

CHAPTER 1

INTRODUCTION

In the rapidly evolving digital landscape, social media platforms have become a primary means of communication and information sharing. However, with the advent of sophisticated artificial intelligence (AI) technologies, the creation and dissemination of deceptive digital content, known as deepfakes, have emerged as significant threats. Deepfakes utilize advanced AI models to fabricate hyper-realistic images, videos, and audio clips, often making it challenging to distinguish between genuine and manipulated media. This has far-reaching implications, including political manipulation, misinformation, and personal harm through activities such as blackmail and revenge porn. The proliferation of deepfakes has heightened the urgency for robust detection mechanisms.

The increasing sophistication of deepfake technology, coupled with the widespread use of mobile cameras and social media, has made it easier than ever to create and propagate manipulated media. As these fake videos become more realistic, the potential for harm escalates. For instance, a deepfake of a political leader declaring war could lead to widespread panic and geopolitical instability. Similarly, deepfakes of celebrities or public figures making controversial statements can damage reputations and incite public unrest. Therefore, developing effective deepfake detection tools is essential to maintain the integrity of digital content and prevent the spread of misinformation.

"Deepfake detection: Safeguarding truth in digital content" is a project aimed at addressing the challenges posed by deepfake technology. Our approach utilizes a combination of advanced machine learning algorithms and image processing techniques to detect deepfakes. Specifically, we employ a Long Short-Term Memory (LSTM) based artificial neural network for sequential temporal analysis of video frames, alongside a pre-trained ResNeXt Convolutional Neural Network (CNN) for frame-level feature extraction. The ResNeXt CNN captures intricate details within each frame, which are then analyzed by the LSTM network to determine the authenticity of the video.

To ensure the model's robustness and accuracy, we trained it on a large, balanced dataset comprising various sources, including FaceForensics++, the Deepfake Detection Challenge dataset, and Celeb-DF. This diverse training data helps the model generalize well to different types of deepfake videos, enhancing its real-world applicability.

To make our solution accessible and user-friendly, we developed a front-end application that allows users to upload videos for analysis. The application processes the uploaded video using our trained model and provides a classification result, indicating whether the video is real or a deepfake, along with the model's confidence level. This intuitive interface enables users, including non-experts, to easily verify the authenticity of digital content.

The development of "Deepfake Detective" is a crucial step towards combating the negative impacts of deepfake technology. By integrating expertise from artificial intelligence, digital forensics, and media studies, our project aims to push the boundaries of technological innovation while addressing significant societal challenges. In an era characterized by truth decay and digital misinformation, our solution provides a means to preserve trust in online content and protect individuals and communities from the dangers of manipulated media.

As deepfake technology continues to evolve, the need for effective detection mechanisms becomes increasingly urgent. "Deepfake Detective" represents a comprehensive approach to this challenge, combining cutting-edge machine learning techniques with practical applications to safeguard the digital landscape. By detecting and preventing the spread of deepfakes, we aim to contribute to a more secure and trustworthy online environment.

CHAPTER 2

LITRATURE SURVEY

Deepfake Video Detection Using Convolutional Neural Network

Authors: Aarti Karandikar, Vedita Deshpande, Sanjana Singh, Sayali Nagbhidkar, Saurabh Agrawal.

Reference Number :1

Published Year: 2020

With the advent of new technological enhancements in artificial intelligence, new sophisticated AI techniques are used to create fake videos. Such videos can pose a great threat to the society in various social and political ways and can be used for malicious purposes. These fake videos are called deepfakes. Deepfakes refer to manipulated videos, or other digital representations produced by sophisticated artificial intelligence, that yield fabricated images and sounds that appear to be real. A deep-learning system can produce a persuasive counterfeit by studying photographs and videos of a target person from multiple angles, and then mimicking its behaviour and speech patterns. Detecting these videos is a massive problem because of the increasing developments in more realistic deepfake creation technologies emerging every now and then. The paper aims to solve this problem by proposing a model that analyses the frames of the videos using deep learning approach to detect inconsistencies in facial features, compression rate and discrepancies introduced in the videos while creating them. The model uses a convolutional neural network along with transfer learning to train the model that can catch these instilled errors in the deepfakes. The neural network is trained on these discrepancies induced during deepfake creation around the face. It uses a dataset called “Celeb-DF: A New Dataset for DeepFake Forensics” to train the model. The paper further discusses methods that can be used, in detail, to improve learning by this model.

DeepFake detection is a major need in today’s world and needs considerable detection techniques as detecting deepfakes will become more challenging in the future. As deepfakes can have major social and political impact improvements should be made continuously in its detection techniques. In this paper, proposed method uses transfer learning on VGG-16 model to train the dataset and focus on facial manipulation for detection of forgery. Transfer learning is essential as the model should be trained in considerable amount of time and should require minimum resources to give the desired accuracy for its classification over varied examples in the dataset. The proposed model works well and is able to successfully gather features required for further processing to test for deepfakes. For improving the performance, further research can be done on detecting temporal and audio discrepancies and then using this combined information with features extracted from image processing module. It is observed that the accuracy of the proposed model decreases with low quality images and with medium quality videos the accuracy needs to be further increased using combined models for training on temporal parameters. Thus better dataset with improved quality will lead to better training. Various ensemble learning techniques can also be implemented to further increase the accuracy

of the model and account for variance in the dataset. Aggregation of results over each frame and over different learning models will thus give best results.

Deepfake detection is critical in the digital age due to the potential social and political impacts. Future research should focus on enhancing detection techniques, including temporal and audio discrepancies, and using ensemble learning methods to improve accuracy.

DETECTING DEEPPAKES WITHOUT SEEING ANY

Authors: Tal Reiss Bar Cavia YedidHoshen.

Reference Number :2

Published Year: 2023

Deepfake attacks, malicious manipulation of media containing people, are a serious concern for society. Conventional deepfake detection methods train supervised classifiers to distinguish real media from previously encountered deepfakes. Such techniques can only detect deepfakes similar to those previously seen, but not zero day (previously unseen) attack types. As current deepfake generation techniques are changing at a breathtaking pace, new attack types are proposed frequently, making this a major issue. Our main observations are that: i) in many effective deepfake attacks, the fake media must be accompanied by false facts i.e. claims about the identity, speech, motion, or appearance of the person. For instance, when impersonating Obama, the attacker explicitly or implicitly claims that the fake media show Obama; ii) current generative techniques cannot perfectly synthesize the false facts claimed by the attacker. We therefore introduce the concept of “fact checking”, adapted from fake news detection, for detecting zero-day deepfake attacks. Fact checking verifies that the claimed facts (e.g. identity is Obama), agree with the observed media (e.g. is the face really Obama’s?), and thus can differentiate between real and fake media. Consequently, we introduce FACTOR, a practical recipe for deepfake fact checking and demonstrate its power in critical attack settings: face swapping and audio-visual synthesis. Although it is training free, relies exclusively on off-the-shelf features, is very easy to implement, and does not see any deepfakes, it achieves better than state-of-the-art accuracy

This paper proposes the concept of fact checking to address the challenge of detecting unseen, zero day attacks. We propose FACTOR for implementing this, and showcase it in three important settings. FACTOR outperforms the state-of-the-art without seeing any fake data, using only pretrained feature encoders and being simple to implement.

A novel approach for detecting deep fake videos using graph neural network

Authors: M. M. El-Gayar, Mohamed Abouhawwash, S. S. Askar and Sara Sweidan.

Reference Number :3

Published Year: 2024

Deep fake technology has emerged as a double-edged sword in the digital world. While it holds potential for legitimate uses, it can also be exploited to manipulate video content, causing severe social and security concerns. The research gap lies in the fact that traditional deep fake detection methods, such as visual quality analysis or inconsistency detection, need help to keep up with the rapidly advancing technology used to create deep fakes. That means there's a need for more sophisticated detection techniques. This paper introduces an enhanced approach for detecting deep fake videos using graph neural network (GNN). The proposed method splits the detection process into two phases: a mini-batch graph convolution network stream and a four-block CNN stream comprising Convolution, Batch Normalization, and Activation function. The final step is a flattening operation, which is essential for connecting the convolutional layers to the dense layer. The fusion of these two phases is performed using three different fusion networks: FuNet-A (additive fusion), FuNet-M (element-wise multiplicative fusion), and FuNet-C (concatenation fusion). The paper further evaluates the proposed model on different datasets, where it achieved an impressive training and validation accuracy of 99.3% after 30 epochs.

In this project, they introduced a novel and interpretable Graph Neural Network (GNN) model designed to detect synthetic facial images created using various deceptive techniques. The model's hierarchical structure captures subtle frame characteristics, enhancing feature representation and detection accuracy through activation recalibration and variable refinement. It leverages both content and subsurface relationships via graph connections for comprehensive data understanding. The detection process involves a mini-batch graph convolution network stream and a four-block Convolutional Neural Network (CNN) stream, integrated through FuNet-A, FuNet-M, and FuNet-C fusion networks. Extensive testing across four diverse datasets demonstrated the model's exceptional ability to identify deepfakes, achieving a training and validation accuracy of 99.3% after 30 epochs. Future work will focus on refining the model for real-time detection, exploring additional fusion techniques, and enhancing its explainability to adapt to evolving deepfake technologies, ensuring robust and effective deepfake detection.

Deepfake detection using deeplearning methods: A systematic and comprehensive review.

Authors: ArashHeidari ,NimaJafariNavimipour, HasanDag, MehmetUnal.

Reference Number :4

Published Year: 2022

Deep Learning (DL) has been effectively utilized in various complicated challenges in healthcare, industry, and academia for multiple purposes, including thyroid diagnosis, lung nodule recognition, computer vision, large data analytics, and human-level control. Nevertheless, developments in digital technology have been used to produce software that poses a threat to democracy, national security, and confidentiality. Deepfake is one of those DL-powered applications that has lately surfaced. Deepfake systems can create fake images primarily by replacing scenes or images, movies, and sounds that humans cannot tell apart from real ones.

This study provides a complete assessment of the literature on deepfake detection strategies using DL-based algorithms. We categorize deepfake detection methods in this work based on their applications, which include video detection, image detection, audio detection, and hybrid multimedia detection. The objective of this paper is to give the reader a better knowledge of (1) how deepfakes are generated and identified, (2) the latest developments and breakthroughs in this realm, (3) weaknesses of existing security methods, and (4) areas requiring more investigation and consideration.

