Deepfake detection
Using Neural Networks

by

Sadman Sakib
16301082
Mir Tarid Al Abid
17101536
Nures Saba Tiana
18101229
Wajida Anwar Asha
18101290
Syed Mahbubul Huq
21141043

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
September 2021

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

[illegible]

Sadman Sakib
16301082

[illegible]

Mir Tarid Al Abid
17101536

Tiana

Nures Saba Tiana
18101229

Wajida Anwar Asha

Wajida Anwar Asha
18101290

[illegible]

Syed Mahbubul Huq
21141043

Approval

The thesis titled "Deepfake Detection Using Neural Networks" submitted by

1. Sadman Sakib (16301082)
2. Mir Tarid Al Abid (17101536)
3. Nures Saba Tiana (18101229)
4. Wajida Anwar Asha (18101290)
5. Syed Mahbubul Huq (21141043)

Of Summer, 2021 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on September, 2021.

Examining Committee:

Supervisor:
(Member)

Iqbal

Dr. Muhammad Iqbal Hossain
Assistant Professor
Department of Computer Science and Engineering
Brac University

Co Supervisor:
(Member)

[illegible]

Mr.Mohammed Abid Abrar
Lecturer
Department of Computer Science and Engineering
Brac University

Program Coordinator:
(Member)

Md. Golam Rabiul Alam
Associate Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chairperson)

Sadia Hamid Kazi,Ph.D.
Chairperson and Associate Professor
Department of Computer Science and Engineering
Brac University

Abstract

Deepfake is a sort of artificial intelligence that forge original image or video and create persuading images, audio and video hoaxes by utilizing two contending AI algorithms-the generator and discriminator that form a generative adversarial network (GAN). The term ‘Deepfake’ started in 2017, when a mysterious Reddit user called himself ”Deepfakes.” The user ”Deepfakes” supplanted genuine faces with celebrity faces. With the rapid advancement of modern technology, Deepfakes have become an emerging problem, as deepfakes can threaten cybersecurity, political elections, companies, individual and corporate finances, reputations, and more. Therefore, this makes deepfake detection more and more urgent. Although, a lot of techniques has been invented to detect deepfake but not all of them works perfectly and accurately for all cases. Also, as more up to date deepfake creation strategies are grown, ineffectively generalizing methodologies should be continually refreshed to cover these new techniques. Our research focuses on the recent techniques that are used to create manipulated videos and detect them though ensembling different CNN models.

Keywords: Deepfake; Detect; Neural Networks; Video; Images

Acknowledgement

At first we would like to thank Almighty Allah for keeping us safe and healthy throughout the whole pandemic. We also would like to thank our supervisor and co supervisor for guiding us and bearing our mistakes throughout the whole span of this research. We were able to conduct this research because of their guidelines and constant support.

Table of Contents

List of Figures

List of Tables

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

CNN	Convolutional Neural Network
RNN	Recurrent Neural Network

Chapter 1

Introduction

1.1 Overview

In recent times deepfake videos are being spread all over the popular social media sites. These manipulative videos are mainly generated by replacing a person's face with another person along with facial expressions to make it realistic. Deepfakes can be far more dangerous than expected as it targets the mass of people and most of the people are unaware of the bluff spread by it. Deepfakes are mostly used in negative purposes such as to spread fake news, fake surveillance videos, pornography, fake evidence and extortion. There are many softwares and apps that can generate near realistic deepfake images and videos and they are openly available to the general people. In our digital world, video is one of the most prime ways of communication passage and also one of the major ways of sharing information. If after hearing or getting any information through video we come to know that information is false and has some negative intentions, our trust from this digital and modern technology-based world would be very much lost. To regain this trust to grow a strong trust for technology, a method and technique to solve this problem is must.Therefore, it has become a prime need to find an algorithm that can efficiently distinguish between the real and the fake videos to reduce the amount of bluffs that can spread due to these manipulated videos. Many institutions and individuals have done several research and found few techniques and methods to solve this problem. Among them the use of GAN is widely seen, besides the use of neural networks in this field is major and very much vast.

1.2 Motivation

Artificial Intelligence (AI) has come a long way since 1956 and with its advancement and arrival of deep learning techniques, manipulated digital image and video contents have increased in a huge number in the recent years. Manipulated footages, audios, images, and videos; which are known as deepfakes can have dangerous and scary impacts and might have the ability to alter the truth and take away people's trust or may lead them to believe in something that is unreal. At the beginning of 2020 a clear shift came across in the response to deepfake technology, when Facebook being one of the most popular social media made a public statement to ban fake videos and images on Facebook's platforms [15]. With the span of time and technology these Deepfake videos have become so persuasive that it has become very hard and challenging to differentiate between a real and fake video by just looking at it. In our digital world, video is one of the most prime ways of communication passage and also one of the major ways of sharing information. If after hearing or getting any information through video we come to know that information is false and has some negative intentions, our trust from this digital and modern technology-based world would be very much lost. To regain this trust to grow a strong trust for technology, a method and technique to solve this problem is must. Rapid development of deep neural network has paved the way for many techniques of detecting deepfake videos or images such as Convolutional Neural Networks (CNN), Recurrent Neural Network (RNN), Generative Adversarial Network (GAN), Attribution Based Confidence (abc) Metrics and so on. Our aim is to find an effective algorithm and enhance the effectiveness to strengthen the generated output.

1.3 Problem Statement

Unlike previous times, producing deepfake videos is much easier in recent times. General people with minimal or no programming knowledge can also make these realistic deepfake photos or videos by using different face manipulation apps. Most of these applications are free and publicly available. Recently, a manipulated video got 1 million views on Twitter in which it was made to look like Joe Biden didn't know what state he is. Most of these popular deepfake videos include clips of a public figure saying or doing silly or scandalous activities. Such manipulated videos can be spread to defame anyone and also can give rise to chaos and political instability. Therefore, to detect these manipulated videos and to prevent the spread of misinformation we will be coming up with a model that can detect Deepfakes.

1.4 Aims and Objectives

Our research target is to develop a robust deepfake detection system for detecting fake clips of people used in a forged video. A lot of methods have been proposed previously for detecting deepfakes but those methods have some limitations. We aim to find solutions for those anomalies, in our research. The objectives of this research are:

- To deeply understand what deepfake is and how it is created.
- To find the shortcomings of previous deepfake detection methods and find and test their possible solution.
- To propose a robust and generalized deepfake detection method; evaluate and test them.
- To compare our method with the existing methods and try to make the existing models work better.
- To offer suggestions to make our model better.

1.5 Thesis Structure

In Chapter 1 we have given an overview and introduction on what deepfake is, how they are created and detected. We have discussed the previous researches and works in this field in Chapter 2. In the beginning of Chapter 3 our proposed model, dataset and pre processing technique description is given. Then we explained our model and how this model works. In chapter 4 we illustrated the implementation of the models and at last we discussed the results obtained from our model.

Chapter 2

Related Work

DeepFakes tremendously hamper the trust of common people towards the information shared in various online platforms. Besides, it puts a huge pressure on the news publishers and other social media because sometimes the outputs from various deepfake algorithms are so realistic that it becomes very hard to differentiate them from the real images or videos. Thus, DeepFakes are used as an effective tool to spread misinformation about mostly famous figures by making them go viral over different platforms. Moreover, deepfakes are also used in committing virtual crimes. For instance, an audio Deepfake was used to scam a CEO out of 243,000 dollars.[6]

According to [13] , through Remote visual photoplethysmography, the change in skin color due to blood circulation can be monitored. The heartbeat rhythms in manipulated videos remain distorted or entirely broken which can be a powerful indicator to detect a manipulated video. A dual-spatial-temporal attention is used in this method to acclimatize to constantly changing face manipulation types. The same source states that by using GAN it is possible to produce such realistic photos or videos that sometimes can fool different detection algorithms. However, the authors claim that pulse signals are not preserved by GANs in exactly the same way they are, so the unaltered and manipulated videos can be differentiated by focusing on the pulse signals.

The problem with most of the existing methods is that in most cases they are dependent on image and video classification based on deep learning based algorithms which take inputs from the raw pixel-domain DeepFake. They might not be very effective in the future in detecting manipulated video and images when the DeepFakes become increasingly naturalistic since the image generation using deep methods themselves are becoming more developed day by day. To make the method robust under several degradation, they have integrated a heart rhythm motion amplification module and a learnable spatial-temporal attention mechanism at different layers of the network.

Again, in another research paper, the authors proposed a system based on watermarking technology that detects Deepfake news generated from existing authentic video clips via face-swap and lip-sync[9]. According to their model, the watermarks are embedded in audio and video tracks at fine granularity and specificity to allow reliable detection at an extremely low false positive rate. The embedded marks are sensitive to changes that target the video integrity. They impart unique identities to the host video linking the video to its source and provenance in a blockchain network. The video and audio watermarks are tightly coupled, and they cross reference each other to verify the integrity of the video and provide extended payload capacity, unique content identification, and an adequate level of security. This coupling allows these watermarks to verify that an audio segment has not been replaced as is often done in making Deepfakes via voice impersonation. The frame watermark can localize frame changes resulting from face swapping, which is commonly used for making Deepfakes. Only authorized entities have access to the watermark embedder. Also, a secure watermark reader employs detection protocol known only to an authorized party. The system can transcribe the suspected audio and check if the transcription matches the transcription stored in the metadata stored in blockchain.

In the paper [12] , they used a method that is established on convolutional neural networks (CNNs) and recurrent neural networks (RNNs). Along with CNN and RNN a weighting mechanism is combined with a Gated Recurrent Unit (GRU). The method works in three steps. At first, face detection operation using MTCNN is performed across multiple frames, Then, features are extracted with a CNN, and finally, they predict and estimate with a layer called Automatic Face Weighting (AFW) along with a Gated Recurrent Unit (GRU). The motivation behind their research was to correctly detect facial forgery in videos by focusing on the face. Their method retrieves visual and temporal features from faces that are detected in videos to accurately detect manipulations by an automatic weighting mechanism to focus on the most reliable regions where faces are found and drop the least reliable ones when a video-level prediction is done. Also, the frame was downscaled by a factor of 4 for the purpose of speeding up the face detection process. Besides, to avoid false positives and detect blurry faces, an automatic weighting mechanism is used. A logit and weight for each frame is estimated by the backbone model by not using information from other frames. If this method is adapted to detect deepfake by focusing on only regions where faces are detected, the accuracy of detection improves significantly and adding an automatic face weighting layer as well as a GRU increase the accuracy more. Also, they evaluated the accuracy in each level of the detection process by comparing their result with existing results and methods.

