DEEP FAKE DETECTION

A Project Report

submitted in partial fulfillment of the requirements

of

Deep Learning

by

Gudditi Kalyan, gudditikalyan123@gmail.com

Guduru Phaneendra, phaneendraguduru6@gmail.com

Madham Kiran Kumar, mkirankumar6839@gmail.com

Under the Guidance of

Saomya Chaudhury

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to total **Techsaksham** team especially **Jay Rathod sir, Saomya Chaudhury sir** and **medinibv mam** for guiding us and providing Valuable resources for completing project. Your support and encouragement helped us stay on track and overcome any challenges we faced. Your expertise and encouragement were instrumental in helping us reach our full potential.

I also like to acknowledge our HOD **P. Rajesh Kumar** sir for fostering a supportive learning environment that encourages collaboration and teamwork. Your leadership has set a positive example for all of us and has set a positive example for all of us and has created a culture of academic excellence.

I would like to express my sincere gratitude to all the members in my group for their hard work, dedication and cooperation throughout the completion of our project.

Abstract of the Project

The growing computation power has made the deep learning algorithms so powerful that creating a indistinguishable human synthesized video popularly called as deep fakes have became very simple. Scenarios where these realistic face swapped deep fakes are used to create political distress, fake terrorism events, revenge porn, blackmail peoples are easily envisioned. In this work, we describe a new deep learning-based method that can effectively distinguish AI-generated fake videos from real videos. Our method is capable of automatically detecting the replacement and reenactment deep fakes. We are trying to use Artificial Intelligence (AI) to fight Artificial Intelligence (AI). Our system uses a Res-Next Convolution neural network to extract the frame-level features and these features and further used to train the Long Short Term Memory (LSTM) based Recurrent Neural Network (RNN) to classify whether the video is subject to any kind of manipulation or not, i.e whether the video is deep fake or real video. To emulate the real time scenarios and make the model perform better on real time data, we evaluate our method on large amount of balanced and mixed data-set prepared by mixing the various available data-set like Face-Forensic++[1], Deepfake detection challenge[2], and Celeb-DF[3]. We also show how our system can achieve competitive result using very simple and robust approach.

Keywords:

Res-Next Convolution neural network,

Recurrent Neural Network (RNN),

Long Short Term Memory (LSTM),

Computer vision.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

Introduction

1.1 Problem Statement

Convincing manipulations of digital images and videos have been demonstrated for several decades through the use of visual effects, recent advances in deep learning have led to a dramatic increase in the realism of fake content and the accessibility in which it can be created. These so-called AI-synthesized media (popularly referred to as deep fakes). Creating the Deep Fakes using the Artificially intelligent tools are simple task. But, when it comes to detection of these Deep Fakes, it is major challenge. Already in the history there are many examples where the deepfakes are used as powerful way to create political tension [14], fake terrorism events, revenge porn, blackmail peoples etc. So it becomes very important to detect these deepfake and avoid the percolation of deepfake through social media platforms. We have taken a step forward in detecting the deep fakes using LSTM based artificial Neural network

1.2 Motivation of the Project

The increasing sophistication of mobile camera technology and the ever growing reach of social media and media sharing portals have made the creation and propagation of digital videos more convenient than ever before. Deep learning has given rise to technologies that would have been thought impossible only a handful of years ago. Modern generative models are one example of these, capable of synthesizing hyper realistic images, speech, music, and even video. These models have found use in a wide variety of applications, including making the world more accessible through text-to-speech, and helping generate training data for medical imaging.

Like any transformative technology, this has created new challenges. So-called "deep fakes" produced by deep generative models that can manipulate video and audio clips. Since their first appearance in late 2017, many open-source deep fake generation methods and tools have emerged now, leading to a growing number of synthesized media clips. While many are likely intended to be humorous, others could be harmful to individuals and society. Until recently, the number of fake videos and their degrees of realism has been increasing due to availability of the editing tools, the high demand on domain expertise.

