Deepfake Detection Using Machine Learning

NEELAKANTAN J[1], C LAKSHITH APPAIAH[2], ROHITHASHWA R[3], PROF. POOJA A[4]
[1, 2, 3] *Students, Dept. CSE, ICEAS, Bangalore*
[4] *Assistant Prof., Dept. CSE, ICEAS, Bangalore*

Abstract- ***Deepfake detection has become increasingly challenging due to advancements in computational power and deep learning algorithms. The creation of highly realistic AI-generated videos, commonly known as deepfakes, poses significant threats, including political unrest, fake terrorism events, revenge porn, and blackmail. This work introduces a novel deep learning-based approach to effectively distinguish AI-generated fake videos from real ones. The proposed system combines a ResNeXt convolutional neural network to extract frame-level features with a Long Short-Term Memory (LSTM) recurrent neural network for video classification. It identifies manipulations such as face replacements and reenactments in videos. To enhance real-world performance, the model is trained and evaluated on a diverse, balanced dataset that integrates multiple sources, including FaceForensics++, the Deepfake Detection Challenge, and Celeb-DF. This straightforward yet robust method demonstrates competitive results in combating deepfake threats using AI.***

Indexed Terms- ***Deepfake Detection, ResNeXt, LSTM, Artificial Intelligence, Video Manipulation, Face Replacement, Reenactment, Convolutional Neural Network, Recurrent Neural Network, FaceForensics++, Celeb-DF, AI-Generated Videos.***

I. INTRODUCTION

In the world of ever-growing social media platforms, Deepfakes are considered as the major threat of AI. There are many Scenarios where these realistic face swapped deepfakes are used to create political distress, fake terrorism events, revenge porn, blackmail peoples are easily envisioned. Some of the examples are Brad Pitt, Angelina Jolie nude videos.

It becomes very important to spot the difference between the deepfake and pristine video. We are using AI to fight AI. Deepfakes are created using tools like FaceApp and Face Swap, which using pre-trained neural networks like GAN or Autoencoders for these deepfakes creation. Our method uses an LSTM based artificial neural network to process the sequential temporal analysis of the video frames and pre-trained Res-Next CNN to extract the frame level fea tures. ResNext Convolution neural network extracts the frame-level features and these features are further used to train the Long Short Term Memory based artificial Recurrent Neural Network to classify the video as Deepfake or real. To emulate the real time scenarios and make the model perform better on real time data, we trained our method with large amount of balanced and combi nation of various available dataset like FaceForensic++, Deepfake detection challenge, and Celeb-DF.

Further to make the ready to use for the customers, we have developed a frontend application where the user will upload the video. The video will be processed by the model and the output will be rendered back to the user with the classification of the video as deepfake or real and confidence of the model

II. LITERATURE REVIEW

Authors: Pawar Reena Vishwas, Yelkar Anjali Rajendra
Publisher: IJCRT.ORG Year: January 2022
This paper explores the development and implementation of AI-based online proctoring systems (AIPS) to maintain academic integrity during remote examinations. By leveraging 360-degree security cameras and browser lockdown technologies, the study highlights methods to detect cheating through biometric authentication, including face and voice recognition. The research addresses challenges such as privacy concerns, scalability, and false positives in AI systems while recommending enhancements for current proctoring technologies. The use of a hybrid proctoring system, integrating

automated tools with human oversight, emerges as a robust solution to ensure examination sanctity.

III. METHODOLOGY

Now it is the time to articulate the research work with ideas gathered in above steps by adopting any of below suitable approaches:

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), Deepfake detection challenge(DFDC)[2], and Celeb-DF. Further we have mixed the dataset with 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 DFDCdataset and removed the audio altered videos from the dataset by running a python script.

After preprocessing 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 Celeb DFdataset. Which makes our total dataset consisting of 3000 Real, 3000 fake videos and 6000 videos in total. Figure 2 depicts the distribution of the data-sets

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 environment. 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.

3. DATA-SET SPLITTING

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.

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. Using 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 performance 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 dimensions.

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 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 batch training. A SoftMax layer is used to get the confidence of the model during prediction.

IV. SYSTEM ARCHITECTURE

In this system, we have trained our PyTorch deepfake detection model on an equal number of real and fake videos in order to avoid the bias in the model. The system architecture of the model is shown in the figure. In the development phase, we have taken a dataset, preprocessed the dataset and created a new processed dataset which only includes the face cropped videos.

Creating Deepfake videos To detect the deepfake videos it is very important to understand the creation process of the deepfake. Majority of the tools including GAN and autoencoders take a source image and target video as input. These tools split the video into frames, detect the face in the video and replace the source face with the target face on each frame. Then the replaced frames are then combined using different pre-trained models. These models also enhance the quality of video by removing the left-over traces by the deep-fake creation model. Which results in the creation of a deepfake looks realistic in nature. We have also used the same approach to detect the deepfakes. Deepfakes created using the pretrained neural networks models are very realistic that it is almost impossible to spot the difference by the naked eyes. But in reality, the deepfakes creation tools leave some of the traces or artifacts in the video which may not be noticeable by the naked eyes. The motive of this paper is to identify these unnoticeable traces and distinguishable artifacts of these videos and classify it as deepfake or real video.