Again, in the paper [16], DeepTag, a deep learning-based approach is used. It is a simple but very effective encoder and decoder design that can retrieve the implanted message from images even after many extreme GAN-based DeepFake transformations with high confidence, is proposed. This method is kind of similar to digital watermarking. Their motivation was to implement a robust encoder and decoder that is DNN based and ensure that the implanted message remains unchanged after several extreme GAN-based transformations and help to detect DeepFake easily. They have proposed a image tagging method that is robust and can prevent image forgery and can protect the safety of facial images in social medias against DeepFake to overcome the limitations of generalization and robustness. The image tagging permits to efforlessly perform DeepFake detection and find the origin of the images using the embedded messages. According to the author, this kind of facial image tagging approach for DeepFake provenance is not used before. According to experimental results, that DeepTag attains an exactness of more than 95.1 percent on the high-resolution facial images. DeepTag could give more than 80 percent accurate results even when the images are compressed 80 percent. This method was applied on three typical GANs including the complete synthesis and partial synthesis to measure the effectiveness in implanting watermarking into facial images and retrieving them from facial images after extreme GAN-based transformation.

According to the paper [19] , they proposed Dynamic Prototype Network (DPNet) that uses dynamic representations to explain deepfake visual dynamics. They formulated temporal logic specifications focusing on these prototypes to examine the acceptance of their model to desired temporal behaviors. Their motivation was to explain why a video has been detected as DeepFake. They have solved the interpretability issue. Though there are many methods to detect deepfake but there are very few works on addressing the interpretability issue. Previous works have made a good progress in detecting DepFakes to a certain extent but it's also important to explain why that prediction is made. This paper has solved that issue. They propose an interpretable prototype-based neural network called DPNet which captures dynamic features like abnormal movements and temporal artifacts, and uses them to describe the reason behind their particular prediction. They checked the robustness of their proposed model, DPNet and interpretable baselines against temporal specifications.Both of their approaches serve the key frame specification with high percentage. DPNet performs somewhat better, particularly with the fake traces.

In another paper [7], ethereum smart contracts is used to trace and track down the origin of a content. Hashes of the interplanetary file system (IPFS) is used to reserve digital content and its metadata. This paper mainly emphasize on video content though they say that their paper is valid to be used with any other digital medium. The motivation of this paper was finding the real artist (Creator of the video) and going to the root even if the video was copied multiple times. Solidity language is compiled and run by using the Remix IDE. Remix IDE enable the user to write and run the codes of a smart contract. It also has facility to debug and test the solidity code. Here after a video is created the information of the creator and everything else are stored in IPFS DB. To edit another, the creator has to give permission to the seeker. If permission is given the list will hold all the entries that it has given permission to. By following this method, the root and source from which the video was made can be found and if a reliable source or root is not found from the video source then it is assumed that the video is not real. As a result, it is always bound to give a solution and result. This paper provides very good results and the efficiency is also high but the major drawback is that it works for a selected network and the paper was also tested for that particular case of ethereum network where vast and major other data sets were ignored.

In the paper [11] they presented a way to compare certain video clips with a full-sized video and tried to find the similarities and differences between them. They tried to analyze and prove that certain audio and visual clips of a video must have similarities in behavior. The motivation of this paper is to find out and detect fakes by comparing clips and the word "emotion" was used to mean the behavioral changes of certain characters within the clip. Besides, they tried to compare audio and visuals. They used Siamese network-based architecture to differentiate if a video is unaltered or manipulated. During the training phase, they passed an original video as well as its deepfake through their network and obtained modality and emotion embedding vectors for the face and speech of the subject from the videos they passed. The obtained embedding vectors were used to calculate triplet loss function to lower the likeness between the modalities from the fake video and increase the likeness between modalities for the real video. In the Siamese network two neural networks are trained together that share the same weights. Besides, OpenFace and pyAudioAnalysis are used. In their method they at first pre-trained their system. If two modalities that represent the same emotion differ in terms of appearance, the identified features are slike and should be correlated, this is the concept used in psychological point of view and they are successful enough in this. Moreover, the datasets used are bigger and the details of the model used to manipulate the manipulated videos from the unalteres videos is not revealed. Also, DFDC is the only dataset which consists of a combination of videos with manipulated faces, audio, or both. All the other datasets are made up of only manipulated faces not manipulated audio. However, there are some drawbacks, only DeepFakeTIMIT dataset and DFDC dataset are used. Besides they only made an assumption that the fake video will have any error as the videos are long enough, but if the clip is very small there is no room for comparison in the sample. This technique can also detect a inconsistency in the modalities of real videos, and erroneously analyze them as fake because of indifferent human emotions and their nature. This method can only detect one person in a single video. Visual artifacts across frames are not used here. Improved techniques for using audio cues could be used. They also mentioned a few failures in their methods even from their selected datasets. They mentioned many ways of improvement of their paper and method by combining multiple other methods in their system.

Also, the paper [3] is about fake image detection using a deep convolutional neural network. The fake images are created through several applications using GAN. The authors used GAN to create fake images then applied DNN to extract the fake images with multiple resolutions datasets of images. They applied DNN for robust face feature extraction. Their motivation was to use their DNN method to extract fake images from real one's rather than previous conventional methods due to the rapid development of identity theft after GAN was introduced. The authors used a mixture of methods and added them to their six layers of image detection and analysis. Firstly, they used DC-GANs to generate images with a resolution of 64x64. Secondly, they used PG-GAN's to generate images with a resolution of 256x256, 1024x1024. After generating fake images, they applied a deep face recognition system. They used VGG-Net for image extraction. Then the layer was connected with K-way softmax after face recognition for fine tuning and classification. The mixture methods they used worked for different resolutions of images and their classification. For example, only using one method like DC-GAN the image had to be enhanced or geometry of those fake images had to be extracted differently. Moreover, they used VGGFace for extraction of images which has five-layer blocks and max-pooling layers in each block for extraction. So, this method works and they have 80 percent accuracy. The drawbacks in this method is that they applied their method on a high-end GPU (GTX 1080) and 16 Gb of ram. So, the environmental cost is relatively higher than conventional methods. Besides, now we have latest gen face recognition APIs such as Kairos Face recognition API, Lambda Labs API, Animetrics Face Recognition API etc. to gain more accuracy then they got which was 80 percent not significantly higher.

In the paper [14], a deep learning method called DeepfakeStack is used by evaluating many DL based state-of-art models used in ensembled fashion. To perform the experiment and assess the proposed method, they used the FaceForensics++ dataset. Visual Computing Group (VGG), which is an active research group on computer vision, computer graphics, and machine learning collected the dataset. They followed Greedy Layer-wise Pretraining (GLP) technique. The GLP repeatedly adds a new hidden layer to a model and redevelop the model. Their motivation was to meet the challenges posed by Deepfake multimedia, they wanted to make something for detecting manipulated videos to defend forged contents, political campaigns, e-crimes etc. According to the authors, their work resolve the binary classification problem. They labeled 0 for real and 1 for Deepfake and calculated accuracy and categorical log loss. DeepfakeStack architecture consists of two or more base learners that is called level-0 models and a meta-learner is called level-1 model. Level-1 model merges the predictions of level-0 models. The level-1 model is trained on the predictions that are made by base models on out-of-sample data. The data that is not used to train the base models is passed through the base models. Then predictions are made and from these predictions, the input and output pairs of the training dataset as well as the expected outputs used to fit the meta-model are found. The proposed model incorporates a series of DL based state-of-art classification models and constructs an improved composite classifier. Based on their observation from experiments, it is found that DeepfakeStack performs much better than other classifiers by achieving an accuracy of 99.65 percent and AUROC score of 1.0 in detecting Deepfake. So, their method produces a solid basis for constructing a Realtime Deepfake detector.

Lastly in the paper [4] , contains the convolutional neural networks(CNN) model for optical flow known as PWC-Net. The base o their method is pyramidal processing and enumerate the optical flow. Moreover, they used three face sythesis methods called Deepfakes, Face2Face and FaceSwap. Their motivation was to utilize possible inter-frame variance to identify the deepfake video by using optical flow fields. Recent Artificial intelligence based technologies have made creating extremely realistic counterfeit videos very easy. They solved the problem using a pretrained network where they represented motion vectors as a 3-channel image and then used them as input for a neural network. This method is not like other state-of-the-art methods which resort to single video frames. Instead they proposed the adoption of optical flow fields to utilize as many as possible inter-frame variance. The purpose of exploiting optical flow field dissimilarities as a clue to differentiate between deepfake videos and real videos has been introduced and analyzed. For the training, they trained a generic net on randomly left-right flipped squared patches of size 224×224 pixels randomly chosen on a bigger patch of 300×300 that contain the face. They used Adam optimizer and used three computerized face manipulation methods like Deepfakes, Face2Face and FaceSwap.

While reviewing the previous works, we have come up with some limitations that we came across:
Haya and Salah et al. [7] proposed a method to trace back a video to its original source to find out if it is fake or whether it is copied multiple times to generate deepfake video with face swapping. They used ethereum smart contracts which uses hashes of the interplanetary file system (IPFS) used to resrve digital content and

its metadata to trace and track down the source of the original video. That's a good method to find the original content creator and check if it is valid or not however there are many cases where there are original videos created without proper ethereum data. The computational cost was relatively higher to buy ether tokens that had to be bought here for the transactions in ethereum networks and works only on ethereum network.