Spreading of the Deep fakes over the social media platforms have become very common leading to spamming and peculating wrong information over the plat form. Just imagine a deep fake of our prime minister declaring war against neighboring countries, or a Deep fake of reputed celebrity abusing the fans. These types of the deep fakes will be terrible, and lead to threatening, misleading of common people.

To overcome such a situation, Deep fake detection is very important. So, we describe a new deep learning-based method that can effectively distinguish AI generated fake videos (Deep Fake Videos) from real videos. It's incredibly important to develop technology that can spot fakes, so that the deep fakes can be identified and prevented from spreading over the internet.

1.3 Goal and Objectives:

Goal and Objectives:

- Our project aims at discovering the distorted truth of the deep fakes.
- Our project will reduce the Abuses' and misleading of the common people on the world wide web.
- Our project will distinguish and classify the video as deepfake or pristine.
- Provide a easy to use system for used to upload the video and distinguish whether the video is real or fake.

1.4 Statement of scope:

There are many tools available for creating the deep fakes, but for deep fake detection there is hardly any tool available. Our approach for detecting the deep fakes will be great contribution in avoiding the percolation of the deep fakes over the world wide web. We will be providing a web-based platform for the user to upload the video and classify it as fake or real. This project can be scaled up from developing a web-based platform to a browser plugin for automatic deep fake detection. Even big application like WhatsApp, Facebook can integrate this project with their application for easy pre-detection of deep fakes before sending to another user. A description of the software with Size of input, bounds on input, input validation, input dependency, I /O state diagram, Major inputs, and outputs are described without regard to implementation de tail.

CHAPTER 2

Literature Survey

Face Warping Artifacts [15] used the approach to detect artifacts by com paring the generated face areas and their surrounding regions with a dedicated Convolutional Neural Network model. In this work there were two-fold of Face Artifacts.

Their method is based on the observations that current deepfake algorithm can only generate images of limited resolutions, which are then needed to be further transformed to match the faces to be replaced in the source video. Their method has not considered the temporal analysis of the frames.

Detection by Eye Blinking [16] describes a new method for detecting the deep fakes by the eye blinking as a crucial parameter leading to classification of the videos as deepfake or pristine. The Long-term Recurrent Convolution Network (LRCN) was used for temporal analysis of the cropped frames of eye blinking. As today the deepfake generation algorithms have become so powerful that lack of eye blinking can not be the only clue for detection of the deepfakes. There must be certain other parameters must be considered for the detection of deep fakes like teeth enchantment, wrinkles on faces, wrong placement of eyebrows etc.

Capsule networks to detect forged images and videos [17] uses a method that uses a capsule network to detect forged, manipulated images and videos in different scenarios, like replay attack detection and computer-generated video detection. In their method, they have used random noise in the training phase which is not a good option. Still the model performed beneficial in their dataset but may fail on real time data due to noise in training. Our method is proposed to be trained on noiseless and real time datasets.

Recurrent Neural Network [18] (RNN) for deepfake detection used the approach of using RNN for sequential processing of the frames along with Image Net pre-trained model. Their process used the HOHO [19] dataset consisting of just 600 videos.

Their dataset consists small number of videos and same type of videos, which may not perform very well on the real time data. We will be training out model on large number of Realtime data.

Synthetic Portrait Videos using Biological Signals [20] approach extract bio logical signals from facial regions on pristine and deepfake portrait video pairs. Applied transformations to compute the spatial coherence and temporal consistency, capture the signal characteristics in feature vector and photoplethysmography (PPG) maps, and further train a probabilistic Support Vector Machine (SVM) and a Convolutional Neural Network (CNN). Then, the average of authenticity probabilities is used to classify whether the video is a deepfake or a pristine.

