DEPT. OF CEN

AMRITA SCHOOL OF ENGINEERING

21AIE211

Deep Learning for Signal and Image Processing

PROJECT REPORT

DeepFake Classification

Submitted by:

TEAM : 13

Penaka Vishnu Reddy (CB.EN.U4AIE20048)

Dharavathu Rohith (CB.EN.U4AIE20060)

Svs Dhanush (CB.EN.U4AIE20068)

Tsaliki Satya Ganesh Kumar (CB.EN.U4AIE20073)

Under the supervision of

Ms. Aswathy

ABSTRACT

Keywords - Deep Learning, Deep Fake Videos, CNN, LSTM, ResNeXt

Deep learning is an effective and useful technique that has been widely applied in a variety of fields, including computer vision, machine vision, and natural language processing. Deepfakes uses deep learning technology to manipulate images and videos of a person that humans cannot differentiate them from the real one.

So in our work we try to develop more efficient models to detect whether a video is a real or manipulated video. We will be using Res-Next Conventional Neural Network to extract frame level features and Long Short Term Memory (LSTM) to classify whether a video is fake or real. We will be a various available datasets to train and test our model.

Contents

Chapter 1

INTRODUCTION

First we will try to study about deep fake videos so that we can have better understanding about deep fake video which helps us to create a effective detection model.

1.1 What are Deep Fake Videos

Deepfake videos refer to manipulated or synthesized videos created using deep learning algorithms. These videos involve replacing or superimposing someone's face onto another person's body, creating highly realistic and often indistinguishable results. Deepfake technology has gained attention due to its potential misuse, including spreading misinformation, creating political unrest, facilitating fraud, and compromising personal privacy.

1.2 How Deep Fake videos generated ?

Majority of the tools including the GAN and autoencoders takes a source image and target video as input. These tools split the video into frames, detect the face in the video and replace the source face with target face on each frame. Then the replaced frames are then combined using different pre-trained models. These models also enhance the quality of video my removing the left-over traces by the deepfake creation model. Which result in creation of a deepfake looks realistic in nature. We will be using a similarly approach in detecting the deep fakes, First we convert the videos into frames and learn the features from the images and classify it whether it is fake or real video.

1.3 Motivation behind the project

Deepfake videos are often used for harmless purposes like in memes, social media filters, or face-swap apps. But deepfakes can also be used maliciously to spread misinformation, create fake news, or launch revenge videos. Deepfake technology can be used for many types of deception, from political manipulation and fake news, to revenge porn and blackmail. Anyone with access to deepfake technology can make anyone else look like they are saying or doing pretty much anything. There are concerns that deepfakes will increasingly be used maliciously in the future, calling into question the security of biometric data, including in the use of facial recognition tools. To overcome these problems, Deep fake detection is very important. So we try to built a new deep learning based model which helps in classifying AI generated videos (Deep Fake videos) and Real videos.

1.4 Methods and existing Deep Fake Detection models

Facial and Body Movements, Deepfake videos often exhibit unnatural or distorted facial and body movements. Advanced algorithms can analyze these movements and compare them to expected patterns to identify anomalies. For example, deepfake videos may have incorrect eye blinking, unnatural facial expressions, or mismatched lip movements.

Image forensics techniques analyze the image or video frames to identify artifacts or inconsistencies that may indicate tampering or manipulation. This includes detecting anomalies in pixel patterns, inconsistencies in lighting and shadows, and inconsistencies in facial landmarks.

1.4.1 XceptionNet

- Release Year: 2017
- Authors: François Chollet

XceptionNet, short for Extreme Inception, is a deep convolutional neural network architecture. It is a variant of the InceptionNet model and is designed to have a more efficient and lightweight structure. XceptionNet leverages depthwise separable convolutions to reduce the number of parameters and computations while maintaining high performance. It has been widely used in various computer vision tasks, including image classification

and feature extraction. Although not specifically developed for deepfake detection, It can be applied as a feature extraction backbone in deepfake detection models to extract discriminative features from manipulated videos.

