INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA DE INTELIGÊNCIA ARTIFICIAL




Um Estudo Comparativo entre CNNs e Vision Transformers para Reconhecimento Facial em Sistemas de Autenticação

GUSTAVO LEVI VIEIRA FERREIRA
Outubro de 2025

P.PORTO

A Comparative Study of CNNs and Vision Transformers for Facial Recognition in Authentication Systems

Gustavo Levi Vieira Ferreira

Aluno nº: 1200829

Dissertação para obtenção do Grau de

Mestre em Engenharia de Inteligência Artificial

Supervisor: Carlos Fernando da Silva Ramos, Full Professor, Institute of Engineering - Polytechnic of Porto (ISEP/IPP)

Juri:

President: Luiz Felipe Rocha de Faria, Associate Professor, Institute of Engineering - Polytechnic of Porto (ISEP/IPP)

Members:

Diogo Emanuel Pereira Martinho, Assistant Professor, Institute of Engineering - Polytechnic of Porto (ISEP/IPP)

Carlos Fernando da Silva Ramos, Full Professor, Institute of Engineering - Polytechnic of Porto (ISEP/IPP)

Porto, Outubro 2024

Dedicatória

Quero dedicar esta tese à minha família, com especial menção aos meus irmãos, Guilherme e Bárbara.

Resumo

O reconhecimento facial tem vindo a consolidar-se como uma das soluções mais promissoras para sistemas de autenticação, ao combinar praticidade, rapidez e ausência de interação explícita por parte do utilizador.

Contudo, o seu uso em ambientes reais levanta desafios críticos, sobretudo no equilíbrio entre produtividade e segurança. Ataques de spoofing e tentativas de login fraudulentas representam ameaças significativas que podem comprometer a fiabilidade e segurança destes sistemas.

Assim sendo, esta tese propõe uma solução orientada para a implementação de um mecanismo de autenticação baseado em reconhecimento facial, capaz de aliar desempenho e resiliência face a diversas tentativas de ataques. Para tal, foram exploradas e comparadas arquiteturas de *Convolutional Neural Networks* (CNNs) e *Vision Transformers* (ViTs), avaliando o seu comportamento em termos de velocidade, desempenho e precisão.

Os resultados evidenciam as vantagens e limitações de cada abordagem, fornecendo uma possível arquitetura e desenvolvimento ao tema em questão, por forma a conseguir uma solução tanto útil como segura.

Palavras-chave: Reconhecimento facial, Autenticação, Convolutional Neural Network, Vision Transformer, Segurança, Spoofing

Abstract

Facial recognition has established itself as one of the most promising solutions for authentication systems, combining practicality, speed, and no explicit interaction on the part of the user.

However, its use in real environments raises critical challenges, especially in balancing productivity and security. Spoofing attacks and fraudulent login attempts pose significant threats that can compromise the reliability and security of these systems.

Therefore, this thesis proposes a solution that aims to implement a facial recognition-based authentication mechanism capable of combining performance and resilience in the face of multiple attack attempts. To this end, Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs) architectures were explored and compared, evaluating their behavior in terms of speed, performance, and accuracy.

The results highlight the advantages and limitations of each approach, providing possible architecture and development for the topic in question in order to achieve a solution that is both useful and secure.

Keywords: Facial Recognition, Authentication, Convolutional Neural Network, Vision Transformer, Security, Spoofing

Agradecimentos

Gostava de agradecer aos meus familiares que não só me possibilitaram toda a minha carreira universitária, como também me apoiaram em todos os momentos. Gostava também de agradecer a todos os colegas e professores com quem me cruzei neste caminho, especialmente aos professores Nuno Silva, que me orientou na licenciatura, e ao professor Carlos Ramos, que me orientou na escrita desta mesma dissertação.

Por último, gostaria de agradecer à instituição ISEP, por me ter aberto as portas e me ter proporcionado todo o ensino até aos dias de hoje. Termino esta minha fase de coração cheio e muito orgulho em tudo o que conquistei.

Contents

List of Figures

List of Tables

List of Code Blocks

Acronyms and Simbols

Lista de Acrónimos

CNN	Convolutional Neural Network
VIT	Vision Transformers
AI	Artificial Intelligence
PRISMA	Preferred Reporting Items for Systematic Reviews and Meta-Analyses
ReLu	Rectified Linear Unit
Seq2Seq	Sequence to Sequence
EMA	Exponential Moving Average
LFW	Labeled Faces in the Wild
AUC	Area Under the Curve
ROC	Receiver Operating Characteristic
MTCNN	Multi-task Cascaded Convolutional Neural Network
GDPR	General Data Protection Regulation
HTTPS	HyperText Transfer Protocol Secure
API	Application Programming Interface
ERP	Enterprise Resource Planning
MES	Manufacturing Execution System
GPU	Graphics Processing Unit
CPU	Central Processing Unit

Lista de Símbolos

η	Taxa de Aprendizagem
Θ	*Threshold*

1 Introduction

This chapter presents the introduction to this study, justifying the relevance of the authentication system based on facial recognition and the motivation for carrying it out. The focus of the research, the structure of the document and the research questions that will be applied will also be presented.

1.1 Context and Motivation

In recent years, Artificial Intelligence (AI) has revolutionized several sectors, providing significant advances in areas such as security, biometrics and human-computer interaction. Among the various applications of AI, facial recognition stands out as a superior technology for user authentication, offering a balance between robust security and convenience in accessing systems and services.

Traditional authentication methods, such as passwords and tokens, have demonstrated several limitations, including vulnerabilities to phishing attacks, complex management of multiple credentials and a less intuitive and slower user experience. In contrast, solutions based on facial recognition use unique biometric characteristics, significantly increasing security and simplifying the authentication process.

The adoption of facial recognition has grown exponentially, supported by significant advances in the accuracy and efficiency of this technology. For example, studies show that, in just four years, accuracy in recognizing and comparing photographs in databases has increased 20 times,

highlighting the rapid technical progress in this area [1]. In addition, advanced systems such as Apple's Face ID have a 1 in 1,000,000 chance of a random user unlocking someone else's device, illustrating the level of security achieved by biometric solutions [2]. Globally, the impact of this technology is evident, with more than 64 countries adopting surveillance systems with facial recognition [3] [4], and more than 131 million users using this technology daily [5]. All this data reinforces the potential and relevance of facial recognition as a superior solution for improving security and efficiency in authentication processes.

This thesis aligns with the research and work conducted to implement a facial recognition authentication system for Sistrade Software Consulting. Sistrade is a company that offers next-generation integrated management solutions that generate appropriate information for rapid decision-making, leading to increased productivity, and is already perfectly aligned with the pillars of Industry 4.0.

This study proposes, using deep learning and computer vision, to create a solution that not only strengthens the security of corporate access, but also simplifies the user experience by eliminating the need for multiple authentications.

1.2 Objectives

From a business perspective, maximizing productivity is a key priority. Reducing time wasted on authentication represents a straightforward yet highly effective way to achieve this goal. Facial recognition, by streamlining the login process, has the potential to not only enhance security but also improve overall efficiency in accessing platforms or websites [6].

Beyond productivity gains, such systems contribute to minimizing operational costs by reducing the dependency on IT support for password resets, which account for up to 20-50% of help desk requests in many organizations [7]. Moreover, by eliminating the need for managing multiple credentials, facial recognition systems enhance user satisfaction, making access to critical tools faster and more intuitive.

Implementing such systems demonstrates a forward-looking approach to technology adoption, positioning organizations as innovative and tech-savvy. As businesses increasingly compete with efficiency and user experience, leveraging advanced tools like facial recognition can offer a decisive competitive edge.

From a security standpoint, biometric solutions significantly reduce risks associated with traditional authentication methods, such as password sharing or phishing attacks, which are responsible for a large percentage of corporate data breaches [8].

This study aims to develop a facial recognition system that enables seamless user authentication. The system will require an initial collection of user images; however, it is essential that this process be lightweight and user-friendly to minimize inconvenience. Additionally, the system must prioritize ease of use and, most importantly, demonstrate robust resilience against intruders or attacks, ensuring a high level of trust and reliability.

To meet these objectives, the study will undertake an in-depth exploration of several critical domains, including:

- **Image Processing and Computer Vision**: Techniques to optimize image quality and ensure accurate facial feature extraction.
- **Neural Network and Their Performance:** Evaluating and leveraging state-of-the-art models for facial recognition tasks.
- **Convolutional Neural Network (CNN) and Vision Transformers (ViTs) and their Performance:** Assessing the potential of ViTs to outperform traditional Convolutional Neural Networks (CNNs) in terms of accuracy, efficiency, and robustness.

1.3 Research Question

This thesis explores the design and implementation of a facial recognition-based authentication system tailored for corporate environments. The study aims to address challenges in security, user experience, and system efficiency, leveraging advanced techniques in deep learning and Vision Transformers (ViTs). Given all these aspects, the main research question is:

How can a facial recognition-based authentication system optimize security, user experience and therefore increase productivity in corporate environments?

1.4 Document structure

This document is structured into six chapters, each detailing a critical component of the research, design, implementation, and evaluation process.

Chapter 1: Introduction - This first chapter provides an overview of the study, including its context and motivation, problem statement, objectives, research questions and a brief outline of the thesis structure. This chapter is critical to identify and understand the problem, develop the research questions and define the structure of the document.

Chapter 2: Literature Review - This chapter focuses on understanding the defined research questions defined above. For that, the PRISMA methodology will be applied with the conduction of a systematic review.

Chapter 3: Proposed System Design - This chapter outlines the conceptual framework and system architecture for the facial recognition-based authentication system.

Chapter 4: Implementation - The implementation chapter describes the practical steps involved in developing the system, including the training and evaluation of models, methods for performance optimization, and integration with Sistrade's corporate system. Lastly, this chapter also discusses security measures against adversarial attacks.

Chapter 5: Experimental Results and Analysis - This chapter presents the results of the system's evaluation, using metrics such as accuracy, precision, recall, and F1-score. It includes a performance comparison between CNNs and ViTs, usability testing outcomes, and a discussion of the system's limitations and areas for improvement.

Chapter 6: Conclusions and Future Work - The final chapter summarizes the study's key findings, evaluates how the objectives were met, and discusses the implications of the results. It also provides recommendations for future research and potential enhancements to the system.

2 State of the art

This chapter provides an overview of the theoretical background and state-of-the-art methods regarding facial recognition systems, followed by a systematic review of the proposed research questions. The theoretical introduction forms the basis for the approach taken in addressing these questions. Since the CNNs and ViTs form the backbone of the facial recognition system to be developed, this chapter outlines the principles of CNNs and ViTs, methodologies involved, and the relevance to the field.

2.1 Facial recognition systems

Facial recognition technologies consist of a class of biometric solutions that identify or authenticate people by comparing and analyzing facial features. Presently, these systems are of wide use, but their popularity will continue to spread even more across domains in the near future: security, healthcare, retail, personal devices, and others. In security, this area is widely used for surveillance and controlling access, enhancing safety in public space and private infrastructures. Consumer technology now sees it as an easy way of authentication for users on smartphones and laptops. It offers much-needed improvements to password-based authentication, which often fails because people forget passwords, reuse them across multiple sites, and are susceptible to hacking.

Facial recognition technology finds its origin in the pioneering work of Woodrow Wilson Bledsoe back in the 1960s when he built a system for the recognition of faces using measurements of the distance manually among various facial features like the eyes, nose, and mouth [9]. This manual approach laid the foundation for future improvements in the field.

In recent years, the fields of computer vision and machine learning have turned this into an artificial intelligence-powered recognition system from what used to be a process of handcrafted features. Nowadays, deep learning-based approaches, which learn features themselves directly from image data through CNNs mostly, are predominantly adopted and achieve unprecedented accuracy and scalability [10]. More recent innovations in incorporating attention mechanisms and ViTs have fostered this capability further, thus enhancing the global

relationship capture ability in facial data. Face Recognition has emerged to be one of the most significant parts of a modern technology ecosystem coupled with the availability of Big Data and increased computational resources. With the rapid growth in technology, challenges remain to be resolved, mainly with ethical considerations, data privacy, and biases in these systems; it is, therefore, a vibrant area of active research and development.

2.1.1 Neural Networks

Neural networks are a type of machine learning that is known for its similarity to the human brain. This association comes from the fact that these networks consist of interconnected neurons that communicate through synapses, meaning the output of each neuron serves as the input for the neurons in the next layer [11]. Each neuron has the function of performing a simple yet important task: it receives input, processes them through a mathematical function, and generates an output, so that the network can ultimately learn and make decisions. At the core of each neuron, functions are weights, biases and an activation function [12]. Weights determine the strength or importance of each input connection. These weights are adjusted and optimized during training of the network. Bias acts as serves as an offset, allowing the neuron to shift the activation function regardless of the result. Basically, the bias acts as a constant term added to the weighted sum ensuring that even if the inputs are zero, you can still adjust it in the future. Lastly, the activation function is what introduces non-linearity to the network, enabling it to learn and represent complex patterns. Without activation functions, a neural network would simply behave as a linear model [13].

Some of the most common activation functions are Sigmoid, ReLu (Rectified Linear Unit) and Softmax.

Sigmoid

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Sigmoid is one of the most used activation functions given its ability to normalize the output. This happens because the values of the output are between zero and one. In addition, as it is visible in Figure 1, the outputs of this function are normally very close to the extremes 0 and 1, making it perfect for binary tasks [13].

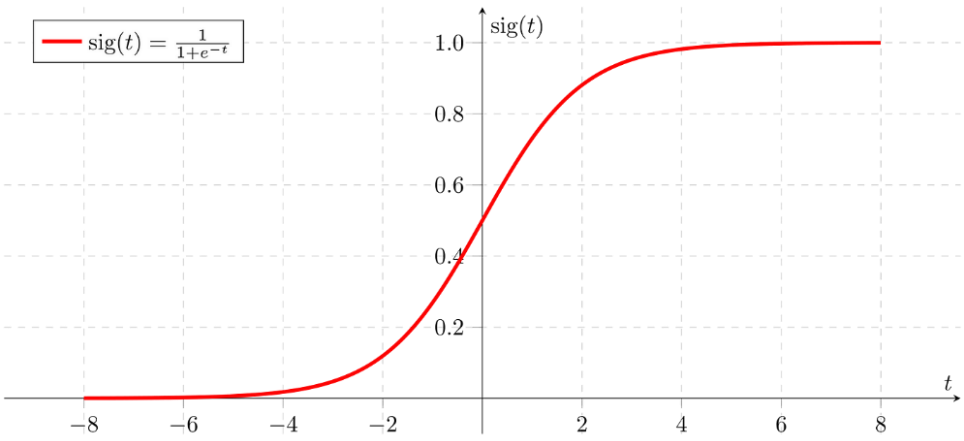

Figure 1 - Sigmoid Function [14]

Relu (Rectified Linear Unit)

$$\boldsymbol{f(x) = max(x)}$$

ReLu is the most used activation function in deep learning models and is normally found in the more internal layers of the models. As it is visible in the equation, this function is very simple yet effective: if the input is negative or zero, then the output is always zero and if the input is bigger than 0, its linear. This function is crucial because it helps avoid the problem of saturation, which occurs in functions like Sigmoid, where the output becomes very small and causes gradients to vanish during training [13]. By not having these small values, the network can give more importance to the neurons that actually matter. In addition, the fact this function is so simple, it speeds up the training of the network. The downside of this function is that it can make some neurons "die" because they will always produce the value zero [15].

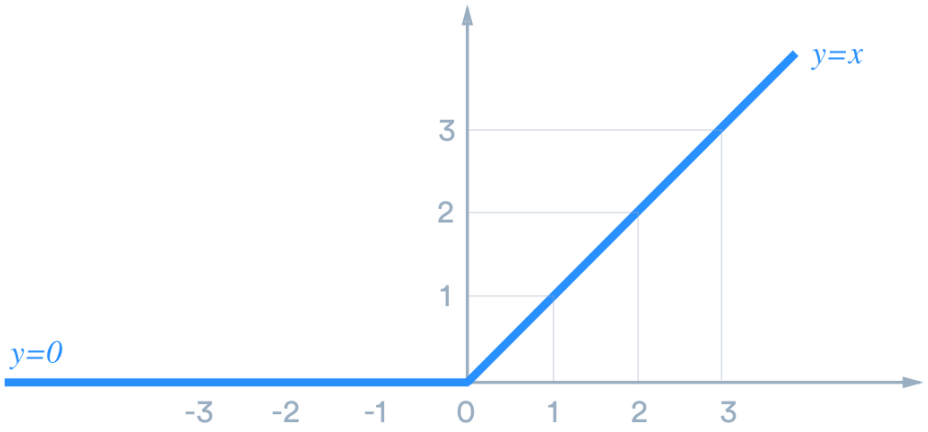

Figure 2 - ReLu activation function [16]

Softmax

$$softmax(x_i) = \frac{e^{x_i}}{\sum_j e^{x_i}}$$

This function also works by generating values between 0 and 1, but unlike Sigmod, it offers a much more diversity of results and not only extreme outputs. Softmax is commonly used in multi-class classification tasks because it converts raw scores into probabilities, where the sum of the probabilities of all classes is equal to 1.

