VISVESVARAYA TECHNOLOGICAL UNIVERSITY

JNANA SANGAMA, BELAGAVI – 590 018, KARNATAKA

A Project Report
on

SYSTEM FOR DETECTING DEEPFAKE IN VIDEOS

Submitted in partial fulfillment of the requirements for the VIII Semester of degree of
Bachelor of Engineering in Information Science and Engineering *of*
VisvesvarayaTechnological University, Belagavi

by

Annapurna R Shanbhag	**1RN18IS019**	**Harshitha G S**	**1RN18IS051**
Vani Shiva Bhat	**1RN18IS116**	**Vardhini B H**	**1RN18IS117**

Under the Guidance of
Mrs.Kusuma S
Associate Professor
Department of ISE

Department of Information Science and Engineering

RNS INSTITUTE OF TECHNOLOGY

Dr. Vishnuvaradhan Road, Rajarajeshwari Nagar Post,
Channasandra, Bengaluru – 560 098

2021-2022

RNS INSTITUTE OF TECHNOLOGY

Dr. Vishnuvaradhan Road, Rajarajeshwari Nagar Post
Channasandra, Bengaluru – 560 098

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the project work entitled ***System for Detecting Deepfake in Videos*** has been successfully completed by **Annapurna R Shanbhag (1RN18IS019), Harshitha G S (1RN18IS051), Vani Shiva Bhat (1RN18IS116), and Vardhini B H (1RN18IS117),** bonafide students of **RNS Institute of Technology, Bengaluru** in partial fulfillment of the requirements for the award of degree **Bachelor of Engineering** in **Information Science and Engineering** of **Visvesvaraya Technological University, Belagavi** during academic year **2021-2022**. The project report has been approved as it satisfies the academic requirements in respect of project work for the said degree.

Mrs. Kusuma S
Project Guide

Dr. Prakasha S / Mrs. Kusuma S
Project Coordinator

Dr. Suresh L
Professor and HOD

Dr. M K Venkatesha
Principal

External Viva

Name of the Examiners	Signature with Date
1. _____	1. _____
2. _____	2. _____

DECLARATION

We, **ANNAPURNA R SHANBHAG (1RN18IS019), HARSHITHA G S (1RN18IS051), VANI SHIVA BHAT (1RN18IS116), and VARDHINI B H (1RN18IS117)** students of VIII Semester B.E. in Information Science and Engineering, RNS Institute of Technology hereby declare that the Project entitled ***System for Detecting Deepfake in Videos*** has been carried out by us and submitted in partial fulfillment of the requirements for the *VIII Semester of degree of* ***Bachelor of Engineering in Information Science and Engineering*** *of Visvesvaraya Technological University, Belagavi* during academic year 2021-2022.

Place: Bengaluru

Date:

ANNAPURNA R SHANBHAG **(1RN18IS019)**

HARSHITHA G S **(1RN18IS051)**

VANI SHIVA BHAT **(1RN18IS116)**

VARDHINI B H **(1RN18IS117)**

ABSTRACT

With the rapid progress in technology over the past few years, techniques that generate and manipulate multimedia content can now guarantee a very advanced level of realism. The boundary between real and synthetic media has become very thin. On the one hand, this opens the door to a series of exciting applications in different fields such as creative arts, advertising, film production, video games. The increasing sophistication of smartphone cameras and the availability of good internet connection all over the world has increased the ever-growing reach of social media and media sharing portals have made the creation and transmission of digital videos easier than ever before.

On the other hand, it poses enormous security threats. Software packages freely available on the web allow any individual, without special skills, to create very realistic fake images and videos. Such realistic looking videos and images are called as “Deepfakes”. Such videos can prove to be disastrous. They can be used to manipulate public opinion during elections, commit fraud, discredit or blackmail people. Potential abuses are limited only by human imagination. Therefore, there is an urgent need for automated tools capable of detecting false multimedia content and avoiding the spread of dangerous false information.

ACKNOWLEDGMENT

The fulfillment and rapture that go with the fruitful finishing of any assignment would be inadequate without specifying the people who made it conceivable, whose steady direction and support delegated the endeavors with success.

We would like to profoundly thank **Management** of **RNS Institute of Technology** for providing such a healthy environment to carry out this Project work.

We would like to express our thanks to our Principal **Dr. M K Venkatesha** for his support and inspiration towards the attainment of knowledge.

We wish to place on record our words of gratitude to **Dr. Suresh L**, Professor and Headof the Department, Information Science and Engineering, for being the enzyme and master mind behind our Project work.

We Would like to thank our project Coordinators **Dr. Prakasha S**, Associate Professor, and **Mrs. Kusuma S**, Assistant Professor, Department of Information Science and Engineering who is also our guide of the project, for their valuable guidance, constructive comments and continuous encouragement throughout theProject work.

We would like to thank all other teaching and non-teaching staff of Information Science& Engineering who have directly or indirectly helped me to carry out the project work.

And lastly, we would hereby acknowledge and thank our parents who have been a source of inspiration and constant support in carrying out this Project work.

ANNAPURNA R SHANBHAG (1RN18IS019)

HARSHITHA G S (1RN18IS051)

VANI SHIVA BHAT (1RN18IS116)

VARDHINI B H (1RN18IS117)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AI	Artificial Intelligence
CNN	Convolution Neural Networks
DF	Deep Fake
DFDC	Deep Fake Detection Challenge
DFT	Discrete Fourier Transform
DWT	Discrete Wavelet Transform
FFT	Fast Fourier Transform
GAN	Generative Adversarial Networks
LSTM	Long Short -Term Memory
RNN	Recurrent Neural Network
SVM	Support Vector Machines

Chapter 1

INTRODUCTION

In recent times, AI-synthesized face swapping videos referred to as deepfake have become an emerging problem. False video is becoming more and more difficult to distinguish, which brings a series of challenges to social security. Deepfake is a combination of fake and deep-learning technology. Deep learning is the function of artificial intelligence that can be used to create and detect deepfakes.

Photo manipulation was developed in the 19th century and soon applied to motion Pictures. The first known attempt at trying to swap someone's face, circa 1865, can be found in one of the iconic portraits of U.S. President Abraham Lincoln. The lithography, as seen in Fig 1.1, mixes Lincoln's head with the body of Southern politician John Calhoun. After Lincoln's assassination, demand for lithographs of him was so great that engravings of his head on other bodies appeared almost overnight. Technology steadily improved during the 20th century, and more quickly with digital videos. Deepfake technology has been developed by researchers at academic institutions beginning in the 1990s, and later by amateurs in online communities. More recently the methods have been adopted by industry.

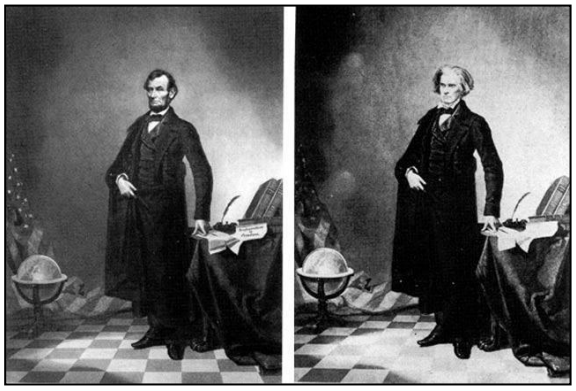

Fig. 1.1 swap of U.S. President Lincoln's head with politician John Calhoun's body were produced in mid-19th century.

Anyone can generate realistic-looking fake faces by utilizing state-of-the-art face manipulation tools such as DeepFake or Face2Face. Fig.1.2 shows the examples of genuine and fake faces, which are almost indistinguishable with the bare eye (The second and fourth rows are fake faces manipulated by DeepFake and Face2Face, respectively. Faces on the first and third rows are genuine). DeepFake replaces a person's face in the original media with the target person's likeness while maintaining the original facial expression.

Face2Face transfers the expression of a source media to a target media while maintaining the original facial identity. Both tools use deep learning and computer vision techniques to generate an image containing the desired manipulation result. The manipulated face is pasted and rendered onto the original image. Then, seamless blending is applied, which adjusts image characteristics such as color, contrast, and brightness to make a realistic manipulation result.

Fig. 1.2: Examples of genuine and fake facial images.

As the need for fake face detection arises, facial image forensics research has been actively conducted in recent years. Most face forensics approaches leverage deep learning, especially convolutional neural networks (CNNs) originally architected for the object recognition task such as AlexNet, VGG19, ResNet and Xception.

These models have a number of stacked layers for hierarchically extracting content features from an image: it first extract low-level content features (e.g., edge, mesh patterns, text) and mixes them to detect and classify object specific features (e.g., dog faces, birds' legs) . So, for fake detection, the whole layers except the classification part are borrowed for transfer learning. Among the object detection CNNs, Xception showed the most powerful performance.

Chapter 2

LITERATURE REVIEW

In **[1],** a temporal-aware pipeline is proposed to automatically detect deepfake videos. It uses aconvolutional neural network (CNN) to extract frame-level features. These features are then used to train a recurrent neural network (RNN) that learns to classify if a video has been subject to manipulation or not.

The main contributions of this work are summarized as follows:

A two-stage analysis composed of a CNN to extract features at the frame level followedby a temporally-aware RNN network to capture temporal inconsistencies between frames introduced by the face-swapping process is proposed.

- A collection of 600 videos to evaluate the proposed method is used, with half of the videos being deepfakes collected from multiple video hosting websites.
- The effectiveness of the described approach is shown experimentally, which allows useto detect if a suspect video is a deepfake manipulation with 94% more accuracy than a random detector baseline in a balanced setting.

In **[2],** The following methodology is used:

Frame Extraction: Most image detection methods cannot be used for videos alone because of the strong degradation of the frame data after video compression. The beneficial way to analyze this media was to implement an extraction of the videos into frames. This is a form of preprocessing before implementing the techniques of detecting DeepFakes by using DWT and CNN. This method will insert frames in the dataset by doing this extraction. The video lengths vary in range so, we took the first

Apply DWT as filtering: DWT, decomposes an image into four parts. DWT when using Python library PyWavelets outputs images as Approximation, Horizontal Detail, Vertical Detail, and Diagonal Detail sub-band images called low-low (LL), low-high (LH) high-low (HL), and high ℎigh (HH). e specific output that will be used in this willbe Vertical Detail also known as high-low, HL. This image decomposition can be understood as an underlying high frequency filter for the frame. The newly filtered frames will be the feeding dataset for the CNN in the next steps.

