Deep Fake Video Detection

Mrs. Kasturi Nikumbh[1], Sahil Karamkar[2], Vaibhav Adhe[3], Shruti Khaire[4], Shivam Ghaware[5]

[1,2,3,4,5]Information Technology, PES Modern College of Engineering, Pune

Abstract

In recent times, the proliferation of free deep learning- based tools has enabled the creation of highly convincing face- swapped videos, commonly known as "DeepFake" (DF) videos. While the manipulation of digital videos has been possible for decades using visual effects, recent advancements in deep learning have significantly enhanced the realism and ease of generating such fake content. These AI- generated media, often referred to as DeepFakes, pose a growing challenge in detecting manipulated content. While generating DeepFakes using AI tools is relatively straightforward, detecting them remains a difficult task due to the complexities involved in training algorithms to accurately identify such manipulations.

This paper proposes a solution to address this challenge by leveraging Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). The CNN extracts features at the frame level, and these features are subsequently used to train the RNN to classify whether a video has been altered or remains authentic. Additionally, the RNN is trained to detect temporal inconsistencies between consecutive frames, which are often introduced by DeepFake creation tools. Our method is tested on a large collection of DeepFake videos sourced from standard datasets, demonstrating competitive performance with a simple yet effective architecture.

Keywords: DeepFake Detection, Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN)

INTRODUCTION

The rapid advancement of smartphone camera technology, coupled The rapid advancement in smartphone camera technology, coupled with widespread access to high-speed internet, has dramatically expanded the reach of social media and video-sharing platforms, making it easier than ever to create and share digital content. At the same time, the increasing computational power of modern systems has made deep learning techniques more accessible and powerful, enabling tasks that were previously considered nearly impossible. Among these transformative technologies, one of the most concerning developments is the rise of "DeepFake" videos, which are generated using deep generative adversarial networks (GANs) to manipulate video and audio content.

The proliferation of DeepFake videos across social media platforms has raised significant concerns, as these videos can be used to spread misinformation, defame individuals, and cause public unrest. The ability to create these hyper-realistic videos has made it difficult for users to differentiate between genuine and manipulated content, posing serious risks to democracy, public trust, and individual privacy. Given these challenges, detecting DeepFake videos has become an urgent necessity to prevent the harmful consequences of such media being disseminated online.

To effectively detect DeepFakes, it is essential to understand how they are created. GANs generate DeepFake videos by taking a source image and video, then replacing the target person's face with that of another individual. These adversarial neural networks are trained on large datasets of facial images and videos, learning to map facial features and expressions from the source to the target. The process involves splitting the video into frames, where each frame is altered to reflect the new face, and then reconstructing the video. This manipulation process is often carried out using autoencoders to ensure that the video appears seamless.

However, due to the computational constraints of GANs, the generated faces must often undergo affine warping to fit the target video, which can lead to noticeable inconsistencies between the manipulated face and the surrounding context. These inconsistencies, or "artifacts," provide key clues that can be leveraged to identify DeepFake videos. Our proposed method builds on this principle by detecting such artifacts. Specifically, we split the video into frames and use a ResNeXt Convolutional Neural Network (CNN) to extract spatial features. Additionally, we employ a Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) to capture the temporal inconsistencies that occur between consecutive frames due to the manipulation. To train the ResNeXt CNN, we simulate resolution inconsistencies in affine face warping directly, allowing the model to effectively learn to identify the subtle artifacts introduced during DeepFake creation.

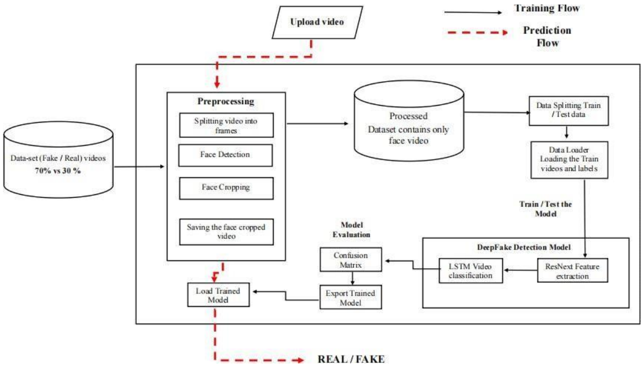
Upload video
Training Flow
Prediction Flow
Data-set (Fake / Real) videos
70% vs 30 %
Preprocessing
Splitting video into frames
Face Detection
Face Cropping
Saving the face cropped video
Processed Dataset contains only face video
Data Splitting Train / Test data
Data Loader Loading the Train videos and labels
Train / Test the Model
Model Evaluation
Confusion Matrix
Export Trained Model
Load Trained Model
DeepFake Detection Model
LSTM Video classification
ResNext Feature extraction
REAL / FAKE