1.4.2 MesoNet

- Release Year: 2018
- Authors: Darius Afchar, Vincent Nozick, Junichi Yamagishi, Isao Echizen

MesoNet is a deep learning-based model designed for the detection of deepfake videos. It focuses on analyzing the subtle artifacts introduced during the compression of manipulated videos. MesoNet utilizes a lightweight CNN architecture and is trained on a large dataset of real and manipulated videos. It can effectively classify videos as either authentic or manipulated, helping to identify deepfake content.

These are some of the methods and existing models for deep fake detection.

1.5 Overview of our project

So in our project we will try to build the model which will predict whether a video is real or deep faked video we will even try to create an interface for easy use of our application and to make it available to everyone. We will try train our model on the collected data from various datasets which helps our model to learn various features.

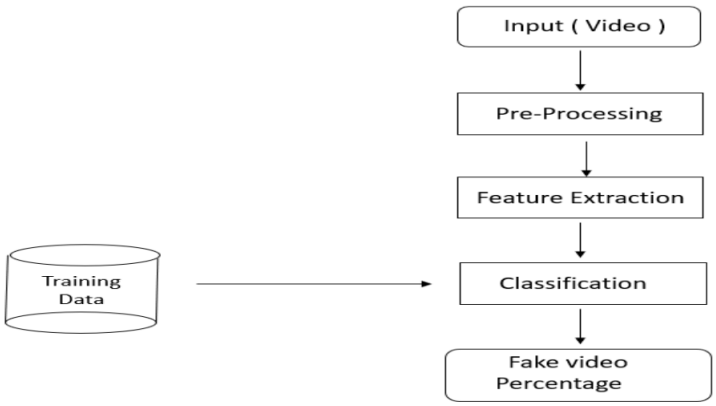

Figure 1.1: Overview of our project

Chapter 2

Literature Survey

Title	Year	Dataset	Contribution to Project
Aggregated Residual Transformations for Deep Neural Networks	2017	Imagenet	We used this model to develop our model
Methods of deepfake detection based on machine learning	2020	Celeb-DF	Found the indicators that can distinguish whether face manipulation is applied on any media
Deep Learning for Deepfakes Creation and Detection: A Survey	2022	–	Studied various algorithms working
Trusted Media Challenge Dataset and User Study	2022	–	Understood about our dataset

The table presented above provides a concise summary of the literature review conducted for our project. It outlines key contributions from various papers and datasets that have significantly influenced our work. Each row in the table represents a specific title, year, dataset, and the corresponding contribution it made to our project. This literature review played a crucial role in guiding our research and understanding the existing methodologies and advancements in the field.

Chapter 3

Dataset

In order to make our model more efficient towards classifying more accurately we will train our model with various datasets. This makes our model to explore and learn various features from the dataset. We will be using the DFDC dataset and Face Forensic++ dataset, the DFDC dataset is a video dataset which is collected by Facebook AI. It contains REAL and FAKE (Deepfaked) videos where each video is around 10 secs and the dataset is stored in AWS S3 so for our project we will be using only a small subset of the dataset by considering our computational power we wil be taking the sample of DFDC dataset as it is very large dataset i.e 3000 videos where there are 1500 REAL and 1500 FAKE videos we try to take equal number of REAL and FAKE videos from both the datasets inorder to avoid bias and 2000 videos from Face Forensic++ dataset where 1000 videos are REAL and 1000 videos are FAKE. So our combined dataset will be containing a total of around 5000 videos where 2500 videos are REAL and 2500 videos are FAKE videos. We will go with the 70, 20, 10 split ratios for training, validation, and testing splits and we try to make sure that there equal number of REAL and FAKE videos in these each splits. Our dataset contains two labels i.e the id of the video and whether that video is FAKE or REAL.

Chapter 4

Architecture

4.1 Pre-processing

The first step in our model design is the pre processsing of the data. This is an important step because we have collected various videos from various datatsets with different video length and resolutions. We will even try to remove the corrupted videos from our dataset and also the audio from the videos and we keep only the the required portion of the video which the face region.

4.1.1 Video to Frames

This is the first step in pre processing process, as we know videos are nothing but a group of stacked images, we convert those video into frames to do the further prep rocessing steps i.e face detection.

