Deepfake Detection using Deep Learning

By

Md. Wahiduzzaman Nayem

201-35-2979

The report presented in partial fulfillment of the requirements for the Bachelor of Science in Software Engineering.

DEPARTMENT OF SOFTWARE ENGINEERING,

DAFFODIL INTERNATIONAL UNIVERSITY.

FALL 2023

APPROVAL

This thesis titled on "DeepFake Detection using Deep Learning", submitted by **Md. Wahiduzzaman Nayem (ID: 201-35-2979)** to the Department of Software Engineering, Daffodil International University, has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science and approved as to style and contents.

Board of Examiners

____

Dr. Imran Mahmud Chairman

Associate Professor & Head In-charge

Department of Software Engineering

Faculty of Science and Information Technology

Daffodil International University

____

Examiner Internal Examiner 1

____

Examiner Internal Examiner 2

____

Examiner External Examiner 1

THESIS DECLARATION

I hereby declare that, this thesis report is done by me under the supervision of Mr. Md. Shohel Arman, Assistant Professor, Department of Software Engineering, Daffodil International University, in partial fulfillment my original work. I am also declaring that neither this thesis nor any part therefore has been submitted else here for the award of Bachelor or any degree.

Supervised By

____

Md. Shohel Arman

Assistant Professor,

Department of Software Engineering,

Daffodil International University.

Submitted By

____

Md. Wahiduzzaman Nayem

ID: 201-35-2979

Department of Software Engineering,

Daffodil International University.

ACKNOWLEDGEMENT

First of all, I want to express my gratitude to Almighty Allah for granting me His divine favor and making it possible for me to finish my undergraduate thesis.

I would like to say thank you to my Supervisor Md. Shohel Arman, Assistant professor in the Department of Software Engineering at "Daffodil International University" in Dhaka. He deserves my sincere gratitude and respect. His extensive knowledge and direction in the section on "Deep Learning" really helped me to finish this entire thesis work. He has made it possible through his unwavering empathy, academic leadership, constant inspiration, diligent monitoring, constructive criticism, helpful advice, and the review of numerous subpar manuscripts that he has corrected at every level.

I wish to extend my sincere appreciation to Dr. Imran Mahmud, the head of the "Software Engineering" Department within the Faculty of Science and Information Technology, as well as to the other professors, faculties, and staff members of the SWE Department at "Daffodil International University" for their thoughtful assistance in completing my work.

Last but not least, I would like to express my gratitude to my parents for their unwavering support and love.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Abstract

Deep learning has shown to be a successful approach for solving several difficult problems, such as huge data analytics, computer vision, and human-level control. However, developments in deep learning have also been used to produce software that compromises democracy, privacy, and national security. Deepfake is the name of one of the recent deep learning applications. Deepfake algorithms have the ability to produce convincingly fake images and videos that are hard for people to tell apart from the real thing. Deepfake is a subfield of artificial intelligence that creates convincing image, audio, and video frauds by combining the generator and discriminator algorithms, two rival AI techniques that comprise a generative adversarial network (GAN). A Reddit user going by the username Deepfakes discussed the origins of the term "Deepfake" in 2017. Although there are numerous techniques for identifying Deepfakes, not all of them are flawless and consistent in every situation. Furthermore, outdated generalizing methods must be updated frequently to keep up with the most recent methods for producing deep fakes. My research focuses on cutting-edge techniques that integrate many Deep Learning models to create changed videos and identify them. We test our method on a large-scale balanced and mixed data-set created by combining the different accessible data-sets, such as Face-Forensic++, Deepfake detection challenge, and Celeb-DF, YouTube Videos, and custom crafted deepfakes, in order to replicate real-time scenarios and improve the model's performance on real-time data. We also demonstrate how our method can produce competitive outcomes in a very straightforward and reliable way. Our model was trained on 400 films in all, 217 train and 183 test videos. We achieved 94.871794% accuracy on the total.

Keywords:

Res-Next Convolution neural network, Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Computer vision.

Chapter 1

Introduction

1.1 Introduction

Deepfake movies have been making their rounds on all of the major social media sites lately. These convincing movies are typically created by adding real-life facial emotions and swapping out a person's visage for another. Deepfakes target large groups of people, most of whom are unaware that the fraud is being conducted, making them far more dangerous than we initially realize. Initially, deepfake movies were produced with good intentions in a number of industries, such as entertainment (Panyatham 2020), advertising (Sachs 2019), and film (Grossman 2018). Deepfakes, on the other hand, are now frequently employed for illegal purposes like extortion, the dissemination of false information, the production of pornographic material, fake surveillance footage, and faked evidence. The general public can download numerous apps and tools for free that have the potential to create remarkably realistic deepfake images and movies. A primary concern is the quick evolution of deepfakes, which are outpacing increasingly complex detection techniques. As such, it is essential to develop ever more complex models that are able to recognize these trick films. Over time, further study has been conducted on these face-modifying techniques (Sven van Asseldonk, 2021). The bulk of this field's study is on identity swap modification techniques since they have the potential to cause big public difficulties (Davide Coccomini, 2022). Convolutional neural networks (CNNs) are used in the majority of Deepfake detection methods to extract frame-level information from images. The information that gradually fades away in a movie cannot be detected by these detection methods, which are often referred to as temporal characteristics. These temporal properties outperformed frame-level based techniques when used in a number of recent studies (de Lima et al. 2020). No studies have been discovered that also integrate hand-crafted facial characteristics to improve classification performance, despite Tianyu, Zhenjiang, and Jianhu's (2018) demonstration that hand-crafted features may provide models with additional information. This work investigates the identification of Deepfake videos by using a Long Short-Term Memory (LSTM) network in conjunction with manually generated face and ResNext derived frame-level data.

1.2 Background

Deepfakes are seen to be the main AI risk in the world of social media platforms that are always growing. These lifelike face-swapping deepfakes are widely employed in scenarios to create revenge porn, plan terrorist assaults, stir up political turmoil, and extort people. The naked videos of Brad Pitt and Angelina Jolie are two examples. An artificial media creation known as a "deepfake" is when a person's likeness is used to substitute a real person in an already-existing picture or video. We call the technique of fabricating DeepFakes "deep fakery." DeepFakes are frequently used in identity theft and news falsification. The terms "deep learning" and "fake" are combined to form the term "deepfake". Artificial neural networks are used in deep learning, a kind of machine learning, to learn from data. A huge collection of photos or videos of the individual whose resemblance is being substituted is often where DeepFakes acquire their data from.

Deepfakes' legal standing is always changing. In certain states, it is unlawful to produce and distribute DeepFakes—which are used to fabricate news or impersonate persons. However, DeepFakes are not expressly forbidden by laws in other nations. Deepfake presents numerous really dangerous scenarios.

a) They might be used maliciously to impersonate someone else or spread false information.

b) They could be employed to disseminate false information or harm someone's reputation.

c) They might be used to produce media that shows sexual abuse of children.

d) They have the power to influence someone's emotions or actions.

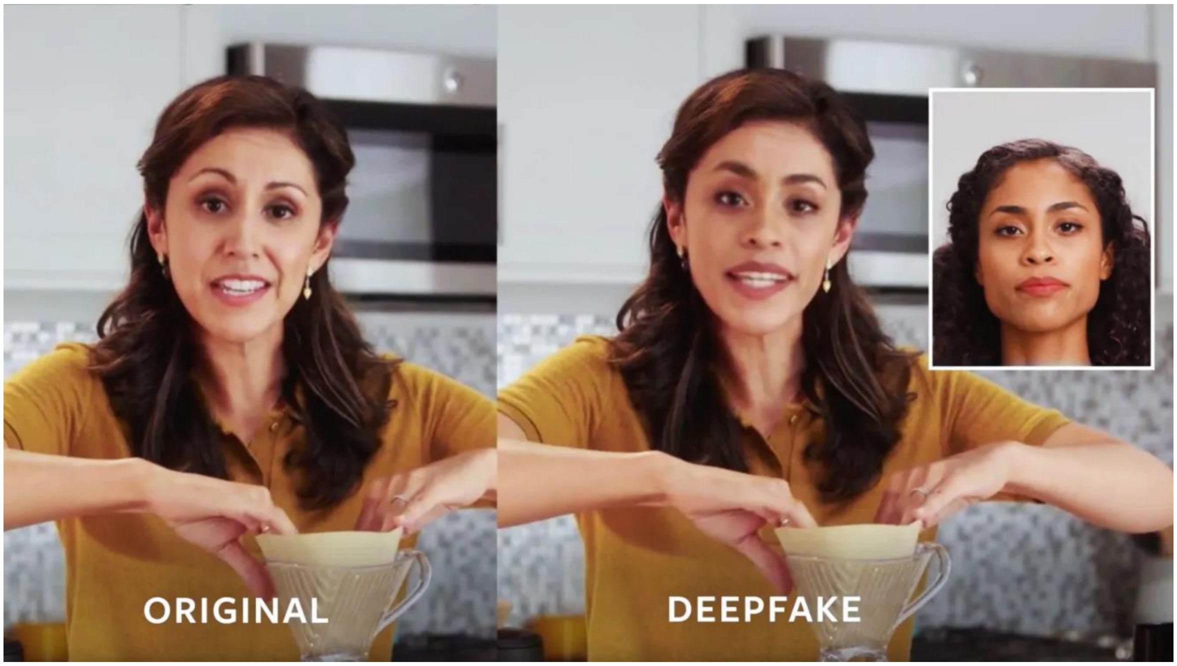


Figure 1.1: Deepfake Image

1.3 Motivation

The need to counter the growing threat posed by this cutting-edge technology is what drives the pursuit of deepfake detection research. Deepfakes, which are distinguished by their capacity to create convincingly comparable but completely fake media material, have become an extremely powerful instrument for manipulation, presenting a complex risk to people, institutions, and social structures. These artificial intelligence-driven manipulations have the potential to cause extensive disinformation, identity theft, and intrusive privacy breaches due to their sophisticated nature.

The urgent necessity to protect information integrity in the face of these dishonest technological breakthroughs is what drives research into deepfake detection. Beyond the immediate issues of disseminating false information, the consequences include the deterioration of public confidence in digital media, which poses a significant social challenge. The possibility that deepfakes could be used for malicious intent emphasizes how urgent it is to provide reliable detection methods in order to mitigate their negative impacts.

This research is a creative and inventive reaction to the rapidly changing technology landscape in the field of academic inquiry. It aims to provide preventive measures that guard against the malevolent uses of deepfake technology in addition to comprehending its complex inner workings. Our goal in conducting this research is to advance our understanding of the ramifications of deepfakes and open the door for the creation of sophisticated and efficient detection technologies. Fundamentally, the goal of this research is to gather information that will enable us to lessen the threats that deepfakes bring, strengthening the groundwork for a reliable digital environment that benefits both people and society as a whole.

