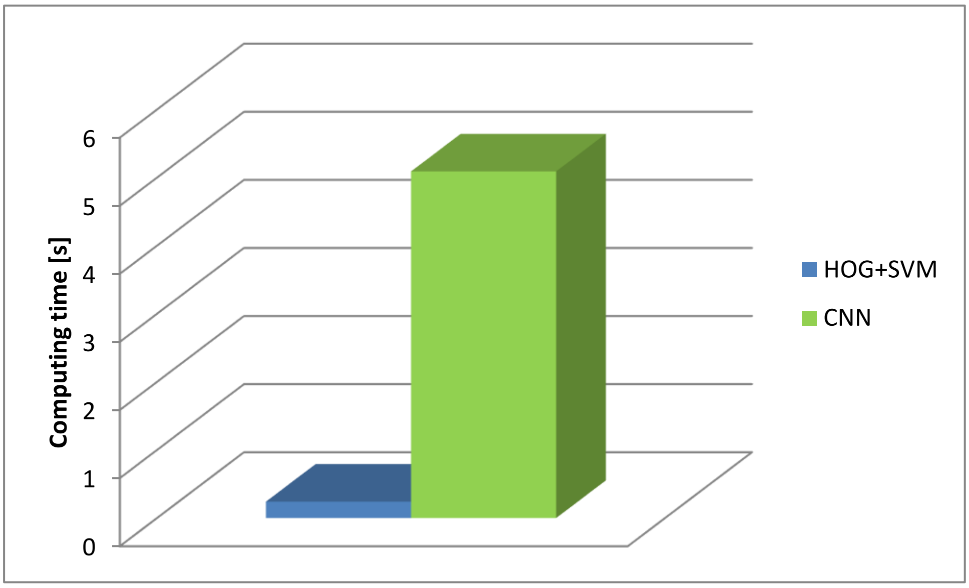


Figure 32 - *Comparison between the processing time of the HOG+SVM and CNN techniques, for facial detection.*

Figure 33 shows that when applying the CNN technique to a second device with greater processing power, there is a significant decrease in the detection time. Thus, it is possible to admit the possibility of implementing face detection in real time through CNN, which is possible by increasing the computational capacity of the devices or by the evolution of the technique itself.

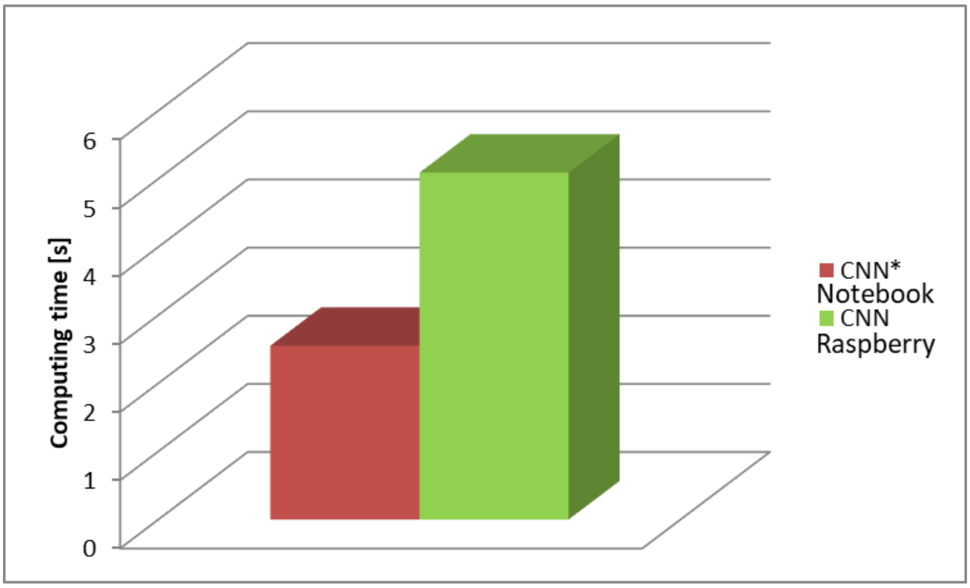


Figure 33 - *Comparison of the processing time of CNN technique on two hardware (Raspberry Pi 3 Model B and Notebook Dell Vostro 3560).*

5.1.3 Scenario 3 - Recognition Accuracy

The scenario 3 aims to verify the accuracy of face recognition of the model used. In this scenario, the following conditions were adopted:

- 13233 images for registration and testing (5749 individuals)
- ResNet Recognition Technique used
- Device running the benchmark (Notebook)

The standard LFW benchmark was used to evaluate the accuracy results. The idea of using a standard protocol is to be able to compare different recognition architectures under the same conditions, and with that we have a fairer comparison that uses the same metric.

The test had 13,233 images of 5,749 different people, of which 1,680 of them have more than one image. And it was done through cross-validation and the whole set is divided into 10 folders. The threshold set for the test was 0.6, following [87], and the closer to 1, the easier it is to detect a false positive, and the closer to 0, the easier a false negative to occur. As previously stated, the definition for a correspondence between faces is made via Euclidean distance.

For each face, the system tries to find its identity in the database by comparing the embedding of the entry with the embedding of all other images, excluding the input image itself. With this he scores the hits and errors to get the accuracy rate. The result is automatically presented, as shown in Table 5, because all faces are labeled and allow automatic computation of errors and right answers.