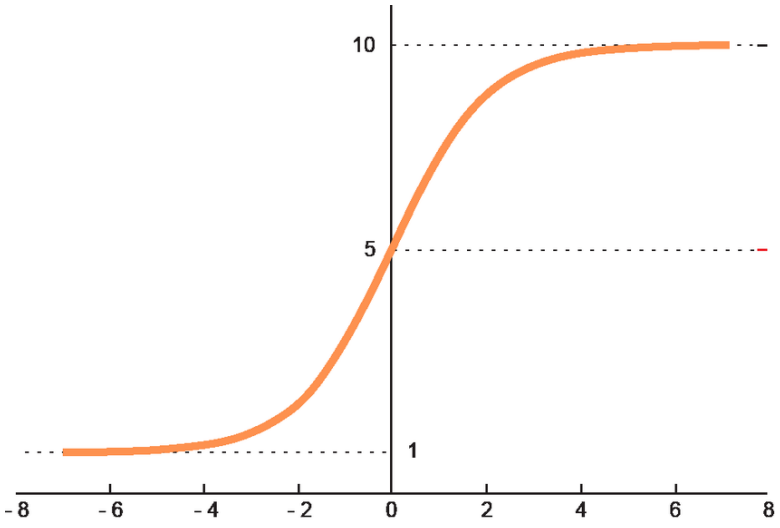

Figure 3 - Softmax activation function [17]

Neural networks operate in a cyclical learning process. After the activation function processes the input and generates an output, the network then enters the training phase. This phase is

where the weights and biases of the neurons are adjusted, using a feedback mechanism known as backpropagation, guided by a key component: the loss function.

The loss function is important because it quantifies how far the network's predicted output is from the expected result, where a lower loss indicates better performance. The loss function gives a value representing the error, and based on this value, the network adjusts the weights and biases [11] [15]. After that, the process repeats itself.

Lastly, it is also important to understand the different types of layers presented in Neural Networks. As was already mentioned above, the input layer is the first layer of the network and is the layer where raw data is fed into the network. After this layer, all the internal layers of the network are called Hidden Layers, and these layers often contain fully connected layers (dense layers) or convolutional layers. A Fully Connected Layer is a layer where every neuron of the layer is connected to every neuron of the next layer, whereas in a Convolutional Layer this does not happen. In the convolutional layer, each neuron is connected only to a small region of the input, called a receptive field. This happens because these layers apply filters (or kernels) to parts of the input data in order to detect patterns and make more localized connections [12] [18]. Finally, the Output Layer is the layer that produces the final predictions.

It is important to say that these filters or kernels are not pre-defined; instead, they are learned during the training process. The network automatically identifies these patterns, like edges, textures, or shapes, and that is what makes convolutional layers particularly useful in image recognition tasks.

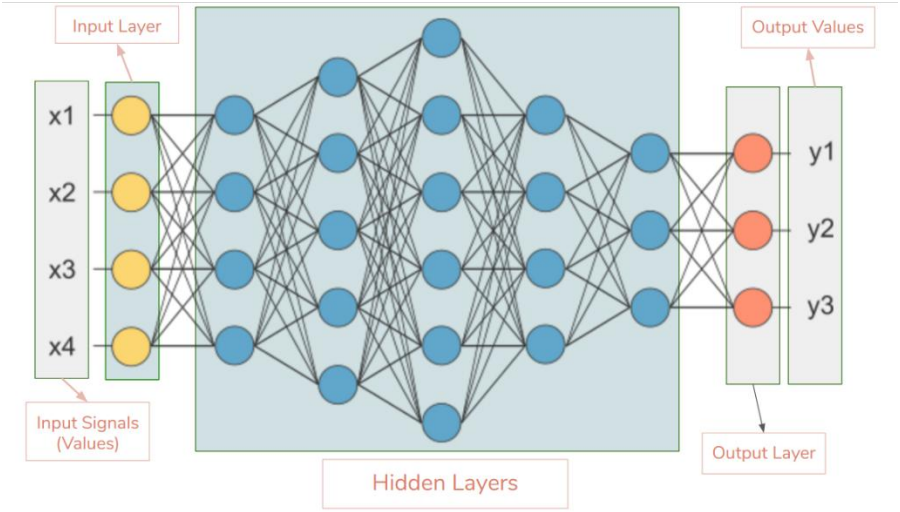

Figure 4 - Neural Network Layers [19]

In summary, the process training of these networks consists in the forward pass, where data is moving through the layers, an output is given, there is a calculation of loss, and then a backpropagation where weights and biases are adjusted, to minimize loss and achieve the optimal balance.

Now that the architecture and functions of neural networks have been discussed, the next section will focus on Convolutional Neural Networks (CNN).

2.1.2 Convolutional Neural Networks and Deep Learning

Convolutional Neural Networks (CNNs) are a type of neural network specifically designed for processing structured grid data, such as images. One of the key advantages of CNNs over other types of artificial neural networks is their ability to significantly reduce the number of parameters by sharing weights across the input data and not being fully connected [20].

The architecture of a CNN consists of the Input Layer, followed by multiple convolutional layers stacked on top of one another, each responsible for detecting features at different levels of complexity. The first layers detect simple features like edges, and the deeper layers detect more complex structures like faces or objects [21]. After these convolutional operations, there is an activation function (usually ReLu), so the model does not get linear.

After, it is normally applied one or more pooling layers, that have the job of reducing the dimension of the feature map. Lastly, the output is flattened, passed to a Fully Connected Layer and then goes to the Output Layer where that produces the network prediction [18]. Figure 5 is a visual representation of this architecture.

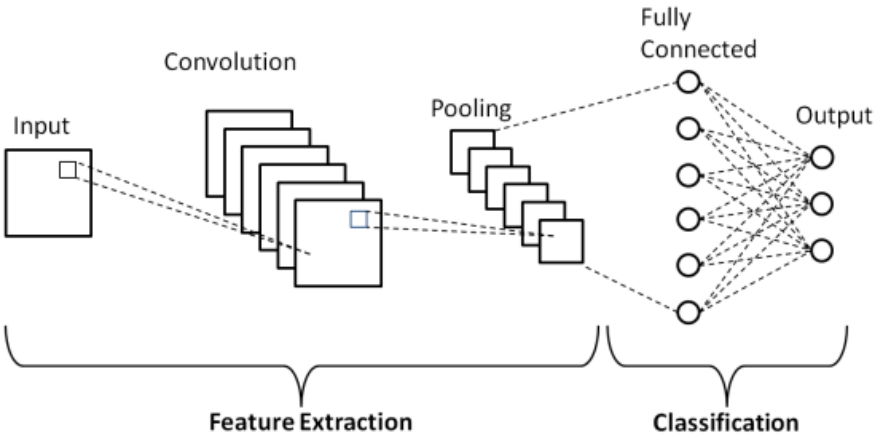

Figure 5 - CNN Architecture [22]

As explained earlier, each convolutional layer applies a set of kernels to the input data in order to extract features from the input. Initially, these filters start with random values, and through backpropagation, they get adjusted to recognize potentially useful patterns. These kernels are applied to small regions of the input data (receptive fields) and slide over each region of the image, which allows the network to focus on local features and learn spatial hierarchies [23]. As the network deepens, the convolutional layers combine simple features into increasingly complex representations.

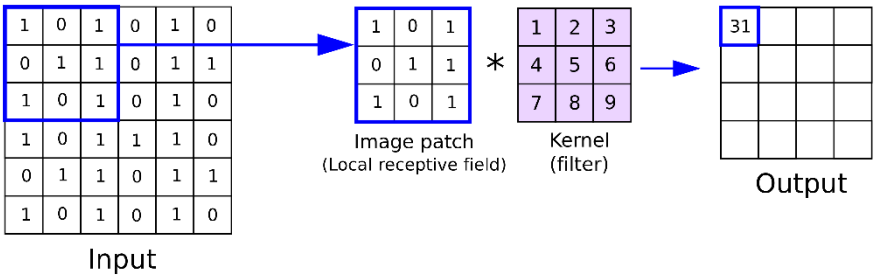

Figure 6 - Kernel in a Convolutional Layer [24]

This ability to learn and detect this hierarchical pattern makes CNN the foundation of many modern computer vision applications [25].

After the set of Convolutional Layers, pooling operations are often used in order to decrease the dimension of the feature maps, which is the output of the convolutional layers. This brings

not only the advantage of reducing the number of parameters and consequentially the computational requirement but also makes the network less invariant to small changes. There are two common types of pooling: Max Pooling and Average Pooling [18].

Max Pooling selects the maximum value in each sub-region of the feature-map, while Average Pooling average value of all the elements in the defined sub-region.

$$\boldsymbol{Max\ Pooling}(\boldsymbol{x}) = \boldsymbol{max}(\boldsymbol{x})$$

$$\boldsymbol{Average\ Pooling}(\boldsymbol{x}) = \frac{\boldsymbol{1}}{\boldsymbol{n}}\sum_{\boldsymbol{i=1}}^{\boldsymbol{n}} \boldsymbol{x_i}$$

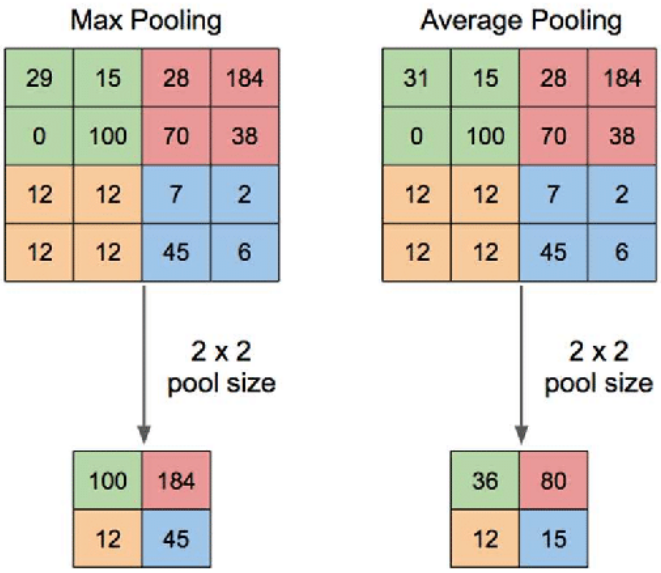

Figure 7 - Pooling Layers [26]

2.1.3 Transformers and Attention Mechanism

Transformers are a new type of neural network architecture, which comes to solve some problems related to the Seq2Seq models. Although it was initially designed for natural language processing tasks, its immense performance made it now dominate various other domains, including computer vision.

Seq2Seq models were often limited when dealing with long-range dependencies because it was hard to keep track and gave the needed importance to all parts of the input. RNNs for example, used to treat input sequentially to keep the order in place, but if the input had a long

dependency, this problem could happen. To solve it, Google Brain came up with the concept of Transformers.

First presented by the seminal paper “Attention Is All You Need" [27], this new model addresses this challenge by having a new groundbreaking mechanism: the self-attention mechanism. The self-attention mechanism allows the model to focus on different parts of the input sequence, instead of doing it sequentially, and ultimately leads to a parallelization.

At their core, transformers architecture consists of an encoder-decoder structure. The encoder is a stack of identical layers, where each layer has two main components: Multi-Head Self-Attention Mechanism that allows the model to weight the importance of different words or tokens in the input sequence relative to each other and Feedforward Neural Network, which servers to fully connect each token of the layer [27]. On the other hand, the decoder is also composed of stacked layers, but each layer has an additional component: an attention mechanism that focuses on the encoder’s output and allows to generate sequences given this output [27].

To summarize, the attention is used in two different parts. First it is used a self-attention mechanism where both the output and input focuses on different part of their own sequence, and then there is the encoder-decoder attention mechanism, also known as cross-attention, that allows to the model focus on parts of the input sequence while predicting the output sequence [27].

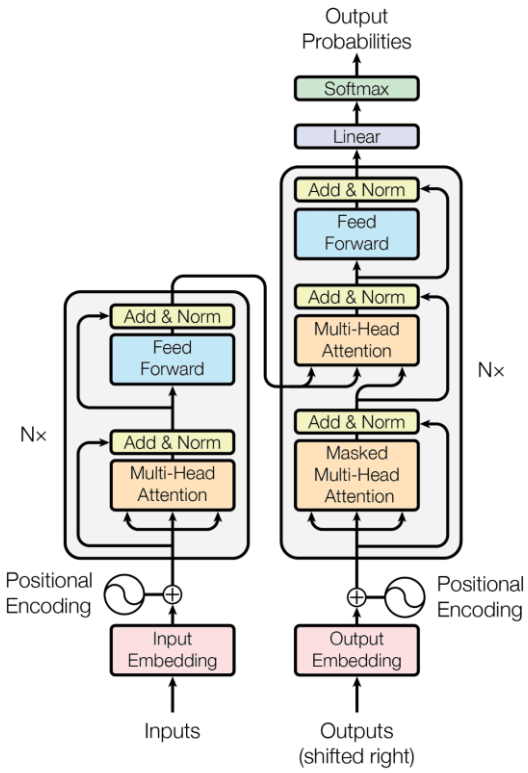

Figure 8 - Transformers Architecture [28]

To determine which part needs to be focused, the self-attention mechanism does a series of calculations. First, each input token is projected into three vectors: Query(Q), Key(K) and Value(V) [29]. These vectors are done by multiplying the input embedding by three matrices that were trained during the training process. For a given token, the attention score with every other token is calculated by taking the dot product of Q vector with the K vector of the other tokens, to understand how much focus the token should give to others [29]. These scores are then divided by the square root of the dimension of the key vectors and passed through a softmax function, converting them into normalized weights that sum to 1. Now that the values are normalized, the relative importance of each token is measured [29]. After, the softmax score multiplies each value vector, so the relative tokens can be emphasized and the irrelevant one discarded. Lastly, the model sums up the weighted value vectors and it results in the encoder of the self-attention layer to one word [29]. The Figure 9 shows a simple example of this process.

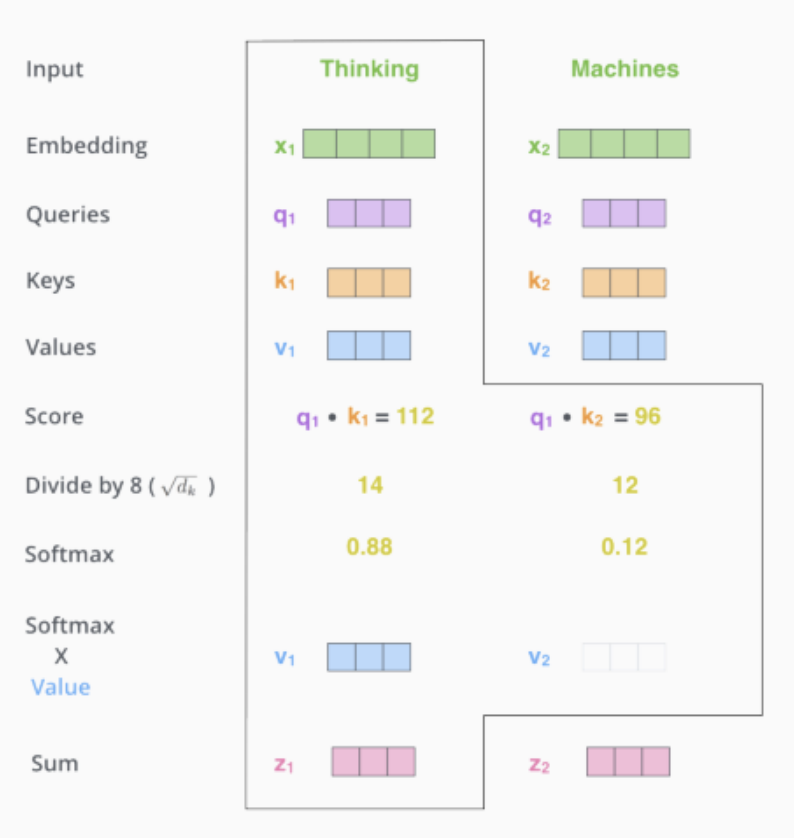

Figure 9 - Self-attention mechanism [30]

2.1.4 Vision Transformers

Vision Transformers, known as ViTs, extend the transformer architecture to image processing tasks. Up until the development of this model, convolutional neural networks had total dominance in computer vision, but the big performance of ViTs in recent years is changing this trend.

In this model, the idea is to represent an image as a sequence of patches and process it using the standard transformer architecture. The input image is divided into fixed-sized patches, and each patch is flattened and linearly projected into and embedding vector, making the image look like a sequence of tokens [31]. Then, the attention mechanisms used in this type of model allow the model to capture global relationships between patches. The difference in Vision Transformers is the Classification Token that is appended to the sequence of patch embeddings and aggregates information from all the other patches, making it practical to classification tasks [31].