1.4 Deepfake Creation

Artificial intelligence (AI) techniques are used to make synthetic media known as "deepfakes." These techniques usually include manipulating photos, video, or audio to create content that looks real but is actually fake. Depending on the type of media, these modifications can be in the form of audio-only, image-based deepfakes, video with sound, or video without sound. Deepfakes are created using a variety of techniques, all of which make use of cutting-edge AI algorithms. These techniques fall into the following general categories:

1. Face-switching:

   This technique is swapping out a person's facial features in a picture or video with another's. Generative adversarial networks (GANs), which are trained to imitate and reproduce the target person's facial expressions, gestures, and physical characteristics, are frequently used to accomplish this.

2. Voice synthesis:

   The goal of audio deepfakes is to mimic a human voice through the use of voice synthesis algorithms. By analyzing the rhythms, intonations, and subtleties of the target voice, these algorithms enable the creation of realistic-sounding speech that can be used to alter or fabricate bogus audio content.

3. Lip-syncing:

   This technique is used in video deepfakes to synchronize changed facial expressions and movements with previously recorded speech or text. The outcome is a convincing film that seems real since the modified face's lip movements were precisely mapped to match the speech patterns.

4. Style transfer:

   Techniques for transferring styles entail superimposing the visual elements of one picture or video onto another. Using this technique, one source's artistic style is transferred to another's content to produce visually appealing and realistic-looking deepfakes.

5. Generative Adversarial Networks (GANs) and Autoencoders:

   Deepfake production methods often heavily rely on GANs. GANs are made up of a discriminator and a generator that compete to produce extremely realistic material. Another kind of neural network called an autoencoder is also used to encode and decode data, which helps create believable deepfakes.

While deepfake technology has many creative uses, it should be noted that there are potential ethical issues and misuses as well, such as the creation of harmful impersonation or deceptive information. In order to address the ethical implications of deepfake technology, efforts are currently being made to establish detection techniques and legislation.

1.5 Detecting Deepfakes

Because deepfakes are created using complex techniques, detecting them can be difficult. Nonetheless, scientists and programmers are presently devising strategies to detect and lessen the influence of deepfake material. Typical methods for identifying deepfakes include the following:

1. Face irregularities:

   Deepfakes can be identified by examining facial features for anomalies. Realistic facial expressions move consistently, whereas deepfake faces can have irregularities in facial characteristics that are hard to recreate, or abnormal blinking or lip-sync problems.

2. Eye movement and blinking:

   Deepfake algorithms might have trouble simulating realistic eye movement and blinking. Analyzing eye activity patterns, such as blink rates and how well one eye moves in tandem with other facial emotions, can reveal signs of artificial manipulation.

3. Audio analysis:

   Tools for audio analysis can be used to find anomalies in audio-based deepfakes, such as strange pauses, artificial intonations, or artifacts that might point to the employment of synthesis techniques. Algorithms for speech analysis can compare the vocal patterns found with the known traits of the purported speaker.

4. Lip-sync detection:

   Since lip-syncing is frequently used in video deepfakes, irregularities can be found by examining how well the audio and lip movements synchronize. Automated systems are able to evaluate how speech patterns match up with lip movements in order to detect possible manipulation.

5. Forensic analysis:

   The metadata of media files can be examined using digital forensics techniques. Anomalies in the dates of file creation, editing history, or compression artifacts could be signs of manipulation or deepfake.

6. Biometric analysis:

   Deepfake-generated information can be distinguished from authentic content by utilizing biometric data, such as examining the distinctive patterns in a person's voice or face.

Disparities that hint to manipulation can be found by machine learning algorithms that have been trained on biometric data.

7. Consistency between frames:

   Because it is difficult to replicate realistic movements, deepfake films frequently have discrepancies between frames. Deepfake creation abnormalities can be found by examining the coherence and consistency of facial characteristics and expressions over video frames.

8. Behavioral analysis:

   It's possible that deepfakes don't display the same organic behavioral signs that real people do. Behavioral analysis can assist in spotting differences that point to intentional manipulation. This analysis involves evaluating gestures, body language, and environmental indicators.

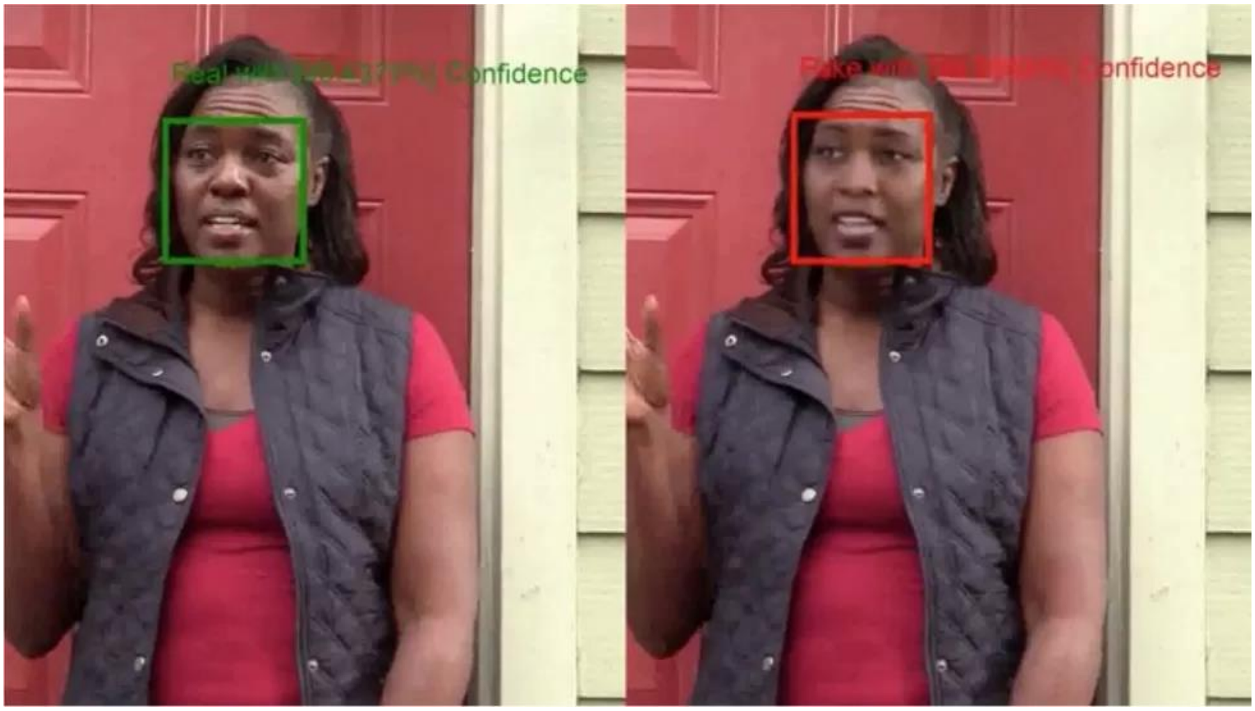

Real with [illegible] Confidence
Fake with [illegible] Confidence

Figure 1.2: Detecting Deepfake

1.6 Problem Statement

Since visual effects have been used for decades to display convincing modifications of digital photos and videos, recent developments in deep learning have significantly increased the realism of fake content and the ease with which it may be produced. These so-called artificial intelligence-generated material, also known as "deep fakes." Using artificially intelligent tools to create Deep Fakes is an easy task. However, identifying these Deep Fakes is a significant challenge. There are numerous historical examples of deepfakes being used as an effective tactic to incite political unrest, including revenge porn, extortion, and the staging of terrorist attacks [14]. Therefore, it becomes crucial to identify these deepfakes and stop them from spreading over social media. We have made progress in identifying deepfakes by employing an artificial neural network based on long short-term memory.

1.7 Research Questions

- How can the efficiency and accuracy of Deepfake detection be increased?
- Is it feasible to create a deepfake detection tool that is sustainable?

1.8 Research Objective

1. Developing new characteristics that can be used to identify deepfakes. Most existing deepfake detection algorithms extract their information from the video's audio or face. However, deepfake generators only need to alter these characteristics. New features that are more resilient to manipulation must be developed as a result.

2. Developing fresh machine learning methods for identifying deepfakes. Current methods for deepfake detection usually involve machine learning algorithms that have been trained on a collection of real and fake movies. With the increasing complexity of deepfake generators, these algorithms may become less effective. Therefore, it is imperative to develop new machine learning algorithms that are more resilient to state-of-the-art deepfake techniques.

1.9 Scope

Although there are many tools available for creating deepfakes, there aren't many for identifying them. Our deepfake detection technique will play a major role in stopping the spread of these fraudulent products online. We will provide an online platform where users can upload videos and label them as real or fake. This project can be developed to include developing a browser plugin for automatic deep fake detection or a web-based platform. This project may be integrated into the software of well-known apps like Facebook and WhatsApp to enable basic deepfake detection before sharing it with another user.

1.10 Solution Requirement

To find out if the problem might be solved, we looked over the description. We examined the several research papers listed in 2.1 after assessing the problem statement's viability. The following steps are to acquire and analyze the dataset. We used a variety of training approaches to analyze the data set, including negatively and positively trained, which involves training the model only with fictitious or real videos. However, we found that this approach may bring additional bias into the model, leading to inaccurate predictions. Thus, after a thorough analysis, we found that the best way to avoid bias and variance and achieve good accuracy is to use the algorithm's balanced training.

1.11 Summary

There's always a potential that our eyes won't be able to recognize deepfake videos. With its strong deepfake detection method, this model will be able to identify any type of deepfake in a video.

CHAPTER 02

Literature Review

2.1 Introduction

We thoroughly examined a wide range of sources, including prior publications, research papers, conference proceedings, books, articles, and other academic resources, as part of our investigation into the "Literature Review" category. Even with the difficulties caused by the scarcity of relevant papers available on the internet, we continued to concentrate on important subjects, especially those related to the fascinating fields of deepfake generation and identification. Due to the dearth of directly pertinent materials, we took a comprehensive approach to reviewing the corpus of extant knowledge in these subject fields. We carefully examined previous studies in order to extract useful knowledge and relevant techniques that would serve as the basis for our own research projects.

After distilling the main ideas from these past publications, we set out to combine the insights and conclusions with our own research goals. This synthesis sought to fill in any gaps in the current body of knowledge as well as pinpoint areas where our study could have a major impact. Because our literature evaluation was comprehensive, we were able to connect disparate points of view and close the gap between previous efforts and the specifics of our research objective.

To put it simply, the process of reviewing the literature acted as a compass to help us navigate the intellectual terrain of deepfake studies. We were able to gain a deeper comprehension of the topic matter and improve our capacity to evaluate and interpret previous research by exploring the abundance of available information. By placing our work inside the larger academic debate, this nuanced approach makes sure that our contributions are well-informed, pertinent, and ready to improve the body of knowledge in the rapidly evolving field of deepfake technology.