Implement CNN: For classification a convolutional neural network is used. This will better help in the indication of an anomaly or flaw within the Deep Fake. Before feeding into the CNN model, the dataset of frames is split into 90% training frames and 10% testing frames. This step is key to training a model. The training process is set to around 50 epochs for a better accuracy. The videos will not train at the same rate; the higher the resolution the longer to train which has been realized.

In **[3],** AI-synthesized face-swapping videos, commonly known as DeepFakes, is an emerging problem threatening the trustworthiness of online information. Current DeepFake datasets suffer from low visual quality and do not resemble Deep- Fake videos circulated on the Internet. DeepFake video dataset, Celeb- DF, which contains 5, 639 high quality DeepFake videos of celebrities generated using improved synthesis process. This paper conducts a comprehensive evaluation of DeepFake detection methods and datasets to demonstrate the escalated level of challenges posed by Celeb-DF.

The paper consists of Backgrounds like Deep Fake Video generation: It is a much simpler method based on the work of neural image style transfer that becomes the tool of choice to create DeepFake videos in scale, with several independent open-source implementations DeepFake Detection Methods: DeepFake detection method require training data and need to be evaluated. Current DeepFake Datasets are:

- UADFV - Contains 49 real YouTube and 49 DeepFake Videos.
- DFTIMIT - Includes 640 deepFake videos
- FF-DF -Subset of Deepfake Videos consists of 1000 real YouTube videos.
- DFD - The Google/Jigsaw DeepFake detection dataset consists of 3068 Deepfake videos and 363 original videos.
- DFDC - has 4, 113 DeepFake videos created based on 1, 131 original videos. Celeb-DF dataset consists of:

Basic Information about the datasets.

Synthesis Method generated using improved DeepFake synthesis algorithm. It evaluates thresolution of synthesized face, color mismatch, temporal flickering.

The Celeb-DF dataset reduces the gap in visual quality of DeepFake datasets and the actual DeepFake videos circulated online. Based on the Celeb-DF dataset, we can perform a comprehensive performance evaluation of current DeepFake detection methods, and show that there is still much room for improvement.

In **[4],** Information has been gathered about previous attempts, analyzed methods used by different researches and considered their actuality right now. Basing on results of the discovery was designed strategy to expose Deep Fake videos that combines previous detection methods with super-resolution algorithms. Results of the research were compared with expected, so recommendations and possible way of continuing developments were given.

The most successful methods are based on:

- eye blinking
- mismatched color profiles and
- face warping artifacts

On the other side, if issue is considered as more complex, require other correlations besides visual ones. Due to this, we have attempts that are directly classifying faked content using actual algorithms of machine and deep learning. Method was designed to expose Deep Fakes basing on mismatch between directions of different face regions. It gave good accuracy on more accurate fakes, but still have problems with low resolution video fakes.

The method used is inherently an alliance of 2 methods: Exposing Deep Fakes using inconsistent head poses and detecting Deep Fake pictures using CNN Resnet50 model. Pipeline of the system consists of 4 blocks: Dataset preprocessing, ResNet Classification, Inconsistent Head Pose estimator and Arbitrage (decision maker)

Step 1: At first, original videos fed to Face Recognition module. There they are separated for frames; face location is estimated on each frame and 68 face landmarks are estimated for each frame.

Step 2: Then all preprocessed frames are going through 2 classifications. In addition to ResNet, the super-resolution algorithm was applied to solve this problem. It aims to increase the accuracy of predictions on low-resolution videos: the quality of Deep Fakes often artificially decreased in order to hide artifacts and make pictures closer to the real one.

s**tep 3**: Results from both classifications are transferred to Decision maker that is announcing the final judgement.

The big part in this result played failure with head pose estimation and further classification. So, this experiment can be considered as consisting only of 2 elements: Resnet classification and preprocessing using FSRCNN model.

In **[5]**, a new method is proposed with clustering-based embedding regularization for deepfake detection. Open-source algorithms are used to generate videos which can simulate distinctive artifacts in the deepfake videos. To improve the local smoothness of the representation space, a clustering-based embedding regularization term is integrated into the classification objective, so that the obtained model learns to resist adversarial examples. The method is evaluated on three latest deepfake datasets. Experimental results demonstrate the effectiveness of the method.

The Xception network is trained for classification, using positive samples, negative samples, and generated samples as input samples. Class number is set to 3 during training process, and in the testing process the generated samples are also classified as negative samples to improve the classification effect. At the same time, a regularization loss is added during the training process to ensure the inter-class distance and intra-class smoothness of the embedded space.

The contributions of the paper are as follows:

- The process of the deepfake videos' generation is simulated using simple image processing operations on an image, and the simulated samples are used as an additional class to train the Xception networks. When testing, the simulated samples are classified as negative samples.
- A clustering-based embedding regularization into adversarial learning is introduced, which further guarantees the inter-class distance and the intra-class smoothness in the embedding space, therefore improves the robustness of the model.
- Experimental results on UADFV, Celeb-DF and Deep Fake Detection datasets demonstrate the effectiveness of the methods in deepfake detection.

In **[6],** Convolutional Neural Network (CNN) is used to extract frame level features, then these features are trained in a recurrent neural network (RNN) which learns to classify if an image has been manipulated or not. While these strategies show promising execution, the GAN can easily learn all these methods and find a countermeasure which is a key problem. The existing approaches are mainly based on CNN (Convolutional Neural Network) whereas proposed approach is based on frequency analysis which reveals different behavior of images at different frequency.

Methods used are as follows:

Frequency Domain Analysis: In frequency domain analysis, the conversion of image from spatial domain to frequency domain is done. For specific applications image enhancement is done after the image filtering. The frequency domain is a space which is characterized by Fourier transform. Image processing is one of the applications of Fourier transform. It is used to indicate how signal energy can be distributed over a range of frequency. Computing the 2D discrete Fourier transform of the image is the principle of frequency domain analysis.

Discrete Fourier Transform: One of the most important tools in digital signal processing is discrete Fourier transform (DFT). A signal's frequency spectrum can be calculated using DFT. The input and output values are discrete samples, which make it convenient for computer manipulation. At each frequency the information about the signal’s amplitude and phase is conveyed by frequency-domain representation of a signal.

Azimuthal Average: The information our method is getting now is in new domain after applying Fourier transform to image and in the same dimensionality. So, as we are working with images, we will still get the output as a 2D data. So, to get a strong 1D representation of the Fast Fourier Transform (FFT) magnitude spectrum our method will apply azimuthal average.

Classifier Algorithms: A classifier algorithm is an important algorithm in machine learning which is used to arrange or classifies information into at least one of a set of classes. Email classifier is one of the basic examples in which classifier algorithm is used to filter them as spam or not spam. At each frequency the information about the signal’s amplitude and phase is conveyed by frequency-domain

Support Vector Machines One of the most powerful and most widely used supervised machine learning algorithms used for data classification is Support Vector Machines (SVMs). Basically, the main goal of SVM is to produce maximum marginal hyperplane in Multidimensional space separating different classes. The error can be minimized by generating the hyperplane in iterative manner.

In **[7],** The use of Deep Learning Concept to detect the Face Manipulation is suggested. Fake detection methods in media forensics have been commonly based on:

- In-camera fingerprints, the analysis of the intrinsic fingerprints introduced by the camera device, both hardware and software, such as the optical lens color filter array and interpolation, and compression, among others.

- Out- camera fingerprints, the analysis of the external fingerprints introduced by editing software, such as copy-paste or copy- move different elements of the image, reduce the frame rate in a video, etc.

This paper provides in depth review of digital manipulation techniques that is applied to facial content. Four manipulations covered in this paper are:

Entire Face Synthesis: Detected Using StyleGAN methods (GAN-Generative Adversarial Networks)

Identify Swap: Two different approaches are usually considered:

- classical computer graphics-based techniques such as FaceSwap,
- novel deep learning techniques known as DeepFakes, e.g., the recent ZAO mobile application. Very realistic videos of this type of manipulation can be seen on Youtube.

Attribute Manipulation: also Known as face editing or face retouching: This manipulation process is usually carried out through GAN such as the StarGAN approach proposed in . One example of this type of manipulation is the popular FaceApp mobile application.

Expression Swap: also known as face reenactment. It works on modifying the facial expression of the person. in this group we focus on the most popular techniques Face2Face and Neural- Textures, which replaces the facial expression of one person in a video with the facial expression of another person.

This Paper more reviews on each facial manipulation group, including manipulation techniques, existing public databases, and key benchmarks for technology evaluation of fake detection methods, including a summary of results from those evaluations.

In **[8]**, the proposed approach is based on exploiting the residual noise which is the difference between original image and its denoised version .The study of residual noise has shown effectiveness in deepfake detection with regards to its distinctive and discriminative features which can be effectively captured by convolutional neural networks with transfer learning Theperformance of our approach is evaluated on two datasets: low resolution video sequences of the FaceForensics++ and high resolution videos from Kaggle Deepfake Detection challenge (DFDC).

The work aims to explore the residual noise of real and fake videos by proposing a classification scheme based on transfer learning using Convolutional Neural Networks. The performance of the proposed method is demonstrated on 2 datasets: FaceForensics++ and

DeepFake DetectionChallenge (DFDC).

Given a video V with frames F having h × w dimension the goal is to detect whether the residual noise obtained from an original video is different from the one obtained from the fake videos. To extract the residual noise from the frame, the denoised version of the frame is subtracted from the frame itself. The wavelet transform function WF is applied for denoising the frames. The residual noise at each frame is computed. Deep learning algorithm InceptionResNetv2 is used as backbone, it is a CNN composed of 164 layers and trained over more than a million images from the ImageNet dataset.