Fake Catcher detects fake content with high accuracy, independent of the generator, content, resolution, and quality of the video. Due to lack of discriminator leading to the loss in their findings to preserve biological signals, formulating a differentiable loss function that follows the proposed signal processing steps is not straight forward process**.**

CHAPTER 3

Proposed Methodology

3.1 System Design

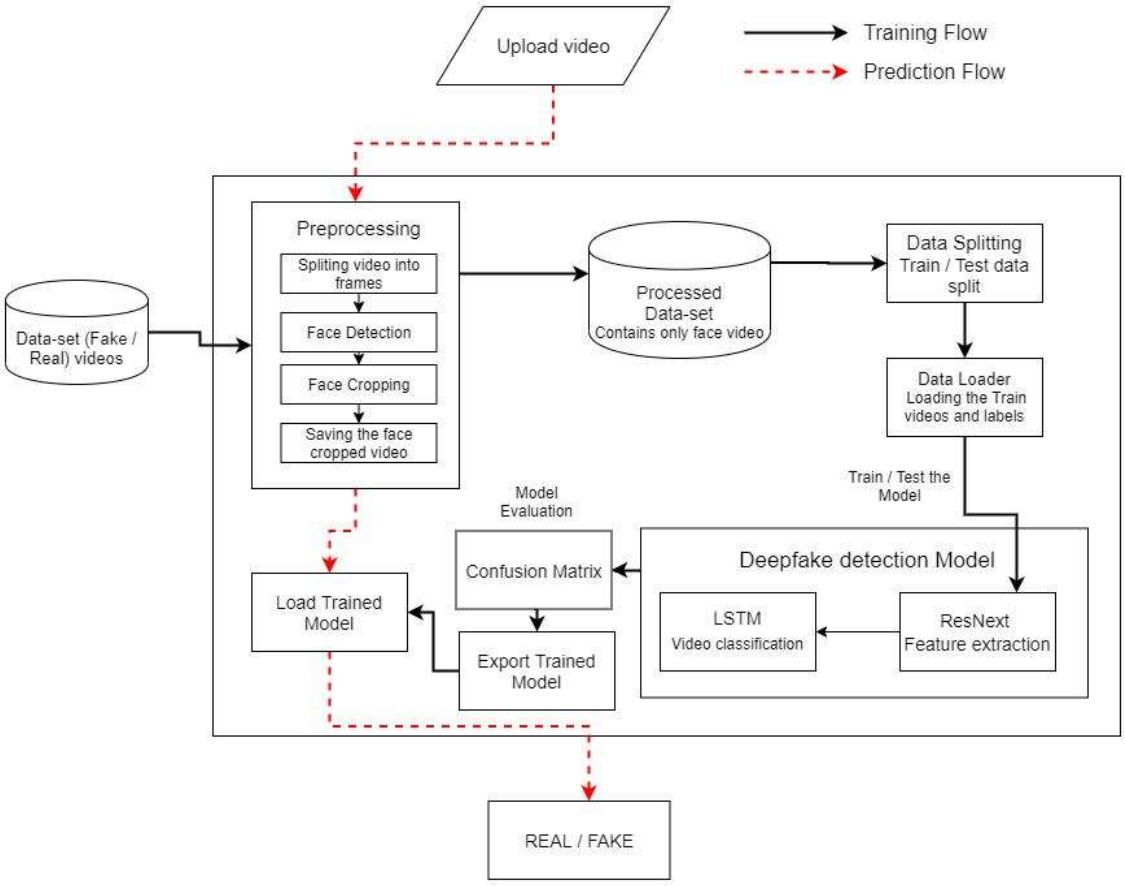

Fig: system design

3.2 Architectural Design

3.2.1 Module 1 : Data-set Gathering

For making the model efficient for real time prediction. We have gathered the data from different available data-sets like FaceForensic++(FF)[1], Deepfake detection challenge(DFDC)[2], and Celeb-DF[3]. Futher we have mixed the dataset the col lected datasets and created our own new dataset, to accurate and real time detectionon different

kind of videos. To avoid the training bias of the model we have consid ered 50% Real and 50% fake videos.