Trisha and Rohan et al. [11] proposed a multi-modal method to compare video clips with a full-sized video checking similarities and differences between them. They analyzed if audio and visual clips of a video have similarities or not. They used Siamese network-based architecture to identify deepfake videos. After obtaining modality and emotion (behavioral changes) embedding vectors, they used these obtained vectors to calculate the triplet loss function to lower the similarity between the modalities from the fake videos and increase the similarity between modalities for the real videos. However, they used two deepfake detection datasets, DeepFake-TIMIT Dataset and DFDC, in the paper. If the clip is short they didn't provide any steps to compare the vectors mentioned above. Visual artifacts across frames are not used. Better methods to use audio cues for motion gesture interaction could've been used.

Nhu-Tai Do and Soo-Hyung Kim et al. [3] used DC-GANs to generate images with a resolution of 64x64. Secondly, they used PG-GAN's to generate images with a resolution of 256x256,1024x1024. After generating fake images, they used VGG-Net for image extraction. That is a good method to extract fake images of multiple resolutions however, the environmental cost was higher than other conventional methods. The gen face recognition API they used was older than the likes of Kairos Face recognition API, Lambda Labs API etc. to get higher accuracy than them.

Felix and Xiaofei et al. [13] said that their method detects DeepFakes by monitoring the heartbeat rhythms. Methods like GAN can alter face attributes or pulse signals. Their method on the other hand DeepRhythm which exploits dual-spatial-temporal attention to adjust to constantly changing face and deepfake types. However, this method performs very well on on JPEG compression and Gaussian noise, but its performance is not well on temporal sampling comapring with Xception and MesoNet.

Franklin and Karwoski et al.[16] said that they constructed a standard reference of the performance of spatiotemporal convolutional methods. They used the Celeb-DF dataset which performs very well on state-of-the-art frame-based detection methods. They took advantage of 3D input, tested all the convolutional networks except RCN and pre-trained on the Kinetics dataset which is a large-scale video classification dataset. They analyzed the methods using fixed lengths video clips. They performed random cropping and temporal jittering on all methods. However, the classical frame based method attained an accuracy of only 66.8 percent. This is probably because of the reduced statistical inconsistency between the real and fake images in Celeb-DF versus FaceForensics++. The results Celeb-DF is not same as the FaceForensics++ results as for the cropped Celeb-DF the relative power for each frequency stays within one standard deviation of the mean between fake and real. Inability to distinguish between the real and fake statistics are responsible for the classifier not performing well on this dataset.

Emily and Daniel et al. [3] proposed a method formed on convolutional neural networks (CNNs) and recurrent neural networks (RNNs) that brings out visual and temporal features from faces exdisting in videos to a perfectly detect manipulations.

Daniel Mas Montserrat, Hanxiang Hao et al. [12] introduced a technique formed on convolutional neural networks (CNNs) and recurrent neural networks (RNNs). With CNN and RNN a weighting mechanism integrated with a Gated Recurrent Unit (GRU) is used. This method actually combines all frame information, weights and logits of every face regions to acquire a final estimate. If it fails to read any data it should have it will eventually fail to detect the deep fake accurately. Though this model performs efficiently on both real and manipulated faces in the DFDC dataset.

In the paper [17], they obtained different models using the base network EfficientNetB4 to tackle the problem of face manipulation. They used two concepts; attention layers and siamese training. To make the prediction rate more accurate, they proposed a method by ensembling XceptionNet, EfficientNetB4, EfficientNetB4ST, EfficientNetB4Att, EfficientNetB4AttST. They evaluated their proposed model on the FaceForensics++ dataset and DFDC dataset. In this paper, they trained and tested each of their models separately on both of the datasets they used and later the models were ensembled. They trained EfficientNetB4 and EfficientNetB4Att with traditional end-to-end technique and EfficientNetB4ST and EfficientNetB4AttST is trained using siamese training approach. For training and testing, 32 frames per second from each sequence were chosen. BlazeFace extractor was used to extract faces as BlazeFace extractor is faster than the MTCNN detector. If multiple faces were detected, they kept the face that has the highest confidence score. They trained the models using Adam optimizer. Data augmentation was also performed on extracted input faces to make their model more robust. For the end-to-end approach and siamese training, the feature extractor was trained for 20k iterations.

Run Wang1 and Felix Juefei-Xu et al. [16] proposed a method called DeepTag. It is a simple but effective encoder and decoder model, which can recover the embedded message after many extreme GAN-based DeepFake transformations with high confidence. They have proposed a robust image tagging method that serves for the protection and safety of facial images in social medias to prevent DeepFake. The image tagging method permits us to effortlessly perform DeepFake detection and find the otigin of the real image or video with the help of embedded message. According to the results from the experiments, it is found that Deep Tag attains an exactness of above 95.1 percent on the high-resolution facial images. DeepTag could provide above 80 percent accurate results even when the picture or videos are compressed by 80 percent. But the problem is It cannot detect real time deep fake and mainly it is an image-based process. But They checked the robustness of DPNet and interpretable baselines against temporal specifications. Overall, both techniques fulfil the key frame specification to a large extent. DPNet performs a bit better mainly for the fake traces.

Md. Sohel Rana and Andrew H. Sung et el. [14] explained their method which is called DeepFakeStack by assessing different state-of-art-model which are DL based . According to them,their method resolve the binary classification problem. They labeled 0 for unaltered and 1 for manipulated and measured correctness and categorical log loss. They have obtained a success rate of 99.65 percent and it mainly works by more than one layer of processing. For any reason if any layer fails to analyze the data it can face some trouble.

Irene Amerini and Leonardo Galteri et el [4] proposed a CNN based method which will detect deepfake through optical flow. It is a good method for real time deepfake detection as there is a frame by frame data processing system. Their method is not like other latest and effective works which exploit only one video frames, they proposed to use optical flow fields to utilize as many as possible inter-frame variance. The concept to utilize optical flow field variance as a indication to differentiate between deepfake videos and real videos were presented and examined. But it will be very difficult to process data when two frames will be blended with each other, It can lose its track. However, this process required a heavy machine to continue its process.

Chapter 3

Proposed Model

Our dataset contains video data containing real and manipulated faces. To classify which videos are fake and which are real we had to extract frames from the videos.Then we detected which of the frames contained faces and extracted them with BlazeFace. We stored frames extracted from a particular video to a particular folder containing the video name. Then we indexed and created our dataframe. Our dataset had 3 folders, among which Celeb-real and Youtube-real contained real videos and Celeb-synthesis contained fake videos. So we labeled the real videos as 0 and fake videos as 1. The dataset contained a text file containing the videos that should be used for training. Those videos were labeled true for test. Then we trained our data with ResNext model. The trained model was thus used for classifying the images.

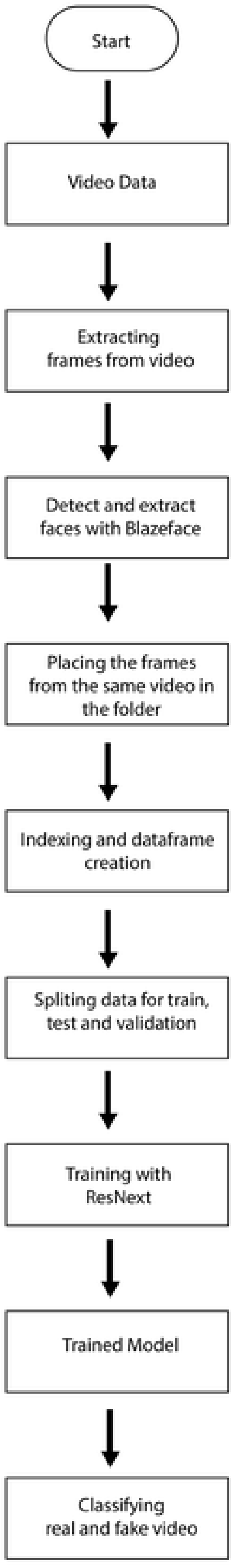

Start
Video Data
Extracting frames from video
Detect and extract faces with Blazeface
Placing the frames from the same video in the folder
Indexing and dataframe creation
Spliting data for train, test and validation
Training with ResNext
Trained Model
Classifying real and fake video

Figure 3.1: Working Mechanism

First we extracted the frames from the video data and stored in such a way that frames extracted from a certain video is stored in a certain folder and the folder is given the same name as the video. Then BlazeFace extractor is used to detect the

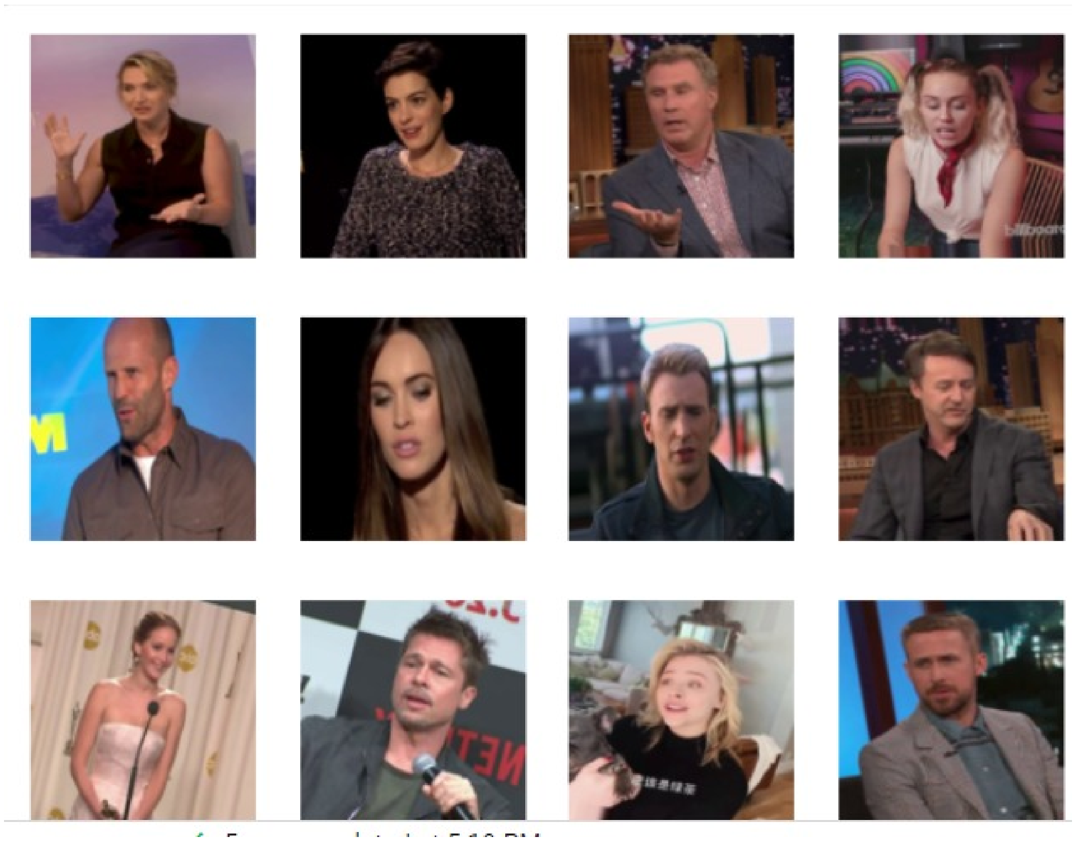


Figure 3.2: Random samples of frames captured from different videos

faces in those frames and extract them. The BlazeFace model creates 6 keypoint of facial coordinates including centers of eye, ear regions, center of mouth and tip of nose tip. which allows to estimate the face rotation and detect faces. Google released the Blazeface algorithm and the algorithm is very fast. The input to the