Deepfakes have eroded people's faith in digital content when seeing them no longer equates to believing in them. This becomes especially important today, even as capabilities for making deepfakes are becoming more accessible, and online platforms can quickly distribute fake content. People's beliefs and truths can be jeopardized in the absence of deepfake detection methods. In this regard, such methods can help prevent the spread of fake multimedia content worldwide while also making it easier for the media to detect them. This work offered a systematic review of the DL mechanisms for deepfake detection. Before presenting the goal of this research, we addressed the advantages and disadvantages of some systematic and peer-reviewed studies about DL-deepfake detection algorithms. Also, the advantages and disadvantages of each mechanism were explored in four categories based on their applicability. The detecting tools and platforms for DL-deepfake were also examined. According to articles based on qualitative characteristics, most publications are assessed based on accuracy, AUC, latency, robustness, and complexity. Meanwhile, some functions, such as security and delay, go unused. Besides, various programming language libraries are utilized to analyse and implement the discussed methods, with Keras accounting for 24% of the effort. . In terms of the prospective findings, applying DL to process deepfake detection takes a lot of time and effort and tight collaboration between government, business, and academia. On the other hand, DL has been acclaimed as a wonderful methodology for developing intelligent solutions to these challenges. The findings of this work could assist in developing deepfake detection-based DL algorithms in real world scenarios.

DeepFake Detection for Human Face Images and Videos: A Survey

Authors: Asad Malik, Minrou Kuribayashi, Ahamad Neyaz Khan, Sani M. Abdullahi.

Reference Number :5**Published Year: 2022**

Techniques for creating and manipulating multimedia information have progressed to the point where they can now ensure a high degree of realism. DeepFake is a generative deep learning algorithm that creates or modifies face features in a super-realistic form, making it difficult to distinguish between real and fake features. This technology has greatly advanced and promotes a wide range of applications in TV channels, video game industries, and cinema, such as improving visual effects in movies. However, it also facilitates a variety of criminal activities, such as generating misinformation by mimicking famous people. To identify and classify DeepFakes, research in DeepFake detection using deep neural networks (DNNs) has attracted increased interest. Basically, DeepFake is regenerated media obtained by injecting or replacing some information within the DNN model.

This article offers a comprehensive survey of a new and prominent technology, namely, DeepFake. It communicates the basics, benefits, and threats associated with DeepFake, including GAN-based DeepFake applications. In addition, DeepFake detection models are also discussed. The inability to transfer and generalize is common in most existing deep learning-based detection methods, which implies that multimedia forensics has not yet reached its zenith. Much interest has been shown by different important organizations and experts that are contributing to the improvement of applied techniques. However, much effort is still needed to ensure data integrity, hence the need for other protection methods. Furthermore, experts are anticipating a new wave of DeepFake propaganda in AI against AI encounters where none of the sides has an edge over the other.

FaceForensics++: Learning to Detect Manipulated Facial Image

Authors: Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, Matthias Niebner.

Reference Number :6**Published Year: 2019**

The rapid progress in synthetic image generation and manipulation has now come to a point where it raises significant concerns for the implications towards society. At best, this leads to a loss of trust in digital content, but could potentially cause further harm by spreading false information or fake news. This paper examines the realism of state-of-the-art image manipulations, and how difficult it is to detect them, either automatically or by humans.

To standardize the evaluation of detection methods, we propose an automated benchmark for facial manipulation detection. In particular, the benchmark is based on DeepFakes, Face2Face, FaceSwap, and NeuralTextures as prominent representatives for facial manipulations at random compression level and size. The benchmark is publicly available [2] and contains a hidden test set as well as a database of over 18 million manipulated images. This dataset is over an order of magnitude larger than comparable, publicly available forgery datasets. Based on this data, we performed a thorough analysis of data-driven forgery detectors. We show that the

use of additional domain-specific knowledge improves forgery detection to unprecedented accuracy, even in the presence of strong compression, and clearly outperforms human observers.

While current state-of-the-art facial image manipulation methods exhibit visually stunning results, we demonstrate that they can be detected by trained forgery detectors. It is particularly encouraging that also the challenging case of low-quality video can be tackled by learning-based approaches, where humans and hand-crafted features exhibit difficulties. To train detectors using domain-specific knowledge, we introduce a novel dataset of videos of manipulated faces that exceeds all existing publicly available forensic datasets by an order of magnitude.

This paper, focus on the influence of compression to the detectability of state-of-the-art manipulation methods, proposing a standardized benchmark for follow-up work. All image data, trained models, as well as our benchmark are publicly available and are already used by other researchers. In particular, transfer learning is of high interest in the forensic community. As new manipulation methods appear by the day, methods must be developed that are able to detect fakes with little to no training data. Our database is already used for this forensic transfer learning task, where knowledge of one source manipulation domain is transferred to another target domain, as shown by Cozzolino et al. [18]. We hope that the dataset and benchmark become a stepping stone for future research in the field of digital media forensics, and in particular with a focus on facial forgeries.

DEEP FAKE VIDEO DETECTION USING RES NEXT CNN AND LSTM

Authors: S Jeevidha, S. Saraswathi, Kaushik J B, Preethi K, NallamVenkataramaya.

Reference Number :7

Published Year: 2023

The proliferation of deepfake videos in today's digital era has sparked significant concerns regarding their potential to undermine the credibility of visual media, posing a substantial threat. Advances in computational power have enabled the creation of highly realistic AI-generated videos, capable of spreading disinformation and causing societal discord. To address this challenge, a new deep learning-based method has been developed to distinguish between AI-generated fake videos and authentic ones. This technique involves fine-tuning the transformer module to explore new feature spaces using Attention-based networks (Res-Next CNN), which selectively focus on critical video features. Frame-level features are extracted using Res-Next Convolutional Neural Networks and utilized to train an LSTM-based Recurrent Neural Network (RNN) for video classification. The system is rigorously evaluated on diverse datasets including Face-Forensic++, Deepfake Detection Challenge, Celeb-DF, and a custom dataset, demonstrating its efficacy in real-time manipulation detection scenarios. Practical applications include mitigating the dissemination of deepfake videos on social media, news platforms, and law enforcement channels to uphold the integrity of online content and combat misinformation.

In conclusion, the development of robust deep learning techniques for detecting deepfake videos represents a crucial step in safeguarding the integrity of visual media in the digital age. By leveraging advanced AI models such as Attention-based networks and LSTM-based RNNs, this approach demonstrates promising capabilities in accurately identifying manipulated videos across various datasets. This method not only addresses current challenges posed by deepfakes but also offers practical solutions for preventing their dissemination on social media, news outlets, and law enforcement platforms. Moving forward, continued research and refinement of these techniques are essential to stay ahead of evolving manipulation tactics and ensure the authenticity and reliability of online content.

DeepFake Detection using InceptionResNetV2 and LSTM

Authors: Priti Yadav, Ishani Jaswal, Jaiprakash Maravi, Vibhash Choudhary and Gargi Khanna.

Reference Number :8

Published Year: 2021

“Seeing is believing” is simply not true anymore and has huge ramifications for many different aspects of our life. As technology improves, it’s becoming easier to develop deepfakes, with some apps making it possible at the palm of a hand. Detecting deepfakes has become increasingly challenging for the human eye. However, researchers are actively pursuing methods to recognize deepfakes. Deepfakes are synthetic media created using AI algorithms that learn attributes from both a target and source image, superimposing the former onto the latter. Our focus is on detecting video deepfakes using deep learning neural networks like LSTM and InceptionResNetV2. We successfully developed a deepfake detection model using transfer learning, where a pretrained InceptionResNetV2 CNN extracts features and forms vectors. The LSTM layer is trained on these features, and evaluation using confusion matrices yielded validation and testing accuracies of 84.75% and 91.48% after 20 and 40 epochs, respectively.

The faith of the masses has started to disintegrate due to the Deepfakes as the streaming content no longer seems to be authentic and real. In our paper we presented an approach that can automatically detect deep fake based on deep learning concept. In deepfakes the target face appears briefly in a video so the model divides user video into frames and these frames were further pre-processed using InceptionResNetV2 and LSTM. The method provided good level accuracy and reliability. The proposed methodology is capable of analysing any video using convolutional LSTM system and also helps in detecting deepfake face which has been manipulated therefore preventing individuals from defaming. We can also hold experiments with a greater number of epochs and Learning Rate to get higher accuracy. In future one can extent this work by exploring more architectures that will help in implementing new detection techniques to detect deepfakes.

DEEFAKE DETECTION USING XCEPTION AND LSTM

Authors: Adil Mohammed Parayil,Ameen Masood, Muhammed Ajas P,Tharun R,Usha

Reference Number :9

Published Year: 2023

Deepfakes are fabricated works of art where a person appearing in an earlier photograph or motion picture is modified to exhibit the face of another person. A deep learning-based generative model called a Generative Adversarial Network (GAN) is a model architecture for training generative models to create fake videos. In the context of GANs, the generation model lends significance to points in a predetermined latent space, enabling fresh points pulled from the latent space to be fed to the generator model as input and utilized to produce brand-new and distinctive output instances. Due to this, it is simple to use GANs to build deep fakes that can be used inappropriately in various contexts. Due to the engrossing misuse of deepfakes, there is a need for appropriate detection tools. Deepfake detection requires a large amount of data to train models and test them. They require large datasets that contain thousands of videos, both real and fake at equal ratios to avoid biased results. This paper works with deep learning-based deep fake detection using Convolutional Neural Network(CNN) and Recurrent Neural Network(RNN), CNN utilizing the Xception network, and ytLSTM as RNN. In the algorithm, the spatial features are recognized by the Xception and LSTM identifies the temporal inconsistency between frames. The models are trained against three standard datasets. The results are also validated with standard datasets resulting in deepfake prediction with a minimal computational time and nominal accuracy.

We presented a neural network-based approach to classify the video as deep fake or real. Our method does the frame level detection using Xception CNN and video classification using RNN along with LSTM. The proposed method is capable of detecting the video as a deep fake or real. We believe that it will provide an average accuracy on real-time data with less computational effort. Overall, the combination of Xception CNN and LSTM RNN is a promising approach for deepfake detection that can provide high accuracy and robustness. However, further research is needed to explore the performance of this approach on more diverse datasets and in different real-world scenarios.

Video Face Manipulation Detection Through Ensemble of CNNs

Authors: Nicolo Bonettini, Edoardo Daniele Cannas, Sara Mandelli, Luca Bondi, Stefano Tubaro, Paolo Bestagini.

Reference Number :10

Published Year: 2020

In the last few years, several techniques for facial manipulation in videos have been successfully developed and made available to the masses (i.e., FaceSwap, deepfake, etc.). These methods enable anyone to easily edit faces in video sequences with incredibly realistic results and very little effort. Despite the usefulness of these tools in many fields, if used maliciously, they can have a significantly bad impact on society (e.g., fake news spreading, cyber bullying through fake revenge porn). The ability to objectively detect whether a face has been manipulated in a video sequence is then a task of utmost importance.