Deep fake detection challenge (DFDC) dataset [3] consist 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)[1] dataset and 500 Real and 500 Fake videos from the Celeb DF[3] dataset. Which makes our total dataset consisting 3000 Real, 3000 fake videos and 6000 videos in total. Figure 2 depicts the distribution of the data-sets.

3.2.2 Module 2 : Pre-processing

In this step, the videos are preprocessed and all the unrequired and noise is removed from videos. Only the required portion of the video i.e face is detected and cropped.The first steps in the preprocessing of the video is to split the video into frames.After splitting the video into frames the face is detected in each of the frame and the frame is cropped along the face. Later the cropped frame is again converted to a new video by combining each frame of the video. The process is followed for each video which leads to creation of processed dataset containing face only videos. The frame that does not contain the face is ignored while preprocessing.

To maintain the uniformity of number of frames, we have selected a threshold value based on the mean of total frames count of each video. Another reason for selecting a threshold value is limited computation power. As a video of 10 second at 30 frames per second(fps) will have total 300 frames and it is computationally very difficult to process the 300 frames at a single time in the experimental envi ronment. So, based on our Graphic Processing Unit (GPU) computational power in experimental environment we have selected 150 frames as the threshold value.While saving the frames to the new dataset we have only saved the first 150 frames of the video to the new video. To demonstrate the proper use of Long Short-Term Memory (LSTM) we have considered the frames in the sequential manner i.e. first 150 frames and not randomly. The newly created video is saved at frame rate of 30 fps and

resolution of 112 x 112.

