Instituto Tecnológico
y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Electrónica, Sistemas e Informática

Maestría en Sistemas Computacionales

DEEPFAKE IMAGE RECOGNITION USING A COMPUTATIONALLY SIMPLE DEEP LEARNING APPROACH: AN EXPLORATORY ANALYSIS.

TRABAJO RECEPCIONAL que para obtener el **GRADO** de

MAESTRO EN SISTEMAS COMPUTACIONALES

Presenta: Naum Jahaziel Nuño Contreras

Director: Dr. Iván Esteban Villalón Turrubiates

Fecha: 21 de mayo de 2026

Instituto Tecnológico y de Estudios Superiores de Occidente

Official Recognition of the Validity of Higher Education Studies in accordance with Secretarial Agreement 15018, published in the Official Gazette of the Federation on November 29, 1976.

Department of Electronics, Systems, and Computer Science

Masters Degree in Computer Systems

DEEPFAKE IMAGE RECOGNITION USING A COMPUTATIONALLY SIMPLE DEEP LEARNING APPROACH: AN EXPLORATORY ANALYSIS.

THESIS SUBMITTED in partial fulfillment for the **DEGREE** of

MASTER OF SCIENCE IN COMPUTER SYSTEMS

Presents: Naum Jahaziel Nuño Contreras

Director: Dr. Iván Esteban Villalón Turrubiates

Date: May 21 2026

Acknowledgments

The authors would like to thank the Ministry of Science, Humanities, Technology, and Innovation (Secretaría de Ciencia, Humanidades, Tecnología e Innovación, SECIHTI) of Mexico for the grant assigned to the CVU/2055386. Also, to the Instituto Tecnológico de Estudios Superiores de Occidente (ITESO) of Mexico, for the resources provided for this research under the project titled “Análisis de Datos para la Extracción de Conocimiento a Partir de Modelos Multidimensionales de Percepción Remota”.

AGRADECIMIENTOS

Los autores desean agradecer a la Secretaría de Ciencia, Humanidades, Tecnología e Innovación (SECIHTI) de México por el apoyo otorgado mediante la beca asignada al CVU/2055386. Asimismo, al Instituto Tecnológico y de Estudios Superiores de Occidente (ITESO) de México, por los recursos proporcionados para la realización de esta investigación, en el marco del proyecto titulado “Análisis de Datos para la Extracción de Conocimiento a Partir de Modelos Multidimensionales de Percepción Remota”.

DEDICATION

The author dedicates this thesis to his parents and family, whose unconditional support, love, and constant guidance have been the foundation throughout this academic journey. Their encouragement, patience, and unwavering belief have been essential in achieving this milestone.

The author also expresses sincere gratitude to his advisor, whose guidance, dedication, and continuous support were fundamental to the development of this project. His mentorship transformed each challenge into an opportunity for learning and personal and academic growth.

DEDICATORIA

El autor dedica esta tesis a sus padres y a su familia, quienes, con su apoyo incondicional, amor y guía constante han sido la base que ha sostenido cada paso de este camino académico. Su motivación, paciencia y confianza han sido fundamentales para alcanzar esta meta.

Asimismo, el autor expresa su sincero agradecimiento a su asesor, cuyo acompañamiento, orientación y compromiso fueron esenciales para el desarrollo de este proyecto. Su guía permitió transformar cada desafío en una oportunidad de crecimiento y aprendizaje.

Abstract

The rapid development of Artificial Intelligence (AI) has enabled the creation of highly realistic Deepfake images, raising concerns about misinformation and digital content integrity. Existing detection methods rely on statistical analysis, blockchain, or traditional machine learning, but often lack robustness. This study proposes a computationally simple Convolutional Neural Network (CNN) approach to detect Deepfake images with improved accuracy; this model will be referred during the paper as Deep Fake CNN (DFCNN). Experimental results demonstrate the model's effectiveness, contributing to reliable digital media authentication and the mitigation of manipulated content dissemination.

RESUMEN

El rápido desarrollo de la Inteligencia Artificial (IA) ha permitido la creación de imágenes Deepfake altamente realistas, lo que ha generado preocupaciones sobre la desinformación y la integridad del contenido digital. Los métodos de detección existentes se basan en análisis estadístico, blockchain o técnicas tradicionales de aprendizaje automático; sin embargo, con frecuencia carecen de robustez. Este artículo propone un enfoque basado en una Red Neuronal Convolucional (CNN) computacionalmente simple para la detección de imágenes Deepfake con una precisión mejorada; a este modelo se le hará referencia a lo largo del documento como Deep Fake CNN (DFCNN). Los resultados experimentales demuestran la efectividad del modelo, contribuyendo a una autenticación más confiable de los medios digitales y a la mitigación de la difusión de contenido manipulado.

TABLE OF CONTENTS

TABLE OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

ACC	Accuracy
Adam	Adaptive Moment Estimation
AI	Artificial Intelligence
AUC	Area Under the Curve
Celeb-DF	Celeb-DF (dataset)
CLRNet	Convolutional LSTM-based Residual Network
CNN	Convolutional Neural Network
CPU	Central Processing Unit
DFCNN	Deep Fake Convolutional Neural Network
DFDC	DeepFake Detection Challenge (dataset)
DL	Deep Learning
F1	F1-Score
FF++	FaceForensics++ (dataset)
GAN	Generative Adversarial Network
KNN	K-Nearest Neighbors
LSTM	Long Short-Term Memory
ML	Machine Learning
RAM	Random Access Memory
ReLU	Rectified Linear Unit
RGB	Red, Green, Blue
RNN	Recurrent Neural Network
SVM	Support Vector Machine

INTRODUCTION

Summary: *This chapter briefly presents the background of the study focused on Deepfake technology, an Artificial Intelligence–based technique that enables highly realistic manipulation of images, videos, and audios. The research problem is defined as improving the accuracy and robustness of manipulated content detection. Therefore, this project proposes the development of a Deep Learning model based on Convolutional Neural Networks (CNNs) to identify Deepfake modifications.*

Development and technological advancement have grown massively in this last decade, and although there are uses for the benefit and progress of humanity, there are also those who misuse these discoveries. One of these misuses is Deepfake, which is relatively new. It refers to the digital manipulation of multimedia content, such as images, videos, or audio. The Deepfake phenomenon emerged in 2017 when advanced artificial intelligence techniques were first used to replace the faces of famous people in adult videos. Since then, technology has advanced rapidly, enabling the creation of fake videos, images, and audios with a high degree of realism. These Deepfakes have raised serious concerns about the authenticity of digital content, as they can be used to spread false information, manipulate public opinion, and damage the reputation of individuals or groups.

Current solutions are based on techniques such as blockchain, which uses immutable records to verify the authenticity of multimedia content, statistical methods that analyze patterns and anomalies in data to identify manipulations, Machine Learning methods, which employ algorithms to detect suspicious features in media, and finally, Deep Learning, which simply identifies and classifies manipulations in images, videos, and audios.

The main objective of this project is to develop a model based on Deep Learning techniques, specifically using Convolutional Neural Networks (CNN), to detect and recognize modifications made by Deepfake in audios, images, or videos. This model will aim to improve accuracy and robustness in identifying manipulated content, thus contributing to the fight against misinformation and the protection of digital integrity.

1.1. Background

In recent years, Artificial Intelligence (AI) has enabled the creation of highly realistic manipulated multimedia, known as Deepfakes, using techniques such as Generative Adversarial Networks (GANs). Early examples appeared around 2017, and since then, methods like Face2Face, FaceSwap, and NeuralTextures have advanced facial and identity manipulation in images and videos.

Deepfake detection has become an important research area, with approaches ranging from Machine Learning and statistical methods to Deep Learning models based on Convolutional Neural Networks (CNNs), which are effective at automatically extracting features from large datasets such as FaceForensics++. This project builds on these developments to create a computationally efficient CNN model for detecting manipulated images.

1.2. Justification

In recent years, there has been an exponential growth in technological advances, particularly in everything related to Artificial Intelligence (AI), largely due to the increase in available computational power. Among the many advances offered by AI is the manipulation of multimedia elements such as images, videos, and audio. While these technologies have been positively leveraged in areas such as education and entertainment, there is also the other side of the coin, where they have been used for illegal and unethical purposes that can seriously affect individuals.