In this paper, we tackle the problem of face manipulation detection in video sequences targeting modern facial manipulation techniques. In particular, we study the ensembling of different trained Convolutional Neural Network (CNN) models. In the proposed solution, different models are obtained starting from a base network (i.e., EfficientNetB4) making use of two different concepts: (i) attention layers; (ii) siamese training. We show that combining these networks leads to promising face manipulation detection results on two publicly available datasets with more than 119000 videos.

Being able to detect whether a video contains manipulated content is nowadays of paramount importance, given the significant impact of videos in everyday life and in mass communications. In this vein, we tackle the detection of facial manipulation in video sequences, targeting classical computer graphics as well as deep learning generated fake videos. The proposed method takes inspiration from the family of EfficientNet models and improves upon a recently proposed solution, investigating an ensemble of models trained using two main concepts: (i) an attention mechanism which generates a human comprehensible inference of the model, increasing the learning capability of the network at the same time; (ii) a triplet siamese training strategy which extracts deep features from data to achieve better classification performances. Results evaluated over two publicly available datasets containing almost 120 000 videos reveals the proposed ensemble strategy as a valid solution for the goal of facial manipulation detection.

CHAPTER 3

PROBLEM STATEMENT AND OBJECTIVES

3.1 PROBLEM STATEMENT

For many years, visual effects have been used to create convincing manipulations of digital images and videos. However, recent advancements in deep learning have significantly increased the realism of fake content and made it more accessible to create. These AI-synthesized media, commonly known as deepfakes, pose a serious threat due to their potential misuse. Creating deepfakes using AI tools has become a straightforward task, but detecting these deepfakes remains a significant challenge. Historically, deepfakes have been used as powerful tools to create political tensions, fabricate fake terrorism events, produce revenge porn, and blackmail individuals. The spread of deepfakes on social media platforms can lead to the dissemination of misinformation and harm to individuals and society. Therefore, it is crucial to develop effective methods to detect deepfakes and prevent their distribution. To address this challenge, we propose a method for detecting deepfakes using a Long Short-Term Memory (LSTM) based artificial neural network. This approach aims to identify and mitigate the impact of deepfakes, thereby enhancing the integrity of digital content shared on social media platforms.

3.2 OBJECTIVES

- Our project aims to uncover the distorted truth behind deepfakes.
- Our project will reduce the abuse and misleading of common people on the World Wide Web
- Our project will distinguish and classify videos as either deepfakes or authentic.
- Provide an easy-to-use system for uploading videos to determine whether they are real or fake.

CHAPTER 4

SYSTEM REQUIREMENT SPECIFICATION

The Deep Fake Detection system aims to accurately identify fake videos generated through deep learning techniques. By utilizing LSTM (Long Short-Term Memory) and ResNet (Residual Networks), the system will analyze video frames and temporal data to determine the authenticity of the content.

1. Functional Requirements

The functional requirements of the proposed work include:

1. Video Upload and Processing:

- Users should be able to upload videos through the system's front-end interface.
- The system will preprocess the uploaded videos, extracting frame-level features using the ResNeXt CNN.

2. Deepfake Detection:

- The preprocessed data will be fed into the LSTM-based artificial neural network for sequential temporal analysis.
- The system will classify the video as either deepfake or authentic, providing a confidence score for the classification.

3. Reporting:

- The system will generate a detailed report indicating the authenticity of the video and the confidence level of the classification.

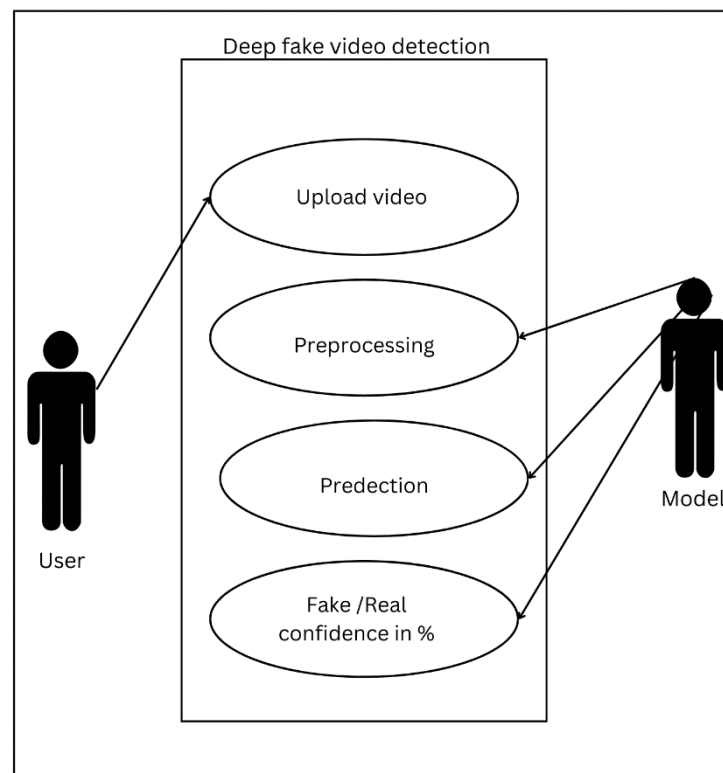


Fig 4.1 Use case diagram

2.Non-Functional Requirements

- **Performance:** The system's performance will be measured using metrics such as accuracy, precision, recall, F1 score, and confusion matrix.
- **Usability:** The user interface should be intuitive and user-friendly, allowing users to interact with the system easily and efficiently.
- **Reliability:** The system should maintain high accuracy in detection and classification, minimizing errors and false predictions.
- **Scalability:** The system should be capable of handling growing volumes of video data and increasing computational demands over time.
- **Security:** Ensure that the uploaded videos and the analysis results are securely stored and processed to protect user privacy and data integrity.

Software requirements and Hardware requirements

Software requirements

- **Operating System:** Windows 7+ 2.
- **Programming Language:** Python.
- **Framework:** PyTorch 1.4.
- **Cloud platform:** Google Cloud Platform.
- **Libraries and Frameworks:**
 - TensorFlow/Keras:** For building and training the deep learning models.
 - PyTorch:** An alternative deep learning framework that can be used based on preference.
- **Machine Learning Libraries:**
 - scikit-learn:** For preprocessing, metrics, and classical machine learning tasks.
- **Computer Vision Libraries:**
 - OpenCV:** For image and video processing.
 - Pillow:** For image manipulation and processing.
- **Data Manipulation and Analysis:**
 - NumPy:** For numerical operations.
 - Pandas:** For data manipulation and analysis.
- **Visualization:**
 - Matplotlib:** For plotting and visualization.
 - Seaborn:** For advanced statistical plots.
 - TensorBoard:** For visualizing training logs and metrics.

Hardware requirements

In this project, a computer with sufficient processing power is needed. This project requires too much processing power, due to the image and video batch processing.

Small-scale Project:

- CPU: Intel i7-10700K or AMD Ryzen 7 3700X
- GPU: NVIDIA GeForce RTX 3060
- RAM: 16 GB
- Storage: 1 TB SSD
- PSU: 650W
- Motherboard: Compatible with chosen CPU and GPU

Large-scale Project:

- CPU: Intel i9-10900K or AMD Ryzen 9 5900X
- GPU: NVIDIA GeForce RTX 3090 or NVIDIA A100
- RAM: 64 GB
- Storage: 2 TB NVMe SSD + 4 TB HDD
- PSU: 850W or higher

CHAPTER 5

SYSTEM ANALYSIS

The prevalence of deepfake technology has necessitated the development of reliable methods for detecting manipulated videos. This project integrates Long Short-Term Memory (LSTM) networks, a type of Recurrent Neural Network (RNN), with Residual Networks (ResNet) for feature extraction to create a robust deepfake detection system. This document provides a detailed system analysis, highlighting the architecture, methodology, data, and evaluation metrics used.

Existing System

The current landscape regarding deepfake detection predominantly relies on a variety of techniques ranging from traditional digital forensics to modern machine learning algorithms. Existing systems typically employ methods such as:

1.FaceForensics++: A benchmark dataset and method for detecting manipulated facial regions in videos.Utilizes Convolutional Neural Networks (CNNs) to identify and localize altered facial areas. Includes various manipulation techniques, providing a comprehensive dataset for training and evaluation.

2.MesoNet: A lightweight CNN architecture specifically designed for deepfake detection. Focuses on capturing mesoscopic properties of face images.Balances between accuracy and computational efficiency.

3.DeepFake Detection Challenge (DFDC) Baseline Model: A baseline model released by Facebook as part of the DFDC.Uses a combination of CNNs and RNNs to classify videos as real or fake.Provides a starting point for researchers participating in the challenge to improve detection methods.

4.Face Warping Artifacts Detection: Detects warping artifacts that are often present in deepfake videos.Utilizes image preprocessing techniques to highlight inconsistencies in facial geometry.Targets specific artifacts commonly produced by face-swapping algorithms..

5.Head Pose Estimation: Analyzes the consistency of head poses in video frames.Uses 3D head pose estimation algorithms to detect anomalies in movement.Identifies discrepancies in head movements that are hard to synthesize accurately.

6.Eye Blinking Patterns: Monitors eye blinking behavior to detect deepfakes.Uses statistical models to compare blinking patterns in videos against natural human behavior. Detects unnatural blinking frequencies and patterns, which are common in poorly synthesized deepfakes.

Limitations of Existing Deepfake Detection Systems

- Requires significant computational resources for training on large datasets.
- Lower accuracy compared to deeper, more complex models.
- May not be robust against diverse and sophisticated deepfake techniques.
- Less effective against high-quality deepfakes with minimal warping artifacts.
- Limited to detecting anomalies in head movement, which can be subtle in some deepfakes.
- Limited to detecting anomalies in eye blinking, which can be bypassed in advanced deepfakes

Proposed:

Our proposed deepfake detection system addresses the complexities and challenges identified through extensive analysis and research. Initially, we evaluated various approaches from existing literature to assess the feasibility and effectiveness of different methodologies. Through this process, we determined that training the model on a balanced dataset—comprising both real and fake videos—was crucial to mitigating bias and variance, thereby ensuring accurate predictions. This approach involved identifying key parameters such as blinking patterns, facial expressions, and lighting conditions, which significantly influence the detection accuracy.

For the design phase, we formulated a robust system architecture, which delineates the layers and their configurations. Leveraging insights from our analysis, we selected PyTorch as our framework of choice due to its CUDA support for GPU acceleration and flexibility in customization. The development phase utilized Python 3 for implementation, with Google Cloud Platform serving as the infrastructure for training our models on a large-scale dataset, including real-time data sourced from platforms like YouTube. Evaluation was conducted using Confusion Matrix methodology to gauge the accuracy of our models across diverse scenarios.

The outcome of our solution is a suite of trained deepfake detection models capable of accurately discerning between authentic and manipulated videos. By providing a reliable means to detect deepfakes, our proposed system aims to safeguard against the spread of misinformation and protect the integrity of digital content shared online.