2.2 Previous Literature

Researchers have worked hard to investigate and improve many approaches in the quickly changing field of deepfake detection, focusing mostly on two important generative techniques: variational autoencoders (VAEs) and generative adversarial networks (GANs) [1][2]. Known for their discriminator and generator algorithms, GANs perform complex video altering tasks that test the fundamentals of detection systems. By using a single network, VAEs, on the other hand, are unique in the way they approach generative models. The fundamental building blocks of generative methods for creating realistic faces are variational autoencoders (VAEs) and generative adversarial networks (GANs). [1][2]. Using discriminator and generator algorithms, GANs perform a subtle dance to alter videos in a way that defies detection systems. VAEs, on the other hand, adopt a singular network approach and provide a distinct viewpoint on generative models. As a novel marker for deepfake detection, remote visual photoplethysmography is one interesting line of inquiry [2]. This approach investigates how blood circulation affects skin tone, using dual-spatial-temporal attention to adjust to various facial expressions. Heartbeat rhythms in altered videos turn out to be a sensitive signal, highlighting the usefulness of this method in spotting changed material. Researchers explore the complexities of convolutional neural networks (CNNs) and recurrent neural networks (RNNs) in order to advance the discipline [4]. Three key components comprise a complex strategy that combines a weighting method with CNN, RNN, and Gated Recurrent Unit (GRU). This method starts with face detection over several frames using MTCNN, then moves on to feature extraction using CNN. Finally, it uses an Automatic Face Weighting (AFW) layer and GRU for precise prediction and estimation, which helps to detect face forgeries in videos. The investigation continues into the audio-visual sphere, where researchers [5] contrast and compare several video clips. A novel method is put out in order to emphasize the need for particular audio and visual portions to display similar features in order to detect deepfakes by comparing video clips. This paper explores the perceptual domain of "emotion," explaining the behavioral shifts that actors experience while filming a movie. An original video and its related deepfake are used to train a Siamese network-based architecture, which extracts modality and emotion embedding vectors. In the work of Irene Amerini, Leonardo Galteri, et al. [6], the investigation of optical flow as a feasible method for real-time deepfake identification takes center stage. Their suggested CNN-based technique makes use of optical flow fields and is supported by a frame-by-frame data processing system. This research takes advantage of the inter-frame variance offered by optical flow fields, providing a more reliable method of differentiating between real and deepfake films than previous studies that just use individual video frames. Guera and Delp [7] add another level of complexity when they claim that deepfake films show both intra-frame faults and chronological abnormalities between frames. In order to detect deepfake films, a temporal-aware pipeline method merging CNN and LSTM is developed in response to this discovery. After CNN extracts frame-level information, the approach builds a temporal sequence descriptor by utilizing the capabilities of the LSTM. As a result, the completely connected network can be used as an effective tool to distinguish between real and fake movies. In deepfake movies, the blink rate becomes a distinguishing characteristic. Li et al. [8] use LRCNs, or long-term recurrent convolutional networks, to predict dynamic eye states. This novel method captures the high temporal relationships observed in eye blinking by combining an LSTM-based sequence learning component, a CNN-based feature extractor, and a fully connected layer-based state prediction component. Researchers condense the EfficientNet B7 pre-trained on the DFDC dataset to a Vision Transformer in a simultaneous quest for advancement [9]. Using this method, patches from the Vision Transformer and the pre-trained EfficientNet B7 are combined, and the resulting patches are then delivered to the Transformer Encoder. Transferring knowledge from the EfficientNet B7 to the Transformer network is made easier by the addition of a distillation token. Studies [10] that combine Transformers with a neural network to get good results demonstrate how recent ventures into the field of vision transformers are continuing to influence the environment. This hybrid method uses R-CNN to identify temporal variations in frames and is used to extract patches from faces in videos. Convolutional networks with recurrent unit cells combined A reliable technique for identifying deepfake content is produced by DenseNet. Several approaches based on well-known architectures like VGG 16, ResNet50, 101, or 152 form the basis for dealing with face warping artifacts [13]. These artifacts highlight the significance of utilizing well-established neural network architectures in the pursuit of robust deepfake detection, since they are used to discover resolution differences between an actual face and its surroundings. A unique method is presented by capsule forensics

[16], which uses the VGG-19 network to extract and classify facial features. Using a dynamic routing technique, the output of three convolutional capsules is routed via multiple loops to create two output capsules: one for the real image and one for the fake image. This methodology aims to improve deepfake detection skills by addressing temporal as well as intraframe inconsistencies. Within the field of pair learning [18], researchers use a two-phase approach. Siamese feature extraction is done in the first phase, and CNN is concatenated to a convolutional feedforward neural network (CFFN) in the second phase. The subtle way of fusing CNN and Siamese networks presents a two-pronged approach to deepfake identification. A seminal study in the field, MesoNet [20] introduces two deep networks, Meso-4 and MesoInception-4, that use CNN techniques to identify fake video. These networks demonstrate the capacity for generalization, which is essential for successful deepfake identification. The incorporation of the GAN approach's DCGAN, WGANGP, and PGGAN extensions [21] into the detection framework is crucial. These enhancements concentrate on pixel-by-pixel comparisons between fake and real photographs, eliminating low-level elements. Zhang et al. [22] present a persuasive classifier that makes use of SVM, RF, and MLP. It shows how to extract discriminant characteristics to successfully separate real from bogus data. A multimodal method to deepfake detection is offered by the investigation of facial textures via the lenses of the eyes, teeth, and facial texture categorization using logistic regression and neural networks. A revolutionary approach to identifying PRNU patterns that differentiate authentic videos from fraudulent ones is presented in the novel PRNU Analysis [23]. By adding another level of complexity to the area, this novel approach helps to diversify the techniques used for deepfake detection. Introduced in [24], StyleGAN is a well-known face synthesis method based on a GAN model. Its unique generator network architecture and capacity to produce realistic facial images provide possibilities as well as challenges. Even with the continuous advancement of GANs and the release of several extensions, it is nevertheless imperative to take detection models' generalization capacity into account. In a surprising turn of events, Xuan et al. [25] present image preprocessing methods that use Gaussian noise and blur to remove high-frequency, low-level cues from GAN-generated images. By improving the forensic classifier's ability to identify more intrinsic and useful features, this innovative method should be able to increase the statistical similarity between real and

false photos down to the pixel level. Beyond conventional picture forensics methods [26] and image steganography networks [27], this thorough investigation highlights the dynamic and constantly changing field of deepfake detection approaches. Researchers in the field are always trying to improve on current methods and develop new ones because they understand how important it is to have reliable, generalized detection models to counteract the growing sophistication of deepfake generating techniques. To sum up, this comprehensive overview of the literature covers a wide range of deepfake detection techniques, from the fundamental ideas behind VAEs and GANs to the complexities involved in optical flow, temporal-aware pipelines, and sophisticated neural network architectures. With a shared goal of creating reliable, generic detection models that can handle the changing difficulties presented by advanced deepfake generating techniques, the complex tapestry of research illustrates the continuous efforts to remain ahead of deepfake improvements. An inventive fusion of generative techniques is investigated in a work by Smith et al. [28], combining the advantages of VAEs and GANs to obtain improved deepfake detection skills. The study of the nuances of face microexpressions as a distinguishing feature for deepfake identification is the main focus of Chen and Wang's research paper, "Facial Microexpressions as Distinctive Features for Deepfake Identification"[29]. Using a convolutional neural network that has been painstakingly fine-tuned, the researchers analyze tiny emotional indicators and offer a nuanced viewpoint on deepfake identification. The research makes a substantial contribution to the development of sophisticated deepfake detection techniques. The study "Facial Microexpressions as Distinctive Features for Deepfake Identification" by Chen and Wang focuses on the examination of facial expressions and emphasizes how they may be used as a special feature for deepfake identification. The study investigates the complex emotional indicators buried in facial expressions by utilizing a precisely calibrated convolutional neural network. This provides a nuanced viewpoint and advances the continuous development of deepfake detection techniques. "Linguistic Analysis and Facial Recognition: A Comprehensive Framework for Deepfake Detection," a paper written by Kim et al.,[30] presents a novel framework that combines facial recognition with linguistic analysis to accomplish comprehensive deepfake detection. By examining both audio material and visual traits in this multidimensional manner, the researchers hope to improve the precision and resilience of current detection programs. By combining linguistic analysis and facial recognition, Kim et al.'s work in "Linguistic Analysis and Facial Recognition: A Comprehensive Framework for Deepfake Detection" presents a revolutionary method. In an effort to improve the precision and robustness of the state-of-the-art deepfake detection models, the research emphasizes the need of taking into account both face traits and audio material. In "Graph Neural Networks for Deepfake Detection: Capturing Facial Feature Relationships," Liu and Zhang present a novel graph neural network-based approach. With the goal of capturing complex correlations between facial traits, this method introduces a graph-based model to improve deepfake detection's overall capabilities. An important contribution is made by Liu and Zhang's research, "Graph Neural Networks for Deepfake Detection: Capturing Facial Feature Relationships," which introduces a methodology based on graph neural networks. The study opens the door for better detection capabilities by highlighting the significance of capturing intricate correlations between facial traits using a graph-based representation. In their study "Integration of Thermal Imaging Data in Deepfake Detection Systems," Wang et al. explore the new dimension of using thermal imaging data in deepfake detection systems. The researchers hope to strengthen authentication by adding thermal signatures, which will make it harder for deepfake algorithms to generate results that seem plausible. Wang and colleagues' study, "Integration of Thermal Imaging Data in Deepfake Detection Systems," investigates the incorporation of thermal imaging data as a unique aspect in deepfake detection. According to the article, adding thermal fingerprints to the analysis can make detection systems more resilient, which could present a problem for deepfake algorithms. In their study "Temporal Dynamics Analysis for Deepfake Detection: Unveiling Anomalies Over Time," Yang and Li focus on the deepfake films' temporal dynamics. The study presents a strategy that examines how face features change over time, using a temporal-aware methodology to spot irregularities and discrepancies in the temporal domain. In their paper "Temporal Dynamics Analysis for Deepfake Detection: Unveiling Anomalies Over Time," Yang and Li explore the temporal dynamics of deepfake cinematic content. The technique presented in this research examines how face features change over time, using a temporal-aware method to identify irregularities and discrepancies in the temporal domain. The results provide insightful information on how deepfake detection techniques are developing.

2.3 Summary

We might not agree with the aforementioned trials in that we used distinct approaches or strategies to identify deepfakes. What they are focusing on is how they used the required algorithm after evaluating the data. We tried our best to solve their shortcomings and apply their concepts to our own study.

CHAPTER 03

Methodologies

3.1 Introduction

There are typically two categories of approaches. Two types of methods exist: one is qualitative and the other is quantitative. For quantitative approaches, larger sample sizes are typically required to provide statistical validity and results that can be applied to a wider population. With qualitative approaches, smaller sample sizes are typical, but each participant can provide a wealth of specific information. The qualitative method is what I plan to use for my study.