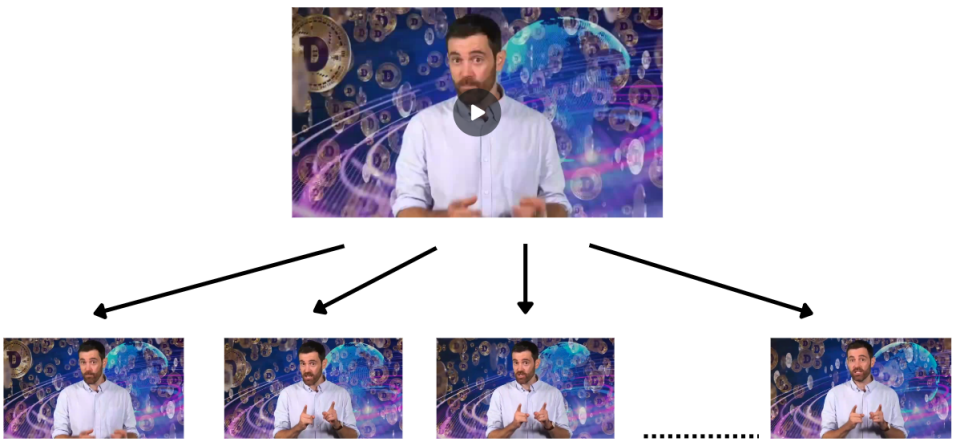

Figure 4.1: Videos are converted into frames

4.1.2 Face Detection

In this process we will detect the face region from the extracted frames from the videos, for this we will be using some face recognition packages available. We do this because it makes our to focus only on the important part of the video which the face and make it easy to learn the features, variations.

4.1.3 Frames to Videos

Now after cropping the frames we stacked them back as videos, we do this step only while training the model, we can neglect this when we try to predict the video through using the interface. The frames which does not contain the face are removed.

Figure 4.2: The processed frames are stacked back

To maintain the uniformity of number of frames and also considering the computational limits, we have selected a threshold value for no. of frames i.e 150 frames at 30FPS while saving the new pre-processed video. To demonstrate the proper use of Long Short-Term Memory (LSTM) we have considered the frames in the sequential manner i.e. first 150 frames and not randomly. So, the newly created video is saved at frame rate of 30 fps which makes it a 5 sec video (150/30) and resolution of 112 x 112.

4.2 Dataset Split

We split our dataset into 70, 20, 10 percentages for train, validation, test respectively. As we have combined two different datasets we will try to train our model in two different ways one with overall 5000 and another with only the Face Forensic++ dataset which consists of 2000 videos. In all the training, testing, validation dataset the no. of real and fake videos contains equally.

4.3 Layer Details

We are using a combination of CNN and RNN layers. We use the pre-trained model ResNeXT for feature extraction from the processed videos and pass it the sequential LSTM layer for classification task.

4.3.1 ResNeXT

We will be using ResNeXt50-32*4d for feature extraction. We will be using the pre-trained model of ResNext instead of building the entire code and training it from scratch. The model architecture that was proposed by researchers at Facebook AI Research (FAIR) in 2017. The ResNeXt architecture is an extension of the ResNet (Residual Network) architecture and is designed to improve the performance of convolutional neural networks (CNNs) on various computer vision tasks. 50 represents the number of layers i.e it has 50 convolutional layers and 32*4d represents the cardinality of the model. The ”32” indicates that the model uses 32 separate paths, which mean the model has 32*4 dimensions.