CHAPTER 6

SYSTEM DESIGN

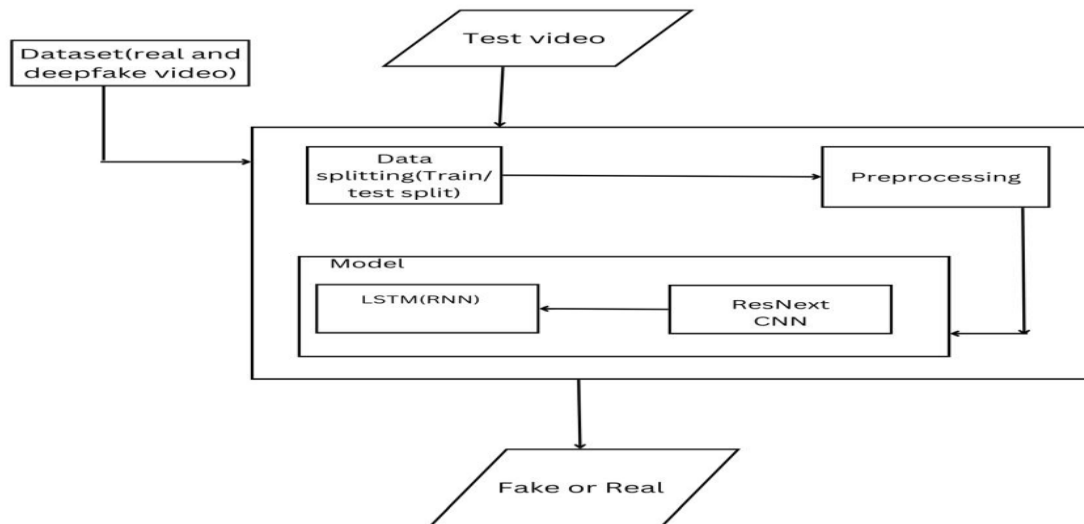


Fig 6.1 System Architecture

A. Dataset:

We are utilizing a diverse dataset comprising an equal number of videos from various sources such as YouTube, FaceForensics++[14], and the Deepfake Detection Challenge dataset[13]. Our newly compiled dataset includes 50% original videos and 50% manipulated deepfake videos. The dataset is divided into 70% for training and 30% for testing.

B. Preprocessing:

The preprocessing stage involves splitting the videos into individual frames. Each frame undergoes face detection, and the detected face is cropped. To maintain consistency in the number of frames, we calculate the mean number of frames per video in the dataset and create a new face-cropped dataset containing this mean number of frames. Frames without detectable faces are ignored during preprocessing. Since processing a 10-second video at 30 frames per second (totaling 300 frames) is computationally intensive, we propose using only the first 100 frames for model training.

C. Model:

Our model architecture includes a ResNext50_32x4d network followed by a single LSTM layer. The Data Loader handles the preprocessed face-cropped videos, splitting them into

training and testing sets. The frames from the processed videos are then passed to the model for training and testing in mini-batches.

D. ResNext CNN for Feature Extraction:

Instead of developing a new classifier, we propose using the ResNext CNN for feature extraction to accurately detect frame-level features. We will fine-tune the network by adding necessary layers and adjusting the learning rate to ensure proper convergence of the gradient descent. The 2048-dimensional feature vectors obtained after the last pooling layer are used as input for the sequential LSTM.

E. LSTM for Sequence Processing:

We assume a sequence of ResNext CNN feature vectors from input frames, and use a 2-node neural network to determine the probabilities of the sequence belonging to a deepfake video or an untampered video. The key challenge is designing a model capable of recursively processing a sequence meaningfully. To address this, we propose using a 2048-unit LSTM with a 0.4 dropout rate, capable of achieving our objective. The LSTM processes frames sequentially, allowing for temporal analysis by comparing the frame at time 't' with the frame at 't-n' seconds, where 'n' represents any number of frames before 't'.

F. Prediction:

A new video is processed through the trained model for prediction. The video undergoes preprocessing to match the format of the trained model. It is split into frames, followed by face cropping. Instead of storing the video locally, the cropped frames are directly passed to the trained model for deepfake detection.

CHAPTER 7

HIGH LEVEL DESIGN

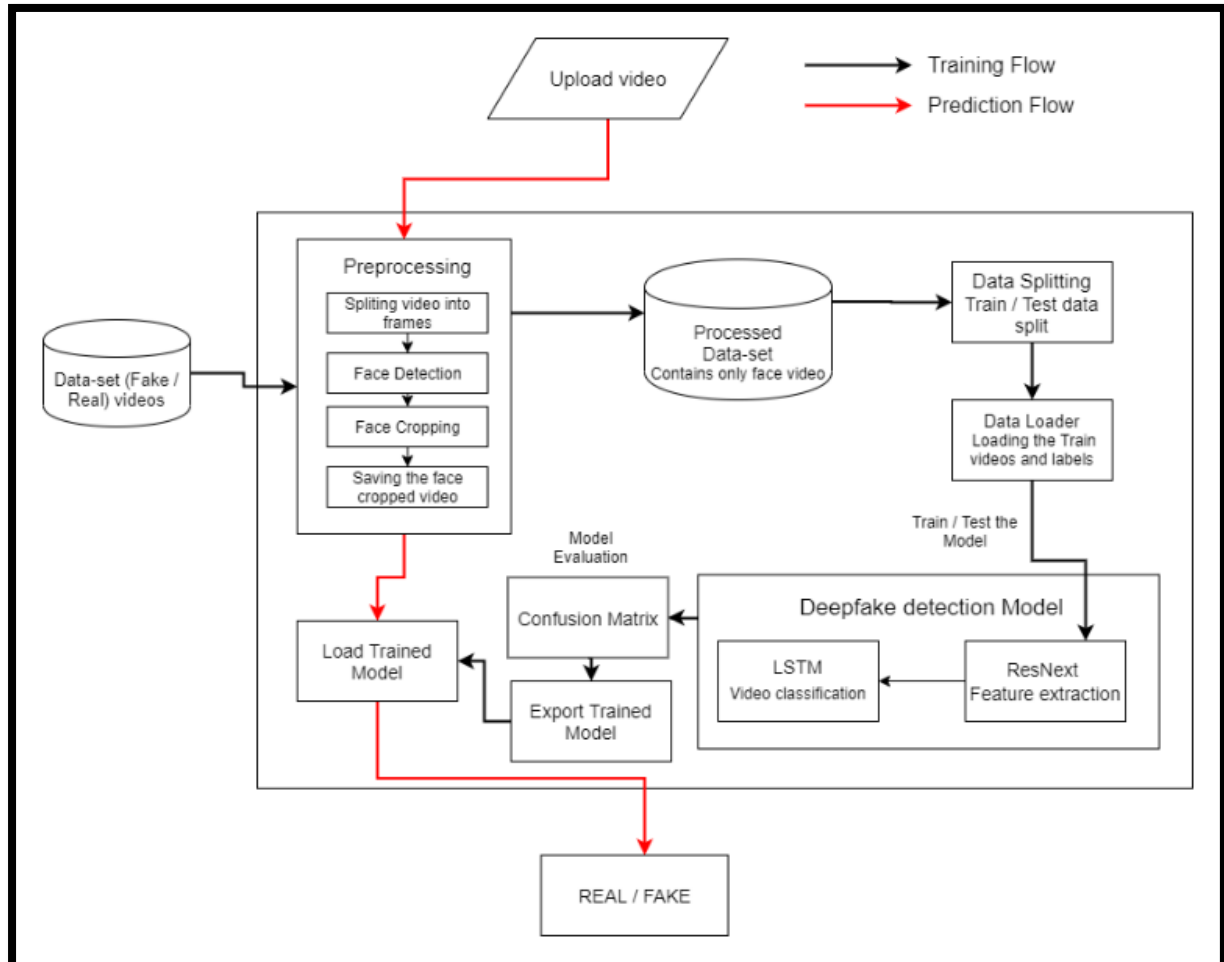


Fig7.1 High level design

1. Dataset (Fake / Real) videos:

- This step involves gathering and storing a dataset consisting of both fake and real videos.
- This data serves as the input for the training flow.

2. Preprocessing:

- **Steps:**
 - **Splitting video into frames:** The video is divided into individual frames for further processing.
 - **Face Detection:** Faces are detected within these frames using face detection algorithms.
 - **Face Cropping:** The detected faces are cropped from the frames.

- **Saving the face cropped video:** The cropped face frames are saved to create a processed dataset.
- The preprocessed data (containing only face videos) is then used in the next steps.

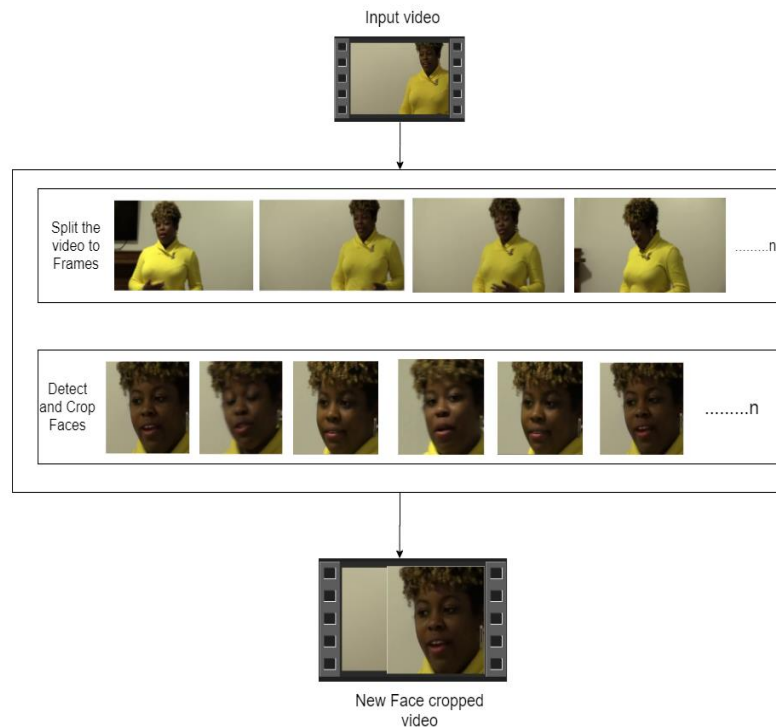


Fig7.2 Pre-processing of video

3. Processed Data set:

- This dataset contains only the face-cropped videos which are ready for data splitting.
- The processed data is split into training and test sets for model training and evaluation.

4. Data Splitting (Train / Test data split):

- The processed dataset is split into training and testing sets.
- This split data is used for loading into the data loader.