In **[9],** the realism of state-of- the-art image manipulations, and how difficult it is to detect them, either automatically or by humans is examined,

- To standardize the evaluation of detection methods, we propose an automated benchmark for facial manipulation detection

- In particular, the benchmark is based on Deep- Fakes, Face2Face, FaceSwap and Neural Textures as prominent representatives for facial manipulations at random compression level and size. The benchmark is publicly available.

- Contains a hidden test set as well as a database of over 1.8 million manipulated images.

This Paper Makes the following contributions:

- An automated benchmark for facial manipulation detection under random compression for a standardized comparison, including a human baseline.
- A novel large-scale dataset of manipulated facial imagery composed of more than 1.8 million images from 1,000 videos with pristine (i.e., real) sources and target ground truth to enable supervised learning. an extensive evaluation of state of the art handcrafted and learned forgery detectors in various scenarios.

The paper intersects several fields in computer vision and digital multimedia forensics.

Face Manipulation Methods: The tracked deformations of the source face are applied to the target face model. As a final step, they blend the altered face on top of the original target video.

Multimedia Forensics: Multimedia forensics aims to ensure authenticity, origin, and provenance of an image or video without the help of an embedded security scheme.

Forensic Analysis Datasets: forensics datasets have been created with significant manual effort under very controlled conditions, to isolate specific properties of the data like camera artifacts.

In **[10]**, a framework for evaluating detection methods under real-world conditions is presented, consisting of cross-model, cross-data, and postprocessing evaluation, and evaluation of state-of-the-art detection methods using the proposed framework is made. Furthermore, the usefulness of commonly used image pre-processing methods is examined. Lastly, human performance on detecting CNN-generated images is evaluated, along with factors that influence this performance, by conducting an online survey. The results suggest that CNN-based detection methods are not yet robust enough to be used in real-world scenarios.

The aim to evaluate state-of-the-art image generation models under an approximation of real-world conditions, using the following three categories:

a *cross-model scenario*, where the type of model used to generate an image is unknown
a *cross-data scenario*, where the data used to train a generative model is unknown
a *post-processing scenario*, where an image is modified with an unknown type of post-processing.

For each category it is examined whether the generalizability of learning-based methods could be improved using commonly used pre-processing methods. The work focuses on facial images, since most applications are targeted on facial generation or manipulation.

The main contributions of the work are:

- A framework is proposed, consisting of three types of evaluation required for robust evaluation under real-world conditions: cross-model, cross-data, and post-processing evaluation
- The most promising state-of-the-art model architectures and pre-processing methods are evaluated.
- A user study is performed with 496 participants and measure human performance of detecting state-of-the-art generated images and factors that influence this performance.

In **[11]**, the use of deep learning for creating as well as detecting deepfakes is explored. It also proposes the use of deep learning image enhancement method to improve the quality of deepfakes created.

Deepfake Creation: Deepfake relies on an autoencoder-decoder pipeline, encoders are widely used in image compression they rely on deep neural networks and by introducing a bottleneck in the network this forces a compressed representation of the original input. as more advanced encoders are introduced, high quality image compression is possible which can facilitate deepfake task as less computational power is required. deepfake creation workby training two autoencoders. One autoencoder learns the features of the source image and the second encoder learns the feature the target image and two encoders share their parameters, then to generate the deepfake image, the target image is reconstructed using the source image's decoder, this will generate an image of the target with features of the source image. This method is the most common way of deepfake image creations and is used by DeepFaceLab and many others.

Deepfake Detection: Mesonet is a neural network designed specifically to detect deepfakes.It relies on an intermediate approach using a deep neural network with small number of layers. This network begins with a pattern of four layers of successive convolutions and pooling, andis followed by a dense network with one hidden layer. convolution and pooling are used to extract features of an image, it common pattern to use a convolution layer followed by a pooling layer as the convolution layer extract the features and the pooling layer creates a down sampled version of the feature map.

To improve generalization, the convolutional layers use ReLU activation functions that introduce non-linearities and Batch Normalization to regularize the output, and the fully connected layers use Dropout to regularize and improve their robustness.

In **[12],** an overview of indicators that can tell us about the fact that face swapping algorithms were used on photos is presented. Main purpose of this paper is to find algorithm or technology that can decide whether photo was changed with Deep Fake technology or not with good accuracy.

As for model DenseNet169 is used with face warping artifacts indicator. It is presumed that the model should work correctly with present algorithms of Deepfake. Another reason of using this kind of model is that the model is not required to be trained any face swapping algorithms to collect training and validation dataset. Only photos of people from the internet need to be collected. For negative examples a photo was used and some noise was added to it. Gaussian blur, exponential blur and Rayleigh blur were tried.

For extracting faces from the picture dlib package is used. After that a random affine

transformation on randomly resized pictures is used. Then a random specific blur is added. Finally, face pictures are resized back and the picture is made whole again. To evaluate the model Celeb-DF dataset is used, which is one of the newest datasets of Deepfake videos. It has about 1 thousand videos with HQ face swapping algorithms used to synthesize part of them. There are good quality synthesized videos with almost none artifacts of original face, small moving parts and other indicators. So, it can be really hard challenge for the model.

In **[13],** an analysis of the methods for visual media integrity verification is presented, that is, the detection of manipulated images and videos. Special emphasis is placed on the emerging phenomenon of deepfakes and, from the point of view of the forensic analyst, on modern data-driven forensic methods. The analysis will help to highlight the limits of current forensic tools, the most relevant issues, the upcoming challenges, and suggest future directions for research.

The review starts with a brief analysis of the most effective manipulation methods proposed in recent years. Then, integrity verification methods are described, beginning with conventional approaches, then moving to deep learning-based approaches, to conclude with specific deepfake detection methods. In the next section, a discussion of the state of multimedia forensics and its perspectives after the advent of deep learning is carried out. A list of the datasets most widespread in the field is presented. Then, the further major themes of counter-forensics and fusion are considered. Finally, future research directions are outlined and conclusions are drawn.

In **[14],** a method is proposed to detect forged videos of faces by placing the method at a mesoscopic level of analysis. Indeed, microscopic analyses based on image noise cannot be applied in a compressed video context where the image noise is strongly degraded. Similarly, at a higher semantic level, human eye struggles to distinguish forged images, especially when the image depicts a human face. That is why it proposes to adopt an intermediate approach using a deep neural network with a small number of layers.

The two following architectures have achieved the best classification scores among all our tests, with a low level of representation and a surprisingly low number of parameters. They are based on well-performing networks for image classification that alternate layers of convolutions and pooling for feature extraction and a dense network for classification.

Meso-4: This network begins with a sequence of four layers of successive convolutions and pooling, and is followed by a dense network with one hidden layer. To improve generalization,

the convolutional layers use ReLU activation functions that introduce non-linearities and Batch Normalization to regularize their output and prevent the vanishing gradient effect, and the fully-connected layers use dropout to regularize and improve their robustness. In total, there are 27,977 trainable parameters for this network.

MesoInception-4: An alternative structure consists in replacing the first two convolutional layers of Meso4 by a variant of the inception module introduced by Szegedy et al.The idea of the module is to stack the output of several convolutional layers with different kernel shapes and thus increase the function space in which the model is optimized. Instead of the 5 × 5 convolutions of the original module,

A 3 × 3 dilated convolutions in order to avoid high semantic is used. This idea of using dilated convolutions with the inception module can be found in as a mean to deal with multi-scale information, but we have added 1×1 convolutions before dilated convolutions for dimension reduction and an extra 1×1 convolution in parallel that acts as skip-connection between successive modules.

In **[15]**, the deepfake detection technologies Xception and MobileNet are considered as two approaches for classification tasks to automatically detect deepfake videos. Training and evaluation datasets from FaceForensics++ comprising four datasets generated using four different and popular deepfake technologies are utilized. The results show high accuracy over all datasets with an accuracy varying between 91-98% depending on the deepfake technologies applied. A voting mechanism that can detect fake videos using the aggregation of all four methods instead of only one is also developed.

In this paper, different deep learning solutions to automatically classify and hence detect deepfake videos are considered. Specifically, FaceForensics++ is utilized as the source video data and used this data to train two neural networks: Xception and MobileNet using pre-processed images. The training of each network produces four models, each corresponding to one of four different mainstream deepfake software platforms. These include Deepfakes, Face2Face, FaceSwap and Neural Textures. The result of the model's evaluations demonstrates a high degree of accuracy in distinguishing real and fake videos however this accuracy is also highly sensitive and depends greatly on the deep fake platform used. To address this, a voting mechanism is presented that leverages the outputs of the various models to provide a more robust solution. Overall literature summery is shown in table 2.1.