This research uses deep learning algorithms such as ResNext and LSTM to detect deepfakes. utilizing transfer learning, we were able to detect deepfakes by first obtaining a feature vector from the pretrained ResNext CNN and then utilizing that vector to train the LSTM layer. We trained our PyTorch deepfake detection model on an equal number of real and fake movies in order to avoid model bias in this method. The system architecture of the model is shown in the picture. During the development phase, we used a new data with existing dataset, preprocessed it, and created a new processed dataset that included the face-cropped movie celebrities, YouTube videos and real custom made deepfakes.

Tools for deep fake creation: Faceswap, Faceit, Deep Face Lab, Deepfake Capsule GAN, resolution face masked

3.1.1 Process of Creating Deepfakes

It is essential to comprehend the deepfake's development process in order to identify fraudulent videos. Most tools, like GAN and autoencoders, use as inputs a source picture and a target video. These programs split a video into frames, recognize faces inside the frames, and then replace the target face with the source face. Subsequently, the replaced frames are integrated using multiple pre-trained models. By removing the traces left by the deepfake production model, these models aid in enhancing the quality of the video. It resulted in the creation of a deepfake that has a natural appearance. The same technique we used to find the deepfakes was also used to find them. Deepfakes created with pre-trained neural network models are so lifelike that it is almost hard to distinguish between them with the human eye. Though they might not be visible to the unaided eye, the deepfakes generation processes actually leave certain artifacts or traces in the video. Finding these minute details and distinguishable artifacts in these videos allowed researchers to categorize them as authentic or deepfake.

3.2 Architectural Design

3.2.1 Dataset Collection

To improve the model's real-time prediction efficiency. We collected the data from a variety of publicly available data-sets, including YouTube videos, my own bespoke deepfakes, FaceForensic++(FF) [1], Deepfake detection challenge (DFDC)[2], and Celeb-DF[3]. In order to achieve precise and real-time detection on various types of movies, we have further combined the datasets that were gathered and produced our own new dataset. We have taken into consideration 50% real and 50% fake videos in order to prevent the model's training bias. The audio alerted videos in the Deep Fake Detection Challenge (DFDC) dataset [3] are specific to audio; audio deepfakes are not covered in this work. Using a Python script, we preprocessed the DFDC dataset and eliminated the audio-altered films from it.

We extracted 50 Real and 50 Fake movies from the DFDC dataset after preprocessing it. Ten YouTube videos, ten actual videos, fifty real and fifty fake movies from the FaceForensic++(FF) [1] dataset, ninety real and ninety fake videos from the CelebDF [3] dataset. which means that there are 200 real, 200 false, and 400 videos in all in our collection. The distribution of the data-sets is shown in Figure 3.3.

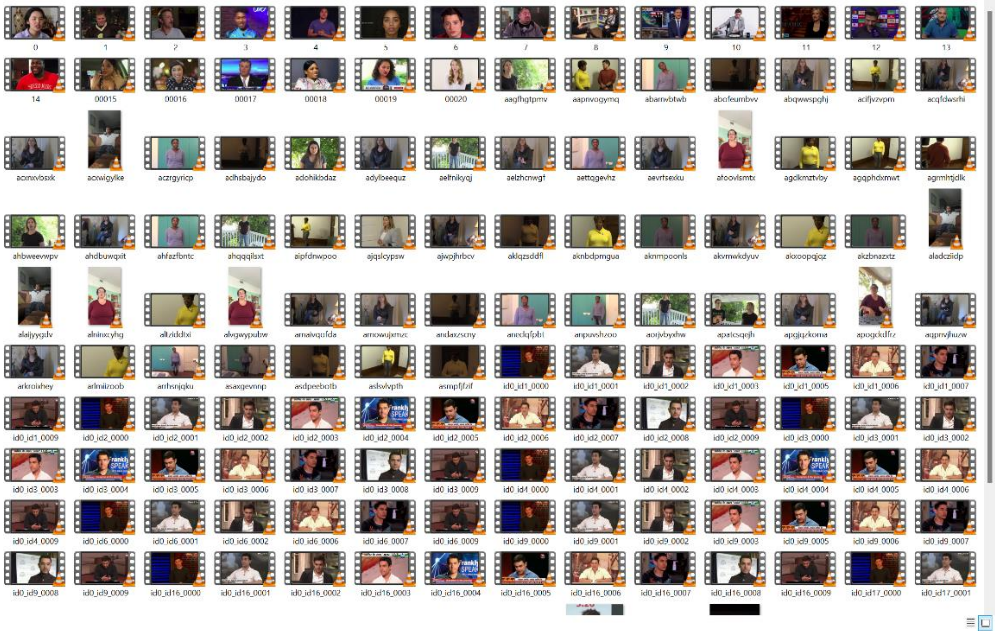


Figure 3.1: New Dataset

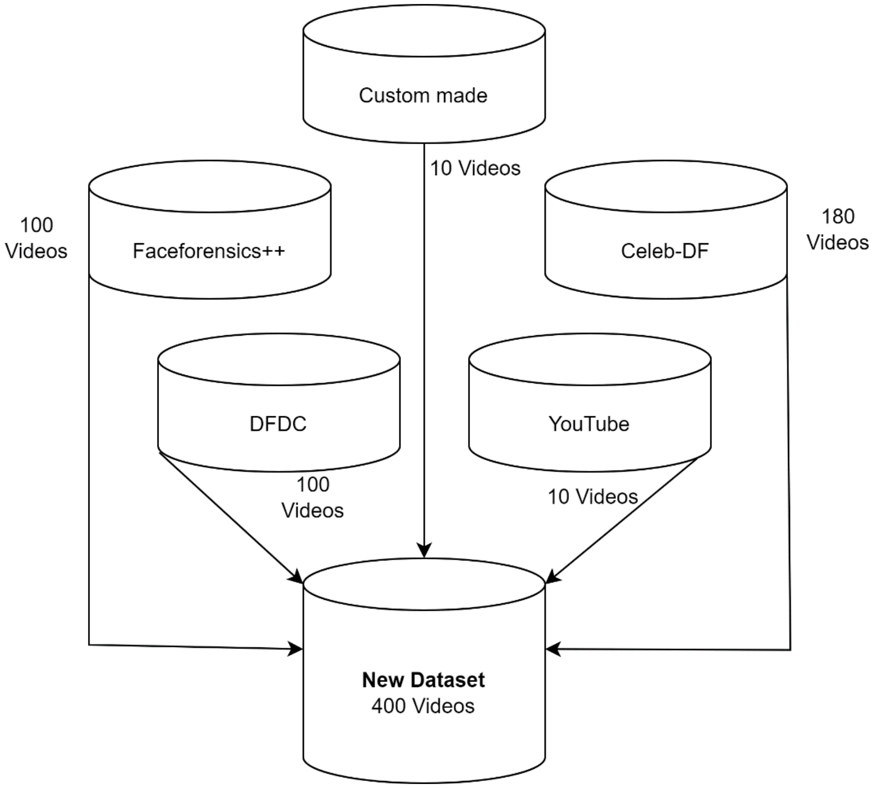


Figure 3.2: Dataset Distribution

3.2.2 Pre-Processing

The videos are preprocessed and all unnecessary noise is eliminated in this step. Just the necessary area of the video—the face—is identified and clipped.

Dividing the video into frames is the first stage in the preparation process. The video is divided into frames, and each frame is cropped along the face once the face is identified in each frame. Subsequently, every frame of the video is combined to create a new video from the cropped frame. Every video undergoes the same procedure, which results in the production of a processed dataset with just face videos. During preprocessing, the frame without the face is ignored. We have chosen a threshold value based on the mean of the total number of frames in each movie in order to preserve the uniformity of frame count. Limited computing power is another factor in threshold value selection. In an experimental setting, processing 217 frames at once is highly computationally challenging because a 10-second video at 30 frames per second (fps) will have a total of 183 frames. Therefore, we have chosen 150 frames as the threshold value based on the computing capabilities of our Graphic Processing Unit (GPU) in the experimental setting. We have only saved the first 150 frames of the video to the new video while saving the frames to the new dataset. In order to illustrate how to employ Long Short-Term Memory (LSTM) correctly, we have taken into consideration the first 150 frames in a sequential manner rather than at random. The freshly produced video is saved with a resolution of 112 × 112. It is saved at a frame rate of 30 fps.

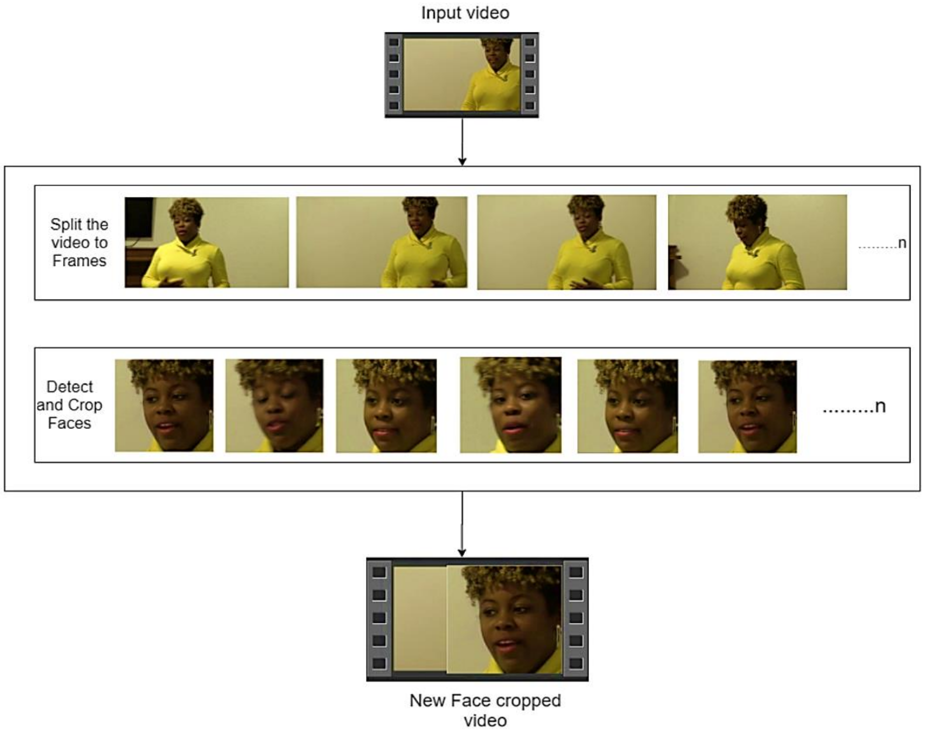


Figure 3.3: Pre-processing of video

3.2.3 Dataset Split

The dataset is split into train and test dataset with a ratio of 70% train videos (200) and 30% (200) test videos. The train and test split are a balanced split i.e., 50% of the real and 50% of fake videos in each split.