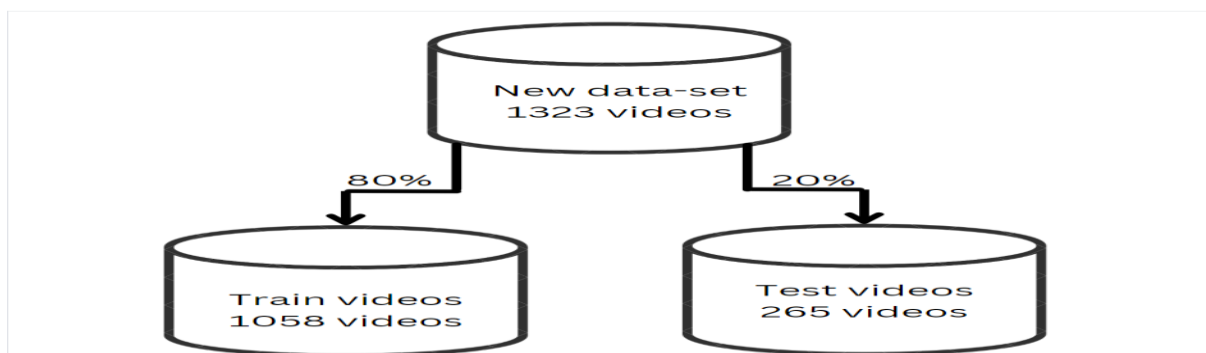


Fig7.3 Train test split

5. Data Loader:

- The data loader is responsible for loading the train videos and their corresponding labels.
- The data loader feeds the data into the deepfake detection model for training and testing.

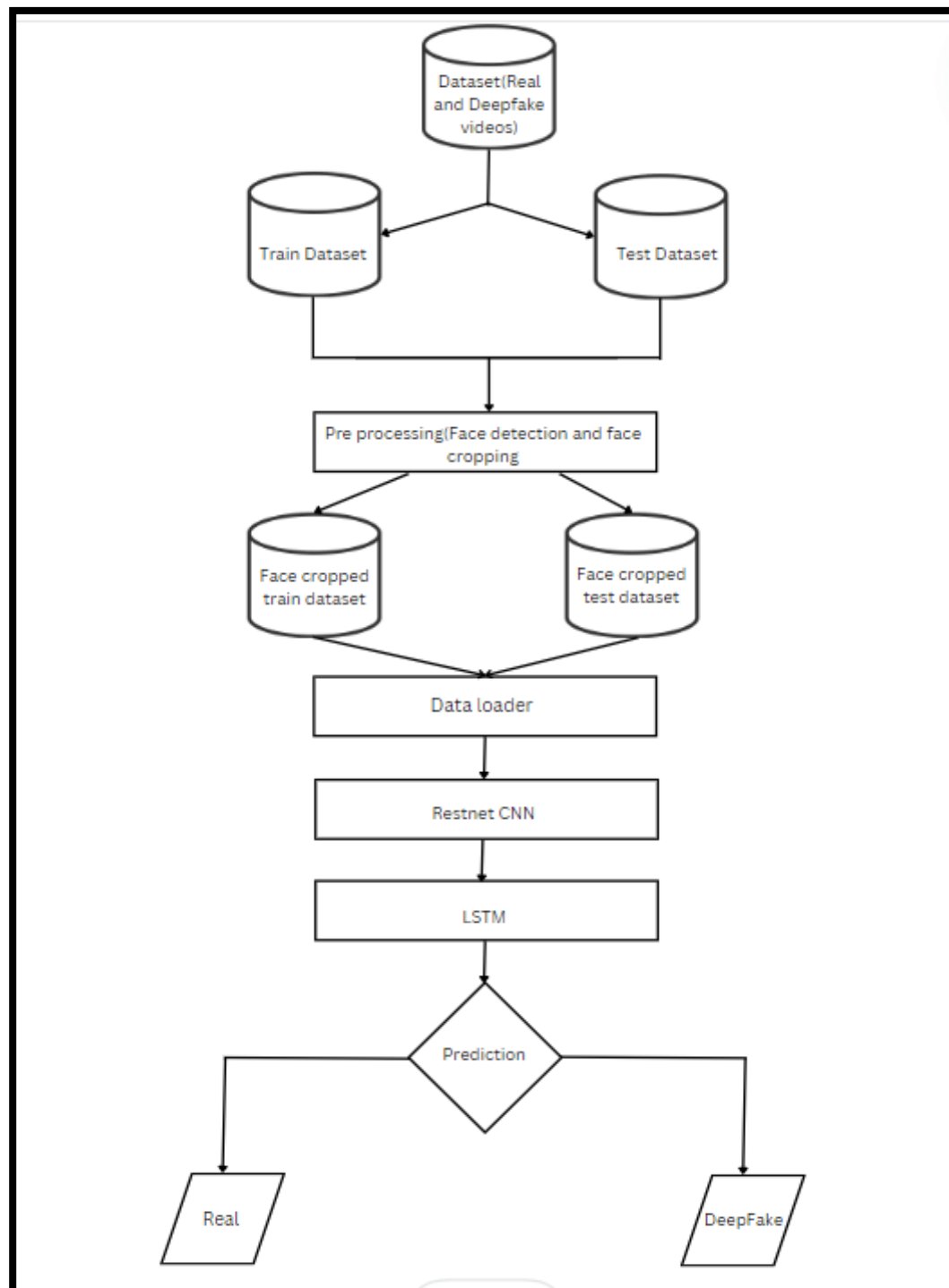


Fig7.4 Training Flow

Training the Model:

Dataset: This is the initial collection of videos, containing both real and deepfake videos.

Train Dataset and Test Dataset: The original dataset is split into two parts: 70% for training and 30% for testing.

Training Set: Used to train the model.

Testing Set: Used to test the model's accuracy and generalization.

Preprocessing (Face Detection and Face Cropping): The videos in both the training and test datasets undergo preprocessing.

Face Detection: Detect faces in each frame of the videos.

Face Cropping: Crop the detected faces from the frames.

Face Cropped Train Dataset and Face Cropped Test Dataset: The preprocessed frames containing only the cropped faces are stored in separate datasets for training and testing. To have a clean and focused dataset for model training and evaluation.

Data Loader: This component loads the face-cropped frames and their corresponding labels from the training and testing datasets. To feed the data into the model during the training and evaluation phases.

LSTM (Long Short-Term Memory): The LSTM network analyzes the temporal sequences of the extracted features from ResNext. To capture the temporal dependencies and dynamics in the video, crucial for detecting deepfakes which often have subtle temporal inconsistencies.

Prediction: The final output of the LSTM network provides a prediction indicating whether the video is real or a deepfake. To classify the video based on the learned patterns and features.

Real/DeepFake: The system provides a final classification output.

Real: Indicates that the video is authentic.

DeepFake: Indicates that the video is a deepfake.

This training flow ensures that the deepfake detection model is properly trained with a comprehensive and focused dataset, leveraging both spatial and temporal features to accurately classify videos as real or fake.

6. Deepfake detection Model:

- **ResNext (Feature extraction):** ResNext CNN is used to extract features from the frames of the videos.
- **LSTM (Video classification):** The extracted features are fed into an LSTM-based Recurrent Neural Network (RNN) to classify videos as real or manipulated.
- The model is trained and tested using the data from the data loader.

7. Model Evaluation:

- After training, the model's performance is evaluated.
- **Confusion Matrix:** Used to evaluate the accuracy and performance of the trained model.
- **Export Trained Model:** The trained model is exported for future use.
- The trained model is then loaded for prediction.

8. Load Trained Model:

- The trained model is loaded for predicting the authenticity of uploaded videos.
- This step is part of the prediction flow where an uploaded video is processed to determine if it is real or fake.

9. Upload Video:

- A new video is uploaded for deepfake detection.
- This video goes through the preprocessing steps and then is fed into the loaded trained model for prediction.

10. REAL / FAKE:

- The final output of the system, indicating whether the uploaded video is real or fake.
- This is the end result of the prediction flow, provided by the loaded trained model after processing the uploaded video.

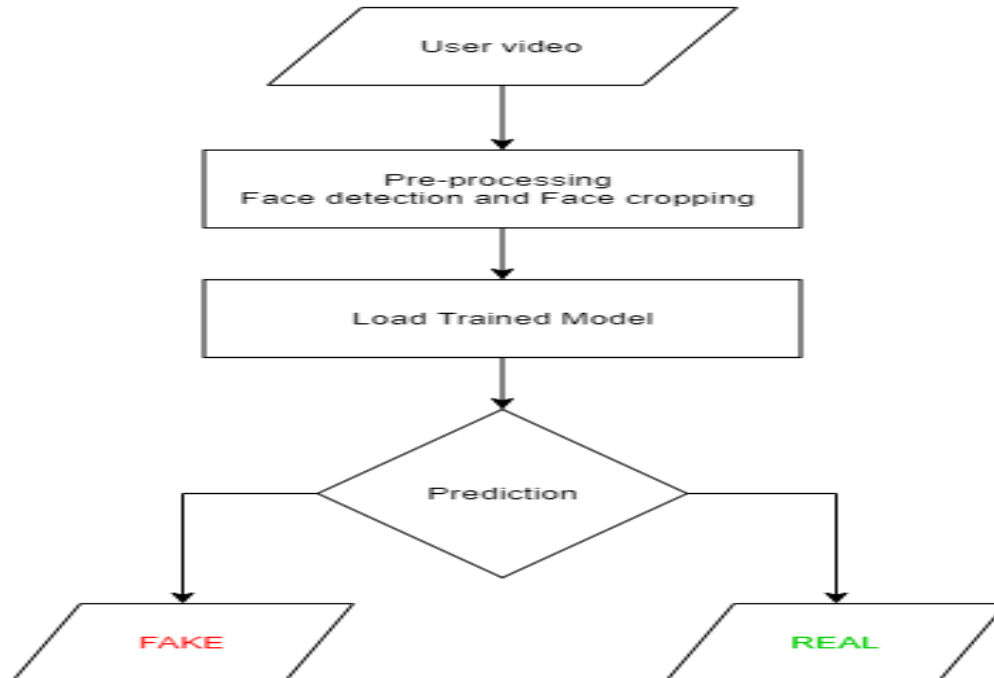


Fig7.5 Testing workflow

CHAPTER 8

SYSTEM IMPLEMENTATION

The advances in the modern open source deep learning frameworks like TensorFlow, Keras, PyTorch along with cheap access to the high computation power has driven the paradigm shift. The Conventional autoencoders[10] and Generative Adversarial Network (GAN) pretrained models have made the tampering of the realistic videos and images very easy. Moreover, access to these pretrained models through the smartphones and desktop applications like FaceApp and Face Swap has made the deepfake creation a childish thing. These applications generate a highly realistic synthesized transformation of faces in real videos. These apps also provide the user with more functionalities like changing the face hair style, gender, age and other attributes. These apps also allow the user to create a very high quality and indistinguishable deepfakes. Although some malignant deepfake videos exist, but till now they remain a minority. So far, the released tools [11,12] that generate deepfake videos are being extensively used to create fake celebrity pornographic videos or revenge porn [13]. Some of the examples are Brad Pitt, Angelina Jolie nude videos. The real looking nature of the deepfake videos makes the celebrities and other famous personalities the target of pornographic material, fake surveillance videos, fake news and malicious hoaxes. The Deepfakes are very much popular in creating the political tension [14]. Due to which it becomes very important to detect the deepfake videos and avoid the percolation of the deepfakes on the social media platforms.