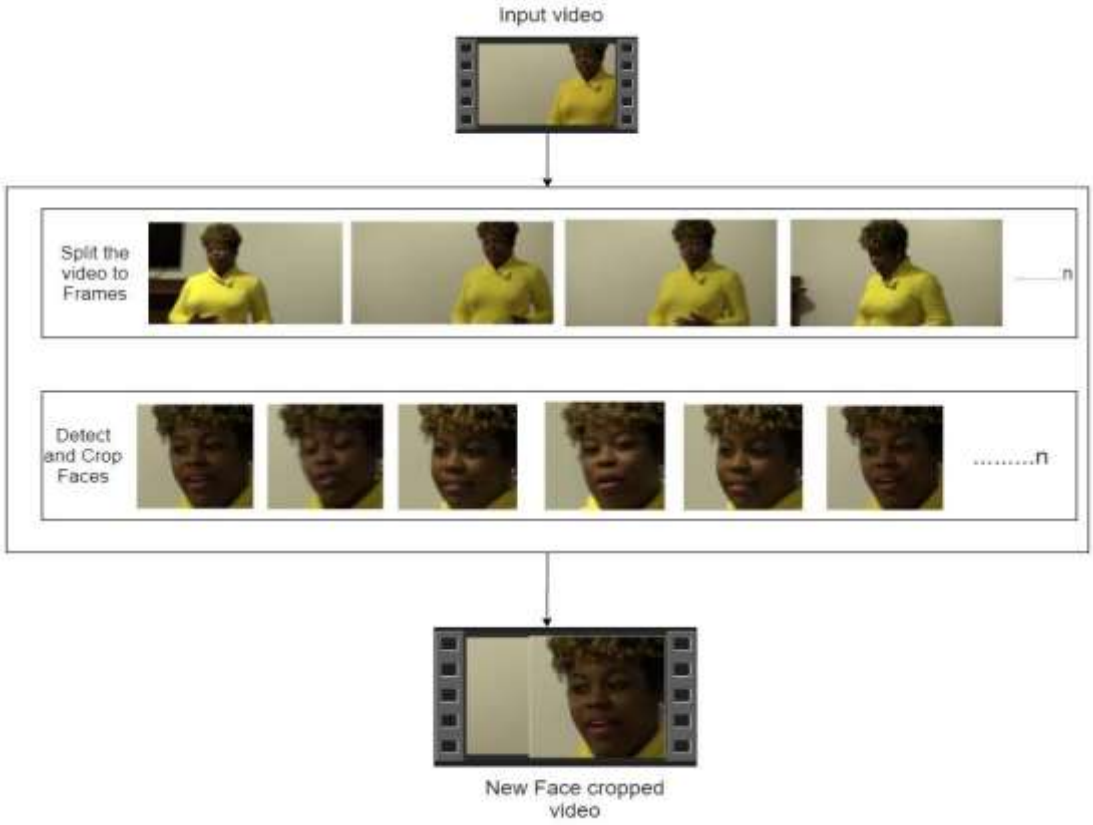

Fig: pre-processing

3.2.3 Module 3: Data-set 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.

3.2.4 Module 4: Model Architecture

Our model is a combination of CNN and RNN. We have used the Pre- trained ResNext CNN model to extract the features at frame level and based on the extracted features a LSTM network is trained to classify the video as deepfake or pristine. Us ing the Data Loader on training split of videos the labels of the videos are loaded and fitted into the model for training.

ResNext :

Instead of writing the code from scratch, we used the pre-trained model of ResNext for feature extraction. ResNext is Residual CNN network optimized for high per formance on deeper neural networks. For the experimental purpose we have used resnext50_32x4d model. We have used a ResNext of 50 layers and 32 x 4 dimen sions.

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 of ResNext is used as the sequential LSTM input.

LSTM for Sequence Processing:

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.The model also consists of Leaky Relu activation function. A linear layer of 2048 input features and 2 output features are used to make the model capable of learning the average rate of correlation between eh input and output. An adaptive average polling layer with the output parameter 1 is used in the model. Which gives the target output size of the image of the form H x W. For sequential processing of the frames a Sequential Layer is used. The batch size of 4 is used to perform the batch training. A SoftMax layer is used to get the confidence of the model during predication.