Fig. 1:System Architecture Diagram

LITERATURE REVIEW

Exposing DF Videos by Detecting Face Warping Artifacts This method focuses on detecting artifacts produced during the face swapping process in DeepFake videos. These artifacts arise from the resolution mismatch between the generated face and the surrounding video content. The authors propose a technique that compares the face regions to the surrounding areas using a dedicated Convolutional Neural Network (CNN). This approach is grounded in the observation that current DeepFake algorithms are limited by the resolution at which they can generate faces.

Consequently, after faces are swapped, they often need to undergo transformations to match the face to the target video, resulting in visible inconsistencies. Although this method shows promise, it mainly detects spatial inconsistencies without considering the temporal dimension, which might limit its performance in videos with more complex manipulations.

Exposing AI-Created Fake Videos by Detecting Eye Blinking The detection of DeepFake videos by analyzing eye blinking is another innovative approach. This method exploits the fact that the synthesized videos often fail to correctly replicate the natural physiological signals like eye blinking, which are rarely captured accurately in DeepFake generation. The method involves detecting the presence (or absence) of blinking, which serves as a key indicator for identifying whether the video is real or fake. Although the technique shows good performance using standard eye-blinking datasets, it relies solely on eye blinking as a detection clue.

However, other inconsistencies such as unnatural facial expressions, wrinkles, and teeth enhancement, which are typical artifacts in DeepFake videos, are not considered in this method. Our proposed approach addresses this limitation by considering a wider range of indicators for DeepFake detection.

Using Capsule Networks to Detect Forged Images and Videos This method employs Capsule Networks (CapsNets) to detect forged images and videos, leveraging their ability to better capture spatial relationships and hierarchical structures. The approach aims to detect manipulations like replay attacks and computer-generated video anomalies. While the model has proven effective for the dataset used in this study, it suffers from a potential issue in the form of random noise introduced during the training phase. This noise could lead to inconsistencies in real-time data analysis, making the model less robust for practical applications. Our proposed method, by contrast, aims to train on noiseless and real- world datasets, which should improve the model's generalizability and detection accuracy in real- world settings.

Detection of Synthetic Portrait Videos using Biological Signals In this method, the authors propose extracting biological signals from facial regions in both authentic and fake portrait videos. The system focuses on detecting discrepancies in biological signals, which include facial blood flow and pupil dilation, that are typically difficult to replicate in synthetic videos. By applying transformations to compute spatial coherence and temporal consistency, the method creates feature sets and PPG (photoplethysmography) maps. A probabilistic Support Vector Machine (SVM) and a CNN are used to aggregate these signals and classify whether the video is authentic or fake. Although the method performs well in distinguishing between real and synthetic videos, it faces challenges due to the lack of a robust discriminator to preserve biological signals in the video. Additionally, formulating a differentiable loss function for signal preservation is not straightforward, which limits the practical deployment of this approach in real-time scenarios.

Fake Catcher

The Fake Catcher model is another notable approach designed to detect fake content across a variety of video types, including different resolutions, qualities, and sources. By extracting biological signals and analyzing inconsistencies, the system has been shown to have high accuracy in distinguishing between real and synthetic content. However, the method is limited by the lack of a discriminator that ensures the integrity of biological signals throughout the analysis process. As a result, the detection system might not be as effective when faced with videos that do not exhibit the typical biological signatures or when these signatures are subtly manipulated. Our approach seeks to improve upon this limitation by directly simulating affine warping artifacts in the training phase, allowing for more accurate and comprehensive detection.

