IDENTIFYING DEEPFAKE IMAGES WITH ARTIFICIAL INTELLIGENCE - ENHANCED CONVOLUTIONAL NEURAL NETWORK

A Thesis Presented

by

Dawit Yetmgeta

to

The Faculty of Informatics

of

St. Mary's University

In Partial Fulfillment of the Requirements
for the Degree of Master of Science

in

Computer Science

January, 2025

ACCEPTANCE

IDENTIFYING DEEPFAKE IMAGES WITH ARTIFICIAL INTELLIGENCE (AI) -ENHANCED CONVOLUTIONAL NEURAL NETWORK (CNN)

By

Dawit Yetmgeta

Accepted by the Faculty of Informatics, St. Mary's University, in partial fulfillment of the requirements for the degree of Master of Science in Computer Science

Thesis Examination Committee:

Internal Examiner
{Full Name, Signature and Date}

External Examiner
{Full Name, Signature and Date}

Dean, Faculty of Informatics
{Full Name, Signature and Date}

January 2025

DECLARATION

I, the undersigned, declare that this thesis work is my original work, has not been presented for a degree in this or any other universities, and all sources of materials used for the thesis work have been duly acknowledged.

Dawit Yetmgeta

Signature

Addis Ababa

Ethiopia

This thesis has been submitted for examination with my approval as advisor.

___________Dr. Million Meshesha___________
Name

________million________
Signature

Addis Ababa, Ethiopia

January 2025

Acknowledgment

To begin with, I would like to express my deepest gratitude to Dr. Million Meshesha for his unwavering support and invaluable guidance throughout my M.Sc. thesis. His expertise, patience, and encouragement were instrumental in shaping my research, and I am incredibly thankful for the wisdom and insight he provided.

I would also like to extend my heartfelt thanks to the FaceForensics++ team for their guidance in selecting the data for my research. Their expertise and collaboration were critical to the success of this project, and I truly appreciate their support.

Special thanks to God for providing me with the strength, wisdom, and perseverance throughout this journey.

Finally, I would like to express my deepest gratitude to my family for their endless love, encouragement, and unwavering belief in me. Their constant support has been my source of strength, and I could not have completed this thesis without them.

Table of Contents

List of Acronyms

3D	Three-Dimensional
AI	Artificial Intelligence
CNN	Convolutional Neural Network
F1	F1 Score
GAN	Generative Adversarial Network
Grad-CAM	Gradient-weighted Class Activation Mapping
Inception	Inception Network (a type of CNN architecture)
PyTorch	An open-source machine learning framework
ResNet	Residual Network
ROC	Receiver Operating Characteristic
VGG	Visual Geometry Group (referring to VGGNet)

List of Tables

List of Figures

ABSTRACT

The rapid advancement of deepfake technology poses significant challenges to the authenticity and integrity of digital media, leading to widespread concerns in various sectors, including politics, media, and personal relationships. This study aims to develop an AI-enhanced Convolutional Neural Network (CNN) model for detecting deepfake images, addressing the limitations of traditional detection methods that struggle against sophisticated manipulation techniques. By leveraging state-of-the-art deep learning architectures, specifically the XceptionNet model, this research explores the efficacy of advanced feature extraction techniques and data augmentation strategies to improve detection accuracy.

The proposed system utilizes the FaceForensics++ dataset, which includes both authentic and manipulated images, to train and evaluate the model. Experimental results demonstrate that the AI-enhanced CNN model significantly outperforms traditional approaches, achieving a binary classification accuracy of 86.91% and a type classification accuracy of 70.50%. These findings indicate that the model effectively identifies subtle artifacts and inconsistencies that are characteristics of deepfake manipulations.

This research not only contributes to the field of digital forensics but also emphasizes the need for ongoing advancements in detection methodologies to combat the evolving landscape of deepfake technology. Future work will focus on expanding the dataset, enhancing real-time detection capabilities, and integrating interdisciplinary approaches to address the broader societal implications of deepfakes. Ultimately, this study aims to empower individuals and organizations with reliable tools to discern authentic media from manipulated content, fostering a safer and more trustworthy digital environment.

Keywords: Deepfake Detection, Convolutional Neural Network (CNN), XceptionNet Model, AI-Enhanced Model, FaceForensics++ Dataset, Binary Classification, Type Classification

CHAPTER ONE

INTRODUCTION

1.1.Background

Deepfakes, a term that combines "deep learning" and "fake," are highly realistic videos designed to portray individuals doing or saying things they never actually did. These videos are created using advanced artificial intelligence (AI), specifically neural networks, which can mimic a person's facial expressions, movements, voice, and tone with incredible accuracy. To create a deepfake, AI systems are trained on videos of two people, allowing the technology to swap one person's face with another's seamlessly. At their core, deepfakes rely on facial mapping and AI algorithms to digitally alter videos [1] [2].

The concept of deepfakes first gained widespread attention in 2017, when a Reddit user shared fake videos that falsely depicted celebrities in compromising situations [2]. These videos are difficult to detect because they blend real visuals with fabricated audio, creating an illusion of authenticity. Once shared on social media, they spread quickly, often misleading viewers into believing they are real.

From a technical perspective, deepfakes are created using a type of AI called Generative Adversarial Networks (GANs) [2]. This technology involves two neural networks working together: the generator, which creates fake content, and the discriminator, which evaluates how convincing the content is. The two systems constantly improve by challenging each other, with the generator striving to produce content that can fool the discriminator. For instance, GANs can analyze a series of photos and create a new image that looks like the subject without directly copying any specific photo. As the technology advances, GANs are expected to require less input data, making it easier to swap faces, voices, or even entire bodies. Researchers have already developed methods to generate deepfake videos using just a single image, such as a selfie, showing how accessible this technology is becoming [3] [4].

Another key technology behind deepfakes is Convolutional Neural Networks (CNNs), which are designed to process image and video data. CNNs consist of layers that work together to identify patterns and features, such as edges, shapes, and textures, in the input data [5].

The convolutional layer is particularly important as it detects these features by applying filters that scan the input data. These filters identify patterns in specific areas of an image or video, creating a feature map that highlights where these patterns occur. This feature map is then processed further by other layers, such as pooling layers, to refine the analysis and improve the system's understanding of the input [3].

In a convolutional neural network (CNN), pooling layers are essential for reducing the spatial dimensions of feature maps created by convolutional layers. This process involves applying operations like max pooling, which selects the largest value within a small region, or average pooling, which calculates the mean value in that region. By compressing the size of the feature maps, pooling layers help the network become more resistant to minor shifts or distortions in the input data. Additionally, this dimensionality reduction decreases the computational load and the number of parameters in the model, improving efficiency and reducing the risk of overfitting [3], [6].

At the end of a CNN's structure, the fully connected layer takes the refined features produced by earlier layers and maps them to the output. In this layer, each neuron connects to every neuron in the preceding layer, enabling the model to integrate the learned features for final decision-making. This is particularly important in classification tasks, where the network predicts a probability for each class, indicating how likely the input belongs to a particular category. The final prediction is based on this probability distribution [5], [7].

CNNs operate by using convolutional layers to identify spatial patterns and relationships within input data. Pooling layers then streamline these outputs by condensing their size while maintaining the most relevant features. The fully connected layers follow, combining these processed features to deliver the network's predictions [5].

The concept of AI-enhanced CNNs involves incorporating advanced techniques to further improve their performance [3]. A key example is transfer learning, which leverages pre-trained models built on large datasets and adapts them for specific applications. This approach is particularly advantageous when working with limited datasets, as it allows models to achieve higher accuracy with less training. Another enhancement involves attention mechanisms, such as self-attention or spatial attention, which enable the network to focus on critical parts of the input and detect finer details.

CNNs have gained widespread use in image and video analysis tasks, including classification and object detection. To expand their capabilities, researchers have introduced multi-scale processing techniques [8]. These approaches enable networks to analyze data at multiple levels of detail, enhancing their ability to recognize intricate patterns. For instance, multi-scale architectures process inputs at various resolutions, while pyramid pooling aggregates information from different spatial scales. These methods allow CNNs to integrate global context with fine-grained details, making them more effective for complex tasks like object tracking and video segmentation. By embracing multi-scale processing, CNNs can better understand the spatial and temporal aspects of data, broadening their applicability across diverse multimedia scenarios.

1.2.Motivation of the study

Deepfake technology has become a source of major concern in modern society, given its ability to manipulate images and videos with remarkable realism. This has profound implications for politics, media, and personal relationships, as these altered videos can mislead and deceive on a large scale. The growing threat underscores the need for robust detection methods to identify and counteract manipulated media effectively [1], [9].

This study focuses on creating a detection system powered by an AI-enhanced Convolutional Neural Network (CNN) to address the challenges posed by deepfakes. The goal is to design a highly accurate and efficient tool that distinguishes authentic media from falsified content. By empowering individuals, organizations, and platforms with reliable detection capabilities, this research seeks to promote a safer and more credible digital environment.

The field of deepfake detection has advanced significantly, transitioning from basic machine learning models to sophisticated deep learning approaches. Early detection techniques often relied on analyzing individual frames or applying simple algorithms, which struggled with issues like video compression artifacts and inconsistencies across frames. The adoption of CNNs has revolutionized the field by enabling models to automatically extract spatial and temporal features from videos, leading to improved detection performance. Innovative methods, such as those proposed by researchers like Jiameng Pu et al. [10] and Patel et al. [3] have introduced groundbreaking frameworks like NoiseScope and Dense CNN architectures. These approaches focus on detecting unique noise signatures and incorporating data augmentation to enhance model robustness. Despite these advancements, challenges remain, particularly in detecting high-quality deepfakes, such as expression-swapping and face2face manipulations. Continued research is essential to refine detection techniques and ensure they can adapt to the rapidly evolving nature of deepfake technology.

1.3.Statement of the problem

The widespread proliferation of deepfake images has emerged as a significant and pressing challenge, as it poses a critical obstacle in accurately discerning between genuine and manipulated visuals [11]. This issue has far-reaching implications, contributing to the potential spread of misinformation and deception across a wide array of contexts. The advent of advanced algorithms utilized in the creation of deepfake manipulations has rendered traditional image forensics methods inadequate in effectively identifying sophisticated alterations. Traditional image forensics methods, which rely on detecting specific image characteristics and artifacts, struggle to keep up with advancements in deepfake technology. Deepfake algorithms can now seamlessly blend and manipulate visual elements, creating highly realistic and convincing images that evade detection by conventional forensic techniques. These advanced deepfake methods can conceal or minimize the telltale signs and distortions that traditional forensics would typically identify, making it increasingly difficult to distinguish real and fabricated visuals. The pace of deepfake content production also poses a challenge to forensic analysis, as the proliferation of manipulated visuals outpaces the ability of traditional methods. Addressing this challenge requires developing robust solutions, such as integrating machine learning and artificial intelligence algorithms to detect deepfake manipulations, and cross-disciplinary collaboration between experts in computer vision, image processing, and digital forensics [12].

Jiameng Pu et al. [10] have developed the NoiseScope detection framework to identify GAN-generated images based on noise patterns, achieving impressive accuracy in various datasets. Similarly, Patel et al. [3] proposed a Dense CNN architecture for detecting deepfake images, improving performance with extensive data augmentation techniques. However, these methods often struggle with the challenges posed by postprocessing operations such as the application of 3D filters and blurriness removal, which enhance deepfake realism

Deepfake detection methods suffer from postprocessing operations that can enhance deepfake videos by removing minor artifacts like blurriness and applying 3D filters, making them highly realistic and challenging to detect. Additionally, existing datasets for deepfake detection have limitations such as video and synthesized audio not being lip-synced, and participants not facing the camera in most recorded deepfake videos, which affects the development of a robust benchmark dataset [9].

It is therefore the aim of this study to develop a robust and efficient deepfake detection system by leveraging AI-enhanced Convolutional Neural Networks (CNNs) so as to address the growing concerns surrounding the authenticity of digital content, particularly in light of the increasingly sophisticated deepfake technologies.

1.4.Research Questions

To investigate and find solution for the above stated problem, the following research questions are formulated.

1. What are the suitable CNN models to detect and classify various types of deepfake manipulations?
2. What specific AI techniques can be integrated into CNN architectures to enhance their performance in deepfake image detection?
3. What is the performance of AI-enhanced CNN models in deepfake image detection?

1.5.Objective of the Study

1.5.1. General Objective

The general objective of this study is to develop an AI-enhanced CNN model for accurately identifying deepfake images and enhancing the detection capabilities of manipulated visuals.

1.5.2. Specific Objectives

To achieve the general objective of the study, the following specific objectives are attempted.

- ✓ To review previous related works to find suitable methods and techniques.
- ✓ To collect data and pre-process images for preparing a dataset for training and testing.
- ✓ To select suitable CNN models for experimentation
- ✓ To identify best AI techniques for integrating to CNN architecture.

- ✓ To construct an optimal model that differentiate between authentic and manipulated images.
- ✓ To evaluate the performance of the AI-enhanced CNN model in detecting various types of deepfake manipulations with high accuracy.

1.6.Scope and Limitation of the study

This study focuses on developing and implementing an AI-enhanced CNN model specifically designed for identifying deepfake images. The research involves training the model on a diverse dataset of authentic and manipulated images to improve its detection capabilities. The study encompasses the evaluation of different deepfake generation techniques and the performance analysis of the CNN model in accurately identifying manipulated visuals including testing phases to assess the effectiveness of the developed deepfake detection system. The study utilizes images from the "FaceForensics" dataset, which contains 224x224 images of deepfake faces, with a total of over 100,000 images from 5,000 videos. The dataset includes a diverse collection of both authentic and manipulated facial images, covering various subjects, poses, and lighting conditions.

By using this comprehensive image dataset, the AI-enhanced CNN model is trained on a representative sample of authentic and deepfake visuals, improving its ability to accurately identify manipulated images. The performance of the developed deepfake detection system extensively evaluated using various metrics to assess its effectiveness. However, this study is limited to the use of the FaceForensics dataset and does not include the evaluation of other datasets or deepfake detection methods beyond the CNN model.

1.7.Methodology of the study

1.7.1. Research Design

This study utilizes an experimental approach, a research method designed to investigate the effects of independent variables on dependent variable by manipulating conditions while maintaining strict control over external influences [4]. This approach is widely recognized for its capacity to establish causal relationships, allowing researchers to derive robust findings about cause and effect. Its structured design is ideal for testing research questions and studying causal mechanisms in environments where features can be carefully regulated. Randomly assigning participants to experimental and control groups eliminates potential biases and reduces the influence of confounding variables, thereby increasing the reliability and internal validity of the results [8].

This methodology is particularly well-suited for systematically exploring the influence of key features in a reproducible manner.

The research plan is built on a systematic process for designing, training, and evaluating the CNN model. It begins by assessing various well-known architectures, such as VGG, ResNet, Inception, and their respective variations, to determine the most suitable structure. The model's performance is further refined through the meticulous adjustment of hyperparameters, including the learning rate, batch size, number of layers, and activation functions. Advanced optimization strategies, such as grid search, random search, and Bayesian optimization, are employed to fine-tune these parameters. To ensure the model's reliability and ability to generalize across diverse data subsets, cross-validation techniques like k-fold cross-validation and leave-one-out cross-validation are applied throughout the training process.

1.7.2. Data Preparation

The FaceForensics dataset is preprocessed rigorously and comprehensively to ensure the optimal format and quality for training the CNN model. This involves normalizing the pixel values to a common range (between 0 and 1) to improve the model's convergence and stability. To further enhance the model's performance, various data augmentation techniques, such as random flipping, rotation, scaling, and color jittering, have been applied to the training data to increase the diversity and size of the dataset, thus improving the model's ability to generalize to unseen data. The preprocessed dataset is then be split into training, validation, and testing subsets, ensuring a balanced distribution of authentic and manipulated images in each subset.

