# CS 535:01 - Deep Fake Video Detection using Deep Learning

GYANA DEEPIKA DASARA
*net id: gd452*
gd452@scarletmail.rutgers.edu

PRANATHI VADDELA
*net id: pv250*
pv250@scarletmail.rutgers.edu

ADITYA KAUSHIK JONNAVITTULA
*net id: aj918*
aj918@scarletmail.rutgers.edu

TEJASHWINI VELICHETTI
*net id: tv186*
tv186@scarletmail.rutgers.edu

ADITYA SEHGAL
*net id: as4099*
as4099@scarletmail.rutgers.edu

PRATEEK MISHRA
*net id: pm883*
pm883@scarletmail.rutgers.edu

***Abstract*—In the recent months, the proliferation of free deep learning-based tools has made it easier to create convincing face swaps in videos, commonly known as "Deep Fake" (DF) videos. While video manipulations have existed for decades, recent advancements in deep learning have significantly enhanced the realism of fake content, making it more accessible to create. Despite the simplicity of generating Deep Fake content using artificial intelligence tools, detecting such manipulations poses a significant challenge. To address this, we have developed an approach that leverages Convolutional Neural Network (CNN) and Recurrent Neural Network (RNN) technologies. Our system utilizes a CNN to extract frame-level features, which are then used to train an RNN. The RNN learns to classify whether a video has undergone manipulation by detecting temporal inconsistencies introduced by DeepFake creation tools. We present competitive results against a large set of fake videos from standard datasets, demonstrating the effectiveness of our approach with a straightforward architecture.**

***Keywords: Deep Learning, Res-Next CNN, Recurrent Neural Network, Long Short Term Memory, Computer Vision.***

## I. Introduction

We assess our technique on a huge and diversified dataset constructed by integrating multiple existing datasets, such as Face-Forensic++, Deepfake Detection Challenge, and Celeb-DF, to improve real-world performance. Our system achieves competitive results by using a simple and robust technique, demonstrating its usefulness in spotting edited movies in real-time circumstances.

Deepfakes are a serious AI-related hazard in the world of fast developing social media platforms. These realistic face-swapped videos, which are frequently created with tools like FaceApp and Face Swap that use pre-trained neural networks like GANs or Autoencoders, are increasingly being used for nefarious purposes like political manipulation, fake terrorism events, and personal harm like revenge porn and blackmail.

It is critical to distinguish between legitimate and deepfake videos. To fight this AI-driven problem, we employ artificial intelligence in our strategy. To assess the sequential temporal characteristics of video frames, we use an artificial neural network based on Long Short-Term Memory (LSTM). A pre-trained Res-Next Convolutional Neural Network (CNN) extracts features at the frame level at the same time.

To improve real-time performance, our model is trained on a large, diversified dataset that includes FaceForensic++, Deepfake Detection Challenge, and Celeb-DF. Users input a video, and our model processes it, producing output that identifies the video as deepfake or real, as well as the model's confidence level in its classification.

## II. Model Architecture

### A. Architecture Components

The proposed model architecture integrates a ResNext50 convolutional neural network (CNN) followed by a single LSTM (Long Short-Term Memory) layer. The Data Loader component preprocesses face-cropped videos, subsequently partitioning them into distinct train and test sets. Within this structure, processed video frames are transmitted to the model for both training and testing purposes, utilizing mini-batch processing for efficiency.

*1) ResNext CNN for Feature Extraction:* Instead of developing a new classifier, our approach advocates leveraging the ResNext CNN classifier for precise feature extraction. This technique enables the accurate detection of frame-level features. Following this extraction, the network undergoes fine-tuning by incorporating additional layers and carefully selecting optimal learning rates, ensuring the model's gradient descent convergence. The resultant 2048-dimensional feature vectors, derived post the final pooling layers, serve as the input for the sequential LSTM.

*2) Sequential Layer:* Sequential Layer, a container for Modules, plays a pivotal role in the model's structure. It facilitates the sequential storage of feature vectors obtained from the ResNext model. This sequential arrangement ensures an organized flow of data for subsequent processing by the Long Short-Term Memory (LSTM) layer.