At this point, the term Deepfake emerges. It refers to applications that misuse technology to generate fake multimedia content with the intent of delivering misleading messages, spreading false propaganda, or, in the worst cases, producing pornographic material [1].

This is where motivation for this project arises. Although there are many ways to detect AI-manipulated content such as deep learning methodologies, traditional machine learning methods, statistical techniques, or blockchain-based approaches this project focuses on using a Deep Learning model based on Convolutional Neural Networks (CNNs) to identify whether a multimedia element has been manipulated by AI. This approach is particularly relevant because, in many cases, humans are unable to distinguish between real and digitally manipulated content [2].

1.3. Problem

The rapid growth of AI technologies has enabled the creation of highly realistic Deepfake multimedia content, including images, videos, and audio. While these technologies have beneficial applications in education and entertainment, they are increasingly being exploited for unethical and illegal purposes, such as spreading misinformation, fake propaganda, or non-consensual content. Detecting these manipulated elements is challenging because human perception is often unable to differentiate between real and AI-generated content.

Existing detection methods, including deep learning, traditional machine learning, statistical analysis, and blockchain-based approaches, have limitations in terms of accuracy, robustness, or computational complexity. Therefore, there is a practical need for a computationally efficient, reliable, and accurate method to detect AI-manipulated multimedia content, particularly Deepfake images, to protect individuals, organizations, and the integrity of digital media.

1.4. Hypothesis

A computationally simple Convolutional Neural Network (CNN), referred to as Deep Fake CNN (DFCNN), can detect Deepfake images with improved accuracy. It is expected that the experimental results will demonstrate the effectiveness of the proposed model, contributing to more reliable digital media authentication and to the mitigation of the dissemination of manipulated content.

1.5. Objectives

1.5.1. General objective

The aim is to develop a model based on deep learning techniques for recognizing modifications known as Deepfake that have been made in audios, images, or videos.

1.5.2. Specific objectives:

- Conduct a review of the state of the art to gain accumulated knowledge on the topic.
- Review and develop the theoretical framework to establish a solid theoretical foundation for the development of the project.
- Implement the code for the model that detects Deepfake generated images.
- Achieve solid model validation and ensure it produces satisfactory results for the project.

1.6. Scientific, Technological Innovation or Contribution

This work presents a computationally simple Convolutional Neural Network (CNN) model, referred to as Deep Fake CNN (DFCNN), designed to detect Deepfake images with improved accuracy. The novelty of this project lies in providing a practical, efficient, and reliable solution for identifying AI-manipulated multimedia content. This contributes to digital media authentication and helps prevent the dissemination of manipulated content, offering valuable applications for organizations, media platforms, and end users concerned with content integrity and security.

2. STATE OF THE ART OR TECHNICAL BACKGROUND

Summary: *This chapter reviews related work on Deepfake generation and detection. Deepfakes, created using deep learning techniques such as Generative Adversarial Networks (GANs), have evolved rapidly since 2017, enabling highly realistic facial manipulation through methods like Face2Face and FaceSwap. As misuse increased, research shifted toward detection of approaches based on Machine Learning and Deep Learning, particularly Convolutional Neural Networks (CNNs). Studies also highlight the importance of combining human judgment with automated models to improve detection accuracy.*

2.1. Deepfake history

In [1], it is mentioned that the term "Deepfake" was coined by combining the words "Deep Learning" and "fake," and it is described as a multimedia product created using Deep Learning tools. Its first reference was in 2017 when a person replaced someone's face in an adult video. This can be done using two neural networks, one generative and the other discriminative, which is known as Generative Adversarial Networks (GANs). The generative network is responsible for creating fake images using encoders and decoders, while the discriminative one determines their authenticity [1].

As mentioned, AI and everything related to it has grown exponentially in recent years, and in the context of Deepfake, many DL researchers have developed a wealth of information in the field of generative modeling. Shohel Rana et al. [1] highlight some of these advances: for instance, a group of researchers from the University of Washington introduced a method in which they synchronize lip movements with speech from an entirely different speech, and another example was when researchers in computer vision introduced the Face2Face method, which mimics a person's facial expressions on an avatar in real-time. In 2018, a group from Stanford developed a method to reanimate video portraits, among others. It was in 2017 when these techniques started to be used for illegal situations and/or to harm a group of people [1].

Research related to Deepfake only began in earnest in 2018. Likewise, Shohel et al. [1] mention four distinct techniques used for recognizing related products in computing. The main methods for Deepfake detection are Machine Learning, Deep Learning-based methods, statistical measurements, and blockchain, with a focus on deep learning modeling. These types of solutions are widely used due to their feature extraction and selection capabilities, learning directly from the data, and primarily making use of Convolutional Neural Networks (CNNs).

On the other hand, Matthew Groh et al., in their study published in [2], take a deeper look into everything surrounding Deepfake. They question the importance and relevance of deep learning models for detecting fake media, investigating whether humans, machine learning models, or the interaction between both are better at detecting media manipulation.

To test who is truly better at detecting media manipulation, experiments were conducted where both authentic and Deepfake videos were presented. The goal was to compare the performance of each, and it was observed that both were accurate in their own way, but they made different types of errors. The group that had access to model predictions was more accurate than either of the two subjects alone. However, the inaccurate predictions from the model often decreased the accuracy of the participants [2].

Something worth highlighting from the results is that, for political videos, humans were found to be more accurate than models due to their ability to use context and prior knowledge. Groh et al. [2] also found that participants were more accurate in low-quality videos, such as those that were grainy, blurry, or very dark, while machine learning models had difficulties in these contexts.

Groh et al. [2] mention that a promising strategy for Deepfake detection is the combination of both machine and human. They suggest that collaborative systems should be carefully designed to balance human and model predictions, as incorrect predictions from the model can lead to less accurate decisions. Furthermore, they emphasize the importance of considering the specific features of videos and the context in which they are presented to improve detection accuracy.

Deepfake has evolved to the point where it is almost impossible to visually detect between a fake and a real video. To achieve this, there are various techniques, one of which is Face2Face, where the goal is to transfer expressions from the source to the destination. Another is FaceSwap, used for facial identity manipulation,

which uses a graphics-based approach to create a model of the source in each frame and then projects this new model onto the target, minimizing the distance between frames. Similarly, one of the oldest techniques is Neural Textures, which uses Generative Adversarial Networks (GANs) to recreate faces [3].

Of the four methodologies available for potential Deepfake detection, Deep Learning will be chosen for the development of the model. Vedant Jolly et al. [3] developed a model based on Convolutional Neural Networks, using the dataset provided by FaceForensics++, which consists of over a thousand manipulated videos and more than a million manipulated images.

The main challenge in detecting this phenomenon lies in how we preprocess an image from the start. Firstly, the subject to be analyzed must be detected using facial recognition algorithms. The next step is to refine the facial features of the current image to eliminate as much noise as possible.

This is where the importance of choosing Convolutional Neural Networks for the analysis comes in, as they use techniques like blurring and Gaussian noise to ignore irrelevant high-frequency noise and sounds, thus allowing the recognition of more significant features and increasing the accuracy of the model [3].

Finally, this phenomenon has led to a decrease in trust in digital media, causing negative effects on individuals and potentially provoking political and social unrest. Therefore, it is crucial to develop methods capable of detecting fraud in the real world. Table 1 [5] shows diverse techniques and datasets used by researchers for the creation of false images/videos.

Table 1 Diverse methods for deepfake detection

Image	Video	Method	CelebA-HQ	CelebA	Other	FID	SSIM
✔	✔	Deep Convolution GAN	•	•	✔	49.3	•
✔	•	Lightweight Identity-aware Dynamic Network	•	•	✔	6.79	•
✔	•	GAN	•	•	✔	29.31	•
✔	•	StarGAN v2	✔	•	•	23.9	•
✔	•	Encoder-Decoder	•	✔	•	12.17	0.1717
✔	•	Feed Forward Network	•	✔	•	29.5	0.74
✔	✔	FSGAN	•	•	✔	•	0.51

2.2. Techniques used for deepfake detection

Same with deep-fake creation, various deep learning techniques can be used for the detection of images and videos. Table 2 [5] shows some of these techniques with their respective accuracy and shows which database was used.