1.7.3. Implementation Tools and Techniques

The research leverages the capabilities of prominent framework named PyTorch to develop the AI-enhanced CNN model. This state-of-the-art tool is widely recognized for their efficiency and versatility, offering optimized functionalities for constructing, training, and evaluating deep learning models. Their support for rapid experimentation and prototyping makes them ideal for this study's needs. In addition to these frameworks, supplementary libraries play a key role in enhancing the workflow. OpenCV is employed for image processing tasks, Scikit-learn is used to assess model performance, and Matplotlib is utilized for visualizing data and results. Together, these tools create a cohesive and efficient environment for model development and analysis.

Table 1.1 below summarizes programming languages with their packages and techniques used in this research.

NO	Tool/Technique	Description
1	Python 3.9	This version provides a good balance of features, performance, and community support. They are widely adopted and have good compatibility with the latest versions of PyTorch library.
2	PyTorch	open-source machine learning framework for developing and training deep learning models, including CNNs for deepfake detection.
3	PyTorch Lightning	a higher-level framework that simplifies training code and handles many routine tasks such as training loops, checkpointing, and logging.
4	OpenCV	A computer vision library that can be used for tasks like face detection, preprocessing, and feature extraction, which are essential for building CNN-based deepfake detection models.
5	Scikit-learn	A machine learning library in Python that provides tools for model evaluation, data preprocessing, and other utilities that can be integrated with CNN-based deepfake detection models.
6	Jupyter Notebook	An interactive web-based notebook environment that allows for the development, testing, and sharing of CNN-based deepfake detection models, as well as the visualization of results
Techniques Used		
7	Data Augmentation	Techniques like image flipping, rotation, scaling, and noise addition that can be used to artificially expand the training dataset and improve the generalization of CNN-based deepfake detection models.
10	Transfer Learning	The technique of using a pre-trained CNN model, such as VGG-19, ResNet, or Inception, as a starting point for building a deepfake detection model, which can improve performance with limited training data.

Table 1 – Implementation Tools and Techniques

1.7.4. Evaluation Methods

The developed deepfake detection system is thoroughly evaluated using a comprehensive suite of performance metrics to assess its effectiveness in identifying both authentic and manipulated images. These metrics includes accuracy, precision, recall, F1-score, and area under the receiver operating characteristic (ROC) curve. The accuracy metric provides an overall assessment of the model's ability to correctly classify images into authentic and manipulated, while precision and recall evaluate its ability to correctly identify manipulated images and detect all manipulated images, respectively. The F1-score, which computed harmonic mean of precision and recall, offers a balanced evaluation of the model's performance. Additionally, the area under the ROC curve provides insights into the model's ability to differentiate between authentic and manipulated images across different decision thresholds. The model's performance is evaluated not only on the overall dataset but also on different types of deepfake generation techniques to assess its robustness in detecting a variety of manipulated visuals.

1.8.Significance of the Study

The outcomes of this research are highly relevant for tackling the growing issue of deepfake imagery and improving the trustworthiness of digital content. By leveraging AI-enhanced CNN models for detecting deepfakes, various groups such as media organizations, individuals, and platforms can enhance their ability to spot and address falsified images. This approach will help limit the spread of misinformation and safeguard the integrity of digital content. The findings from this study are expected to make a significant contribution to fields like image forensics, media credibility, and cybersecurity, by providing cutting-edge tools to detect and mitigate the impact of deepfake technology.

AI-enhanced CNN models offer a promising solution to the complexities of detecting altered images. These systems are poised to greatly improve the accuracy of distinguishing real images from fake ones. By strengthening the ability to identify manipulated content, this research aims to equip users with effective tools to fight the spread of doctored visuals. The widespread use of AI-enhanced CNNs could lead to stronger protections against the harmful effects of deepfakes, ensuring that the authenticity of visual media is maintained across various sectors.

Looking ahead, there are several potential areas for advancing deepfake detection methods. One key area for improvement is the expansion and diversification of datasets, incorporating a wider range of sources from different parts of the world. This would help the models become more adaptable to detecting deepfakes in diverse environments. Additionally, exploring more sophisticated image preprocessing techniques could help improve detection accuracy by refining the quality of images before they are processed by deep learning models.

By integrating advanced CNN models with innovative AI techniques, this research seeks to improve detection accuracy and efficiency, providing a reliable tool for identifying manipulated images. Additionally, this study aims to establish a foundation for future work in the field by addressing critical gaps in current detection methodologies and encouraging the exploration of alternative architectures and datasets.

1.9.Organization of the thesis

This thesis is structured into five main chapters: Introduction, Literature Review, Methodology, Design/Implementation/Experimental Results/Discussions, and Conclusions and Future Directions.

The first chapter introduces the thesis, providing a detailed background on the development and challenges associated with deepfake technology. It outlines the specific research problems addressed in this study, clarifies the goals of the research, and emphasizes the broader relevance of the work in addressing the increasing prevalence of deepfake content. Additionally, this chapter defines the scope and limitations of the study, specifying the boundaries within which the research operates. The chapter also includes an overview of the thesis structure, helping readers navigate through the detailed exploration of deepfake detection using AI-powered CNN models.

In the second chapter, a comprehensive review of existing literature and related studies is presented. It covers the key developments in deepfake technology, AI-driven CNN models, and the methods employed to detect deepfakes. The chapter provides an analysis of the current state of research, highlighting the major trends, challenges, and significant advancements in the field. By reviewing and synthesizing the available literature, this chapter offers a deeper understanding of the landscape surrounding deepfake technology and the strategies evolving to mitigate its harmful impact.

Chapter three details the research methodology, describing the data collection process, image preprocessing methods for both real and manipulated images, the training procedures for AI-enhanced CNN models, and the evaluation metrics used to assess model performance. This chapter also examines the specific AI techniques integrated into the CNN architectures, outlining the technical foundations that support the deepfake detection approach used in the study.

The fourth chapter presents the design, implementation, and experimental results of the AI-enhanced CNN model developed for deepfake detection. It includes a discussion on the model’s performance, analysis of the experimental outcomes, and the broader implications of these findings in the context of deepfake detection.

The fifth and final chapter summarizes the conclusions drawn from the research findings and explores possible directions for future research. It reflects on the significance of the study, recaps the major conclusions, and suggests potential areas for further investigation and advancements in the detection of deepfakes using AI-enhanced CNN models.

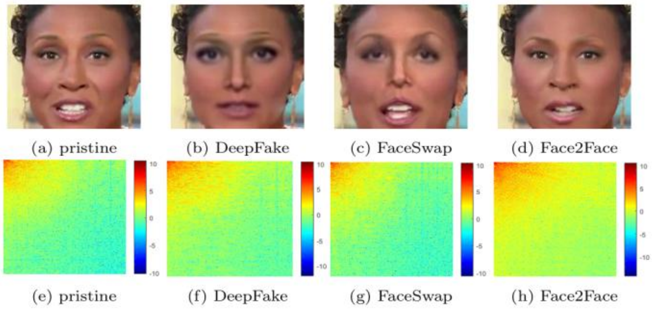
CHAPTER TWO
LITERATURE REVIEW AND RELATED WORKS

This chapter explores the use of AI-enhanced Convolutional Neural Networks (CNNs) for detecting deepfake images, harnessing the advanced image analysis abilities of CNNs to uncover subtle artifacts and irregularities. By analyzing existing research on the creation and detection of deepfakes, assessing the performance of various CNN architectures, and investigating the benefits of advanced techniques, this study seeks to improve the accuracy and reliability of deepfake detection systems. In doing so, it aims to contribute to the ongoing efforts to preserve the authenticity and integrity of digital media.

2.1. Deep Fakes

Deepfake is a technique for creating synthetic content by naturally changing the human face of the original content using an autoencoder and generative adversarial network (GAN).This technology leverages advanced neural networks that analyze extensive datasets to replicate a person's facial expressions, mannerisms, voice, and inflections[1], [2], [13].

The technology behind deepfakes primarily relies on neural networks, which learn from large sets of data to replicate a person's unique characteristics. At the core of this process are Generative Adversarial Networks (GANs), which consist of two neural networks: the generator and the discriminator [14]. The generator creates new, realistic samples, while the discriminator evaluates their authenticity, with this interplay enhancing the realism of the generated media. Additionally, facial mapping and AI are integral to the process, as they involve feeding footage of two individuals into a deep learning algorithm that trains to swap faces seamlessly, overlaying one person’s face onto another’s in a video.[13]



(a) pristine (b) DeepFake (c) FaceSwap (d) Face2Face

(e) pristine (f) DeepFake (g) FaceSwap (h) Face2Face

Figure 1 - Original video vs. Deep fake images [15]

2.1.1. The evolution of Deep Fakes

Deepfakes are a significant advancement in artificial intelligence and digital manipulation, utilizing deep learning techniques to create hyper-realistic content. Since the introduction of Generative Adversarial Networks (GANs) in 2014, research on deepfakes has surged, particularly between 2018 and 2021(See figure 2)[12]. Analysis of the SDO21 dataset reveals four main research trends: niche topics, which are well-developed but marginally significant; motor topics, crucial for the field's advancement; emerging topics that are gaining traction; and basic topics that remain important but underdeveloped.

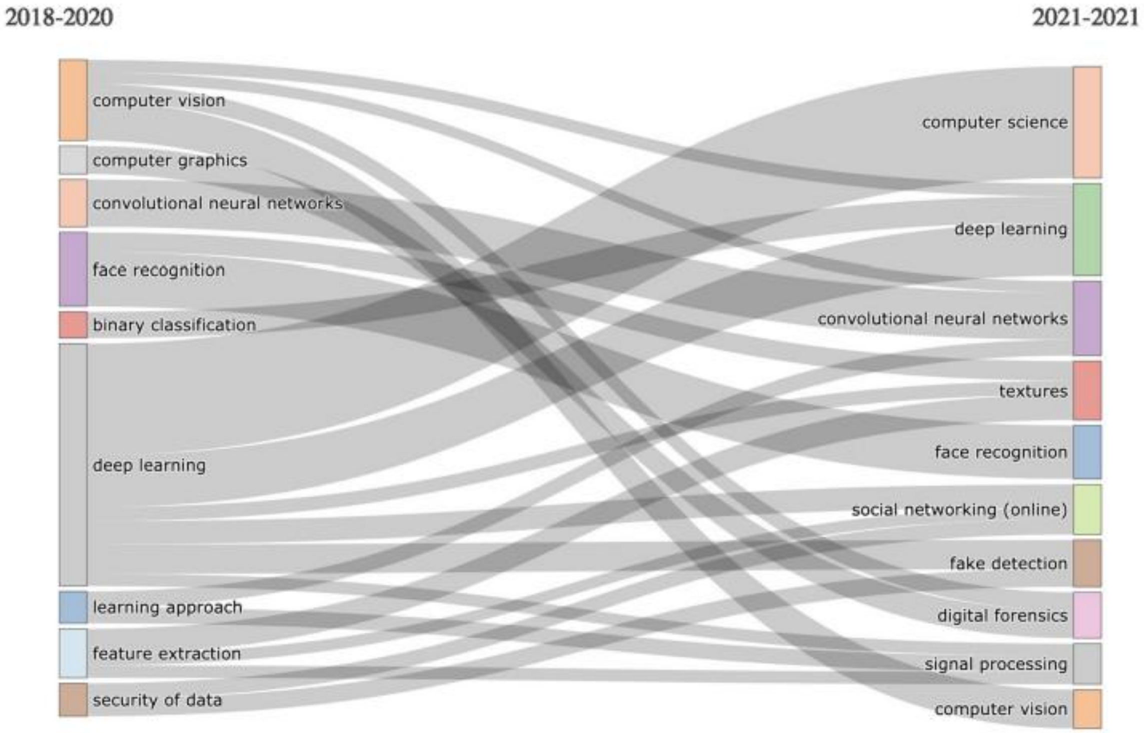


Figure 2- The evolution of research topics from 2018-2020 to 2021, highlighting shifts in focus areas such as deep learning, convolutional neural networks, and face recognition [12]

The thematic evolution of deepfakes illustrates how research areas have diversified over time. For instance, the prominent topic of computer vision has split into specialized subtopics, reflecting the field's growing complexity. Technologies such as Convolutional Neural Networks (CNNs) play a dual role in both creating and detecting deepfakes, highlighting an ongoing need for sophisticated detection methods as the technology advances[12], [13].

The societal implications of deepfakes are profound, raising ethical concerns about privacy and misinformation. The ability to fabricate realistic content can undermine trust in media and lead to misuse in various contexts, such as politics and personal relationships. Moreover, the proliferation of deepfakes presents legal and security challenges, creating a continuous arms race between creators and detectors in an effort to mitigate malicious uses of this technology [12].

2.2. Deep Fake Generation

The creation of deepfakes involves the use of advanced AI techniques, especially deep learning models such as Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs), to generate synthetic content, including highly realistic images, videos, and audio. Common techniques in deepfake generation include face swapping, where one person's face in a video is replaced with another's, lip-syncing, which adjusts mouth movements to align with different audio, and puppet-mastery (or face reenactment), where facial expressions and movements of one person are transferred to another's face. Additionally, face synthesis and attribute manipulation generate fake facial images and attributes, while audio deepfakes alter or imitate someone's voice. While these methods have valuable applications in areas such as entertainment and training, they also raise serious concerns regarding their potential misuse in producing misleading media [16].

2.2.1. Deepfake Generation Techniques

2.2.1.1. GAN-Based Deepfake Generation

GAN-based deepfake generation is an advanced image synthesis technique that utilizes Generative Adversarial Networks (GANs) to create highly realistic fake images. Unlike traditional image-to-image translation methods that require paired training data, deepfakes leverage unpaired datasets, making them powerful for tasks like style transfer and object transfiguration. In this process, two adversarial components—the generator and the discriminator—work against each other, with the generator creating fake images that the discriminator attempts to distinguish from real ones, thus improving the quality of generated images[17].

The Cycle-GAN, a specific type of GAN used in deepfake generation, includes two GAN networks to handle unpaired image-to-image translation. This setup includes a cycle-consistency loss to ensure that the transformation from one domain to another and back again maintains the original image features, resulting in realistic outputs. The Cycle-GAN's ability to learn these transformations has been effectively demonstrated in tasks like converting photographs into Van Gogh-style paintings or transforming between images of handbags and backpacks, showing the versatility and power of GAN-based deepfake technologies [14].

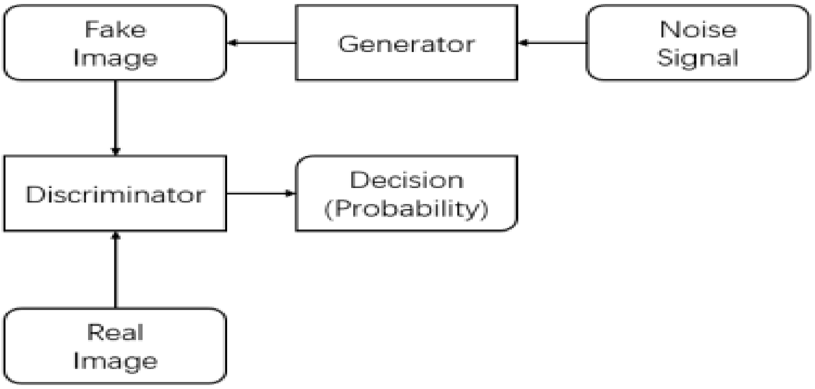


Figure 3 - A block diagram of GAN [14].

A Generative Adversarial Network (GAN) consists of two primary components: the generator (G) and the discriminator (D) [14]. The generator's task is to create realistic images G(z) from random noise z, while the discriminator's role is to evaluate whether an image is real or synthetic. During the training process, the generator attempts to minimize the likelihood that the discriminator will identify its generated images as fake, while the discriminator works to improve its ability to distinguish between real and fake images. This adversarial process is represented by the following minimax value function:

$$\min_{G} \max_{D} V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$

In this equation, V(D,G) is the value function that measures the interplay between the generator and the discriminator. The term $E_{x\sim pdata(x)}$[logD(x)] represents the expected value across the distribution of real data $_{pdata}$(x), where D(x) is the discriminator's probability estimate that x is a real image. The generator, on the other hand, aims to create samples that will deceive the discriminator.