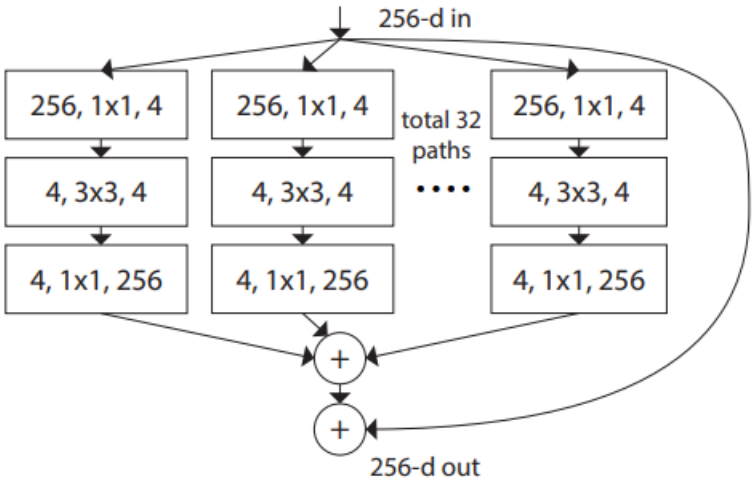

Figure 4.3: ResNeXT50 32*4d Architecture

4.3.2 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.

4.3.3 LSTM

LSTM (Long Short-Term Memory) is a type of recurrent neural network (RNN) architecture that is commonly used for sequential data processing tasks, such as natural language processing and time series analysis. LSTM contains three gates input gate, forget gate, output gate. The input gate determines how much information from the current time step should be stored in the memory state. The forget gate controls the extent to which the previous memory state is retained or forgotten. The output gate determines the extent to which the current memory state influences the output of the LSTM at the current time step.

4.3.4 Leaky ReLU

Leaky ReLU is an activation function that introduces a small slope to negative values, preventing the complete elimination of negative inputs. Its formula is LReLU(x) = max(a * x, x), where ”a” is a small positive constant, we have taken the default value which is 0.01. This small slope helps mitigate the ”dying ReLU” problem and allows for better learning convergence in neural networks.

4.3.5 Droupout

The dropout layer is a regularization technique commonly used in neural networks, including recurrent neural networks (RNNs) such as LSTM. It is designed to prevent overfitting, which occurs when a model learns to memorize the training data instead of generalizing well to unseen data.

4.3.6 Adaptive average pooling

The adaptive average pooling layer is a type of pooling layer commonly used in convolutional neural networks (CNNs). Unlike traditional pooling layers with fixed pooling sizes, adaptive average pooling allows the network to dynamically adapt the pooling operation based on the input size and the target output size. In average pooling, the purpose is to reduce the spatial dimensions of the input feature maps while preserving important information.

4.3.7 Model Architecture

Our model contains 6 layers (considering ResNeXT as a single layer). Below is the structure of the and order of the layers.

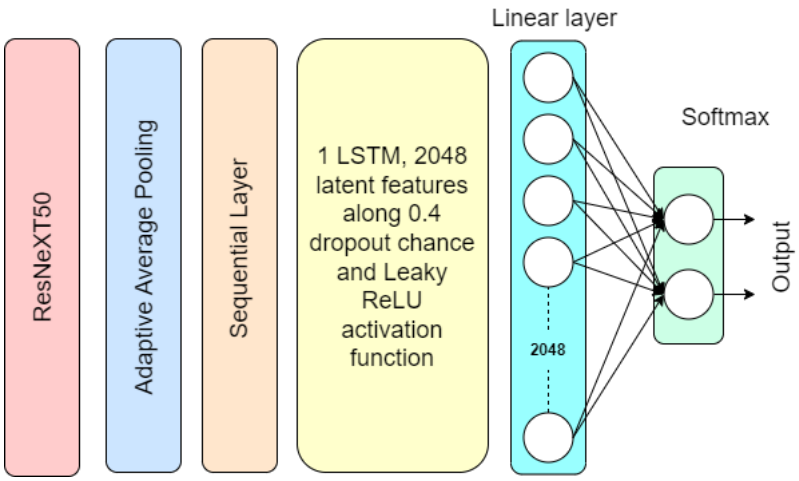

Figure 4.4: Model Architecture

The in detailed architecture of ResNeXT50 32*4d is as follows and it has 25×10^6 parameters.

stage	output	**ResNeXt-50 (32×4d)**
conv1	112×112	7×7, 64, stride 2
conv2	56×56	3×3 max pool, stride 2 [1×1, 128 3×3, 128, C=32 1×1, 256] ×3
conv3	28×28	[1×1, 256 3×3, 256, C=32 1×1, 512] ×4
conv4	14×14	[1×1, 512 3×3, 512, C=32 1×1, 1024] ×6
conv5	7×7	[1×1, 1024 3×3, 1024, C=32 1×1, 2048] ×3
	1×1	global average pool 1000-d fc, softmax
# params.		**25.0**×10^6

Figure 4.5: ResNeXT50 layers

Chapter 5

Results and Discussions

As we have trained our model in two ways, on with FF++ dataset with a frame rate of 20 and other with combined dataset with frame rate of 60. We have obtained a 88.38 percent accuracy for model trained with FF++ dataset and the results as follows.