Table 2 Deepfake detection techniques

Modality	Method	Evaluation Dataset	Performance		
to	Video -	SVM	Other/Custom ✓	ACC = 88.33	AUC -
Image -	Video ✓	SVM	Other/Custom ✓	ACC -	AUC FF++ = 56.8
Image ✓	Video -	Canny Edge Detection, Hough Transform	Face Forensics++ ✓	ACC -	AUC = 94
Image ✓	Video -	K-NN	Other/Custom ✓	ACC = 90.22	AUC -

Image ✓	Video -	SVM	Other/Custom ✓	ACC -	AUC = 89
Image ✓	Video -	Hybrid CNN	Other/Custom ✓	ACC = 95.75	AUC -
Image ✓	Video ✓	CNN-LSTM	FF++ ✓, Celeb-DF ✓, DFDC ✓	ACC FF++=91.21, Celeb-DF=79.49, DFDC=66.26	AUC -
Image -	Video ✓	Convolutional LSTM-based Residual Network (CLRNet)	Other/Custom ✓	ACC = 93.86	AUC -
Image -	Video ✓	Xception Networks	Other/Custom ✓	ACC -	AUC = 75
Image -	Video ✓	LSTM	Other/Custom ✓	ACC = 94.29	AUC -
Image -	Video ✓	CNN	Other/Custom ✓	ACC -	AUC = 99

The first method to investigate is the Support Vector Machine (SVM). These are learning models used for regression and classification tasks. Their functionality is based on detecting the hyperplane that maximizes the margin between points from different classes/groups, which is why they are very useful for Deepfake detection, due to their ability to handle high-dimensional spaces. Their main steps are feature extraction, model training, and finally, classification [6].

On the other hand, we have CNN models (convolutional neural networks), which are formed by various layers: the first one being the input layer, in which basically we have the input image. This layer is followed by the convolutional layer; here we apply filters to the input image for feature extraction for discovering relevant patterns. Subsequently, to each output of the convolutional layers we have activation functions that apply a transformation to each outcome. At the end, final layers process the last output to make a final prediction [6].

In [6], Stankov et al. talks about doing a hybrid model mixing two different deep learning methodologies. The hybrid approach proposed includes mixing both CNN and SVM models. The purpose of making a hybrid approach is to leverage the strengths of both. CNN is used for feature extraction and uses the extracted features as input for the SVM model for classifying the input as real or fake. When comparing the algorithms, it is shown that the CNN model has high precision due to all the features mentioned earlier, while the SVM, although effective, falls short. The combination of both has better precision, accuracy, and sensitivity, which is an evaluation metric in classification models that measures the model's ability to correctly identify positive instances (in the case of face manipulation detection, for example, "deepfake" instances or manipulations). Table [6] shows the results of image detection using this approach. The test showed that the CNN-SVM approach offers the best performance in terms of accuracy, precision, recall, and F1 score.

Table 3 Model metrics

Model	**Accuracy**	**Precision**	**Recall**	**F1-score**
SVM	0.92	0.91	0.93	0.92
CNN	0.95	0.94	0.96	0.95
CNN+SVM	0.97	0.96	0.98	0.97

Another strategy used for fake image detection was the use of the technique Canny edge detector, which explores the use of corneal specular highlight as a cue to expose GAN synthesized human faces. Under normal circumstances, a human face catch by the camera exposes the following features under the same light environment: the anatomic parameters of the two eyes including the distance between the centers of the pupils

and the diameters of the corneal limbus, the relative location of the eyes due to head position, and the distance the light sources to both eyes [7].

Li, et al [7] explains the methodology behind extracting corneal activity. First, they run a face detector to isolate the human face, then the area of both eyes is cropped. Applying a Canny edge detector and a Hough transform to find the corneal limbus. Then by extracting and analyzing corneal specular highlights we can calculate the similarity between both eyes with a score that goes from [0-1], the lower the score, we can deduce that this corresponds to a fake image, figure 1 [7] shows some results of the experiment. The AUC obtained was 0.94, which indicates that this method is very effective at detecting fake images.

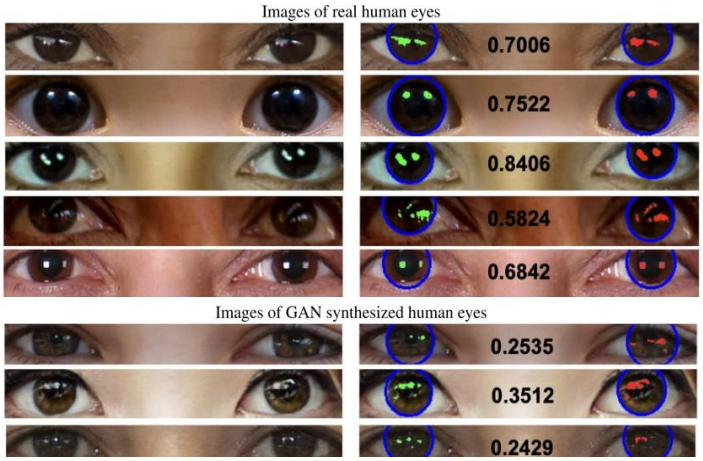


Figure 1 Canny Edge detector

Another course of action is the use of KNN models. In Deepfake detection using "DeepFake Detection by Analyzing Convolutional Traces," [8] the KNN method is used to classify an image or video as real or fake based on the convolutional traces (features) extracted by the CNN. The KNN algorithm works by searching for the nearest labeled examples (real or fake) in the feature space and predicts the class based on the majority vote of the nearest neighbors. An accuracy of 90.2% was achieved.

It is well known that both machine and deep learning techniques are leading the way in deep fake detection. We have gone through CNN, SVM, KNN learning models, and other techniques such as canny edge detectors. Agrawa, et al [9] also tested Recurrent Neural networks (RNN), which is an architecture that handles sequential data by retaining an internal state or memory. Furthermore, they are an excellent choice for sequence-related applications including speech recognition, natural language processing, and time-series prediction.

Another alternative was the use of Xception models, which is a deep convolutional neural network architecture designed to improve existing Inception models. Comparing it to CNN models, Xception uses less parameters, less computer capacity, and its accuracy is the same or even greater. Agrawa, et al [9], tested 4 metrics on these models. First one being accuracy, which is the ratio of properly predicted cases (both true positives and true negatives), to the total number of occurrences. Secondly, precision is the percentage of expectedly positive cases that turn out to be positive. Then, recall, which is the model's ability to properly forecast positive cases. And finally, we have the F-1 score metric, which basically is the means of recall and accuracy. Figure 2 [9] shows these results. When looking at recall and precision metrics, we can see that CNNs are invaluable in image related tasks.

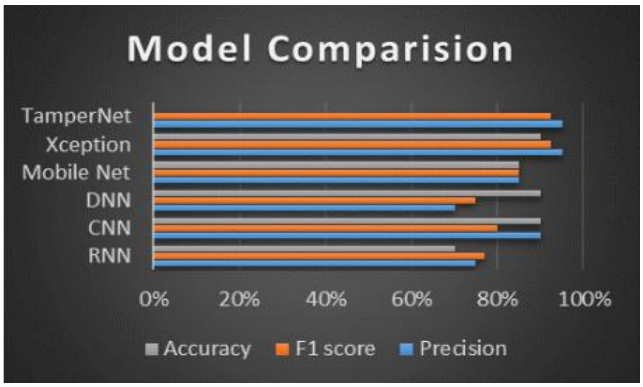


Figure 2 Model comparison

There have also been studies where they compare diverse CNN models. Using FaceForensics++ dataset, four CNN models were trained: ResNet-152, MobilenetV3, Convnext Large, and EffecientNetB7. The last one, Efficient-NetB7 achieved the highest testing accuracy, making it the best model to use when it comes to detecting deepfakes. This can be seen in table 4 [10]. Furthermore, to enhance deepfake detection, researchers of this study propose to explore techniques such as data augmentation, regularization, and larger datasets

Table 4 CNN Model metric comparison

CNN Model	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss	Testing Accuracy
EfficientNet_B7	87.88%	0.3877	67.5%	0.6260	75%
MobileNetV3	85.25%	0.4079	70.75%	0.6299	61.33%
ConvNeXt	94.44%	0.2077	73.25%	1.2783	67.17%
ResNet-152	91.19%	0.2993	80.75%	0.6678	61%

Naveen Chandra, et al [11] precisely used data augmentation in their CNN model. These authors suggest a method for identifying fake images by looking for convolutional traces that are left behind by Generative adversarial Networks (GAN). In this study, all the images from the dataset were resized to 256x256 pixels, and through data augmentation, more data is introduced to improve accuracy and reduce overfitting, which was a problem seen in study [10]. Three augmentations were used: rescale, which is the process where each pixel is divided in value by 255 to rescale the image's pixel values from 0 to 1. The second one being a horizontal flip, which can aid in the diversity of the training set and the bright range (help improving the capacity of the model to new data and improve its recognition of images having different levels of brightness).