Over time, through this iterative process, the generator becomes proficient at producing highly convincing images, while the discriminator enhances its ability to accurately classify real and fake images[14], [18].

2.2.1.2. Autoencoder-based Deepfakes

Autoencoders, initially developed in 2017 and later implemented as user-friendly applications like FakeApp, are fundamental models for generating deepfake content[18]. Traditionally, they have been employed for tasks such as dimensionality reduction, image compression, and learning generative models. Their ability to create compact representations of images while minimizing reconstruction loss makes them superior to other image compression methods. This feature of autoencoders enables them to effectively learn compressed representations of images, leading to their adoption as the primary model for face-swapping methods in deepfake generation. Mathematically, an autoencoder consists of an encoder function E: $R^m \rightarrow R^l$ and a decoder function D : $R^l \rightarrow R^m$, which aim to minimize the reconstruction loss between the original and reconstructed images, as represented in the equation[14]:

$$\arg\min_{E,D} \mathbb{E}[e^2(x, (D \circ E)(x))]$$

Where e (x, y) represents the error between the input xxx and the output of the composition of the encoder E and decoder D, denoted as (D∘E) (x), the objective is to minimize the expected squared error, E[e2(x ,(D∘E)(x)], over the distribution of x.[18]

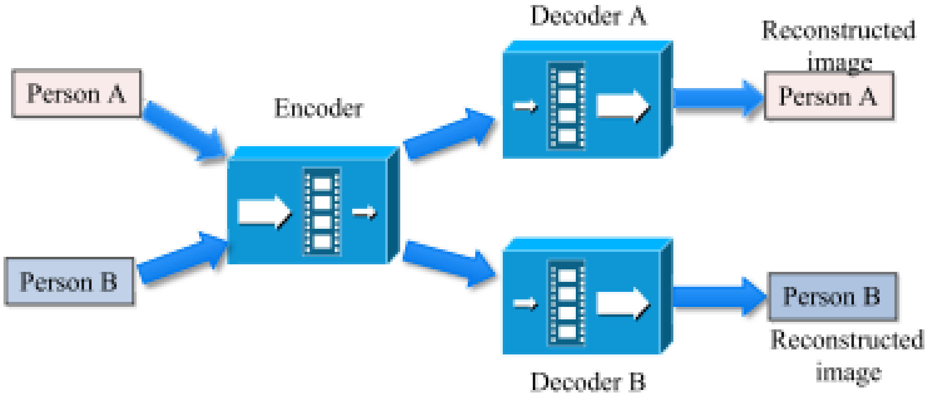


Figure 4 - Working principle of autoencoders [19]

The operation of an autoencoder involves three main stages: encoding, representation in the latent space, and decoding. During the encoding phase, the input image is compressed, capturing essential features like skin tone, texture, facial expressions, and structural details. This compressed representation is passed to the latent space, where patterns and relationships between data points are learned. In the decoding phase, the system reconstructs the original image as accurately as possible using the information stored in the latent space. For deepfake generation, the process involves training two separate autoencoders for two different faces, with a shared encoder and distinct decoders. To generate a deepfake, the image of Person A is encoded and then decoded using Person B's decoder, creating an image of Person B with the facial features of Person A. This methodology forms the basis for various deepfake technologies, including DFaker, DeepFaceLab, and TensorFlow-based tools. The same principle applies to generating deepfake videos, where faces are swapped frame by frame to produce realistic results [18].

2.2.2. Types of Face Manipulation in Deepfake

There are different types of face manipulation in Deepfake. Hereunder a brief description of each is given.

Face Swap

Face-swapping is a type of visual manipulation where the face in a source image or video is replaced with a target face. Traditional face-swap techniques typically follow three steps: face detection, blending the source and target face features, and adjusting lighting and color for a seamless transition. However, these methods often produce rigid and unnatural results as they do not preserve facial expressions.[16]

With the rise of deep learning, modern face-swap methods using neural networks, particularly autoencoder-decoder pairs, have significantly improved the realism of deepfakes. These methods extract and reconstruct facial features from source and target images, allowing for seamless face swaps while maintaining facial expressions. Popular applications such as FakeApp, ZAO, and REFACE have made this technology accessible to general users, allowing them to embed faces into various media content[7].

Advanced approaches using Generative Adversarial Networks (GANs) have further enhanced face-swap techniques, producing more realistic results. For instance, FSGAN enables real-time face-swapping and reenactment by manipulating pose, expression, and identity in a coherent manner. Other methods like FaceShifter use adaptive attention layers to handle facial occlusions, preserving target attributes such as pose, lighting, and expression. Despite the advancements, challenges remain, particularly in handling occlusions and creating deepfakes that are indistinguishable from reality.[7], [16]

Lip-syncing

Lip-syncing deepfakes involve generating videos where the lip movements of a target person match an arbitrary audio input. This technique focuses on synthesizing realistic mouth movements and expressions to create visually coherent and convincing speech. Lip-syncing has practical applications in entertainment, such as dubbing films, creating digital characters, and generating content for hearing-impaired audiences through lip-reading.[16]

Traditional methods often rely on frame reselection and transcription-based approaches, which are limited in their ability to generalize to new faces or emotional states. However, deep learning models have significantly advanced this field. Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) can now predict mouth shapes and generate natural-looking videos. For example, models like Speech2Vid and LipGAN use audio features to drive lip movements, and can produce photorealistic videos directly from still images or video clips. More recent approaches like Wav2Lip improve synchronization accuracy by using pre-trained lip-sync discriminators and temporal correlation, resulting in more natural facial expressions and better video quality. These advanced models enable the creation of highly realistic lip-synced videos with improved audio-visual alignment.[16], [20]

Puppet-Master

Puppet-master, or face reenactment, is a deepfake technique that allows the manipulation of a person's facial expressions by transferring gestures, head movements, and eye motions from a source actor to a target. This method has practical applications in areas like dubbing films, altering facial expressions in video conferences, and creating photorealistic animations for movies and games. Early approaches relied on 3D facial modeling to accurately capture geometry and facial movements. Thies et al. [4]introduced one of the first real-time systems, using 3D face models tracked with RGB-D sensors to transfer facial expressions, while Face2Face extended this concept to RGB video streams, making it feasible with basic webcams.

Generative Adversarial Networks (GANs) have significantly advanced face reenactment, enabling highly realistic image synthesis. Methods like Pix2pixHD and ReenactGAN improved image fidelity and expression transfer by using multi-scale GAN architectures and latent feature spaces to encode facial details. Some approaches, like GANimation and GANnotation, incorporated emotion-specific action units and facial landmarks to create more controlled, natural expressions. However, these methods often required extensive training data for each target identity and struggled with varying facial angles and large pose changes [16].

More recent innovations in few-shot and one-shot learning, such as X2face and MarioNETte, enable face reenactment using just a few images or even a single image of the target. These models employ self-supervised learning and advanced attention mechanisms to transfer poses and expressions more effectively while preserving the target's identity. Approaches like FSGAN have also improved real-time capabilities, making it possible to generate smooth reenactment at high frame rates. As these techniques evolve, face reenactment is expanding beyond facial expressions to include full-body reenactment, transferring more complex movements like head gestures and body postures in real-time.

Face Synthesis and Attribute Editing

The advancements in facial synthesis and attribute modification technologies have introduced transformative capabilities across various domains, including art, animation, and the entertainment industry. These technologies facilitate the creation of lifelike human faces that do not exist in reality, offering significant benefits for applications such as video game character design and 3D modeling. Central to these innovations are generative models like Generative Adversarial Networks (GANs) and Variational Autoencoders (VAEs). Over time, GAN architectures have evolved remarkably, with models such as StyleGAN and StyleGAN2 producing high-resolution, photorealistic facial images. This progress has enhanced the realism of synthetic faces, widening the scope of their applications—both positive and potentially harmful [16].

Attribute editing, a critical subset of facial synthesis, involves modifying specific features of a face while maintaining its overall identity. Early approaches, such as Invertible Conditional GAN (IcGAN) and Fader Networks, enabled changes to attributes like age, gender, and expressions. However, these methods often faced issues with preserving finer image details, leading to visual inconsistencies or blurring. More advanced models like StarGAN and its successor, StarGAN-v2, have addressed these shortcomings by allowing multiple attribute transformations within a single framework. These innovations have significantly improved the coherence and visual quality of edited images while reducing unwanted artifacts [16], [17].

Despite the advancements, achieving flawless attribute manipulation that preserves identity and intricate details remains a challenge. Approaches like AttGAN and STGAN have introduced enhanced architectures and constraints to better handle attribute relationships within the latent space. Nevertheless, problems such as detail loss and artifact generation persist. As the field continues to advance, it is vital to weigh the advantages of these technologies against ethical considerations. Issues related to privacy, misinformation, and digital identity require ongoing attention to ensure responsible use. Future research must focus on refining methodologies to achieve high-quality results while mitigating the risks of misuse [16].

2.3. Approaches of Deepfake Detection

Nowadays Deep learning has revolutionized deepfake detection, with CNNs playing a crucial role. CNNs automatically learn relevant features from video data, improving accuracy and robustness [19]. In general, Deepfake detection approaches can be classified into two, such as traditional and deep learning approaches.

2.3.1. Traditional Approaches

Early approaches to detecting deepfakes predominantly utilized traditional machine learning techniques. These methods generally focused on analyzing individual video frames without considering the sequential nature of video data. By isolating frame-level features, these approaches faced limitations in identifying temporal patterns and inconsistencies. Furthermore, the effects of lossy video compression, which often degrade the quality of frames, posed significant challenges to distinguishing authentic content from manipulated material. Consequently, these techniques exhibited reduced accuracy, particularly in video scenarios where temporal coherence and variable frame quality are crucial factors [16], [19].

Initial detection systems also relied on traditional machine learning algorithms such as Support Vector Machines (SVMs) and Decision Trees. These methods required extensive manual feature extraction to identify patterns within the data. However, their performance was hindered by difficulties in generalization and achieving high accuracy, especially when applied to diverse datasets with varying characteristics [12], [16], [19].

2.3.2. Deep Learning Approaches

With the advent of deep learning, detection techniques have evolved significantly. Many contemporary approaches leverage convolutional neural networks (CNNs) to extract frame-level features, combined with recurrent neural networks (RNNs) like long short-term memory (LSTM) networks to capture temporal sequences. For example, Guera and Delp [17] proposed a hybrid architecture that employs CNNs to extract intra-frame features and LSTMs to analyze the temporal relationships among frames. Their method successfully identified discrepancies in Deepfake videos by focusing on features such as eye blinking, which typically occur more frequently in genuine videos than in manipulated ones [18], [19].

Another notable technique is the use of recurrent convolutional networks (RCNs), which take advantage of spatiotemporal features in video data. These networks enable more effective detection by analyzing both the spatial and temporal dimensions of video content. Furthermore, ensemble learning strategies, such as the DeepfakeStack proposed by Rana and Sung [20], integrate multiple deep learning models to enhance detection accuracy. By training a meta-learner on the predictions of various base-learners, DeepfakeStack achieved an impressive accuracy of 99.65%, demonstrating the potential for combining deep learning methods to improve detection outcomes [19].

Deep learning has revolutionized deepfake detection, with CNNs playing a crucial role. CNNs automatically learn relevant features from video data, improving accuracy and robustness [19]. Notable deep learning models include the following [19]:

- **XceptionNet**: Utilizes depthwise separable convolutions for efficient learning and high classification accuracy.
- **FaceForensics++**: Provides a benchmark dataset for training deep learning architectures, enhancing model performance.
- **Multi-Modal Models**: Integrate audio and visual data, leading to improved detection capabilities.

2.4. Steps of Deepfake Detection

The process of Deepfake detection involves several key steps (as shown in figure 5 below that enhance the robustness of the detection system [19]:

1. **Data Collection**: The initial step is the compilation of datasets, such as the FF++ dataset, which contains a diverse range of Deepfake videos. However, inconsistencies in dataset size and composition across studies can impact the generalizability of findings.

2. **Feature Extraction**: The next step involves extracting relevant features from the videos. Traditional methods focused solely on frame-level analysis, but current approaches utilize CNNs and LSTMs to capture both spatial and temporal features. This includes identifying physical indicators, such as eye blinking patterns, that can signal manipulation.

3. **Model Training**: Once features are extracted, deep learning models are trained on labeled datasets. Many studies have indicated that deep learning-based models outperform traditional machine learning methods, leading to a preference for deep learning techniques in recent research.

4. **Model Evaluation**: After training, models are evaluated using various metrics, including accuracy and area under the receiver operating characteristic curve (AUROC). However, a lack of standardization in measurement metrics among studies can diminish the reliability of comparisons.

5. **Framework Development**: Given the limitations identified in existing literature, establishing a unique framework for evaluating Deepfake detection methods is essential. This framework should address the inconsistencies in datasets, measurement metrics, and experimental procedures to enhance the credibility of research outcomes.

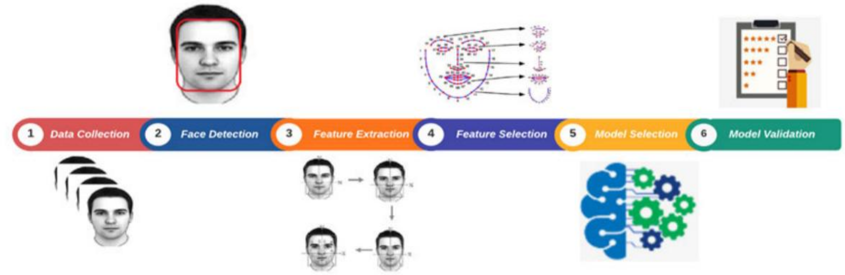


Figure 5 - Steps of Deepfake detection [19].

2.5. Deepfake Detection Models

Deepfake detection has evolved with advancements in machine learning, transitioning from traditional methods to deep learning approaches, particularly Convolutional Neural Networks (CNNs)[19].