Figure 5.1: Train vs Validation loss

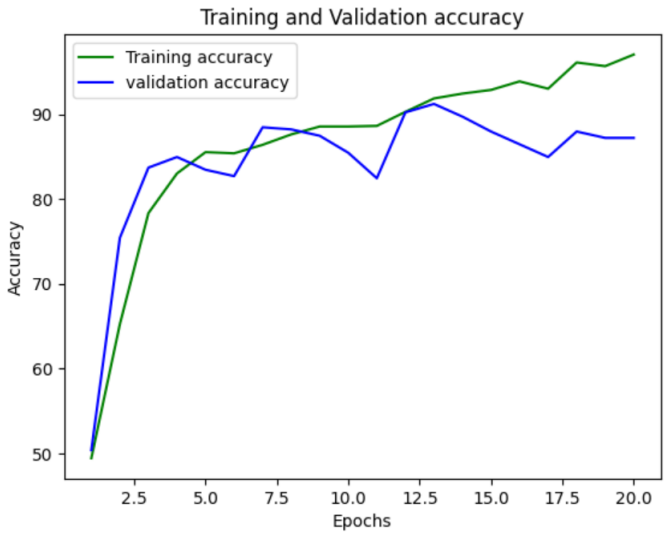

Figure 5.2: Train vs Validation accuracy

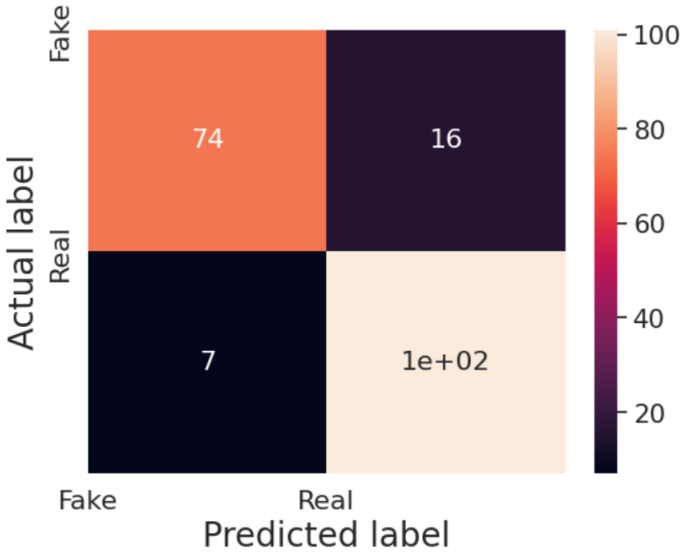

Figure 5.3: Confusion matrix for the test data

Below image is an example of based on what regions the models tries to predict whether a video is fake or real. The regions various from frame to frame and video to video.

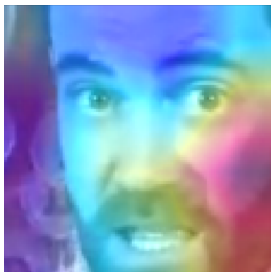

Figure 5.4: Heat map of one of the frames

With the 5000 dataset (the combined dataset) we have obtained an accuracy of 91.59 percent. Below is some of the images on what this model tries to learn and predict from the frames.