Table 5 - *Results of a LFW Benchmark.*

Folders	**Mean Accuracy**
Fold 1	99.5000%
Fold 2	99.1667%
Fold 3	99.1667%
Fold 4	99.0000%
Fold 5	99.6667%
Fold 6	99.6667%
Fold 7	99.0000%
Fold 8	99.5000%
Fold 9	99.6667%
Fold 10	99.5000%
Overall LFW Accuracy	**99.3833%**

To elucidate the performance results obtained, the expression that defines how accuracy was defined is displayed in the Equation (3). The equation is evaluated by the values of true positive (TP), true negative (TN), false positive (FP), and false negative (FN). In this experiment, false positives and false negatives correspond to the faces that were misclassified, while the true positives correspond to the assertiveness of the solution.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \tag{3}$$

5.1.4 Scenario 4 - Average Recognition Time

The objective of this scenario is to obtain the average time for face recognition. In this scenario, the following conditions were adopted:

- 100 samples for testing (20 individuals with 5 samples each).
- ResNet Recognition Technique used.
- Running on IoT Device (Raspberry).

Considering that the execution time is one of the crucial points of our system, we evaluated the model using time markers in the recognition stage. With the results obtained in this step, we can say that the real-time characteristic remains intact, since the recognition step has added only about 0.4 seconds in the total system time, reaching 0.64 seconds if added to the detection time. With that, the system still manages to deliver 1.56 fps, with a variation between approximately 1 to 3 fps in the practical tests, considering the main parameter that affects this value as the number of people in each frame.

5.2 Evaluation of the System with interference inclusion

The quality of an image is an essential parameter for its recognition to be satisfactory, in a real environment, some interferences that can generate low quality, which makes this recognition difficult. Examples of this are: poor lighting outside, low resolution cameras, blur, among others.

In this work, two analyzes are performed to measure the robustness of the system in the face of some type of interference. The first is performed by manually decreasing the lighting of the collected images and the second is done through the application of Blur.

As mentioned in [87], lighting and blur or low image resolution represent problems related to unrestricted face detection. Therefore, initially the following graphics are based on the 100 original photos without alteration and then consider an increase in noise, such as a decrease in brightness.

5.2.1 Scenario 5 - Assessment of blur interference

The objective of this scenario is to verify the behavior of the proposed image processing architecture, considering the application of gradual blur to the images. In this scenario, the following conditions were adopted:

- 100 samples for testing (20 individuals with 5 samples each).
- HOG+SVM Detection Technique used
- ResNet Recognition Technique used.
- Gradual blur being applied to the test set.

The analysis performed to assess the robustness of the system is done by applying a slight blur to the images and then this blur effect is increased according to the size of the filters. The graph in Figure 34 shows the variation in accuracy for different levels of blur. This test is performed to analyze the robustness of the system given the low quality that the cameras can have in practice.

