Revista ARACÊ
ISSN: 2358-2472

ANALYSIS AND EVALUATION OF FACIAL DETECTION AND RECOGNITION MODELS

https://doi.org/10.56238/arev7n3-003

Submitted on: 02/03/2025 **Publication date:** 03/03/2025

Rhuan Lima Ruiz de Oliveira[1], Sálvio Roberto Freitas Reis[2] and Rafael Oliveira Vasconcelos[3].

ABSTRACT

With technologies such as the Internet of Things constantly evolving, the opportunity arises for other areas, such as facial recognition, to incorporate new concepts into their operation and conquer new applications in smart cities. Public safety is one of the aspects that make up cities, with creative ways to improve emerging all the time, they also enjoy the benefits that the implementation of facial recognition systems brings. These systems are capable of capturing images of individuals and detecting faces through algorithms, which are submitted to models capable of recognizing these people by comparing them with a database. This work conducts a study on the facial recognition process, aiming at the analysis and evaluation of facial detection and recognition models in the context of public security.

Keywords: Facial Recognition. Face Detection. Safety. Internet of Things.

[1] Graduated in Computer Engineering
Department of Computing – Federal University of Sergipe
E-mail: rhuan.ruiz@dcomp.ufs.br
ORCID: https://orcid.org/0009-0008-8864-6194
[2] Master of Science in Computer Science
Graduate Program in Computer Science – Federal University of Sergipe
Email: salvio.reis@dcomp.ufs.br
ORCID: https://orcid.org/0009-0002-2924-3084
[3] Dr. in Computer Science (DI/PUC-Rio)
Graduate Program in Computer Science – Federal University of Sergipe
E-mail: rafael @ dcomp.ufs.br
ORCID: https://orcid.org/0000-0001-7974-304X

Revista ARACÊ
ISSN: 2358-2472

INTRODUCTION

With the advancement of technologies and the great impact of the Internet of Things and smart cities on daily life, a portfolio of possibilities presents itself in the most diverse areas that make up society. According to (SALAM, et al., 2019), the internet of things will revolutionize how society will work by connecting devices capable of making smart decisions with minimal human intervention.

Internet of Things *(IoT)* refers to any and all objects incorporated into other technologies, such as software and sensors, with the aim of promoting communication between these and other devices. From everyday items to industrial tools, as noted by (ILYAS, 2021), these objects are capable of monitoring and collecting a wide range of information, which is then processed using *edge computing* or cloud for decision-making. According to (ORACLE, 2020), the number of IoT devices is expected to reach 22 billion by 2025.

In this context, the Internet of Things emerges as a crucial component of smart cities. According to (ILYAS, 2021), smart cities take advantage of all available technologies to improve the quality of life of their inhabitants, encouraging greater community engagement, in addition to reducing operating costs and optimizing the use of public resources. The communication promoted by IoT devices directly benefits smart cities, and it is possible to observe its applications in the most diverse areas that compose it, including public safety.

With the continuous advancement of technologies, new approaches are emerging all the time to improve the public safety of smart cities. Closed Circuit Television (CCTV) systems represent one of these approaches, responsible for capturing and transmitting images from cameras, allowing recording and viewing of these images. For (DOSHI et al., 2022), the implementation of cameras is able to protect both goods and individuals in the region where the CCTV system is installed, offering benefits in preventing theft and detecting suspicious activity.

However, as emphasized by (ALSHAMMARI; RAWAT, 2019), the growing amount of data from video monitoring brings with it an increase in human errors, given the limit with which teams are able to process vast amounts of information. Therefore, for intelligent monitoring, it is necessary to implement facial recognition software. These tools are able to identify the face of individuals by measuring the facial components of an image.

Revista ARACÊ
ISSN: 2358-2472

Thus, the use of CCTV systems combined with the implementation of facial recognition software provides several opportunities for improving public safety. (DOSHI et al., 2022) says that facial recognition is responsible for making the system intelligent, facilitating the identification of suspicious individuals. These systems are already present in large volume in homes and establishments, making it feasible to use them for identification, sending images and information to the agencies responsible for the security of the cities.

JUSTIFICATION

Faced with the need to improve public safety in cities in a scenario where traditional methods are insufficient, the use of technologies emerges as an alternative to achieve such a feat. Today, individuals are already identified in different ways on a daily basis, through passwords, electronic keys, fingerprint identification, voice, among others. However, many of these methods may have security flaws or not be as effective in identifying citizens.

In this sense, facial recognition is a method on the rise, capable of capturing images of the faces of individuals and attributing them to an identified individual. It is a technology with great versatility, having applications in several areas, with the greatest benefit, compared to other methods, the ability to identify individuals passively, without the need for user interaction with the system. This feature is advantageous in identifying individuals who have criminal histories and who have their images submitted to a facial recognition system.

Given this scenario, the opportunity arises to carry out a study on the entire facial recognition process and carry out an analysis and evaluation of some of the available facial detection and recognition models, contextualizing it for public safety.

OBJECTIVES

General Objective

The objective of this work is to perform an analysis of the facial detection and recognition models available in the context of public security, aiming to use evaluation metrics to determine the most efficient models.

Specific Objectives

- Conduct a study on facial recognition;
- Analyze and discuss impacts of the use of facial recognition on public security and privacy;

Revista ARACÊ
ISSN: 2358-2472

- Analyze databases available for the purposes of the work;
- Discuss facial detection and recognition models;
- Analyze facial detection and recognition models.

THEORETICAL FOUNDATION

FACIAL RECOGNITION

The use of facial recognition techniques is widespread in the most diverse areas of society. In the context of public security, it is used for access control in restricted areas and for crowd control at public or private events, in addition to being used to identify suspicious activities. However, given the sensitivity of the data collected, a series of questions arise as to possible injuries to the privacy and integrity of individuals that may eventually be monitored.

In Brazil, monitoring with facial recognition is already widely used and shows a promising future, with several projects in the area under development. According to (ABDALA, 2023), almost 48 million Brazilians are potentially under monitoring with camera systems that use facial recognition, with a greater concentration in the Southeast and Northeast regions, respectively. Some scholars question the extent of this use, given that facial recognition has not yet shown significant practical results.

Facial recognition is a technique capable of recognizing individuals, through the analysis of the physical characteristics of their faces, enabling their identification. It is based on the principles and techniques of biometrics, having biometric patterns built from the unique facial features of an individual, such as eyes, eyebrows, nose, lips and jaw (BELUCO; FILHO, 2023). These patterns are then compared with previously stored records so that identification can be made.

Every facial recognition system has an operational model, divided into steps that will make up the entire face recognition process. In general, (CAMARA, 2021) says that the first stage consists of the general capture of the image, followed by facial detection. In this step, the human face is detached and segmented from the rest of the captured image, allowing for better analysis of facial features.

Then, the image is submitted to the extraction of attributes, concentrating on the acquisition of the geometric characteristics of the face. Subsequently, the facial image proceeds to the analysis stage itself, where the face is actually detected and the image is classified, so that it can be associated with an identity pre-registered in the system.

Revista ARACÊ
ISSN: 2358-2472

Depending on the system implemented, the image can go through extra steps throughout the recognition process.

Figure 1 – Facial Recognition Steps

Source: IDEC (2020)

Face Detection

The facial detection stage is responsible for identifying the human faces in the image and everything that may or may not be relevant to the recognition process. In other words, as mentioned by (IDEC, 2020), it is the moment responsible for categorizing the image or portions of it, segmenting the faces of the image from the rest of the frame and categorizing them as faces or not. As portions of the image may contain elements that are not of interest to the analysis, such as animals and objects, categorization is of paramount importance to the process.