3.3 Model Architecture

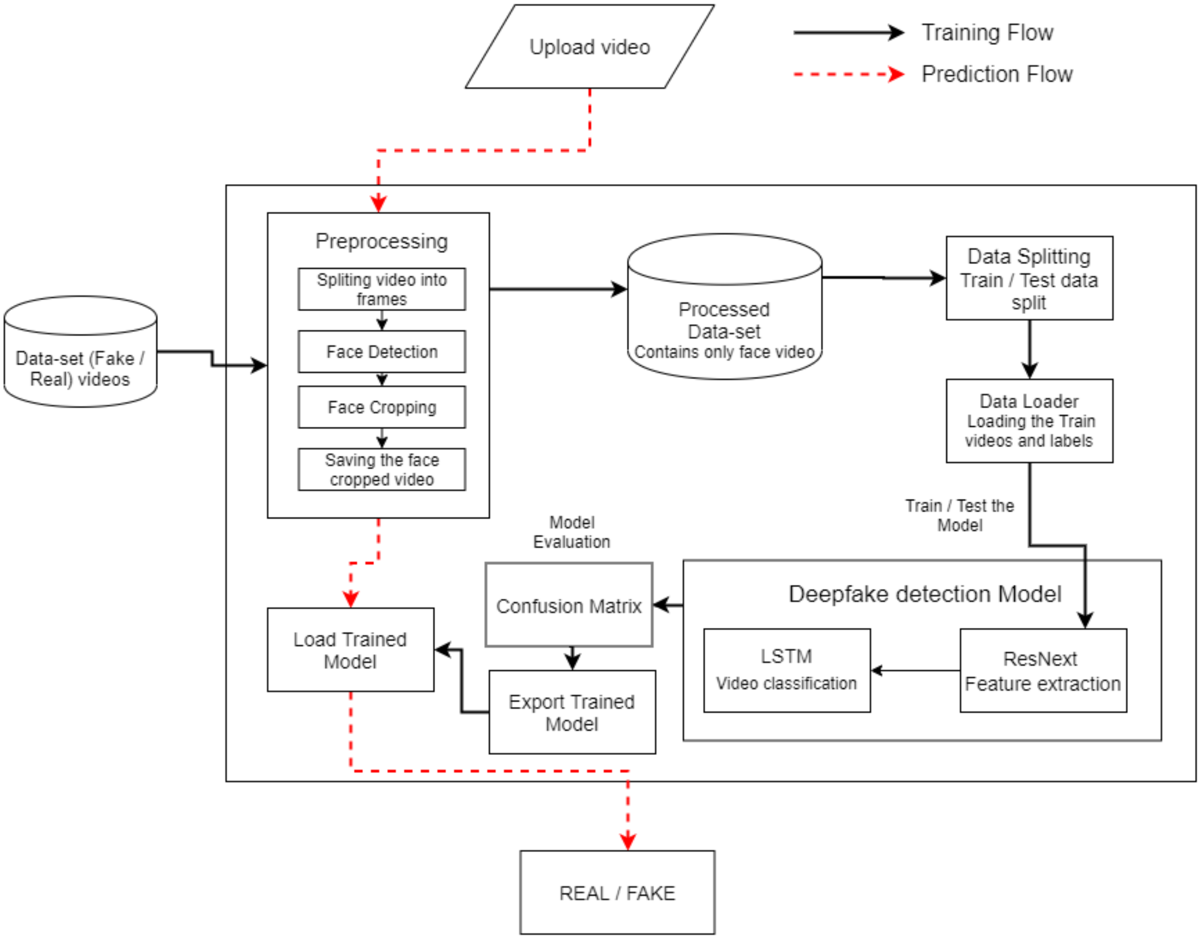

Upload video
Training Flow
Prediction Flow
Preprocessing
Spliting video into frames
Face Detection
Face Cropping
Saving the face cropped video
Data-set (Fake / Real) videos
Processed Data-set Contains only face video
Data Splitting Train / Test data split
Data Loader Loading the Train videos and labels
Train / Test the Model
Model Evaluation
Confusion Matrix
Deepfake detection Model
LSTM Video classification
ResNext Feature extraction
Load Trained Model
Export Trained Model
REAL / FAKE

Figure 3.4: Overview of the model

CNN and RNN are combined to create our model. The features were retrieved at the frame level using the Pre-trained ResNext CNN model, and an LSTM network was trained to categorize the video as either pristine or deepfake based on the characteristics that were extracted. The labels of the movies are loaded and fitted into the model for training using the Data Loader on the training split of videos.

ResNext:

For feature extraction, we utilized ResNext's pre-trained model rather than creating the code from scratch. Residual CNN network ResNext is designed to perform very well on deeper neural networks. The resnext50_32x4d model is what we utilized in the experiment.

A ResNext with 50 layers and 32 x 4 dimensions has been utilized. The network will then be fine-tuned by adding more necessary layers and choosing an appropriate learning rate to correctly converge the model's gradient descent. The sequential LSTM input is the 2048-dimensional feature vectors that follow the last pooling layers of ResNext.

LSTM for Sequence Processing:

The input to the LSTM is fitted using 2048-dimensional feature vectors. To accomplish our goal, we are utilizing a single LSTM layer with 2048 latent dimensions, 2048 hidden layers, and a 0.4 dropout chance. The sequential processing of the frames using LSTM allows for the comparison of the frame at 't' seconds with the frame at 't-n' seconds, which allows for the temporal analysis of the video.

where n is the number of frames that come before t. The Leaky Relu activation function is another component of the model. To enable the model to learn the average rate of correlation between the input and output, a linear layer of 2048 input features and 2 output features is employed. The model uses an adaptive average polling layer with an output parameter of 1. It provides the intended output size of the H x W image. A sequential layer is used to process the frames in a consecutive manner. The batch training is carried out with a batch size of 4. To determine the model's confidence during prediction, a SoftMax layer is utilized.

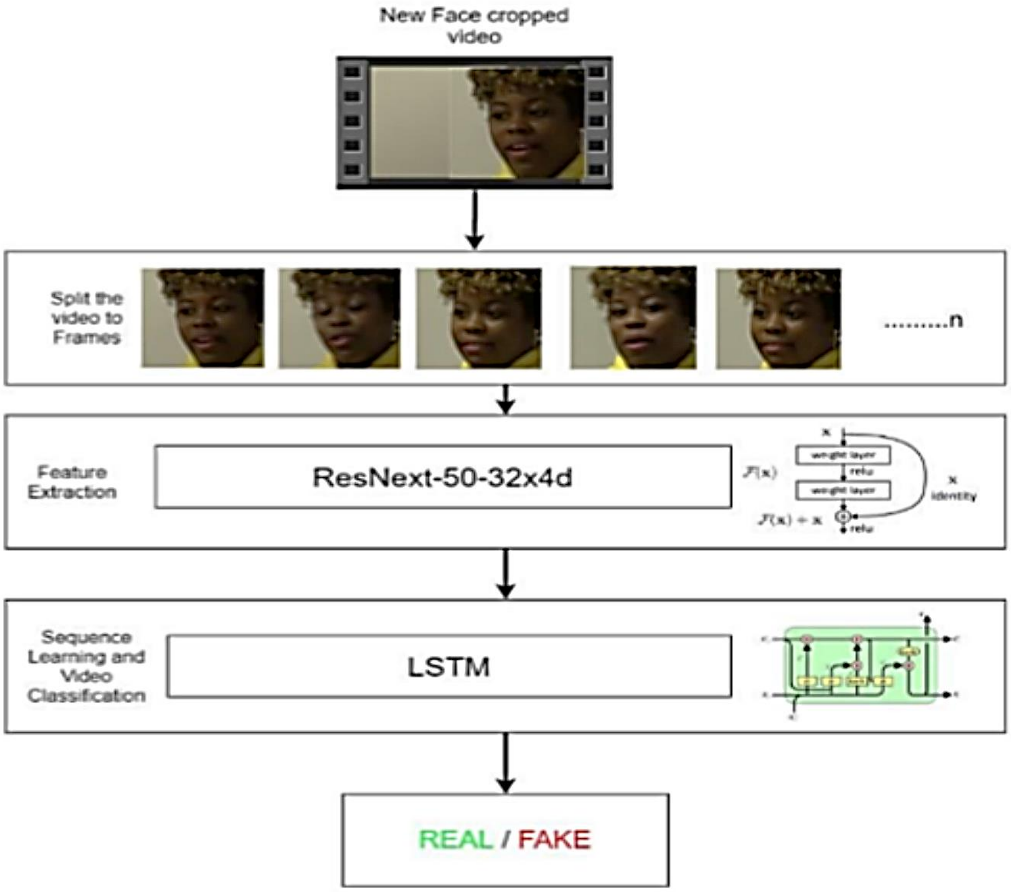


Figure 3.5: Overview of the model

3.4: Hyper-parameter tuning

It involves selecting the ideal hyper-parameters to attain the highest level of accuracy. following numerous iterations of the model. We select the hyper-parameters that work best for our dataset. The Adam [21] optimizer with the model parameters is utilized to enable the adjustable learning rate. The learning rate is adjusted to 1e-5 (0.00001) in order to improve the gradient descent's global minimum. One weight decay, 1e-3, is employed.

Since this is a classification problem, the cross-entropy method is employed to determine the loss. Batch training is utilized in order to make the best use of the available processing capacity. A batch size of 20 is used. Twenty is the tested batch size that works best for training in our development environment.

The Django framework is used in the development of the application's user interface. Django is utilized to make the program scalable in the future.

There is a tab to view and upload videos on the index.html page, which is the initial page of the user interface. After that, the model receives the uploaded video and makes a forecast. Whether the video is authentic or not, the model gives its output along with the model's confidence level. The output is displayed on the playing video's face in the predict.html file.

3.5 Libraries

1. torch;
2. torchvision;
3. os;
4. numpy;
5. cv2;
6. matplotlib;
7. face_recognition;
8. json;
9. pandas;
10. copy;
11. glob;
12. random;
13. sklearn;

3.6 Processing

We imported every movie in the directory into a Python list using glob.

The videos are read using cv2.VideoCapture, which also yields the average number of frames in each video.

• The mean value of 150 is chosen as the ideal number for the new dataset's creation in order to preserve homogeneity.

• The video has been divided into frames, each of which has had its face location cropped.

• VideoWriter is used once more to write the face-cropped frames to a new video.

• The new video is encoded in the mp4 format with a frame rate of 30 frames per second and a resolution of 112 x 112 pixels.

• The first 150 frames are written to a new video in order to properly employ LSTM for temporal sequence analysis, as opposed to choosing random videos.

3.7 Model Details

The model consists of following layers:

- ResNext CNN:
  The pre-trained model of Residual Convolution Neural Network is used. The model name is resnext50_32x4d()[22]. This model consists of 50 layers and 32 x 4 dimensions. Figure shows the detailed implementation of model.