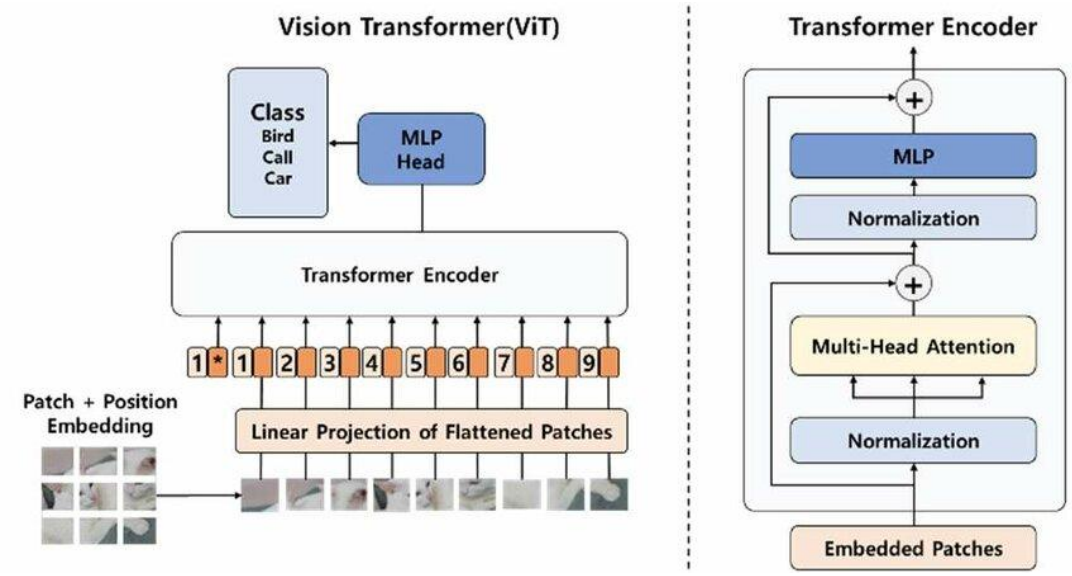

Figure 10 - Vision Transformers Architecture [32]

In Summary, transformers and their attention mechanism have revolutionized machine learning in all of its areas and applications. Particularly in computer vision, this model brings a new approach by creating relationships in the image, but it is worth noting that it requires a relatively large dataset to actually be more effective than Convolutional Neural Networks.

2.1.5 Metrics for Facial Recognition Systems

Major performance metrics of various facial recognition systems are done based on this. These metrics identify the robustness and fairness of the system in identifying a person correctly under various conditions. Both the CNN and Vision transformer models are evaluated on the same set of fundamental metrics; hence, comparisons across architectures are feasible.

2.1.5.1 Accuracy

Accuracy measures the proportion of correctly classified cases (true positives and true negatives) relative to the total number of predictions. It provides an overall sense of system performance but may not be sufficient in datasets with class imbalances.

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

The TP and TN symbolize the true positive and true negative results, and FP and FN the false positive and false negative results.

Practical Example: Out of 100 logins, 95 were correctly identified. The accuracy was 95%.

2.1.5.2 Precision

Precision, or positive predictive value, indicates how many of the predicted positive cases are correct. It is particularly useful in scenarios where false positives need to be minimized.

$$Precision = \frac{\text{Number of sucessfull results (TP)}}{\text{Number of results (TP + FP)}}$$

Practical Example: Out of 100 logins made, there were 20 times where the person was not the user who logged in. The precision would be 80%.

2.1.5.3 Recall (Sensitivity)

Recall measures the system's ability to identify all relevant cases, focusing on minimizing false negatives.

$$Recall\ = \frac{\text{Number of sucessfull results (TP)}}{\text{Number of relevant results (TP + FP)}}$$

Practical Example: Out of 100 logins made, there were 10 times that the actual user was wrongfully not given access. The Recall would be 90%.

2.1.5.4 F1-Score

The F1-Score provides a harmonic meaning of precision and recall, balancing the trade-off between false positives and false negatives.

$$F1\ = 2 \times \frac{\text{Precision}\ \times\ \text{Recall}}{\text{Precision}\ +\ \text{Recall}}$$

2.1.5.5 False Acceptance Rate (FAR) and False Rejection Rate (FRR)

These metrics evaluate biometric-specific concerns:

- **FAR**: Measures the likelihood of incorrectly accepting an unauthorized user.

$$FAR = \frac{\text{Number of False Acceptances}}{\text{Total Number of Imposters}}$$

- **FRR**: Measures the likelihood of rejecting an authorized user.

$$FRR = \frac{\text{Number of False Rejections}}{\text{Total Number of Genuine Users}}$$

2.2 Systematic Review

This section describes the systematic review to answer the defined research questions. The methodology and guidelines followed during the review concern the search strategy, inclusion and exclusion criteria, and final evaluation of the selected papers regarding how they can answer the questions.

2.2.1 Methodology

This systematic review was performed by using the PRISMA methodology. PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) is a methodology based on an item's checklist preferred for improving quality, enhancing the transparency of reporting in systematic reviews, and meta-analyses.

2.2.2 Search Terms and Data Sources

A targeted set of search terms was utilized to conduct a comprehensive search for relevant studies concerning facial recognition authentication systems and their associated security. These terms were selected with due care in order to catch the wide-ranging literature in terms of technologies, methodologies, and varying performance metrics in the context of face recognition-based authentication systems.

The keywords were selected in a manner that they would capture each critical area of interest in facial recognition systems. Table 1 details the keywords associated with each area of focus for this review.

Table 1 - Keywords by topic

Topic	Keywords
Facial Recognition Technology	"Facial recognition", "face identification", "deep learning", "machine learning", "Computer Vision"
Facial Recognition Models	"CNN", "Vision Transformer"
Facial Authentication	"biometric", "access control", "authentication"
Security and Usability	"productivity", "security", "robustness", "protection", "efficiency", "usability"

Based on the keywords identified for each theme, the final search term was developed by combining all the keywords using logical operators. This was a strategy to ensure that the review would include only those papers that could be highly relevant to the research objectives. The final search query was:

("facial recognition" OR "face identification") AND ("deep learning" OR "machine learning" OR "Computer Vision") AND ("CNN" OR "Vision Transformer") AND ("biometric" OR "access control" OR "authentication") AND ("productivity") AND ("security") AND ("robustness" OR "protection" OR "efficiency" OR "usability")

Once the search query was determined, the next important step was to select the databases in which the literature search would be conducted. For the purpose of this review, three databases were selected: IEEE Xplore, SpringerLink, and B-on. These databases were chosen because they contain vast collections of peer-reviewed articles, conference proceedings, and other scholarly works highly suitable for retrieving relevant studies on facial recognition authentication systems and their associated topics.

Given that B-On is an online library platform comprising several resources under one umbrella, the initial size of the results was gigantic in number. As such, to be assured of both the quality and relevance of studies retrieved, the findings were filtered through for it to include, in its end, only papers that were peer reviewed and published under an academic journal. The results are hereby given in the table below.

Table 2 - Databases results for the search query

DATABASE	RESULTS
IEEE Xplore	80
SpringerLink	41
B-on	192

2.2.3 Inclusion and Exclusion Criteria

Once the papers were selected, it was necessary to evaluate and determine which ones were relevant to include in the systematic review. The first step in this process involved reviewing the *titles* and *abstracts* of all the papers to perform an initial filtration. During this step, duplicate papers were also identified and removed.

Following this first selection process, the results were as follows:

Table 3 - First Papers Selection

DATABASE	PAPERS APPROVED	PAPERS NOT APPROVED
IEEE Xplore	27	53
SpringerLink	13	28
B-on	6	186

Table 3 identifies which papers were approved or discarded during the first filtration. Also, the excluded non-approved papers that must be mentioned include 9 papers excluded for being duplicates.

It is noteworthy that most of the approved papers came from IEEE Xplore and SpringerLink, given that B-On produced many more results. The main reason for such a situation comes from the essence of B-On because it is just an online library facility that holds immense volumes of papers, many of which did not fall directly into the context of this review.

Now that the relevant papers have been identified, outlining specifically, then the application of inclusion and exclusion criteria is in order.

2.2.3.1 Inclusion Criteria

- The source is peer-reviewed
- The source describes how facial recognition systems can be implemented using deep learning techniques, such as CNN, Vision Transformers, or Attention Mechanisms
- The source discusses the security, robustness, or efficiency of facial recognition authentication systems
- The source compares or measures different approaches to the implementation of facial recognition systems

2.2.3.2 Exclusion Criteria

- The source has not been published in the last 5 years.
- The source is duplicated
- The source is either a book chapter, dissertation, systematic review or thesis.
- The focus of the source is not facial recognition systems
- The source doesn't approach the topic using CNN or Vision Transformers

The papers that were deemed relevant after applying the inclusion and exclusion criteria are presented in the table below:

2.2.4 Synthesis and final papers

A comprehensive search across the selected databases yielded a total of 313 sources. After removing the 9 duplicates found, 297 papers moved to the initial screening phase. Through this phase, which used the already predefined inclusion and exclusion criteria, 251 papers were discarded. A sum of 46 papers was considered potentially relevant, and further assessment of their results and conclusions was needed. This process resulted in the exclusion of 22 papers, leaving 24 relevant papers identified and used for answering the research questions. The following diagram shows the whole process.

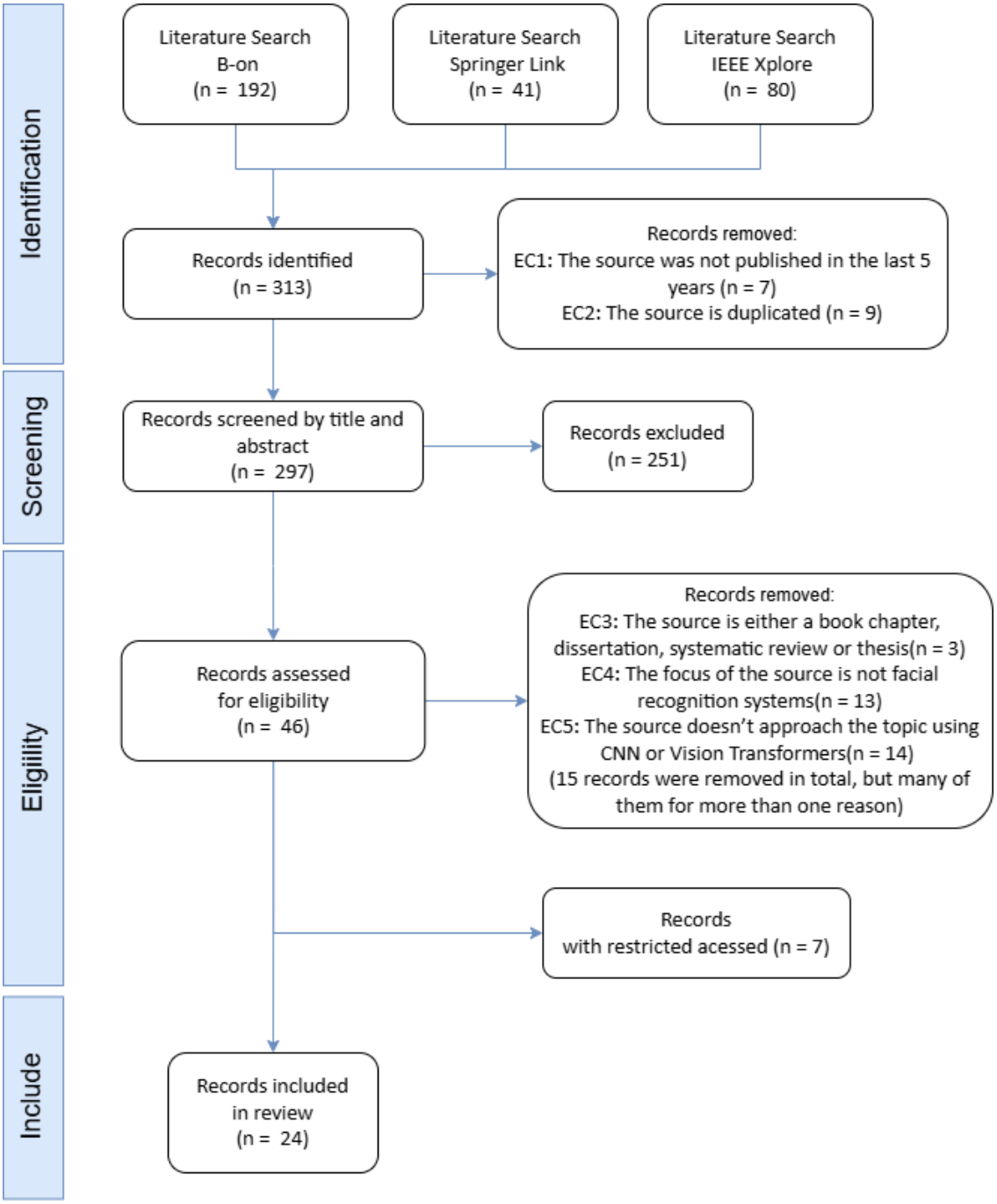

Figure 11 - PRISMA Diagram

2.2.5 Research Questions

After selecting the relevant papers for this systematic review, it is crucial to assess how these studies can help answer the research questions. The research questions guiding this review are as follows:

Q1: How can the system ensure robust security against adversarial attacks and unauthorized access, while maintaining high usability standards?

Q2: Which models, between CNN and Vision Transformers, demonstrate superior performance in terms of accuracy, efficiency, and resilience for facial recognition-based authentication?

Q3: How do different evaluation metrics (e.g., accuracy, precision, recall, F1-score) impact the performance assessment of facial recognition systems, and which metric is most indicative of real-world effectiveness?

2.2.6 Answers to Research Questions

Q1: How can the system ensure robust security against adversarial attacks and unauthorized access, while maintaining high usability standards?

Facial recognition systems have to be able to balance security and usability. For that, there is multiple strategies to achieve this balance, whether it is focusing on mitigating adversarial attacks, preventing unauthorized access, or enhancing usability.

As more and more criticalities are being taken towards the process of ensuring that the systems used to recognize faces actually perform in trustworthy ways, counteracting the adversarial attack has a prime role in the state-of-the-art anti-spoofing. These also include more advanced ones with CNNs, with various other optimized architectures, such as MobileNetV2. Those will detect and counter 2D and 3D presentation attacks ranging from the usage of printed photos to video and deepfakes. For example, anti-spoofing solutions, as emphasized in Smart Visage, through liveness detection and texture analysis, have been able to find manipulated inputs quite effectively. Moreover, deepfake attack detection techniques, as “Detecting Deep-Fake Videos from Appearance and Behavior” shows, investigate minute deviations in facial appearances and behavioral mannerisms. With deepfake technology growing rapidly, these approaches became very important. Integrating multi-encryption mechanisms ensures various options, whereby methods like homomorphic encryption may even be developed as mentioned in the concept of real-time face recognition via Computer Vision to realize industrial safety that ensures safety due to strict privations assurance for data undergoing different processing while their respective storage in appropriate data field facilities.

Limiting unauthorized access also goes a long way in tightening security. The hybrid biometric systems, where face identification and any other additional layers are combined, like passwords or 2FA, have enhanced security levels to a new level. In the case of “Enhancing Bank Locker Security”, for example, face recognition with PINs were deployed in bank locker systems and successfully reduced vulnerabilities by a large margin. Another relevant aspect is ensuring robust feature extraction, that can be done with techniques that prevent unauthorized access by handling issues like changes in illumination, facial angles, and occlusions. In addition, advanced attention mechanisms, such as those presented in “Attention-Mechanism-Based Face Feature Extraction Model for WeChat Applet on Mobile Devices”, also improve them feature extraction processes, enhancing the accuracy of authentication while reducing false positives and false negatives.

Usability has remained critical for the adoption of face recognition systems in different applications. In Improved Facial Biometric Authentication Using MobileNetV2, lightweight models like MobileNetV2 have been optimized for resource-constrained devices such as mobile phones. Such lightweight models achieve high efficiency with low latency without compromising on the security aspect and hence are very suitable for real-time applications related to banking or industrial security. Human-centered design principles ensure usability mainly in non-technical settings. For instance, Smart Retrofitting for Human Factors presents the need for system design that is actually meant for industrial retrofitting applications, ensuring ease of use in enhancing safety and access control. Besides, real-time performance capabilities, as explained in CFAS-M, ensure smooth operation without adding any latency, which is critical in dynamic environments.

Integration of such strategies follows a holistic path in balancing the dual sides of security and usability in face recognition systems. As presented in the broad work in Artificial Intelligence-Based Biometric Authentication Systems for Facial Recognition and Identification, and A Novel Technique for Facial Recognition Based on the GSO-CNN Deep Learning Algorithm, developments in hybrid CNN architecture and optimized methodologies of feature extraction have indeed made both the aspects of security and functionality strong. Ultimately, the continuing development of facial recognition technologies underlines the need to address concomitant adversarial attacks, unauthorized access, and usability concerns in a coherent and integrated manner for such technologies to prove their worth in realistic applications.

Q2: Which models, between CNN and Vision Transformers, demonstrate superior performance in terms of accuracy, efficiency, and resilience for facial recognition-based authentication?