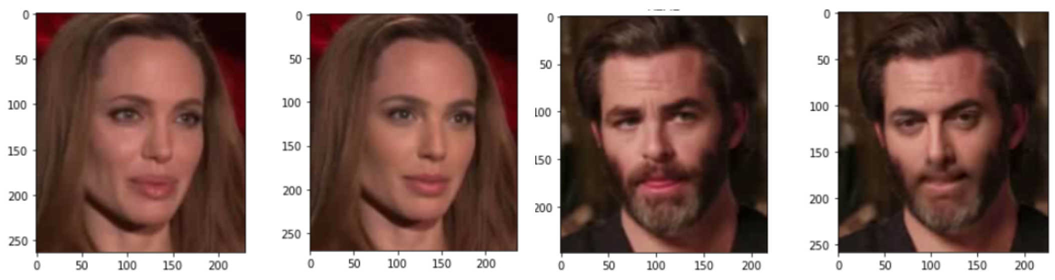
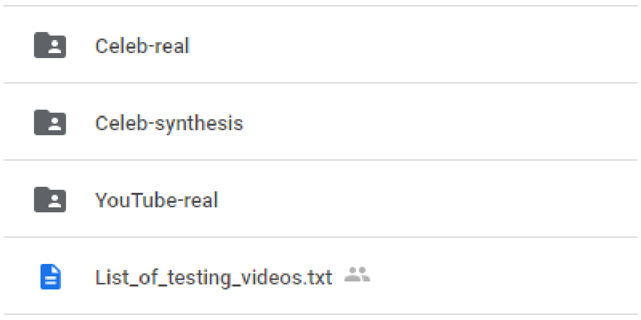


Figure 3.3: Extracted faces from BlazeFace extractor

model is a 128x128 dimensional image. So we split the frames into several tiles which were resized to 128x128. Then Blazeface returns a list of Pytorch tensors for each tile. The detections are then converted back to the original frame size. Since each frame has several tiles, the predictions are then combined and thus the predictions are calculated for each frame.The overlapping predictions are filtered out since one face may be detected in more than one tile. Afterwards, the faces are cropped from the original frames. The the real videos and frames extracted from them are labeled as 0 and the fake ones are labeled as 1. Our dataset contained a text file which contained the name of the videos that should be used for training. so those videos were marked as true for testing in the indexing part. Afterwords we used the ResNext model to train our data. The trained model is then used to identify whether a video is unaltered or manipulated.

3.1 Dataset Description

There are many publicly available dataset for Deepfake detection; DFDC, FaceForensics++, DFD, UADFV, Celeb-DF etc. In this paper, we have used Celeb-DF (v2)[8] dataset to implement our model, which is a freely available dataset. The Celeb-DF (v2) dataset is significantly larger than Celeb-DF dataset (v1). This dataset is the recent and challenging dataset containing deepfake video which is made by using an enhanced DeepFake synthesis method. It is made of 5639 DeepFake videos corresponding to more than 2 million frames derived from YouTube video clips that are available publicly and contains the clips of 59 celebrities of a great deal of variety in ages,genders and ethnicities. The Celeb-DF (v2) dataset comprises real and DeepFake generated videos that have similar visual quality. This large scale dataset solves the problem of other existing datasets' low visual quality and lack of resemblance of DeepFake videos that are widespread in a particular area or at a particular time on the Internet. The datasets that are available and existing have synthesized faces of low-quality, have splicing boundaries that are visible, mismatch of color, have visible parts from the original face, synthesized face orientations are inconsistent and are not convincing and are improbable of having any sufficiently great impact. The in general length of the videos are about 13 seconds and they have the frame rate of 30 frame-per-second. In the unaltered videos, 56.8 percent of the individuals are male, while 43.2 percent are female. 8.5 percent of subjects are of 60 or above, 30.5 percent are between the ages of 50 and 60, 26.6 percent are in their 40s, 28.0 percent are in their 30s, and 6.4 percent are under the age of 30. 5.1 percent of the subjects are Asians, African Americans make up 6.8 percent, and Caucasians make up 88.1 percent. DeepFake videos available in this dataset are made by switching faces of each of the persons present in the clip of the 59 people and the the video format is MPEG4.0. The Celeb-DF (v2) dataset can be downloaded from: https://drive.google.com/file/d/1iLx76wsbi9itnkxSqz9BVBl4ZvnbIazj/view The structure of the Celeb-DF (v2) dataset and the distribution of real and fake videos in the dataset is shown in the Figure 3.1 and 3.2. The false labeled videos are real videos and the true labeled videos are fake videos.



Celeb-real

Celeb-synthesis

YouTube-real

List_of_testing_videos.txt

Figure 3.4: Structure of Celeb-Df dataset

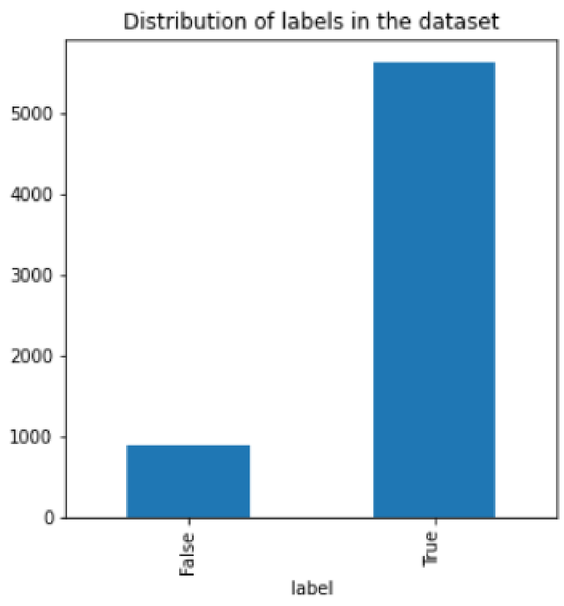


Figure 3.5: Distribution of labels in the dataset

3.2 Data Pre-Processing

To reduce the amount of processing data and time we have discarded those frame information that are not necessary for detecting deep fakes. Besides, we only need the location of the frame that contains the face part, the rest of the frame is unnecessary so we need to extract the faces. For extracting the faces from the frames BlazeFace extractor [5] was used.BlazeFace extractor runs at 200–1000+ FPS speed; which enables it to provide accurate geometry estimation of the facial location, facial expression or feature classification, and segmentation of face region as input to be fed to the models. In case it finds multiple faces, we have selected the face with the highest confidence score. It is also evident to work faster compared to MTCNN as experimented in the paper [19]. The output from the BlazeFace extractor was an image of size 224x224 pixels which was used as input to the model. In case, no face is detected in a frame, the frame is discarded. Moreover, data augmentation was also performed on the inputs during training and validation to make our model more resilient. For this, horizontal flipping, downscaling, brightness contrast, noise addition, hue saturation, noise addition and JPEG compression was used. Albumentation [10] was used as the library for data augmentation. Albumentations library is flexible for image augmentations and transformation is also possible. It is also very fast and easy to use.

3.3 Model Description

3.3.1 Convolutional Neural Network

Convolutional Neural Network (CNN) is a category of neural network used in detecting and recognizing image. Multiple layers of artificial neurons are used in making CNN. CNN are very efficient and less computational powers are also used in detection of image. The working process of CNN is very simple and involves in layers of neuron to conduct its detection.

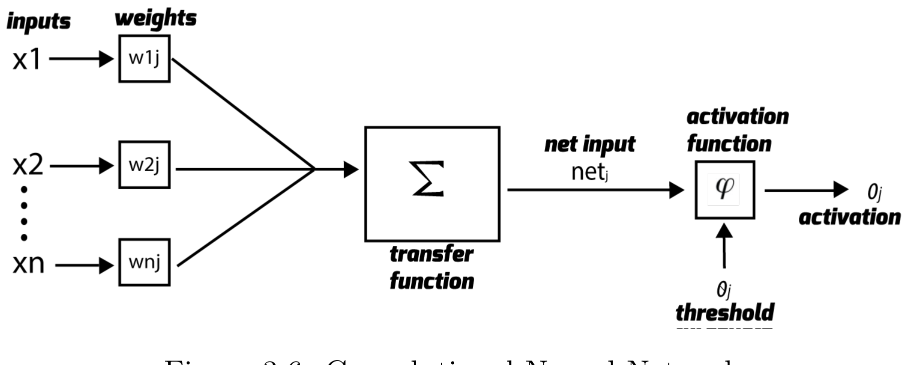


Figure 3.6: Convolutional Neural Network

At first pixel values of images are passed through the weights and when weight is mixed with the input value of the pixel different visual features are got. Only the relevant and noticeable image feature are considered by the network and that are multiplied by the weights. After that it is summed up in the transfer function and a net input is passed to the activation function. Lastly a map is created with all the combined activation that we got from the layers. Depending on the activation map

of the final layer, the inputted image is classified and the system gives us a result on a specified value range. The efficiency of the system depends on the value that we are getting.
The function of multiplying extracted pixel values with weights and lastly summing them up is known as convolution and a CNN contains multiple layers of this kind of convolutional layers. Initial layers of this system detents basic features including vertical, horizonal and diagonal edges. Gradually the system starts detecting more complex details including corners of the images, edges etc. Lastly the bottom most layer does high complex image detection like facial and object recognition etc.