During the following steps, the ReLu activation function was employed, as well as the Adaptive Moment Estimation (Adam) algorithm as optimization methods. This study was able to identify deep-fake images with a high degree of accuracy thanks to the combination of the Adam optimization method and the ReLU activation function. The CNN model used was formed by 5 convolutional layers and 2 dense layers. Results are shown in table 5 [11]. We can see that the best method to use for deep fake detection is the CNN model.

Table 5 Model deepfake detection results

	Precision	Recall	F1 Score
Real	94.25%	96.29%	95.17%
Deepfake	96.17%	94.99%	95.28%

3. THEORETICAL / CONCEPTUAL FRAMEWORK

Summary: *This chapter provides the theoretical and conceptual foundations necessary to understand the development of the project. It introduces the key principles underlying convolutional neural networks, including their structure, operation, and common applications. Additionally, it reviews essential concepts related to image processing, machine learning, and classification tasks that support the methodological choices made later in the work. Relevant prior research and established models are also discussed to contextualize the project within existing literature. Altogether, this framework offers the conceptual grounding required for interpreting the methodology, results, and decisions throughout the study.*

Artificial Intelligence (AI) has emerged as a transformative field within computer science, driving significant advancements across domains such as computer vision, natural language processing, and data analytics. Its primary objective is to design computational systems capable of emulating aspects of human cognition, including learning, reasoning, and decision-making, through the application of mathematical models and algorithmic techniques. Within this broad discipline, Machine Learning (ML) and Deep Learning (DL) have become foundational paradigms, enabling systems to learn from large-scale data and iteratively improve their performance without explicit task-specific programming. In this context, a rigorous understanding of these methodologies is essential for exploring advanced applications, such as the generation and detection of synthetic media, including deepfakes, which rely on increasingly sophisticated neural network architectures.

3.1. Machine Learning

Let's start with machine learning basics. Machine Learning (ML) is a branch of artificial intelligence focused on developing algorithms capable of identifying patterns and making predictions from data. In traditional ML, models learn by mapping input features to expected outputs through statistical inference and optimization processes. This learning paradigm can be broadly categorized into supervised learning, where the model is trained using labeled examples; unsupervised learning, which focuses on discovering hidden structures in unlabeled data; and reinforcement learning, where an agent improves its behavior through interactions with an environment. Regardless of the category, the core idea of ML is to enable systems to improve their performance over time without being explicitly programmed for each task. These foundational principles set the stage for more advanced approaches such as Deep Learning, which extends ML by leveraging neural network architectures capable of automatically extracting complex representations from raw data [12].

3.2 Deep Learning

Building upon these foundations, Deep Learning (DL) emerges as a specialized area within Machine Learning that uses multilayer artificial neural networks to automatically learn hierarchical representations from large volumes of data. Unlike traditional ML models, which rely on handcrafted features, DL architectures can extract increasingly abstract patterns through successive layers of computation. This capability makes DL particularly effective for tasks involving high-dimensional and unstructured data, such as images, audio, and video. Within DL, Convolutional Neural Networks (CNNs) have become the dominant architecture for visual recognition and image-based applications due to their ability to capture spatial dependencies through convolutional operations and shared weights. These properties allow CNNs to learn robust feature maps that encode textures, edges, shapes, and more complex visual patterns, forming the basis for many modern computer vision systems, including deep fake generation techniques [12].

3.3 Convolutional Neural Networks

As mentioned above, the method to be used for the development of the Deepfake model will be based on Deep Learning, specifically on Convolutional Neural Networks (CNNs). These networks consist of multiple layers, each of which contributes unique features to the overall architecture. The first layer receives the image, also known as the input layer. The next layer is referred to as the convolutional layer, in which filters are applied to the image to extract multiple features along with their relevant patterns. Another important layer is the pooling layer, which reduces the dimensionality of the extracted features through sampling. The final layers process the output from the previous layers to generate a final prediction [12].

Face detection and alignment are essential preprocessing steps in any deepfake generation pipeline, as they ensure that facial regions are consistently extracted and normalized before being fed into a neural network. Modern face detection approaches such as Multi-task Cascaded Convolutional Networks (MTCNN), Histogram of Oriented Gradients (HOG)-based detectors, and RetinaFace—localize facial regions with high precision, even under challenging conditions involving occlusions, extreme poses, or low lighting. Following detection, facial alignment techniques use landmark localization to rotate, scale, and crop the face into a standardized

orientation. This normalization step minimizes variability across training samples, allowing the encoder and decoder networks to focus on identity-relevant features rather than pose or spatial inconsistencies. Proper alignment significantly enhances model performance and stability, making it a critical component of effective deepfake generation [13].

The use of Deep Learning to detect these manipulations is driven by several key factors that highlight the effectiveness and accuracy of this technology. As stated by Vendant Jolly et al. [3], convolutional neural networks are extremely effective at extracting complex and subtle features from images and videos, particularly through pixel-level analysis. These models can adapt and learn over time, as their ability to identify new forms of manipulation improves when more data is introduced.

According to Albawi Saad et al. [13], CNN has had major breakthroughs in various fields related to pattern recognition, for example: image processing and voice recognition. Convolutional Neural Networks (CNNs) are a class of deep neural network architectures specifically designed for processing data with spatial or temporal structure, such as images, videos, and sequential signals. Their main advantage over traditional artificial neural networks lies in their ability to significantly reduce the number of trainable parameters while maintaining high representational power.

These neural networks have the following basic elements. The core component of a CNN is the convolutional layer, which applies to a set of learnable filters (kernels) over local regions of the input. Instead of full connectivity, CNNs rely on local receptive fields and weight sharing, allowing the network to detect meaningful patterns such as edges, textures, and shapes regardless of their spatial position in the input. The convolution operation consists of an element-wise multiplication between the filter and the input region, followed by a summation that produces a feature map [13].

An important parameter in the convolution process is the stride, which defines the step size of the filter as it moves across the input. Increasing the stride reduces the spatial resolution of the output feature maps and lowers computational cost, at the expense of potentially losing fine-grained spatial information. To address boundary information loss and control output dimensions, padding is commonly applied by adding zero-valued pixels around the input. Padding enables deeper architecture by preventing excessive shrinking of feature maps across layers. Following the convolutional operation, a nonlinear activation function is applied to introduce nonlinearity into the network. Among various activation functions, the Rectified Linear Unit (ReLU) has become the most widely used due to its computational simplicity, ability to mitigate the vanishing gradient problem, and tendency to produce sparse activations that improve training efficiency in deep networks [13].

Another essential element is the pooling layer, which performs down-sampling feature maps to reduce dimensionality and computational complexity. The most common pooling method, max pooling, retains the maximum value within local regions, providing a degree of translation invariance and helping to prevent overfitting. However, pooling reduces spatial resolution, which may affect precise localization tasks. Finally, Albawi Saad et al. [13] stated that CNN architecture typically includes one or more fully connected layers at the final stages of the network. These layers combine the high-level features learned by the convolutional and pooling layers to perform classification or regression tasks. While effective, fully connected layers contain most of the network parameters and are often regularized using techniques such as dropout to reduce overfitting. Overall, the integration of convolutional layers, nonlinear activations, pooling operations, and fully connected layers enables CNNs to learn hierarchical feature representations, progressing from low level patterns to high level semantic concepts. This hierarchical learning capability explains the strong performance of CNNs in computer vision, pattern recognition, and related machine learning applications.