PROPOSED METHODOLOGY

The detection of DeepFake (DF) videos has become increasingly critical due to the rapid proliferation of manipulated content across social media platforms and the web. Unlike the abundance of tools available for creating DF videos, there is a significant lack of reliable tools for their detection. Our approach aims to fill this gap by providing a web-based platform for users to upload and classify videos as either real or fake. This platform will serve as an accessible solution to prevent the widespread dissemination of manipulated videos. Additionally, we envision the scalability of this project, with the potential to develop browser plugins for automatic DF detection. Major applications such as WhatsApp and Facebook can integrate this tool, allowing for the pre-detection of DeepFake content before it is shared. A key objective is to evaluate the system's performance based on accuracy, security, user- friendliness, and reliability, ensuring that it provides a robust solution for detecting various types of DeepFakes, including replacement DF, retrenchment DF, and interpersonal DF.

A. Dataset

Our system uses a carefully curated mixed dataset composed of an equal number of real and manipulated videos, sourced from diverse datasets such as YouTube, FaceForensics++ [14], and the DeepFake Detection Challenge dataset [13]. This dataset includes 50% real videos and 50% manipulated DeepFake videos, providing a balanced training foundation. The dataset is split into a 70% training set and a 30% testing set, ensuring that the model is trained on a broad variety of authentic and fake video data.

B. Preprocessing

The preprocessing phase includes several crucial steps to prepare the data for model training. Initially, the video is split into individual frames, after which face detection is performed on each frame. The detected faces are cropped to focus only on the region of interest. To ensure uniformity in the number of frames, we calculate the average number of frames in the dataset and resize the face-cropped videos to match this number. Frames that do not contain faces are excluded from the dataset. Given that processing an entire 10-second video at 30 frames per second (i.e., 300 frames) would be computationally intensive, we limit the number of frames used for training to the first 100 frames of each video for experimental purposes.

C. Model

The model architecture is designed to leverage the power of deep learning for accurate DF detection. It begins with a ResNext50_32x4d model, which is well-suited for feature extraction from the preprocessed video frames. This model is followed by a Long Short-Term Memory (LSTM) layer to process the temporal relationships between frames. The data loader manages the preprocessing steps, splitting the videos into training and testing sets, and feeding them into the model in mini-batches.

D. ResNext CNN for Feature Extraction

To extract frame-level features, we utilize the ResNext50_32x4d CNN model, which is pre-trained for efficient feature extraction. Rather than creating a custom classifier from scratch, we adapt this model to the task of detecting DeepFake videos by fine-tuning it. The ResNext model generates a 2048-dimensional feature vector after the final pooling layers, which represents the high-level features of each frame. These feature vectors are then used as input for the subsequent LSTM layer to capture temporal relationships between frames.

E. LSTM for Sequence Processing

The core challenge in DeepFake detection lies in identifying temporal inconsistencies between consecutive video frames. This is achieved using an LSTM (Long Short-Term Memory) network, which is ideal for processing sequential data like video frames. The LSTM unit processes the sequence of 2048-dimensional feature vectors generated by ResNext. It takes into account the temporal differences between frames, comparing frame 't' with frame 't-n' (where n represents the number of frames before 't'). The LSTM has 2048 units with a dropout rate of 0.4, which helps prevent overfitting and ensures that the network captures long-term dependencies between frames. This setup enables the model to effectively identify inconsistencies in the video caused by DeepFake manipulations.

F. Prediction

Once the model is trained, it is used to classify new videos as either real or fake. A new video is first preprocessed in the same way as the training data, including frame splitting, face cropping, and feature extraction. Instead of storing the cropped frames locally, they are directly passed to the trained model for prediction. The system evaluates the video frame-by-frame, analyzing the temporal relationships between frames, and outputs a classification decision on whether the video is authentic or manipulated. This prediction is presented to the user on the web-based platform, offering a reliable tool for DF detection.