Convolution Layer

One of the major layers and blocks used in CNN where the extracted pixel and the weights are multiplied is convolution layer which is generally a linear operation. In this layer all the pixel in the receptive field is converted into a single value which is used for comparing further on. In case of this layer, 2-dimensional inputs are taken in many cases kernels are used to perform the multiplication. A smaller filter is used which is very small compared to the input data and the dot multiplication is performed in this multiplication process. There is a huge diversity for CNN and in the convolutional layer which learns 32 to 512 filters in parallel and give a diversity in result. Filters, such as line detectors, can be handmade, but convolutional neural networks are unique in that they learn the filters while training in the context of a specific prediction issue

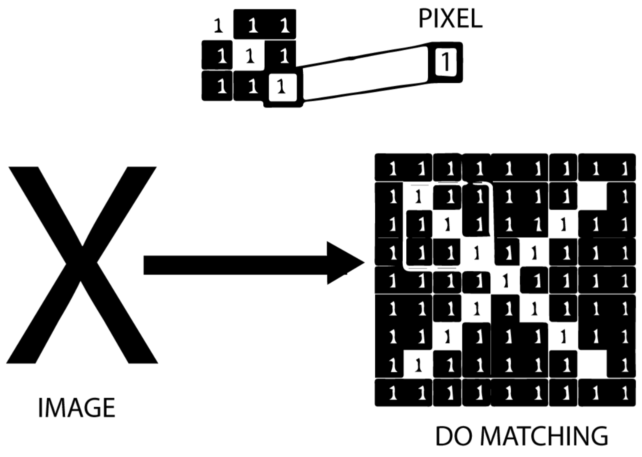


Figure 3.7: Convolution Layer

Many high-level features can be found in an input image. We can extract the high-level features by performing this convolution procedure. The most common type of convolution is 2D convolution, which is sometimes abbreviated as conv2D. A kernel "slides" through the 2D input data in a conv2D layer, performing element-wise multiplication. As a result, the results will be totaled to provide an output pixel. Then kernel will repeat the method for every point it slides over, by converting a 2D matrix consisting of features into a separate 2D matrix consisting of features.

Padding

The amount of extra added pixel to an image while it is being processed by the kernel of a CNN is known as padding. In padding multiple numbers of layers can be added to the image pixel. Accuracy to a greater extent can be achieved by adding padding to a image which makes it more easier for the system to analyze. Padding allows more space for the kernel to cover the image and do the analyzing. By bringing the pixels near the original image's border into the middle of the padded image, this boosts their contribution. As a result, both the information on the image's edges and the information in the middle are maintained.

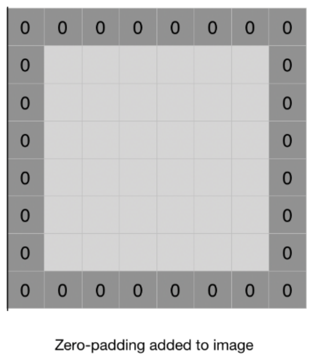


Figure 3.8: Zero-padding added to image

Activation Function

An activation function in a neural network states how the weighted sum of the input from a node or nodes in a layer is converted into an output . An output is delivered in this function based on the input. Sigmoid, tanh, ReLU are examples of activation function. Among them ReLU has many advantages over the other mentioned two. ReLU shows better convergence performance and also it is not a vanishing gradient. Though ReLU has a disadvantage of blowing up activation unlike Sigmoid.

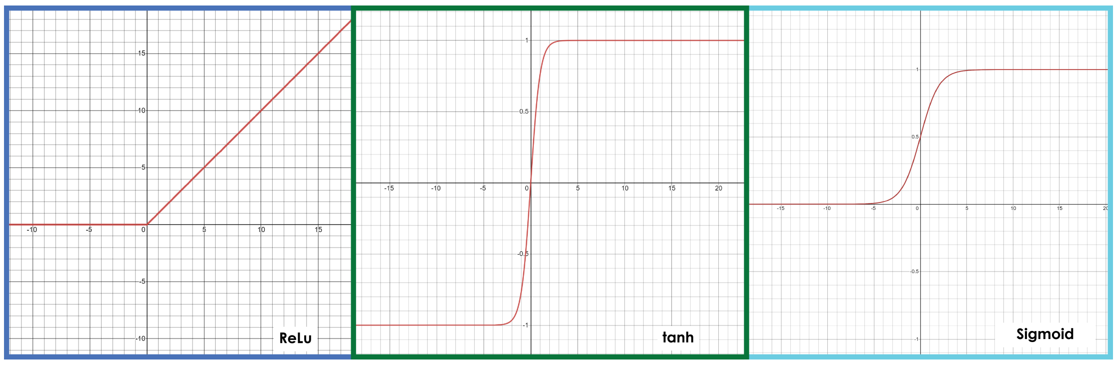


Figure 3.9: Activation Function

Pooling Layer

The job of Pooling Layer in CNN is to decrease the spatial dimension of the representation so that the parameter numbers network is reduced and computation decreases.

The pooling layer operates on each of the feature maps separately, resizes it spatially and combines the results of neuron clusters at one layer into one neuron in the next layer. Another reason to insert this layer is to control overfitting. Usually, for pooling layer, no input padding using zero-padding is done. The pooling layer needs two hyperparameters: spatial extent F and also stride S. It accepts and produces a volume of size W2×H2×D2 where, W2=(W1F)/S+1 H2=(H1F)/S+1 and D2=D1 Max Pooling is the most popular and most used pooling operation. Max Pooling

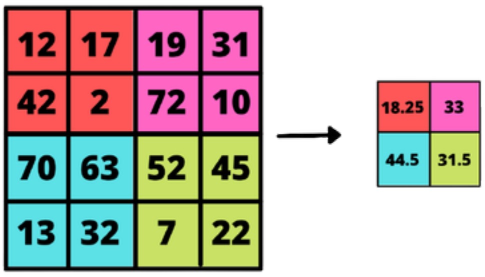


Figure 3.10: Average Pooling

can be one-dimensional, two-dimensional and three-dimensional. It extracts low-level features like edges, points, etc. For each block or pool, the maximum value of the block is calculated. Max pooling ignores the less important elements of a block or pool by picking the maximum value. On the other hand, for Average Pooling, the average value of each pool or block is calculated rather than the max value. It includes all the information in a block or pool and blends them in.

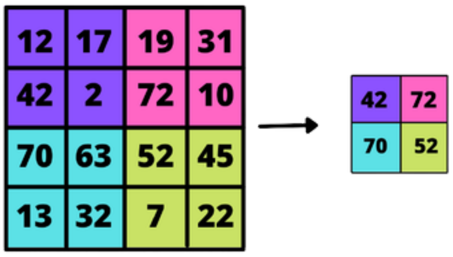


Figure 3.11: Max Pooling

Fully Connected Layer

Fully linked layers are those layers where every input from one layer are attached to every activation unit of the upcoming layer. A model can learn diverse non-linear combinations of high-level characteristics obtained from convolutional layers by adding a fully connected layer. Between two consecutive fully connected layers, an activation function and a dropout layer are employed to bring non linearity and decrease the tendency to over-fit.

3.3.2 EfficientNet

The convolutional neural network architecture where scaling method is used and that method uniformly scales the resolution, width and depth using compound coefficient is known as EfficientNet. In case of EfficientNet novel model scaling methods

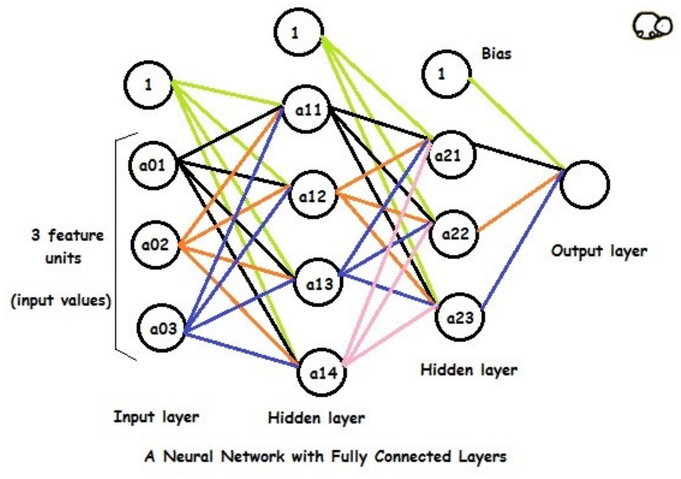


Figure 3.12: Fully Connected Layer

are used. This method are highly efficient in spite of being very simple. EfficientNet uses a very simple architecture were a baseline network is initially used to perform the neural architecture search which is an automating technique for designing neural network. As a result of this structure both the accuracy and the efficiency are increased and optimized to a great extent. EfficientNet tends to give us far better result compared to other models and also the total FLOPs here is minimized to a great extent. Different frameworks of EfficientNet are available and the efficiency and accuracy are different in all of the cases. EfficientNet is substantially smaller than other models that achieve equivalent ImageNet accuracy. For instance, the ResNet50 model, as seen in the Keras application, has a total of 23,534,592 parameters, but it still outperforms the smallest EfficientNet, which has only 5,330,564 parameters. The basic building component of EfficientNet is the mobile inverted bottleneck MBConv, which was first presented in MobileNetV2. By combining direct shortcuts between the bottleneck layers with depthwise separable convolution, which effectively decreases computation by almost a factor, we can connect a significantly less number of channels than expansion layers.