3.4 Autoencoders

Autoencoders represent a fundamental component in traditional deepfake generation pipelines, as they provide an efficient framework for learning compressed latent representations of facial images. An autoencoder consists

of two primary components: an encoder, which maps the input image into a lower-dimensional latent space, and a decoder, which reconstructs the original image from this compressed representation. In deepfake systems, a shared encoder is trained to extract identity-invariant features—such as facial structure, pose, and illumination—while two separate decoders are trained to reconstruct the faces of two different individuals. During inference, the encoder processes the source face and the decoder corresponding to the target identity reconstructs it, effectively producing a synthesized face that maintains the expressions and movements of the source while adopting the identity of the target. This encoder–decoder mechanism allows deepfake models to generalize across large variations in pose, lighting, and facial expressions, making autoencoders a foundational tool in early and modern face-swapping architectures [13].

3.5 Generative Adversarial Networks

Generative Adversarial Networks (GANs), introduced by Goodfellow et al. in 2014, have significantly advanced the field of synthetic image generation and are widely used in deepfake creation and detection. A GAN consists of two competing neural networks: a generator, which attempts to produce realistic synthetic images, and a discriminator, which learns to distinguish between real and generated samples. Through this adversarial training process, the generator progressively improves its ability to create highly detailed and photorealistic outputs. GAN-based architectures such as StyleGAN, StyleGAN2, and StyleGAN3 have demonstrated state-of-the-art performance in facial synthesis, enabling unprecedented control over facial attributes, pose, and realism. While the methodology employed in this work focuses primarily on convolutional neural networks and autoencoder-based structures, GANs remain a crucial reference in the theoretical landscape due to their impact on the evolution of deepfake technology and their relevance in both generative modeling and forensic detection research [13].

4. METHODOLOGY

Summary: *This chapter describes the methodology followed to develop the project, detailing the decision-making process behind the design of the CNN architecture and the evaluation of different structural alternatives. It explains the preprocessing steps applied to the data prior to model training, including dataset selection, cleaning, and normalization procedures. The section also outlines the challenges and technical blockers encountered during development, as well as the tools, datasets, and requirements needed to ensure a robust and reproducible workflow. Together, these elements provide a comprehensive view of how the final methodological framework was constructed.*

The primary focus of this exploratory analysis is to develop and evaluate the performance of a CNN model in detecting Deepfake images (DFCNN). The experiments were conducted on a system equipped with an Intel(R) Core (TM) i5-1035G1 CPU 1.00GHz (base frequency 1.20 GHz) and 8 GB of RAM. This hardware configuration was sufficient for training and evaluating the convolutional neural network on a limited dataset, though more complex models or larger datasets may benefit from more powerful computational resources.

4.1 Database

For the development and evaluation of the proposed Deep Learning model, the FaceForensics++ database was employed as the primary source of training and testing data. FaceForensics++ is one of the most widely used benchmarks in the field of Deepfake detection, offering a large collection of both authentic and manipulated facial videos generated using state-of-the-art Deepfake techniques.

In this study, real images were extracted directly from the original videos provided in the dataset. To create the corresponding manipulated dataset, the fake images were generated by performing a frame-by-frame extraction of the manipulated videos, ensuring a precise alignment between the real and fake images. The resulting dataset consists of a total of 2,360 images, both real and fake, divided into a training set of 1,888 images and a testing set of 472 images.

4.2 Preprocessing of database

First figures 3 and 4 illustrate an example of an original image and a deepfake image.

Figure 3 Original photo from database

Figure 4 Manipulated photo

All images in the dataset were resized to 128 × 128 pixels as a preprocessing step. To address overfitting and improve classification accuracy (an issue noted in study [9]), a series of data augmentation techniques were applied. First, images were normalized so that pixel intensity values ranged from 0 to 255. Each image was then rotated from 0° to 180° in 10° increments, increasing variability in the training set. For each rotation, brightness levels were adjusted to enhance robustness against illumination changes. In addition, image filtering and the Canny edge detector were applied to further diversify the dataset. Together, these augmentations produced a more comprehensive and resilient training set, leading to improved model performance in detecting fake images.

It is worth mentioning that in another preprocessing step, images from both real and fake datasets were read in binary format and processed to isolate the facial region. First, the rembg library was employed to remove the background from each image, ensuring that only the subject remained. The resulting images were then converted to RGB and subsequently transformed into grayscale. Face detection was then carried out using the Haar cascade classifier (haarcascade_frontalface_default.xml), and when one or more faces were identified, the first detected region was cropped and extracted. Each cropped face was subsequently converted into grayscale and stored for further analysis. This procedure standardized the input data by focusing exclusively on facial regions, thereby reducing background noise and emphasizing discriminative features relevant to classification. Figure 5 and figure 6 both illustrate an example of this.