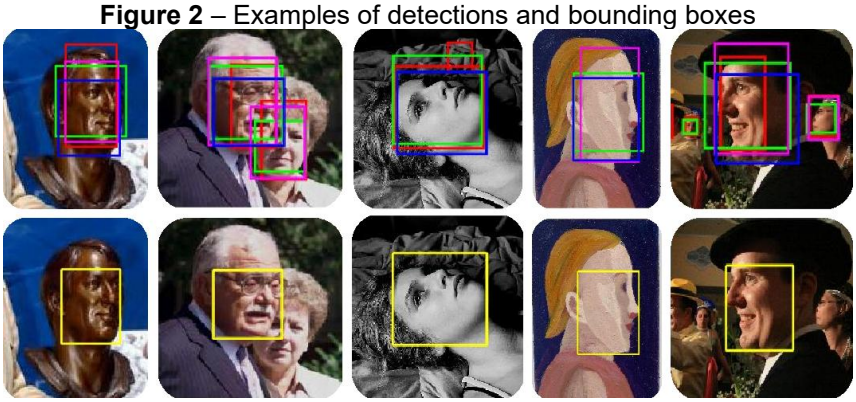
After detection, algorithms use *bounding boxes* to demarcate the faces. The bounding boxes are defined by the points of the upper and lower corners, are widely used in the most diverse detection tasks, in addition to describing the position and size of objects in the image. According to (AYADATA, 2023), boxes are used by algorithms so that they can learn about the content of an image, so that it is possible to collect interesting characteristics for the model and label them, facilitating the recognition of similar objects.

However, a number of factors can negatively affect the detection of one or more faces in the image. As mentioned, images may contain elements that are not interesting for analysis and that are discarded by algorithms, but (IDEC, 2020) shows how elements of the environment can also alter the perception of the image. Aspects such as lighting, face rotation are also some of the elements that can affect the reading of a face's features.

Revista ARACÊ
ISSN: 2358-2472

In this sense, systems can also incorporate intermediate steps between facial detection and the extraction of attributes. Some of the problems mentioned above can be minimized through the normalization process. According to (IDEC, 2020), during this process, patterns such as color, rotation, and lighting are changed in the image through a standard cropping, aiming to perform a more consistent analysis.

Figure 2 – Examples of detections and bounding boxes



Source: (FENG et al., 2017)

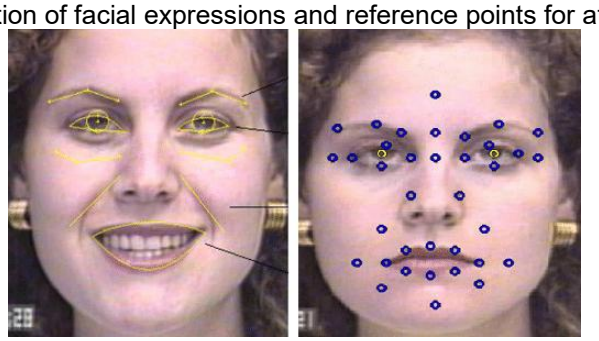
Attribute Extraction

After the face detection step is complete, the image proceeds to the attribute extraction process. During this step, all facial landmarks are analyzed, with relevant information about the image being extracted for later steps. As shown (IDEC, 2020), during the extraction phase, data on the geometric features of the faces such as shape, location, and distance of the facial components (mouth, nose, eyebrows, etc.) are identified and recorded for subsequent use.

The obtained characteristics are then organized and used to form an attribute vector. Properties and size of this vector may vary depending on the algorithm in use. According to (TIAN; KANADE; COHN, 2011), after the implementation of the sequence, the individual's face and the approximate location of the facial attributes are detected in the initial frame, with the rest of the attributes being identified after the algorithm is initialized. The effectiveness of the algorithm has a direct impact on the accuracy with which the faces of individuals under monitoring are identified.

Revista ARACÊ
ISSN: 2358-2472

Figure 3 – Extraction of facial expressions and reference points for attribute extraction



Source: (Tian; thorns; Cohen, 2011)

The role played by the attribute extraction step is crucial to the facial recognition process, as the attributes identified by the algorithms drive the rest of the steps. With the data obtained, facial recognition systems will have in their bases essential information that will contribute to the identification and comparison of the individual characteristics of each user who is submitted to the recognition process.

Logging and Analysis

Subsequent, depending on the properties of the implemented system, the data can be discarded or forwarded to the registration step. In it, data is effectively recorded and trained so that it can be compared between what is stored and what is obtained in real time. In this way, it is possible to identify an individual whose image has been captured by the system.

The facial recognition analysis stage goes through the categorization, authentication, and identification stages, so that they can be used independently or integrated, depending on the system implemented. In addition to the analysis itself, biometric models and images stored in the database are also compared to make identification.

Emphasized by (CAMARA, 2021), during categorization, the biometric image is classified into parameters such as age and gender. In authentication, it is where the verification of the individual occurs, comparing two biometric models and using statistical data to identify the probability of the individual being what the system expected. Finally, in identification, the image is compared with others stored in the database to identify the individual more comprehensively.

Revista ARACÊ
ISSN: 2358-2472

As illustrated in Figure 4, in the event of a successful verification, the individual is recognized and classified by the system as identified. On the other hand, if the verification is unsuccessful, the individual is classified as unknown.

CONVOLUTIONAL NEURAL NETWORKS

Convolutional neural networks, also known as CNNs (Convolutional Neural Networks), are a type of deep learning model designed to process data, such as images, structured in a grid format. According to (YAMASHITA et al., 2018), models are inspired by the animal visual cortex and designed so that they can automatically adaptively learn spatial hierarchies of both low- and high-level pattern characteristics. CNNs are commonly used in the most diverse tasks related to computer vision, in particular, they are successful solutions in the fields of facial recognition and object detection.

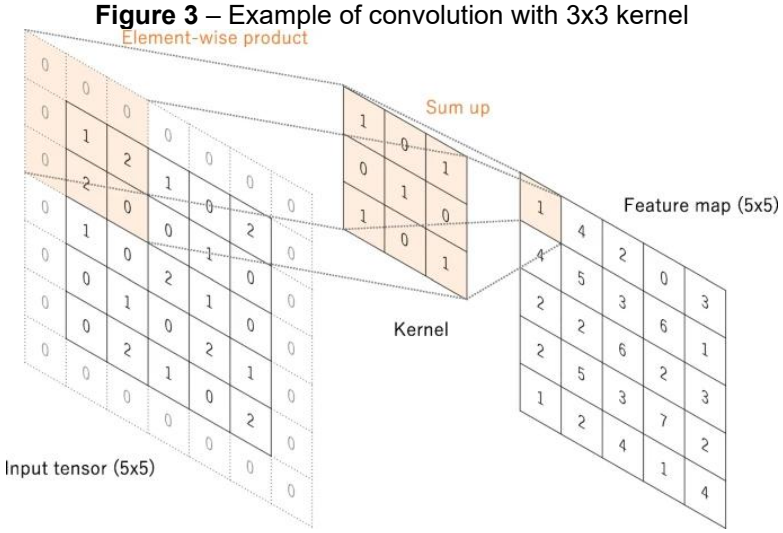
In addition, CNNs offer a more sophisticated solution compared to other techniques that still use manual attribute extraction. According to (ALBAWI; MOHAMMED; AL-ZAWI, 2017), CNNs do not use manual extraction of attributes and do not require the segmentation of anatomical properties by specialists, that is, it is a computationally more expensive solution, since it requires a much larger volume of data to learn its parameters. During their operation, CNNs must be able to locate attributes wherever they are in the array.

The convolution layer is one of the fundamental components of the architecture of CNNs. In it, filters are applied to the image, also known as kernels, which consist of vectors of numbers that slide throughout the image, generating the attribute map at the end of the process. For (INDOLIA et al., 2018), the convolution process occurs by sliding the filter both horizontally and vertically throughout the entire image, extracting attributes in a single layer, which represents distinct attributes. Therefore, at the end of the process, there will be N filters for N attribute maps.