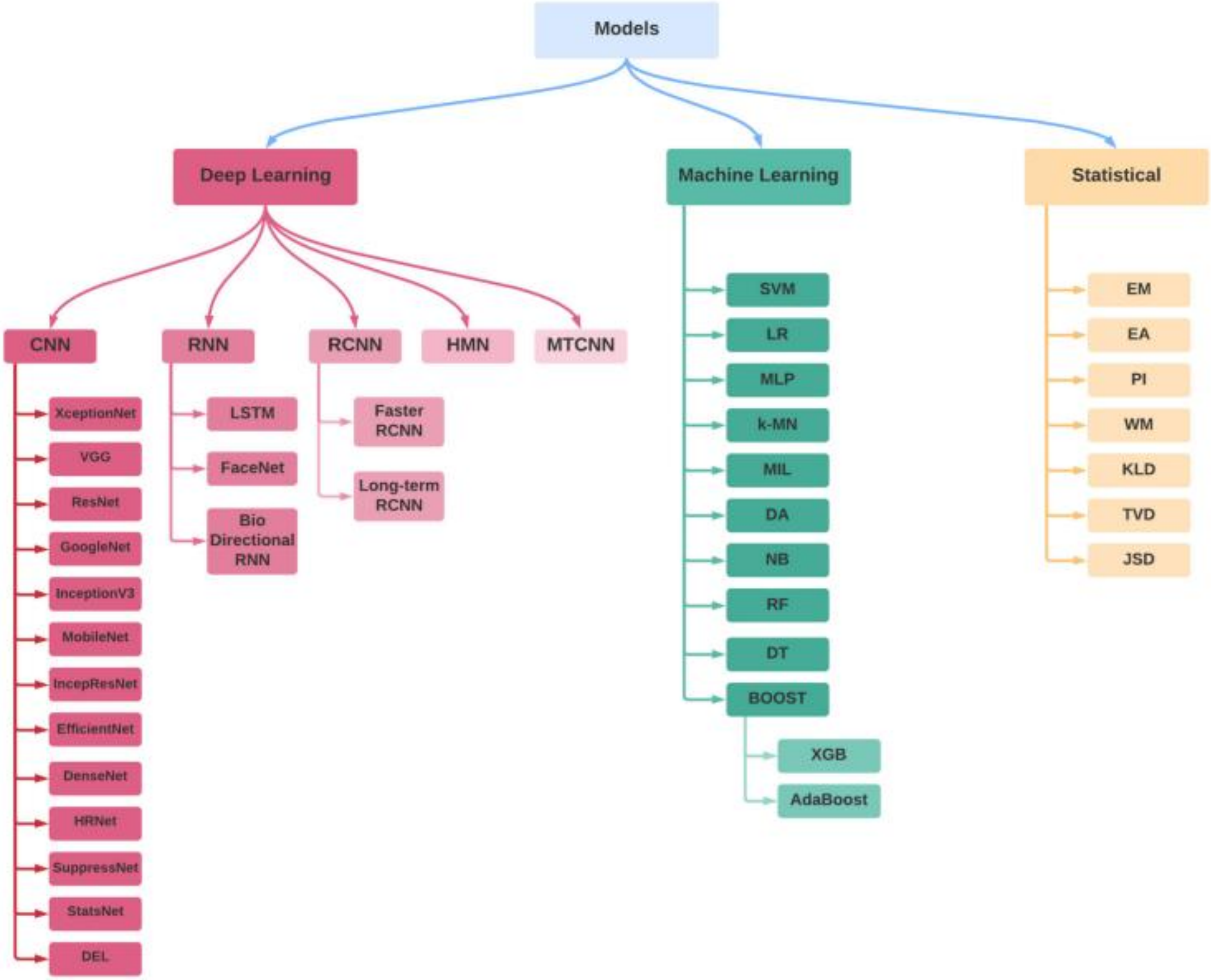


Figure 6 - The list of Deepfake detection models [19].

2.6. Importance of Using CNNs in Deepfake Detection

Patel et al. [3], highlights several key advantages of applying Convolutional Neural Networks (CNNs) for detecting deepfake media:

1. Automated Pattern Recognition: CNNs are highly effective at identifying subtle features within video frames, such as unnatural facial movements or inconsistencies in expressions. Their ability to apply convolutional filters allows them to detect manipulation without relying on manual feature extraction.
2. Integration of Time-Based Analysis: When combined with sequential models like Recurrent Neural Networks (RNNs) or Long Short-Term Memory (LSTM) networks, CNNs can analyze changes and correlations across multiple video frames. This enables a deeper understanding of temporal anomalies, significantly enhancing detection accuracy.
3. Superior Accuracy Levels: Research demonstrates that CNN-based methods achieve much higher performance compared to older machine learning algorithms. For instance, the DeepfakeStack model created by Rana and Sung achieved remarkable accuracy rates, exceeding 99%, which highlights the reliability of CNNs in classification tasks.
4. Customization and Transfer Learning: CNNs can integrate pre-trained models to fine-tune performance for specific datasets, such as FF++, which includes diverse deepfake videos. This capability ensures the model adapts effectively to varying data requirements.
5. Streamlined Learning Process: Unlike traditional approaches requiring separate steps for feature extraction and prediction, CNNs provide an end-to-end learning framework. This simplifies the detection process, reduces manual effort, and allows researchers to concentrate on refining the model itself.

2.7. Related works

Jiameng Pu et al. [10], introduced an innovative deepfake detection framework that addresses the limitations of prior methods by focusing on GAN-generated images without requiring specific training on such data. Their "NoiseScope" technique leverages distinct noise patterns unique to GAN outputs, achieving a remarkable F1 score of 99.68% across various datasets. This method showcases a robust approach, balancing qualitative and quantitative analyses while addressing potential countermeasures to strengthen its effectiveness. Despite its promising results, the paper could benefit from a deeper exploration of the computational demands when scaling this technique for widespread application in media verification systems.

Malik, Asad, et al[6], proposed a cutting-edge detection method utilizing the Expectation-Maximization (EM) algorithm to extract convolutional traces left by GANs. Their approach demonstrated exceptional accuracy rates, reaching 99.81% across classifiers like K-Nearest Neighbor (K-NN) and Support Vector Machine (SVM). The study emphasized the challenges of dataset diversity and highlighted the need for broader testing environments to validate their methodology's scalability and practical application in digital forensics. While the results are compelling, expanding dataset coverage and providing further insights into the computational feasibility of this method would strengthen its utility.

Patel, Y., et al.[3] , presented an enhanced Dense CNN architecture tailored for deepfake detection. By addressing limitations in earlier models, such as MesoNet and MesoInception, the researchers introduced data augmentation techniques (e.g., rotation, flipping, and scaling) to ensure robustness across varying resolutions and data sources. Their model achieved a high accuracy rate of 97.2% on a diverse dataset and performed well even with imbalanced data. This study sets a strong precedent for future advancements, though further exploration into cross-dataset validation would bolster its real-world applicability.

Abhinav Gupta, et al. [15], highlighted the importance of frequency-based CNNs (fCNNs) in deepfake detection, focusing on features in the frequency domain to differentiate real and manipulated images. Evaluated on the FaceForensics++ dataset, the fCNN achieved an accuracy of 78.3% across multiple forgery types, including DeepFake, Face2Face, and FaceSwap. While the model excels in high-resolution scenarios, it faces challenges in detecting neural texture manipulations, suggesting opportunities for improvement in handling nuanced forgery techniques.

Their study underscores the value of frequency-domain methodologies, paving the way for more refined and adaptive systems.

Zhang, Yi, et al. [21] examined the evolution of deepfake generation methods, focusing on GANs and VAEs. They categorized deepfake techniques into face transfer, face swap, face reenactment, and face editing, emphasizing the need for robust detection mechanisms. Their experiments on the ForgeryNet dataset revealed EfficientNetV2-M's superior accuracy of 81.1% for specific methods, while ViT-Base exhibited consistent generalization across diverse approaches. These findings highlight the potential of Vision Transformers in real-world applications, given their ability to balance performance across multiple manipulation types.

Kumar,Shinde, and Verma[4], explored the capabilities of face recognition models in detecting deepfakes, particularly those employing identity-swapping techniques. Their study demonstrated that models trained with advanced loss functions like Combined Margin and CosFace outperformed the CNN-based Face-Xray model by a significant margin, achieving superior results on datasets such as Celeb-DF and FaceForensics++. However, expression-swapping manipulations, which alter facial movements while retaining identity features, remain a challenge. This research underscores the need for novel methodologies to detect such subtle manipulations effectively.

Abir, Wahidul Hasan, et al.[22], proposed leveraging Explainable AI (XAI) for deepfake detection, incorporating the LIME algorithm to enhance interpretability in CNN-based models. Their approach achieved an impressive accuracy of 99.87% for detecting real versus fake images. However, the study identifies gaps in its application to video and audio deepfakes, as well as limited evaluation of user understanding of the model's predictions. Further exploration into XAI techniques and dataset quality would strengthen the development of robust, user-friendly deepfake detection systems.

Hereunder Table 2.1 presents summary of related works done on deepfake detection from real images.

Author(s) and Year	Techniques Used	Application (Focus Area)	Dataset	Results
[Jiameng Pu et al., 2021]	NoiseScope detection framework, GAN noise pattern analysis	Detection of GAN-generated deepfake images	Multiple GAN-generated datasets	F1 Score of 99.68% across datasets
[Malik et al., 2020]	Expectation-Maximization (EM) algorithm, K-NN and SVM classifiers	Detecting convolutional traces in GAN images	Custom deepfake dataset	Accuracy of 99.81% using SVM and K-NN
[Patel et al., 2019]	Dense CNN architecture with extensive data augmentation	Detection and classification of deepfake images	Diverse real and deepfake images	97.2% Accuracy on diverse dataset
[Gupta et al., 2021]	Frequency-based CNN (fCNN), activation map visualizations	Detecting facial forgery in high-quality videos	FaceForensics++	78.3% detection accuracy
[Zhang et al., 2022]	EfficientNetV2-M and Vision Transformer (ViT)	Detecting deepfakes across multiple methods	ForgeryNet, Deepfakes	EfficientNetV2-M: 81.1% accuracy, ViT: 62.0% with lower variance

Table 2.1 Summary of related works

2.7.1. Research gap

Based on related works review, the following research gaps are identified for further investigation and advancement of deepfake detection.

- ✓ Limited Focus on Non-Visual Deepfakes: Most existing research, including work by Jiameng Pu et al. [10] and Patel et al. [3] has concentrated on detecting visual (image-based) deepfakes. However, deepfake technology extends beyond images and videos into audio and text-based manipulations. There is a need for research that addresses multi-modal deepfake detection, including audio-visual and text-based fakes.
- ✓ Generalization Across Deepfake Generation Methods: Studies like Zhang et al. show that different detection models, such as EfficientNetV2 and Vision Transformers, perform

inconsistently across different deepfake generation techniques (FaceSwap, FaceReenactment). These models often specialize in detecting specific types of deepfakes, leading to lower generalizability across new or unseen methods. There is a gap in developing models that can generalize well across a broader range of deepfake generation techniques without overfitting to specific datasets.

- ✓ Scarcity of Large-Scale and Diverse Datasets: Many studies, such as Malik et al. [6] rely on custom or limited datasets, often focused on specific types of manipulations. The datasets lack diversity in terms of image quality, demographic variability, and manipulation techniques. More comprehensive datasets are needed, with a wider range of manipulated content from diverse sources, to ensure models are robust and applicable to real-world scenarios.
- ✓ Integration of Explainability in Detection Models: While Abir et al. [22] introduced explainable AI (XAI) techniques like LIME to improve interpretability, there is limited research on the systematic evaluation of different explainability methods in deepfake detection. A gap exists in evaluating how explainability can improve trust in AI models, especially for real-world deployment in critical applications like media verification and digital forensics.
- ✓ Handling High-Quality and Real-Time Deepfakes: Although models like the fCNN by Gupta et al. [15]perform well in detecting high-quality forgeries, their detection accuracy still suffers with high-resolution, real-time deepfakes, particularly those using neural texture manipulations. Further research is needed to develop real-time, high-accuracy detection models that can handle high-quality, high-resolution manipulations in both video and still images.
- ✓ Practical Implementation Challenges and Clinical Integration: Despite the promising results of CNN-based models, few studies address the practical implementation challenges in large-scale, real-world environments. The studies often neglect the computational costs, scalability, and regulatory challenges of deploying these models in professional environments such as digital forensics or media verification agencies.
- ✓ Bias and Fairness in Detection Models: Research shows that deepfake detection models, particularly CNNs, often suffer from performance biases when applied across different demographic groups. There is limited exploration of fairness in detection algorithms,

leading to potential ethical concerns when these systems are deployed. There is a gap in developing unbiased detection models that perform equitably across diverse populations.

This study focuses on two critical gaps in deepfake detection research: the scarcity of large-scale and diverse datasets and the challenge of generalization across deepfake generation methods. Existing datasets often lack diversity in image quality, demographic representation, and manipulation techniques, limiting the robustness and real-world applicability of detection models. Additionally, current models, such as EfficientNetV2 and Vision Transformers, tend to specialize in detecting specific types of deepfakes, struggling to generalize effectively to new or unseen techniques. By addressing these gaps, this study aims to enhance the adaptability and reliability of detection systems, laying the groundwork for more robust and scalable solutions to combat evolving deepfake technologies.

CHAPTER THREE

PROPOSED ARCHITECTURE AND METHODS

3.1.Overview

This chapter outlines the conceptual architecture and system structure developed for detecting deepfake images using Convolutional Neural Networks (CNNs) enhanced with artificial intelligence. The architecture is crafted to recognize specific patterns and features unique to deepfake images, enabling it to reliably distinguish between manipulated and authentic content. A system flowchart provides an overview of the process, showing how the CNN-based model ingests input images, extracts relevant features, and classifies them as real or fake. The selection of CNNs for this task stems from their proven capability to identify subtle irregularities and distortions in images, which are often imperceptible to the human eye.

3.2.Proposed Architecture

The complex nature of deepfake manipulation necessitates the use of CNNs for their efficiency in automated feature extraction and high accuracy across diverse datasets. The system begins by processing an input image through successive network layers, each of which extracts increasingly sophisticated features. Ultimately, the system outputs a classification label indicating whether the input image is authentic or manipulated. Key aspects of the system, such as data preprocessing steps, model architecture, and evaluation metrics, are described in detail. This system has the potential to significantly aid in identifying deepfakes, thereby supporting efforts to reduce misinformation and enhance digital content security for various stakeholders.

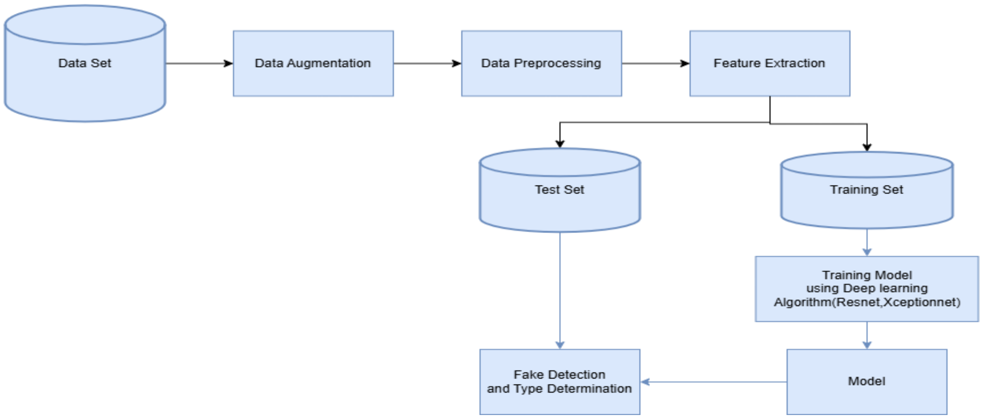


Figure 2. Proposed architecture

3.3.Data Collection

The FaceForensics++ dataset serves as the primary resource for training and testing the detection model. This dataset includes both unaltered and manipulated video sequences. The unaltered sequences consist of genuine videos featuring individuals performing various actions and expressions, providing a reliable baseline for comparison.

The manipulated content in the dataset has been generated using several deepfake techniques, including:

- ✓ **Deepfakes:** Techniques that use autoencoder-based models to modify or replace facial expressions.
- ✓ **FaceSwap:** A method that involves swapping faces between two different individuals within a video.
- ✓ **Face2Face:** A real-time reenactment method that adjusts the expressions in a target video to mimic those of a source individual.
- ✓ **NeuralTextures:** Techniques that apply artificial textures or subtly alter facial features, making them more challenging to detect visually.

3.4.Preprocessing

Preparing the dataset for model training is a crucial step in building an effective deepfake detection system. The preprocessing pipeline consists of the following steps:

1. **Extracting Frames:** Videos are converted into individual frames to isolate static images. These frames, representing both original and manipulated content, form the primary input for the CNN model.

2. **Resizing Images:** All extracted frames are resized to 224x224 pixels to ensure consistency in input dimensions across the dataset. Standardizing image size is essential for effective training.

3. **Normalizing Pixel Values:** Pixel intensities are scaled to fall within a range of [0, 1] by dividing each value by 255. Normalization ensures compatibility with the model and facilitates faster and more stable training.

4. **Data Augmentation:** Various augmentation techniques, such as flipping, rotation, scaling, and adjustments to brightness or contrast, are applied to diversify the dataset. These enhancements help the model generalize better and resist overfitting.