Facial recognition-based authentication completely improved when started to use deep learning architectural implementations. Common techniques generally involve variants of CNNs or ViTs, each with different relative strengths and best suited for various issues that relate to the specific task of face recognition. Thus, the current answer combines the advantages of these two approaches

CNNs have established themselves as a reliable choice in facial recognition due to their capability of extracting spatial hierarchies in images quite effectively. As it's shown in "A Cascade CNN Model Based on Adaptive Learning Rate Thresholding for Reliable Face Recognition" that Cascade CNNs achieve an accuracy as high as 99.65% in controlled environments. Other CNNs, like LR-LGBM-CNN from the review article "Artificial Intelligence-Based Biometric Authentication Systems for Facial Recognition and Identification," perform very well with accuracy rates up to 87%, mainly on smaller datasets and constrained devices. However, in poor lighting conditions, and occlusions, the performance of CNNs degrades reach an accuracy of 98.76% under favorable conditions, that is proved in the paper "Deep Face Recognition for Biometric Authentication".

While all the CNN architectures summarized above work very well, ViTs have changed the game by presenting self-attention mechanisms that can create long-range dependencies within images. Intrinsically, this makes them very robust for cases with complex spatial relationships or of low quality. For instance," A Dual Approach with CycleGANs-based Face Reconstruction" and "ViT-based Classification" show the robustness of the ViTs against challenging conditions like low-quality images and non-frontal facial views. In addition, "Face Detection and Recognition Using OpenCV Vision Transformer" presents a comparison of ViTs against CNNs in difficult environments, where the results prove that the Vision Transformers outperform CNN. Lastly, in more recent research it is shown that Swin Transformers, one of the variants of ViTs, did an even better job than its predecessors in more complicated tasks than their traditional counterparts, which were made up of CNN architecture.

Given these advantages of each architecture, recent works are trying to focus on hybrid methods using both CNN and ViT for better exploitation of their complementary advantages.

Attention mechanisms are being integrated into CNNs as an attempt to improve feature extraction without giving up the speed. These hybrid architectures show great promise for achieving high accuracy with robustness across varied conditions and avoid many limitations that may arise with standalone CNN or ViT models.

In summary, CNNs make sure that efficiency and performances are achieved under controlled environments, whereas in the more complex open-set scenario, a proper breakthrough was warranted by ViT-based architectures, opening the way to hybrid solutions, combining the best of both paradigms for real-world challenging applications.

Q3: How do different evaluation metrics (e.g., accuracy, precision, recall, F1-score) impact the performance assessment of facial recognition systems, and which metric is most indicative of real-world effectiveness?

The best way to know if a facial recognition system is reliable is by using evaluation metrics. These metrics can make the developer understand how well the system operates under various conditions, so they can understand where it needs to be improved.

Starting with accuracy, as a straightforward and widely used metric, it measures the percentage of correct predictions out of all predictions made by the system. This metric is a basic metric that can give a general indication of the performance of the model, but it is important to understand that this metric can be insufficient mainly when using imbalanced datasets.

Next, precision and recall are particularly valuable in applications where the cost of errors varies significantly. Given that precision measures the proportion of true positives among all positive predictions, and recall evaluates the ability to correctly identify the true positives, these two metrics are essential for systems where false positives need to be almost none. A false positive would mean that the system identifies a face as the wrong person, meaning that someone would have access to an account of another person. False Negatives are not that impactful given that it would only mean the authentication failed, which doesn't necessarily mean a lack of security.

Lastly, AUC-ROC evaluates the balance between true positive and false positive rates across various thresholds, and FPR, as talked before, evaluates how often the access is granted incorrectly and is particularly important in environments with high risk.

3 Methods and Experimentation

The following chapter covers the research methodologies and materials involved in this paper, which mainly begins with analyzing and selecting datasets. Discussions on the application of machine learning methodologies are introduced, as well as the basic framework and related libraries applied to perform these techniques. The experiment procedure will then be explained in the later section, emphasizing several important ethical and security considerations developed within the entire process.

3.1 Introduction

With facial recognition systems growing in importance, modern authentication provides a secure, effective, and convenient solution. By verifying the identity of an individual automatically, it eliminates the need to manually check someone's identity and reduces human errors, hence increasing productivity. Smooth integration into the work processes of every day not only facilitates operations but is also a great convenience to users turning into the perfect solution of choice for high-security, high-efficiency environments.

Conventionally, CNNs have been in the limelight in facial recognition due to their ability to capture complex features of images coupled with well-set architecture. Models like FaceNet and ResNet have achieved incredible accuracy, benchmarking for real-world applications. However, this trend has arguably changed with the recent development in Vision Transformers that promises an arguably better approach. Unlike CNNs, which have relied on convolutional operations to deal with local features, ViT relies on self-attention mechanisms that capture a global relationship among the images. This provides a topological difference where ViTs can perform impressively in understanding detailed patterns and contextual information, thereby making them quite feasible for challenging situations in occlusions, varying lighting conditions, and extreme viewing angles.

This chapter will provide a performance comparison of pre-trained CNN and ViT models in the context of face recognition systems for authentication purposes. This work will attempt to establish if ViTs could possibly outperform CNNs in aspects of accuracy, efficiency, and

adaptability through evaluating the performances of each against diverse datasets providing an analysis of each model's strengths and weaknesses.

3.2 Methods and Tools

As outlined earlier, this chapter presents a comparison between two state-of-the-art machine learning approaches for face recognition, Convolutional Neural Networks and Vision Transformers. Both methodologies are evaluated using pre-trained models, focusing on their architectural differences regarding strong and weak points in an authentication system. In order to implement these CNNs, the FaceNet model is used and to implement the Vision Transformer architecture, a pre-trained version known as DINOv2 was used. This comparison helps us to better understand what solution could work better, so it can then be approached more smoothly.

3.2.1 FaceNet

FaceNet, introduced by Google researchers Florian Schroff, Dmitry Kalenichenko and James Philbin in 2015, revolutionized face recognition by mapping the facial images to a compact Euclidean space where the faces can be compared using the distance between them as the measure of similarity directly [33]. FaceNet maps the facial images in a 128-dimensional Euclidean embedding space so that direct comparison of facial similarities through distance metrics is possible. This innovative approach eliminates reliance on intermediate layers or additional processing. The embedding mechanism from FaceNet scales well and efficiently, making it a touchstone for many facial verification, recognition, and clustering.

Architecture-wise, FaceNet has used deep convolutional neural networks for scalability with accuracy. Major use in the architecture of the model is done with the help of two architectural styles, namely, the Zeiler and Fergus network, along with the Inception model. The Zeiler and Fergus network relies on multiple convolutional, pooling, and normalization layers to capture intricate facial features, while the Inception model leverages multi-path convolutions in order to optimize computational efficiency by reducing the number of parameters and FLOPs significantly [33]. These design elements make FaceNet adaptable for deployment on diverse platforms, from data centers to mobile devices.

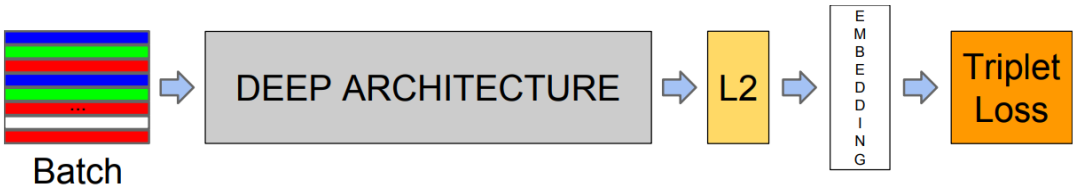

Figure 12 – FaceNet Architecture [34]

The key to its success lies at the heart of how FaceNet achieves effectiveness: the triplet loss function specifically created for embedding learning. Instead of classifying images into predefined labels, as done by most previous methods, the triplet loss optimizes the embedding space directly to reflect facial similarities. This is achieved by minimizing the Euclidean distance between embeddings of the same individual, or anchor and positive pair, and maximizing the distance between embeddings of different individuals, or anchor and negative pair [33]. The loss function is regularized through a margin parameter that enforces a measurable separation between identities. The formulation of the triplet loss can be expressed as:

$$L = \sum_i [||f(x_i^a) - f(x_i^p)||_2^2 - ||f(x_i^a) - f(x_i^n)||_2^2 + \alpha]_+$$

Where x_a, x_p, x_n represent the anchor, positive, and negative images, respectively, and α is the margin that enforces separation.

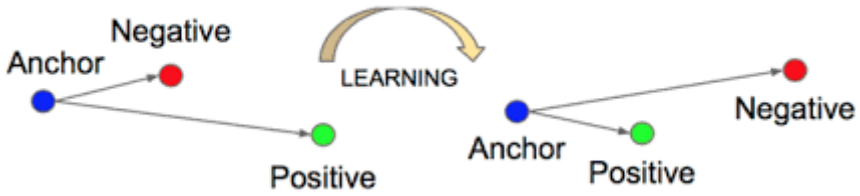

Figure 13 – Triplet Loss [35]

The performance of FaceNet is outstanding, state-of-the-art results on widely used benchmarks. For example, it achieves an accuracy of 99.63% on the Labeled Faces in the Wild dataset and 95.12% on the YouTube Faces database (YTDB) [33]. These results represent a 30% reduction in the error rates compared to previous leading models. The compactness of FaceNet's 128-dimensional embeddings also makes it ideal for large-scale applications where storage and computational resources are at a premium.

3.2.2 DINOv2

DINOv2 (Self-Distillation with No Labels v2) is a pre-trained Vision Transformer (ViT) model designed for self-supervised learning of image representations without requiring human-labeled data. DINOv2 leverages a self-distillation framework, meaning the model learns by varying its own representations over augmented views of an image.

DINOv2 is fundamentally rooted in the Vision Transformer (ViT) architecture introduced by Dosovitskiy [36]. Unlike CNNs, that processes images as a hierarchy of features, ViT represents an image as a sequence of fixed-size patches. Each patch is embedded into a vector using a linear embedding layer and added to positional encodings. These embeddings pass through multiple transformer blocks (multi-head or self-attention layers) to develop a deep understanding of complex global dependencies.

DINOv2 follows this architecture, but with the distinction that it has a self-supervised training approach. This model uses a teacher-student framework, where two versions of the same network are trained simultaneously and learn from each other with the teacher guiding the student. Basically, the teacher model is updated using an EMA (exponential moving average) of the student parameters, generating soft pseudo-labels for training the student and allowing the model to learn high-quality image features without explicit class labels.

As mentioned above, a critical element of this model is the use of Exponential Moving Average (EMA) to update the teacher model weights. This technique prevents the teacher from being an exact copy of the student, enabling stable learning dynamics.

The teacher model's weights (θ_t) are updated using an EMA of the student model's weights (θ_s), following the formula:

$$\theta_t \leftarrow \lambda\theta_t + (1 - \lambda)\theta_s$$

λ represents a momentum parameter, which is typically close to 1, in order to create a smoothed version of the student's weight and not change it dramatically.

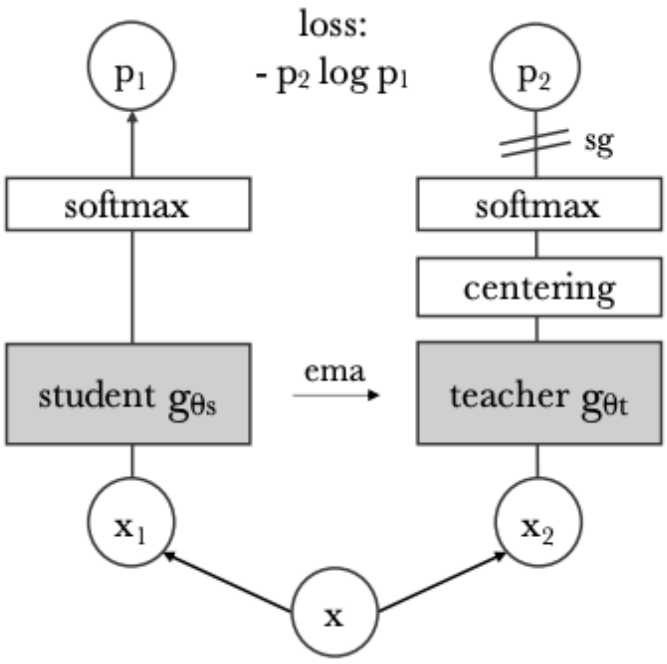

Figure 14 - DINOv2 architecture

3.3 Datasets

When talking about facial recognition systems and their primary concerns, datasets should be right at the top. Selecting the right dataset is critical to ensure good and meaningful for both self-evaluations and comparisons between models. In this section, we will discuss some of the most established datasets in this area of face recognition.

Labeled Faces in the Wild (LFW)

Labeled Faces in the Wild (LFW) is a database of face photographs designed for studying the problem of unconstrained face recognition. Created and maintained by researchers at the University of Massachusetts, this dataset contains 13,233 images of 5,749 individuals, detected and aligned using the Viola-Jones face detector. This dataset is well known in face verification tasks due to its challenging real-world variations, such as pose, lighting, and expression.

Lastly, according to the researchers, deep-funneled images produce superior results for most face verification algorithms compared to the other image types. Hence, the dataset considered for the thesis is going to be the deep-funneled version.

YouTube Faces Database (YTDB)

The YouTube Faces Database is a database of face videos designed for studying the problem of unconstrained face recognition in videos. It includes 3,425 videos of 1,595 individuals where all the data is from YouTube [37]. With a focus on challenges like temporal variation, dynamic facial expressions, and varying resolutions, YTDB provides a unique opportunity to evaluate models in dynamic, video-based scenarios, complementing static-image datasets like LFW presented above.

CelebA

CelebFaces Attributes Dataset (CelebA) is a large-scale collection of over 200,000 celebrity images, each labeled with 40 distinct facial attributes. The dataset features diverse poses and complex backgrounds, making it an excellent resource for research in dynamic and unstructured environments. CelebA is widely used for various computer vision tasks and is recognized for its flexibility and effectiveness in advancing research. [38].

MS-Celeb-1M

The Microsoft Celeb Dataset (MS-Celeb-1M, or MS1M) is one of the largest-known face recognition datasets and was released by Microsoft Research in 2016 [39]. It contains over 10 million face images from approximately 100,000 individuals collected from the Internet. It serves to advance face recognition technologies by providing an initial training dataset necessary for developing models capable of recognizing larger populations. The dataset has also widely been used for research and commercial use, especially biometric data, raising several ethical concerns [40].

VGGFace2

VGGFace2 is a large-scale face recognition dataset designed for developing and evaluating face recognition algorithms. The images were sourced from Google Image Search and include identities spanning a wide range of demographics, including diverse ethnicities, accents, professions, and age groups. All images are captured "in the wild", featuring variations in pose, emotion, lighting, and occlusion conditions, which closely resemble real-world scenarios. The dataset provides a varying number of images per identity, ranging from 87 to 843 images, with an average of 362 images per subject, offering a balanced representation for robust training and evaluation [41].

3.3.1 Dataset Decision

In this experiment, FaceNet, based on CNN and DINOv2, a model based on ViT are the two pre-trained models that are going to have their performance compared. Among the choices of datasets mentioned above, since this evaluation will focus on their accuracy, adaptability, and robustness, LFW would be most appropriate for this research.

The work will focus on training and testing using the LFW dataset since it has become one of the prominent databases in the study of unconstrained face recognition challenges. This dataset provides a considerable number of images with pose, light, emotion, and occlusion variation, which is one of the most significant factors for face recognition systems, making this dataset ideal to evaluate models for practical applications. In addition, since it has been widely used by the research community, it guarantees a standard benchmark for comparison. Finally, one of the advantages of LFW is that it can offer all these benefits while providing enough data for a useful comparison without requiring excessive computation power.

The LFW dataset also includes a set of.csv files that could be utilized for training or testing the model by providing pairs of images that could either be the same person or different individuals. The file specifies the image file names in the format name1/name1_xxxx.jpg for each image in the pair, along with a label indicating whether the pair is a match (1) or a mismatch (0). For that, each line of the file contains the name and path to the first image, the name and path to the second image and a binary label that indicates whether the pair is a match or not. Some examples of line of this dataset are in Table 4.

Table 4 - LFW test file format

NAME	IMAGE1	NAME	IMAGE2	RESULT
AJ_Cook	AJ_Cook_0001	Linda_Dano	Linda_Dano_0001	0 (Mismatched)
Ali_Naimi	Ali_Naimi_0013	Ali_Naimi	Ali_Naimi_0008	1 (Match)

4 Experimentation and Analysis of Results

4.1.1 FaceNet

To evaluate the performance of FaceNet model, it was first decided the model from PyTorch library was the one to use. Giving this the most common and trustworthy library, the decision was very straightforward.

The evaluation process starts by initializing the *InceptionResnetV1* FaceNet model, which is pre-trained on the VGGFace2 dataset provided by PyTorch, and also the MTCNN (Multi-task Cascaded Convolutional Networks), that is used to detect and align faces. This algorithm helps to improve and ensure consistency input to the model. After loading the model, the data for evaluation needs to be prepared.

To achieve the file format presented above in Table 4, the files *matchpairsDevTest.csv* and *mismatchpairsDevTest.csv* had to be concatenated and processed. These files were processed using the Pandas library, a powerful and well-known tool for data manipulation and preparation. The two files were concatenated into a single file, and it was added a column “same_person” that defined the result: 1(Match) and 0 (Mismatch).