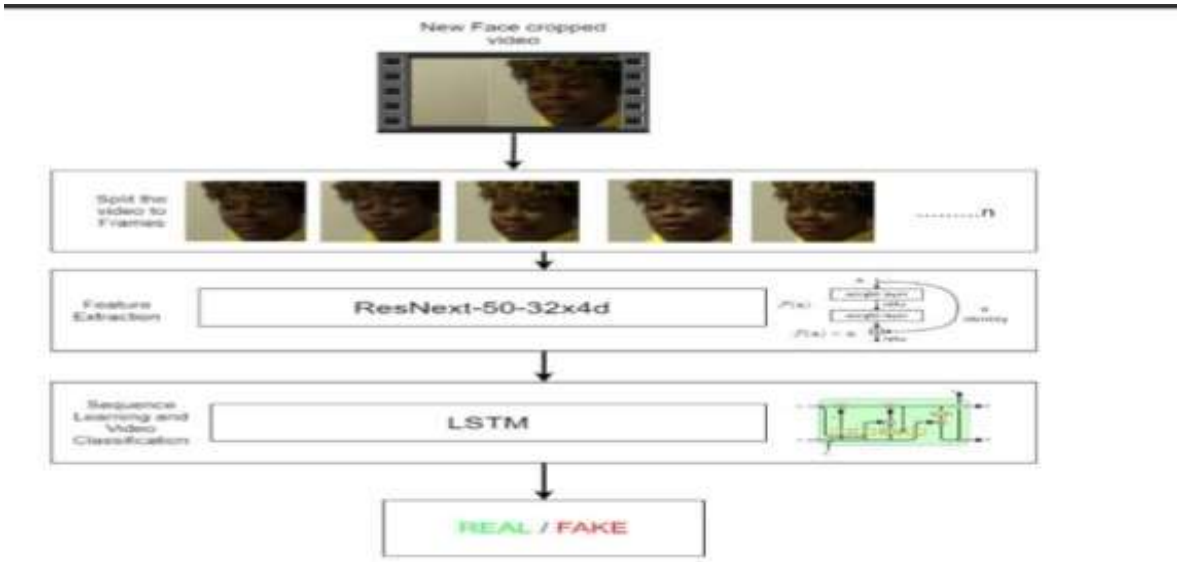

Fig: overview of architecture

3.3 Data Flow Diagram

3.3.1. DFDLevel-0

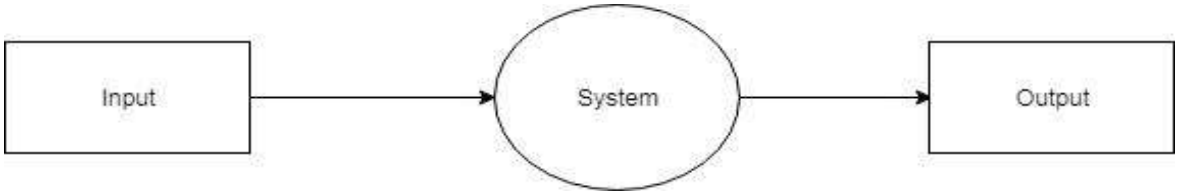

Fig: DFD level-0

DFD level– 0 indicates the basic flow of data in the system. In this System Input is given equal importance as that for Output.

- Input: Here input to the system is uploading video.
- System: In system it shows all the details of the Video.
- Output: Output of this system is it shows the fake video or not.

Hence, the data flow diagram indicates the visualization of system with its input and output flow.

3.3.2. DFDLevel-1

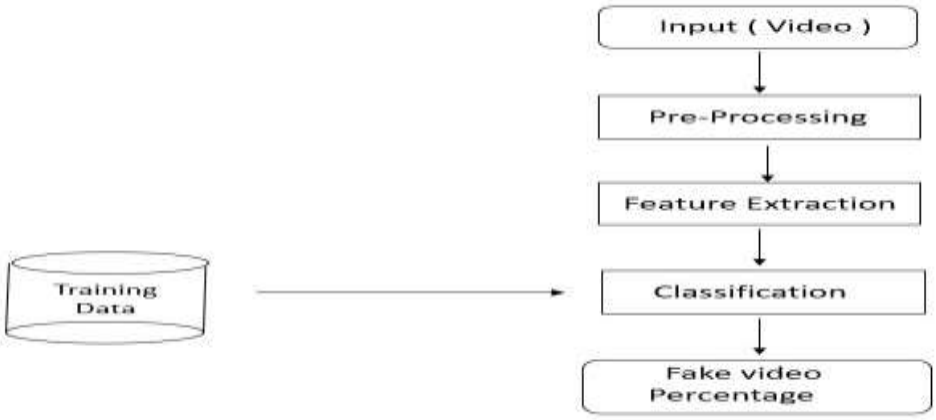

Fig: DFD level-1

[1] DFD Level– 1 gives more in and out information of the system.

[2] Where system gives detailed information of the procedure taking place.