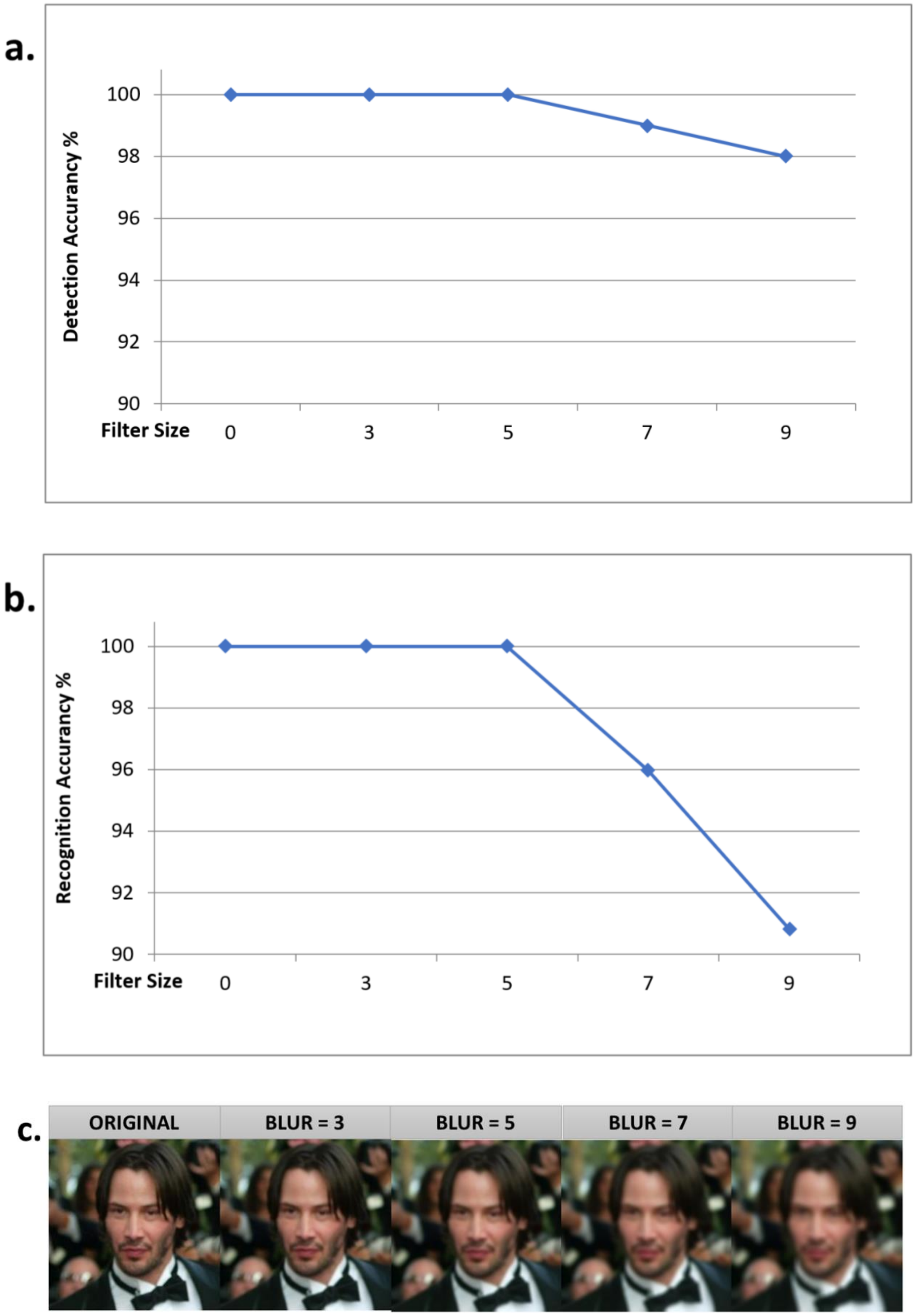


Figure 34 - *(a) Decay of detection accuracy for blur variations (b) Decay of recognition accuracy for blur variation (c) Image examples of blur variation.*

Considering the results obtained, it is possible to conclude that both techniques - detection and recognition - are highly robust to blur, presenting a more pronounced drop only in the last condition in which the blur is applied more intensely. Thus, it is possible to state that the system can operate satisfactorily even in cameras that do not have high resolutions.

5.2.2 Scenario 6 - Assessment of light interference

The objective of this scenario is to verify the behavior of the proposed image processing architecture, considering the gradual reduction of lighting in the images. In this scenario, the following conditions were adopted:

- 100 samples for testing (20 individuals with 5 samples each).
- HOG+SVM Detection Technique used
- ResNet Recognition Technique used.
- Gradual darkening being applied to the test set.

The analysis performed to assess the robustness of the system is done by manually decreasing the brightness of the images gradually. Initially we started with the natural image and a gradual variation of 20% is applied to the 100 images and increased with each new test. The Figure 35 shows low degradation of recognition rates with increasing interference. This proves that the system is even more robust against variations in light compared to tests with blur.

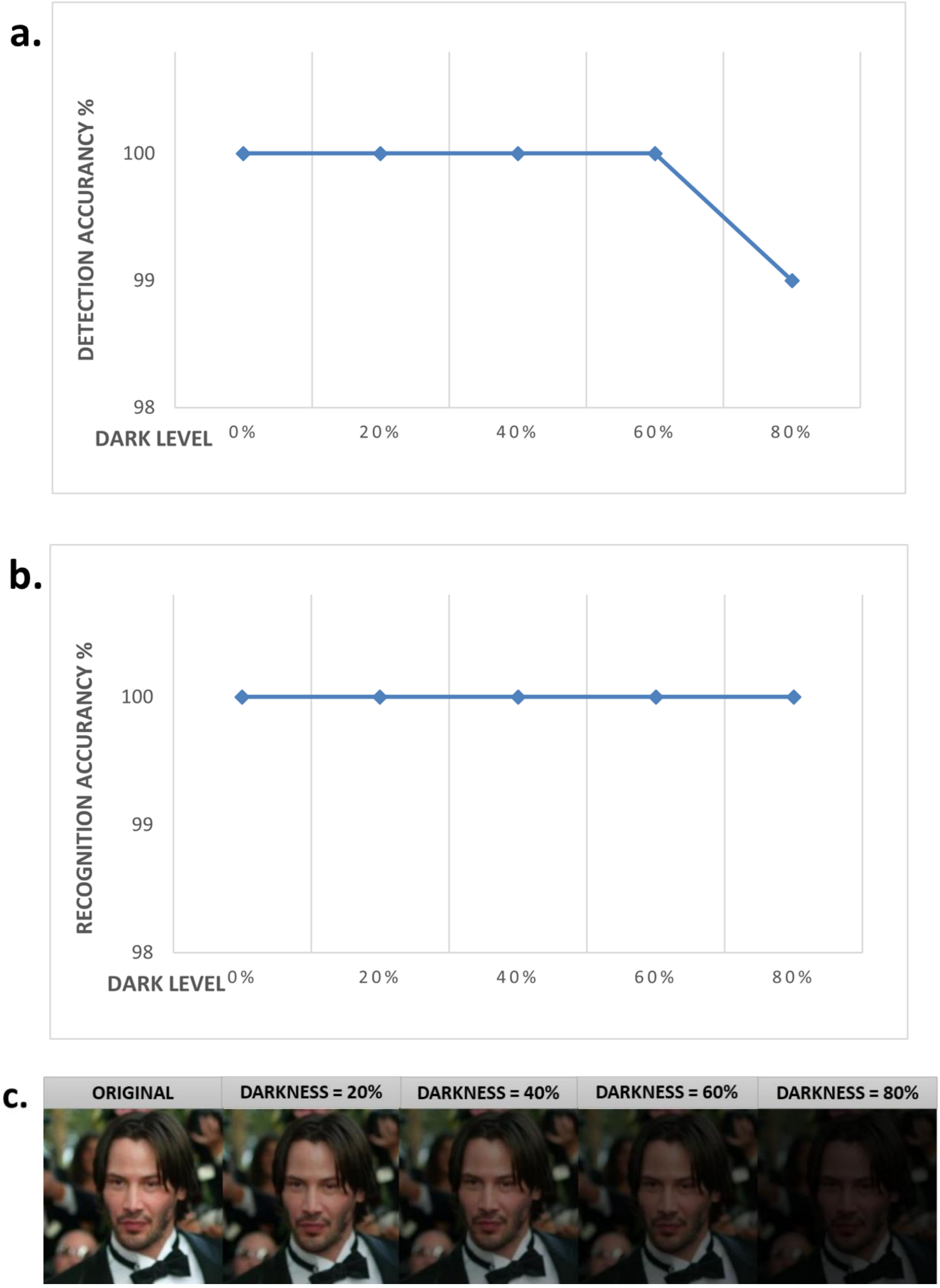


Figure 35 - *(a) Decay of detection accuracy for darkness variations (b) Decay of recognition accuracy for darkness variation (c) Image examples of darkness variation.*

6. Conclusion and Future Work

This chapter presents the lessons learned, key conclusions and future work of this study.