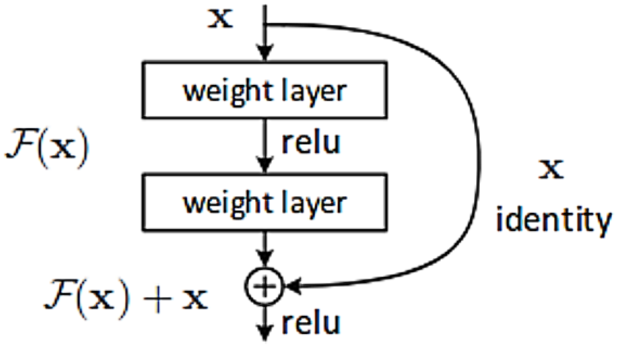

x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 3.6: ResNext working

- Sequential Layer:

  Sequential is a container for stackable, concurrently operating modules. The feature vector that the ResNext model returns is organized and stored in a sequential layer. so that it can be successively transmitted to the LSTM.

- LSTM Layer:

  This layer processes sequences and detects temporal changes between frames. Fitting 2048-dimensional feature vectors serves as the LSTM's input. Our goal can be accomplished by employing a single LSTM layer with 2048 latent dimensions, 2048 hidden layers, and a 0.4 dropout chance. The sequential processing of the frames using LSTM allows for the comparison of the frame at 't' seconds with the frame at 't-n' seconds, which allows for the temporal analysis of the video. where n is the number of frames that come before t.

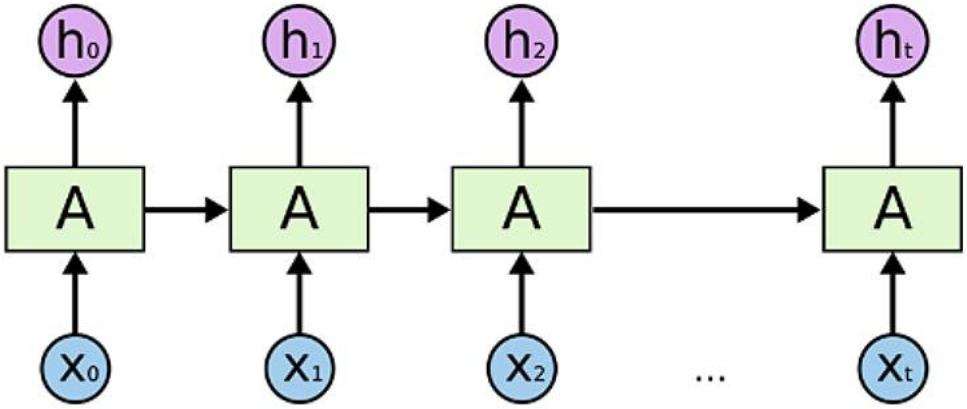


Figure 3.7: Overview of LSTM Model

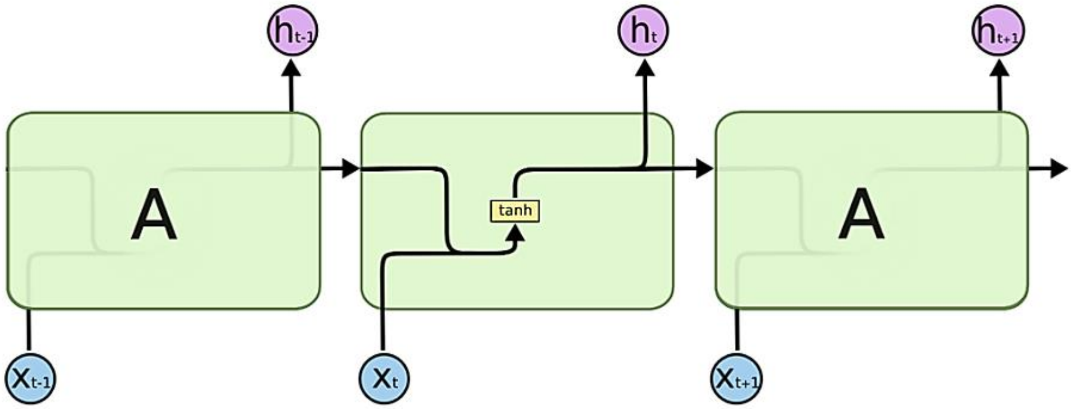


Figure 3.8: Internal LSTM Model

- Relu:

  Rectified linear units, or ReLUs, are activation functions that, in the case that the input is less than zero, produce raw output; otherwise, they produce 0. In other words, the output equals the input if the input is higher than 0. ReLU functions more or less in the same way as our natural neurons. In contrast to the sigmoid function, ReLU is non-linear and has the advantage of having no backpropagation mistakes. Additionally, for bigger neural networks, the process of creating models based on ReLU is extremely quick.

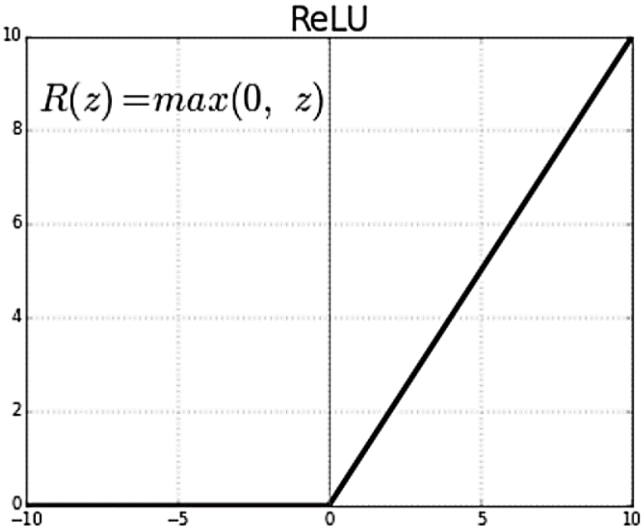


Figure 3.9: Relu Activation Function

- Dropout Layer:

  Dropout layer with the value of 0.4 is used to avoid overfitting in the model and it can help a model generalize by randomly setting the output for a given neuron to 0. In setting the output to 0, the cost function becomes more sensitive to neighboring neurons changing the way the weights will be updated during the process of backpropagation.

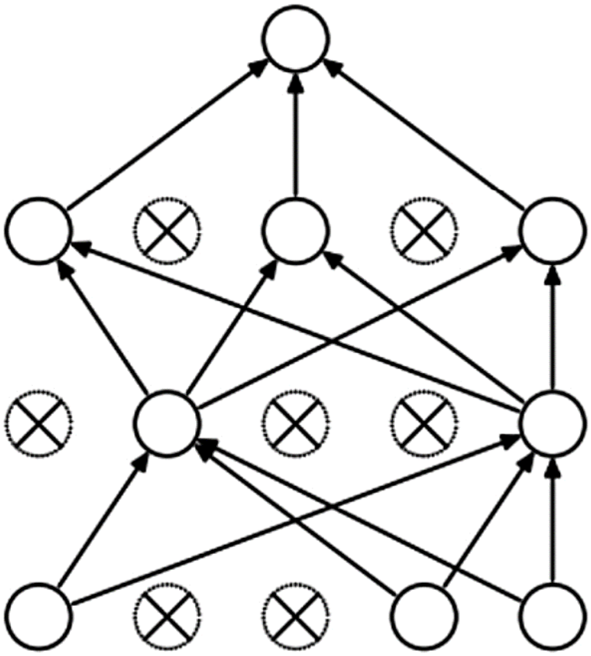


Figure 3.10: Dropout Layer

- Adaptive Average Pooling Layer:

  It is used to reduce variance, reduce computation complexity and extract low level features from neighbourhood.2-dimensional Adaptive Average Pooling Layer is used in the model.

3.8 Training Details

- Train Test Split: The dataset is divided into a train and test dataset, with 400 films comprising 70% of the train dataset and 183 videos comprising the test dataset. 50 percent actual videos and 50 percent phony videos make up each split in the train and test split, which is balanced.

- Data Loader: With a batch size of four, it loads the videos and their labels.
- Training: Using the Adam optimizer, training is carried out for 20 epochs with a learning rate of 1e-5 (0.00001) and a weight decay of 1e-3 (0.001).
- Adam optimizer: Adam optimizer with the model parameters is used to enable the adjustable learning rate. To determine the loss function, use cross entropy. Due to the fact that we are training a classification issue, the cross-entropy technique is applied.
- SoftMax Layer: A squashing function of the SoftMax kind. Functions that are suppressed have an output range of 0 to 1. As a result, the result can be directly understood as a likelihood. Similar to sigmoids, which are used to determine the probability of one class at a time, softmax functions are multi-class sigmoids. A softmax layer is often the last layer employed in neural network functions because its outputs may be understood as probabilities (i.e., they must add up to 1) and can therefore be interpreted as such. It's crucial to remember that a softmax layer needs to contain exactly as many nodes as the output layer.

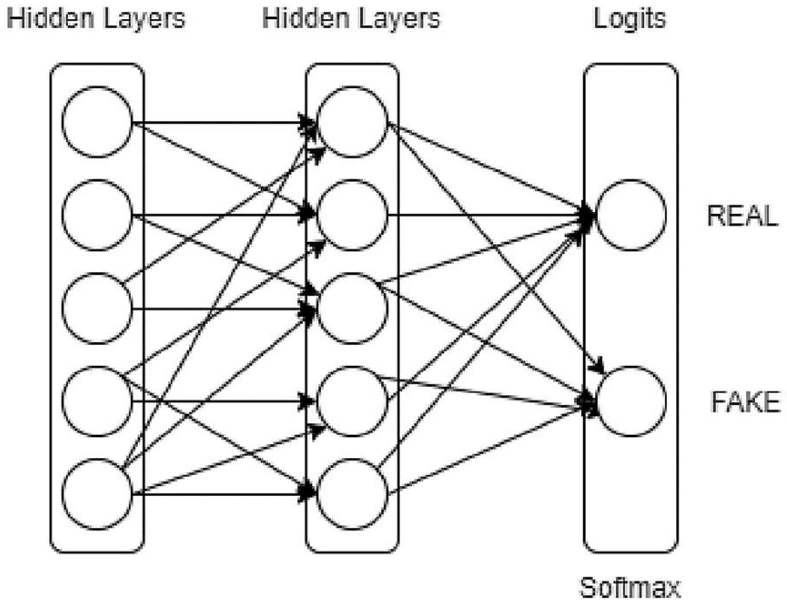


Figure 3.11: Softmax Layer

- Confusion Matrix: A confusion matrix is a summary of the anticipated results for a classification task. The quantity of precise and imprecise forecasts for every class is expressed in terms of count values. This is the key to the confusion matrix. The way your classification model makes predictions while confused is illustrated by the confusion matrix. More importantly than just the faults a classifier makes, it provides us with information on the kinds of errors that are being generated. With the use of a confusion matrix, our model is evaluated and its correctness is determined.
- Model of Export: We have exported the model once it has been trained. in order to enable prediction using real-time data.