Furthermore, the sharing of *kernels* at all points in the image is one of the fundamental elements of the convolution layer. According to (YAMASHITA et al., 2018), sharing allows the extracted attributes to be translation-invariant, as kernels detect learned patterns as they traverse the image. In addition, it is also possible for them to learn spatial hierarchies of attribute patterns and increase the effectiveness of the model by reducing the number of parameters to be learned.

Revista ARACÊ
ISSN: 2358-2472

Figure 3 – Example of convolution with 3x3 kernel



Source: (YAMASHITA et al., 2018)

The convolution layer is followed by the sub-sampling step, representing, in general, a downsampling operation. This layer is responsible for reducing the dimensions of the formed attribute maps by decreasing the number of parameters that can be learned, allowing direct learning from the data. For (INDOLIA et al., 2018), the biggest advantage of the subsampling stage is precisely the reduction of the attributes that can be trained, in addition to introducing invariance to translation.

The ability to invariance to translation and distortions, in addition to the precision in extracting attributes from facial images with their hierarchical learning, are some of the characteristics of CNNs. In this way, convolutional neural networks are powerful for facial recognition systems, allowing the extraction of discriminative attributes from faces and facilitating the recognition of individuals.

FACE DETECTION MODELS

Viola-Jones

The Viola-Jones algorithm is a popular method for detecting objects or faces in images, and is recognized for its speed of detection. It is capable of dealing with the aversions of the image environment and facial expressions, and is commonly used in facial recognition systems. Despite this, it is an algorithm that does not deal well with extreme variations in lighting and with the large differences caused by the variation of distinct objects (LU; YANG, 2019).

Its operation occurs over a few stages. Initially, the extraction of attributes takes place using filters called Haar, where the identification of patterns between faces, such as

Revista ARACÊ
ISSN: 2358-2472

eye position and nose, is carried out through the application of the filter in the image that calculates the different intensities between its pixels. Then, through the *Adaboost* algorithm, a waterfall classifier is trained by making a combination of weaker classifiers that are dependent on a single attribute in order to form a stronger classifier (VIOLA; JONES, 2001). At each stage of the cascade, a classifier with another, more specific attribute is applied until it reaches the point of detection or rejection for the rest of the flow.

Single Shot Multibox Detector (SSD)

Single Shot Multibox Detector (SSD) is a neural network designed to perform detection of several objects of different classes in the same step. It stands out for its detection accuracy, reaching a mAP (mean Average Accuracy) of 74.3% and 76.9% in inputs with 300x300 and 512x512 resolution, respectively, in tests carried out with the VOC2007 dataset (LIU et al., 2021), performing the tasks of location and classification in a single pass (single-shot) over the network. These characteristics make the SSD a neural network with more precision and accuracy compared to other single-shot detectors, such as YOLO (You Only Look Once), or detectors that use techniques such as pooling and RPN (Region Proposal Network) (LIU et al., 2021).

The SSD architecture is characterized by its multiscale detection, involving successive convolutional layers that are composed of convolutions of different scales responsible for detecting objects in different size ranges. The model divides the input image into a grid format, delimits bounding boxes into different sizes, and makes other predictions that include the location of the object and the probability of it being present in each boundary (LIU et al., 2021).

The SSD uses convolutional bases (VGG-16) to extract the characteristics of the image before being subjected to the multiscale step, in addition to using pre-defined bounding boxes similar to the anchor boxes used in the RPN (LIU et al., 2021). This pre-definition avoids the need for the model to pool the extracted facial attributes and, instead, assigns a score to each object in each of the bounding boxes, performing the detection and classification of the objects simultaneously.

Despite having a focus on general object detection, (YE et al., 2015) shows the model's potential for real-time facial detection. Its implementation is favorable for cases where speed is a priority, such as navigation systems, facial authentication on mobile

Revista ARACÊ
ISSN: 2358-2472

devices and especially for CCTV surveillance systems. However, the model loses performance in situations where it is necessary to manipulate multiple faces.

Multi-task Cascaded Convolutional Networks (MTCNN)

Multi-Task Cascaded Convolutional Networks (MTCNN) is one of the most recognized and widely used facial detection neural networks, standing out with the correlation between facial detection and alignment methods (RAJPUT, 2020). Proposed by Zhang et al. (2016), the model works in a cascade between three neural networks that are responsible for the process of detecting and refining bounding boxes. The model achieved a false positive rate of 0.9504 and a recall of 0.851 in tests conducted during its development (ZHANG et al., 2016).

Its waterfall architecture is composed of the P-Net (Proposal Network), R-Net (Refine Network) and O-Net (Output Network) networks. P-Net is the network responsible for identifying the candidate faces of the image and their initial bounding boxes through a suppression process called NMS, where redundant bounding boxes are eliminated and a regression vector is formed. Then, the R-Net network refines the input received, being responsible for reducing the number of false positives, calibrating the regression vector of the bounding boxes, and performing another NMS step. Finally, O-Net makes the final adjustments to the bounding boxes and describes five positions of facial references (ZHANG et al., 2016).

Systems that need to perform both detection and facial recognition tasks simultaneously can benefit from using MTCNN due to its accuracy in terms of face alignment. Because of this, its applications are commonly present in biometric systems, security systems in general, in addition to surveillance, CCTV or mobile systems themselves. Overall, the good alignment resulting from the model promotes more effective later facial recognition.

In (JOSE et al., 2019), the system is capable of using multiple cameras, identifying and keeping a record of suspects, sustaining an accuracy of 97% during the process. However, in situations where the model works with images in adverse situations it can result in a less efficient cascade, consequently affecting the overall performance of the model.

Revista ARACÊ
ISSN: 2358-2472

RetinaFace

RetinaFace is a model capable of calculating bounding boxes and key points of faces under varying conditions and sizes. The model excels in high-quality images, detecting faces in different size conditions, performing the detection of faces in a hierarchical manner through feature pyramids and predicting key points that facilitate detection and alignment for later recognition (DENG et al., 2020).

Its architecture uses the FPN (Feature Pyramid Network) and ResNet50 as another deep backbone network. Together, they provide attribute vectors from ResNet50 for the detection phase, contributing to the identification of faces of different scales, in addition to the alignment of the key points of the faces found throughout the process (COCHARD, 2024). This methodology makes the model extremely effective in detecting small faces in the image, in addition to ensuring greater accuracy in cases where there is a variation in poses, lighting, among other adverse conditions.

In light of this, the model flourishes in situations where multifaceted detection is mandatory. It is widely used by surveillance systems due to its characteristics of having high accuracy in situations of uncontrolled environment or scenarios with a high number of people, surpassing models such as MTCNN in direct tests (DENG et al., 2020). However, other models can still be prioritized in these cases due to the higher computational cost involved in implementing it.

MediaPipe

MediaPipe is a set of libraries and tools developed by Google with applicability in the fields of artificial intelligence and machine learning. In the field of facial detection, it provides solutions optimized especially for mobile devices, due to its ability to operate at a lower computational cost, being designed for web, Android and iOS platforms. The Face Detector is part of the library, being a model capable of locating faces, their attributes, and defining the key points of single images or a continuous flow of images (GOOGLE, 2021).

MediaPipe's architecture unfolds through a pipeline that uses a set of algorithms as it progresses through the detection stages. Using neural networks, it is able to identify the facial attributes of images and videos by delimiting the bounding boxes and key points of the face. This approach contributes to the versatility of the model, especially in situations that deviate from the standard of environments with large volumes and movement of individuals.

Revista ARACÊ
ISSN: 2358-2472

The model acts through data flow graphs, which allows concurrent processing of several steps of the detection process, as well as a higher execution speed. In addition, it is also capable of accurately tracking faces in motion or with changes in angle, allowing the model to act by performing continuous detection. The combination of precision and adaptive monitoring make the model efficient and accurate, and it is widely used in systems that require this type of task (MANJHI; RAJAWAT; SRIVASTAVA, 2023).