3.3.3 ResNext

ResNeXt architecture is a highly modularized, homogeneous neural network. This architecture is an extended version of deep residual network and is inherited from ResNet, VGG, and Inception. This architecture is very similar to ResNet, only the ResNet blocks are replaced by ResNeXt blocks. In this architecture, a new dimension named cardinality is added which means the dimension of the transformations set[2]. Comparing with ResNet-50, it has been found that if cardinality C is increased from 1 to 32 while keeping complexity, the validation error rate of ResNeXt-50 keeps reducing and the training error of 32×4d ResNeXt is also very much lower[2]. Again, comparing with ResNet-101, it has been found that, if complexity is made double, the error reduction rate is very small when going deeper or wider of the blocks. On the contrary, ResNext gives a much better result than ResNet if the cardinality is increased. This architecture is very simple compared to Inception models and very few hyper-parameters are needed to be set manually. It increases accuracy

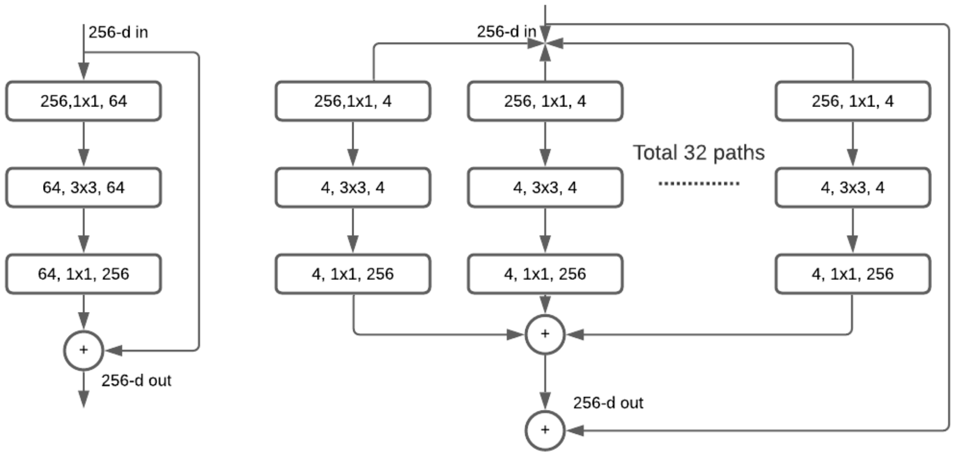


Figure 3.13: Left A ResNet block. Right A ResNeXt block with cardinality = 32

reducing the complexity of the network and the parameter numbers. Same topology blocks are stacked parallelly to create a deep architecture model. Repeating blocks in ResNext aggregates a set of transformations. Input is split into multiple blocks, which means convolution is not performed over the whole input feature map, and later the blocks are merged together which is similar to an Inception module. Before merging the results, convolutional filters are applied individually on the previously split blocks. Hyper-parameters in this architecture is shared between blocks.

Generic equation: F(x) $= \sum_{i=1}^{C} \tau_i(x)$

3.4 Attention mechanism

In different tasks, attention mechanisms are now used an significant feature of appealing sequence modeling and transduction models, which allows for the modeling of dependencies regardless of their length of space in the input or output sequences. The concept of attention mechanisms is a game-changing concept that is revolutionizing the way we apply deep learning. One of the most important achievements in Deep Learning research in the previous decade is the attention mechanism. It has produced a slew of recent natural language processing advancements (NLP). The LSTM encoder processes the full input sentence and encodes it into a context vector, which is the LSTM/final RNN's hidden state and the LSTM or RNN units in the decoder output the words in a sentence one by one Here in the above diagram

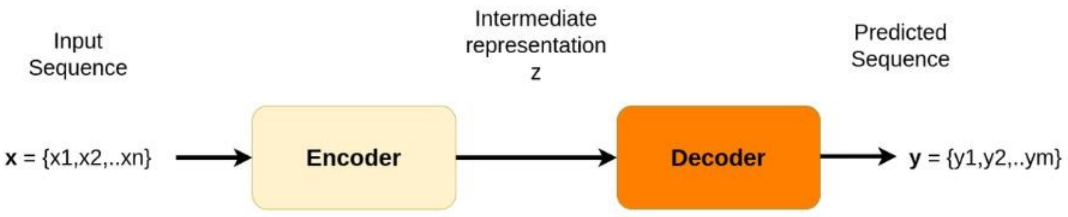


Figure 3.14: How attention mechanism works

the RNNs encoder and decoder function is shown, where it has many limitations and drawbacks. The intermediate representation z is unable to encode all of the input timesteps. The bottleneck problem is what it's called. To solve this limitation, attention mechanism are implemented. As a result of this implementation the

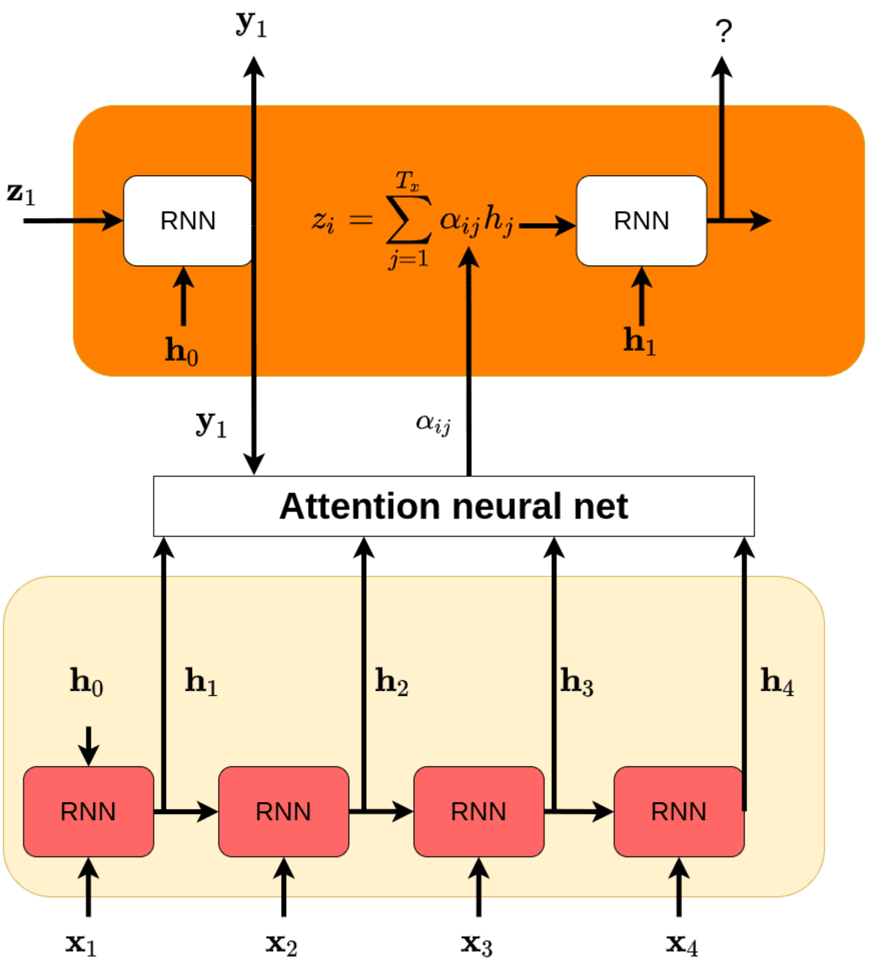


Figure 3.15: Attention mechanism

outcome becomes very much efficient and the problem of bottleneck is solved. The newly formed equation for this is:

Which is far more efficient and error-free than the previous one. $z_i = \sum_{j=1}^{T} a_i j h_j$ Apart from addressing the bottleneck problem, there are numerous reasons for attention to be effective. Because they provide direct connections between the encoder states and the decoder, it frequently eliminates the vanishing gradient problem. Besides, we can acquire insights into the model's behavior as well as identify its limitations by looking at the distribution of attention weights which proves the explainability power of this mechanism.[1]

3.5 Siamese Training

Siamese network is used to compare between two inputs they are the same or different using the Similarity score. Siamese network produce good results even with a small amount of training data. In a Siamese network, two or more subnetworks are passed as inputs. These subnetworks are identical, which means they include the similar configuration with the similar architecture, weights and parameters. The inputs we want to compare, are passed through one of two identical subnetworks that share weights. Then, features are extracted from the input.
Then, the output feature vectors from each subnetwork are combined using subtraction and differencing layer is built to calculate the Euclidian distance. A sigmoid operation is used to convert the value of Euclidian distance to a probability having values 0 and 1 which indicates if the inputs are similar or different.
In the training of Siamese networks, mainly Triplet loss and Contrastive Loss functions are used. Triplet Loss requires three inputs instead of pairs. The triplet consists of an anchor, a positive sample, and a negative sample. In the Triplet loss, the distance between the anchor and negative sample encoding is maximized and the distance between the anchor and positive sample encoding is minimized.

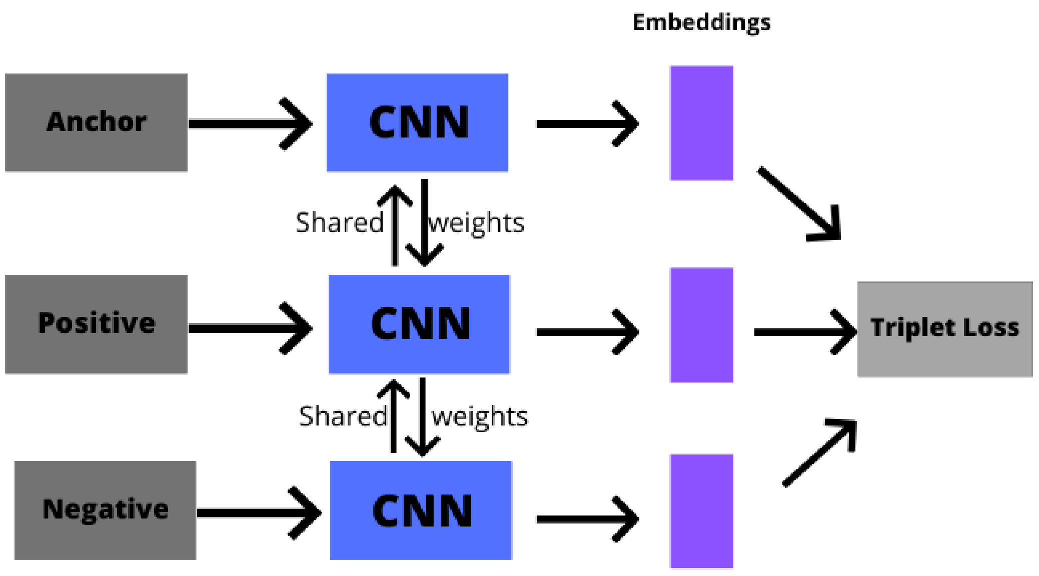


Figure 3.16: Triplet Loss

Triplet Loss Function,
L= max(d(a,p)- d(a,n)+margin,0) Here, d(a,p) = embedding length of space between the Anchor and the positive selection
d(a,n) = embedding length of space between the Anchor and the negative selection
Contrastive Loss Function is helpful when there's not much training data and all the classes are not known at training time. It needs a pair of positive and negative training data. It is distance-based loss and its objective is that two similar points have a small Euclidean distance and two dissimilar points have a greater distance.