3.3.3. DFDLevel-2

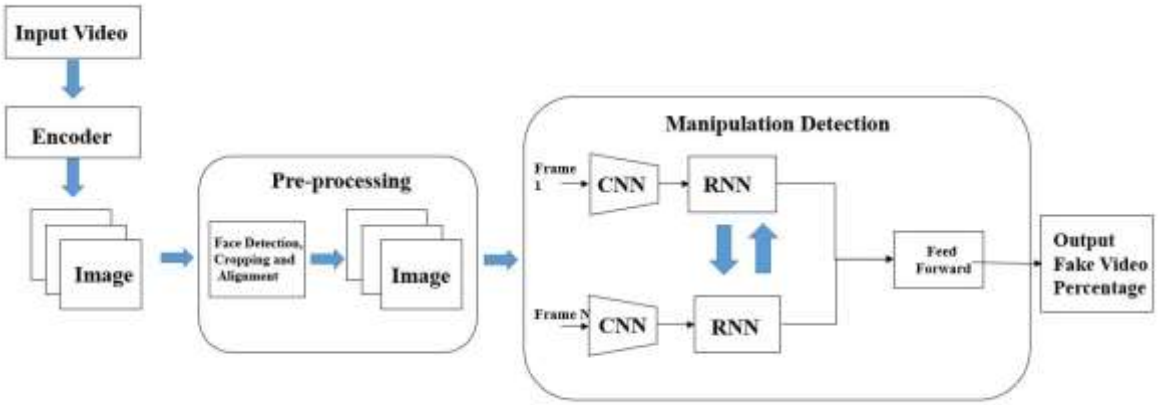

Fig: DFD level-2

[1] DFD level-2 enhances the functionality used by user etc.

3.2 Advantages

Deepfake detection has several key advantages, including:

1. Prevents Misinformation & Fake News

- Stops the spread of misleading or manipulated videos that can influence public **opinion, elections, and social stability.**

2. Protects Personal & National Security

- Prevents identity theft, blackmail, and propaganda by detecting fraudulent videos impersonating individuals or officials.

3. Reduces Financial Fraud

- Prevents scammers from using deepfake audio or video to trick people into making financial transactions or divulging sensitive information.

4. Safeguards Businesses & Brands

- Protects company executives and employees from deepfake scams, ensuring corporate security and brand reputation.

5. Enhances Cybersecurity

- Helps prevent unauthorized access through biometric manipulation, voice cloning, and video-based identity fraud.

6. Ensures Legal Evidence Integrity

- Law enforcement and courts can verify the authenticity of video evidence, preventing wrongful convictions based on manipulated content.

7. Protects Public Figures & Celebrities

- Prevents the misuse of deepfake technology in spreading false endorsements, defamatory content, or reputation-damaging media.

8. Maintains Trust in Digital Media

- Ensures that people can differentiate real from fake content, preserving credibility in journalism and media.

9. Supports Ethical AI Development

- Encourages responsible AI usage and research, discouraging unethical applications of deepfake technology.

10. Improves Online Safety

- Helps detect and remove manipulated content used for cyberbullying, harassment

3.5 Requirement Specification

3.5.1 Hardware Resources Required

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

3.5.2 Software Resources Required

Platform:

1. Operating System: Windows 7+

2. Programming Language: Python 3.0

3. Framework: PyTorch1.4, Django 3.0

4. Cloud platform: Google Cloud Platform

5. Libraries: OpenCV, Face-recognition

CHAPTER 4

Implementation and Result

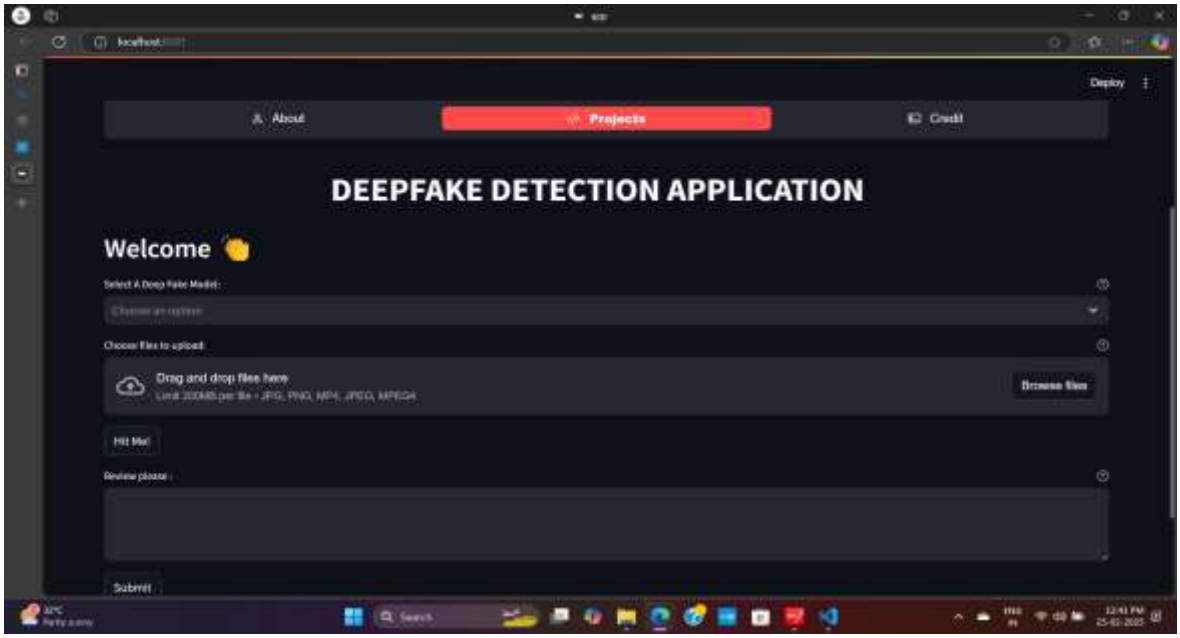

Fig: interface of project

Fig: frames extraction

CHAPTER 5

Discussion and Conclusion

5.1. Git Hub Link of the Project: : https://github.com/M6839/Deep_Fake_Detection.git

5.2. Video Recording of Project: : https://drive.google.com/file/d/1XffTspWWa6isclsNoZ17ZI6rW-H6-04-/view?usp=drivesdk

5.3. Limitations

Deepfake detection models have several limitations, including:

1. Evolving Deepfake Technology
   - As deepfake algorithms improve, they become harder to detect, requiring continuous updates to detection models.
2. High False Positives & False Negatives
   - Detection models may incorrectly classify real videos as deepfakes (false positives) or fail to detect advanced deepfakes (false negatives).
3. Limited Generalization
   - Many models are trained on specific datasets and struggle to detect deepfakes created using different techniques or unseen manipulations.
4. Computational Cost & Resource Intensity
   - Deepfake detection requires significant processing power, making real-time detection challenging, especially on large-scale platforms.
5. Vulnerability to Adversarial Attacks
   - Attackers can modify deepfake videos in subtle ways to evade detection models, reducing their effectiveness.
6. Difficulty in Real-World Applications
   - Factors like video compression, poor resolution, lighting changes, and background noise can degrade detection accuracy.
7. Lack of Standardized Evaluation Metrics

- Different studies and models use varied benchmarks, making it hard to compare performance across detection techniques.

8. Ethical & Privacy Concerns

- Some detection methods require access to biometric data or personal videos, raising concerns about data privacy and security.

9. Scalability Issues

- Detecting deepfakes across massive social media platforms in real time remains a significant challenge.

10. Dependence on Labeled Datasets

- High-quality labeled datasets are needed for training, but deepfake generation methods evolve quickly, making existing datasets outdated.

5.4 Future Scope

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

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

5.5 Conclusion

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

REFERENCES

Data set:

- Face Forensics ++ Real and fake processed videos(https://github.com/ondyari/FaceForensics)
- Celeb-DF Fake processed videos (https://github.com/yuezunli/celeb-deepfakeforensics)

- Celeb-DF Real processed videos https://github.com/yuezunli/celeb-deepfakeforensics
- DFDC Fake processed videos https://www.kaggle.com/c/deepfake-detection-challenge/data
- DFDC Real processed videoshttps://www.kaggle.com/c/deepfake-detection-challenge/data

My github link: https://github.com/M6839/Deep_Fake_Detection.git

Video of project: https://drive.google.com/file/d/1XffTspWWa6isclsNoZ17ZI6rW-H6-04-/view?usp=drivesdk

Appendices (if applicable)

Include any additional information such as code snippets, data tables, extended results, or other supplementary materials.