Overall, MediaPipe is a model that stands out for its optimization for mobile devices and its ability to detect continuously, which makes it an efficient model. These characteristics make it a solution aimed at situations where it is necessary to operate with more restricted resources, not reaching the level of robustness and precision of other models such as RetinFace and MTCNN. Its area of expertise is efficient in situations where speed and fluidity are required (GOOGLE, 2021).

FACIAL RECOGNITION MODELS

Eigenfaces

The Eigenfaces model offers a more simplified approach to performing facial recognition, being widely considered one of the most widely used models and serving as the basis for other facial recognition algorithms. For (ALAKKARI; COLLINS, 2013), Eigenfaces represents one of the first attempts to create a facial space and compress the data obtained from the faces, being used as a model for facial recognition based on principal component analysis (PCA).

The operation proposed by (TURK; PENTLAND, 1991) occurs from the stages of learning and recognition. During learning, the algorithm receives one or more of so-called training images, containing the faces of individuals so that the algorithm can learn about their model. Then, in the recognition phase, the training images are applied to the PCA technique, transforming the faces into vectors, reducing the dimensionality of the image and extracting the key attributes of the faces.

Subsequently, the image is projected into the subspace and the distances between the vectors of the image to be recognized and the training images are calculated. In the end, facial recognition is done based on the distance found, that is, closer distances represent an identified individual, while farther distances represent the face of an individual that the algorithm still needs to obtain more knowledge of (TURK; PENTLAND, 1991).

Revista ARACÊ
ISSN: 2358-2472

FisherFaces

Fisherface shares similarities with other models, especially Eigenfaces, but is considered more efficient than other algorithms because of its class separation during the training stage. According to Reddy and Kumar (2021), similar to the Eigenfaces model, Fisherface also uses principal component analysis (PCA) for dimensionality reduction, but only considers some image attributes to perform detection using linear discriminant analysis (LDA). Its class-separation feature makes the algorithm handle common aversions present in images well, such as variation in facial expressions and lighting.

With the application of PCA and the reduction of dimensionality, the LDA is applied to the attributes in order to find the best projections that offer greater distinction between the classes. Then, images of the same class are classified, distributing them more closely if the algorithm sees them as belonging to one individual and more distant if they belong to another. Thus, with the variance between the classes, the model is able to calculate the similarity between the classified images and determine the identity of the individual.

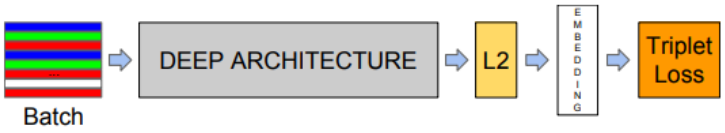
FaceNet

FaceNet is a deep learning model, developed by Google (2021), that works in the field of facial recognition using the so-called embeddings for learning the representations of faces. Embeddings form vectors of attributes that can be used in the identification and comparison of faces, standing out for their effectiveness in measuring similarity between faces. The faces identified and assigned to the same individual are stored in agropamentos called face clusterings (SCHROFF; KALENICHENKO; PHILBIN, 2015).

The functioning of the model occurs through neural networks that work directly on top of the embeddings, generating vectors of attributes per identified face (SCHROFF; KALENICHENKO; PHILBIN, 2015). These vectors contain general information about faces, such as eye shape, facial structure, among others. Through the triplets, it defines three embeddings: the anchor, which is the entrance face; the positive one that is another face but also belongs to the anchor; and the negative belonging to another individual. In this way, the model is able to measure and compare the faces with each other and determine if they belong to the same individual by comparing the distance between the vectors, so that the closer the vectors to the anchor and positive, the greater the chances of a true positive.

Revista ARACÊ
ISSN: 2358-2472

Figure 4 – Illustration of how the FaceNet model works



Source: (SCHROFF; KALENICHENKO; PHILBIN, 2015)

FaceNet has applications in biometric systems, authentication on mobile devices and in security, surveillance and monitoring systems. In tests carried out by the authors, accuracies of 99.63% and 95.12% were obtained in the LFW (Labeled Faces in the Wild) and Youtube Faces DB (SCHROFF; KALENICHENKO; PHILBIN, 2015), which highlights the model's ability to identify and compare similar faces. Such efficiency comes with the higher cost of pre-processing, given the quality of alignment required for the model to work accurately.

OpenFace

OpenFace is a neural network facial recognition model with Python and Torch-based implementation. It is capable of detecting facial attributes, making estimates related to the position of the individual's head and direction of gaze, among others. It is a flexible tool capable of being performed in real time, eliminating the need to implement an environment with specialized hardware.

Developed by (AMOS; LUDWICZUK; SATYANARAYANAN, 2016) and inspired by the FaceNet model, uses conjoined neural networks, which are designed to check the similarity between two images, to learn about the attributes of the faces that are submitted. It uses Torch, and its training was done using FaceNet to avoid training large volumes of data, since it focuses on real-time execution, such as mobile devices. Their training generates 128 facial embeddings representing attributes of the faces, as well as compressing the data to make their overall performance faster.

VGG-Face

VGG-Face is a deep learning model based on neural networks, also designed for the task of facial recognition. Using the architecture of the VGG (Visual Geometry Group), it performs small 3x3 convolutions, extracting more detailed and discriminative attributes throughout the recognition process. Like FaceNet and OpenFace, these attributes are also compressed into so-called facial embeddings.

Revista ARACÊ
ISSN: 2358-2472

In this way, embeddings make it easier to compare the face found in the input image with a face already known in the database. By calculating the distance between the embeddings, it is possible to determine the similarity between the faces, making the facial recognition process more agile. Because of this, VGG-Face is also better suited for applications aimed at real-time execution, such as public safety, authentication, and monitoring systems.

EXECUTION ENVIRONMENT

In facial recognition systems, the entire process can be done completely both locally and through the cloud. In the first case, the system is able to perform recognition directly from homes and establishments, offering advantages in terms of speed of detection and protection of sensitive images of users that may eventually be submitted to the system, given that there is no data sharing with the network. However, as the detection capacity is directly linked to the local device, this represents a limitation to the scalability of the system that may have a more restricted database compared to cloud solutions that have more complete infrastructures.

On the other hand, a cloud-running environment contrasts with the benefits offered by an on-premises running environment. Using users' sensitive data without their permission can bring legal obstacles, so that it opens the possibility of the system infringing the general law for the protection of personal data (LGPD) (DIÁRIO OFICIAL DA UNIãO, 2018). However, as the system does not depend only on the hardware of the local device, the system can take advantage of more scalable resources that the cloud can offer, with a larger database and the possibility of using several facial recognition models together.

The use of cloud-running environments in solutions focused on surveillance and security also promotes more versatility and processing power. In view of the vast amount of data required for optimal operation, the cloud allows the use of more robust models in its operation, such as MTCNN, FaceNet, among others. These features simplify the process of maintaining and integrating new features, making the system more accurate, efficient and up to date with technological advances.

Another factor to consider is the capacity to access the network necessary for its implementation. Taking into account scalability, local execution promotes more reliability since the system does not suffer from the instabilities and absence of network that some

Revista ARACÊ
ISSN: 2358-2472

areas have. Cloud systems can also suffer from higher response times as they require good communication with the server, which can impact their efficiency.

On-board devices

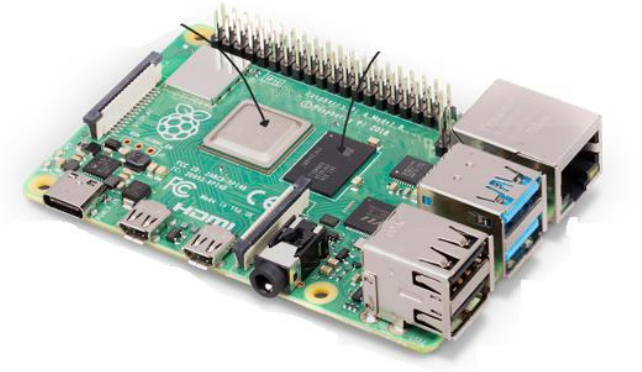
Embedded systems are those in which the computer is completely encapsulated or dedicated to a device, having a microprocessor to perform the control (SOUZA, 2018). They are developed to perform a specific task, being optimized in order to reduce size, computational and product costs, facilitating connection and use in the most diverse types of projects. It has application in practically all technological branches, such as mobile devices, televisions, automobiles, among others.