Since the files only specify the name of the person and the image index, a function was developed to construct the full path to each image in the dataset directory, so the FaceNet Model could access the images.

```
# Load the MTCNN and Facenet model
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
mtcnn = MTCNN(keep_all=True, device=device)
facenet_model =
InceptionResnetV1(pretrained='vggface2').eval().to(device)

# Load matched and mismatched pairs
matched_pairs = pd.read_csv('../lfw-dataset/matchpairsDevTest.csv')
mismatched_pairs = pd.read_csv('../lfw-dataset/mismatchpairsDevTest.csv')
matched_pairs['name2'] = matched_pairs['name']  # name2 is the same as
name for matched pairs
mismatched_pairs.columns = ['name', 'imagenum1', 'name2', 'imagenum2']  #
Renaming the second 'name' to 'name2'
matched_pairs['same_person'] = 1
mismatched_pairs['same_person'] = 0
pairs = pd.concat([matched_pairs, mismatched_pairs])

# Base directory for the dataset
base_dir = "../lfw-dataset/lfw-deepfunneled/lfw-deepfunneled"

# Function to construct image paths
def construct_image_path(name, image_number):
    return f"{base_dir}/{name}/{name}_{str(image_number).zfill(4)}.jpg"
```

Code 1 - FaceNet dataset preparations

After the data and model were prepared, the first step of the evaluation was processing each image in MTCNN, that used various scales until it found the face or faces in the image. If more than one face is detected, the system decides to use the first one. The “unsqueeze” operation adds a batch dimension to the tensor, so it can be interpreted by the model. Then, the model generated the embeddings of both images.

```
# Function to preprocess image and extract face
def preprocess_image(image_path, target_size=(160, 160)):
    img = cv2.imread(image_path)
    if img is None:
        raise FileNotFoundError(f"Image not found: {image_path}")

    scales = [0.5, 1.0, 1.5]  # Scaling the image for face detection
    faces = None
    for scale in scales:
        img_resized = cv2.resize(img, (int(img.shape[1] * scale),
int(img.shape[0] * scale)))
        faces = mtcnn(img_resized)
        if faces is not None and len(faces) > 0:
            break  # Stop when faces are detected

    if faces is None or len(faces) == 0:
        raise ValueError(f"No faces detected in image: {image_path}")

    face = faces[0]  # Assuming the first detected face is the target
face
    return face.unsqueeze(0)  # Add batch dimension

# Function to get embedding from Facenet model
def get_embedding(model, image_path):
    processed_image = preprocess_image(image_path)
    embedding = model(processed_image)
    return embedding[0].detach().cpu().numpy()
```

Code 2 - FaceNet image processing and embedding generation

After the model generates the embeddings, they are compared to each other using the Euclidean distance, and given the threshold defined, it is decided whether a pair is classified as matched or mismatched. The system then compares this result with the actual result given in the “same_person” column.

```
# Function to compute Euclidean distance between two face embeddings
def compute_pair_distance(model, name1, img_num1, name2, img_num2):
    img1_path = construct_image_path(name1, img_num1)
    img2_path = construct_image_path(name2, img_num2)
    emb1 = get_embedding(model, img1_path)
    emb2 = get_embedding(model, img2_path)
    return euclidean(emb1, emb2)

# Function to evaluate pairs (matched or mismatched)
def evaluate_pairs(model, pairs, threshold=1.16):
    y_true = []
    y_pred = []
    y_scores = []
    missed_count = 0
    total_count = 0

    for _, row in pairs.iterrows():
        try:
            name1 = row["name"]
            img_num1 = row["imagenum1"]
            name2 = row["name2"]
            img_num2 = row["imagenum2"]

            # Compute pair distance
            distance = compute_pair_distance(model, name1, img_num1,
name2, img_num2)
            prediction = 1 if distance < threshold else 0

            # Append actual and predicted labels
            y_true.append(row['same_person'])
            y_pred.append(prediction)
            y_scores.append(distance)
            total_count += 1

        except ValueError as e:
            missed_count += 1
            continue
```

Code 3 - FaceNet Euclidean distance calculation and pair evaluation

Lastly, the system loops through all rows of the file and in the end calculates various performance metrics given the results obtained.

```
accuracy = accuracy_score(y_true, y_pred) if y_true else 0  # Avoid
division by zero
    # Calculate precision, recall, f1 score
    precision = precision_score(y_true, y_pred)
    recall = recall_score(y_true, y_pred)
    f1 = f1_score(y_true, y_pred)

    # Calculate confusion matrix
    cm = confusion_matrix(y_true, y_pred)
    tn, fp, fn, tp = cm.ravel()  # Extract confusion matrix components

    # Calculate FPR and FNR
    fpr = fp / (fp + tn) if (fp + tn) > 0 else 0
    fnr = fn / (fn + tp) if (fn + tp) > 0 else 0

    # Calculate ROC Curve and AUC
    fpr_roc, tpr_roc, thresholds_roc = roc_curve(y_true, y_scores)

    return {
        'accuracy': accuracy,
        'precision': precision,
        'recall': recall,
        'f1_score': f1,
        'fpr': fpr,
        'fnr': fnr,
        'confusion_matrix': cm,
        'missed_count': missed_count,
        'fpr_roc': fpr_roc,
        'tpr_roc': tpr_roc,
        'y_true': y_true,
        'y_scores': y_scores,
        'evaluated_count': total_count
    }
```

Code 4 - FaceNet creates metrics for evaluation

The evaluation was conducted with a threshold of 1.0, which provided 95% accuracy, along with other metrics that can be seen in the following images:

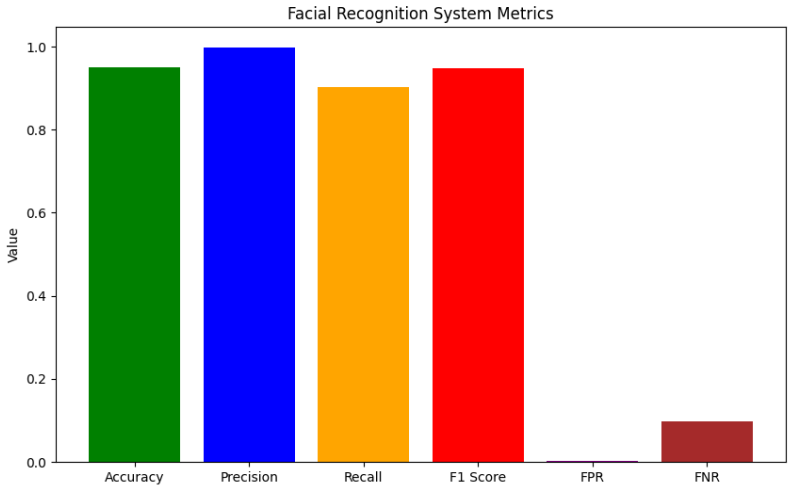

Figure 15 - FaceNet evaluation metrics for threshold 1.0

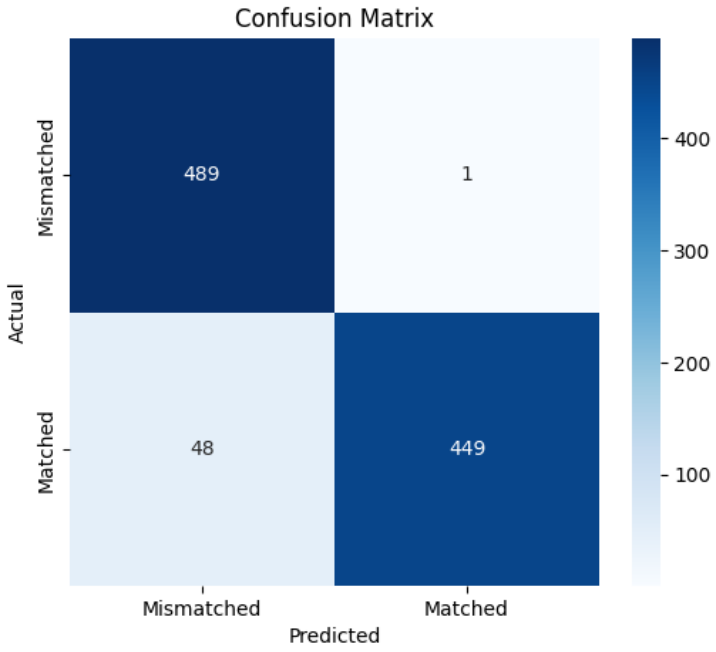

Figure 16 - FaceNet confusion matrix for threshold 1.0

The graphs displaying the metrics were generated using the sklearn library.

Figure 15 shows the model achieved significantly good results for all the main metrics with the threshold of 1.0. Beside the accuracy of 95%, that shows the model classified almost all pairs correctly, it is important to look at other metrics that are important to this type of model. First, the model predicts correctly a pair as a match 98% of the time, which is an impressive indicator

in authentication models, where security is the priority. As Figure 16 shows, out of 450 pairs that the model predicted as being a matched pair, only one of them was not actually a pair, meaning that theoretically, in 450 attempts to authenticate via facial recognition, the system only let a user log in wrongfully once. On the other hand, the False Negative Rate of 9.7% is significantly higher, meaning that almost 10% of the time the model was not able to identify an actual match pair of images. Either this could require a better refining of the model or better preparation of the data sent to the model.

4.1.2 DINOv2

To evaluate the performance of the Vision Transformer (ViT) model, it was decided to use the DINOv2 model from Meta, as it is one of the most recent and robust self-supervised learning models available in the PyTorch library.

The evaluation process begins by loading the DINOv2 model, which is pre-trained on a large dataset, and the corresponding image processor, both provided by the Hugging Face Transformers library. Unlike FaceNet, which relies on the MTCNN algorithm for face alignment, ViT processes entire images without explicit face detection, what can cause variability in feature extraction depending on image quality and framing.

After loading the model, the dataset must be structured in a suitable format. Like the FaceNet evaluation, the files matchpairsDevTest.csv and mismatchpairsDevTest.csv were concatenated into a single dataset and a new column, “same_person,” was added to distinguish between matched (1) and mismatched (0) pairs.

```
# Load the DINOv2 model
processor = AutoImageProcessor.from_pretrained("facebook/dinov2-base")
dino_model = AutoModel.from_pretrained("facebook/dinov2-base")
dino_model.eval()

# Load matched and mismatched pairs
matched_pairs = pd.read_csv('../lfw-dataset/matchpairsDevTest.csv')
mismatched_pairs = pd.read_csv('../lfw-dataset/mismatchpairsDevTest.csv')

# Adjust dataframe structure
matched_pairs['name2'] = matched_pairs['name']  # name2 is the same as
name for matched pairs
mismatched_pairs.columns = ['name', 'imagenum1', 'name2', 'imagenum2']
matched_pairs['same_person'] = 1
mismatched_pairs['same_person'] = 0
pairs = pd.concat([matched_pairs, mismatched_pairs])

# Base directory for the dataset
base_dir = "../lfw-dataset/lfw-deepfunneled/lfw-deepfunneled"

# Function to construct image paths
def construct_image_path(name, image_number):
    return f"{base_dir}/{name}/{name}_{str(image_number).zfill(4)}.jpg"
```

Code 5 - DINOv2 data and model preparation

Once the dataset was prepared, the next step involved image preprocessing. The images were resized to 224x224 pixels to match the model's input requirements, and then some additional transformations were also done. These transformations include horizontal flipping and color jittering that were applied to the images to introduce variability in the dataset and consequentially, improve the model's robustness against light or positions changes.

Lastly, to help process the image, a pooling operation was used. Between the max or mean pooling, it was decided to go with mean polling, because it creates a more stable and representative embedding, reducing local noises present in the image.

```
transform = transforms.Compose([
    transforms.Resize((224, 224)),
    transforms.RandomHorizontalFlip(p=0.5),  # Augmentation for
robustness
    transforms.ColorJitter(brightness=0.2, contrast=0.2, saturation=0.2,
hue=0.05),
    transforms.ToTensor()
])

# Function to preprocess an image
def preprocess_image(image_path):
    image = Image.open(image_path).convert("RGB")
    image = transform(image)  # Apply new augmentations
    inputs = processor(images=image, return_tensors="pt")
    return inputs['pixel_values']

# Function to extract embeddings from DINOv2
def get_embedding(model, image_path):
    img_tensor = preprocess_image(image_path)
    with torch.no_grad():
        outputs = model(img_tensor)
        features = outputs.last_hidden_state.mean(dim=1)  # Mean pooling
of all patch embeddings
    return features.flatten().cpu().numpy()
```

Code 6 - DINOv2 image processing and transformation

Once the embeddings are generated, it is needed to compare the similarity between the images. For that, the cosine distance was used. Just like CNN, if the distance was inferior to the defined threshold, the pair was classified as a match, otherwise it would be classified as a mismatch.

```
# Normalize embedding for better distance calculation
def normalize_embedding(embedding):
    return embedding / np.linalg.norm(embedding)

def compute_pair_distance(model, name1, img_num1, name2, img_num2):
    img1_path = construct_image_path(name1, img_num1)
    img2_path = construct_image_path(name2, img_num2)
    emb1 = normalize_embedding(get_embedding(model, img1_path))
    emb2 = normalize_embedding(get_embedding(model, img2_path))
    return cosine(emb1, emb2)  # cosine distance
```

Code 7 - DINOv2 embedding comparison

Finally, the system computed performance metrics based on the classification results.

```
    # Compute evaluation metrics
        accuracy = accuracy_score(y_true, y_pred)
        precision = precision_score(y_true, y_pred)
        recall = recall_score(y_true, y_pred)
        f1 = f1_score(y_true, y_pred)
        cm = confusion_matrix(y_true, y_pred)
        tn, fp, fn, tp = cm.ravel()

        # False Positive & False Negative Rates
        fpr = fp / (fp + tn) if (fp + tn) > 0 else 0
        fnr = fn / (fn + tp) if (fn + tp) > 0 else 0

        # Compute ROC Curve
        fpr_roc, tpr_roc, thresholds_roc = roc_curve(y_true, y_scores)

        return {
            'accuracy': accuracy,
            'precision': precision,
            'recall': recall,
            'f1_score': f1,
            'fpr': fpr,
            'fnr': fnr,
            'confusion_matrix': cm,
            'missed_count': missed_count,
            'fpr_roc': fpr_roc,
            'tpr_roc': tpr_roc,
            'y_true': y_true,
            'y_scores': y_scores,
            'evaluated_count': total_count
        }
```

Code 8 - DINOv2 evaluation metrics generation

The DINOv2 model achieved an accuracy of 81.1%, with a precision of 87.29% and a recall of 72.8% for a threshold of 0.4. These metrics along with the confusion matrix can be seen below.

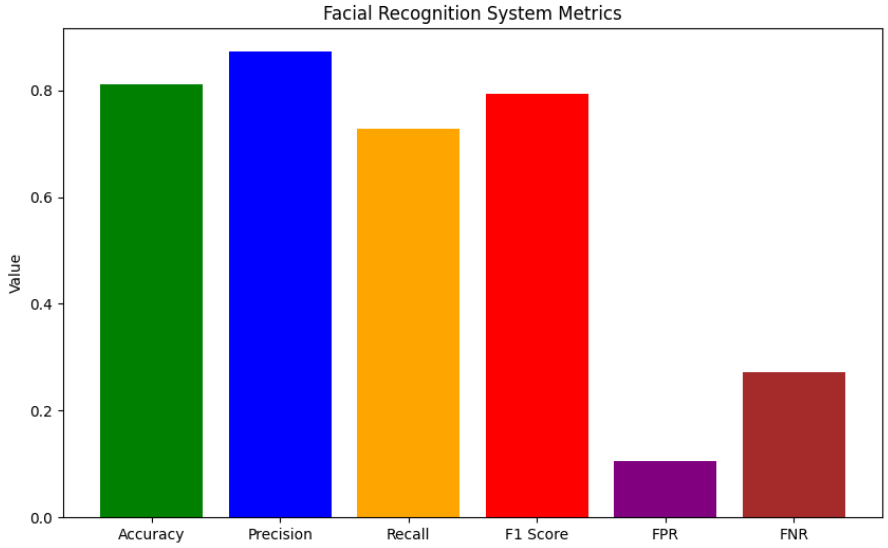

Figure 17 - DINOv2 evaluation metrics

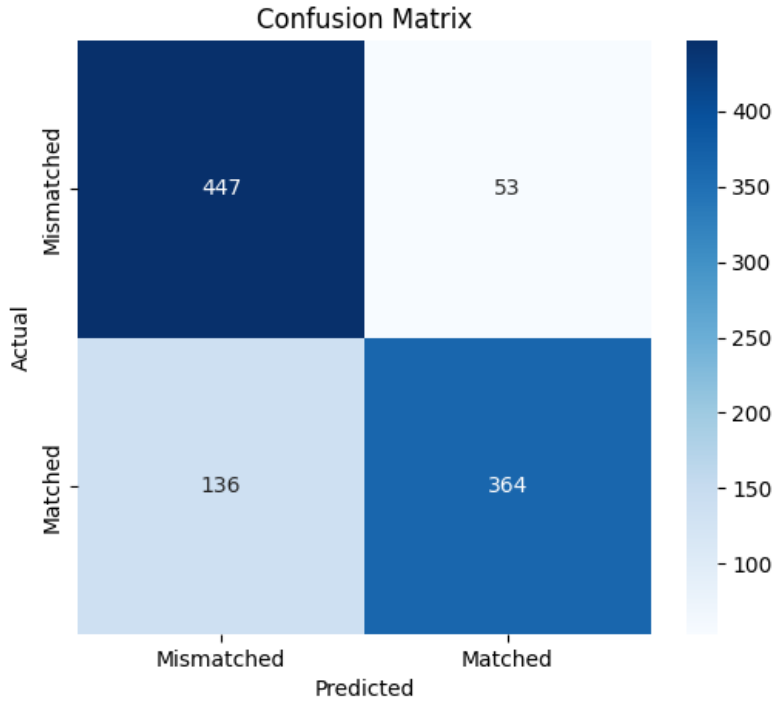

Figure 18 - DINOv2 confusion Matrix

Figure 17 indicates that while this vision transformer-based model achieved reasonable performance in face recognition, it did not surpass the performance of FaceNet. In terms of accuracy, FaceNet performed significantly better, having 95% compared with the 81.1% of

DINOv2 model. Beside accuracy, for the other metrics important to facial recognition, it can be seen that in terms of precision, DINOv2 was very good, with 87.3%. The model was also worse than FaceNet in performance, having 27% of False Negative Rate meaning that almost a third of the times the model was not able to identify an actual match pair of images, and in security, where out of 417 times the system considered the pair a match, 53 times this was not correct, meaning that 7% of the times this system would have security breaches.