This dissertation was the result of analyzes of previous works related to the theme, computational experiments and practical implementations that allowed the learning of valuable lessons. In this sense, the main difficulties and relevant considerations about the research work carried out are pointed out below, in order to score their most important findings and allow the continuation of this and other similar research in future studies.

6.1 Learned Lessons

Throughout this study, important lessons have been learned regarding FR systems applied in an internet of things environment. These lessons will be scored below and may contribute to future research on the subject.

It was evident that the choice of detection and recognition techniques, especially in embedded scenarios, is a crucial factor in defining the response time of the application. The effectiveness of face detection and accuracy during the recognition step are also causally linked factors choosing these techniques.

As mentioned in Chapter 2, computer vision techniques have been undergoing constant evolution with the increasing emergence of new models. And in this scenario, the ILSVRC results show that CNN stands out in relation to other techniques, especially when item is precision.

From the detailed description of the architecture and technologies used, it will be possible for other researchers to be able to easily replicate the results obtained in this work, allowing a comparison with other models and techniques of detection and recognition. Moreover, the detailed flowchart of each process also clearly allows the understanding of the flow of the algorithm, even for a lay user in programming. It was also possible to note that the integration of Middleware with database allowed greater connectivity and availability of data to the IoT environment, and indirectly, provided reflections on the advantages of the different architectures that could be used to provide this type of data management and storage.

It is note point that the use of more popular tools has made development simpler, since many authors of articles and other academic-level works make open versions of their projects available in repositories. Discussion forums of these repositories are extremely indicated, both at the beginning of development and during the process, since they provide valuable information that helps solve some of the problems that researchers encounter during their implementations and studies.

The difficulty of finding implementations centrally hinders the learning process. Thus, it is suggested the centralization of implementations on a consultation basis, as well as academic articles. The lack of standardization of programming languages can also be a hindrance for a researcher who has no mastery over different languages, but among the searches for implementations in repositories, one of the most found languages was "python".

Still, CNN functions as a black box system, since they can be analyzed in terms of their inputs and outputs, without the need for knowledge of their inner workings. Knowing the inner workings of a CNN is exceedingly difficult due to the large number of nonlinear parts that interact with each other. The size of modern CNN, which can have up to tens of millions of parameters, makes this analysis even more difficult. A better understanding of the inner workings could lead to the development of more powerful architectures.

6.2 Main Conclusions

In this dissertation, a study of different techniques, tools and solutions was presented that can be used to implement an IoT-based facial recognition system that can meet real-time processing requirements.

During these analyses, we can conclude that the detection technique chosen allowed the system a rapid response and made it able to operate in real-time, even considering the subsequent step of recognition. And, that the numerical comparison between the techniques addressed allows us to observe in a practical way the difference in processing time between the two. It was also concluded that CNN's require a longer processing time, and it is important to use devices with a reasonable processing capacity to deliver a system that responds in real-time. Something that can be easily verified through the results, which demonstrate an improvement in processing time when using a device with greater processing power. In addition, a variation in the accuracy of the system was observed when the image undergoes some type of variation, such as luminosity or distortions of the image itself. This is a reasonable factor since even the human eye presents difficulties in the processing of poor-quality images or with more severe luminosity variations.

To evaluate the performance of the system, a real prototype was built, with the techniques embedded in a Raspberry Pi. This results in performance analysis by means of processing time and accuracy measurements. The experiments were carried out through suggested metrics for accuracy analysis in computer vision system, with the detailing of the software and hardware elements used to make up the developed prototype.

Middleware allows the system to make the generated data available so that it can be used by other applications on the Internet, a trend that is constantly growing since the increase in IoT devices already mentioned in this work. It also offers chart construction tools, tables, and other objects, which allow the user to build their own interface using the data provided by Raspberry.

With the measurements obtained during the experiments, quantitative and qualitative analyzes of the performance of each part of the system were made in an isolated and combined way. The results of these experiments allowed to find a solution that met the requirements of real time, high accuracy, and excellent robustness against external interference.

This does not mean that the techniques chosen are the best or faster ones, but rather that the system manages to deliver what has been proposed, despite its limitations and processing power.

6.3 Future Works

For future research and work, given the constant evolution of detection and recognition techniques, combined with the improvement of the processing power of the devices, it is proposed the analysis of different combinations of techniques, which offer interesting proposals and performances, not only for FR, but other tasks of detection, recognition, and classification within computer vision. And, given the number of innovations and new work on the subject that arise almost monthly in recent years suggests, delve into other FR techniques, such as ResNet.

Another analysis that can be explored is the possibility of implementing 3D computer vision techniques in IoT devices, considering their limited processing power. Three-dimensional analyses, rather than the 2D used in this work, can offer interesting features to increase the accuracy of systems.

Integrating the system with external applications in a large IoT scenario can generate a study on the main challenges of IoT-based computer vision systems, such as Smart Cities. Thus, it is possible to better evaluate the different models of DBMS and Middleware's that can be used to make up the structure of this system.

It is also necessary to research adaptive computer vision structures, embedded in IoT devices, that have the ability to autonomously choose between different techniques and resources for the processing of images, in order to better adapt to variations in the environment, such as in luminosity or even according to the number of objects processed at the same time in the image.