Algorithms Details

Preprocessing Details

- Using glob we imported all the videos in the directory in a python list.
- cv2.VideoCapture is used to read the videos and get the mean number of frames in each video.
- To maintain uniformity, based on mean a value 150 is selected as idea value for creating the new dataset.
- The video is split into frames and the frames are cropped on face location.
- The face cropped frames are again written to new video using VideoWriter.
- The new video is written at 30 frames per second and with the resolution of 112 x 112 pixels in the mp4 format.
- Instead of selecting the random videos, to make the proper use of LSTM for temporal sequence analysis the first 150 frames are written to the new video.

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.

| stage | output | ResNet-50 | ResNeXt-50 (32×4d) |
|-----------|---------|---|---|
| conv1 | 112×112 | 7×7, 64, stride 2 | 7×7, 64, stride 2 |
| conv2 | 56×56 | 3×3 max pool, stride 2 | 3×3 max pool, stride 2 |
| | | $\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$ | $\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128, C=32 \\ 1 \times 1, 256 \end{bmatrix} \times 3$ |
| conv3 | 28×28 | $\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$ | $\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256, C=32 \\ 1 \times 1, 512 \end{bmatrix} \times 4$ |
| conv4 | 14×14 | $\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$ | $\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512, C=32 \\ 1 \times 1, 1024 \end{bmatrix} \times 6$ |
| conv5 | 7×7 | $\begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$ | $\begin{bmatrix} 1 \times 1, 1024 \\ 3 \times 3, 1024, C=32 \\ 1 \times 1, 2048 \end{bmatrix} \times 3$ |
| | 1×1 | global average pool 1000-d fc, softmax | global average pool 1000-d fc, softmax |
| # params. | | 25.5×10^6 | 25.0×10^6 |
| FLOPs | | 4.1×10^9 | 4.2×10^9 |

Fig8.1 ResNext Architecture

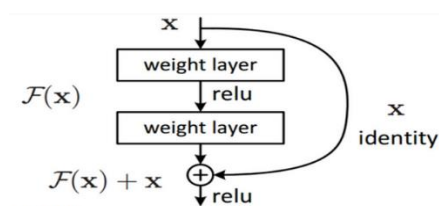


Fig8.2 ResNext Working

- **Sequential Layer** : Sequential is a container of Modules that can be stacked together and run at the same time. Sequential layer is used to store feature vector returned by the ResNext model in a ordered way. So that it can be passed to the LSTM sequentially.
- **LSTM Layer** : LSTM is used for sequence processing and spot the temporal change between the frames. 2048-dimensional feature vectors is fitted as the input to the LSTM. We are using 1 LSTM layer with 2048 latent dimensions and 2048 hidden layers along with 0.4 chance of dropout, which is capable to do achieve our objective. LSTM is used

to process the frames in a sequential manner so that the temporal analysis of the video can be made, by comparing the frame at 't' second with the frame of 't-n' seconds. Where n can be any number of frames before t.

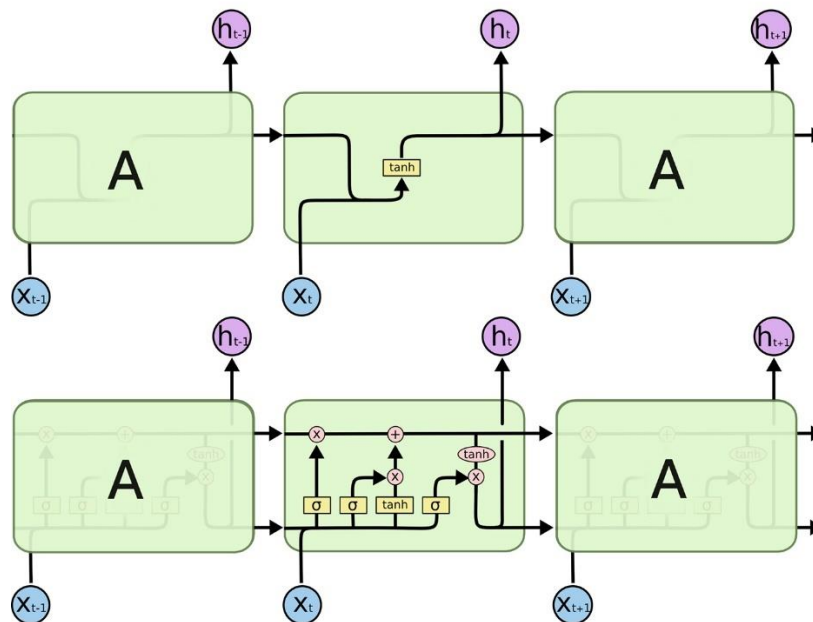


Fig8.3 LSTM Architecture

- **ReLU:** A Rectified Linear Unit is activation function that has output 0 if the input is less than 0, and raw output otherwise. That is, if the input is greater than 0, the output is equal to the input. The operation of ReLU is closer to the way our biological neurons work. ReLU is non-linear and has the advantage of not having any backpropagation errors unlike the sigmoid function, also for larger Neural Networks, the speed of building models based off on ReLU is very fast.

- **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 neighbouring neurons changing the way the weights will be updated during the process of backpropagation.

Model Training Details

- **Train Test Split:**The dataset is split into train and test dataset with a ratio of 70% train videos (4,200) and 30% (1,800) test videos. The train and test split is a balanced split i.e 50% of the real and 50% of fake videos in each split.
- **Data Loader:** It is used to load the videos and their labels with a batch size of 4.

- **Training:** The training is done for 20 epochs with a learning rate of $1e-5$ (0.00001), weight decay of $1e-3$ (0.001) using the Adam optimizer.
- **Adam optimizer[21]:** To enable the adaptive learning rate Adam optimizer with the model parameters is used.
- **Cross Entropy:** To calculate the loss function Cross Entropy approach is used because we are training a classification problem.
- **Confusion Matrix:** A confusion matrix is a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarized with count values and broken down by each class. This is the key to the confusion matrix. The confusion matrix shows the ways in which your classification model is confused when it makes predictions. It gives us insight not only into the errors being made by a classifier but more importantly the types of errors that are being made. Confusion matrix is used to evaluate our model and calculate the accuracy.
- **Export Model:** After the model is trained, we have exported the model. So that it can be used for prediction on real time data.

Model Prediction Details

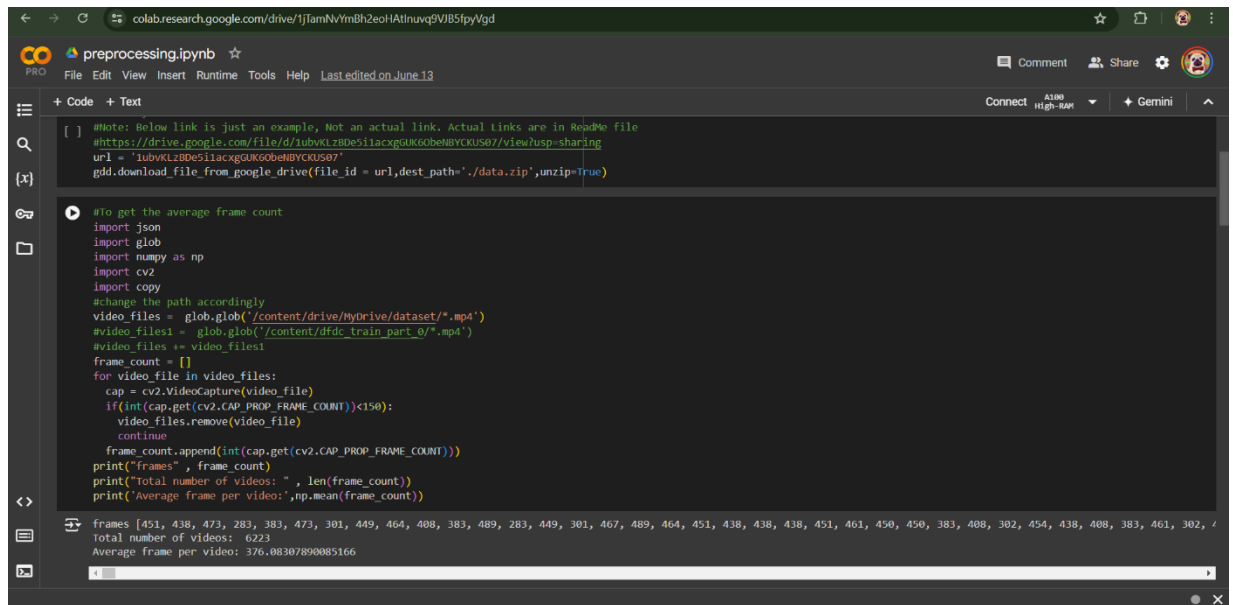
- The model is loaded in the application.
- The new video for prediction is preprocessed and passed to the loaded model for prediction.
- The trained model performs the prediction and return if the video is a real or fake along with the confidence of the prediction.

CHAPTER 9

RESULTS AND SNAPSHOTS

- The outcome of the solution is trained deepfake detection models that will help the users to check if the new video is deepfake or real.