5. **Assigning Labels:** Frames derived from authentic videos are tagged as "real," while those from manipulated sequences, such as Deepfakes, FaceSwap, and NeuralTextures, are labeled as "fake." These labels act as ground truth during model training.

6. **Splitting the Dataset:** To train and evaluate the model effectively, the dataset is divided into training, validation, and testing subsets. Typically, 70% of the data is used for training, 15% for validation, and 15% for testing. This division ensures that the model learns from diverse examples and is assessed on unseen data for accuracy. Additionally this **split** ensures sufficient data for effective training, robust validation for tuning and generalization, and unbiased testing for reliable performance evaluation.

By following this systematic data preparation approach, the FaceForensics++ dataset is transformed into a suitable format for use with the proposed CNN model. This rigorous preprocessing ensures that the model is exposed to a variety of authentic and manipulated images, enhancing its ability to detect deepfakes with high reliability.

3.5.Modeling

The detection system is built around the XceptionNet model, a specialized convolutional neural network is chosen for its strength in feature extraction. The system analyzes video data from the FaceForensics++ dataset, where video frames are systematically extracted and processed. Preprocessing includes resizing the frames, normalizing pixel values, and applying techniques to augment the data, all aimed at enhancing the model's accuracy and robustness.

Once the frames are prepared, they are passed through the XceptionNet model, which uses its unique architecture—including depthwise separable convolutions—to detect fine-grained details in facial features and textures. These intricate features help the system identify subtle manipulations in the images. The extracted features are then flattened into a single-dimensional vector, further processed through dense layers, and passed through a dropout mechanism to reduce overfitting. A softmax layer at the end classifies the images as either genuine or manipulated, specifying the type of manipulation (e.g., deepfake, FaceSwap, or NeuralTextures).

To further refine performance, transfer learning is applied by adapting a pre-trained XceptionNet model to the specific properties of the FaceForensics++ dataset. Incorporating attention mechanisms such as spatial and channel attention enables the model to focus on critical areas of the face, improving detection precision. Additionally, techniques like dropout, L2 regularization, and early stopping are used to strengthen the model’s generalization and reduce overfitting risks.

The system's workflow, as illustrated in Figure 8, outlines the step-by-step process for detecting manipulated images extracted from video data in the FaceForensics++ dataset. This process is described below:

1. **Extracting Frames:** Individual frames are taken from videos at consistent intervals. These frames represent either genuine content or various types of manipulated content, such as deepfakes, FaceSwaps, or NeuralTextures. The frames act as the primary dataset for training and testing.

2. **Data Preparation:** The extracted frames are resized to 299x299 pixels, matching the input dimensions required by the XceptionNet model. Pixel values are normalized to fall between 0 and 1 to standardize the dataset. To improve the system's ability to generalize, data augmentation is performed, introducing variations like rotations, flipping, and brightness adjustments.

3. **Feature Extraction:** The processed frames are fed into the XceptionNet model. Using its convolutional layers, the model extracts hierarchical features, identifying intricate details such as textures, patterns, and edges that distinguish genuine images from manipulated ones. Depthwise separable convolutions play a key role in capturing these fine details efficiently.

4. **Flattening and Regularization:** The hierarchical features obtained from convolutional layers are transformed into a one-dimensional vector. Dropout is applied at this stage to improve the system’s resilience against overfitting by randomly omitting neurons during the training phase.

5. **High-Level Feature Processing:** The flattened feature set is passed through dense layers that combine the extracted information to form a more comprehensive representation of the input image. These layers help uncover relationships between the features that indicate whether an image is authentic or fake.

6. **Final Classification:** The dense layer outputs are passed to a softmax classifier, which predicts whether an image is real or manipulated. Manipulated images are further classified into specific categories, including deepfake, FaceSwap, or NeuralTextures. The softmax layer assigns a probability to each class, with the highest probability determining the output.

7. **Model Training and Testing:** The model undergoes training using labeled datasets, where each frame is identified as real or manipulated. Training optimizes the categorical cross-entropy loss function with the Adam optimizer. Following this, the model is evaluated using metrics like precision, recall, accuracy, and F1-score to validate its effectiveness in detecting manipulations.

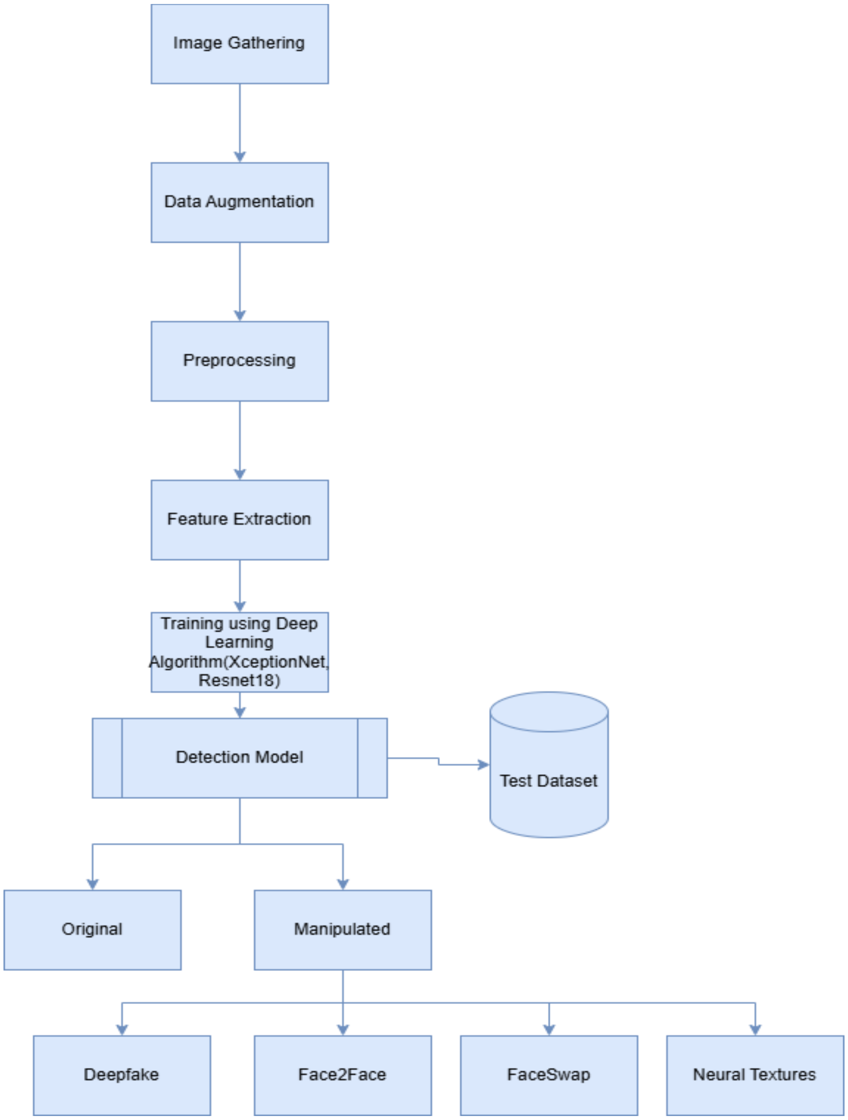
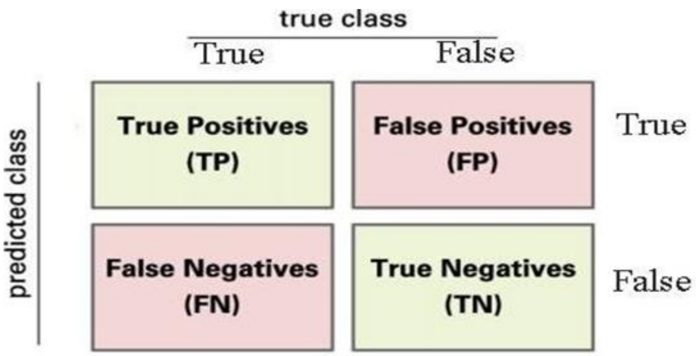


Figure 7- Workflow of the proposed deepfake detection model

3.6.Evaluation Methods

3.6.1. Confusion Matrix

The confusion matrix provides a detailed breakdown of classification outcomes into true positives, true negatives, false positives, and false negatives. This visualization offers deeper insights into where the model is excelling and where it might be prone to errors. For example, it can help identify if the model struggles more with certain categories of manipulated images, such as deepfakes or faceswap manipulations.



predicted class	true class: True	true class: False
True	True Positives (TP)	False Positives (FP)
False	False Negatives (FN)	True Negatives (TN)

Figure 8 - Confusion Matrix

3.6.2. Evaluation Metrics

There are difference model effectiveness measures used in this study, such as accuracy, recall, precision and F-score.

Accuracy

Accuracy is a fundamental evaluation metric that measures the proportion of correctly classified images over the total number of images. It provides a simple yet effective measure of the model's overall performance. The formula for accuracy is:

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

where TP represents true positives, TN true negatives, FP false positives, and FN false negatives. While accuracy is helpful for assessing performance, it may not fully reflect the model's effectiveness in imbalanced datasets where one class dominates.

Precision, Recall, and F1-Score Precision, recall, and F1-score are critical metrics that go beyond accuracy, particularly in scenarios where the cost of false positives or false negatives varies.

Precision

Precision measures the proportion of true positives among all images classified as manipulated. It is a key metric in scenarios where minimizing false positives is crucial. High precision ensures that most of the detected manipulated images are genuinely fake:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Recall

Recall (Sensitivity) evaluates the model's ability to identify all actual manipulated images. It measures the proportion of true positives out of all manipulated images in the dataset:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

F1-Score

F1-Score combines precision and recall into a single metric by calculating their harmonic mean. It balances the trade-off between false positives and false negatives, offering a comprehensive view of the model's performance:

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

This makes the F1-score particularly useful when the dataset has imbalanced classes or when equal importance is placed on precision and recall.

Area Under the Receiver Operating Characteristic (ROC-AUC)

The ROC-AUC metric evaluates the model's capacity to distinguish between real and manipulated images across different classification thresholds. The ROC curve plots the true positive rate (sensitivity) against the false positive rate, and the area under this curve quantifies the model's performance. A higher AUC indicates better discrimination capability, with an AUC of 1 representing perfect classification.

CHAPTER FOUR

EXPERMENTAL RESULT AND DISCUSSION

4.1. Overview

This chapter explores the application of Convolutional Neural Networks (CNNs) and transfer learning techniques for detecting Deepfakes. The investigation is presented through various scenarios, detailing the methodologies employed. Additionally, the chapter provides a comprehensive explanation of the hyperparameters utilized in the experiments.

4.2. Experimental Setup

Hardware

The experiments were conducted using a high-performance hardware setup that included an NVIDIA RTX 4050 GPU, 16GB of system RAM, and a 13th-generation Intel i5 processor. The NVIDIA RTX 4050, with its powerful parallel processing capabilities, significantly accelerated training and inference, making it ideal for handling the computational demands of the CNN models, which involves complex convolutional operations and large datasets. The 16GB RAM ensured smooth execution of memory-intensive tasks, while the 13th-gen i5 processor provided robust support for data handling and system-level operations, enabling efficient and seamless experimentation.

Software

The experimental setup for detecting Deepfakes utilized various software libraries to streamline data preprocessing, model training, evaluation, and visualization. PyTorch and its associated modules (torch, torch.nn, torch.optim, torch.utils.data) were employed for building, training, and optimizing deep learning models, with CUDA settings configured to enhance GPU performance. Torchvision was used for image transformations and dataset handling, while Timm provided access to pre-trained models such as Xception. Pandas facilitated data manipulation and CSV file handling, and Pillow (PIL) was used for image loading and processing. NumPy supported numerical computations, and Matplotlib was leveraged for visualizing training history and performance metrics. Additionally, Tqdm tracked progress during training and evaluation, Psutil monitored memory usage, and the OS and Garbage Collection (gc) libraries managed file paths and optimized memory usage. The Datetime library was also used for timestamping and tracking experiment runtime.

4.3. Dataset Preparation

The preparation of the dataset for deepfake detection involved several systematic steps to ensure high-quality, diverse, and well-structured data. The dataset initially consisted of 5,000 videos, equally divided between original (real) and manipulated (fake) content. The fake videos were further categorized into four distinct manipulation techniques: Deepfakes, Face2Face, FaceSwap, and NeuralTextures, each contributing 1,000 videos. From each video, 20 frames were uniformly extracted to create a balanced and representative set of images. This process resulted in an initial dataset of 100,000 images, evenly distributed among the five video categories.

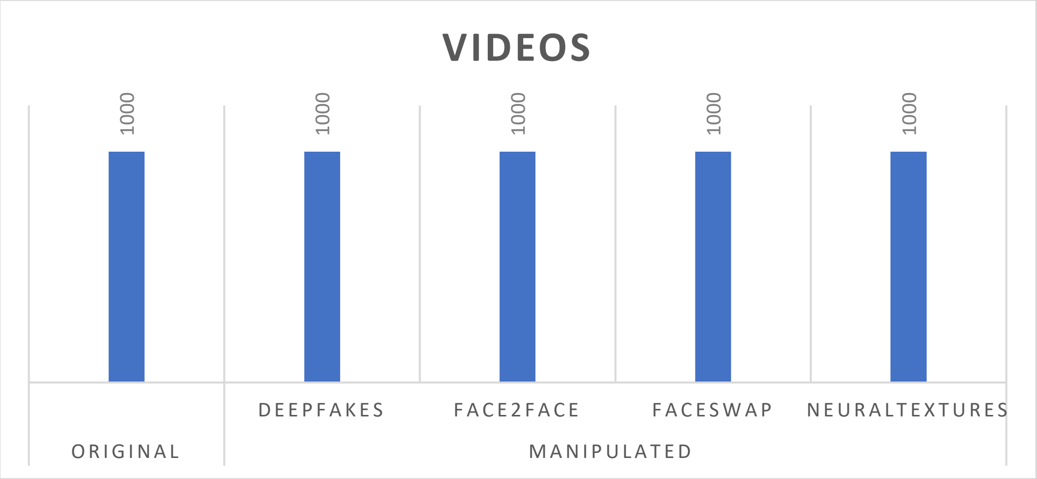


Figure 9 - FaceForensics Video Dataset

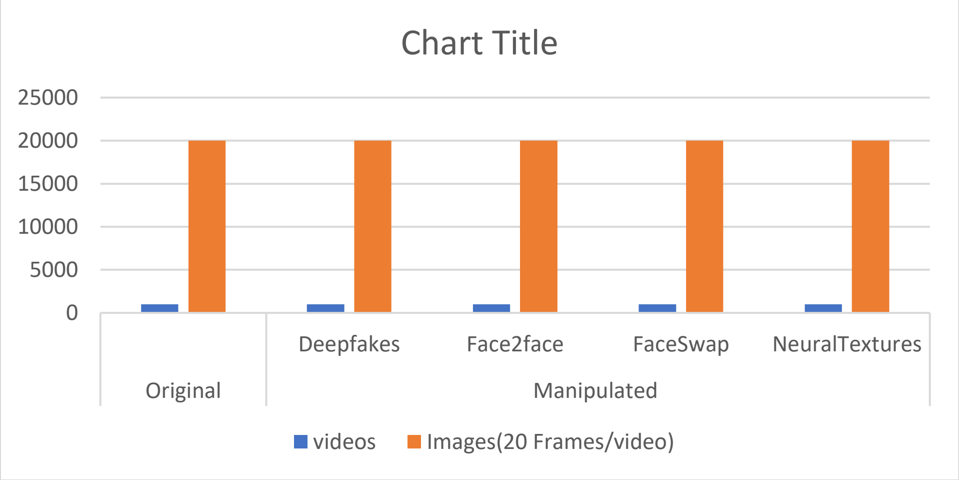


Figure 10- FaceForensics Image dataset extracted from each videos.