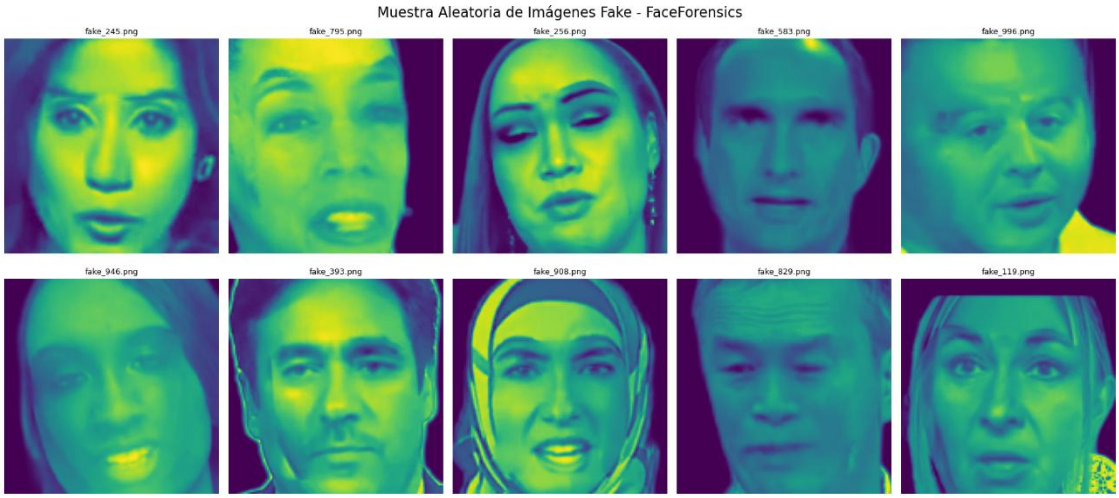


Figure 5 Face detection with Haar cascade classifier fake images

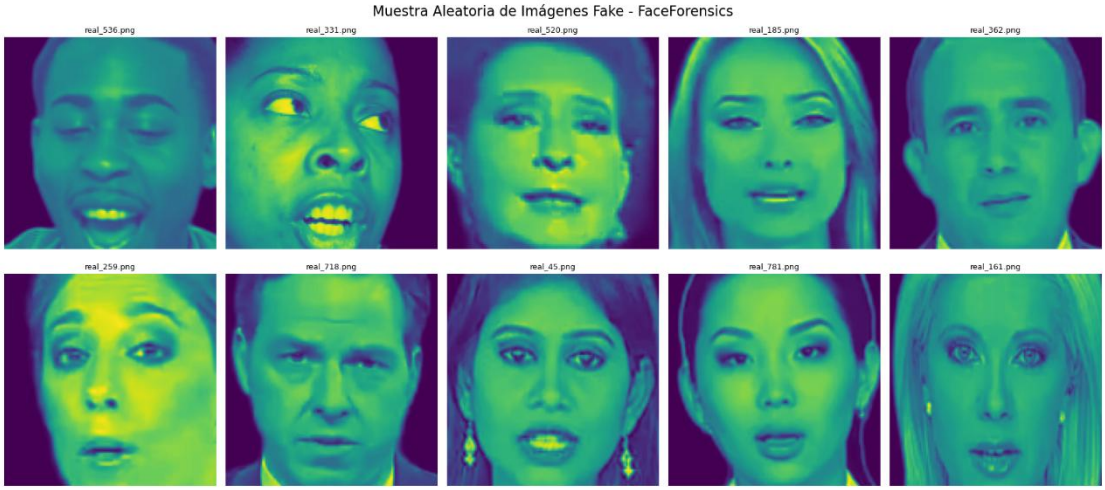


Figure 6 Face detection with Haar cascade classifier real images

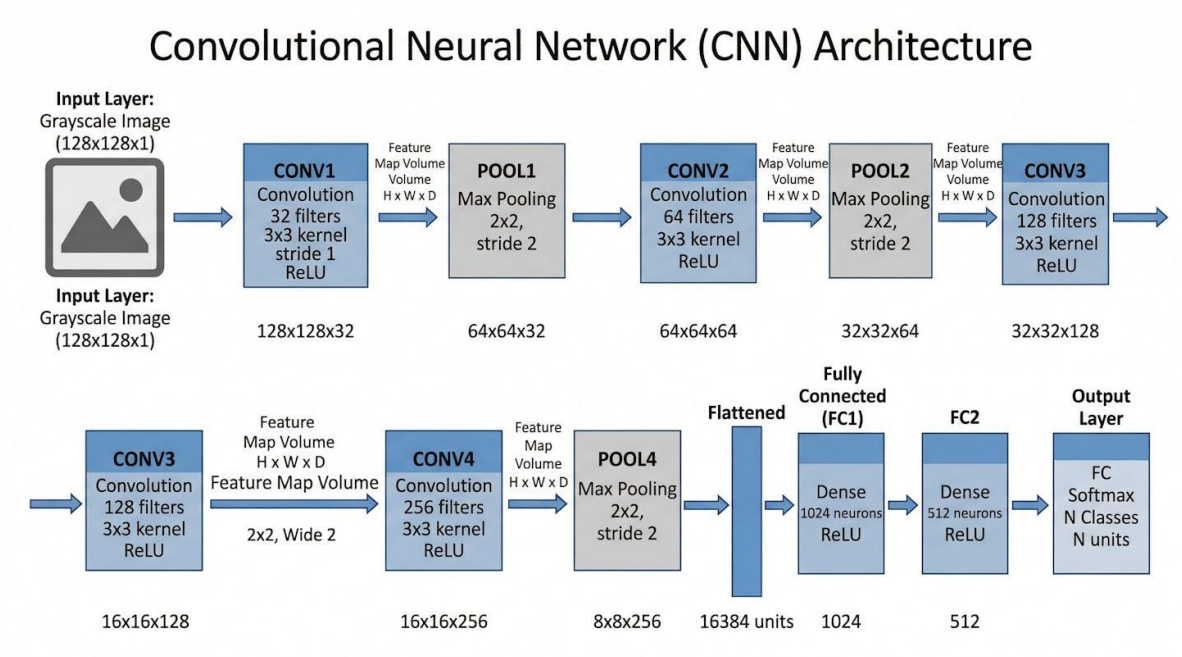
4.3 Model experimentation

All datasets were divided into training and test sets (80% for training data and 20% for test data). The architecture of the proposed model is based on a deep convolutional neural network (CNN) designed for binary image classification (real vs. fake). The input layer accepts RGB images of size 128 × 128 × 3. The feature extraction stage consists of five convolutional blocks. Each block applies a two-dimensional convolution using a 5 × 5 kernel with "same" padding and ReLU activation, followed by a 2 × 2 max-pooling operation. The

number of filters is progressively increased across the blocks (32, 64, 128, 256, and 256), which allows the network to capture low- to high-level spatial features in a hierarchical manner.

After convolutional processing, the feature maps are flattened and passed to the classification stage. This stage begins with a fully connected dense layer of 512 units with ReLU activation. To improve generalization and training stability, batch normalization is applied immediately after this dense layer. Furthermore, a dropout layer with a rate of 0.5 is introduced to reduce overfitting by randomly deactivating half of the neurons during training.

The final classification layer consists of two output neurons with a SoftMax activation function, enabling the network to estimate class probabilities for the binary task. The model is implemented using the TensorFlow library and trained for 50 epochs using the Adam optimizer. DFCNN completed training in 43 minutes and 9 seconds. Categorical cross-entropy is used as the loss function, and classification accuracy is monitored as the primary performance metric. This configuration ensures a balance between depth and regularization, enabling the model to extract robust discriminative features while maintaining stability during training and reducing the risk of overfitting.



4.4 Requirements

This section describes the main requirements for the development, training, and evaluation of the Convolutional Neural Network (CNN) designed for deep-fake detection. These requirements are categorized into dataset requirements, computational resources, and software tools.

4.4.1 Dataset Requirements

The performance of a deep-fake detection model strongly depends on the quality, diversity, and size of the training data. The dataset must contain both authentic (real) and manipulated (deepfake) samples to enable supervised learning.

Key dataset requirements include:

- Balanced classes: A comparable number of real and fake samples to avoid classification bias.

- Diversity of manipulations: Deepfake videos generated using different methods (e.g., face swapping, facial reenactment, GAN-based synthesis) to improve generalization.
- Variation in conditions: Differences in lighting, resolution, compression levels, facial expressions, poses, and ethnicities.
- Frame extraction: Videos are decomposed into frames to allow CNN-based image-level processing.
- Ground truth labels: Each sample must be accurately labeled as real or fake.
- Publicly available datasets commonly used for deepfake detection include FaceForensics++, DeepFake Detection Challenge (DFDC), and Celeb-DF, which provide large-scale, labeled data suitable for CNN training.

4.4.2 Computational Requirements

The experiments were conducted on a system equipped with an Intel(R) Core (TM) i5-1035G1 CPU 1.00GHz (base frequency 1.20 GHz) and 8 GB of RAM. This hardware configuration was sufficient for training and evaluating the convolutional neural network on a limited dataset, though more complex models or larger datasets may benefit from more powerful computational resources.

4.4.3 Software and Framework Requirements

The development of CNN relies on modern deep learning frameworks and supporting libraries.

Required software components include:

- Programming language: Python, due to its extensive ecosystem for machine learning.
- Deep learning frameworks: TensorFlow or PyTorch for CNN implementation and training.
- Libraries for video processing and frame extraction.
- Data handling libraries: NumPy and Pandas for numerical operations and dataset management.
- Evaluation tools: Scikit-learn for performance metrics such as accuracy, precision, recall, F1-score, and ROC-AUC

5. RESULTS AND DISCUSSION

Summary: *This chapter presents the results obtained from the experiments and analyzes their significance in relation to the objectives of the project. It discusses the performance of the CNN model, comparing metrics such as accuracy, loss, and other relevant indicators to evaluate its effectiveness. The section also interprets the observed behaviors, highlighting trends, strengths, and limitations revealed through testing. Furthermore, it contrasts the findings with expectations and related work, providing context to understand their relevance. Overall, this chapter offers a comprehensive discussion of the outcomes and their implications for the problem addressed.*

5.1. Results

After training the model for 50 epochs, it was evaluated using a dataset ranging from approximately 500 to 4000 images. The results showed a clear improvement in performance on both the training and validation sets. Table 3 illustrates how the model's accuracy has progressively improved.

Table 6 Deepfake model metrics

Images	Time	Accuracy	Loss	Epochs
500	20min	0.8118	0.4	50
1000	31min	0.9127	0.2039	50
1500	32min	0.8988	0.2244	50
2000	46min	0.9276	0.2	50
3000	48min	0.9259	0.1789	50
4000	102min	0.9716	0.081	50

Figure 7 provides a clearer visual representation of how the model's performance improved as the number of training images increased.