*3) LSTM for Sequential Processing:* The model operates on a sequence of ResNext CNN feature vectors derived from input frames. It employs a 2-node neural network to determine the likelihood of a sequence belonging to either a deep fake or a prestine video. The main challenge addressed

is designing a model capable of meaningfully processing sequences. Our proposed solution involves employing a 2048-unit LSTM architecture, with a 0.4 dropout probability. This LSTM configuration effectively achieves our objectives. Specifically, the LSTM processes frames in a sequential manner, facilitating temporal video analysis. It accomplishes this by comparing the frame at 't' seconds with frames occurring at 't-n' seconds, with 'n' representing any number of frames preceding 't'.

### *B. Activation Functions and Layers*

The model's architecture harnesses distinct activation functions and layers in specific ways to enhance performance.

1) **Leaky ReLU**: Leaky ReLU, assumes a prominent role within our model. Unlike traditional ReLU, Leaky ReLU allows a small gradient for negative inputs, thereby addressing the 'dying ReLU' problem and promoting greater information flow within the network. In our architecture, Leaky ReLU is strategically applied, enabling faster model building, especially advantageous for larger Neural Networks.
2) **Dropout Layer**: The Dropout Layer significantly contributes to mitigating overfitting concerns within our model. With a set probability of 0.4, this layer selectively drops out neurons during training, effectively preventing the model from becoming overly reliant on specific nodes and ensuring robust generalization during predictions.
3) **Adaptive Average Pooling Layer**: Adding the Adaptive Average Pooling Layer in our model has reduced variance, computational complexity, and extracting low-level features from the surrounding elements of the data. Its integration has optimized the model's ability to discern subtle nuances within the video frames, enhancing overall comprehension and analysis

## III. Dataset Gathering and Pre-Processing

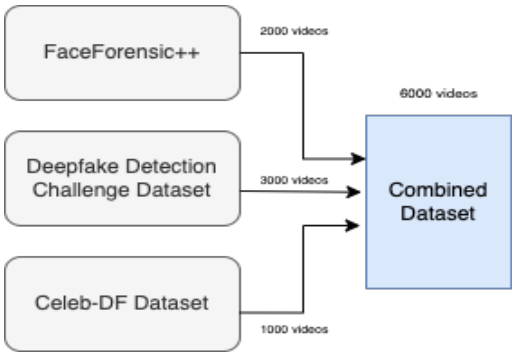


**Fig. 1:** Data Gathering

### *A. Data*

To enhance the model's efficiency for real-time prediction, we curated data from diverse sources, including FaceForensic++ (FF), the Deepfake Detection Challenge (DFDC), and Celeb-DF. Combining these datasets, we created a new dataset aimed at achieving accurate and real-time detection across various video types. To mitigate training bias, our dataset comprises an equal distribution of 50% real and 50% fake videos.

The DFDC dataset initially contained audio-altered videos, to address this, we preprocessed the DFDC dataset by implementing a Python script to remove audio-altered videos.

Following the preprocessing of the DFDC dataset, we selected 1500 real and 1500 fake videos from DFDC, 1000 real and 1000 fake videos from the FaceForensic++ (FF) dataset, and 500 real and 500 fake videos from the Celeb-DF dataset. This aggregation results in a comprehensive dataset comprising 3000 real videos, 3000 fake videos, and a total of 6000 videos overall.

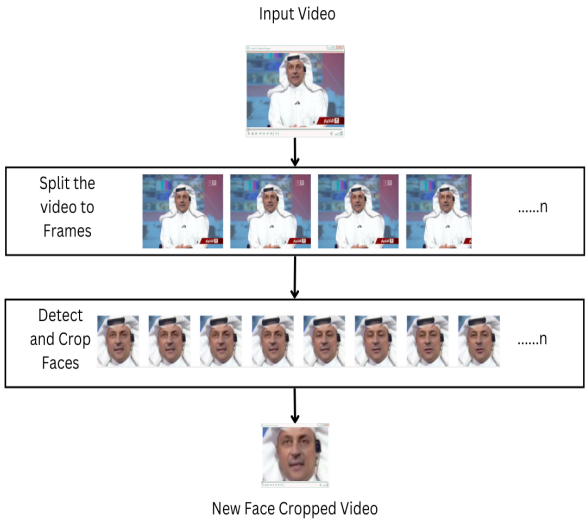


**Fig. 2:** Data Pre-processing

### *B. Pre-Processing*

In this phase, the videos undergo preprocessing to eliminate unnecessary noise and retain only the essential content, specifically the face. The initial step involves splitting the video into frames. Subsequently, facial detection is applied to each frame, and the identified face is cropped. The cropped frames are then compiled to reconstruct the video. This entire process is iteratively applied to each video, resulting in the generation of a processed dataset featuring videos containing only the detected faces. Frames without discernible faces are disregarded during preprocessing.