Microcontrollers are encapsulated semiconductor circuits reduced to very small physical sizes that integrate CPU, memory, and I/O devices, and that make connections through terminals, aimed at controlling embedded systems. This set formed by the CPU, memory and peripherals is described as an integration (embedded) on the same chip, which makes microcontrollers ideal for applications in which the cost of the product and space occupied are more relevant considerations than computational power (MAZIDI; MCKINLAY; CAUSEY, 2008). The Arduino and Raspberry Pi are some of the boards available and affordable for testing.

Figure 18 illustrates a Raspberry Pi, which is a low-cost microcomputer described as an SBC (Single-Board Computer), with great versatility and applications in learning, automation and robotics. Its proposal is to offer limited accessible computing power, with development focused on making systems cost-effective and efficient (GUPTA et al., 2016). Due to these characteristics and the vast support offered, the Raspberry Pi is widely used in applications aimed at education and prototyping, making it powerful in facial recognition tasks when coupled with computer vision tools. Despite the processing limitation that Raspberry has, it is able to act efficiently in security, monitoring, and surveillance systems. Less complex facial detection and recognition models are capable of performing well, ideal for scenarios with limited resources. In addition, by incorporating other sensors and peripherals, such as GPUs, into its operation, the system's potential can be greatly enhanced.

Revista ARACÊ
ISSN: 2358-2472

Figure 5 – Raspberry Pi 4



Source: (RASPBERRY PI TRADING, 2019)

DATASETS

Datasets are datasets used in deep learning and machine learning projects so that algorithms can learn and perform the function for which they are proposed. Usually arranged in tabular format, organized in the format of rows and columns, they are exported in formats ranging from TXT text files to CSV (Comma-Separated Values) files. The data is processed by libraries such as Python's Pandas, which offer a wide range of tools to perform processing and visualization. In addition, the database is usually branched so that one set of data can be used in the training of the algorithms and another set during the testing stages, and may even use data from other sources after the training stage.

Regarding facial recognition, datasets are fundamental elements for the training and evaluation of models. For an efficient training of the models, it is necessary to have well-structured datasets that best fit the task proposed by the system, and should include a sufficient number of images, variations of poses, expressions, as well as diversity in the images in relation to age, ethnicity and gender. Some of the databases widely used in facial detection and recognition tasks include:

LFW (Labeled Faces in the Wild)

LFW (HUANG et al., 2007) is a database of face images focused on the task of facial recognition. It consists of more than 13,000 images of web faces with more than 5,000 identities, containing annotations for the names of the people in each image, in addition to ensuring that at least 1,680 people have one or more photos in the entire database. In addition, it has images in uncontrolled situations, with variations in lighting, pose and facial expression, making it a good choice for training systems aimed at

Revista ARACÊ
ISSN: 2358-2472

surveillance. It is a reference dataset in the evaluation of facial recognition models, aimed at studies and research that can contribute to the advancement of facial verification techniques.

IMDB-WIKI

It is common for datasets to be small to medium in size. In this context, IMDBWIKI (ROTHE; TIMOFTE; GOOL, 2018) aims to break away from this pattern by collecting images of more than 100,000 celebrities from IMDb and Wikipedia sites, grouping the name, gender, and date of birth data of all images. The result is a database composed of more than 500,000 images with great diversity of age, gender and ethnicity, with a special focus on age prediction and demographic analysis tasks. However, the large amount of data and its more than ten attributes present in the notes allow the database to also be used in facial recognition tasks.

CelebA (CelebFaces Attributes Dataset)

CelebA (LIU et al., 2015) contains more than 200,000 images of celebrities so that each image contains more than 40 annotation attributes. The database has a large number of images and annotations, as well as a vast diversity of individuals with images containing variations of pose, environment, facial expressions, accessories, and is widely used for the study of facial attributes due to the richness of the annotations. The base can be implemented for training and testing datasets for various computer vision tasks such as facial detection, facial attribute recognition, facial recognition, identification of key points, among others.

VGGFace

The VGGFace (PARKHI; VEDALDI; ZISSERMAN, 2015) contains more than 2.6 million images of faces collected from more than 2,622 identities containing annotations for the location of each face and coordinates of the bounding boxes. Created by the VGG (Visual Geometry Group) team, the base is focused on the training of neural networks in facial recognition, being fundamental in the development and training of several models, such as FaceNet and ArcFace. VGG works on the assumption that the celebrity identities found are public, providing a large amount of images and data for study purposes, making it ideal for facial recognition tasks.

Revista ARACÊ
ISSN: 2358-2472

UTKFace

UTKFace (ZHANG et al., 2017) is a database that has more than 20,000 images of faces ranging in age from 0 to 116 years, containing annotations for their own age, as well as gender and ethnicity of each of the images. The images contain variation in pose, lighting, resolution, occlusion, facial expression, among others, in addition to the aligned and cropped images with their respective key points noted. The dataset focuses on the tasks of detection and facial recognition aimed at age prediction, as well as analysis and evaluation of the accuracy of population groups.

Wider Face

Wider Face (YANG et al., 2016) is a database composed of images selected from another public dataset, Wider. The database contains 32,203 images with more than 393,000 faces, varying in scale, pose, occlusion, makeup, lighting, in addition to annotations of the coordinates of each of the faces present. Wider Face is organized into 61 event classes so that for each class 40% of the data is randomly selected to be used in training, 10% for validation and 50% for testing. Due to the large volume of faces present in the images of the base, it is excellent for performing tests and evaluating more complex detection models, such as MTCNN and RetinaFace.

FDDB (Face Detection Dataset and Benchmark)

THE FDDB (JAIN; LEARNED-MILLER, 2010) is an offshoot of the LFW dataset, containing 5,171 faces in a total of 2,845 images. It is a basis for the study of facial detection in uncontrolled environments, with the faces annotated in the shape of ellipses in order to make the detection task more arduous for the models. The base also provides images with angle, occlusion, and illumination variations, making it ideal for evaluating the performance of models using the metrics in section 2.7 as an evaluation methodology.

EVALUATION AND METRICS

Metrics are evaluation methods used to ensure the efficiency of various computer vision tasks, including facial detection and recognition. They are used to obtain comparative parameters, in addition to measuring the quality of the models, evaluating the detection power of the faces or the ability of a model to correctly identify the faces of individuals. For this, true and false positives are used for the calculation, achieving more

Revista ARACÊ
ISSN: 2358-2472

significant results depending on the model, task, cost of different incorrect classifications, and whether the database is in balance with the objective of the work (GOOGLE, 2024). The following are the main concepts related to model evaluation:

- True Positive (VP): Indicates a correct prediction that occurs when there is actually a real face in a detection performed by the model;
- True Negative (VN): Indicates a correct prediction that occurs when the model correctly identifies the absence of a face in a region or throughout the image;
- False Positive (FP): Indicates an incorrect prediction that occurs when there is no real face in a detection performed by the model;
- False Negative (FN): Indicates an incorrect prediction that occurs when the model fails and is unable to detect an existing face in the image.