In conclusion, the vision Transformer based model was clearly inferior to the CNN based model, having a worse performance on every aspect. One possible reason for this discrepancy is that FaceNet is specifically designed for face recognition, employing CNN-based architecture trained on the VGGFace2 dataset, which contains a vast number of facial images, whereas DINOv2, although highly effective in general visual representation learning, is not explicitly designed to this task. The perfect example is the use of MTCNN for face alignment in FaceNet that ensures consistent input quality, reducing variability in facial positioning and orientation. Additionally, knowing the highly effectiveness of Vision Transformers, fine-tuning this model on a facial dataset could significantly improve its performance and make it much more competitive compared to FaceNet and CNN models.

4.2 Data Protection, Security and Ethics

Facial recognition systems for authentication purposes raise significant ethical and data privacy concerns. Given that this model deals with biometric data, it is naturally critical to keep this sensitive information safe and safeguard the integrity of the entire recognition infrastructure against potential misuse or intrusion.

About data protection, the model makes sure to follow the General Data Protection Regulation (GDPR), which governs the process and storage of personal data within the European Union. The dataset used in this study was the Labeled Faces in the Wild (LFW) dataset, which is publicly available, licensed, and does not contain any sensitive identifiers apart from names and public images. As such, no personal or confidential data was collected or processed. The dataset is only for research purposes, ensuring ethical and legal standards.

Now, looking from a security perspective, it is equally important to look for potential vulnerabilities of facial recognition systems. Intrusions or malicious attacks of the recognition

model could allow unauthorized users to gain access to protected areas or data. Therefore, the implementation of some security measures is fundamental to maintaining system integrity. Some of these measures could be encrypted communication between client and server, secure storage of model parameters or multi-factor access controls.

Regarding ethics and fairness, facial recognition technologies have faced a lot of criticism due to privacy intrusions, potential misuse, and mainly algorithmic bias, where models tend to perform disproportionately worse on underrepresented demographic groups, such as individuals with darker skin tones or specific ethnic backgrounds. To mitigate this issue, this study relied on the LFW dataset, which provides a broad distribution of images across gender, ethnicity, and age. Nevertheless, continuous evaluation of fairness metrics is necessary to ensure equitable performance.

In conclusion, several measures were implemented to ensure that all ethical, legal, and technical dimensions of security were considered. Protecting both the biometric data and the integrity of the system itself contributes to the reliability, credibility, and social acceptance of the proposed solution.

4.3 Conclusions

In chapter 3 and 4, the goal was to present the study, analyzing different approaches, evaluate datasets, tools and ultimately, facilitate the implementation to be done.

Chapter 3 presented a theoretical explanation of both approaches compared, along with the tools and frameworks needed for their development. Also, various datasets were compared and evaluated, understanding what each of them had to offer to a model like the one being developed.

The implementation of a pre-trained model and analysis of its metrics is critical to understanding how the data should be prepared and what algorithm should be used to calculate the embeddings generated by the model. Also, the understanding and analysis of the performance metrics gives a good overview of what should be the main concerns performance-wise.

Ultimately, this pre-trained model with such good results serves as a benchmark to validate and evaluate any model trained and tested from scratch. It sets a performance standard and offers valuable insights into optimizing the system's architecture.

5 Architecture

In this chapter, the best architecture for the implementation will be explored and analyzed in-depth, in a way to maximize efficiency and user experience.

This system supports real-time facial recognition authentication tailored for integration into the Enterprise Resource Planning (ERP) and the Manufacturing Execution System (MES) systems of Sistrade Software Consulting. At the factory level, where frequent login by factory-floor operators can slow down production workflows, enabling the possibility to authenticate faster and easier, would naturally boost the levels of efficiency and productivity.

Designed for speed, accuracy, and user privacy, the solution employs a client-server model to separate lightweight user-side operations from the more intensive processing tasks handled by the backend. At the front-end, an .asp page embedded into the ERP login interface manages the image capture process. Using a JavaScript module powered by the FaceMesh library, facial landmarks are detected in real time, guiding the user through various poses. This approach ensures that the images captured are of high quality and suitable for recognition, while keeping the experience fast and seamless. Once the images are captured, they are securely transmitted to a Python-based backend API using HTTPS. Each request includes session tokens and is protected against replay attacks through timestamping and rate-limiting.

On the server, MTCNN is used to detect and align faces. The aligned image is then processed using the FaceNet model, which generates 128-dimensional embedding. This embedding is compared to the reference embedding stored for the user, with authentication success determined by a similarity threshold. The backend sends a simple Boolean response back to the .asp page, which integrates with the ERP's authentication logic. A successful match initiates the user session, bypassing traditional credential checks. If the authentication fails, the system defaults to standard login mechanisms such as passwords or OTPs. To maintain system responsiveness, FaceNet is preloaded into memory and image batches are processed asynchronously. For scalability, the backend supports containerization with Docker and load balancing, ensuring low latency even under high demand. In most scenarios, the entire authentication process is completed in under one second. Security and privacy are central to this architecture.

All embeddings are encrypted with AES-256 and stored only with explicit user consent. The system complies fully with GDPR, allowing users to opt out and delete their facial data at any time, aligning with Articles 5 and 17. Communication is protected via TLS, and all data flows are carefully logged and monitored.

Figures 27 and 28 illustrate the sequence of user authentication and registration, respectively. Together, architecture delivers a robust and secure facial recognition system that enhances the security and usability of ERP access without compromising user autonomy or data protection.

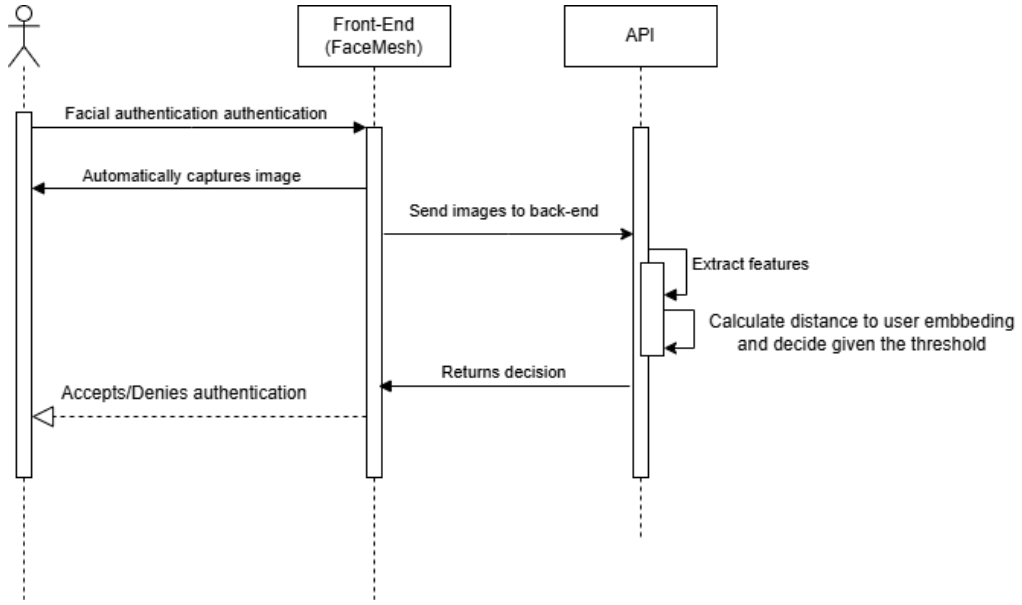

Figure 19 – Sequence Diagram of user facial authentication

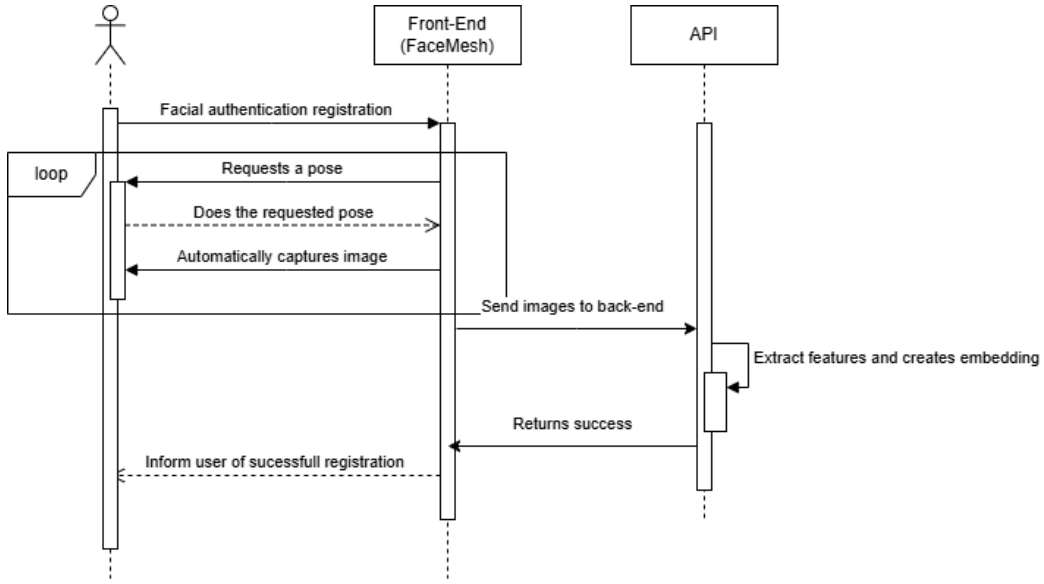

Figure 20 - Sequence Diagram of user facial registration

6 Implementation and Results

In this chapter, we present the authentication model based on the CNN approach, given it showed a better performance in comparison that was previously done.

The authentication system is centered around the creation and usage of a unique facial embedding per user. This embedding is generated during user registration and is subsequently used for verifying the user during authentication attempts. The core idea is to compare a newly captured image with the previously stored embedding to determine whether the person attempting to log in is indeed the registered user.

6.1 User Registration

During the registration process, the system will capture automatically a set of images of the user in different facial poses. First, the system takes multiple photos of the user for 10 seconds, 1 per second, and then it asks the user for some specific poses so it can construct a more robust embedding. The front-end application is responsible for this capture process, using FaceMesh, a library capable of real-time facial geometry tracking. The system uses the nose of the user to check the current position so it can automatically detect the user pose and take photos. Once the images are captured, they are transmitted to the back-end via an API.

In the back-end, each image is processed individually by the CNN model to extract facial features and generate embeddings. These embeddings are then averaged to form a consolidated and unique digital signature for the user. This average number of multiple images and poses is crucial to not only ensure a good embedding but also to remove possible noise.

The final embedding is stored securely in a folder on the server. An important note is that this authentication method is only activated with the user's explicit consent. In compliance with the General Data Protection Regulation (GDPR) in Europe, if the user decides to disable this feature, the stored embedding file must be permanently deleted from the server.

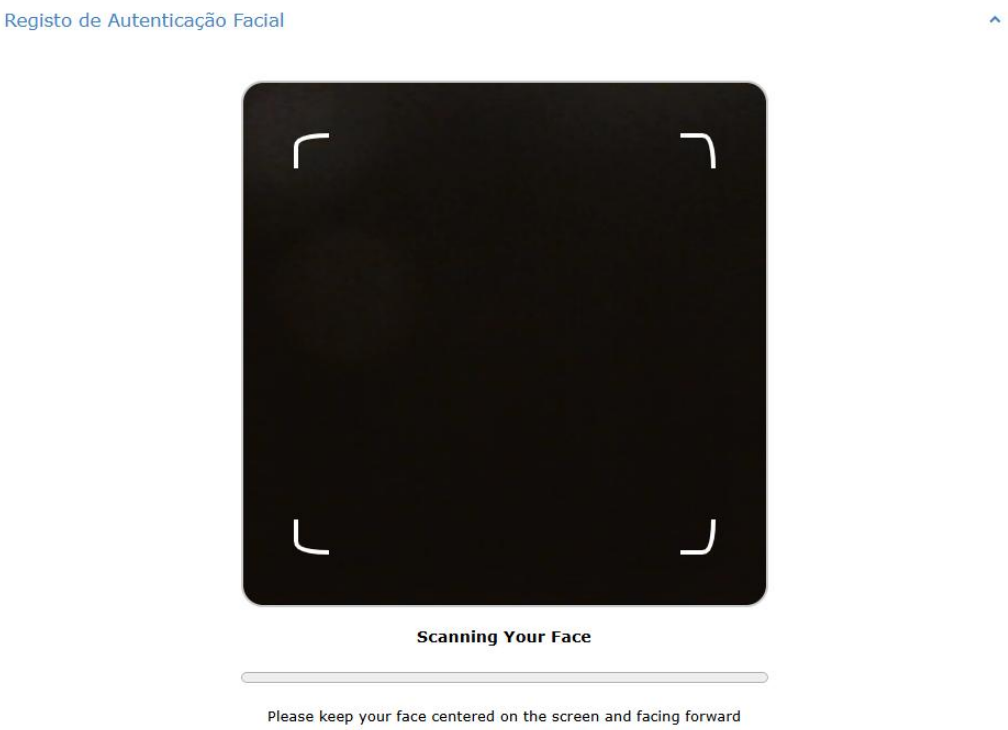

Figure 21 – First phase of user images capture

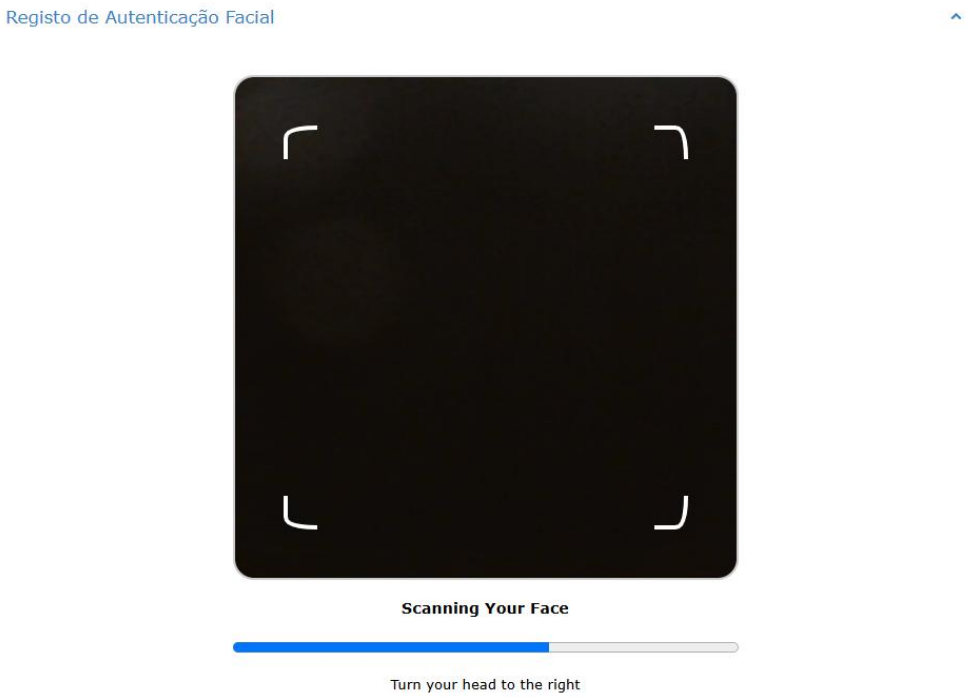

Figure 22 - Second phase of user images capture

6.2 Authentication

When a user attempts to authenticate, the front-end captures a small number of facial images in real time. These new images are sent to the back-end, where the same CNN model processes them to extract fresh embeddings.

These new embeddings are then compared to the stored embedding generated during registration and if the calculated distance between the new embeddings and the stored embedding falls within a predefined threshold, the system considers the user to be successfully authenticated. Naturally, the smaller the threshold, the less margin there is. Although a smaller threshold can look better and safer, it can also cause wrong results if the user is a little different from how it was when the registration was made (different hair, glasses, etc...).