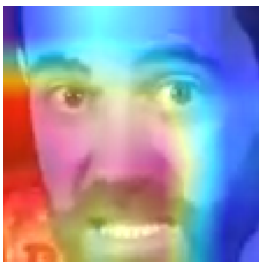

Figure 5.5: REAL video predicted as REAL

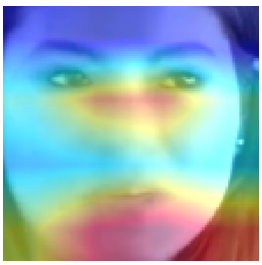

Figure 5.6: FAKE video predicted as REAL

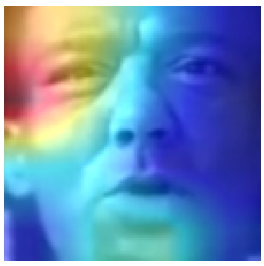

Figure 5.7: FAKE video predicted as FAKE

Some of times even our model predicts/classifies the videos wrongly.

5.0.1 GUI

We have created a basic and simple GUI using gradio package available in python, first you need to upload the video into the drive and then type the video name in the input cell available on the interface, it takes sometime do all the preprocessing and classification steps and at the output it displays whether it is a FAKE or REAL video. Below is an glimpse of the GUI.

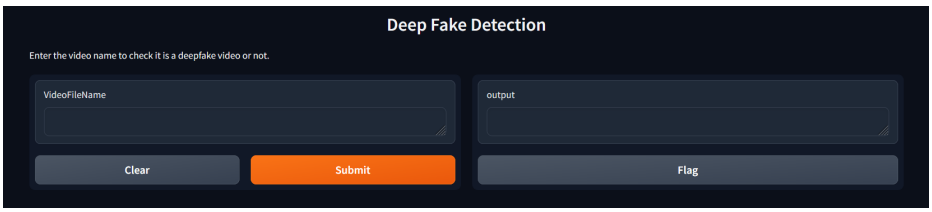

Figure 5.8: GUI Interface

Now upload a video into the drive account with is connected with the code and enter the video name in the input blank. Below is single frame from the fake video sample we have taken and its respective output.

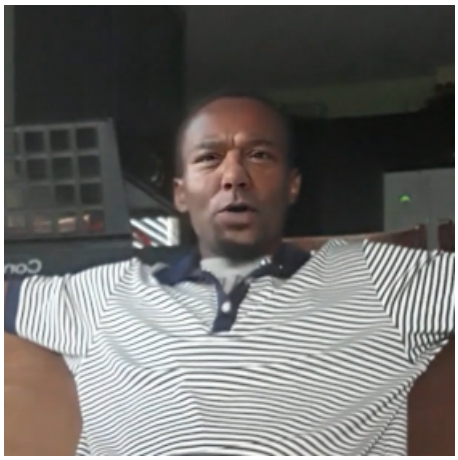

Figure 5.9: Frame from a fake video sample

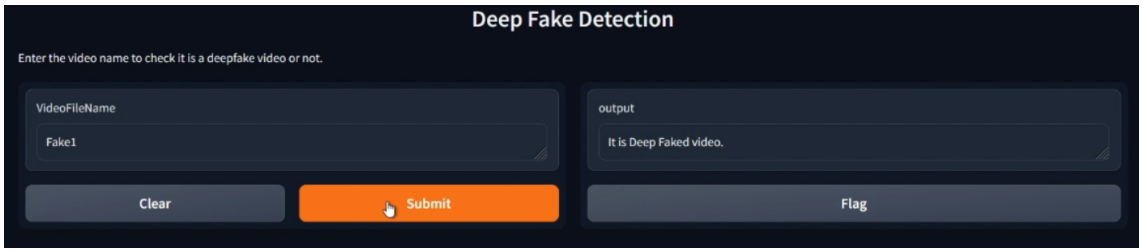

Figure 5.10: GUI output displaying it is a fake video

Chapter 6

Conclusion

We treid to built a model using ResNeXT50 (CNN) for feature extraction and LSTM for temporal sequence processing to spot the changes between the t and t-1 frame based on the features we have extracted and we have tried train our model on different datasets and frames rates 20, 60 and based on the results we noticed more the number of frames more the information more the accuracy. Many new ways have been developing to generate deep fake videos, so do we need new ways to detect them too. Some of the draw backs of our model is that it doesnt classify based on the audio.

Chapter 7

References

Aggregated Residual Transformations for Deep Neural Networks

Methods of deepfake detection based on machine learning

Trusted Media Challenge Dataset and User Study

– END –