Table 2.1 Literature survey summary

Sl. No	Title of thepaper	Techniques used	Dataset used	Limitation
1.	Deepfake Video Detection Using Recurrent Neural Networks	uses LSTM andRNN, CNN forfeature Extraction, LSTM for sequence processing	HOHA	The robustness of the system can still be increased against manipulated videos usingunseen techniques.
2.	DeepFakes Detection in Videos using Feature Engineering Techniques in Deep Learning Convolution Neural Network Frameworks	DWT, CNN+SIFT	Fraudulent mediawhich can be in the form of fake texts, fake voices,fake videos, and fake photographs.	Detecting why DWT is generalizing at a different rate and other factors needed to be explored to detect morecomplex DeepFakes
3.	Combining Deep Learning and Super-Resolution Algorithms for Deep Fake Detection	UADFV, CelebA	CNN Resnet50 model with super-resolution algorithms.	At first, research could berepeated applying another popular method for Deep Fake detection such as one using RNN models. At the same time, for ResNet50v2 model couldbe replaced another ResNet model

4.	Deepfake Detection with Clustering-based Embedding Regularization	Meso4 algorithm,FWA algorithm, The EVA algorithm, the Multitask algorithm and thelast algorithm Xception-c23	UADFV, Celeb- DF and DeepFakeDetection	
5.	Deepfake Detection UsingSVM	DeepFake, Image Processing, SVM, GenerativeAdversarial Net,DFT	CelebA dataset.	The proposed method is that it shows odd accuracies with fewer amounts of training data.
6.	DeepFakes and Beyond: A Survey of FaceManipulation and Fake Detection	CNN + RNN	DFFD database	
7.	Fighting deepfake by residual noise using convolutional neural networks	InceptionResNetV2, CNN Deep learning algorithm	FaceForensics dataset, DFDC	should be applied to the whole videos available inthe datasets
8.	Deepfakes Creation and Detection Using Deep Learning	MesoNet CNN	Deepfake detection challenge	
9	FaceForensics++: Learning toDetect Manipulated Facial Images	Detection using Steganalysis method	FaceSwap, DeepFakes, Face2Face,Neural Textures, Post processing VideoQuality	
10	Detecting CNN-Generated Facial Images in Real-World Scenarios	Uses Generative Adversarial Networks (GANs)	CelebA-HQ (CAHQ) and Flickr-FacesHQ (FFHQ)	

Chapter 3

ANALYSIS

3.1 Problem Identification

A realistic-looking video that seemed to show former President Barack Obama (Fig.3.1) cussing and calling President Donald Trump a "total and complete dips---," posted by Buzzfeed went viral in 2018, bringing attention to the dangers of a controversial video editing technology that many have called "the future of fake news." About halfway through the video, it is revealed that Obama had actually not uttered those words, and that they were actually said by "Get Out" Director and Writer Jordan Peele, whose voice and mouth had been digitally inserted into an original-much less scandalous-video of the former president.

Fig. 3.1 A snapshot from the deepfake video of Barack Obama

Peele, Buzzfeed, and Monkeypaw Productions used a controversial, but widely available software to make the video, in an effort to demonstrate the potential dangers of "deepfakes," aka digitally manipulated videos that have the power to "make it look like anyone is saying [or doing] anything at any point in time," that didn't actually happen. According to Buzzfeed, the video took roughly 56 hours to make, along with the assistance of a professional video editor.

Unfortunately, this technology is already being used by non-experts for nefarious purposes, including inserting the faces of celebrities into pornographic videos creating-in some instances-very convincing and disturbing results.

Through the literature survey we have inferred that there exist many platforms for the creation of deepfakes which are easily available to people but only a few platforms exist for the detection of such videos. People who are not aware of the existence of such technologies

might find it difficult to differentiate between real and deepfake videos. Even if a person is suspicious about the authenticity of a video, there are very few means for them to check whether the video is legitimate. All these problems are what led to the idea of a deepfake detection system which can be useful to the people in bifurcating real and fake videos.

3.2 Objectives

- The proposed system can be used as a tool in digital media forensics which can prove to be vital in the real-world scenario, considering the adversary's ever-evolving techniques in generating more deceptive deepfakes.
- To pre – process the datasets and detect face in the video and to crop the face so that the resultant "Face-only" video can be used to feed into the algorithm as input.
- To design and develop a Deep Learning algorithm which can detect whether a video is a deepfake or pristine.

3.3 Methodology

Methodology is the systematic, theoretical analysis of the methods applied to a field of study. It comprises the theoretical analysis of the body of methods and principles associated with a branch of knowledge. Typically, it encompasses concepts such as paradigm, theoretical model, phases and quantitative or qualitative techniques.

A methodology does not set out to provide solutions, it is therefore, not the same as a method. Instead, a methodology offers the theoretical underpinning for understanding which method, set of methods, or best practices can be applied to a specific case, for example, to calculate a specific result.

The methodology is the general research strategy that outlines the way in which research is to be undertaken and, among other things, identifies the methods to be used in it. These methods, described in the methodology, define the means or modes of data collection or, sometimes, how a specific result is to be calculated. Methodology does not define specific methods, even though much attention is given to the nature and kinds of processes to be followed in a particular procedure or to attain an objective.

3.4 System Design Specification

We analyzed the problem statement and found the feasibility of the solution of the problem. We read different research paper as mentioned in the survey. After checking the feasibility of

the problem statement. The main step is the dataset gathering and analysis. We analyzed the data set in different approach of training like negatively or positively trained i.e training the model with only fake or real video's but found that it may lead to addition of extra bias in the model leading to inaccurate predictions. So, after doing lot of research, we found that the balanced training of the algorithm is the best way to avoid the bias and variance in the algorithm and get a good accuracy.

3.4.1 Hardware Requirements

In this project, a computer with sufficient processing power is needed. This project requires too much processing power, due to the image and video batch processing. • Client-side Requirements: Browser: Any Compatible browser device.

Table 3.1 Hardware requirements

Sl.no	Parameter	Minimum Requirement
1	Intel Xeon E5 2637	3.5 GHz
2	RAM	8 GB
3	Hard Disk	100 GB
4	Graphic card	NVIDIA GeForce GTX Titan (12 GB RAM)

3.4.2 Software Requirements

- Operating System: Windows 7+
- Programming Language: Python 3.0+
- Framework: PyTorch 1.4, Django 3.0
- Cloud platform: Google Cloud Platform
- Libraries: OpenCV, Face-recognition

Chapter 4

SYSTEM DESIGN

The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces. Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development.

4.1 System Architecture

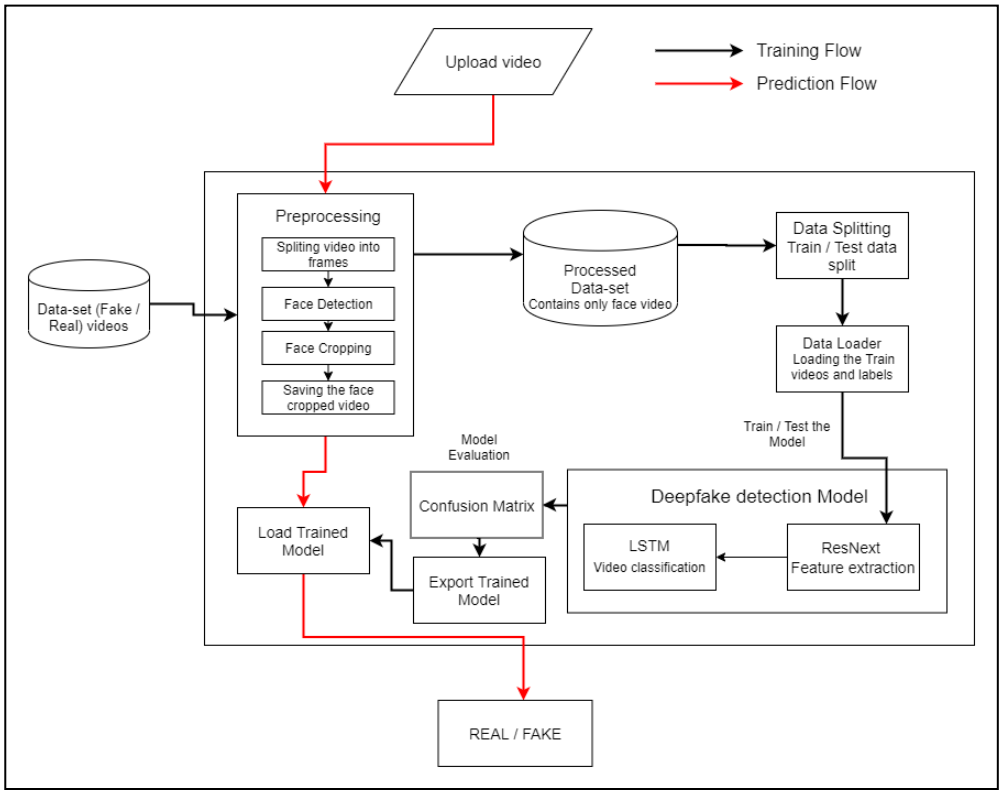

Fig. 4.1 The System architecture of the proposed system.

In the proposed method (Fig. 4.1), a video is uploaded to check whether it is a Deepfake or real. The video is initially Pre-processed to remove the unrequired noise and to detect the features which are necessary. The given video is split into several frames. After splitting, the face of the subject is detected. The detected face is then cropped. After cropping, the frames are combined again to form the cropped video.

This process is followed for each video which leads to the creation of processed dataset which contains the "Face-Only" videos. The videos which do not contain a face are ignored. As a video of 10 seconds at 30 frames per second will have a total of 300 frames, it is computationally difficult to process 300 frames at a single time in the experimental environment. Therefore, a threshold value of 150 frames is selected. The processed video is passed to the model for training.

For training the Deep Neural Network in an optimized way a ResNeXt-50 (32x4d) pre-trained model is used for feature extraction. ResNeXt-50 consists of 50 layers with 32 nodes in each layer and is 4 -dimension model which is capable of learning 25.0 x 10^6 number of parameters. The output of the ResNeXt model is a pooling layer is a feature vector which is then fed into a sequential layer. The sequential feeds the input to the LSTM. 1 LSTM layer is used with 2048 latent dimension and 2048 hidden layers along with 0.4 chance of dropout. The output of the LSTM is further processed by the Linear and Adaptive Average Pooling layers and finally by the softmax layer which produces the output if the video is real or fake

4.2 System Overview

The proposed system aims to use Celeb DF, FaceForencis++ etc as datasets. These datasets contain both real and deepfake videos. The datasets are combined into one and the combined dataset is divided into training and testing data. After the dataset is divided, the pre -processing phase starts. Each video is pre-processed Where the video is converted into individual frames. Face is detected from these frames and the frames are cropped around the detected face. For the Pre-processing of the videos, OpenCV model is used. OpenCV is an open-source library which provides us with the tools to perform almost any kind of image and video processing. OpenCV is written in C++ and its primary interface is in C++. The cropped frames are then combined again to form a video. For computational feasibility, the first 150 frames are selected

We use the Edge detection methods present in OpenCV to detect face. Edge detection is a useful technique to detect he edges of surfaces and objects in the video. Edge detection involves the following steps:

- Noise reduction
- Gradient calculation
- Non-maximum suppression
- Double threshold
- Edge tracking by hysteresis

If the Pre-Processed video is a part of the training data, then it is used to the train model.If the video is a part of the testing data, then it is passed to the trained model. The trained model uses a combination of a CNN, for frame feature extraction combined with an LSTM for temporal sequence analysis to exploit the inconsistencies that exist between the frames which are introduced while creating deepfake videos. Finally, the systems predicts whether the video is a deepfake or real.The below Fig.4.2 shows the overall training workflow of the model.