Additionally, the back-end includes mechanisms to detect spoofing attempts, such as photos or videos being used in place of a real face, which will be exploited deeper in the next chapter.

6.3 Anti-Spoofing Strategies

Since the system under development is intended to be used as an authentication method, its ability to prevent spoofing and resist fraudulent access attempts become clinical. Security is not just a feature, but it is the foundation of trust in biometric systems. In the context of facial recognition, spoofing attacks frequently involve the use of static photographs, prerecorded videos, or even deepfakes to impersonate legitimate users, and so ensuring the system can reliably detect and reject such attempts is essential for its real-world viability.

6.3.1 Client-Side vs Server Side

The first important aspect is to determine if this verification should be done on the server side or client side by understanding what each of them has to offer.

Client-Side Detection: Running anti-spoofing on the client, which is the browser in this specific case, can reduce server load, latency and detect obvious fake inputs locally, saving bandwidth and server computation [42]. And even though this approach gives instant feedback to the user and can operate regardless of the network velocity, the results are not 100% trustworthy.

Relying solely on client-side liveness checks therefore poses security risks [42], given an attacker could modify the JavaScript to skip the liveness test or feed the system pre-recorded images.

Server-Side Detection: Naturally, performing this verification on the server side is far more secure because the image analysis happens in a "controlled environment". Apart from that, the server can use more powerful hardware (GPU and CPU) to run advanced and heavy models. On the negative side, the detection takes more time and the latency is bigger.

In conclusion, and what many systems adopt, is a hybrid approach, where a lightweight check is done on the client side, so it can filter out the most obvious spoofing attacks, followed by a heavier and more complex verification on the server side [42].

6.3.2 State-of-the-Art Models and Techniques for Real-Time Anti-Spoofing

This chapter focuses on evaluating the effectiveness of both deep learning architectures Convolutional Neural Networks and Vision Transformers for face anti-spoofing. Each approach has its strengths and weaknesses, which influence their suitability depending on the deployment context and the nature of the spoofing threats being addressed**.**

Given CNNs are highly efficient at capturing local spatial features, it allows fast inference and robust performance, especially in real-time scenarios. Also, they perform well on smaller datasets, making them accessible for practical applications where large-scale annotated spoofing data may not be available [43] [44]. The most common CNN-based models for anti-spoofing are ResNet50, MobileNetV3, and EfficientNet-B4, who have consistently demonstrated strong results in anti-spoofing tasks.

However, CNNs do have limitations. Their localized focus may miss more subtle, globally distributed artifacts, like those introduced by deepfake generation techniques [45]. Additionally, they can be susceptible to adversarial attacks, particularly those crafted to mimic facial patterns with high fidelity [46].

On the other hand, Vision Transformers represent a newer and increasingly promising direction. Unlike CNNs, ViTs process an entire image at once, allowing them to capture broader contextual information [47]. This global perspective makes them particularly well-suited to detect smooth

inconsistencies and pixel-level manipulations present in deep-fake images or 3D mask attacks. Also, they tend to generalize well without requiring as much data augmentation as CNNs [48].

Nevertheless, the strengths of ViTs also come with trade-offs. They typically require larger training datasets to perform effectively [49], and their computational demands are significantly higher, making them less ideal for real-time applications or devices with limited hardware capacity. Inference times are slower, and the need for powerful GPUs can be a barrier to implementation in mobile or embedded systems [50].

To better understand the differences between these two approaches, Table 5 demonstrates the biggest differences between some CNN and ViT models. The Accuracy of the models were determined based on CelebA-Spoof database [51].

Table 5 - Anti-spoofing model comparison

MODEL	ACCURACY	SPEED	BEST FOR
ResNet50 (CNN)	95%	Fast	General Anti-Spoofing
MobileNetV2 (CNN)	93%	Very Fast	General Anti-Spoofing
EfficientNet-B4 (CNN)	96%	Fast	High Accuracy, Low Compute
ViT-Base Transformer (ViT)	97%	Slow	Deepfake Detection
Swin Transformer (ViT)	98%	Slow	3D Mask Spoofing

In summary, the choice between CNN and ViT for anti-spoofing should be guided by the specific context in which the system will operate. Given the context of what's being developed, where the number of data is limited, and is being implemented in a real-time application, it makes more sense to go with a CNN based model.

6.3.3 Datasets and Fine-Tuning to Improve Performance

For the success and accuracy of anti-spoofing model, a good and robust dataset is clinical. Face liveness detection can fail if the model only "memorizes" specific attack conditions, so having a diverse dataset and fine-tuning it to the target is essential.

Having said this, there are several public datasets for anti-spoofing research. Some of the better ones are:

MSU MFSD: From MSU, including printed photos and replay attacks using mobile devices.

CelebA-Spoof: A dataset with over 625K images covering many identities, poses, and spoof types, like prints, cutouts, screens, etc.

OULU-NPU: A dataset of video clips with print and video replay attacks under different lighting conditions.

Given that each dataset has its own bias, models trained on one may not generalize to others. A common practice is cross-dataset evaluation, which consists of training the model on two of the above and testing on the third to see how well the model handles unseen spoofs [52].

On the other hand, fine-tuning a pre-trained model on specific data can significantly boost performance. Instead of training from scratch, starting with a model pre-trained in a large dataset and then fine-tuning it on a smaller dataset that reflects the environment of the project normally results in better performance and less effort [53].

6.3.4 Development of an anti-spoofing model

In line with the previous chapters, Convolutional Neural Networks (CNNs) were selected for the implementation of the anti-spoofing component given their efficiency and suitability for real-time applications. MobileNetV3-Small was chosen for the backbone as it balances accuracy with computational efficiency perfectly.

To establish baseline, the MobileNetV3-Small model was first tested in its pre-trained state on ImageNet, with only the final classification head adjusted to output two classes: real and spoof. For that, it was developed an evaluation script able to evaluate the model.

```
    mean=[0.485,0.456,0.406]; std=[0.229,0.224,0.225]
    tf = transforms.Compose([
        transforms.Resize(int(args.img_size*1.1)),
        transforms.CenterCrop(args.img_size),
        transforms.ToTensor(),
        transforms.Normalize(mean,std),
    ])

    ds = datasets.ImageFolder(args.data, transform=tf)
    # Remap to 0=spoof, 1=real if names present
    class_to_idx = ds.class_to_idx
    if 'real' in class_to_idx and 'spoof' in class_to_idx:
        real_idx  = class_to_idx['real']
        spoof_idx = class_to_idx['spoof']
        def remap(samples):
            return [(p, 1 if y==real_idx else 0) for p,y in samples]
        ds.samples = remap(ds.samples)

    loader = torch.utils.data.DataLoader(ds, batch_size=1, shuffle=False,
num_workers=2)

    y_true, y_prob = [], []

    ckpt = torch.load(args.model, map_location='cpu')
    model = models.mobilenet_v3_small(weights=None)
    in_feats = model.classifier[-1].in_features
    model.classifier[-1] = torch.nn.Linear(in_feats, 2)
    model.load_state_dict(ckpt['model_state'], strict=True)
    model.eval()
    for imgs, labels in loader:
        with torch.no_grad():
            logits = model(imgs)
            prob_real = torch.softmax(logits, dim=1)[:,1].numpy()
        y_true.extend(labels.numpy().tolist())
        y_prob .extend(prob_real.tolist())
```

Code 9 - Evaluation of MobileNetV3-Small

The evaluation routine begins by reproducing the exact preprocessing used during training so that the model is assessed under identical conditions. Images are resized slightly larger than the target size and then center-cropped to the final resolution. They are converted to tensors and normalized with the standard ImageNet channel statistics. The dataset is then loaded with a folder-based convention where each subdirectory name defines a class. After, a remapping is

done to eliminate common sources of failure in biometric classifiers, where metrics appear poor simply because the predicted index is interpreted with the wrong semantic class. Then, the data loader iterates with batch size 1 and with no shuffle, and finally, the model is reconstructed with a 2-unit classification head. Once the checkpoint is loaded and the model set to evaluation mode, each image is passed through the network to obtain the probability of belonging to the "real" class. These outputs are then collected alongside the ground-truth labels and later used to compute standard biometric performance measures.

As expected, this version performed poorly. The model had never been exposed to spoofing-specific features and so it generalized poorly.

```
AUC=0.2267  ACC=0.3750  APCER=0.5667  BPCER=0.8000  EER=0.6583 @ thr=0.440
Confusion matrix [rows: spoof, real] -> [[tn fp],[fn tp]]:
[[26 34]
 [16  4]]
```

Figure 23 - MobileNetV3 evaluation metrics

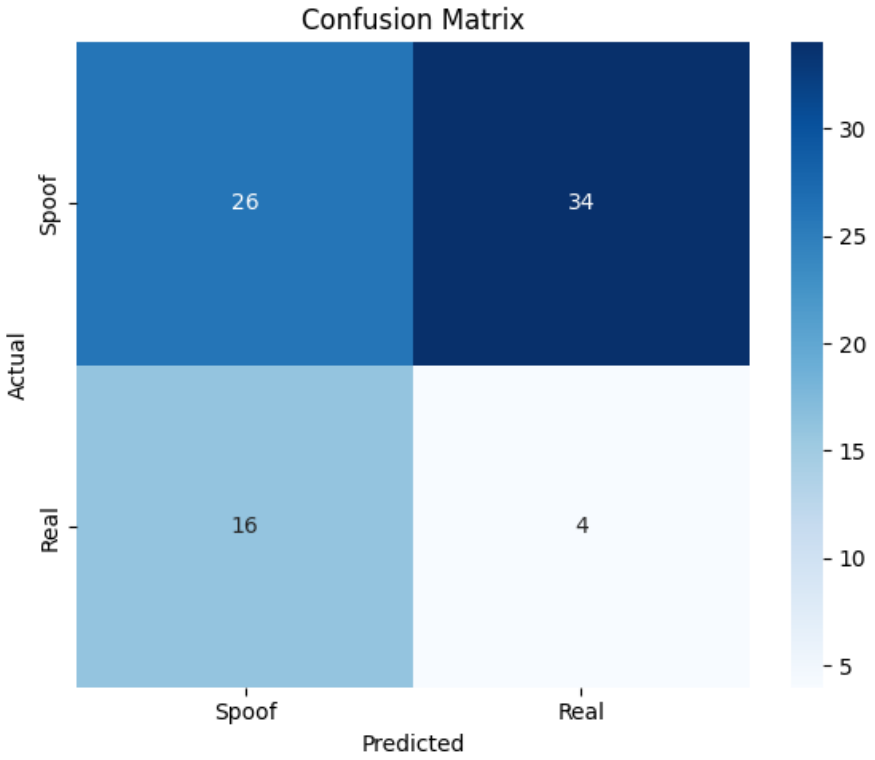

Figure 24 - MobileNetV3 confusion matrix

The initial evaluation yielded very low performance, with an accuracy of 37.5% (Figure 23). The corresponding confusion matrix (Figure 24) confirmed the problem: the classifier struggled to distinguish spoof attempts from real attempts, misclassifying a large proportion of real samples as spoof and vice versa.

To improve performance, the model was fine-tuned on a task-specific dataset. The fine-tuning process retained the ImageNet-pretrained convolutional backbone but updated its weights using labeled real and spoof face images. Using data augmentation strategies to simulate common spoofing conditions like brightness and contrast variation or color jitter, the model was now ready to identify different lighting and capture conditions. Additionally, class imbalance was reduced using a weighted sampler.

```
# train_antispoof.py (excerpt)
train_tf, val_tf = build_transforms(args.img_size, imagenet_norm=True,
cropper=cropper)

train_set = datasets.ImageFolder(os.path.join(args.data, 'train'), trans-
form=train_tf)
val_set = datasets.ImageFolder(os.path.join(args.data, 'val'), trans-
form=val_tf)

# Remap: 0=spoof, 1=real
class_to_idx = train_set.class_to_idx
if 'real' in class_to_idx and 'spoof' in class_to_idx:
    real_idx, spoof_idx = class_to_idx['real'], class_to_idx['spoof']
    def remap(samples): return [(p, 1 if y==real_idx else 0) for p,y in
samples]
    train_set.samples, val_set.samples = remap(train_set.samples),
remap(val_set.samples)

# DataLoaders with weighted sampling
train_loader = DataLoader(train_set, batch_size=args.batch_size,
   sampler=WeightedRandomSampler(compute_class_weights(train_set),
   num_samples=len(train_set), replacement=True))
val_loader = DataLoader(val_set, batch_size=args.batch_size, shuf-
fle=False)

# Model: MobileNetV3-Small pre-trained on ImageNet
model = models.mobilenet_v3_small(weights=models.Mo-
bileNet_V3_Small_Weights.IMAGENET1K_V1)
model.classifier[-1] = nn.Linear(model.classifier[-1].in_features, 2)

# Train / validate
optimizer = optim.AdamW(model.parameters(), lr=args.lr, weight_de-
cay=args.wd)
criterion = nn.CrossEntropyLoss()
for epoch in range(args.epochs):
    train_one_epoch(model, train_loader, optimizer, criterion, device)
    validate(model, val_loader, criterion, device)
```

Code 10 - Fine tune of MobileNetV3-Small

The model starts with ImageNet pre-trained weights, and the final fully connected layer is replaced with a two-class head. Training uses AdamW optimization with cross-entropy loss, and both training and validation metrics are tracked per epoch. Whenever a new best validation AUC is achieved, the model checkpoint is saved and additionally exported to the ONNX format, which can be directly used beyond the training environment.

The fine-tuning was conducted for 12 epochs and a learning rate of 3e-4. During training, the model showed rapid convergence, reaching near-perfect training accuracy after a few epochs (Figure 25).

Then the model was submitted to the same validation as the last model. The results demonstrated substantial improvement compared to the baseline (Figure 26). The model achieved an accuracy of 73.75% and a much better confusion matrix (Figure 27).

```
[01] tr_loss=0.6681 acc=0.524 | val_loss=0.7933 acc=0.325 auc=0.458 APCER=0.800 BPCER=0.300 EER=0.500 thr=0.572 (14.3s)
[02] tr_loss=0.5954 acc=0.698 | val_loss=0.7725 acc=0.362 auc=0.524 APCER=0.750 BPCER=0.300 EER=0.508 thr=0.564 (13.0s)
[03] tr_loss=0.4446 acc=0.857 | val_loss=0.7806 acc=0.350 auc=0.590 APCER=0.800 BPCER=0.200 EER=0.450 thr=0.588 (14.9s)
[04] tr_loss=0.3561 acc=0.905 | val_loss=0.7684 acc=0.388 auc=0.659 APCER=0.783 BPCER=0.100 EER=0.400 thr=0.602 (14.9s)
[05] tr_loss=0.2803 acc=0.952 | val_loss=0.7469 acc=0.425 auc=0.716 APCER=0.733 BPCER=0.100 EER=0.325 thr=0.617 (13.0s)
[06] tr_loss=0.2289 acc=0.968 | val_loss=0.7416 acc=0.438 auc=0.758 APCER=0.733 BPCER=0.050 EER=0.300 thr=0.617 (15.1s)
[07] tr_loss=0.1716 acc=1.000 | val_loss=0.7538 acc=0.412 auc=0.789 APCER=0.767 BPCER=0.050 EER=0.317 thr=0.632 (13.8s)
[08] tr_loss=0.1432 acc=0.968 | val_loss=0.7644 acc=0.412 auc=0.817 APCER=0.783 BPCER=0.000 EER=0.258 thr=0.659 (14.6s)
[09] tr_loss=0.1724 acc=0.952 | val_loss=0.7503 acc=0.438 auc=0.839 APCER=0.750 BPCER=0.000 EER=0.233 thr=0.678 (15.1s)
[10] tr_loss=0.1706 acc=0.952 | val_loss=0.6996 acc=0.512 auc=0.860 APCER=0.650 BPCER=0.000 EER=0.267 thr=0.641 (13.1s)
[11] tr_loss=0.0802 acc=1.000 | val_loss=0.6549 acc=0.562 auc=0.877 APCER=0.583 BPCER=0.000 EER=0.267 thr=0.621 (15.0s)
[12] tr_loss=0.0649 acc=1.000 | val_loss=0.6133 acc=0.637 auc=0.887 APCER=0.467 BPCER=0.050 EER=0.267 thr=0.594 (14.1s)
Training completed.
Best model (by AUC): ./runs/antispoof_mnv3\best_antispoof_mnv3.pt
ONNX exported alongside it.
```

Figure 25 - Training epochs of improved model

```
AUC=0.8875  ACC=0.7375  APCER=0.2667  BPCER=0.2500  EER=0.2667 @ thr=0.594
Confusion matrix [rows: spoof, real] -> [[tn fp],[fn tp]]:
[[44 16]
 [ 5 15]]
```