Accuracy

Accuracy is the proportion of all positive classifications in the model that are actually positive, that is, the proportion of true positives (PV) among all detections performed by the model, indicating how many of the detected faces are actually true. Mathematically, it is defined as:

$$Accuracy = \frac{VP}{VP + FP}$$

Recall

It is the proportion of positives that were correctly classified as positive, that is, the proportion of true positives (PV) in relation to the total number of positive cases, indicating how many of the real faces in the image were actually detected. Recall is defined as:

$$Recall = \frac{VP}{VP + FN}$$

Acuration

Accuracy is the ratio of all correct classifications, true positives, and true negatives. It typically measures the model's ability to correctly associate faces with true identity, and is the primary metric for non-generic or unspecified tasks. However, it is also recommended to use other metrics for real-world cases, where the data set is unbalanced. It is defined as:

Revista ARACÊ
ISSN: 2358-2472

$$Acuration = \frac{VP + VN}{VP + VN + FP + FN}$$

F1-Score

Provides balance between accuracy and recall metrics by calculating the harmonic mean between both metrics. The F1-Score takes into account both false positives (FP) and false negatives (FN) and is defined as:

$$F1 - Score = 2\ x\ \frac{Accuracy\ x\ Recall}{Accuracy + Recall}$$

DEVELOPMENT

DATASET

Initially, in the previous scope of the work, it was thought to use the VIRAT Video Dataset. The dataset consists of videos recorded outdoors of everyday actions and other recordings made by unmanned aircraft, with more than 20 types of events spread over more than 29 hours of video (OH et al., 2011), designed mainly for the analysis of video activities. Despite this, it was not ideal for facial detection and recognition tasks because its focus is not on capturing faces, but on monitoring and surveillance of human activity, with faces captured at long distances, which makes it difficult to detect and recognize them. In view of this, VIRAT proved to be inadequate for the scope proposed by the work.

Next, LFW (Labelled Faces in the Wild) was considered, given its reputation and efficiency in facial recognition tasks. However, the absence of annotations for bounding boxes proved to be a major obstacle to face detection tasks. This is because the LFW was designed mainly for recognition, not having annotations for the coordinates of the faces of its more than 13,000 images, which makes it unfeasible to use it for detection tasks (HUANG et al., 2007). We tried to annotate manually, but for the scope of the tests we chose to follow another dataset, considering that labeling them manually would affect the consistency and quality of the results.

Given this, the dataset chosen was IMDB-WIKI, containing more than 500,000 images with photos of celebrities from IMDB and Wikipedia. However, the sheer volume of images proved challenging, increasing the time required to process all tasks, including adjustments, pre-processing, and resizing. Considering the computational limitations, it was then decided to use only the WIKI portion of the dataset, which contains a total of 62,328 images, reducing the number of images by almost 90%.

Revista ARACÊ
ISSN: 2358-2472

During the process of analyzing the WIKI portion, a number of defects were observed in the dataset. The images lacked size standardization, and others were corrupted or completely absent of content relevant to both detection and recognition tasks. In addition, it was found that many of the bounding box annotations were inaccurate, resulting in flaw detections and inconsistent metrics during testing. This scenario revealed that, despite its abundance, IMDB-WIKI did not offer what was necessary to test the facial detection and recognition models, being focused on other deep learning activities (ROTHE; TIMOFTE; GOOL, 2018).

Later, other datasets were contemplated, such as VGGFace, UTKFace and WiderFace. The VGGFace (PARKHI; VEDALDI; ZISSERMAN, 2015) also contains thousands of images of faces with various annotations, but faced problems similar to those found on IMDB-WIKI, such as the large volume of images and the variability in capture conditions. UTKFace (ZHANG et al., 2017) does not have information on the names of individuals or detailed notes of bounding boxes, which makes it inadequate for both detection and recognition. WiderFace (YANG et al., 2016) also lacks essential information such as people's names, making it unfeasible to use it for the facial recognition task.

In view of these difficulties, it was concluded to use two distinct datasets: one for detection and the other for recognition. For detection, the FDDB (Face Detection Data Set and Benchmark) was selected due to the quality of the bounding box annotations as well as its moderate size, facilitating processing and meeting the needs of an environment with limited resources. The dataset has about 5,171 faces in a total of 2,845 images, with a diversity of images at challenging angles and conditions (JAIN; LEARNED-MILLER, 2010).

Although the LFW proved to be inadequate for facial detection, it was decided to continue with the same for the recognition task also because of the quality of the annotations, but this time focused on the identification of individuals, such as name, but also because of its moderate size. The combination of these two datasets avoids many of the problems faced and enhances the results of the tests, making it more dynamic and less computationally costly.

PREPARATION OF THE ENVIRONMENT

Initially, the idea of conducting the tests of this work in a more practical environment, using Raspberry Pi, was contemplated. However, aiming at time optimization, the experiments were carried out in a local environment configured on a personal computer.

Revista ARACÊ
ISSN: 2358-2472

The machine had reasonable computing power, with 8 GB of RAM and a Ryzen 5 5600g processor with integrated video.

The absence of a dedicated graphics card and more memory, however, represented a limitation to the work. Many of the datasets covered in section 2.6 have large volumes of data, requiring more processing power, which resulted in even longer execution times. In addition, the 8 GB of RAM available limited the ability to load and process images in larger batches, forcing the data to be divided into smaller batches and increasing the total execution time.

For the implementation of the models, version 3.10.9 of Python was used, along with fundamental libraries such as TensorFlow, PyTorch, NumPy, OpenCV, Matplotlib and Pandas. The libraries were used for the implementation of the models, as well as manipulation, analysis and pre-processing of the images.

Jupyter Notebook was chosen as the development environment because of the convenience it offers for code adjustments and its more visual feedback, which was useful during the development and initial analysis of the models. However, for the tests themselves and the collection of performance metrics, it was decided to execute the codes directly in the terminal so that the computer would operate with as few applications running as possible, reducing the impact of memory consumption and processing during the experiments.

MODEL PREPARATION

Seeking to ensure the uniformity of the input data, the initial stage consisted of preparing the models by pre-processing the images of both datasets. All images were resized to the standard dimension of 224x224 pixels in order to reduce computational cost, in addition to standardizing data entry, enhancing the performance of the algorithms.

In addition, a Gaussian filter was applied to the images in order to soften them and reduce noise. Additionally, in order to eliminate unnecessary information during execution, corrupted images (or with inconsistent annotations provided by the authors of the datasets) were discarded to prevent compromising the performance of the models and that the metrics reflected their actual performance.

The implementation of the models was based on libraries such as TensorFlow and PyTorch. However, during this stage some challenges arose when trying to implement some models, such as SSD and ArcFace, due to their requirements with specific versions

Revista ARACÊ
ISSN: 2358-2472

of libraries not compatible with the rest of the environment, specifically TensorFlow, making it difficult to perform some planned tests, since the adjustment of one model could make others unfeasible. To overcome these difficulties, it was decided to focus on the models that presented the greatest feasibility of implementation without compromising the integrity of the environment configured so that it was possible to optimize the execution of the work during the proposed time.

EXECUTION OF THE TESTS

The models used in this work were implemented already pre-trained, skipping the training stage. This strategy was taken into account by time optimization, as well as the objective of testing the potential of the models against the chosen datasets, collecting results that directly reflected their performance. In this way, it was possible to collect the metrics and analyze more objectively the points in which each model stands out.

In view of this, the evaluated models obtained the results shown in Table 1 and Table 2, with variations focused on the way they are implemented as well as the context in which they are being tested, in the case of this work, the LFW and FDDB datasets. The results obtained also highlight the capabilities of the models in the tasks of detection and facial recognition, but it is also possible to discuss their applications in scenarios closer to the real world, such as the context of public security.

Table 1 - Results of Face Detection models

Model	Precision	Recall	Accuracy	F1-Score
MTCNN	0,92701	0,97021	0,94355	0,98488
MediaPipe	0,99311	0,92032	0,89623	0,95851
RetinaFace	0,99474	0,65925	0,65922	0,79463

The MTCNN obtained the best overall result, reaching values above 92% in all metrics, showing that it was able to generate few false positives, locating most of the faces that make up the LFW dataset. MTCNN's performance highlights the smooth functioning of its cascade of P-Net, R-Net and O-Net networks, from the identification of candidate faces, to the structuring of bounding boxes to final refinements. Therefore, it is possible to say that the strategy proposed by the model promotes a more refined detection, not being so affected by the variations that will eventually occur in everyday scenarios, such as pose and lighting.