9.1 SNAPSHOTS



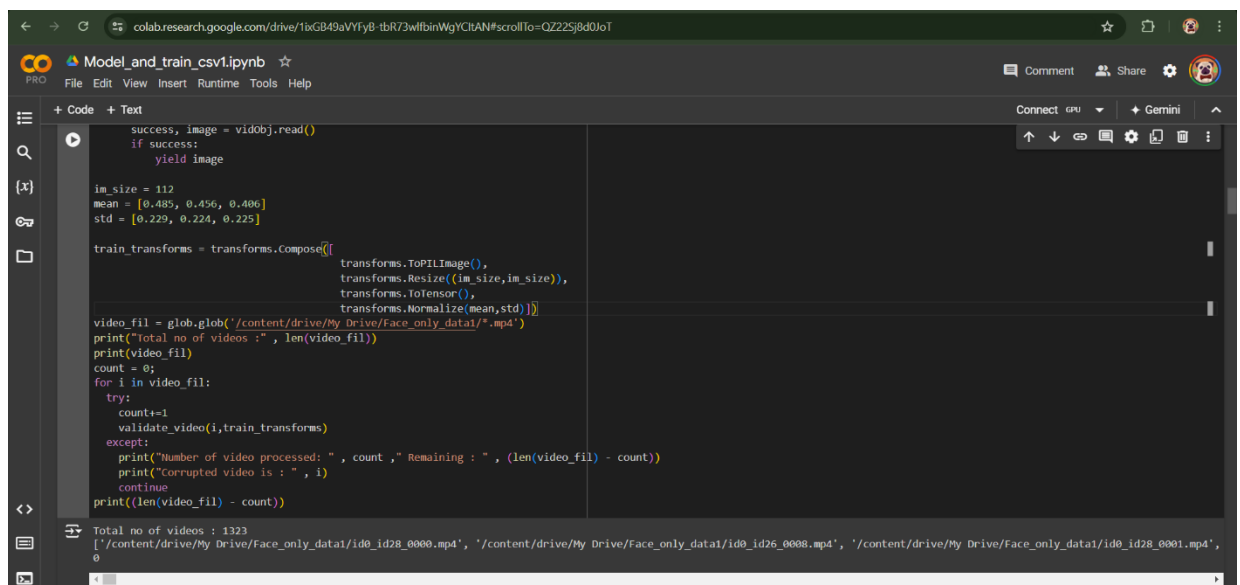
```

#Note: Below link is just an example, Not an actual link. Actual links are in README file
#https://drive.google.com/file/d/1ubvKL78De51IacxgGUK60bEhBYCKU507/view?usp=sharing
url = '1ubvKL78De51IacxgGUK60bEhBYCKU507'
gdd.download_file_from_google_drive(file_id = url,dest_path='./data.zip',unzip=True)

#To get the average frame count
import json
import glob
import numpy as np
import cv2
import copy
#change the path accordingly
video_files = glob.glob('/content/drive/MyDrive/dataset/*.mp4')
#video_files1 = glob.glob('/content/dfdc_train_part_0/*.mp4')
#video_files = video_files1
frame_count = []
for video_file in video_files:
    cap = cv2.VideoCapture(video_file)
    if(int(cap.get(cv2.CAP_PROP_FRAME_COUNT))<150):
        video_files.remove(video_file)
        continue
    frame_count.append(int(cap.get(cv2.CAP_PROP_FRAME_COUNT)))
print("frames", frame_count)
print("Total number of videos: ", len(frame_count))
print("Average frame per video:",np.mean(frame_count))

frames [451, 438, 473, 283, 383, 473, 301, 449, 464, 408, 383, 489, 283, 449, 301, 467, 489, 464, 451, 438, 438, 438, 451, 461, 450, 450, 383, 408, 302, 454, 438, 408, 383, 461, 302, 4
Total number of videos: 6223
Average frame per video: 376.08307890085166
  
```

Fig 9.1.1 To get the average frame count and total number of videos



```

success, image = vidObj.read()
if success:
    yield image

im_size = 112
mean = [0.485, 0.456, 0.406]
std = [0.229, 0.224, 0.225]

train_transforms = transforms.Compose([
    transforms.ToPILImage(),
    transforms.Resize((im_size,im_size)),
    transforms.ToTensor(),
    transforms.Normalize(mean,std)])

video_file = glob.glob('/content/drive/My Drive/Face_only_data1/*.mp4')
print("Total no of videos :", len(video_file))
print(video_file)
count = 0;
for i in video_file:
    try:
        count+=1
        validate_video(i,train_transforms)
    except:
        print("Number of video processed: ", count, " Remaining : ", (len(video_file) - count))
        print("Corrupted video is : ", i)
        continue
print((len(video_file) - count))

Total no of videos : 1323
['/content/drive/My Drive/Face_only_data1/id0_id28_0000.mp4', '/content/drive/My Drive/Face_only_data1/id0_id26_0000.mp4', '/content/drive/My Drive/Face_only_data1/id0_id28_0001.mp4',
0
  
```

Fig 9.1.2 To check if there are any corrupted videos in the dataset.

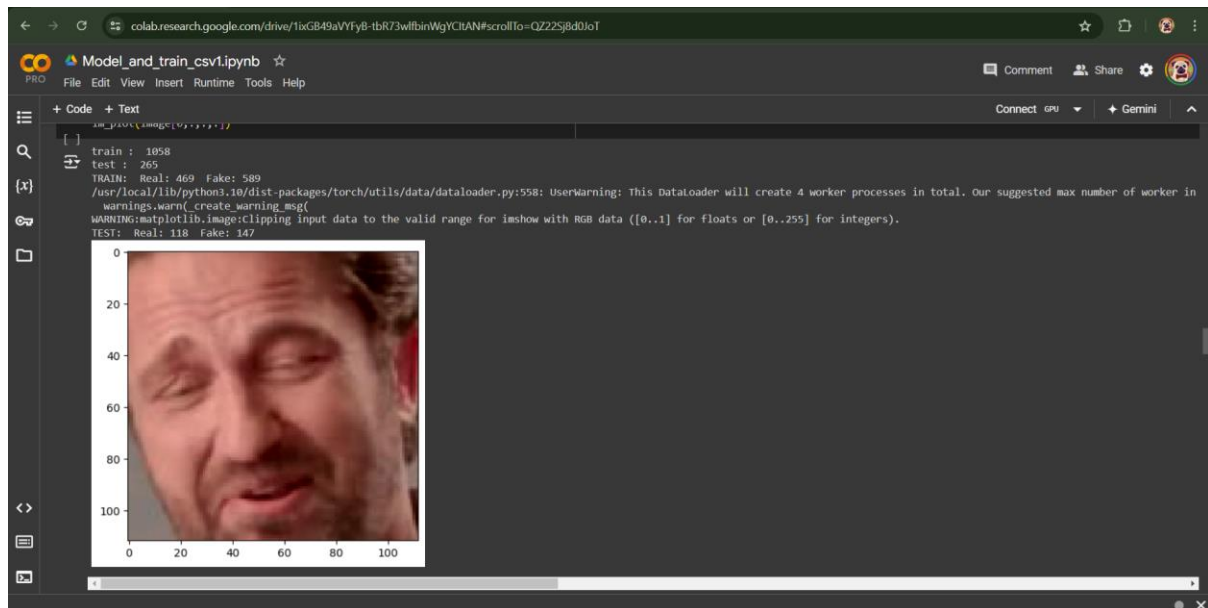


Fig 9.1.3 Number of training and testing videos

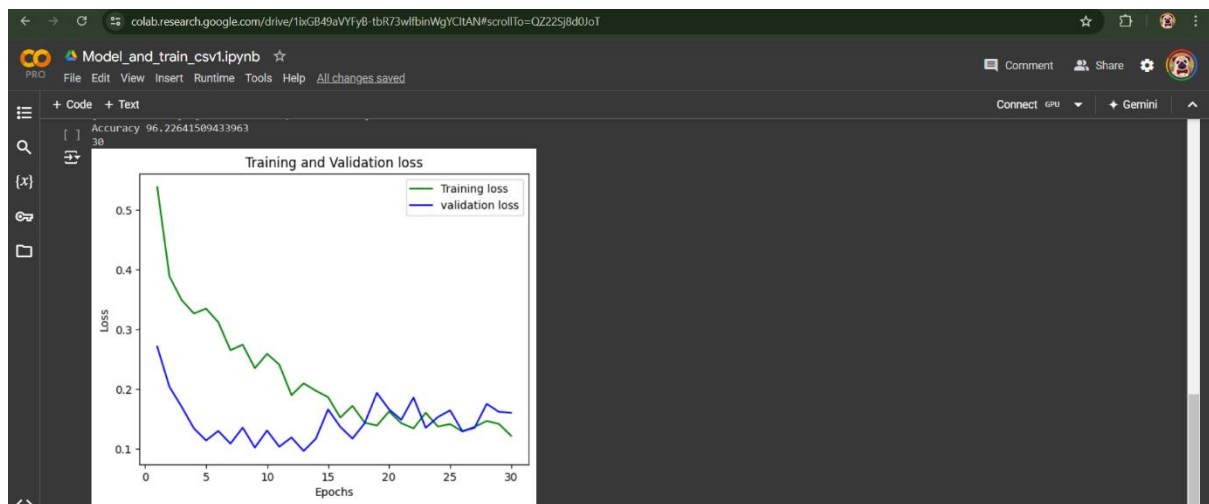


Fig 9.1.4 Graph of training and validation loss where x-axis represents epochs and y-axis represents loss

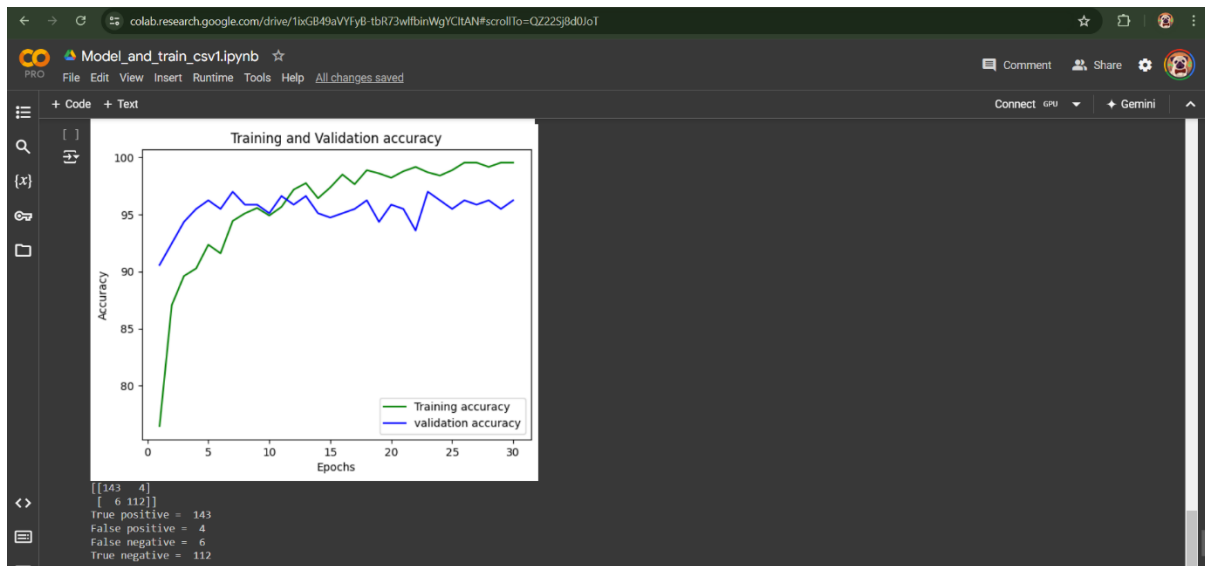


Fig 9.1.5 Graph of training and validation accuracy where x-axis represents epochs and y-axis represents accuracy