Another gap observed during the construction of this work was the assessment of the robustness and accuracy of other computer vision techniques, in relation to variations in the environment that the device is inserted. Several articles perform their analysis in environments with good lighting and few objects being processed at the same time, which can be different when we consider a real environment.

References

[1] A. R. Syafeeza, M. K. Mohd Fitri Alif, Y. Nursyifaa Athirah, A. S. Jaafar, A. H. Norihan, and M. S. Saleha, “IoT based facial recognition door access control home security system using raspberry pi,” *Int. J. Power Electron. Drive Syst.*, vol. 11, no. 1, pp. 417–424, 2020.

[2] Lavanya Sharma and Pradeep K. Garg, Ed., “From Visual Surveillance to Internet of Things: Technology and Applications,” Taylor & Francis Group, New York, 2020.

[3] A. A. Affonso, “Reconhecimento facial em ambientes não controlados por meio do " High-Boost Weber Descriptor " na região periocular,” Universidade de São Paulo, 2018.

[4] “IEEE Xplore Search Results.” [Online]. Available: https://ieeexplore.ieee.org/search/searchresult.jsp?newsearch=true&queryText =%22FACE RECOGNITION%22 AND %22IOT%22. [Accessed: 23-Nov-2020].

[5] A. Bertillon, “La photographie judiciaire : avec un appendice sur la classification et l’identification anthropométriques,” Editeurs de la Bibliotheque photographique, Paris, 1890.

[6] H. Han, C. Otto, X. Liu, and A. K. Jain, “Demographic Estimation from Face Images: Human vs. Machine Performance,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 6, pp. 1148–1161, Jun. 2015.

[7] E. H. Teixeira, S. B. Mafra, J. J. P. C. Rodrigues, W. A. A. N. Da Silveira, and O. Diallo, “A Review and Construction of a Real-time Facial Recognition System,” *An. do Simpósio Bras. Comput. Ubíqua e Pervasiva*, pp. 191–200, Jun. 2020.

[8] T. O. De Santana, “Comparação entre técnicas de aprendizado de máquina no processo de identificação biométrica através de imagens da orelha,”

Universidade Federal de Ouro Preto, 2019.

[9] J. Galbally, S. Marcel, and J. Fierrez, “Image Quality Assessment for Fake Biometric Detection: Application to Iris, Fingerprint, and Face Recognition,” *IEEE Trans. Image Process.*, vol. 23, no. 2, pp. 710–724, Feb. 2014.

[10] M. Alva, A. Srinivasaraghavan, and K. Sonawane, “A Review on Techniques for Ear Biometrics,” *Proc. 3rd IEEE Int. Conf. Electr. Comput. Commun. Technol.*, 2019.

[11] A. K. Jain and A. Kumar, “Biometric Recognition: An Overview,” in *Second generation biometrics: The ethical, legal and social context*, Dordrecht: Springer Netherlands, 2012, pp. 49–79.

[12] S. A. Alvi, B. Afzal, G. A. Shah, L. Atzori, and W. Mahmood, “Internet of multimedia things: Vision and challenges,” *Ad Hoc Networks*, vol. 33, pp. 87–111, Oct. 2015.

[13] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications,” *IEEE Commun. Surv. Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015.

[14] Y. Bin Zikria, M. K. Afzal, and S. W. Kim, “Internet of Multimedia Things (IoMT): Opportunities, Challenges and Solutions,” *Sensors*, vol. 20, no. 8, p. 2334, Apr. 2020.

[15] Z. H. Lin and Y. Z. Li, “Design and Implementation of Classroom Attendance System Based on Video Face Recognition,” *IEEE Int. Conf. Intell. Transp. Big Data Smart City, ICITBS*, pp. 385–388, 2019.

[16] Y.-H. Chuo, R.-K. Sheu, and L.-C. Chen, “Design and Implementation of a Cross-Camera Suspect Tracking System,” *Int. Autom. Control Conf.*, pp. 1–6, Nov. 2019.

[17] J. Gu *et al.*, “Recent advances in convolutional neural networks,” *Pattern*

Recognit., vol. 77, pp. 354–377, May 2018.

[18] S. Ahmad Radzi, M. Khalil-Hani, and R. Bakhteri, “Finger-vein biometric identification using convolutional neural network,” *Turkish J. Electr. Eng. Comput. Sci.*, 2016.

[19] K. He, X. Zhang, S. Ren, and J. Sun, “ResNet,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2016.

[20] Z. Lu, X. Jiang, and A. Kot, “Deep Coupled ResNet for Low-Resolution Face Recognition,” *IEEE Signal Process. Lett.*, 2018.

[21] X. Yu, Z. Yu, and S. Ramalingam, “Learning Strict Identity Mappings in Deep Residual Networks,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2018.

[22] A. Taherkordi, F. Eliassen, and G. Horn, “From IoT big data to IoT big services,” *Proc. Symp. Appl. Comput.*, pp. 485–491, 2017.

[23] K. Hong, D. Lillethun, U. Ramachandran, B. Ottenwälder, and B. Koldehofe, “Mobile Fog: A Programming Model for Large–Scale Applications on the Internet of Things,” *Proc. 2nd ACM SIGCOMM Work. Mob. Cloud Comput.*, pp. 15–20, 2013.

[24] N. Abbas, Y. Zhang, A. Taherkordi, and T. Skeie, “Mobile Edge Computing: A Survey,” *IEEE Internet Things J.*, vol. 5, no. 1, pp. 450–465, 2018.

[25] W. Li, X. Zhu, and S. Gong, “Harmonious Attention Network for Person Re-Identification,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, no. II, pp. 2285–2294, Feb. 2018.