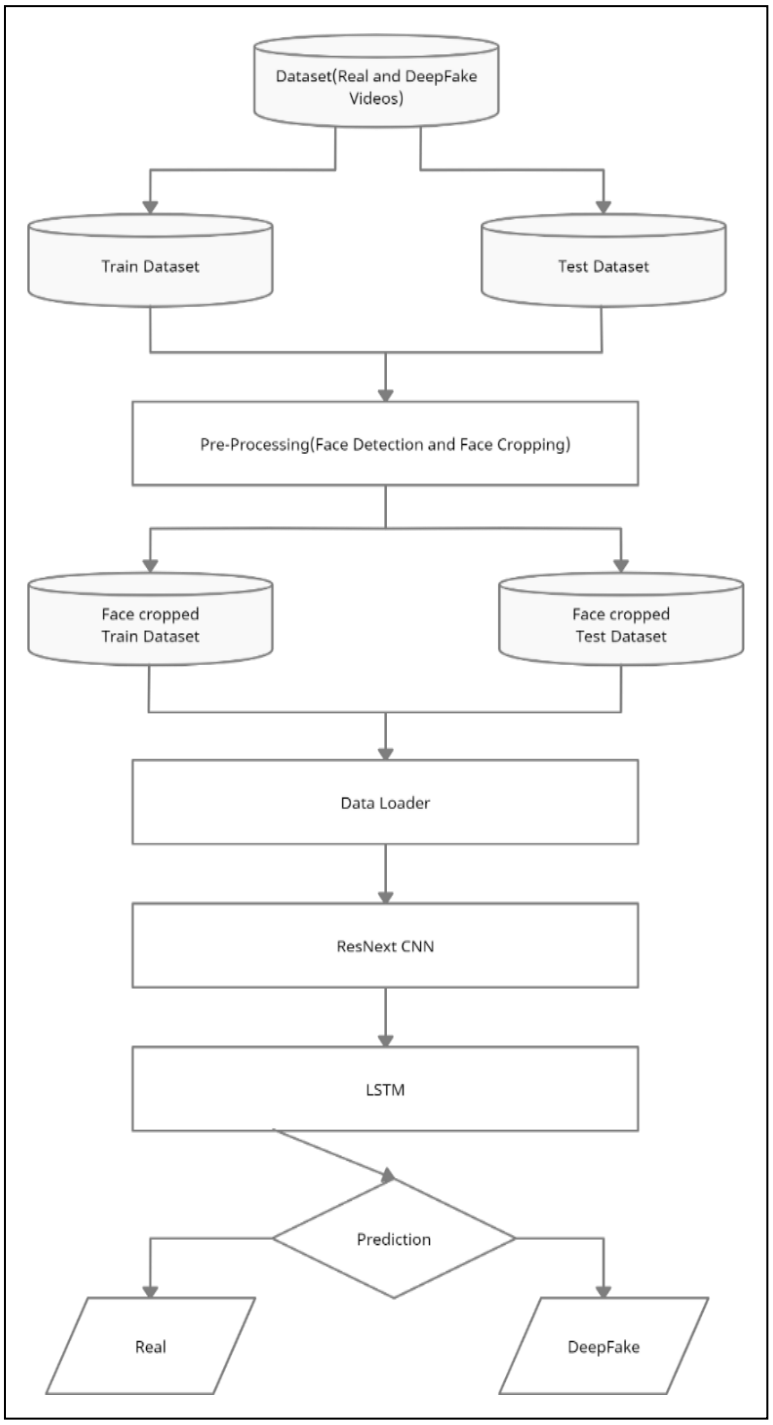

Fig. 4.2 Training workflow

4.2.1 Use case diagram

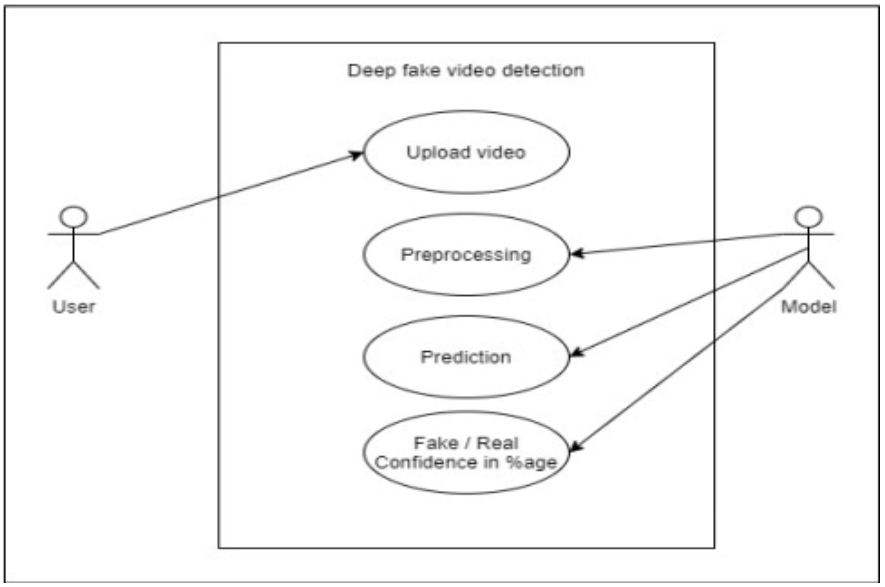

Fig. 4.3 Use case diagram

4.2.2 Sequence diagram

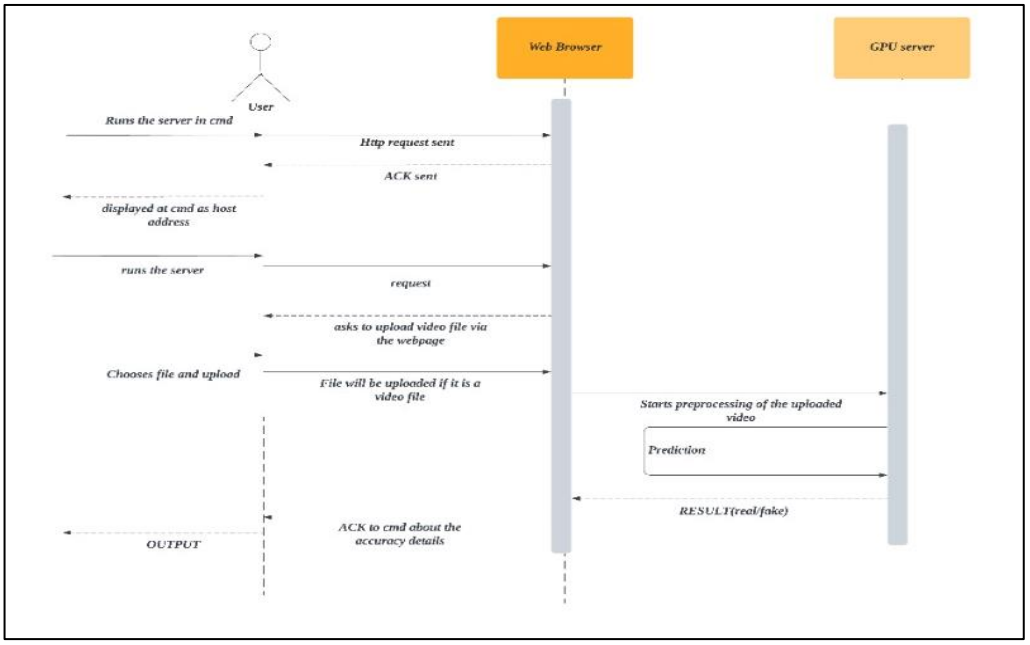

Fig. 4.4 Sequence diagram

4.3 Dataset

Some of the dataset used for Detection are listed below

FaceForensics++: FaceForensics++ is a forensics dataset consisting of 1000 original video sequences that have been manipulated with four automated face manipulation methods: Deepfakes, Face2Face, FaceSwap and NeuralTextures. All videos contain a trackable mostly frontal face without occlusions which enables automated tampering methods to generate realistic forgeries.

Celeb DF: A Large-scale Challenging Dataset for Deepfake Forensics Celeb-DF dataset includes 590 original videos collected from YouTube with subjects of different ages, ethnic groups and genders, and 5639 corresponding Deepfake videos.

Fig 4.5 shows a list of the popular Deepfake datasets.

Database	Real Videos	Fake Videos
UADFV (2018) [16]	49 (YouTube)	49 (FakeApp)
DeepfakeTIMIT (2018) [17]	320 (VidTIMIT)	640 (faceswap-GAN)
FaceForensics++ (2019) [18]	1,000 (YouTube)	1,000 (DeepFake)
Deepfake Detection (2019) [19]	363 (Actors)	3,068 (DeepFake)
Celeb-DF (2019) [20]	590 (YouTube)	5,639 (DeepFake)
DFDC Preview (2019) [21]	1,131 (Actors)	4,119 (Unknown)
DFDC (2019) [22], [23]	19,154 (Actors)	100,000 (4 Different Methods)

Fig. 4.5 Available Deepfake Datasets

4.4 Pre-Processing

It is important to extract frames and perform pre-processing on the videos to reduce computational complexity. The preprocessing steps can be summarized as follows:

1) Frame Selection: Every video has 30 frames per second and ~10sec long i.e., ~300 frames, using all the frames is both computationally expensive and resource consuming. Therefore, we select 10 frames from each video to be used to train and evaluate the models. We perform three different methods of frame selection such that we can compare the performance of our model on each selection method and see whether the selection of frames impacts our results. Additionally, we select a single frame from each video to compare the impact of a single frame and multiple frames on our results.

2) Face detection: As we want to focus on the task of deepfake detection, we would like to only detect the face part with the facial manipulations. Fig 4.6 (a) shows different frame selection methods.