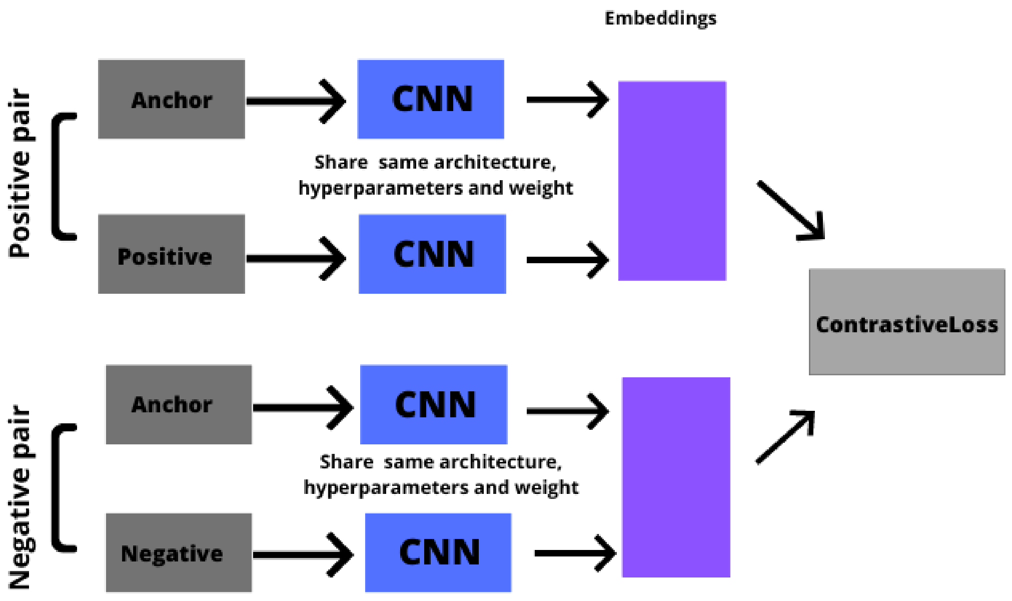

Embeddings
Positive pair
Anchor
CNN
Share same architecture, hyperparameters and weight
Positive
CNN
ContrastiveLoss
Negative pair
Anchor
CNN
Share same architecture, hyperparameters and weight
Negative
CNN

Figure 3.17: Contrastive Loss

Contrastive Loss: $(1-Y)\frac{1}{2}(D_w)^2 + (Y)\frac{1}{2}max(0, m - D_w)^2$
Here, Dw= Euclidean distance Y= 0, if inputs are from the same class Y=1, if inputs are from different class m= margin, which is always greater than 0

Chapter 4

Implementation and Result Analysis

4.1 Implementing the model

Implementation for ResNext

We applied the ResNeXt-50 (32×4d) model for video classification. At first the input images are shaped to 224 by 224.We have used 64 filters, 2 strides and 7x7 convolution in the first convolution layer. In the second convolution layer, we used max pooling with 3x3 pool size and 2 strides.Here the cardinality is 32 which represent 32 groups of group convolution. The grouped convolutional layer conducts convolutions of 32 groups where there are 4 dimensional input and output channels. These are then concatenated as the outputs of the layer. The network has 4 bottleneck layers and each layer has cardinality 32.

Implementation for EfficientNetB4

From the previous works that we studied we found out that efficientNet is considered to be the state of the art network having most accuracy in image classification. In paper [17], they have used the efficientNet B4 model with 2 training strategies-siamese and attention and the datasets used by them are FF++ and DFDC. Being inspired by their paper we wanted to test and tried to increase the performance of their models by bringing slight modification to the structure of models and training parameters on our dataset Celeb-DF(v2). For this, instead of adding the attention layer to the 3rd MBConv block we added the attention layer to the 4th MBConv block having size 14x14x112. Also, we increased the maximum iteration number to 30k. The network is fed a colored image of face extracted from the video and gives a feature vector as output which contains 1792 images. The features from the images are passed to the classifier; which gives the score related to the face as output. To implement the attention layer, the output of 4th MBConv block having size 14x14x112 is selected. These features are then processed with kernel size 1 and the activation function Sigmoid is used. After that the attention map is multiplied with the feature maps. For the siamese training strategy, triplet margin loss is used and a classification layer is used at the top of the network with the same training procedure as before.

4.2 Training the model

We have used adam optimizer for optimization because we wanted to update the weights at the same time our model is training. In order to handle overfitting we have used early stopping with patience of 4.We have also used initial learning rate as 10^{-5}. In order to make our model more flexible we have used data augmentation so that our model can classify images in any form or state. For this we used compression, brightness, contrast, addition of noise, horizontal and vertical flipping, downsampling, hue and saturation. To achieve better performance and efficiency we have used the Albumentations. [10] library. The input to the network is a sample of images containing the faces.The network returns a output score y. For updating the weights we have used the LogLoss function

$$L_L = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(S(\hat{y}_i)) + (1 - y_i)\ (1 - S(\hat{y}_i))] \tag{4.1}$$

During training, the model learn parameters, W and b such that the predicted output , which is $\hat{y}_i$ is closer to the actual output, y_i.
It is conducted by updating the W and b constantly through backpropagation. To update the weights, the model has to know how well or poor it's performance is going.
Log loss is an efffective way of measuring the performance. The more the difference is between predicted and actual output, the larger the log loss is.
(log) Loss function measures discrepency between predicted output $\hat{y}_i$ and actual output. In the equation y_i denotes the score of ith face which is a value between 0 and 1. The sigmoid function is S and N is used as total number of faces. The faces that are from real frames have the label 0 and faces from fake frames have the label 1.

4.3 Result Analysis

The tables below show the results obtained from different models for both frames list and video list

Performance Evaluation of ResNext

We trained our data using ResNext model for a maximum of 30k iterations. The tables below show the performance of the model for both test and validation on both frames basis and video list. While the frames are considered, the model detected 3535 real and 6800 fake frames on the test data. And accuracy acquired for the frames was 72% for test data.

To determine if a video is real or manipulated, the average of the pre-sigmoid scores from the frames was taken and then we computed the sigmoid on the mean score of the whole video. Thus the model detected 113 unaltered videos and 715 manipulated videos on the validation data and 177 real videos and 340 manipulated videos on the test data. The accuracy received in this case was 77% for the test data.

net	**split**	**real**	**fake**	**loss**	**acc**	**accbal**	**rocauc**
Resnext	val	2258	14300	0.687168	0.659319	0.730412	0.822789
Resnext	test	3535	6800	0.623719	0.725980	0.758419	0.834087

Table 4.1: Result per frame list for ResNext

net	split	real	fake	loss	acc	accbal	rocauc
Resnext	val	113	715	0.603565	0.68599	0.766025	0.859917
Resnext	test	177	340	0.563863	0.77176	0.802094	0.864174

Table 4.2: Result per video list for ResNext

Performance Evaluation of EfficientNet models

We trained our data using EfficientNetB4 model for a maximum of 30k iterations. we also used the attention mechanism and siamese training strategy used in the paper [17]. But instead of adding the attention layer to the 3rd MBConv block we added the attention layer to the 4th MBConv block having size 14x14x112 and increased the maximum iteration to 30k. The tables below show the performance of the model for both test and validation on both frames basis and video list. While the frames are considered, the model detected 3535 real and 6800 fake frames on the test data. And accuracy acquired for the frames was 96.67% for test data.While trained with siamese training strategy, the model acquired 97.23 % accuracy, and while used the attention mechanism, it acquired 97.35 % accuracy on test data.

net	split	real	fake	loss	acc	accbal	rocauc
EfficientNetB4	test	3535	6800	0.0941	0.9672	0.9644	0.9945
EfficientNetB4	val	2258	14300	0.1377	0.9458	0.9449	0.9889
EfficientNetB4-Att-ST	test	3535	6800	0.0840	0.9723	0.9697	0.9945
EfficientNetB4-Att-ST	val	2258	14300	0.1061	0.9523	0.9517	0.9914
EfficientNetB4-Att	test	3535	6800	0.0905	0.9735	0.9688	0.9954
EfficientNetB4-Att	val	2258	14300	0.1336	0.9551	0.9516	0.9901

Table 4.3: Result per frame from the EfficientNetB4 models

For the video classification, similar strategy was followed as the ResNext model. Thus the EfficientNetB4 model detected 113 unaletred videos and 715 manipulated videos on the validation data and 177 real videos and 340 manipulated videos on the test data. The accuracy received for the EfficientNet B4 model on test data was 99 % while when trained with siamese strategy, it acquired 98.83 % accuracy and when attention mechanism was used it achieved 99.41 % accuracy.