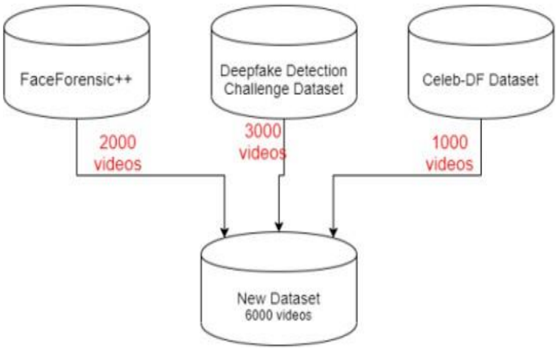

Fig. 2: Datasets

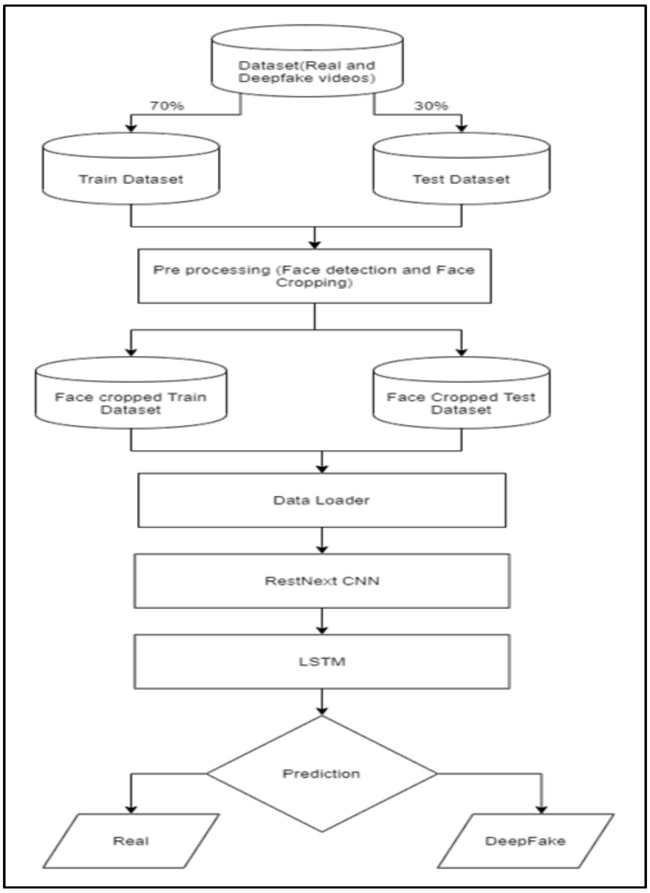
Dataset(Real and Deepfake videos)
70%
30%
Train Dataset
Test Dataset
Pre processing (Face detection and Face Cropping)
Face cropped Train Dataset
Face Cropped Test Dataset
Data Loader
RestNext CNN
LSTM
Prediction
Real
DeepFake