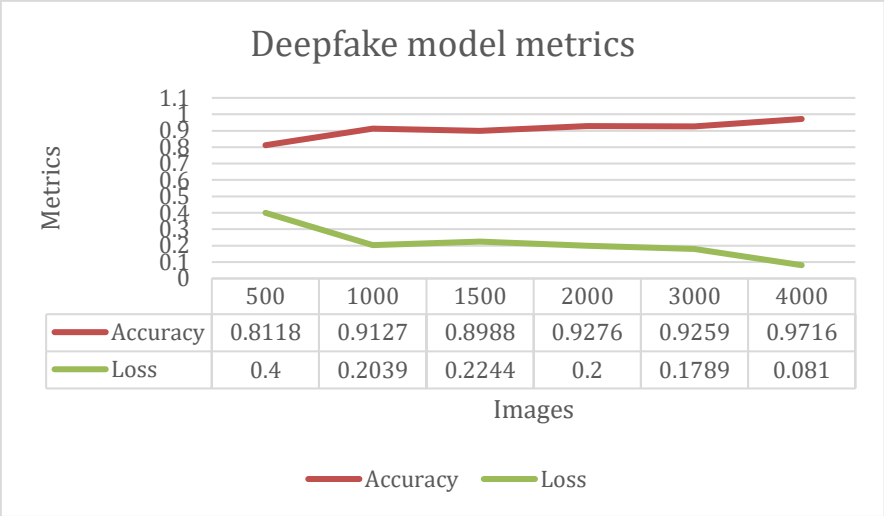


Figure 7 Deepfake model metrics

After training the model for 50 epochs using a dataset of 4,000 images, the model demonstrated strong performance on both the training and validation sets. Figure 8 illustrates how the model's accuracy progressively improved throughout the training process, indicating that the model was able to learn meaningful patterns from the data.

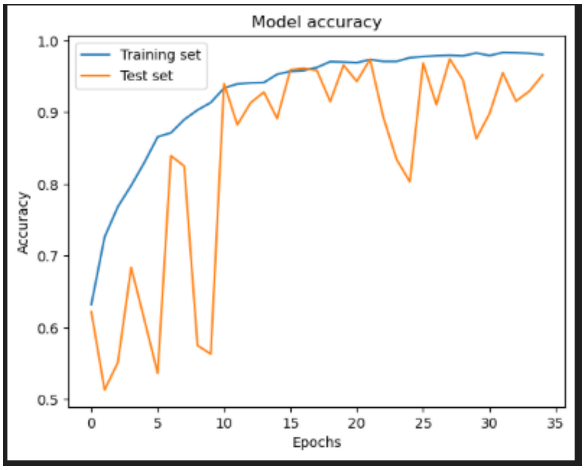


Figure 8 Model accuracy dataset 4000 images

In addition to accuracy, it is important to analyze the model's loss during training. Figure 9 shows the evolution of the model's loss across the training epoch. The decreasing trend in the loss value indicates that the model gradually reduced its prediction errors as learning progressed.

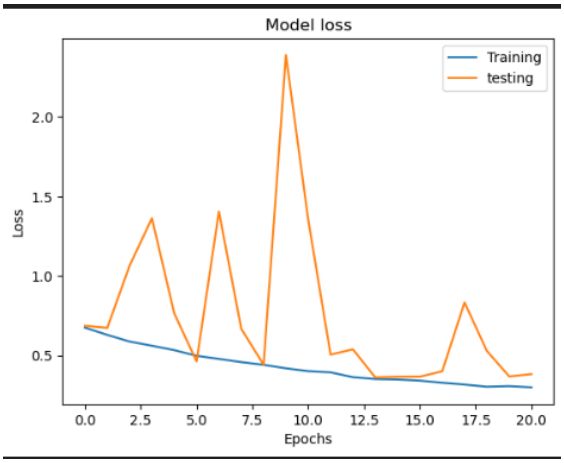


Figure 9 Model loss dataset 4000 images

To further evaluate the classification performance, a confusion matrix was generated using the test dataset. The confusion matrix displays the number of correct and incorrect predictions made by the model for each class.

Figure 10 presents the confusion matrix obtained from the model trained with the dataset of 4,000 images. The matrix compares the predicted labels with the true labels for the two classes: real and fake images. The results show that most samples were correctly classified, as indicated by the high values along the diagonal of the matrix. This suggests that the model can effectively distinguish between real and deepfake images.

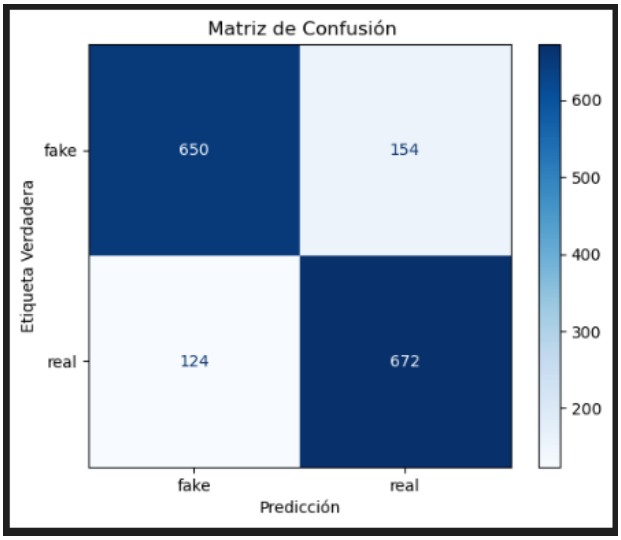


Figure 10 Confusion matrix 4000 images

As shown in the previous figures, both the training and validation sets exhibited consistent improvement in performance with each completed epoch. After the model finished training, it was evaluated using randomly

selected images from the dataset. Figure 11 illustrates an experiment performed on a randomly chosen fake image, where the model successfully classified the image into the correct category.

To further evaluate the model's performance, additional tests were conducted using multiple random samples from the dataset. Figure 12 shows five randomly selected images evaluated by the model, along with their corresponding predictions. The results demonstrate the model's ability to correctly classify most of the images as either real or fake, providing further evidence of its effectiveness in detecting deepfake content.

Figure 11 DFCNN Experiment on fake image

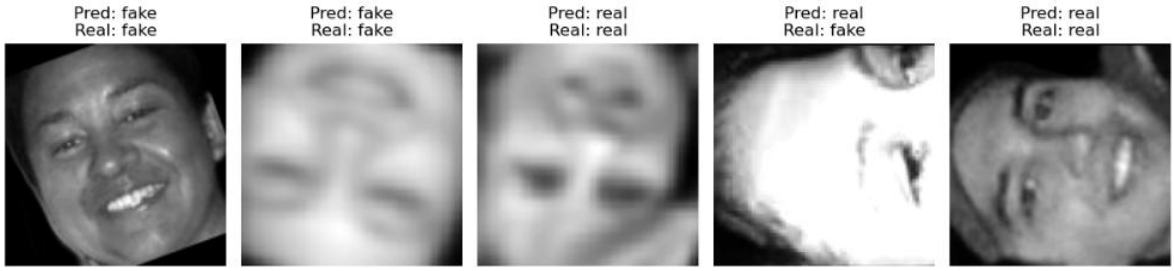


Figure 12 Five randomly selected images evaluated by the model

5.2. Discussion

When comparing the proposed DFCNN model with the baseline models presented in Table 1, it becomes evident that this approach offers a promising direction for advancing deepfake detection. The architecture demonstrates competitive performance, suggesting that CNN-based models, when properly configured and trained, can effectively distinguish between real and manipulated images.

Table 3 provides a comparative analysis of performance metrics across previous models and DFCNN. While the results show that DFCNN outperforms or matches existing methods in key areas such as accuracy, it is important to recognize that this is only a starting point, and there is much left to be done.

Table 7 Comparative analysis of performance metrics across previous models and the DFCNN

Model	**Performace**	
	Accuracy	

SVM	83.33	89
CannyEdge Detection	-	94
KNN	90.22	-
CNN	-	99
Xception Networks	-	75
Hybrid CNN	95.75	-
DFCNN	99.21	-

The results reported on this paper show the qualitative analysis of the overall performance of the CNN method applied to Deepfake image detection. However, the implementation performance and the optimization of the CNN architecture (DFCNN) are a matter of further studies. In the following link you can find both the python code and the dataset used in [14].

6. CONCLUSIONS

Summary: *This chapter presents the overall conclusions of the work carried out, discussing how the initial objectives were successfully fulfilled while highlighting the main contributions and impact of the project. It also reflects the limitations encountered throughout the development process and the lessons learned from addressing them. In addition, the chapter outlines potential directions for future work that could extend, refine, or build upon the results achieved, offering opportunities for further exploration and improvement.*

This research focused on the design, implementation, and evaluation of a deep learning model for the detection of deepfake media. The proposed Deep Fake Convolutional Neural Network (DFCNN) was developed as a computationally efficient architecture capable of identifying manipulated visual content.