Figure 26 - Evaluation metrics of improved model

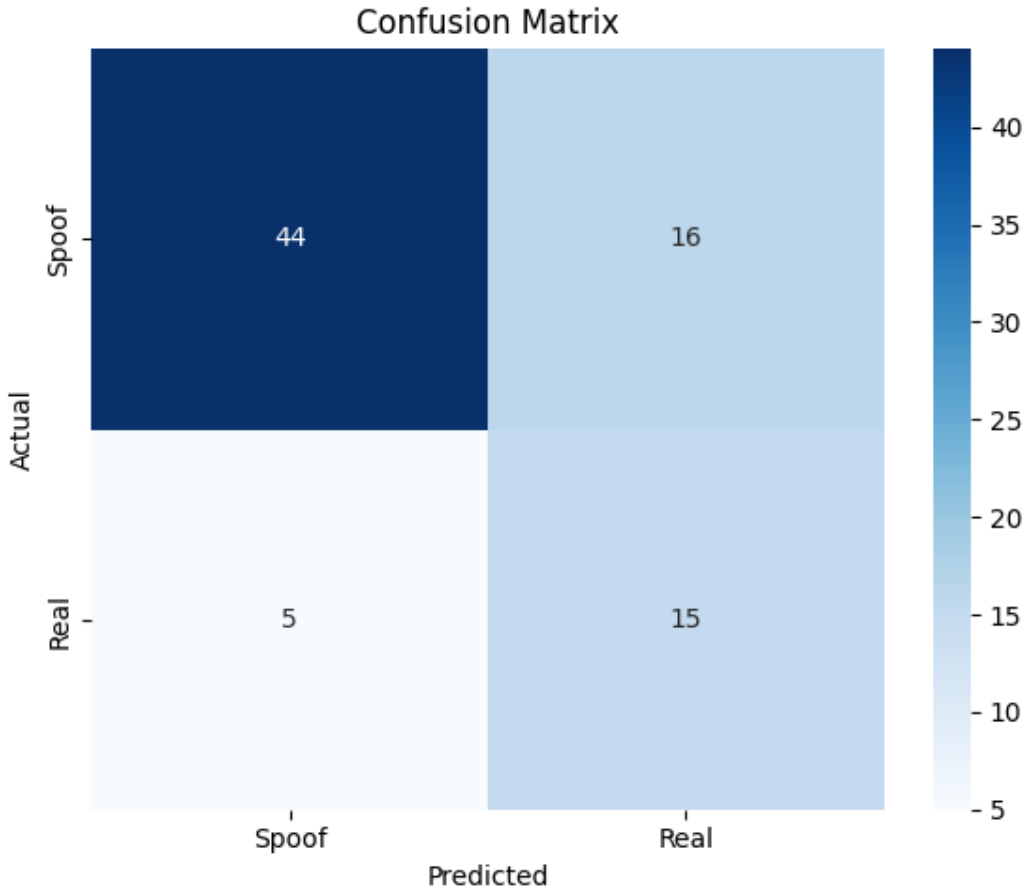

Figure 27 - Confusion matrix of improved model

While the first basic model misclassified most real samples, the fine-tuned model significantly reduced false rejections, correctly identifying 15 out of 20 real faces and 44 out of 60 spoof attempts. Although errors remain (particularly in distinguishing some sophisticated spoofs from real inputs), the system demonstrated clear robustness improvements after fine-tuning.

These findings support the conclusion that fine-tuning a pre-trained CNN is essential. Training from scratch would require much larger datasets to achieve comparable performance, whereas fine-tuning leverages the general visual representations already learned from ImageNet and adapts them to the specific domain of liveness detection. The improvement from an unusable baseline to a model with an accuracy close to 75% demonstrates an improvement for real-time deployment in the ERP authentication system.

7 Conclusion

7.1 Summary and Objectives Achieved

This thesis aims to study, evaluate possibilities and implement a facial recognition authentication system for corporate environments. The research focused on the limitations of traditional authentication mechanisms, which are often vulnerable not only to security attacks, but also to biased and not diverse learning.

Having said so, the work began with a state of the art and a systematic review, examining the evolution of facial recognition systems, their architecture, and associated evaluation metrics. This review confirmed both the potential and the challenges of biometric authentication, particularly around fairness, privacy, and robustness to spoofing. It also highlighted the promise of convolutional neural networks (CNNs) and the emerging Vision Transformers (ViTs), which were both evaluated and compared later.

This systematic review finished with 3 questions that needed to be answered to better understand what needed to be done:
Q1: How can the system ensure robust security against adversarial attacks and unauthorized access, while maintaining high usability standards?

Q2: Which models, between CNN and Vision Transformers, demonstrate superior performance in terms of accuracy, efficiency, and resilience for facial recognition-based authentication?

Q3: How do different evaluation metrics (e.g., accuracy, precision, recall, F1-score) impact the performance assessment of facial recognition systems, and which metric is most indicative of real-world effectiveness?

Building on these answers, the thesis continued to develop and compare the effectiveness and performance of CNNS and ViTs. After evaluating both in the same conditions, the decision came to advance with Convolutional Neural Networks given their historical success and proven efficiency.

Later, the thesis advanced for the architecture of real-time ERP authentication. The system integrates client-side image capture and server-side embedding generation to create a secure, and user-friendly interface, while maintaining GDPR compliance.

Another central contribution of the work was the exploration of anti-spoofing strategies. First, some of the most common models were compared and explored. After choosing the model, it was first evaluated without task-specific adaptation, revealing weak performance and confirming the necessity of fine-tuning. Naturally, through targeted training with augmentation and class balancing, the fine-tuned model achieved substantially higher accuracy and robustness, correctly distinguishing real users from spoofing attempts in most cases. While errors remain, this process showed how fine-tune can change everything.

In summary, the objectives set out at the start of the project ended up being achieved. The thesis reviewed and compared state-of-the-art models, implemented a complete facial authentication pipeline, developed and fine-tuned an anti-spoofing model, and demonstrated improvements in accuracy, security, and usability. Obviously, there are still improvements to make, this work can be a start of future research. In the end, the study demonstrates that facial recognition, when carefully designed with attention to ethics and robustness, can serve as a secure and user-friendly authentication solution for enterprise environments.

7.2 Limitations and Future Work

Naturally, despite the promising developments made, there are still some limitations.

Starting on spoofing detection, the first limitation is related to the dataset size and diversity. The anti-spoofing model was trained and evaluated on a relatively small database, which makes it more probable of generalizing different demographic groups, while having a low capacity of identifying small differences. Obviously, training it on a larger dataset would improve robustness and reduce its bias.

A second limitation is the range of attacks the model can identify. The model was primarily tested against print and replay attacks, and that makes it vulnerable against more sophisticated threats like 3D masks, silicone masks, and AI-generated deepfakes. Given that these complex attacks are growing fast, some future work should include exposure to more detailed and varied attacks to make sure that the system boosts its resilience.

Lastly, for the anti-spoofing developments, the use of broader architecture could also bring significant improvements. CNNs are efficient and suitable for real-time applications, but they focus primarily on local features, making them less ready than Vision Transformers (ViTs), that process images in their entirety and have shown promise in detecting globally distributed spoofing attacks, particularly in deepfakes and 3D mask scenarios. Having said so, exploring hybrid models that combine CNNs with some self-attention mechanisms could offer a better result in efficiency and robustness.

Beyond anti-spoofing, the face authentication workflow itself could also be improved. When the system starts being tested in real-world scenarios and not only under laboratory conditions, but some problems could also arise. Some of potential problems are latency under production loads or even scalability when handling large numbers of concurrent authentication requests.

Finally, apart from anti-spoofing, some privacy and security features could also be improved. While the system already complies with GDPR principles, future work could use other techniques to improve it even more on this matter, like safely storing and communicating data. Some examples are using encryption for storing the embeddings or using federated learning to improve the anti-spoofing model without ever exposing raw facial data to the server.

7.3 Final Considerations

This chapter marks the end of this thesis. The work done here represents a period of hard work and a lot of learning, where I had the opportunity to improve my knowledge of artificial intelligence applied to a practical and relevant problem. The process had many challenges, but those challenges were precisely what drove the development of new skills, from technical implementation to critical analysis and problem-solving. Even if some goals could not be totally achieved, the work showed valuable contributions to the academic community and a future application in industry. Hopefully, soon, this work will be used to implement a facial recognition system of authentication for an ERP and MES software.

References

[1] P. Grother, M. Ngan and K. Hanaoka, "Ongoing face recognition vendor test (FRVT) Part 2," 2018.

[2] Apple Inc., "Apple platform security," Apple, 2021. [Online]. Available: https://support.apple.com/guide/security/optic-face-touch-passcodes-passwords-sec9479035f1/web. [Accessed 12 December 2024].

[3] Statista, "Security & surveillance technology," Statista, 2024.

[4] S. SLiu, "Facial recognition market size worldwide in 2019 and 2024," Statista, 2020. [Online]. Available: https://www.statista.com/outlook/tmo/artificial-intelligence/computer-vision/facial-recognition/worldwide. [Accessed 14 December 2024].

[5] Cyberlink, "FACE OF THE FUTURE - KEY FINDINGS," Cyberlink, 2022.

[6] BleepingComputer, "The true (and surprising) cost of forgotten passwords," 14 11 2024. [Online]. [Accessed 15 December 2024].

[7] BleepingComputer, "Password reset calls are costing your org big money," 6 12 2022. [Online]. Available: https://www.bleepingcomputer.com/news/security/password-reset-calls-are-costing-your-org-big-money/. [Accessed 16 December 2024].

[8] Deloitte, "Cybersecurity insights for FS-ISAC members," Deloitte, 2020.

[9] C. &. E. J. Libby, "Facial Recognition Technology in 2021: Masks, Bias, and the Future of Healthcare.," Journal of medical systems, 2021.

[10] I. Adjabi, A. Ouahabi, A. Benzaoui and A. Taleb-Ahmed, *Past, Present, and Future of Face Recognition: A Review.,* Electronics, 2020.

[11] "Basic Learning Principles of Artificial Neural Networks," in *Foreign-Exchange-Rate Forecasting With Artificial Neural Networks*, Springer US, 2007.

[12] V. Asadpour, "Neural Network Principles and Applications," in *Digital Systems*, IntechOpen, 2018.

[13] L. a. E. G. a. C. O. Sára Pusztaházi, "Parametric Activation Functions for Neural Networks: A Tutorial Survey," vol. 12, 2024.

[14] Medium, "Derivative of the Sigmoid function," Medium, 7 7 2018. [Online]. [Accessed 03 02 2025].

[15] S. F. Frauke Guenther, "neuralnet: Training of Neural Networks," *The R Journal,* 2010.

[16] Medium, "A Practical Guide to ReLU," Medium, 30 11 2017. [Online]. [Accessed 03 02 2025].

[17] Es-sabery, F. a. HairM, A. a. Qadir, J. a. S. d. Abajo, B. a. Garcia-Zapirain, B. a. D. l. T. Díez and Isabel, "Sentence-Level Classification Using Parallel Fuzzy Deep Learning Classifier," 01 2011.

[18] A. Khan, A. Sohail and U. e. a. Zahoora, "A survey of the recent architectures of deep convolutional neural networks.," *Artif Intell Rev,* 2020.

[19] Medium, "Everything you need to know about Neural Networks," Medium, 4 12 2017. [Online]. [Accessed 03 02 2025].

[20] Li, Z. a. Liu, F. a. Yang, W. a. Peng, S. a. Zhou and Jun, "A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects," *IEEE Transactions on Neural Networks and Learning Systems,* vol. 33, pp. 6999-7019, 2022.

[21] S. Khan, M. H. Javed, E. Ahmed, S. A. A. Shah and S. U. Ali, "Facial Recognition using Convolutional Neural Networks and Implementation on Smart Glasses," IEEE, 2019.

[22] Medium, "My Goto Notes on Convolutional Neural Networks for any Deep Learning Interview," Medium, 8 10 2024. [Online]. [Accessed 03 02 2025].

[23] A. V. G. Dhillon, "Convolutional neural network: a review of models, methodologies and applications to object detection," *Prog Artif Intell,* 2020.

[24] A. H. Reynolds, "Convolutional Neural Networks (CNNs)," 2019. [Online]. Available: https://anhreynolds.com/blogs/cnn.html. [Accessed 03 02 2025].

[25] B. Kumar, K. Manisha, A. Sinha, A. Kumar and J. Kumar, "Deep Learning Based Image Classification for Automated Face Spoofing Detection Using Machine Learning: Convolutional Neural Network," IEEE, 2024.

[26] V. Rajput, "Pooling layers in Neural nets and their variants," Medium, 10 1 2022. [Online]. [Accessed 03 02 2025].

[27] A. Vaswani, ""Attention is all you need." Advances in Neural Information Processing Systems," 2017.

[28] S. Cristina, "Machine Learning Mastery," 6 1 2023. [Online]. [Accessed 03 02 2025].

[29] L. Faria, *Transformers - Língua Natural e Sistemas Conversacionais,* ISEP, 2024.

[30] L. Faria, "Transformers," 2023. [Online]. [Accessed 2025].

[31] K. Li and S. Meng, "TransGait: Vision Transformer Based Gait Recognition Network," IEEE, 2023.

[32] Bang, J.-H. a. Park, S.-W. a. Kim, J.-Y. a. Park, J. a. Huh, J.-H. a. S. Hoon, J. a. Sim and Chun-Bo, "CA-CMT: Coordinate Attention for Optimizing CMT Networks," 2023.

[33] D. K. J. P. Florian Schroff, "FaceNet: A Unified Embedding for Face Recognition and Clustering," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[34] GeeksforGeeks, "FaceNet – Usando o Sistema de Reconhecimento Facial," 13 06 2022. [Online]. [Accessed 03 02 2025].

[35] E. Shrestha, "Triplet Loss and Siamese Neural Networks," 24 10 2019. [Online]. [Accessed 03 02 2025].

[36] A. Dosovitskiy, "An Image is Worth 16x16 Words: Transformers for Image Recognition," *CoRR,* vol. abs/2010.11929, 202.

[37] "YouTube Faces DB," [Online]. Available: https://www.cs.tau.ac.il/~wolf/ytfaces/. [Accessed 15 01 2025].

[38] Z. Liu, P. Luo, X. Wang and X. Tang, "Deep Learning Face Attributes in the Wild," 2015.

[39] Guo, Y. a. Zhang, L. a. Hu, Y. a. He, X. a. Gao and Jianfeng, "MS-Celeb-1M: A Dataset and Benchmark for Large-Scale Face Recognition," in *ECCV*, 2016.

[40] A. L. J. Harvey, "MS-Celeb-1M (MS1M)," Exposing.ai, 01 01 2021. [Online]. Available: https://exposing.ai/msceleb/. [Accessed 15 01 2025].

[41] Q. Cao, L. Shen, W. Xie, O. M. Parkhi and A. Zisserman, "VGGFace2: A dataset for recognising faces across pose and age," vol. abs/1710.08092, 2017.

[42] Zhang, X. a. Ye, H. a. H. Z. a. Ye, X. a. Cao, Y. a. Zhang, Y. a. Yang and Min, "Understanding the (In)Security of Cross-side Face Verification Systems in Mobile Apps: A System Perspective," in *2023 IEEE Symposium on Security and Privacy (SP)*, 2023, pp. 934-950.

[43] Y. Liu, A. Jourabloo and X. Liu, *Learning Deep Models for Face Anti-Spoofing: Binary or Auxiliary Supervision,* 2018.

[44] Patel, K. a. Han, H. a. Jain and A. K., *Secure Face Unlock: Spoof Detection on Smartphones,* 2016.

[45] Nguyen, H. H., Y. J. and I. Echizen, "Multi-task learning for detecting and segmenting manipulated facial images and videos," in *Proceedings of the Biometrics Security and Privacy Workshop (BSPA)*, Florida, USA, 2019.

[46] M. Sharif, S. Bhagavatula, L. Bauer and M. K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in *ACM SIGSAC Conference on Computer and Communications Security*, Vienna Austria, 2016.

[47] A. S. Keresh, "Liveness Detection in Computer Vision: Transformer-based Self-Supervised Learning for Face Anti-Spoofing," 2024.

[48] A. Dosovitskiy, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," 2021.

[49] T. d.-e. i. t. &. d. t. attention, "Touvron, H., Cord, M., Douze, M.," in *ICML*, 2021.

[50] L. Yuan, "Tokens-to-Token ViT: Training Vision Transformers from Scratch on ImageNet," in *ICCV*, 2021.

[51] J. Chen and Y. Wang, "CelebA-Spoof: Large-Scale Face Anti-Spoofing Dataset with Rich Annotations," 2022.

[52] Shin and Y. L. a. Y. K. a. Jinho, "Robust face anti-spoofing framework with Convolutional Vision Transformer," 2023.

[53] Y. a. M.-V. H. a. L. L. S. a. G.-M. M. Martínez-Díaz, "Exploring the Effectiveness of Lightweight Architectures for Face Anti-Spoofing," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 10.1109/CVPRW59228.2023.00680, 2023, pp. 6392-6402.

[54] C. Ryando, "Comparison of Machine Learning Algorithms for Face Classification Using FaceNet Embeddings," *The Indonesian Journal of Computer Science,* vol. 13, 2024.

[55] M. C. Hugo Touvron, "Training data-efficient image transformers & distillation through attention," vol. abs/2012.12877, 2020.

[56] O. V. J. D. Geoffrey Hinton, "Distilling the Knowledge in a Neural Network," 2015.

[57] M. C. M. D. F. M. A. S. H. J. Hugo Touvron, "Training data-efficient image transformers & distillation through attention," *Cornell University,* 2021.