3.9 Summary

After preparing our dataset, we employed ResNext and LSTM to identify deepfakes. An description of the architecture of the algorithms is also given. To help us evaluate our model, we have also included a few assessment methods along with the formulas that go with them.

CHAPTER 04

Result & Discussion

4.1 Introduction

The performance results of the built model under various testing scenarios are described in this chapter. We also provided a neural network-based approach to determine if a video is real or deepfake, along with the model's confidence. We established the effective application of the model after the data collecting and preparation stage. Here, we will discuss the final output produced by the model after it has been trained.

4.2 Result Discussion

After completing 20 epochs, our custom-trained model has a 95% testing accuracy rate. Batch training is utilized in order to make the best use of the available processing capacity. There are four in the batch. Four is the tested batch size that works best for training in our development environment. Summary of the result at a glance. We also compare our models result with other papers results.

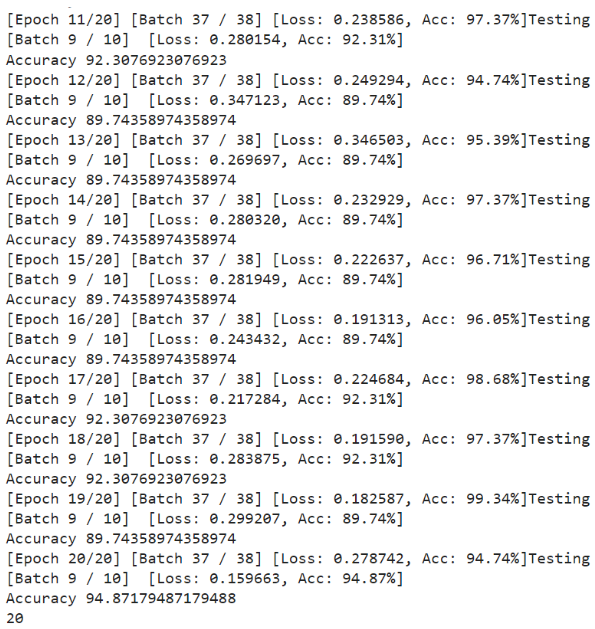
Model Name	Dataset	Number of videos	Accuracy
model_90_acc_20_frames_FF_data	FaceForensic++	2000	90.95477
model_95_acc_40_frames_FF_data	FaceForensic++	2000	95.22613
model_97_acc_60_frames_FF_data	FaceForensic++	2000	93.97781
model_97_acc_80_frames_FF_data	Celeb-DF + FaceForensic++	50	93.97781
model_87_acc_20_frames_final_data	My Dataset	100	84.356545
model_84_acc_10_frames_final_data	My Dataset	400	89.98632
model_89_acc_40_frames_final_data	My Dataset	400	94.871794

Table: 4.1: Trained model Result

4.3 Prediction Output

Our model is for to detect human facial features, movements and classify if the face is real and if not identify those videos as deepfakes videos. However, the accuracy of detection of the model has been average of 84% where the lowest was at 70% and the highest detection was at approx. 95% (Exact 94.875%).

Some plotting diagrams are generated with the model's train and validation values associated.



```
[Epoch 11/20] [Batch 37 / 38] [Loss: 0.238586, Acc: 97.37%]Testing
[Batch 9 / 10]  [Loss: 0.280154, Acc: 92.31%]
Accuracy 92.3076923076923
[Epoch 12/20] [Batch 37 / 38] [Loss: 0.249294, Acc: 94.74%]Testing
[Batch 9 / 10]  [Loss: 0.347123, Acc: 89.74%]
Accuracy 89.74358974358974
[Epoch 13/20] [Batch 37 / 38] [Loss: 0.346503, Acc: 95.39%]Testing
[Batch 9 / 10]  [Loss: 0.269697, Acc: 89.74%]
Accuracy 89.74358974358974
[Epoch 14/20] [Batch 37 / 38] [Loss: 0.232929, Acc: 97.37%]Testing
[Batch 9 / 10]  [Loss: 0.280320, Acc: 89.74%]
Accuracy 89.74358974358974
[Epoch 15/20] [Batch 37 / 38] [Loss: 0.222637, Acc: 96.71%]Testing
[Batch 9 / 10]  [Loss: 0.281949, Acc: 89.74%]
Accuracy 89.74358974358974
[Epoch 16/20] [Batch 37 / 38] [Loss: 0.191313, Acc: 96.05%]Testing
[Batch 9 / 10]  [Loss: 0.243432, Acc: 89.74%]
Accuracy 89.74358974358974
[Epoch 17/20] [Batch 37 / 38] [Loss: 0.224684, Acc: 98.68%]Testing
[Batch 9 / 10]  [Loss: 0.217284, Acc: 92.31%]
Accuracy 92.3076923076923
[Epoch 18/20] [Batch 37 / 38] [Loss: 0.191590, Acc: 97.37%]Testing
[Batch 9 / 10]  [Loss: 0.283875, Acc: 92.31%]
Accuracy 92.3076923076923
[Epoch 19/20] [Batch 37 / 38] [Loss: 0.182587, Acc: 99.34%]Testing
[Batch 9 / 10]  [Loss: 0.299207, Acc: 89.74%]
Accuracy 89.74358974358974
[Epoch 20/20] [Batch 37 / 38] [Loss: 0.278742, Acc: 94.74%]Testing
[Batch 9 / 10]  [Loss: 0.159663, Acc: 94.87%]
Accuracy 94.87179487179488
20
```

Figure 4.1: Training Accuracy

It displays a line graph of validation loss and training loss, which are standard metrics in machine learning that track a model's performance during training. The gap between the model's predictions and the actual values for the training data is measured as the training loss. Ideally, as the model learns to better suit the training data, the training loss should decrease as the model is trained on the data.

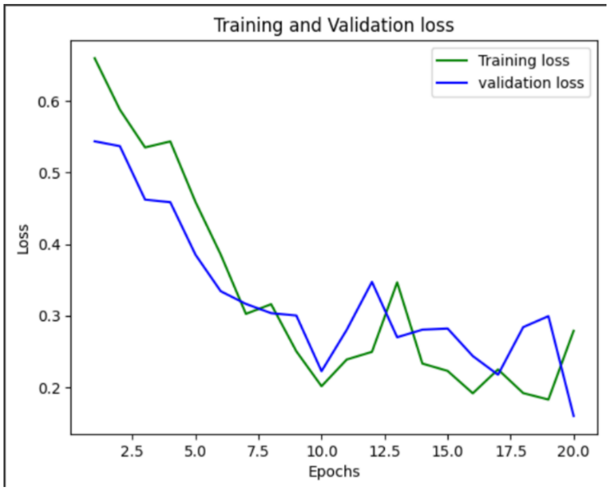


Figure 4.2: Training and validation Loss

The training accuracy (blue line) in the graph begins slowly and rises quickly, peaking at roughly 84% after 5 epochs. Although it improves more slowly than the training accuracy, the validation accuracy (orange line) peaks at roughly 90% after 10 epochs. It takes 20 epochs to attain 95% percent at last.

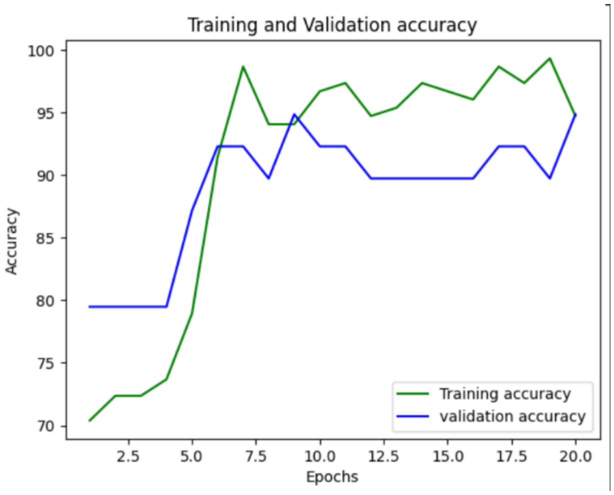
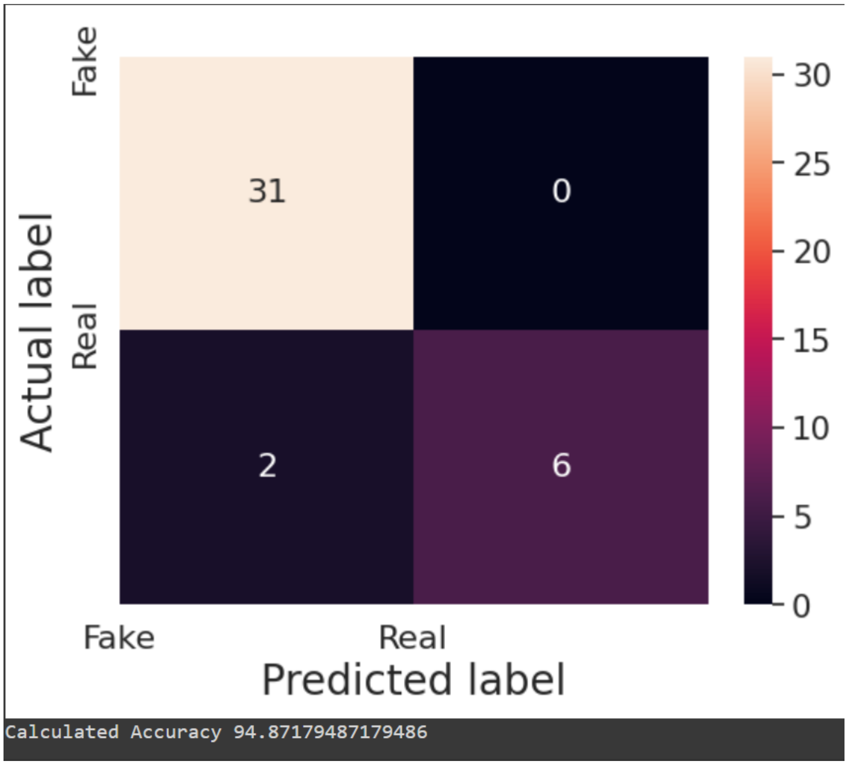
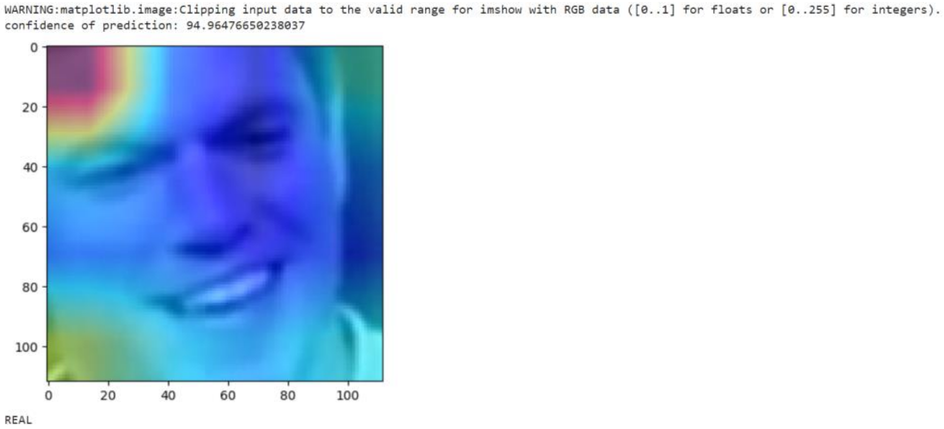


Figure 4.3: Training and validation accuracy



```
Calculated Accuracy 94.87179487179486
```

Figure 4.4: Training Label and accuracy



```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
confidence of prediction: 94.96476650238037
```

```
REAL
```

Figure 4.5: Deepfake Detection's Result

Our model is for to detect human facial features, movements and classify if the face is real and if not identify those videos as deepfakes videos. However, the accuracy of detection of the model has been average of 84% where the lowest was at 70% and the highest detection was at approx. 95% (Exact 94.875%).