The first objective of this research, which consisted of conducting a state-of-the-art review, was successfully accomplished through the analysis of existing deepfake detection techniques and deep learning approaches. This review provided a comprehensive understanding of the current challenges and methodologies in the field, allowing the project to identify effective strategies for the design of the proposed model.

The second objective, the development of a theoretical framework, was also achieved. Key concepts related to deep learning, convolutional neural networks, digital image manipulation, and deepfake generation techniques were examined and integrated to establish a solid conceptual basis for the project. This theoretical foundation supported the methodological decisions taken during the model design and implementation stages.

Another major objective involved the implementation of a deep learning model capable of detecting deepfake content. This objective was fulfilled through the development of DFCNN architecture using deep learning libraries and tools. The model was trained on a dataset containing both real and manipulated images, allowing it to learn relevant visual features associated with deepfake content.

The final objective focused on validating the model and ensuring that it produced satisfactory results. This objective was achieved through multiple experimental evaluations. The model was trained for 50 epochs using a dataset of 4,000 images, during which both training and validation accuracy showed consistent improvement over time. Additionally, the loss function demonstrated a decreasing trend throughout the training process, indicating that the model progressively reduced prediction errors and improved its learning performance. These

objectives mentioned have exhaustive results shown in figures 7 through 10, where it is shown that after each epoch accuracy is improving up until getting an overall of >90% accuracy.

Further evaluation using a confusion matrix revealed that most of the images were correctly classified, as evidenced by the high values along the diagonal of the matrix. Additional testing using randomly selected images from the dataset also demonstrated the model's ability to correctly identify whether an image was real or fake. These results provide empirical support for the initial hypothesis, confirming that a computationally simple convolutional neural network such as the proposed DFCNN can effectively detect deepfake images with promising accuracy.

The findings of this research highlight the potential of lightweight CNN-based architecture as practical tools for improving digital media authentication and helping mitigate the spread of manipulated visual content. By demonstrating that effective detection can be achieved without extremely complex architectures, this work contributes to the development of more accessible deepfake detection solutions. For code please refer to GitHub repository in references [14].

6.1. Future work

Despite the encouraging results, there remains considerable room for improvement. Future work may involve refining the model architecture, optimizing preprocessing techniques, and exploring alternative configurations such as different activation functions, normalization strategies, and more advanced regularization methods. These improvements could enhance the model's ability to generalize to more complex or previously unseen datasets.

Additionally, training the model using more powerful computational resources would allow the exploration of deeper architecture and the use of larger and more diverse datasets. Future research could also incorporate temporal analysis for video deepfake detection, multi-frame feature extraction, or hybrid architectures that combine convolutional neural networks with attention mechanisms or transformer-based models.

In conclusion, the proposed DFCNN model demonstrates strong potential as an effective approach for deepfake detection. While the field continues to evolve rapidly, the results of this study show that carefully designed convolutional neural networks can serve as a solid foundation for the development of more robust and scalable deepfake detection systems.

References

[1] M. S. Rana, M. N. Nobi, B. Murali and A. H. Sung, "Deepfake Detection: A Systematic Literature Review," in IEEE Access, vol. 10, pp. 25494-25513, 2022, doi: 10.1109/ACCESS.2022.3154404

[2] M. Groh, Z. Epstein, C. Firestone and R. Picard, "Deepfake Detection by Human Crowds, Machines, and Machine-Informed Crowds," in Psicological and Cognitive Sciences, vol. 119, no. 1, pp. 1-11, 2022, doi: 10.1073/pnas.2110013119.

[3] V. Jolly, M. Telrandhe, A. Kasat, A. Shitole and K. Gawande, "CNN based Deep Learning model for Deepfake Detection," 2022 2nd Asian Conference on Innovation in Technology (ASIANCON), Ravet, India, 2022, pp. 1-5, doi: 10.1109/ASIANCON55314.2022.9908862.

[4] D. Garg and R. Gill, "Deepfake Generation and Detection - An Exploratory Study," 2023 10th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), Gautam Buddha Nagar, India, 2023, pp. 888-893, doi: 10.1109/UPCON59197.2023.10434896

[5] D. Garg and R. Gill, "Deepfake Generation and Detection - An Exploratory Study," 2023 10th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), Gautam Buddha Nagar, India, 2023, pp. 888-893, doi: 10.1109/UPCON59197.2023.10434896. keywords:

{Deep learning;Deepfakes;Computational modeling;Benchmark testing;Artificial intelligence;Faces;Audio-visual systems;Deepfakes;Deep Learning;Forgery;Audio Deepfake;Video Deepfake;Image Deepfake;Face swap},

[6] I. S. Stankov and E. E. Dulgerov, "Detection of Deepfake Images and Videos Using SVM, CNN, and Hybrid Approaches," 2024 XXXIII International Scientific Conference Electronics (ET), Sozopol, Bulgaria, 2024, pp. 1-5, doi: 10.1109/ET63133.2024.10721497. keywords: {Support vector machines;Deepfakes;Computational modeling;Scalability;Machine learning;Media;Streaming media;Vectors;Convolutional neural networks;Security;Deepfake;SVM;CNN;Combined Approach;Image Analysis;Video Analysis;Machine Learning}.

[7] S. Hu, Y. Li, and S. Lyu, “Exposing gan-generated faces using in-consistent corneal specular highlights,” in ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2021, pp. 2500–2504.

[8] L. Guarnera, O. Giudice and S. Battiato, "DeepFake Detection by Analyzing Convolutional Traces," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 2020, pp. 2841-2850, doi: 10.1109/CVPRW50498.2020.00341. keywords: {Gallium nitride;Generative adversarial networks;Videos;Computer architecture;Data models;Convolution;Feature extraction},

[9] M. Vashistha, S. Jain, S. Pandey, A. Pradhan and S. Tarwani, "A Comparative Analysis of Machine Learning and Deep Learning Approaches in Deepfake Detection," 2024 IEEE Region 10 Symposium (TENSYMP), New Delhi, India, 2024, pp. 1-8, doi: 10.1109/TENSYMP61132.2024.10752209. keywords: {Deep learning;Deepfakes;Visualization;Machine learning algorithms;Accuracy;Nearest neighbor methods;Media;Prediction algorithms;Feature extraction;Classification algorithms;Deepfake Detection;Face Morphing;Convolutional Neural Networks;Machine Learning;Deep Learning;Support Vector Machine;K-Nearest Neighbour;Face Recognition},

[10] P. Ritter, D. Lucian, Anderies and A. Chowanda, "Comparative Analysis and Evaluation of CNN Models for Deepfake Detection," 2023 4th International Conference on Artificial Intelligence and Data Sciences (AiDAS), IPOH, Malaysia, 2023, pp. 250-255, doi: 10.1109/AiDAS60501.2023.10284611. keywords: {Training;Deepfakes;Analytical models;Computational modeling;Forensics;Focusing;Streaming media;deepfake detection;CNN models;comparative analysis;accuracy;LSTM layer},

[11] N. C. Gowda, V. H N and D. R. Ramani, "Efficient Identification of DeepFake Images using CNN," 2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), Bengaluru, India, 2024, pp. 1542-1547, doi: 10.1109/IDCIoT59759.2024.10467555. keywords: {Training;Resistance;Deepfakes;Privacy;Neural networks;Data augmentation;Data models;DeepFake;Generative Adversarial Networks (GANs);Deeplearning;Convolutional Neural Network (CNN)},

[12] D. Garg and R. Gill, "Deepfake Generation and Detection - An Exploratory Study," 2023 10th IEEE Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON), Gautam Buddha Nagar, India, 2023, pp. 888-893, doi: 10.1109/UPCON59197.2023.10434896.

[13] S. Albawi, T. A. Mohammed and S. Al-Zawi, "Understanding of a convolutional neural network," 2017 International Conference on Engineering and Technology (ICET), Antalya, Turkey, 2017, pp. 1-6, doi: 10.1109/ICEngTechnol.2017.8308186. keywords: {Convolution;Neurons;Convolutional neural networks;Feature extraction;Image edge detection;machine learning;artificial neural networks;deep learning;convolutional neural networks;computer vision;Image recognition},

[14] jahaziel2903, “DeepfakeDetection,” GitHub repository. Available: https://github.com/jahaziel2903/DeepfakeDetection