4.3.1. Augmenting the Dataset

To enhance the diversity of the dataset, several data augmentation techniques were applied to the extracted images. These included horizontal flipping, random rotations within ±15 degrees, brightness and contrast adjustments, addition of Gaussian noise, and random cropping and resizing. Augmentation significantly increased the dataset size to 599,154 images, with approximately 119,802 images contributed by each category. This substantial expansion ensured the dataset was robust and reflective of real-world variations in manipulated content.

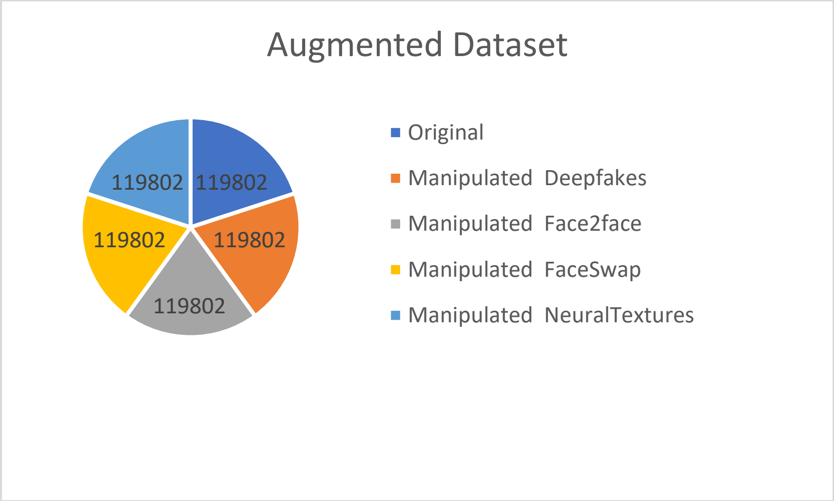


Figure 11- FaceForensics Augmented Dataset

Data augmentation is a crucial technique employed in training deep learning models, particularly in computer vision tasks. By applying a variety of transformations such as resizing, random rotations, and color adjustments, the model is exposed to a wider range of image variations. This process not only enhances the diversity of the training dataset but also helps in preventing overfitting by ensuring that the model learns to generalize better from the training data. Techniques like random horizontal flipping and sharpness adjustments introduce additional randomness, allowing the model to become more robust against different input conditions. Ultimately, these augmentations contributed to improved performance and accuracy when the model is deployed on unseen data.

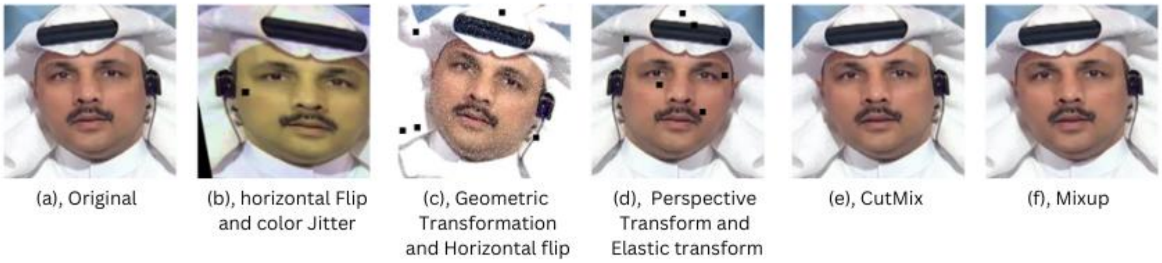
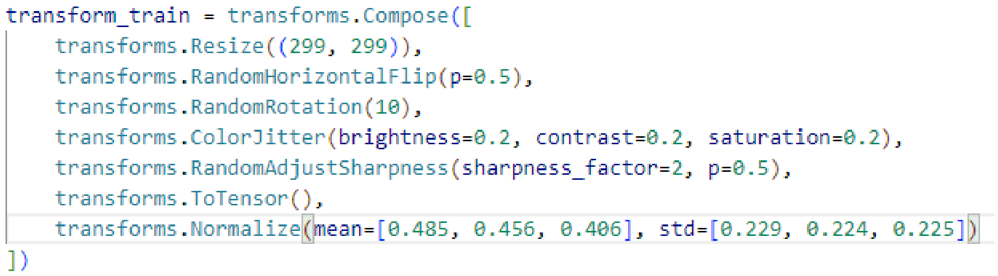

(a), Original
(b), horizontal Flip and color Jitter
(c), Geometric Transformation and Horizontal flip
(d), Perspective Transform and Elastic transform
(e), CutMix
(f), Mixup

Figure 12- Augmented Images

This technique is done on dataset preparation and on Model training.



```
transform_train = transforms.Compose([
    transforms.Resize((299, 299)),
    transforms.RandomHorizontalFlip(p=0.5),
    transforms.RandomRotation(10),
    transforms.ColorJitter(brightness=0.2, contrast=0.2, saturation=0.2),
    transforms.RandomAdjustSharpness(sharpness_factor=2, p=0.5),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])
])
```

Figure 13 - Augmentation used on Model training.

4.3.2. Image Enhancement

Following augmentation, face enhancement techniques were applied to improve the quality and focus of the images. A pre-trained face detection model, such as OpenCV's cv2.CascadeClassifier, was employed to detect and crop facial regions from each frame. Facial landmarks were used to align the cropped faces consistently, and sharpening filters were applied to enhance facial details, improving feature clarity. Additionally, pixel intensity values were normalized to ensure consistency and optimize model training.

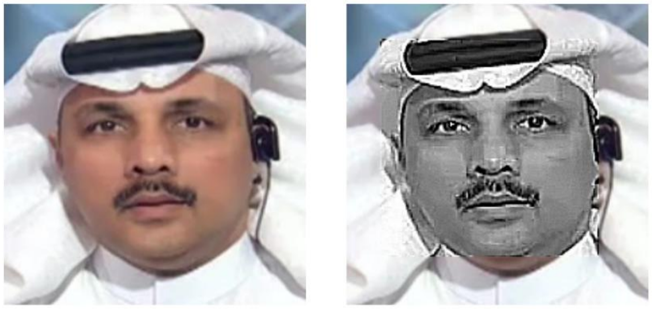


Figure 14 - Original Image vs. Enhanced Image

4.3.3. Defining Labels

The images were labeled with three key attributes: the file path, the label (either "Real" for original images or "Fake" for manipulated ones), and the subcategory for fake images, identifying the specific manipulation technique (e.g., Deepfakes, Face2Face). This labeled dataset provided a comprehensive structure for organizing and analyzing the data. A sample record from the labeled dataset is illustrated in the following table:

Image_Path	**Label**	**Subcategory**
deepfakes\000_003\processed_frame_000.png	Fake	deepfakes
neuraltextures\737_719\processed_frame_007.png	Fake	neuraltextures
face2face\719_737\processed_frame_019_aug_2.png	Fake	Face2face
faceswap\425_485\processed_frame_006_cutmix.png	Fake	faceswap
original\430\processed_frame_002_aug_0.png	Real	none

Table 3 - Labeled FaceForensics dataset with (image_path, label, and subcategory)

The dataset was then split into training, validation, and test subsets using a stratified sampling approach. This ensured a balanced representation across classes and subcategories. The training set comprised 70% of the dataset, while the validation and test sets each accounted for 15%. This split resulted in approximately 419,408 images in the training set, 89,873 images in the validation set, and 89,873 images in the test set. The splitting process was implemented using Python's scikit-learn library, with stratification applied to maintain class distributions.

The final prepared dataset of 599,154 images was diverse, well-labeled, and systematically organized for deepfake detection tasks. The following pie chart (Figure 15) illustrates the distribution of images across the dataset splits.

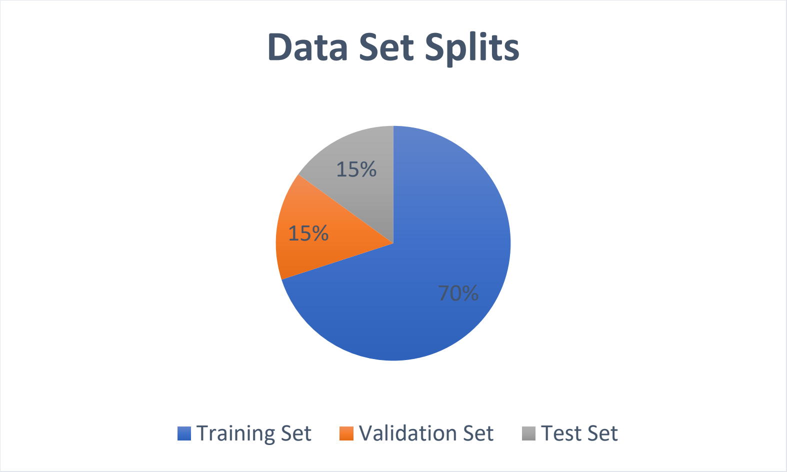


Figure 15 - Dataset Splits to training, validation, and test set.

This preparation process ensured that the dataset was well-suited for training, validating, and testing advanced deepfake detection models, enabling robust and reliable performance evaluations.

4.4. Training Components of the Proposed CNN model

The proposed Convolutional Neural Network (CNN) architecture is tailored specifically for deepfake detection, leveraging the pre-trained Xception model as a foundation. Through extensive experimentation, the model was fine-tuned by adjusting key hyperparameters such as the learning rate, batch size, and the number of epochs. The training process utilized the Adam optimizer, chosen for its efficiency in managing sparse gradients and dynamically adapting the learning rate.

The model was trained over 15 epochs with a batch size of 32, striking a balance between memory efficiency and performance. A learning rate of 1×10^{-4} was employed to ensure stable convergence throughout the training process. The architecture supports both binary classification (real vs. fake) and multi-class classification (various deepfake types), reflecting a careful balance between complexity and computational constraints.

4.4.1. Model Architecture Overview

The CNN model integrates convolutional and fully connected layers, optimized for extracting and analyzing features specific to deepfake detection. Key elements of the architecture include:

- ✓ **Input Layer:** The model processes RGB images with dimensions 299×299×32. The input layer passes the images directly to the first convolutional layer without altering the spatial dimensions, ensuring that all input data is preserved for subsequent processing.
- ✓ **Convolutional Layers:** The network incorporates several convolutional layers, utilizing depthwise separable convolutions to reduce computational complexity while maintaining performance. The first convolutional layer applies 32 filters of size 3×33, generating an output volume of 299×299×32. As the data progresses through additional layers, spatial dimensions are reduced using strides and pooling layers, enabling the model to focus on salient features.
- ✓ **Activation Function:** Each convolutional layer is paired with the Rectified Linear Unit (ReLU) activation function, which introduces non-linearity into the model. ReLU helps the network capture intricate data patterns and speeds up training by addressing the vanishing gradient issue.
- ✓ **Pooling Layers:** Max pooling layers are interspersed between convolutional layers to reduce the spatial dimensions of feature maps. These layers enhance computational efficiency while retaining critical features. For instance, max pooling layer follows the second convolutional layer, decreasing the spatial dimensions and streamlining subsequent processing.
- ✓ **Fully Connected Layers:** The architecture concludes with two fully connected (dense) layers. The first layer comprises 256 neurons, processing the flattened feature maps derived from the convolutional layers to learn high-level representations. The final output layer employs a softmax activation function with two neurons, which produces a probability distribution over the classes, facilitating binary classification.

The adoption of the Xception model, based on depthwise separable convolutions, allows the architecture to minimize the number of parameters while preserving high accuracy. This efficiency is critical for scaling the model across datasets and ensuring robust performance in deepfake detection tasks.

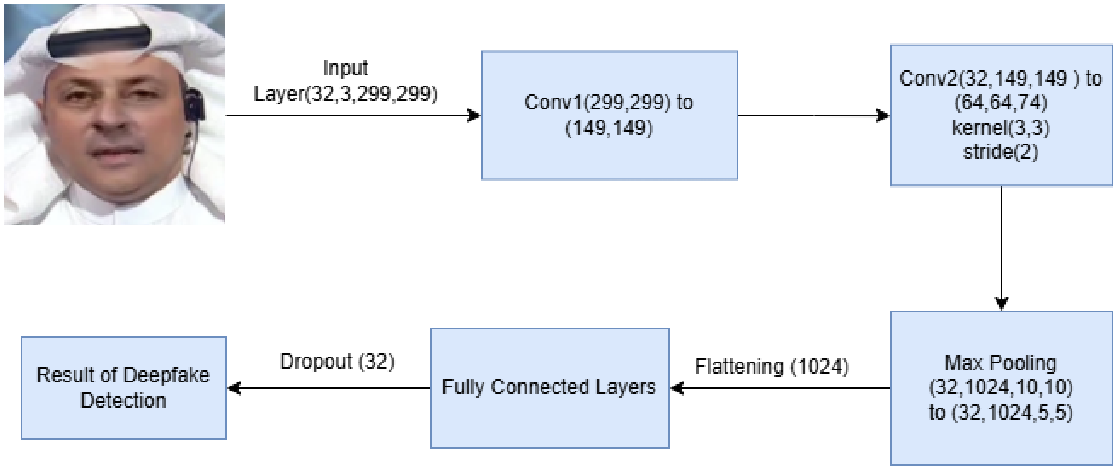

Input
Layer(32,3,299,299)
Conv1(299,299) to (149,149)
Conv2(32,149,149) to (64,64,74) kernel(3,3) stride(2)
Max Pooling (32,1024,10,10) to (32,1024,5,5)
Flattening (1024)
Fully Connected Layers
Dropout (32)
Result of Deepfake Detection

Figure 16 – Model Architecture overview

4.5. Experimental Result

The experiment commenced by utilizing both ResNet and XceptionNet architectures to evaluate their performance on the deepfake detection dataset. The XceptionNet model, leveraging depthwise separable convolutions, demonstrated superior performance compared to ResNet in terms of accuracy and robustness against overfitting. While both models were trained on the same dataset, XceptionNet consistently outperformed ResNet in validation accuracy, achieving a significant improvement in detecting subtle artifacts commonly found in deepfake images.

The training process involved fine-tuning the pre-trained XceptionNet model, which allowed it to adapt effectively to the specific characteristics of the deepfake dataset. With a learning rate set at **0.0001** and a batch size of **32**, the model underwent **15 epochs** of training, incorporating advanced data augmentation techniques to enhance its generalization capabilities. Despite the challenges posed by a limited dataset, the XceptionNet model exhibited commendable training and validation accuracy, indicating its ability to learn essential features from the images. The experimental setup included various scenarios to rigorously assess the models, demonstrating that while starting from scratch can yield limited results, leveraging pre-trained architectures like XceptionNet can significantly enhance performance in complex tasks such as deepfake detection.

The experimental setup involved three primary experiments to evaluate deepfake detection and classification using the FaceForensics++ dataset. These experiments utilized two CNN-based architectures: ResNet and XceptionNet. The first experiment tested the models without data augmentation, using the original dataset and training for 10 and 15 epochs to establish baseline performance. The second experiment addressed class imbalance by employing undersampling and oversampling techniques, training both models for 15 epochs to assess the impact of balancing. The third experiment introduced data augmentation, incorporating transformations like resizing, rotations, and intensity adjustments, with models trained for 15 epochs to evaluate the enhancement in accuracy. Both architectures were optimized with the Adam optimizer and ReLU activation, with hyperparameters including a batch size of 32 and a learning rate of 1e-4. Results demonstrated that XceptionNet consistently outperformed ResNet, particularly in the augmented dataset scenario, highlighting its suitability for deepfake detection tasks.

4.5.1. Detection and Classification of Deepfakes using Resnet and XceptionNet

Scenario 1: Testing Without Augmentation (10 and 15 Epochs)

In the first scenario, the models were evaluated without any data augmentation. The training was conducted for **10 epochs**, yielding a binary accuracy of **80.33%** and a type accuracy of **47.93%** for XceptionNet. When extending the training to **15 epochs**, XceptionNet improved slightly, achieving a binary accuracy of **81.33%** and a type accuracy of **48.72%**. In contrast, ResNet showed a binary accuracy of **77.95%** and a type accuracy of **40.32%** after **10 epochs**. After training for **15 epochs**, ResNet's performance marginally increased, resulting in a binary accuracy of **78.01%** and a type accuracy of **43.72%**.