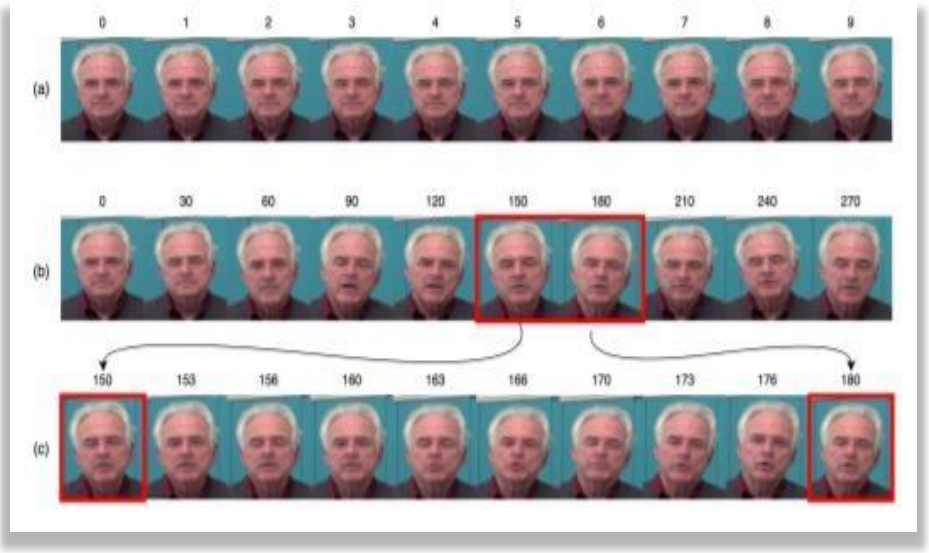

Fig. 4.6(a) Frame Selection methods

Frame selection methods: (a) First 10: First 10 frames are selected (b) Equal Interval: 10 frames extracted at equal interval from ~10sec (~300 frames) video. (c) Most Changes: Frame 150 and Frame 180 have the least similarity score. 10 frames extracted between Frame 150 and Frame 180.

i) Face detection: We use a deep learning-based face detection mode.

ii) Selecting the same face across multiple frames: When detecting multiple faces across multiple frames, faces are swapped while detection due to different faces having higher confidence in different frames. To avoid this and keep all the faces uniform, we keep a bounding box for each face found in a frame and search for faces in that bounding box across all frames. If there is a miss in any faces across multiple frames, we drop that face from the video. This also avoids false detection of the face across multiple frames.

3) Cropping face: As only facial features correspond to facial manipulations cropping the image only to focus on the face reduces the complexity of the model and should improve the model performance. Adding pixel padding to the crop window in a range of 1.5-2.0 times the crop window helps in capturing the spatial differences around the face boundaries.

4) Rescale: After cropping, as all the images should be of uniform size, we rescale them to the same scale and create new face cropped videos.

5) Saving: Save the face cropped video.

Fig 4.6(b) summarizes the steps involved in Pre- processing.

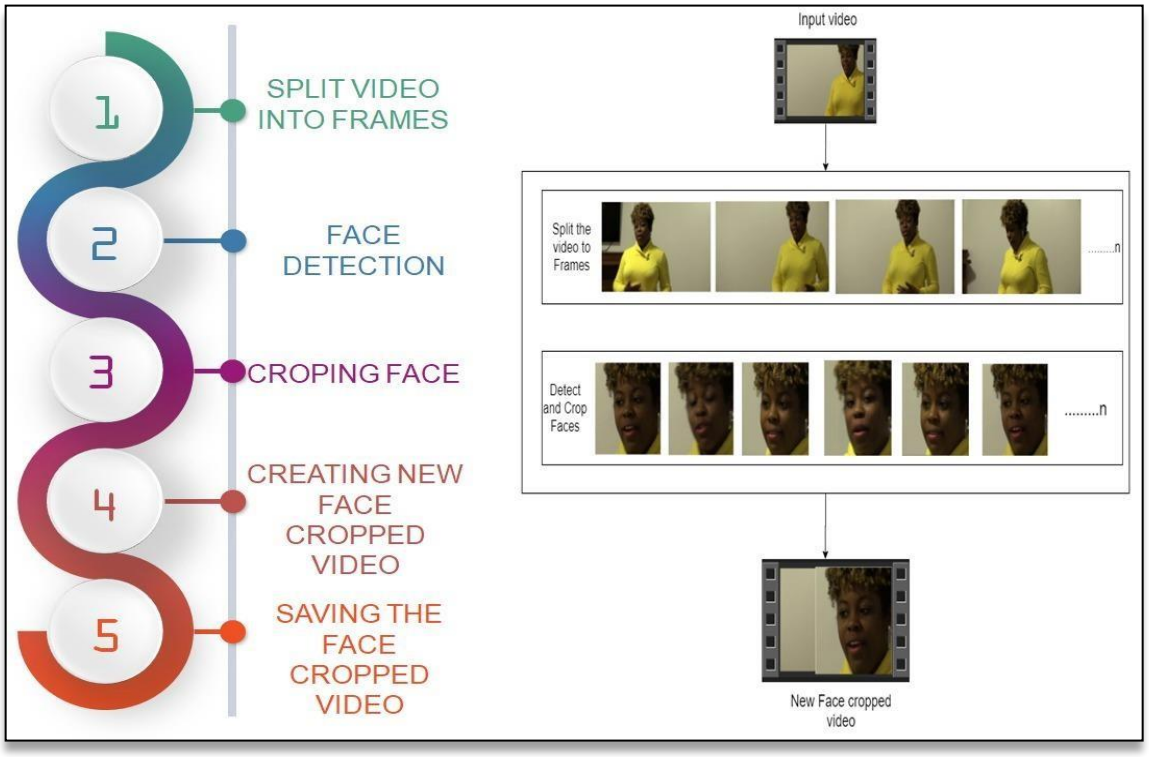

Fig. 4.6 (b) Steps involved in Pre-Processing

4.5 Model and Train

The model consists of resnext50_32x4d followed by one LSTM layer. The Data Loader loads the preprocessed face cropped videos and split the videos into train and test set. Further the frames from the processed videos are passed to the model for training and testing in mini batches.

4.5.1 ResNeXt

ResNeXt is a simple, highly modularized network architecture for image classification. Our network is constructed by repeating a building block that aggregates a set of transformations with the same topology. Our simple design results in a homogeneous, multi-branch architecture that has only a few hyper-parameters to set.

4.5.2 CNN and LSTMs

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network capable of learning order dependence in sequence prediction problems. This is a behavior required in complex problem domains like machine translation, speech recognition, and more.

Convolutional LSTM is a temporal-aware pipeline to identify deepfakes generated and validated on videos collected from multiple video-hosting websites. The proposed model consists of a combination of a CNN, for frame feature extraction combined with an LSTM for temporal sequence analysis. As the deepfakes are generated frame-by-frame, each frame has a new face generated which will have inconsistencies when compared to every other frame and therefore, lacks temporal awareness between frames. These temporal inconsistencies such as flickering in frames and inconsistent choice of illuminants are used to detect deepfakes and results in an accuracy of ~97% on their data-set.

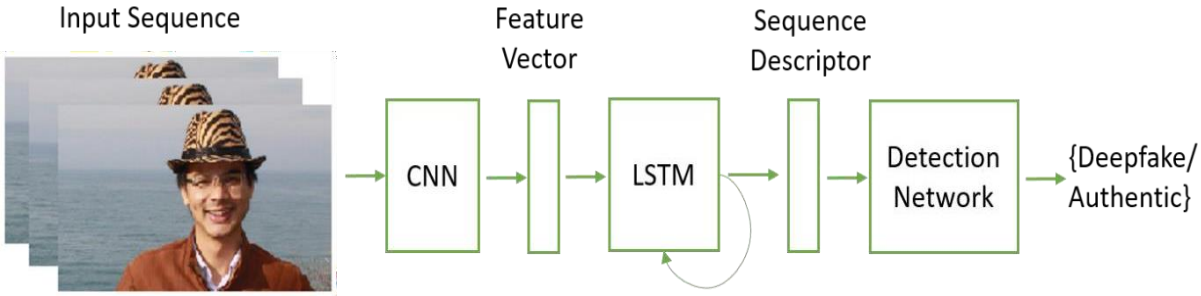

Fig. 4.7 Model Pipeline of DeepFake Detection.

4.5.3 ResNeXt CNN for Feature Extraction

Instead of writing the rewriting the classifier, we are proposing to use the ResNext CNN classifier for extracting the features and accurately detecting the frame level features. Following, we will be fine-tuning the network by adding extra required layers and selecting a proper learning rate to properly converge the gradient descent of the model. The 2048-dimensional feature vectors after the last pooling layers are then used as the sequential LSTM input.

4.5.4 LSTM for Sequence Processing

Let us assume a sequence of ResNeXt CNN feature vectors of input frames as input and a 2-node neural network with the probabilities of the sequence being part of a deep fake video or an untampered video. The key challenge that we need to address is the design of a model to recursively process a sequence in a meaningful manner. For this problem, we are proposing to the use of a 2048 LSTM unit with 0.4 chance of dropout, which is capable to do achieve our objective. LSTM is used to process the frames in a sequential manner so that the temporal analysis of the video can be made, by comparing the frame at ‘t’ second with the frame of ‘t-n’ seconds. Where n can be any number of frames before ‘t’.

4.6 Prediction

A new video is passed to the trained model for prediction. A new video is also preprocessed to bring in the format of the trained model instead of storing the video into local storage the cropped frames are directly passed to the trained model to detect whether it is pristine or fake.

Chapter 5

IMPLEMENTATION

5.1 Introduction

There are many examples where deepfake creation technology is used to mislead the people on social media platform by sharing the false deepfake videos of the famous personalities like Mark Zuckerberg Eve of House A.I. Hearing, Donald Trump's Breaking Bad series where he was introduces as James McGill, Barack Obama's public service announcement and many more. These types of deepfakes creates a huge panic among the normal people, which arises the need to spot these deepfakes accurately so that they can be distinguished from the real videos.