Revista ARACÊ
ISSN: 2358-2472

However, it is observed that despite the high metrics, the accuracy was the lowest found, compared to the other algorithms. It is possible to justify this fact by stating that MTCNN is more dependent on the quality of the input images, and it is necessary to have images with the clearest facial attributes for the P-Net network to perform a satisfactory initial identification of the faces, something that LFW may not offer in its entire dataset extension.

By contrast, MediaPipe's results show that the model is slightly more conservative than MTCNN. During detection, the model was the second most accurate, with few false positives, however, it was recalled with a value of 92%, which suggests greater care to avoid incorrect predictions. This behavior can be explained by its architecture that unfolds through a pipeline using neural networks, being optimized for contexts focused on real-time execution.

This approach is advantageous in embedded and mobile applications, as mentioned in subsection 2.3.5, but may not be ideal for detection in more challenging contexts, such as public safety-oriented monitoring. In these situations, it is common for the input images to be presented in a wide range of variations such as lighting, sharpness, unusual angles, among others, which would make it difficult for the model to act. Therefore, the model proved to be reliable in contexts where accuracy and reduction of false positives are priorities.

RetinaFace achieved the highest accuracy among the models analyzed at 99%, evidencing its ability to reduce false positives. However, the recall was considerably lower, suggesting that a significant portion of faces in the images were not identified, also generating accuracy and F1-Score values close to 65% and 79% respectively, indicating a balance between correct and incorrect detection. The lower than expected performance can be justified by the fact that it is a model developed to minutely capture facial details and face challenges such as unfavorable lighting and diverse facial expressions. Therefore, this specialization can compromise its performance in situations with low-quality images or with several small faces, underestimating the existence of faces in the image.

Another possible explanation for the results obtained by RetinaFace may be related to the format of the FDDB dataset annotations, which uses bounding boxes in elliptical format. Although the annotations were converted to the expected format, this transformation may have introduced inconsistencies. Like all the models evaluated, RetinaFace is also dependent on the accuracy of annotations for its deep learning tasks,

Revista ARACÊ
ISSN: 2358-2472

which may have led to incorrect interpretations of facial regions, negatively impacting the aforementioned metrics.

Evaluation of Facial Recognition Models

Table 2 – Results of the Facial Recognition models.

Model	Precision	Recall	F1-Score	Accuracy	Average Time(s)
VGG-Face	0,90327	0,78075	0,87684	0,78549	0,34730
FaceNet	0,85068	0,67107	0,80313	0,67895	0,35880
OpenFace	0,74015	0,22748	0,44302	0,23221	0,15730

VGG-Face stood out with the strongest performance among the facial recognition models analyzed. It obtained 90% accuracy, following the pattern of the other algorithms analyzed, with its architecture demonstrating effectiveness in the context employed, with its methodology based on large and varied data sets, which most likely favored its ability to identify more facial attributes. However, the rest of the metrics were suboptimal, a behavior that was observed in all the models evaluated.

One of the reasons that can explain this phenomenon lies in the obstacles presented by the LFW dataset, which has images with a wide variety of facial expressions, lighting and angles, which can offer resistance to the performance of the models. Even so, it is important to highlight the favorable average completion time, even in the face of a challenging scenario, evidencing the adaptability of VGG-Face in the contexts in which it is implemented.

FaceNet's performance was slightly lower than VGG-Face's, but still satisfactory, with the model achieving 85% accuracy and F1-Score, while recall and accuracy were 67% and 80%, respectively. FaceNet stood out for its versatility, proving to be a consistent model during the evaluation, balancing computational cost and good results. Thus, considering the need to optimize resources, FaceNet proved to be a model capable of acting in monitoring systems aimed at public security.

Among the models evaluated, OpenFace had the most modest performance, with an accuracy of 74%, recall of 22% and F1-Score of 44%. These values reflect that while the model was able to avoid many false positives, it faced considerable difficulty in identifying true positives. It is likely that the cause of this was during its implementation, related to libraries necessary for the proper functioning of the model, indicating that with optimization and adjustments to the code or configurations of its dependencies, the model can perform more efficiently.

Revista ARACÊ
ISSN: 2358-2472

However, in the general context of the evaluation, and taking into account the performance of the three models, it is important to take into account that the tests focused on the performance of the model only on the basis used, with the absence of a stage focused on training. In addition, the annotations present in the datasets may diverge from the results obtained by the models, affecting the final performance during the collection of metrics.

CONCLUSION

The objective of this work was to explore and discuss concepts that incorporate the task of facial recognition focused on convolutional neural networks, as well as to implement and evaluate the performances of some models. For this, some of the most popular approaches were analyzed, focusing on pre-trained models, without specific training in the context of this work, in order to evaluate the performance and effectiveness of facial detection and recognition techniques in a simulated local environment, aiming to approach a more practical context, similar to that found by public security cameras.

In the initial scope of the work, the idea was to build a system focused on public security, capable of identifying suspects related to criminal activity, using the cameras of residential and commercial establishments. However, this more ambitious approach gave way to another with more focus on the discussion and evaluation of model performance, aiming at an analysis of the potentialities, limitations and applicability of these algorithms in the context of public security.

To achieve this feat, a wide range of datasets have been tested and implemented. The FDDB datasets were properly used for facial detection evaluations and LFW for facial recognition. The FDDB dataset was chosen for its quality and variety for detecting faces in cluttered environments, while the LFW provided a good basis for evaluating face recognition.

The face detection models evaluated were MTCNN, MediaPipe and RetinaFace, which achieved results ranging from high to more modest, depending on the characteristics and challenges of the respective models and dataset. RetinaFace, in particular, was more sensitive to variations in dataset annotations, while MediaPipe stood out for its accuracy. MTCNN, on the other hand, obtained the best results, although it still had some limitations in detecting faces in more complex environments.

Revista ARACÊ
ISSN: 2358-2472

The facial recognition tests were carried out on the VGG-Face, FaceNet and OpenFace models. VGG-Face stood out with good metrics and better overall performance, while FaceNet, even with difficulties in identifying true positives, was still accurate. OpenFace, on the other hand, underperformed the rest, with low accuracy and recall, possibly due to interlibrary conflicts or inconsistencies with LFW annotations.

In conclusion, this work showed the great potential that the evaluated facial detection and recognition models have in the most diverse applications. These algorithms have the quality to contribute significantly to public safety systems, offering an effective and high-performance platform and, in some cases, low computational cost. Linked to a specific training stage focused on the context in which its implementation is conceived, the performances collected here in this work can be further enhanced, contributing even more to monitoring systems aimed at public security.

Revista ARACÊ
ISSN: 2358-2472

REFERENCES

1. Abdala, V. (2023). Over 47 mi may be under facial recognition surveillance in Brazil. Rio de Janeiro, Brasil. Retrieved on March 19, 2024.

2. Alakkari, S., & Collins, J. J. (2013). Eigenfaces for face detection: A novel study. Retrieved on March 22, 2024.

3. Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). Understanding of a convolutional neural network. Retrieved on March 21, 2024.

4. Alshammari, A., & Rawat, D. B. (2019). Intelligent multi-camera video surveillance system for smart city applications. Available at: https://ieeexplore.ieee.org/document/8666579. Retrieved on March 12, 2024.

5. Amos, B., Ludwiczuk, B., & Satyanarayanan, M. (2016). Openface: A general-purpose face recognition library with mobile applications.

6. Ayadata. (2023). Bounding boxes in computer vision: Uses, best practices for labeling, and more. Retrieved on March 19, 2024.

7. Beluco, D. C., & Filho, J. L. F. (2023). Reconhecimento facial aplicado para registro de ponto. Retrieved on March 19, 2024.

8. Camara, G. (2021). Do reconhecimento facial: Estudo exploratório e análise comparativa entre Brasil e Portugal. Retrieved on March 19, 2024.