Epoch	**ResNet Binary Accuracy**	**ResNet Type Accuracy**	**XceptionNet Binary Accuracy**	**XceptionNet Type Accuracy**
10	77.95%	40.32%	80.33%	47.93%
15	78.01%	43.72%	81.33%	48.72%

Table 4 - Testing before augmentation with different epoch sizes

Scenario 2: Testing Using a combination of Undersampling and Oversampling (15 Epochs)

In the second scenario, the models were trained using a combination of undersampling and oversampling techniques for **15 epochs**. The results indicated that ResNet's binary accuracy closely approached that of XceptionNet, showcasing a binary accuracy of **78.01%** and maintaining a type accuracy of **43.72%**. XceptionNet, on the other hand, continued to outperform ResNet with a binary accuracy of **81.33%** and a type accuracy of **48.72%**. This scenario demonstrated the effectiveness of balancing techniques in enhancing model performance while maintaining type accuracy.

Epoch	**ResNet Binary Accuracy**	**ResNet Type Accuracy**	**XceptionNet Binary Accuracy**	**XceptionNet Type Accuracy**
15	78.01%	43.72%	81.33%	48.72%

Table 5 - Testing result using the combination of undersampling and Oversampling

Scenario 3: Testing After Augmentation (15 Epochs)

The third scenario compared the models before and after applying data augmentation techniques, both trained for **15 epochs**. For ResNet, the binary accuracy improved from **78.01%** to **80.67%**, while the type accuracy increased significantly from **43.72%** to **52.23%** after augmentation. Similarly, XceptionNet showed an increase in binary accuracy from **81.16%** to **86.91%**, with type accuracy rising from **54.70%** to **70.50%**. This scenario highlighted the substantial impact of data augmentation on enhancing model performance, particularly in type accuracy.

Epoch	**ResNet Binary Accuracy**	**ResNet Type Accuracy**	**XceptionNet Binary Accuracy**	**XceptionNet Type Accuracy**
15(Before Augmentation)	78.01%	43.72	81.33%	48.72%
15(After Augmentation)	80.67%	52.23%	86.91%	70.50%

Table 6 - Testing results after augmentation

4.5.2. Result analysis for the XceptionNet Model for deepfake detection and classification.

As shown in Fig. , the XceptionNet model exhibited strong performance in the deepfake detection and classification task. In the binary classification metrics, the model demonstrated a steady increase in both the training and validation binary accuracy, reaching around 86.91% by the end of the 15 epochs. This high binary classification accuracy indicates that the XceptionNet model was able to effectively distinguish between real and deepfake samples, showcasing its potential as a robust deepfake detection system.

The type classification results further highlight the model's capabilities. The accuracy for classifying real samples was 60.12%, while the accuracy for deepfake samples was 78.68%. The model also achieved high accuracy for specific deepfake types, with 71.18% for Face2Face, 79.51% for FaceSwap, and 62.99% for NeuralTextures. The overall type classification accuracy reached 70.50%, suggesting that the XceptionNet model was able to effectively learn the distinctive features that differentiate the various deepfake types. These results demonstrate the model's potential for reliable and nuanced deepfake classification, which is crucial in combating the growing threat of manipulated media.

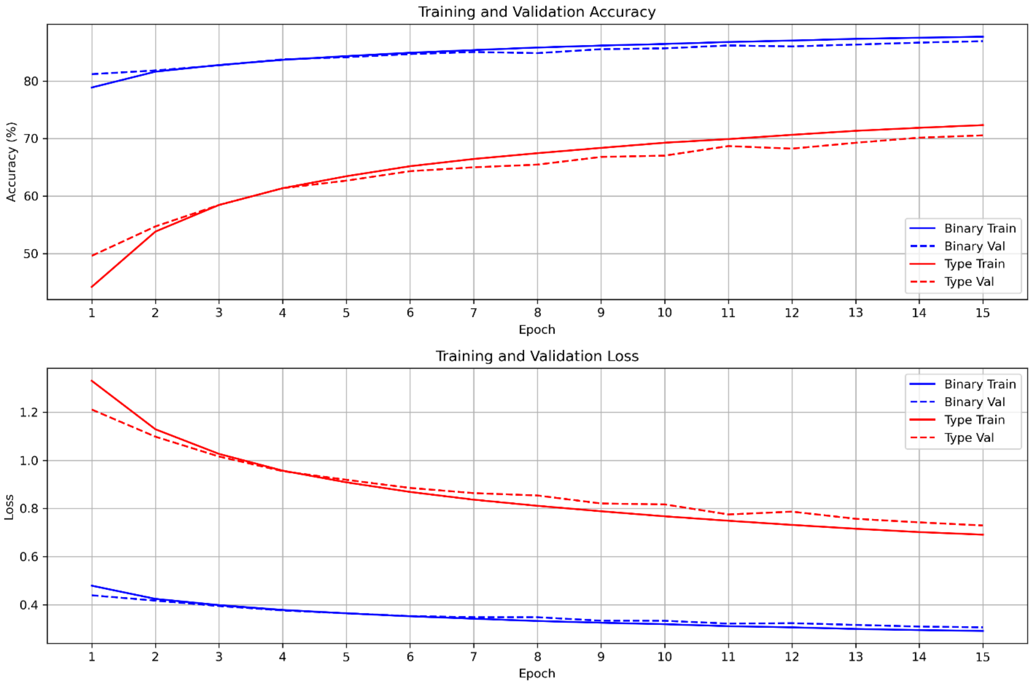


Figure 17 - XceptionNet Model Training and Loss representation

4.5.3. Result analysis for the Resnet Model for deepfake detection and classification.

As shown in fig, The ResNet model's performance in the deepfake detection and classification task, as shown in the provided image, exhibits a more modest improvement compared to the XceptionNet model.

In the binary classification metrics, the training binary accuracy starts around 78% and gradually increases to around 80.67% by the end of the 14 epochs. The validation binary accuracy, on the other hand, remains relatively stable, fluctuating between 77-78% throughout the training process.

The type classification metrics for the ResNet model show a more gradual improvement. The training type accuracy starts at around 40% and reaches approximately 52% by the final epoch. The validation type accuracy also increases, but at a slower pace, rising from around 44% in the early epochs to 52% by the end of the training.

While the ResNet model demonstrates an overall positive trend in both binary and type classification metrics, the performance gains are less pronounced compared to the XceptionNet model. This suggests that the ResNet architecture may not be as well-suited for the specific task of deepfake detection and classification as the XceptionNet model, which appears to have a stronger capability in learning the distinctive features that differentiate the various types of deepfakes.

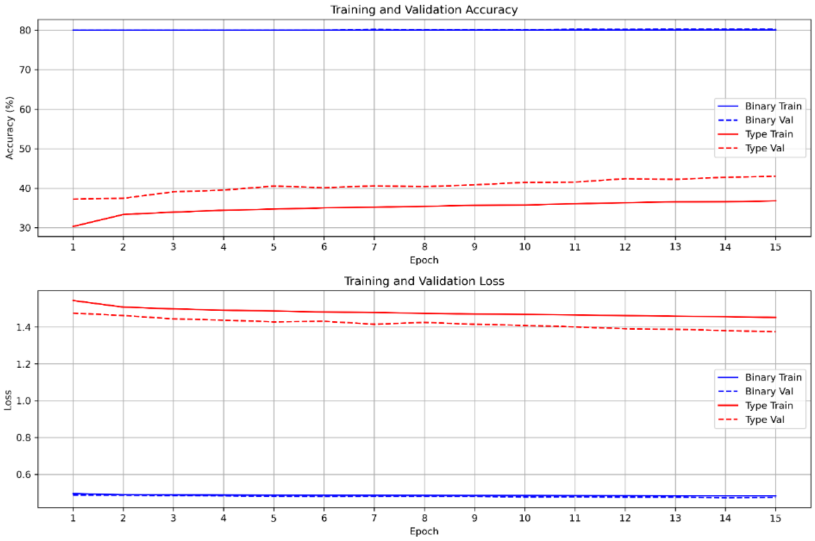


Figure 18 - Resnet18 Model Training and Loss Representation

4.6. Performance Analysis

4.6.1. Performance evaluation for XceptionNet

The binary classification report showcases strong performance, with an overall accuracy of 81%. The model demonstrates high precision (0.81) and recall (0.99) for the positive class (1), indicating its ability to accurately identify deepfake samples. However, the model struggles with the negative class (0), with a relatively low precision (0.70) and recall (0.10), suggesting room for improvement in detecting non-deepfake samples.

The type classification report reveals a more nuanced picture. The model achieves a moderate overall accuracy of 55%, suggesting challenges in accurately classifying the different types of deepfakes. The F1-scores for the individual types range from 0.44 to 0.66, with the highest performance on Type 1 and the lowest on Type 0 and Type 4. This indicates that the model has learned to differentiate certain deepfake types better than others, and further optimization may be necessary to enhance its type classification capabilities.

The analysis of these performance metrics provides valuable insights into the strengths and weaknesses of the XceptionNet model, guiding future model refinements and highlighting areas for targeted improvement in the deepfake detection and classification task.

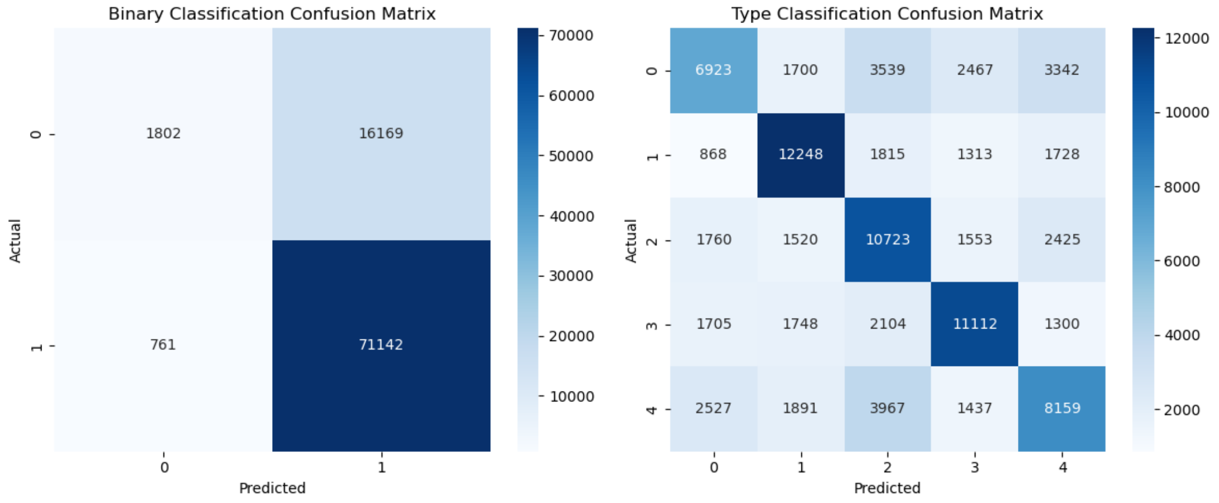
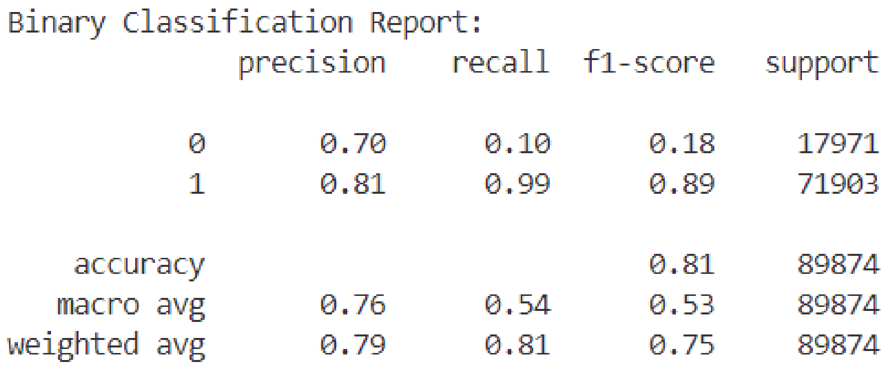


Figure 19 - XceptionNet Confusion metrics

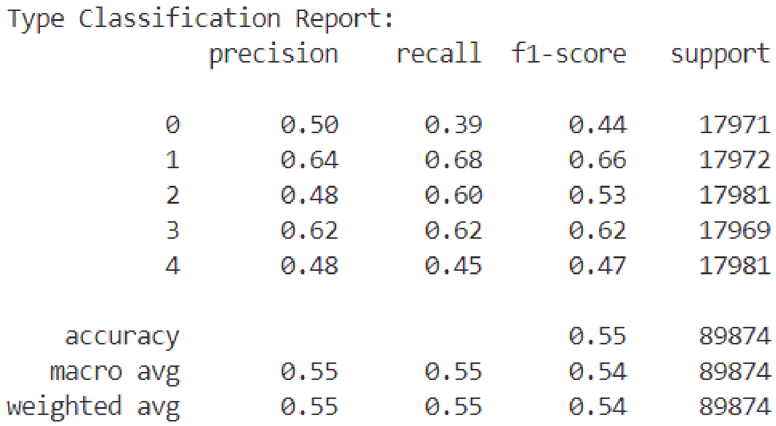


```
Binary Classification Report:
              precision    recall  f1-score   support

           0       0.70      0.10      0.18     17971
           1       0.81      0.99      0.89     71903

    accuracy                           0.81     89874
   macro avg       0.76      0.54      0.53     89874
weighted avg       0.79      0.81      0.75     89874
```

Figure 20 - XceptionNet Binary Classicication Report



```
Type Classification Report:
              precision    recall  f1-score   support

           0       0.50      0.39      0.44     17971
           1       0.64      0.68      0.66     17972
           2       0.48      0.60      0.53     17981
           3       0.62      0.62      0.62     17969
           4       0.48      0.45      0.47     17981

    accuracy                           0.55     89874
   macro avg       0.55      0.55      0.54     89874
weighted avg       0.55      0.55      0.54     89874
```

Figure 21 - XceptionNet Type Classicication Report

Metric	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**ROC AUC**
Binary Classification Metrics	0.8116	0.8148	0.9894	0.8937	0.7375
Type Classification Metrics	0.5470	0.5460	0.5470	0.5436	

Table 7 - XceptionNet Accuracy, Precision, recall, F1 Score and ROC AUC for binary and type classification metrics

4.6.2. Performance evaluation for ResNet

The binary classification report shows strong performance, with an overall accuracy of 80%. The model demonstrates high recall (1.00) for the positive class (1), indicating its ability to accurately identify deepfake samples. However, the precision for the positive class (0.80) and the overall performance on the negative class (0) are relatively lower, with a precision of 0.73 and a recall of only 0.02. This suggests that the model may be biased towards classifying samples as deepfakes, leading to a higher rate of false positives.

The type classification report reveals more nuanced performance. The overall accuracy is 43%, indicating challenges in accurately classifying the different types of deepfakes. The F1-scores for the individual types range from 0.38 to 0.52, with the highest performance on Type 1 and the lowest on Type 0 and Type 4. This suggests that the model has learned to differentiate certain deepfake types better than others, and further optimization may be necessary to enhance its type classification capabilities.

The analysis of these performance metrics provides valuable insights into the strengths and weaknesses of the ResNet model. While the binary classification performance is reasonably strong, the type classification results highlight the need for further model refinement and optimization to improve the model's ability to accurately distinguish between different deepfake types.