[26] M. K. Bhuyan, *Computer Vision and Image Processing Fundamentals and Applications*. Boca Raton: CRC Press/Taylor & Francis Group, 2019.

[27] A. K. Jain, “Computer Vision,” in *Cutting Edge Technologies and Microcomputer Applications for Developing Countries*, 2nd ed., A. B.

TUCKER, Ed. Routledge, 2019, pp. 109–117.

[28] A. Vinay, B. V Sai Krishna, P. N. Manoj, A. Rao Nishanth, K. Balasubramanya Muthy, and S. Natarajan, “Person Identification in Smart Surveillance Robots using Sparse Interest Points,” *Procedia Comput. Sci.*, vol. 133, pp. 812–822, 2018.

[29] A. Goel, A. Khurana, P. Sehgal, and K. Suganthi, “Vision based Office Automation and Security System using Machine Learning and Internet of Things,” *Int. J. Eng. Technol.*, vol. 7, no. 2.24, p. 42, Apr. 2018.

[30] M. Valera and S. A. Velastin, “Intelligent distributed surveillance systems: A review,” *IEEE Proc. Vision, Image Signal Process.*, 2005.

[31] J. Guerrero-Ibáñez, S. Zeadally, and J. Contreras-Castillo, “Sensor Technologies for Intelligent Transportation Systems,” *Sensors*, vol. 18, no. 4, p. 1212, Apr. 2018.

[32] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications,” *IEEE Commun. Surv. Tutorials*, 2015.

[33] H. Arasteh *et al.*, “Iot-based smart cities: A survey,” *IEEE 16th Int. Conf. Environ. Electr. Eng.*, pp. 1–6, Jun. 2016.

[34] P. B. Balla and K. T. Jadhao, “IoT Based Facial Recognition Security System,” *Int. Conf. Smart City Emerg. Technol. ICSCET*, Nov. 2018.

[35] A. R. Syafeeza, M. Khalil-Hani, S. S. Liew, and R. Bakhteri, “Convolutional neural network for face recognition with pose and illumination variation,” *Int. J. Eng. Technol.*, vol. 6, no. 1, pp. 44–57, 2014.

[36] P. Viola and M. J. Jones, “Robust Real-Time Face Detection,” *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, May 2004.

[37] H. Z. Nafchi and S. M. Ayatollahi, “A Set of Criteria for Face Detection

Preprocessing," *Procedia Comput. Sci.*, vol. 13, pp. 162–170, 2012.

[38] A. Karishma et al.,, A. Karishma et al., A. Karishma, K. V. Anand Krishnan, A. Kiran, E. M. Dalin, and S. Shivaji, "Smart Office Surveillance Robot using Face Recognition," *Int. J. Mech. Prod. Eng. Res. Dev.*, vol. 8, no. 3, pp. 725–734, Jun. 2018.

[39] V. Jain and D. Patel, "A GPU Based Implementation of Robust Face Detection System," *Procedia Comput. Sci.*, vol. 87, pp. 156–163, 2016.

[40] R. Sharma, T. S. Ashwin, and R. M. R. Guddeti, "A Novel Real-Time Face Detection System Using Modified Affine Transformation and Haar Cascades," *Adv. Intell. Syst. Comput.*, vol. 707, pp. 193–204, 2019.

[41] S. R. Bodhi and S. Naveen, "Face Detection, Registration and Feature Localization Experiments with RGB-D Face Database," *Procedia Comput. Sci.*, vol. 46, pp. 1778–1785, 2015.

[42] M. Sajjad, M. Nasir, F. U. M. Ullah, K. Muhammad, A. K. Sangaiah, and S. W. Baik, "Raspberry Pi assisted facial expression recognition framework for smart security in law-enforcement services," *Inf. Sci. (Ny).*, vol. 479, pp. 416–431, Apr. 2019.

[43] V. Ruhitha, V. N. Prudhvi Raj, and G. Geetha, "Implementation of IOT based attendance management system on raspberry pi," *Proc. Int. Conf. Intell. Sustain. Syst.*, pp. 584–587, Feb. 2019.

[44] N. R. Brahmbhatt, H. B. Prajapati, and V. K. Dabhi, "Survey and analysis of extraction of human face features," *Innov. Power Adv. Comput. Technol.*, pp. 1–8, Apr. 2017.

[45] Z. Xiang, H. Tan, and W. Ye, "The Excellent Properties of a Dense Grid-Based HOG Feature on Face Recognition Compared to Gabor and LBP," *IEEE Access*, vol. 6, pp. 29306–29319, 2018.

[46] A. Patil and M. Shukla, "Implementation of Classroom Attendance System

Based on Face Recognition in Class," *Int. J. Adv. Eng. Technol.*, vol. 7, no. 3, pp. 974–979, 2014.

[47] P. Yadav, A. Poonia, S. K. Gupta, and S. Agrwal, "Performance analysis of Gabor 2D PCA feature extraction for gender identification using face," *2nd Int. Conf. Telecommun. Networks*, pp. 1–5, Aug. 2017.

[48] M. Sajjad *et al.*, "Raspberry Pi assisted face recognition framework for enhanced law-enforcement services in smart cities," *Futur. Gener. Comput. Syst.*, vol. 108, pp. 995–1007, Jul. 2020.

[49] A. A. Wazwaz, A. O. Herbawi, M. J. Teeti, and S. Y. Hmeed, "Raspberry Pi and computers-based face detection and recognition system," *4th Int. Conf. Comput. Technol. Appl.*, 2018.