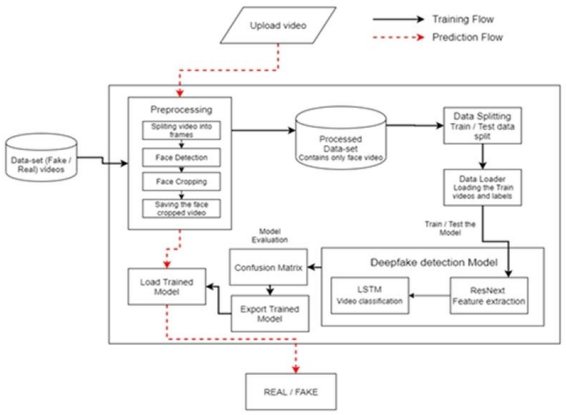

Fig 4.6.1 System Architecture

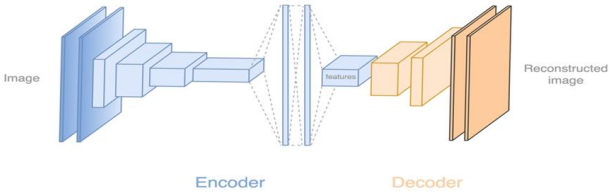

Fig 4.6.2 Deepfake Generation

VI. RESULT AND DISCUSSION

A result is the outcome of actions or occurrences, represented subjectively

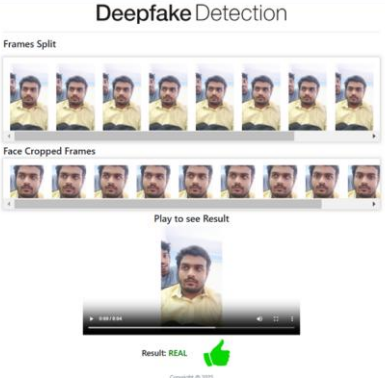

Figure 6.2.3 Real video Output

Case id	Test Case Description	Expected Result	Actual Result	Status
1	Upload a word file instead of video	Error message: Only video files allowed	Error message: Only video files allowed	Pass
2	Upload a 200MB video file	Error message: Max limit 100MB	Error message: Max limit 100MB	Pass
3	Upload a file without any faces	Error message:No faces detected. Cannot process the video.	Error message:No faces detected. Cannot process the video.	Pass
4	Videos with many faces	Fake / Real	Fake	Pass
5	Deepfake video	Fake	Fake	Pass
6	Enter /predict in URL	Redirect to /upload	Redirect to /upload	Pass
7	Press upload button without selecting video	Alert message: Please select video	Alert message: Please select video	Pass
8	Upload a Real video	Real	Real	Pass
9	Upload a face cropped real video	Real	Real	Pass
10	Upload a face cropped fake video	Fake	Fake	Pass

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 can predict the output by processing 1 second of video (10 frames per second) with good accuracy. We implemented the model by using pre-trained ResNext CNN model to extract the frame level features and LSTM for temporal sequence processing to spot the changes between the t and t-1 frame. Our model can process the video in the frame sequence of 10,20,40,60,80,100.

REFERENCES

[1] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies,Matthias Nießner, “FaceForensics++: Learning to Detect Manipulated Facial Images” in arXiv:1901.08971.

[2] Deepfake detection challenge dataset : https://www.kaggle.com/c/deepfake-detection challenge/data Accessed on 26 March, 2024

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

[4] Deepfake Video of Mark Zuckerberg Goes Viral on Eve of House A.I. Hearing : https://fortune.com/2019/06/12/deepfake-mark-zuckerberg/ Accessed on 26 March, 2024

[5] 10 deepfake examples that terrified and amused the internet: https://www.creativebloq.com/features/deepfake-examples Accessed on 26 March, 2024

[6] TensorFlow: https://www.tensorflow.org/ (Accessed on 26 March, 2024)

[7] Keras: https://keras.io/ (Accessed on 26 March, 2024)

[8] PyTorch : https://pytorch.org/ (Accessed on 26 March, 2024)

[9] G. Antipov, M. Baccouche, and J.-L. Dugelay. Face aging with conditional gen erative adversarial networks. arXiv:1702.01983, Feb. 2017

[10] J. Thies et al. Face2Face: Real-time face capture and reenactment of rgb videos. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2387–2395, June 2016. Las Vegas, NV.

[11] Face app: https://www.faceapp.com/ (Accessed on 26 March, 2024)

[12] Face Swap : https://faceswaponline.com/ (Accessed on 26 March, 2024)