Fig. 3: Training Flow

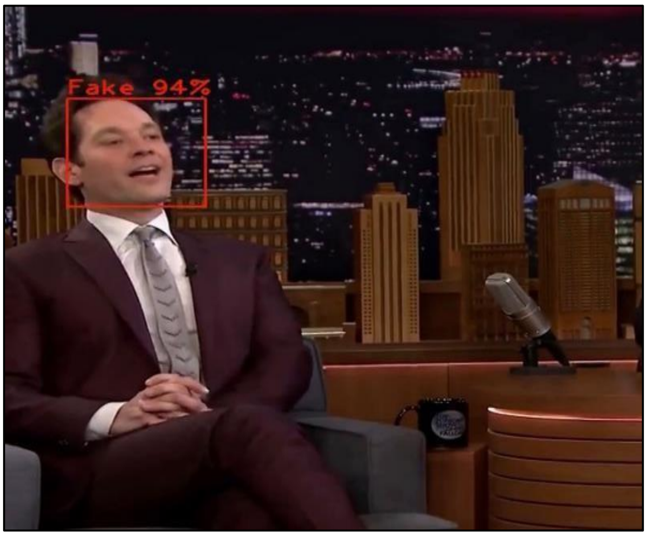

Fig. 4: Expected Results

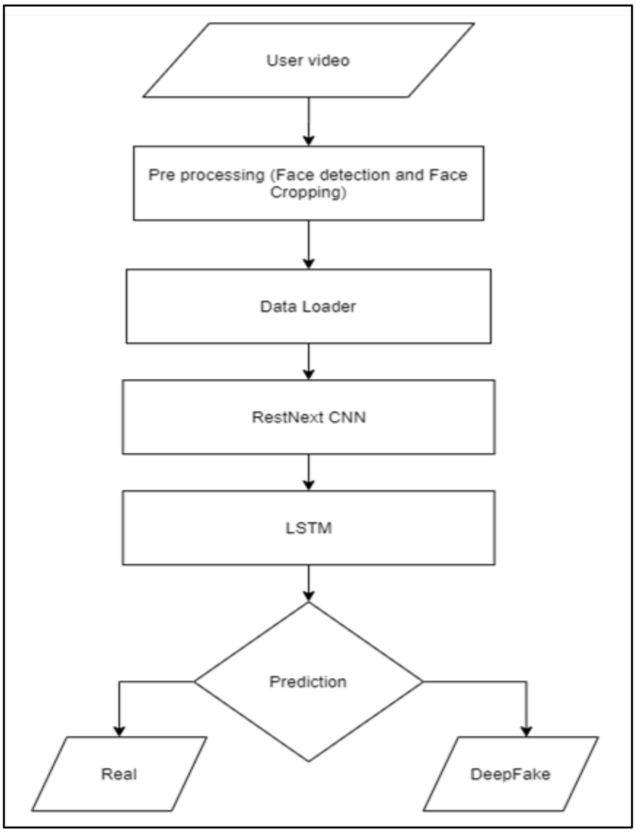

Fig. 5: Prediction flow

tested on real-time data, making it a promising solution for deep fake detection in practical scenarios.

RESULT

The output of the model is going to be whether the video is deepfake or a real video along with the confidence of the model. One example is shown in the figure 4.

CONCLUSION

We introduce a neural network-based approach for classifying videos as either deep fake (DF) or real, with the added feature of providing the confidence level of the model's predictions. This approach is inspired by the techniques used in generating deep fake videos, particularly the process involving Generative Adversarial Networks (GANs) and Autoencoders. Our method performs frame- level analysis using ResNext CNN to extract relevant features from individual frames, and utilizes Recurrent Neural Networks (RNNs) combined with Long Short-Term Memory (LSTM) for video classification and temporal analysis. By processing video frames sequentially, the model is able to detect inconsistencies and artifacts typically introduced during the manipulation process. The proposed model is capable of accurately distinguishing between real and deep fake videos based on the defined parameters outlined in the paper. Given the architecture and methods used, we expect the model to achieve high accuracy when tested on real-time data, making it a promising solution for deep fake detection in practical scenarios

LIMITATIONS

Our current method focuses solely on video-based deep fake detection and does not account for audio manipulation. As a result, it is unable to detect deep fake audio, which remains an area for future exploration. However, we are planning to extend our approach to include audio deep fake detection in the future. This would involve developing specialized models to analyze and classify manipulated audio, further enhancing the overall capability of our system. By incorporating audio analysis, we aim to provide a more comprehensive solution to detecting both visual and auditory deep fakes across a variety of media formats.

References

1. Yuezun Li, Siwei Lyu, "Exposing DF Videos By Detecting Face Warping Artifacts," *arXiv:1811.00656v3*
2. Yuezun Li, Ming-Ching Chang, and Siwei Lyu, "Exposing AI Created Fake Videos by Detecting Eye Blinking," *arXiv*
3. Huy H. Nguyen, Junichi Yamagishi, and Isao Echizen, "Using Capsule Networks to Detect Forged Images and Videos," *arXiv*.
4. Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, and Weipeng Xu, "Deep Video Portraits," *arXiv:1901.02212v2*.
5. Umur Aybars Ciftci, Ilke Demir, Lijun Yin, "Detection of Synthetic Portrait Videos using Biological Signals," *arXiv:1901.02212v2*.
6. Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative Adversarial Nets," *NIPS, 2014*.
7. David Güera and Edward J. Delp, "Deepfake Video Detection Using Recurrent Neural Networks," *AVSS, 2018*.
8. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep Residual Learning for Image Recognition," *CVPR, 2016*.
9. "An Overview of ResNet and its Variants," *Towards Data Science*, https://towardsdatascience.com/an-overview-of-resnet- and-its-variants-5281e2f56035.
10. "Long Short-Term Memory: From Zero to Hero with Pytorch," *FloydHub*, https://blog.floydhub.com/long-short-term- memory-from-zero-to-hero-with-pytorch/.

11. “Sequence Models And LSTM Networks,” *PyTorch Tutorials*, https://pytorch.org/tutorials/beginner/nlp/sequence_models_tuto rial.html.
12. “Confused About Image Preprocessing in Classification?,” *PyTorch Discussion*, https://discuss.pytorch.org/t/confused- about-the-image-preprocessing-in-classification/3965.
13. Kaggle Deepfake Detection Challenge Data, https://www.kaggle.com/c/deepfake-detection-challenge/data.