net	split	real	fake	loss	acc	accbal	rocauc
EfficientNetB4	test	177	340	0.0262	0.9941	0.9928	0.9993
EfficientNetB4	val	113	715	0.0550	0.9637	0.9641	0.9982
EfficientNetB4-Att-ST	test	177	340	0.0385	0.9883	0.9871	0.9988
EfficientNetB4-Att-ST	val	113	715	0.0494	0.9770	0.9792	0.9989
EfficientNetB4-Att	test	177	340	0.0297	0.9941	0.9915	0.9995
EfficientNetB4-Att	val	113	715	0.0512	0.9649	0.9685	0.9983

Table 4.4: Result per video from the EfficientNetB4 models

After ensembling the EfficientNetB4, EfficientNetB4 model trained with siamese strategy and EfficientNetB4 model trained with attention mechanism we can see that we get an AUC score of 99.67%. AUC score is used to measure the potentiality of the classifier to distinguish between different classes. Higher AUC score means better performance a the model on differentiating between positive and the negative classes.

net	loss	auc
(EfficientNetB4)	0.09418	0.9945
(EfficientNetB4-Att-ST)	0.08403	0.9945
(EfficientNetB4-Att)	0.09050	0.9954
(EfficientNetB4, EfficientNetB4-Att-ST)	0.07454	0.9960
(EfficientNetB4, EfficientNetB4-Att)	0.07638	0.9965
(EfficientNetB4-Att-ST, EfficientNetB4-Att)	0.07677	0.9964
(EfficientNetB4, EfficientNetB4-Att-ST, EfficientNetB4-Att)	0.07157	0.9967

Table 4.5: Ensemble

The face images having score closer to zero are real frames and the face images having score closer to 1 are manipulated images. To classify a video we had to calculate the average of the pre sigmoid scores and then calculate the sigmoid to find the mean score of the video.

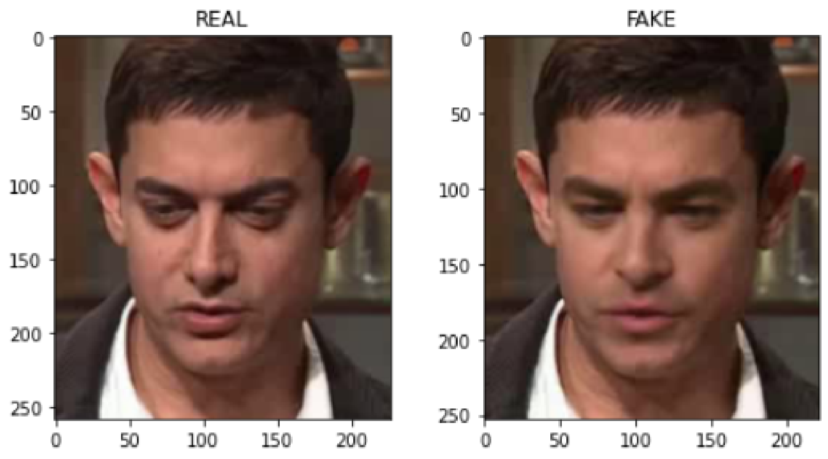


Figure 4.1: Classification of Real and fake frames

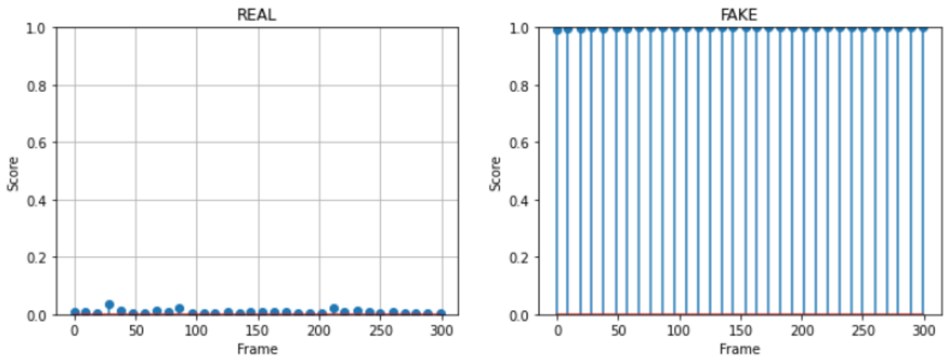


Figure 4.2: Face Score for real frames vs. fake frames

4.3.1 Observation and Comparison of Results

We can see that we got 77% accuracy on the video list while we used ResNext model for training.Based on some related works conducted in this topic, we learnt that ensembling a few models work better. In paper [17] they have trained the XceptionNet and ensembled EfficientNetB4 models with attention and siamese training for which they obtained 94% AUC score for FF++ dataset and 87% AUC score for the DFDC dataset. But instead of adding the attention layer to the 3rd MBConv block we added the attention layer to the 4th MBConv block having size 14x14x112. Also, we increased the maximum iteration number to 30k; which resulted in an accuracy of 96 % and AUC score of 99% based on frames result when trained without attention and 97% accuracy on test data when trained with attention. In the paper,[18] they used the Celeb-DF (v2) dataset also but they created different classifiers for different parts of the face to detect any manipulations and they got an AUC score of 66 % on nose, 65% on mouth, 63% on eyes and 64% on chin which was 63% on average for this dataset. Thus from the results obtained, it is visible that the number of fake and real frames detected and the number of fake and real videos detected from all the models are same whereas, the accuracy achieved by the EfficientNet model is more.

Chapter 5

Conclusion and Future work

Focusing on the fact that deepfake images and videos are created to spread chaos and rumors by misleading people, it has become necessary to detect and ban them from the social media platforms. In spite of all the existing algorithms prevailing to detect deepfake, manipulated videos are still circulating over the internet. Moreover, it can be conjectured that in near future with the advancement of deepfake tools it will be possible to create such realistic images or videos that will be able to fool the detection algorithms. In this paper we have worked with the Celeb-Df (v2) dataset containing real and manipulated videos of the celebrities. At first we preprocessed the data;for which we extracted frames from the videos and then extract the faces from them.We classified the real and fake images from the dataset by training the model ResNext by feeding the faces extracted from the video frames. For this, we had to study how different CNN models work and find the best model for training. Furthermore, we not only used the EfficientNetB4 but also modified the EfficientNetB4 model with attention layer and also trained the EfficientNetB4 model using Siamese training strategy to test and evaluate the results for Celeb-DF(v2) dataset. We then were able to classify the real and fake images and videos and also visualize how the result changes for different models and training strategies.
To conclude, we would like to state that the algorithms presented in this paper by us is an attempt to find an efficient model and to improve the existing models which is merely based on the research done by us relating to our field. Moreover, we discussed the methods that we used in our dataset for training, experimentation and validation. For future work we wish to combine a few video datasets together to add a variety in the data, ensemble a few other models to check the changes in result, implement transfer learning to save training time and implement majority voting scheme where we will like to consider all the frames from the video to make our model work better.

Bibliography

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[2] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1492–1500.

[3] N.-T. Do, I.-S. Na, and S.-H. Kim, "Forensics face detection from gans using convolutional neural network," in *2018 International Symposium on Information Technology Convergence (ISITC 2018), South Korea*, 2018.

[4] I. Amerini, L. Galteri, R. Caldelli, and A. Del Bimbo, "Deepfake video detection through optical flow based cnn," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0–0.

[5] V. Bazarevsky, Y. Kartynnik, A. Vakunov, K. Raveendran, and M. Grundmann, "Blazeface: Sub-millisecond neural face detection on mobile gpus," *arXiv preprint arXiv:1907.05047*, 2019.

[6] J. Damiani, *A voice deepfake was used to scam a ceo out of 243,000, forbes*, 2019.

[7] H. R. Hasan and K. Salah, "Combating deepfake videos using blockchain and smart contracts," *Ieee Access*, vol. 7, pp. 41 596–41 606, 2019.

[8] Y. Li, X. Yang, P. Sun, H. Qi, and S. Lyu, "Celeb-df (v2): A new dataset for deepfake forensics," *arXiv preprint arXiv:1909.12962*, 2019.

[9] A. Alattar, R. Sharma, and J. Scriven, "A system for mitigating the problem of deepfake news videos using watermarking," *Electronic Imaging*, vol. 2020, no. 4, pp. 117–1, 2020.

[10] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin, "Albumentations: Fast and flexible image augmentations," *Information*, vol. 11, no. 2, p. 125, 2020.

[11] T. Mittal, U. Bhattacharya, R. Chandra, A. Bera, and D. Manocha, "Emotions don't lie: An audio-visual deepfake detection method using affective cues," in *Proceedings of the 28th ACM international conference on multimedia*, 2020, pp. 2823–2832.

[12] D. M. Montserrat, H. Hao, S. K. Yarlagadda, S. Baireddy, R. Shao, J. Horváth, E. Bartusiak, J. Yang, D. Guera, F. Zhu, *et al.*, "Deepfakes detection with automatic face weighting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 668–669.

[13] H. Qi, Q. Guo, F. Juefei-Xu, X. Xie, L. Ma, W. Feng, Y. Liu, and J. Zhao, "Deeprhythm: Exposing deepfakes with attentional visual heartbeat rhythms," in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 4318–4327.

[14] M. S. Rana and A. H. Sung, "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)*, IEEE, 2020, pp. 70–75.

[15] A. E. Venema and Z. J. Geradts, "Digital forensics deepfakes and the legal process," *TheSciTechLawyer*, vol. 16, no. 4, pp. 14–23, 2020.

[16] R. Wang, F. Juefei-Xu, Q. Guo, Y. Huang, L. Ma, Y. Liu, and L. Wang, "Deeptag: Robust image tagging for deepfake provenance," *arXiv preprint arXiv:2009.09869*, 2020.

[17] N. Bonettini, E. D. Cannas, S. Mandelli, L. Bondi, P. Bestagini, and S. Tubaro, "Video face manipulation detection through ensemble of cnns," in *2020 25th International Conference on Pattern Recognition (ICPR)*, IEEE, 2021, pp. 5012–5019.

[18] S. Schwarcz and R. Chellappa, "Finding facial forgery artifacts with parts-based detectors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 933–942.

[19] L. Trinh, M. Tsang, S. Rambhatla, and Y. Liu, "Interpretable and trustworthy deepfake detection via dynamic prototypes," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 1973–1983.