9. Cochard, D. (2024). Retinaface: A face detection model for high resolution images.

10. Deng, J., et al. (2020). Retinaface: Single-shot multi-level face localisation in the wild.

11. Diário Oficial da União. (2018). Lei nº 13.709. Lei Geral de Proteção de Dados Pessoais (LGPD). Brasília, DF. Available at: https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2018/lei/l13709.htm

12. Doshi, M., et al. (2022). Smart surveillance system using face detection for residentials. Available at: https://ieeexplore.ieee.org/document/9825346. Retrieved on March 12, 2024.

13. Feng, Z., et al. (2017). Face detection, bounding box aggregation and pose estimation for robust facial landmark localisation in the wild. Available at: https://www.researchgate.net/publication/316788799_Face_Detection_Bounding_Box_Aggregation_and_Pose_Estimation_for_Robust_Facial_Landmark_Localisation_in_the_Wild. Retrieved on March 19, 2024.

14. GeeksforGeeks. (2021). Ml | face recognition using eigenfaces (PCA algorithm). Available at: https://www.geeksforgeeks.org/ml-face-recognition-using-eigenfaces-pca-algorithm/. Retrieved on March 22, 2024.

15. Google. (2021). Guia de soluções do Mediapipe. Available at: https://ai.google.dev/edge/mediapipe/solutions/guide?hl=pt-br

16. Google. (2024). Classificação: Precisão, recall, precisão e métricas relacionadas. Available at: https://developers.google.com/machine-learning/crash-course/classification/accuracy-precision-recall?hl=pt-br

17. Gupta, I., et al. (2016). Face detection and recognition using Raspberry Pi. Available at: https://ieeexplore.ieee.org/document/8009092

18. Huang, G. B., et al. (2007). Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Available at: https://vis-www.cs.umass.edu/lfw

Revista ARACÊ
ISSN: 2358-2472

19. IDEC. (2020). Reconhecimento facial e o setor privado: Guia para adoção de boas práticas. Available at: https://idec.org.br/sites/default/files/reconhecimento_facial_diagramacao_digital_2.pdf. Retrieved on March 19, 2024.

20. Ilyas, M. (2021). IoT applications in smart cities. Retrieved on March 11, 2024.

21. Indolia, S., et al. (2018). Conceptual understanding of convolutional neural network - A deep learning approach. Available at: https://www.sciencedirect.com/science/article/pii/S1877050918308019. Retrieved on March 21, 2024.

22. Jain, V., & Learned-Miller, E. (2010). FDDB: A benchmark for face detection in unconstrained settings. Available at: https://vis-www.cs.umass.edu/fddb/

23. Jan, A., et al. (2017). Artificial intelligent system for automatic depression level analysis through visual and vocal expressions. Available at: https://www.researchgate.net/publication/318798243_Artificial_Intelligent_System_for_Automatic_Depression_Level_Analysis_Through_Visual_and_Vocal_Expressions

24. Jose, E., et al. (2019). Face recognition based surveillance system using FaceNet and MTCNN on Jetson TX2. Available at: https://ieeexplore.ieee.org/document/8728466

25. Liu, W., et al. (2021). SSD: Single shot multibox detector. Available at: https://arxiv.org/abs/1512.02325

26. Liu, Z., et al. (2015). Deep learning face attributes in the wild. In Proceedings of International Conference on Computer Vision (ICCV). Available at: https://mmlab.ie.cuhk.edu.hk/projects/CelebA.html

27. Lu, W. Y., & Yang, M. (2019). Face detection based on Viola-Jones algorithm applying composite features. Available at: https://ieeexplore.ieee.org/document/8806572. Retrieved on March 21, 2024.

28. Manjhi, A. K., Rajawat, A. S., & Srivastava, A. (2023). Design and analysis of programmed face monitoring system. Available at: https://ieeexplore.ieee.org/document/10370072

29. Mazidi, M. A., McKinlay, R. D., & Causey, D. (2008). PIC microcontroller and embedded systems.

30. MultiComp Lab. (2018). Openface 2.2.0: A facial behavior analysis toolkit. Available at: https://github.com/TadasBaltrusaitis/OpenFace

31. Oh, S., et al. (2011). AVSS 2011 demo session: A large-scale benchmark dataset for event recognition in surveillance video. Available at: https://ieeexplore.ieee.org/document/6027400. Retrieved on March 26, 2024.

32. Oracle. (2020). O que é IoT? Available at: https://www.oracle.com/br/internet-of-things/what-is-iot/. Retrieved on March 11, 2024.

33. Parkhi, O. M., Vedaldi, A., & Zisserman, A. (2015). VGG face dataset. Available at: https://www.robots.ox.ac.uk/~vgg/data/vgg_face/

34. Rajput, S. (2020). Face detection using MTCNN. Available at: https://medium.com/@saranshrajput/face-detection-using-mtcnn-f3948e5d1acb

35. Raspberry Pi Trading. (2019). Raspberry Pi 4. Available at: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/. Retrieved on March 26, 2024.

36. Reddy, N. V. M. C., & Kumar, K. (2021). Comparison of HOG and Fisherfaces based face recognition system using MATLAB. Available at: https://ieeexplore.ieee.org/document/9456366. Retrieved on March 22, 2024.

Revista ARACÊ
ISSN: 2358-2472

37. Rodrigues, S. R. dos S. (2020). Desenvolvimento de um sistema de reconhecimento facial. Available at: https://recipp.ipp.pt/handle/10400.22/17594. Retrieved on March 21, 2024.

38. Rothe, R., Timofte, R., & Gool, L. V. (2018). Deep expectation of real and apparent age from a single image without facial landmarks. International Journal of Computer Vision, 126(2–4), 144–157. Available at: https://data.vision.ee.ethz.ch/cvl/rrothe/imdb-wiki/

39. Saini, Y. (2020). Face recognition using Fisherfaces. Available at: https://iq.opengenus.org/face-recognition-using-fisherfaces/#google_vignette. Retrieved on March 22, 2024.

40. Salam, A., et al. (2019). The future of emerging IoT paradigms: Architectures and technologies. Retrieved on March 11, 2024.

41. Schroff, F., Kalenichenko, D., & Philbin, J. (2015). Facenet: A unified embedding for face recognition and clustering. Available at: https://arxiv.org/abs/1503.03832

42. Silva, A. L., & Cintra, M. E. (2015). Reconhecimento de padrões faciais: Um estudo. Available at: https://www.researchgate.net/publication/341625381_Reconhecimento_de_padroes_faciais_Um_estudo. Retrieved on March 20, 2024.

43. Souza, M. (2018). Sistemas operacionais de sistemas embarcados. Available at: https://medium.com/@matheussouza_42815/sistemas-operacionais-de-sistemas-embarcados-892edfc37cd

44. Tian, Y., Kanade, T., & Cohn, J. F. (2011). Facial expression recognition. Available at: https://www.researchgate.net/publication/227031714_Facial_Expression_Recognition. Retrieved on March 20, 2024.

45. Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. Available at: https://ieeexplore.ieee.org/document/6793549. Retrieved on March 22, 2024.

46. Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. Available at: https://ieeexplore.ieee.org/document/990517. Retrieved on March 21, 2024.

47. Yamashita, R., et al. (2018). Convolutional neural networks: An overview and application in radiology. Available at: https://insightsimaging.springeropen.com/articles/10.1007/s13244-018-0639-9. Retrieved on March 21, 2024.

48. Yang, S., et al. (2016). Wider face: A face detection benchmark. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Available at: http://shuoyang1213.me/WIDERFACE/

49. Ye, B., et al. (2015). Face SSD: A real-time face detector based on SSD. Available at: https://ieeexplore.ieee.org/document/9550294

50. Zhang, K., et al. (2017). Age progression/regression by conditional adversarial autoencoder. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Available at: https://susanqq.github.io/UTKFace/

51. Zhang, K., et al. (2016). Joint face detection and alignment using multitask cascaded convolutional networks. Available at: https://ieeexplore.ieee.org/document/7553523