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

5.2 Tools and Technologies Used

UML Tools: Lucid Chart

Programming Languages: Python3, JavaScript

Programming Frameworks: PyTorch, Django.

IDE: Google Colab, Jupyter Notebook, Visual Studio Code.

Cloud Services: Google Cloud Platform

Application and Web servers: Google Cloud Engine

5.3 Libraries

- torch
- torchvision
- os
- numpy
- cv2
- matplotlib
- face_recognition
- json
- pandas
- copy
- glob
- random
- sklearn

5.4 Process Details

5.4.1 Dataset Details

For making the model efficient for real time prediction. We have gathered the data from different available data-sets like FaceForensic++(FF), Deepfake detection challenge (DFDC), and Celeb-DF. Further we have mixed the dataset the collected datasets and created our own new dataset, to accurate and real time detection on different kind of videos. To avoid the training bias of the model we have considered 50% Real and 50% fake videos. Deep fake

detection challenge (DFDC) dataset consists of certain audio alerted video, as audio deepfake are out of scope for this paper. We preprocessed the DFDC dataset and removed the audio altered videos from the dataset by running a python script. After preprocessing of the DFDC dataset, we have taken 1500 Real and 1500 Fake videos from the DFDC dataset. 1000 Real and 1000 Fake videos from the FaceForensic++(FF) dataset and 500 Real and 500 Fake videos from the CelebDF dataset. Which makes our total dataset consisting 3000 Real, 3000 fake videos and 6000 videos in total.

5.4.2 Preprocessing Details

- Using glob, we imported all the videos in the directory in a python list.

- cv2.VideoCapture is used to read the videos and get the mean number of frames in each video. Deepfake Video Detection

- To maintain uniformity, based on mean a value 150 is selected as idea value for creating the new dataset. • The video is split into frames and the frames are cropped on face location.

- The face cropped frames are again written to new video using VideoWriter.

- The new video is written at 30 frames per second and with the resolution of 112 x 112 pixels in the mp4 format.

- Instead of selecting the random videos, to make the proper use of LSTM for temporal sequence analysis the first 150 frames are written to the new video.

5.4.3 Model Details

The model consists of following layers:

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

Sequential Layer: Sequential is a container of Modules that can be stacked together and run at the same time. Sequential layer is used to store feature vector returned by the ResNext model in a ordered way. So that it can be passed to the LSTM sequentially.

LSTM Layer: LSTM is used for sequence processing and spot the temporal change between

the frames.2048-dimensional feature vectors is fitted as the input to the LSTM. We are using 1 LSTM layer with 2048 latent dimensions and 2048 hidden layers along with 0.4 chance of dropout, which is capable to do achieve our objective. LSTM is used to process the frames in a sequential manner so that the temporal analysis of the video can be made, by comparing the frame at 't' second with the frame of 't-n' seconds. Where n can be any number of frames before t.

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

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

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

5.4.4 Model Training Details

Train Test Split: The dataset is split into train and test dataset with a ratio of 70% train videos (4,200) and 30% (1,800) test videos. The train and test split is a balanced split i.e 50% of the real and 50% of fake videos in each split.

Data Loader: It is used to load the videos and their labels with a batch size of 4.

Training: The training is done for 20 epochs with a learning rate of 1e-5 (0.00001), weight decay of 1e-3 (0.001) using the Adam optimizer.

Adam optimizer: To enable the adaptive learning rate Adam optimizer with the model parameters is used.

Cross Entropy: To calculate the loss function Cross Entropy approach is used because we are

training a classification problem.

Softmax Layer: A Softmax function is a type of squashing function. Squashing functions limit the output of the function into the range 0 to 1. This allows the output to be interpreted directly as a probability. Similarly, softmax functions are multi-class sigmoids, meaning they are used in determining probability of multiple classes at once. Since the outputs of a softmax function can be interpreted as a probability (i.e.they must sum to 1), a softmax layer is typically the final layer used in neural network functions. It is important to note that a softmax layer must have the same number of nodes as the output later. In our case softmax layer has two output nodes i.e REAL or FAKE, also Softmax layer provide us the confidence(probability) of prediction.

Confusion Matrix: A confusion matrix is a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarized with count values and broken down by each class. This is the key to the confusion matrix. The confusion matrix shows the ways in which your classification model is confused when it makes predictions. It gives us insight not only into the errors being made by a classifier but more importantly the types of errors that are being made. Confusion matrix is used to evaluate our model and calculate the accuracy.

Export Model: After the model is trained, we have exported the model. So that it can be used for prediction on real time data.

5.4.5 Model Prediction Details

The model is loaded in the application

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

Chapter 6

TESTING

6.1 Introduction

Software Testing is the process used to help identify the correctness, completeness, security and quality of the developed computer software. Testing is the process of technical investigation and includes the process of executing a program or application with the intent of finding errors. It is an investigation conducted to provide stakeholders with information about the quality of the software product or service under test. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. Test techniques include the process of executing a program or application with the intent of finding software bugs (errors or other defects), and verifying that the software product is fit for use.

Software testing involves the execution of a software component or system component to evaluate one or more properties of interest. In general, these properties indicate the extent to which the component or system under test:

- Meets the requirements that guided in its design and development.
- Responds correctly to all kinds of inputs.
- Performs its functions within an acceptable time.
- It is sufficiently usable.
- Can be installed and run in its intended environments.
- Achieves the general result its stakeholder ‘s desire.

As the number of possible tests for even simple software components is practically infinite, all software testing uses some strategy to select tests that are feasible for the available time and resources.

Software testing can be conducted as soon as executable software (even if partially complete) exists. The overall approach to software development often determines when and how testing is conducted. For example, in a phased process, most testing occurs after system requirements have been defined and then implemented in testable programs. In contrast, under an agile approach, requirements, programming, and testing are often done concurrently.

6.2 Unit Testing

It is a level of software testing where individual units or components of software are tested. The purpose is to validate that each unit of software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc.

In object-oriented programming, the smallest unit is the method, which may belong to base or super class, abstract class or derived or child class. Unit testing frameworks, drivers, stubs, and mock/fake objects are used to assist in unit testing. It is performed by using white box testing method.

Unit testing is the first level of software testing and is performed prior to integration testing. It is normally performed by software developers themselves or their peers. In rare cases, it may be performed by independent software testers. The below tables 6.1 show the unit testing for the different models.

Table 6.1 Unit testing

Case No.	Description	Input Data	Actual Output	Expected Output	Status
1	Opening a page to upload video	Upload option is selected	It should display a available videos to upload	Available videos are displayed	PASS
2	Opening a page to upload video	If upload option not selected	It should display a message that upload a video	Display a message to upload a video	PASS
3	Opening a page to upload video	Upload an image	It should display a message that upload a video	Display a message to upload a video	PASS

6.3 Integration Testing

Integration testing is also taken as integration and testing this is the major testing process where the units are combined and tested. Its main objective is to verify whether the major parts of the program are working fine or not. This testing can be done by choosing the options in the program and by giving suitable inputs it is tested.

Table 6.2 Integration testing

Case No.	Description	Input Data	Actual output	Expected Output	Status
1	After Video Upload	Video Split into frames	Split frames undergo pre-processing and prediction to give accuracy	Accuracy of the generated new video is displayed	PASS
2	After Video Upload	Video Split into frames	Face Cropped image with new video displayed	Face Only Image and new face only video is displayed	PASS

6.4 System Testing

System testing is defined as testing of a complete and fully integrated software product. This testing falls in black –box testing where in knowledge of the inner design of the code is not a pre-requisite and is done by the testing team. System testing is done after integration testing is complete. System testing should test functional and nonfunctional requirements of the software. It is the third level of software testing performed after integration testing and before acceptance testing. Itis the process of testing integrated systems to verify that it meets specified requirements. Usually, black box testing method is used. Table 6.3 shows the System testing.

Table 6.3 System testing

Case No.	Description	Input Data	Actual Output	Expected Output	Status
1	Upload a video	If a video is Real	It displays video is real	Video is real	PASS
2	Upload a video	If a video is Fake	It displays video is fake	Video is fake	PASS

Chapter 7

DISCUSSION OF RESULTS

In this section, we will discuss the end user outputs of our system. It depicts how the system will be visible to a user and how the results are displayed to the user.

7.1 Upload Video

The system is designed to detect deepfakes in videos which has human faces. Therefore, all the videos must have at least one clearly detectable face for the system to make prediction. Figure 7.1(a) shows a video which is correctly uploaded.

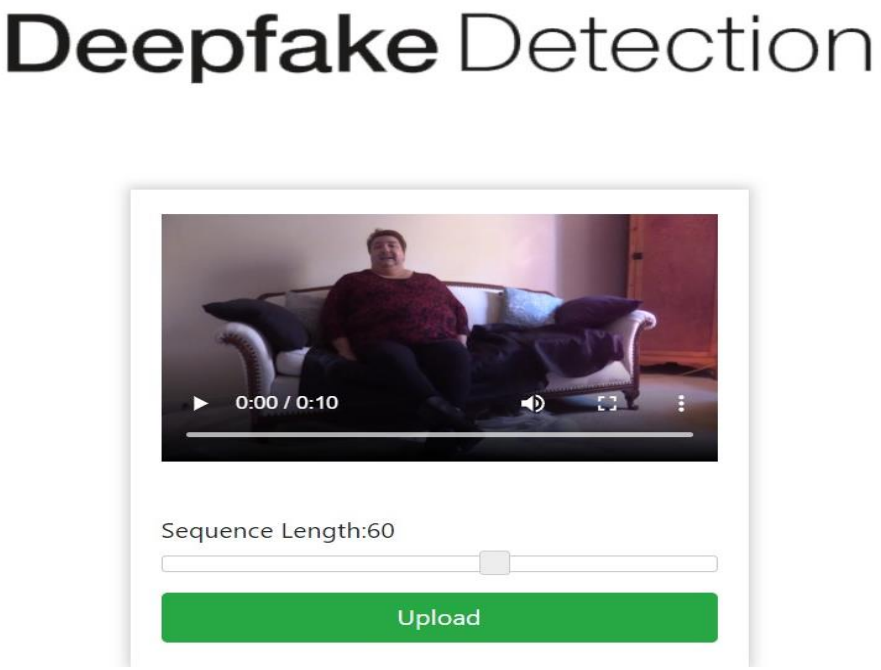

Fig. 7.1 (a) Uploading a proper video

In order to handle a situation where someone uploads a video that has no faces, as shown in figure 7.1(b).