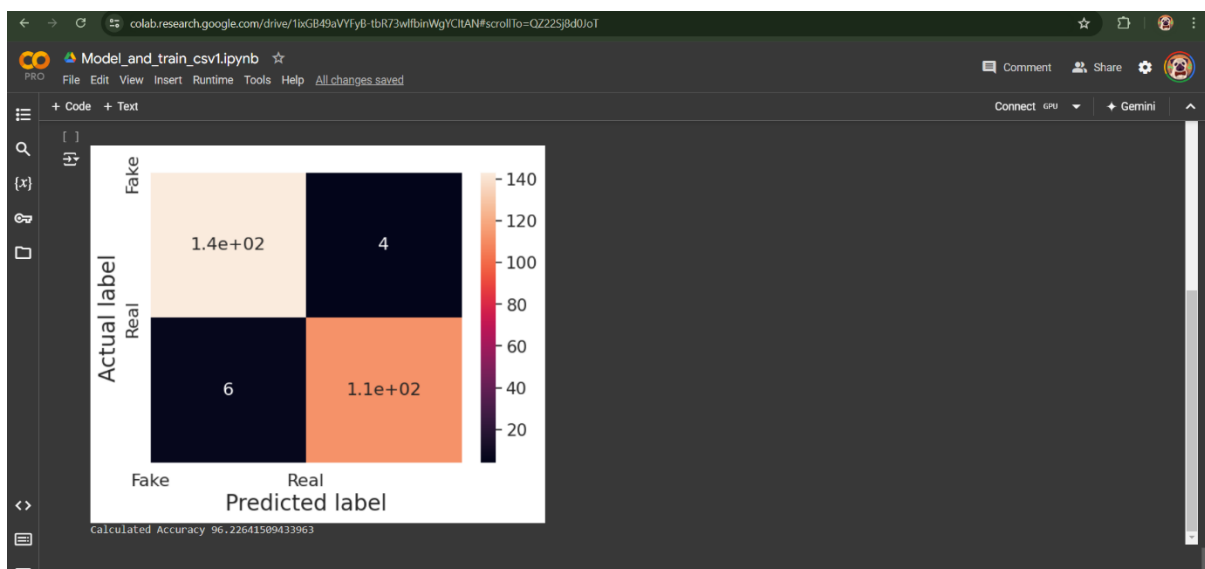


Fig 9.1.6 confusion matrix and accuracy

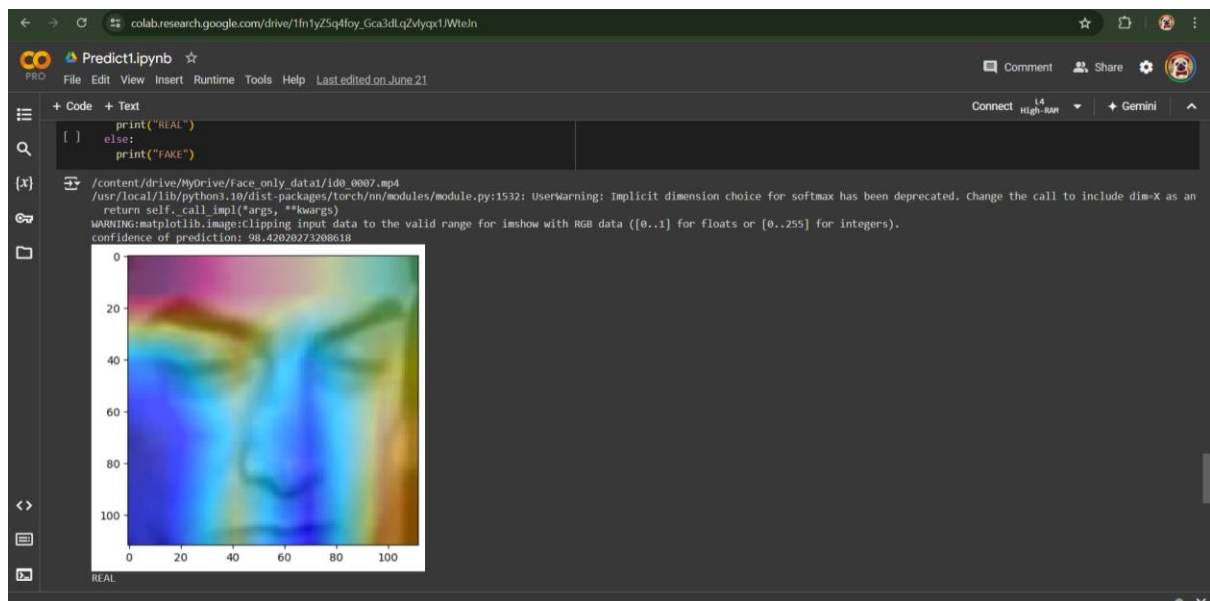


Fig 9.1.7 Real video output

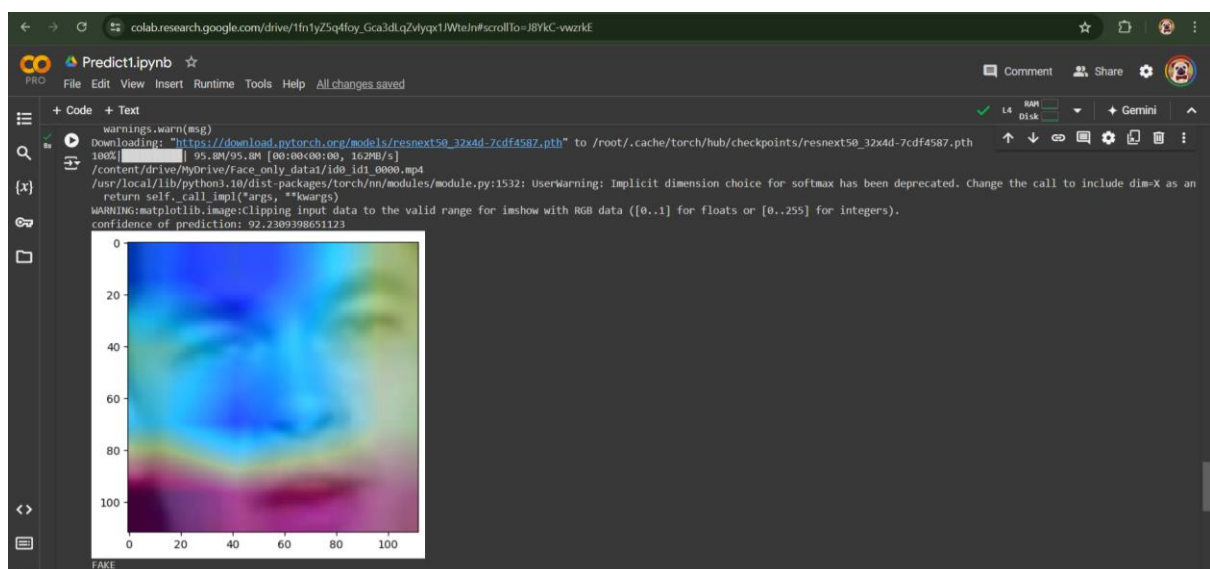


Fig 9.1.8 Fake video output

CHAPTER 10

CONCLUSION

10.1 Conclusion

We presented a neural network-based approach to classify the video as deep fake or real, along with the confidence of proposed model. Our method is capable of predicting the output by processing 1 second of video (10 frames per second) with a good accuracy. We implemented the model by using pre-trained ResNext CNN model to extract the frame level features and LSTM for temporal sequence processing to spot the changes between the t and $t-1$ frame. Our model can process the video in the frame sequence of 10,20,40,60,80,100.

10.2 Future Scope

There is always a scope for enhancements in any developed system, especially when the project build using latest trending technology and has a good scope in future.

- Web based platform can be upscaled to a browser plugin for ease of access to the user.
- Currently only Face Deep Fakes are being detected by the algorithm, but the algorithm can be enhanced in detecting full body deep fakes.

REFERENCES

- [1] Afchar, D., Nozick, V., Yamagishi, J., & Echizen, I. (2018). Mesonet: A compact facial video forgery detection network. Paper presented at the 2018 IEEE International Workshop on Information Forensics and Security (WIFS).
- [2] Ahmed, I., Ahmad, M., Rodrigues, J. J., & Jeon, G. (2021). Edge computing-based person detection system for top view surveillance: Using CenterNet with transfer learning. *Applied Soft Computing*, 107, 107489.
- [3] Y. Mirsky and W. Lee, The creation and detection of deep fakes: A survey, *ACM Comput. Surv.*, vol. 54, no. 1, pp. 141, Jan. 2022.
- [4] M.Masood,M.Nawaz,K.M.Malik,A.Javed,andA.Irtaza, Deepfakes generation and detection: State-of-the-art, open challenges, countermeasures, and way forward, 2021, arXiv:2103.00484.
- [5] R. Tolosana, R. Vera-Rodriguez, J. Fierrez, A. Morales, and Deepfakes and beyond: A survey of face manipulation and detection, *Inf. Fusion*, vol. 64, pp. 131148, Dec. 2020.
- [6] An Overview of ResNet and its Variants : Vision and Pattern Recognition, pages 5967–5976, July 2017. Honolulu, HI.
- [7] R. Raghavendra, Kiran B. Raja, Sushma Venkatesh, and Christoph Busch, “Transferable deep-CNN features for detecting digital and print-scanned morphed face images,” in *CVPRW*. IEEE, 2017.
- [8] Tiago de Freitas Pereira, André Anjos, José Mario De Martino, and Sébastien Marcel, “Can face anti spoofing countermeasures in a real world scenario?,” in *ICB*. IEEE, 2013.
- [9] Nicolas Rahmouni, Vincent Nozick, Junichi Yamagishi, and Isao Echizen, “Distinguishing computer graphics from natural images using convolution neural networks,” in *WIFS*. IEEE, 2017.
- [10] F. Song, X. Tan, X. Liu, and S. [18] P. Chen, J. Liu, T. Liang, G. Zhou, H. Gao, J. Dai, and J. Han, “Fsspotter: Spotting face-swapped video by spatial and temporal clues,” in 2020 IEEE international conference on multimedia and expo (ICME). IEEE, 2020, pp. 1–6.47, no. 9, pp. 2825–2838, 2014.
- [11] D. E. King, “Dlib-ml: A machine learning toolkit,” *JMLR*, vol. 10, pp. 1755–1758, 2009.
- [12] S. A. Khan, A. Artusi, and H. Dai, “Adversarially robust deepfake media detection using fused convolutional neural network predictions,” arXiv preprint arXiv:2102.05950, 2021.