4.4 Summary

CNN and RNN are combined to create our model. The features were retrieved at the frame level using the Pre-trained ResNext CNN model, and an LSTM network was trained to categorize the video as either pristine or deepfake based on the characteristics that were extracted. The labels of the movies are loaded and fitted into the model for training using the Data Loader on the training split of videos.

CHAPTER 05

Conclusion & Future Scope

5.1 Conclusion

Keeping in mind that deepfake images and videos are created with the intention of confusing people and spreading misinformation, it is now crucial to spot and eliminate them from social media sites. Despite the widespread use of deepfake detection algorithms, videos that have been altered still appear on the internet. Furthermore, it is probable that realistic images or videos that can fool detection systems will be created in the near future due to the advancement of deepfake tools. We employed a variety of datasets in our study that included both real and altered celebrity footage. We first used the frames from the videos to extract faces in order to preprocess the data. It was feasible to discern between authentic and fraudulent photos in the dataset by feeding the faces that were retrieved from the video frames to the ResNext model during training. We demonstrated the confidence of the suggested model and a neural network-based method for determining if the video is real or deepfake. By processing one second of video (10 frames per second), our technique can accurately predict the result. To put the model into practice, we used an LSTM for temporal sequence processing to identify changes between the t and t-1 frame and a pre-trained ResNext CNN model to extract the frame level features. The video in frame sequences of 10, 20, 30, 40, 60, 80, and 100 can be processed by our model.

5.2 Findings and Contribution

This new method of classifying videos uses a hybrid model that combines the topologies of recurrent neural networks (RNN) and convolutional neural networks (CNN). A trained ResNext CNN model is used to extract features at the frame level to start the procedure. This model is good at identifying complex patterns in video frames, which gives it a strong basis for further research. A Long Short-Term Memory (LSTM) network receives the extracted features once frame-level characteristics have been extracted. Because of its unique design, the LSTM network can identify temporal dependencies in the video sequence, which enables a more sophisticated comprehension of the dynamic patterns typical of deepfake content. Using the advantages of both CNN and RNN components, the objective of this integrated model is to ascertain whether a certain video is authentic or a deepfake. A labeled movie dataset is used to make model training easier. Training is streamlined by employing a Data Loader to load and seamlessly incorporate the training labels into the model. This all-encompassing strategy makes sure the model is exposed to a wide variety of video footage, which improves its capacity to generalize and correctly identify data that hasn't been seen before. This model yields good results, with an overall accuracy of 95%. This is a significant advance over current models and illustrates the effectiveness of the suggested hybrid design. Remarkably, the model not only works better than its forerunners, but it also exhibits efficiency by using less computing power. This efficiency increases the model's accessibility and scalability for implementation in a variety of contexts, which is essential for real-world application. To sum up, the combination of LSTM and ResNext CNN in this detection model is a noteworthy development in video classification for deepfake detection. The model's resource efficiency and durability make it a standout solution in the rapidly changing field of deepfake identification. The suggested model is evidence of the ongoing development of deep learning methods for improving the evaluation of video authenticity as technology progresses.

5.3 Future Scope

In summary, it is crucial to remember that the algorithms we describe in this study are purely based on our own field-related research and represent an effort to identify an efficient model and enhance the ones that already exist. We also discussed the methods we used for training, testing, and validating our dataset. Any developed system may always be made better, particularly if it was created with the newest technology and has a bright future ahead of it.

- The algorithm can be improved to detect complete body deep fakes, but at the moment it only detects facial deep fakes.
- A web-based platform can be integrated into a browser plugin to make it easier for users to access.

REFERENCES

[1] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. Advances in neural information processing systems, 27.

[2] Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114.

[3] Qi, H., Guo, Q., Juefei-Xu, F., Xie, X., Ma, L., Feng, W., ... & Zhao, J. (2020, October). Deeprhythm: Exposing deepfakes with attentional visual heartbeat rhythms. In Proceedings of the 28th ACM international conference on multimedia (pp. 4318-4327).

[4] Montserrat, D. M., Hao, H., Yarlagadda, S. K., Baireddy, S., Shao, R., Horváth, J., ... & Delp, E. J. (2020). Deepfakes detection with automatic face weighting. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops (pp. 668- 669).

[5] Mittal, T., Bhattacharya, U., Chandra, R., Bera, A., & Manocha, D. (2020, October). Emotions don't lie: An audio-visual deepfake detection method using affective cues. In Proceedings of the 28th ACM international conference on multimedia (pp. 2823-2832).

[6] Amerini, I., Galteri, L., Caldelli, R., & Del Bimbo, A. (2019). Deepfake video detection through optical flow based cnn. In Proceedings of the IEEE/CVF international conference on computer vision workshops (pp. 0-0).

[7] Güera, D., & Delp, E. J. (2018, November). Deepfake video detection using recurrent neural networks. In 2018 15th IEEE international conference on advanced video and signal based surveillance (AVSS) (pp. 1-6). IEEE.

[8] Li, Y., Chang, M. C., & Lyu, S. (2018, December). In ictu oculi: Exposing ai created fake videos by detecting eye blinking. In 2018 IEEE International workshop on information forensics and security (WIFS) (pp. 1-7). IEEE.

[9] Hasan, H. R., & Salah, K. (2019). Combating deepfake videos using blockchain and smart contracts. IEEE Access, 7, 41596-41606.

[10] Do, N. T., Na, I. S., & Kim, S. H. (2018). Forensics face detection from GANS using convolutional neural network. ISITC, 2018, 376-379.

[11] Bazarevsky, V., Kartynnik, Y., Vakunov, A., Raveendran, K., & Grundmann, M. (2019). Blazeface: Sub-millisecond neural face detection on mobile gpus. arXiv preprint arXiv:1907.05047.

[12] Li, Y., Yang, X., Sun, P., Qi, H., & Lyu, S. (2020). Celeb-df: A large-scale challenging dataset for deepfake forensics. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 3207-3216).

[13] Rana, M. S., & Sung, A. H. (2020, August). Deepfakestack: A deep ensemble-based learning technique for deepfake detection. In 2020 7th IEEE international conference on cyber security and cloud computing (CSCloud)/2020 6th IEEE international conference on edge computing and scalable cloud (EdgeCom) (pp. 70-75). IEEE.

[14] Bonettini, N., Cannas, E. D., Mandelli, S., Bondi, L., Bestagini, P., & Tubaro, S. (2021, January). Video face manipulation detection through ensemble of cnns. In 2020 25th international conference on pattern recognition (ICPR) (pp. 5012-5019). IEEE.

[15] Wang, R., Juefei-Xu, F., Luo, M., Liu, Y., & Wang, L. (2021, October). Faketagger: Robust safeguards against deepfake dissemination via provenance tracking. In Proceedings of the 29th ACM International Conference on Multimedia (pp. 3546-3555).

[16] Alattar, A., Sharma, R., & Scriven, J. (2020). A system for mitigating the problem of deepfake news videos using watermarking. Electronic Imaging, 32, 1-10.

[17] Mangaokar, N., & Prakash, A. (2021). Dispelling misconceptions and characterizing the failings of deepfake detection. IEEE Security & Privacy, 20(2), 61-67.

[18] Trinh, L., Tsang, M., Rambhatla, S., & Liu, Y. (2021). Interpretable and trustworthy deepfake detection via dynamic prototypes. In Proceedings of the IEEE/CVF winter conference on applications of computer vision (pp. 1973-1983).

[19] Venema, A. E., & Geradts, Z. J. (2020). Digital Forensics, Deepfakes, and the Legal Process.'. The SciTech Lawyer.

[20] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., & Polosukhin,I. (2017). Attention is all you need. Advances in neural information processing systems, 30.

[21] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint

arXiv:1412.6980.

[22] Ciftci, U. A., Demir, I., & Yin, L. (2020). Fakecatcher: Detection of synthetic portrait videos using biological signals. IEEE transactions on pattern analysis and machine intelligence.

[23] Laptev, I., Marszalek, M., Schmid, C., & Rozenfeld, B. (2008, June). Learning realistic human actions from movies. In 2008 IEEE Conference on Computer Vision and Pattern Recognition (pp. 1-8). IEEE.

[24] Güera, D., & Delp, E. J. (2018, November). Deepfake video detection using recurrent neural networks. In 2018 15th IEEE international conference on advanced video and signal based surveillance (AVSS) (pp. 1-6). IEEE.

[25] Lyu, S. (2018). Detecting deepfake videos in the blink of an eye. The Conversation, 29.

[26] Zi, B., Chang, M., Chen, J., Ma, X., & Jiang, Y. G. (2020, October). Wilddeepfake: A challenging real-world dataset for deepfake detection. In Proceedings of the 28th ACM international conference on multimedia (pp. 2382-2390).

[27] Younus, M. A., & Hasan, T. M. (2020, April). Effective and fast deepfake detection method based on haar wavelet transform. In 2020 International Conference on Computer Science and Software Engineering (CSASE) (pp. 186-190). IEEE.

[28] Nirkin, Y., Keller, Y., & Hassner, T. (2019). Fsgan: Subject agnostic face swapping and reenactment. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 7184-7193).

[29] Li, L., Bao, J., Yang, H., Chen, D., & Wen, F. (2019). Faceshifter: Towards high fidelity and occlusion aware face swapping. arXiv preprint arXiv:1912.13457.

[30] Nguyen, T. T., Nguyen, Q. V. H., Nguyen, D. T., Nguyen, D. T., Huynh-The, T., Nahavandi, S., ... & Nguyen, C. M. (2022). Deep learning for deepfakes creation and detection: A survey. Computer Vision and Image Understanding, 223, 103525.