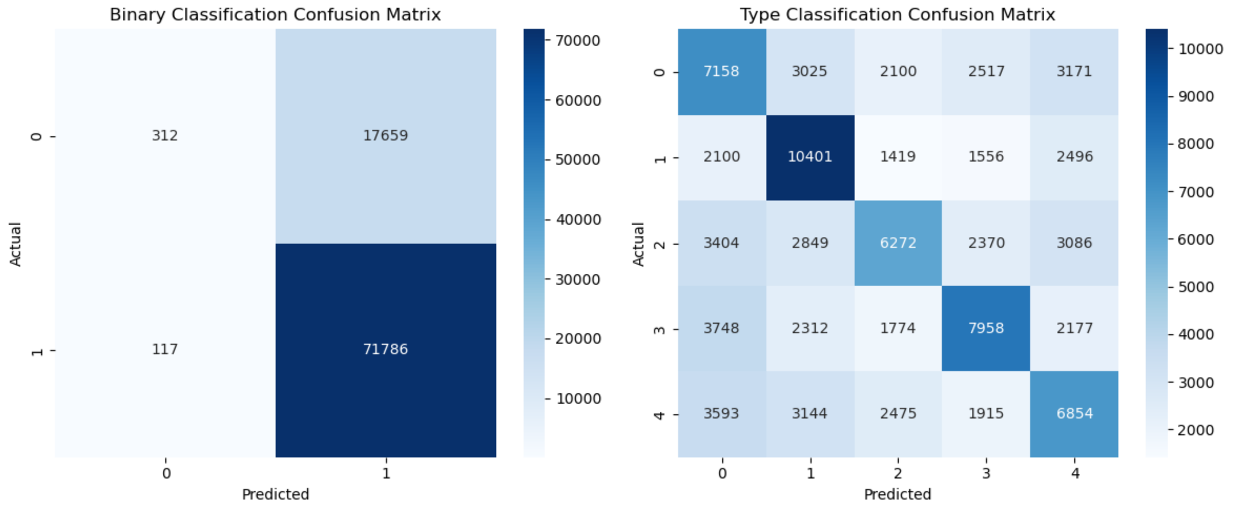
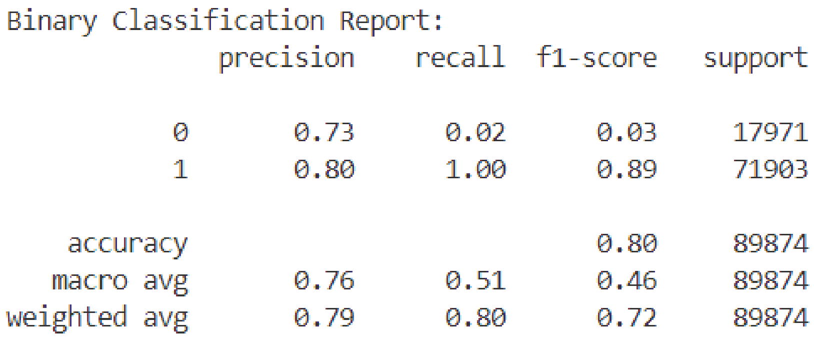


Figure 22 -ResNet Confusion metrics

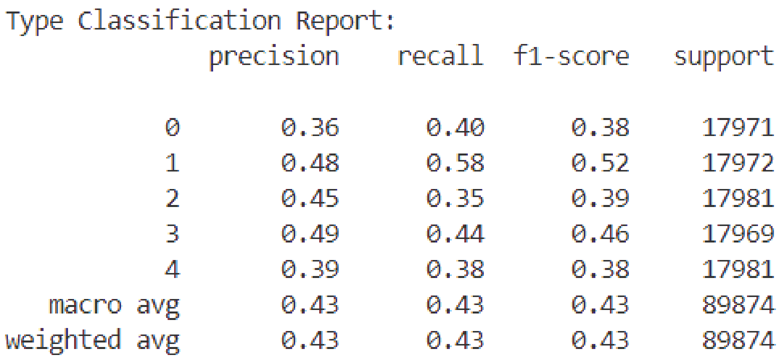


```
Binary Classification Report:
              precision    recall  f1-score   support

           0       0.73      0.02      0.03     17971
           1       0.80      1.00      0.89     71903

    accuracy                           0.80     89874
   macro avg       0.76      0.51      0.46     89874
weighted avg       0.79      0.80      0.72     89874
```

Figure 23 - ResNet Binary Classification Report



```
Type Classification Report:
              precision    recall  f1-score   support

           0       0.36      0.40      0.38     17971
           1       0.48      0.58      0.52     17972
           2       0.45      0.35      0.39     17981
           3       0.49      0.44      0.46     17969
           4       0.39      0.38      0.38     17981
   macro avg       0.43      0.43      0.43     89874
weighted avg       0.43      0.43      0.43     89874
```

Figure 24 - ResNet Type Classification Report

Metric	**Accuracy**	**Precision**	**Recall**	**F1 Score**	**ROC AUC**
Binary Classification Metrics	0.8022	0.8026	0.9984	0.8898	0.6642
Type Classification Metrics	0.4300	0.4313	0.4300	0.4280	

Table 8 - ResNet Accuracy, Precision, recall, F1 Score and ROC AUC for binary and type classification metrics

4.6.3. Model Comparison

When comparing the performance of XceptionNet and ResNet in deepfake detection and classification, XceptionNet clearly outperforms ResNet across all evaluated metrics. XceptionNet's use of depthwise separable convolutions enables it to extract fine-grained features more effectively, resulting in higher binary classification accuracy (86.91% vs. ResNet's 80.67%) and superior type classification accuracy (70.50% vs. ResNet's 52.23%) after augmentation. ResNet, while demonstrating reasonable accuracy, struggled to achieve comparable performance in distinguishing between various manipulation techniques, likely due to its less sophisticated feature extraction capabilities. Additionally, XceptionNet exhibited stronger robustness to augmentation and improved precision and recall, making it more reliable for nuanced detection tasks. Based on these findings, XceptionNet is the recommended model for deepfake detection and classification due to its higher accuracy, robustness, and ability to generalize across multiple manipulation types.

4.7. Discussion of Result

The results presented showcase the comparative performance of the XceptionNet and ResNet models in addressing the task of deepfake detection and classification. Overall, the findings indicate that the XceptionNet model outperforms the ResNet model in both binary classification and type classification tasks.

In the binary classification task, the XceptionNet model demonstrated a consistently strong performance, achieving an overall accuracy of 81.16% before augmentation and further improving to 86.91% after data augmentation. The model's high precision and recall for the positive class (deepfake samples) suggest its ability to accurately identify deepfake instances. However, the relatively lower precision and recall for the negative class (non-deepfake samples) indicate that the model may still struggle with some false positives.

The type classification results for the XceptionNet model reveal a more nuanced picture. The overall type classification accuracy of 54.70% before augmentation and 70.50% after augmentation suggests that the model has learned to effectively differentiate between the various deepfake types, but there is still room for improvement. The F1-scores for the individual types highlight the model's varying abilities in classifying different deepfake types, which could be further optimized through targeted model refinements. In comparison, the ResNet model exhibits a more modest performance improvement, with its type classification accuracy remaining relatively low even after data augmentation.

The research identified XceptionNet and ResNet as suitable CNN architectures for detecting and classifying various types of deepfake manipulations. XceptionNet emerged as the more effective model, achieving higher binary accuracy (86.91% after augmentation) and type classification accuracy (70.50%) compared to ResNet, which achieved 80.67% and 52.23%, respectively. XceptionNet's depthwise separable convolutions allow it to extract fine-grained features, making it particularly adept at detecting subtle artifacts introduced by deepfake techniques like Face2Face and FaceSwap. ResNet, while effective, demonstrated limitations in distinguishing between different manipulation types, likely due to its reliance on traditional residual learning techniques. These findings suggest that XceptionNet is a more suitable choice for comprehensive deepfake detection and classification tasks.

Data augmentation was a key AI technique integrated into the CNN architectures to enhance performance. Techniques such as resizing, rotations, and color adjustments enriched the training dataset, leading to improved model robustness and generalization. For instance, augmentation boosted binary and type classification accuracies for both XceptionNet and ResNet, with the former showing greater gains (e.g., type accuracy increased from 54.70% to 70.50%). Additionally, hyperparameter tuning and the inclusion of advanced loss functions were instrumental in optimizing model performance. Future enhancements could include integrating attention mechanisms or transformer modules to enable models to focus on the most informative regions of the images. These AI techniques collectively enhance the ability of CNN architectures to detect and classify deepfakes more effectively.

AI-enhanced CNN models, particularly XceptionNet, demonstrated strong performance in deepfake image detection. XceptionNet achieved a binary classification accuracy of 86.91% and a type classification accuracy of 70.50% after incorporating data augmentation and hyperparameter tuning. Its precision, recall, and F1-scores were highest for detecting manipulated images, particularly for deepfake categories like Face2Face and FaceSwap. ResNet, though improved by augmentation, lagged behind with a binary classification accuracy of 80.67% and type classification accuracy of 52.23%. The results highlight the effectiveness of AI-enhanced

CNN models in identifying deepfake manipulations, with XceptionNet outperforming ResNet in both binary detection and type classification tasks.

CHAPTER FIVE

CONCLUSION AND FUTURE WORK

5.1. Conclusion

This thesis addresses the critical issue of identifying deepfake images by employing AI-enhanced Convolutional Neural Networks (CNNs). As deepfake technology continues to advance, its misuse in areas such as politics, media, and personal interactions raises serious concerns about the credibility and authenticity of digital content. This research focuses on tackling these challenges by designing an effective detection system using state-of-the-art AI techniques.

The study revealed that traditional deepfake detection approaches are becoming less effective due to the increasing complexity of modern manipulation techniques. A thorough review of existing literature highlighted significant gaps in current detection strategies, underscoring the necessity for more robust solutions. The proposed framework integrates advanced CNN architectures with innovative AI methodologies, resulting in notable improvements in both accuracy and efficiency. Experimental results demonstrated the model's ability to distinguish between genuine and manipulated images, indicating its potential as a dependable tool to combat the spread of misinformation.

This work makes a significant contribution to computer science by introducing a novel deepfake detection approach. By leveraging AI-enhanced CNNs, the model not only identifies manipulated images but also adapts to various forms of deepfake techniques. The findings emphasize the importance of continuous innovation in detection technologies to keep up with the fast-evolving landscape of digital manipulation. This research lays a foundation for future advancements in the field, highlighting the pivotal role of technology in preserving the integrity of digital content.

As a limitation, this study has a few important constraints. While it utilizes a large dataset, it may still be insufficient to capture the full spectrum of deepfake techniques, which could hinder the model's ability to generalize effectively. The findings suggest that accuracy can significantly improve with larger datasets. And the study does not address new emerging deepfake generation methods that could introduce new challenges for detection in the future, potentially limiting the model's applicability in real-world scenarios.

5.2. Future work

Building on the outcomes of this research, several directions can be explored for advancing deepfake detection:

- ✓ Expanding the dataset to include more diverse and comprehensive samples is crucial for improving the model's robustness. Developing localized datasets for testing and identifying fake images is also a key area of focus.
- ✓ As deepfake generation techniques become increasingly sophisticated, it is vital for future studies to adapt detection methods to counter emerging manipulation strategies, ensuring the continued effectiveness of detection systems.
- ✓ Incorporating adversarial training techniques could significantly strengthen the model's resilience against advanced deepfake generation methods. Exposing the system to adversarial examples during training may improve its ability to recognize and mitigate new manipulation strategies. This proactive approach is essential for staying ahead of evolving threats posed by digital manipulation technologies.

Reference

[1] M. Westerlund, "The emergence of deepfake technology: A review," *Technology innovation management review*, vol. 9, no. 11, 2019.

[2] A. S. George and A. S. H. George, "Deepfakes: The Evolution of Hyper realistic Media Manipulation," *Partners Universal Innovative Research Publication*, vol. 1, no. 2, pp. 58–74, 2023.

[3] Y. Patel *et al.*, "An improved dense CNN architecture for deepfake image detection," *IEEE Access*, vol. 11, pp. 22081–22095, 2023.

[4] S. Ramachandran, A. V. Nadimpalli, and A. Rattani, "An experimental evaluation on deepfake detection using deep face recognition," in *2021 International Carnahan Conference on Security Technology (ICCST)*, IEEE, 2021, pp. 1–6.

[5] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: an overview and application in radiology," *Insights Imaging*, vol. 9, pp. 611–629, 2018.

[6] A. Malik, M. Kuribayashi, S. M. Abdullahi, and A. N. Khan, "DeepFake detection for human face images and videos: A survey," *Ieee Access*, vol. 10, pp. 18757–18775, 2022.

[7] I. Korshunova, W. Shi, J. Dambre, and L. Theis, "Fast face-swap using convolutional neural networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 3677–3685.

[8] S. Em, "Exploring experimental research: Methodologies, designs, and applications across disciplines," *Designs, and Applications Across Disciplines*, 2024.

[9] S. Salman, J. A. Shamsi, and R. Qureshi, "Deep fake generation and detection: Issues, challenges, and solutions," *IT Prof*, vol. 25, no. 1, pp. 52–59, 2023.

[10] J. Pu, N. Mangaokar, B. Wang, C. K. Reddy, and B. Viswanath, "Noisescope: Detecting deepfake images in a blind setting," in *Proceedings of the 36th Annual Computer Security Applications Conference*, 2020, pp. 913–927.

[11] S. M. Qureshi, A. Saeed, S. H. Almotiri, F. Ahmad, and M. A. Al Ghamdi, "Deepfake forensics: a survey of digital forensic methods for multimodal deepfake identification on social media," *PeerJ Comput Sci*, vol. 10, p. e2037, 2024.

[12] R. Gil, J. Virgili-Gomà, J.-M. López-Gil, and R. García, "Deepfakes: evolution and trends," *Soft comput*, vol. 27, no. 16, pp. 11295–11318, 2023.

[13] A. S. George and A. S. H. George, "Deepfakes: the evolution of hyper realistic media manipulation," *Partners Universal Innovative Research Publication*, vol. 1, no. 2, pp. 58–74, 2023.

[14] T. Shen, R. Liu, J. Bai, and Z. Li, "'deep fakes' using generative adversarial networks (gan)," *Noiselab, University of California, San Diego*, 2018.

[15] A. Kohli and A. Gupta, “Detecting deepfake, faceswap and face2face facial forgeries using frequency cnn,” *Multimed Tools Appl*, vol. 80, no. 12, pp. 18461–18478, 2021.

[16] M. Masood, M. Nawaz, K. M. Malik, A. Javed, A. Irtaza, and H. Malik, “Deepfakes generation and detection: State-of-the-art, open challenges, countermeasures, and way forward,” *Applied intelligence*, vol. 53, no. 4, pp. 3974–4026, 2023.

[17] Y. Wei *et al.*, “Maggan: High-resolution face attribute editing with mask-guided generative adversarial network,” in *Proceedings of the Asian Conference on Computer Vision*, 2020.

[18] Y. Patel *et al.*, “Deepfake Generation and Detection: Case Study and Challenges,” *IEEE Access*, vol. 11, pp. 143296–143323, 2023, doi: 10.1109/ACCESS.2023.3342107.

[19] M. S. Rana, M. N. Nobi, B. Murali, and A. H. Sung, “Deepfake Detection: A Systematic Literature Review,” *IEEE Access*, vol. 10, pp. 25494–25513, 2022, doi: 10.1109/ACCESS.2022.3154404.

[20] S. J. Park, M. Kim, J. Hong, J. Choi, and Y. M. Ro, “Synctalkface: Talking face generation with precise lip-syncing via audio-lip memory,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022, pp. 2062–2070.

[21] D. A. Coccomini, R. Caldelli, F. Falchi, C. Gennaro, and G. Amato, “Cross-forgery analysis of vision transformers and cnns for deepfake image detection,” in *Proceedings of the 1st International Workshop on Multimedia AI against Disinformation*, 2022, pp. 52–58.

[22] W. H. Abir *et al.*, “Detecting Deepfake Images Using Deep Learning Techniques and Explainable AI Methods”.