[50] G. S. Nagpal, G. Singh, J. Singh, and N. Yadav, "Facial Detection and Recognition using OpenCV on Raspberry Pi Zero," *Proc. - IEEE Int. Conf. Adv. Comput. Commun. Control Netw.*, pp. 945–950, 2018.

[51] P. Espinosa, J. Pilataxi, L. Morales, and V. Benavides, "Vehicle Security and Alert System, Based on Facial Recognition and GPS Location," *Proc. - 2019 Int. Conf. Inf. Syst. Comput. Sci. INCISCOS 2019*, pp. 222–229, 2019.

[52] N. Purohit, S. Mane, T. Soni, Y. Bhogle, and G. Chauhan, "A computer vision based smart mirror with virtual assistant," *Int. Conf. Intell. Comput. Control Syst.*, no. Iciccs, pp. 151–156, 2019.

[53] M. A. Amaro, "Performance Evaluation IoT Middleware," Instituto Nacional de Telecomunicações, 2017.

[54] S. Haykin, *Redes neurais: princípios e prática*, 2nd ed. Porto Alegre: Bookman, 2007.

[55] O. Russakovsky *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[56] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[57] M. H. Ferreira, “Redes Neurais Artificiais: Princípios Básicos,” *Rev. Eletrônica Científica Inovação e Tecnol.*, vol. 1, no. 13, pp. 47–57, 2016.

[58] I. Goodfellow, B. Yoshua, and A. Courville, *Deep Learning*, 1st ed. Cambridge, MA, USA: The MIT Press, 2016.

[59] E. L. Faria, “Redes Neurais Convolucionais e Máquinas de Aprendizado Extremo Aplicadas ao Mercado Financeiro Brasileiro,” Universidade Federal do Rio de Janeiro, 2018.

[60] A. R. Bianchini, “Arquitetura de Redes Neurais para o Reconhecimento Facial Baseado no Neocognitron,” Universidade Federal de São Carlos, 2001.

[61] G. Chen, P. Xiao, J. R. Kelly, B. Li, and R. Tafazolli, “Full-Duplex Wireless-Powered Relay in Two Way Cooperative Networks,” *IEEE Access*, vol. 5, pp. 1548–1558, 2017.

[62] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, “A survey of the recent architectures of deep convolutional neural networks,” *Artif. Intell. Rev.*, vol. 53, no. 8, pp. 5455–5516, Dec. 2020.

[63] A. Solazzo, E. Del Sozzo, I. De Rose, M. De Silvestri, G. C. Durelli, and M. D. Santambrogio, “Hardware Design Automation of Convolutional Neural Networks,” *IEEE Comput. Soc. Annu. Symp. VLSI*, pp. 224–229, Jul. 2016.

[64] Y. Le Cun, J. S. Denker, and S. A. Solla, “Optimal brain damage,” *Adv. Neural Inf. Process. Syst.*, pp. 598–605, 1990.

[65] Y. Le Cun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[66] F. Chollet, *Deep Learning with Python*, 3rd ed. 2017.

[67] E. Ebermam, G. G. De Angelo, H. Knidel, and R. A. Krohling, “Empirical Mode Decomposition, Extreme Learning Machine and Long Short-Term Memory for Time Series Prediction: A Comparative Study,” *7th Brazilian Conf. Intell. Syst.*, pp. 492–497, Oct. 2018.

[68] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986.

[69] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” *ICML Work. Deep Learn. Audio, Speech Lang. Process.*, pp. 1–6, 2013.

[70] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, “Activation Functions: Comparison of trends in Practice and Research for Deep Learning,” *Comput. Vis. Pattern Recognit.*, p. 20, 2018.

[71] H. Ide and T. Kurita, “Improvement of learning for CNN with ReLU activation by sparse regularization,” *Int. Jt. Conf. Neural Networks*, vol. 5, pp. 2684–2691, 2017.

[72] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “2012 AlexNet,” *Adv. Neural Inf. Process. Syst.*, 2012.

[73] D. C. Cireşan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber, “High-Performance Neural Networks for Visual Object Classification,” *Adv. Neural Inf. Process. Syst.*, Feb. 2011.

[74] M. D. Zeiler and R. Fergus, “ZefNet,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, 2014.

[75] C. Szegedy *et al.*, “GoogLeNet,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2014.

[76] C. Szegedy *et al.*, “Going deeper with convolutions,” in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2015.

[77] G. Zeng, Y. He, Z. Yu, X. Yang, R. Yang, and L. Zhang, “InceptionNet/GoogLeNet - Going Deeper with Convolutions,” *Comput. Vis. Pattern Recognit.*, 2016.

[78] K. Simonyan and A. Zisserman, “VGGNet,” *3rd Int. Conf. Learn. Represent.*, 2015.

[79] K. He, X. Zhang, S. Ren, and J. Sun, “ResNet V1,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2015.

[80] K. Greff, R. K. Srivastava, and J. Schmidhuber, “Highway and residual networks learn unrolled iterative estimation,” in *5th International Conference on Learning Representations*, 2017.

[81] D. Valeriani and R. Poli, “Cyborg groups enhance face recognition in crowded environments,” *PLoS One*, vol. 14, no. 3, p. e214557, Mar. 2019.

[82] S. Liao, Zhen Lei, Dong Yi, and S. Z. Li, “A benchmark study of large-scale unconstrained face recognition,” *IEEE Int. Jt. Conf. Biometrics*, pp. 1–8, Sep. 2014.

[83] S. Liao, A. K. Jain, and S. Z. Li, “A Fast and Accurate Unconstrained Face Detector,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 211–223, 2016.

[84] A. Özdil and M. M. Özbilen, “A survey on comparison of face recognition algorithms,” *8th IEEE Int. Conf. Appl. Inf. Commun. Technol.*, 2014.