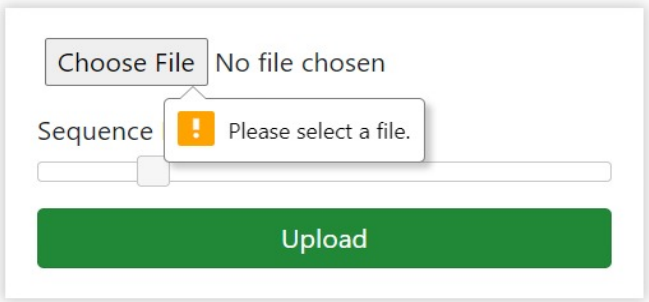

Fig. 7.1(b) If user clicks the upload button without selecting a video

7.2 Frames split and cropped frames

Figure 7.2 shows how the how the uploaded video is split into frames and the face is detected in the video and it is cropped

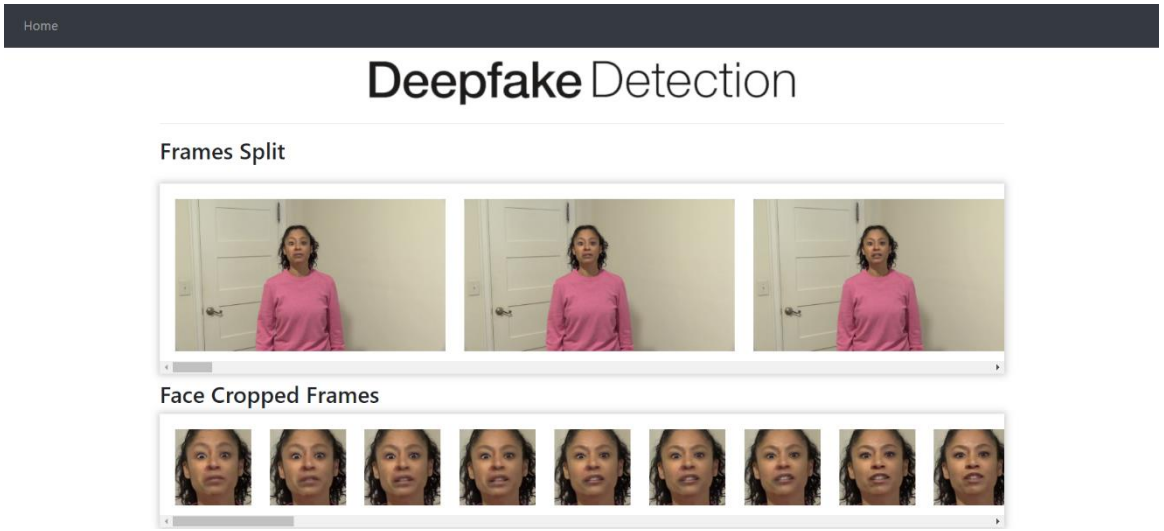

Fig. 7.2 Frames split and face cropped frames

.

7.3 Displaying the prediction

Once the prediction process is completed, the outcome whether the video is fake or real is displayed to the user. In fig 7.3, the uploaded video is detected as fake.

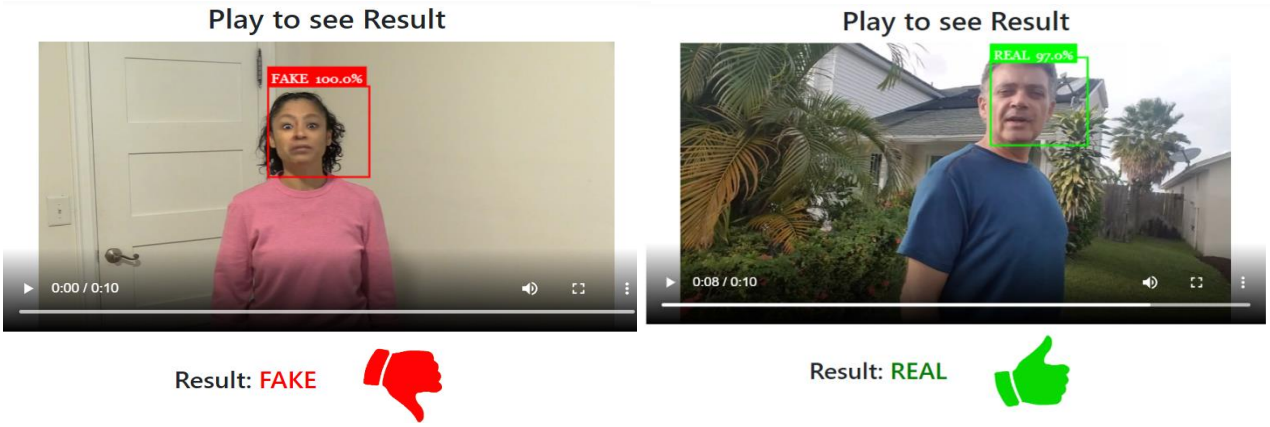

Fig. 7.3 Displaying the predicted result.

Chapter 8

CONCLUSION AND FUTURE ENHANCEMENT

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

There is always a scope for enhancements in any developed system, especially when the project build using latest trending technology and has a good scope in future. Currently only Face Deep Fakes are being detected by the algorithm, but the algorithm can be enhanced in detecting full body deep fakes and also for detecting audio in videos. Accuracy can be increased in case of highly sophisticated videos.

REFERENCES

[1] David Guera, Edward J. Delp. “Deepfake Video Detection Using Recurrent Neural Networks”.

[2] Sonya J. Burroughs, Balakrishna Gokaraju, Kaushik Roy and Luu Khoa. “DeepFakes Detection in Videos using Feature Engineering Techniques in Deep Learning Convolution Neural Network Frameworks”. IEEE Applied Imagery Pattern Recognition Workshop (AIPR) in *2020*

[3] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi and Siwei Lyu Celeb-DF: “A Large-scale Challenging Dataset for DeepFake Forensics”. *arXiv:1909.12962v4 [cs.CR] 16 Mar 2020.*

[4] Nikita S. Ivanov, Anton V. Arzhskov, Vitaliy G. Ivanenko. “Combining Deep Learning and Super-Resolution Algorithms for Deep Fake Detection”.

[5] Kui Zhu, Bin Wu and Bai Wang. “Deepfake Detection with Clustering-based Embedding Regularization” in IEEE Fifth International Conference on Data Science in Cyberspace (DSC) 2020.

[6] Harsh Agarwal, Ankur Singh and Rajeswari D. “Deepfake Detection Using SVM” in Proceedings of the Second International Conference on Electronics and Sustainable Communication Systems (ICESC-2021).

[7] Ruben Tolosana, Ruben Vera-Rodriguez, Julian Fierrez, Aythami Morales and Javier Ortega-Garcia. “DeepFakes and Beyond: A Survey of Face Manipulation and Fake Detection”. arXiv:2001.00179v3 [cs.CV] 18 Jun 2020

[8] Marwa Chendeb El Rai, Hussain Al Ahmad, Omar Gouda, Dina Jamal, Manar Abu Talib and Qassim Nasir. “Fighting Deepfake by Residual Noise using Convolution Neural Networks”. 3rd International Conference on Signal Processing and Information Security (ICSPIS) in 2020.

[9] Andreas Rossler, Davide Cozzolino, Justus Thies, Luisa Verdoliva, Matthias Nießner, Christian Riess. FaceForensics++: Learning to Detect Manipulated Facial Images in *arXiv:1901.08971v3 [cs.CV] 26 Aug 2019.*

[10] Nils Hulzebosch, Sarah Ibrahimi, Marcel Worring. “Detecting CNN-Generated Facial Images in Real-World Scenarios”

[11] Hady A. Khalil, Shady A. Maged. "Deepfakes Creation and Detection Using Deep Learning" in International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC) in 2021.

[12] Artem A. Maksutov, Viacheslav O. Morozov, "Aleksander A. Lavrenov and Alexander S. Smirnov Methods of Deepfake Detection Based on Machine Learning".

[13] Luisa Verdoliva "Media Forensics and DeepFakes": an overview in *arXiv:2001.06564v1 [cs.CV] 18 Jan 2020*

[14] Darius Afchar, Vincent Nozick, Junichi Yamagishi and Isao Echizen. "MesoNet: a Compact Facial Video Forgery Detection Network" in *arXiv:1809.00888v1 [cs.CV] 4 Sep 2018.*

[15] Deng Pan, Lixian Sun, Rui Wang, Xingjian Zhang and Richard O. Sinnott. "Deepfake Detection through Deep Learning" at IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT) in 2020.

[16] Ekraam Sabir, Jiaxin Cheng, Ayush Jaiswal, Wael AbdAlmageed, Iacopo Masi and Prem Natarajan. Recurrent Convolutional Strategies for Face Manipulation Detection in Videos. *arXiv:1905.00582v3 [cs.CV] 16 May 2019.*

[17] Eunji Kim and Sungzoon Cho. "Exposing Fake Faces Through Deep Neural Networks Combining Content and Trace Feature Extractors".

[18] Luca Guuarnera, Oliver Giudice and Sebastiano Battiato. "Fighting Deepfake by Exposing the Convolutional Traces on Images".

[19] Yushaa Shafqat Malik, Nosheen Sabahat, Muhammad Osama Moazzam. "Image Animations on Driving Videos with DeepFakes and Detecting DeepFakes Generated Animations" at IEEE 23rd International Multitopic Conference (INMIC) in 2020.

[20] Mohammed Akram Younus, Taha Mohammed Hasan. "Effective and Fast DeepFake Detection Method Based on Haar Wavelet Transform" at International Conference on Computer Science and Software Engineering (CSASE), Duhok, Kurdistan Region – Iraq in 2020.