[85] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” *Proc. - IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition, CVPR 2005*, 2005.

[86] K. P. Feng and F. Yuan, “Static hand gesture recognition based on HOG characters and support vector machines,” *Proc. - 2nd Int. Symp. Instrum. Meas. Sens. Netw. Autom.*, pp. 936–938, 2013.

[87] D. E. King, “Max-Margin Object Detection,” *arXiv*, pp. 1–8, 2015.

[88] H. Li, Z. Lin, X. Shen, J. Brandt, and G. Hua, “A convolutional neural network cascade for face detection,” *IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 5325–5334, Jun. 2015.

[89] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao, “Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks,” *IEEE Signal Process. Lett.*, vol. 23, no. 10, pp. 1499–1503, 2016.

[90] V. Kazemi and J. Sullivan, “One millisecond face alignment with an ensemble of regression trees,” *IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 1867–1874, Jun. 2014.

[91] Y. Wu and Q. Ji, “Facial Landmark Detection: A Literature Survey,” *Int. J. Comput. Vis.*, vol. 127, no. 2, pp. 115–142, Feb. 2019.

[92] B. Amos, B. Ludwiczuk, and M. Satyanarayanan, “OpenFace: A general-purpose face recognition library with mobile applications,” 2016.

[93] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 770–778, 2016.

[94] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *3rd Int. Conf. Learn. Represent. Conf. Track Proc.*, 2015.

[95] X. Zhang, J. Zou, K. He, and J. Sun, “Accelerating Very Deep Convolutional Networks for Classification and Detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 1943–1955, 2016.

[96] D. Menotti *et al.*, “Deep Representations for Iris, Face, and Fingerprint

Spoofing Detection," *IEEE Trans. Inf. Forensics Secur.*, vol. 10, no. 4, pp. 864–879, 2015.

[97] D. E. King, "Dlib-ml: A machine learning toolkit," *J. Mach. Learn. Res.*, vol. Vol. 10, pp. 1755–1758, 2009.

[98] H.-W. Ng and S. Winkler, "A data-driven approach to cleaning large face datasets," *IEEE Int. Conf. Image Process.*, pp. 343–347, Oct. 2014.

[99] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep Face Recognition," *Br. Mach. Vis. Conf.*, pp. 1–12, 2015.

[100] D. Evans, "The Internet of Things: how the next evolution of the internet is changing everything," *CISCO white Pap.*, no. April, pp. 1–11, 2011.

[101] K. Pardini, "IoT-based Waste Managment: A New Approach for Smart Cities," Instituto Nacional de Telecomunicações, 2019.

[102] M. A. A. da Cruz, J. J. P. C. Rodrigues, J. Al-Muhtadi, V. V. Korotaev, and V. H. C. de Albuquerque, "A Reference Model for Internet of Things Middleware," *IEEE Internet Things J.*, vol. 5, no. 2, pp. 871–883, Apr. 2018.

[103] A. H. H. Ngu, M. Gutierrez, V. Metsis, S. Nepal, and M. Z. Sheng, "IoT Middleware: A Survey on Issues and Enabling technologies," *IEEE Internet Things J.*, pp. 1–1, 2016.

[104] V. A. C. Figueiredo, S. B. Mafra, and J. J. P. C. Rodrigues, "A Proposed IoT Smart Trap using Computer Vision for Sustainable Pest Control in Coffee Culture," *arXiv*. 2020.

[105] "in.IoT | Inatel." [Online]. Available: https://inatel.br/in-iot/. [Accessed: 24-Nov-2020].

[106] G. Fersi, "Middleware for Internet of Things: A Study," *Int. Conf. Distrib. Comput. Sens. Syst.*, pp. 230–235, Jun. 2015.

[107] S. Rautmare and D. M. Bhalerao, "MySQL and NoSQL database comparison

for IoT application," *IEEE Int. Conf. Adv. Comput. Appl.*, pp. 235–238, Oct. 2016.

[108] M. G. Jung, S. A. Youn, J. Bae, and Y. L. Choi, "A study on data input and output performance comparison of MongoDB and PostgreSQL in the big data environment," *Proc. - 8th Int. Conf. Database Theory Appl.*, 2016.

[109] M. M. Patil, A. Hanni, C. H. Tejeshwar, and P. Patil, "A qualitative analysis of the performance of MongoDB vs MySQL database based on insertion and retriewal operations using a web/android application to explore load balancing — Sharding in MongoDB and its advantages," *Int. Conf. I-SMAC (IoT Soc. Mobile, Anal. Cloud)*, pp. 325–330, Feb. 2017.

[110] M. M. Eyada, W. Saber, M. M. El Genidy, and F. Amer, "Performance Evaluation of IoT Data Management Using MongoDB Versus MySQL Databases in Different Cloud Environments," *IEEE Access*, vol. 8, pp. 110656–110668, 2020.

[111] "DB-Engines Ranking - popularity ranking of database management systems." [Online]. Available: https://db-engines.com/en/ranking. [Accessed: 29-Nov-2020].

[112] Y. S. Kang, I. H. Park, J. Rhee, and Y. H. Lee, "MongoDB-Based Repository Design for IoT-Generated RFID/Sensor Big Data," *IEEE Sens. J.*, vol. 16, no. 2, pp. 485–497, Jan. 2016.

[113] P. Wehner, C. Piberger, and D. Gohringer, "Using JSON to manage communication between services in the Internet of Things," *9th Int. Symp. Reconfigurable Commun. Syst.*, pp. 1–4, May 2014.

[114] N. Zhang and W. Deng, "Fine-grained LFW database," *Int. Conf. Biometrics*, Aug. 2016.