To ensure uniformity in the number of frames across videos, a threshold value is determined based on the mean total frame count of each video. This threshold is set at 150 frames, taking into consideration both the desire for uniformity and the computational limitations of the experimental environment, particularly concerning the available Graphic Processing Unit (GPU) capacity. Given that a 10-second video at 30 frames per second (fps) would entail 300

frames, processing such a volume of frames simultaneously proves computationally challenging. Therefore, we opt to work within the constraints of our GPU, limiting the selection to the first 150 frames for inclusion in the new dataset.

It is worth noting that the frames are saved sequentially, adhering to the order of appearance in the original video, rather than being chosen randomly. The resulting videos are stored at a frame rate of 30 fps and a resolution of 112 x 112, illustrating the proper application of Long Short-Term Memory (LSTM) in the context of video frame analysis.

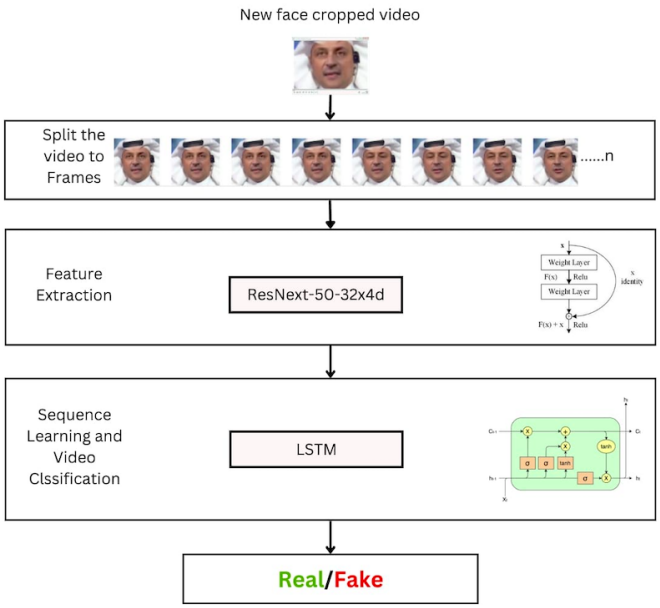


**Fig. 3:** Modelling Overview

### C. Train Test Split

The dataset is split into training and testing subsets using a 80:20 ratio, comprising 4,200 videos for training and 1,800 videos for testing. A balanced distribution ensured an equal representation of real and fake videos in both sets, maintaining parity between classes.

### D. Data Loader

A specialized data loader was implemented to efficiently load videos and their corresponding labels, operating with a batch size of 4 for optimized processing during training.

### E. Training Configuration

The training process spanned 20 epochs, employing a learning rate of 1e-5 (0.00001), weight decay set at 1e-3 (0.001), and the Adam optimizer. The utilization of the Adam optimizer, known for its adaptive learning rate capabilities, facilitated effective convergence of the model parameters.

### F. Loss Function and Softmax Layer

The loss function employed for this classification problem was the Cross Entropy approach, suited for training and evaluating the model's performance in distinguishing between real and fake videos. A Softmax layer, featuring two output nodes corresponding to 'REAL' or 'FAKE', was utilized as the final layer in the neural network architecture. The Softmax function, acting as a multi-class sigmoid, provided probabilities for each class and facilitated confidence estimation in predictions.

### G. Confusion Matrix for Model Evaluation

The evaluation process incorporated the use of a confusion matrix, an essential tool summarizing the model's predictive performance on the classification task. By delineating correct and incorrect predictions for each class, the confusion matrix provided invaluable insights into the classifier's performance, elucidating the nature and types of errors encountered during predictions. This analysis was instrumental in assessing the model's accuracy and identifying specific areas for improvement.

### H. Model Export

Upon successful training, the model was exported, rendering it deployable for real-time prediction tasks, thereby extending its utility beyond the training phase.

This systematic approach to training, evaluation, and model exportation establishes a robust framework for effective classification of videos into 'REAL' or 'FAKE' categories, ensuring both model accuracy and real-world applicability.

## IV. Project WorkFlow

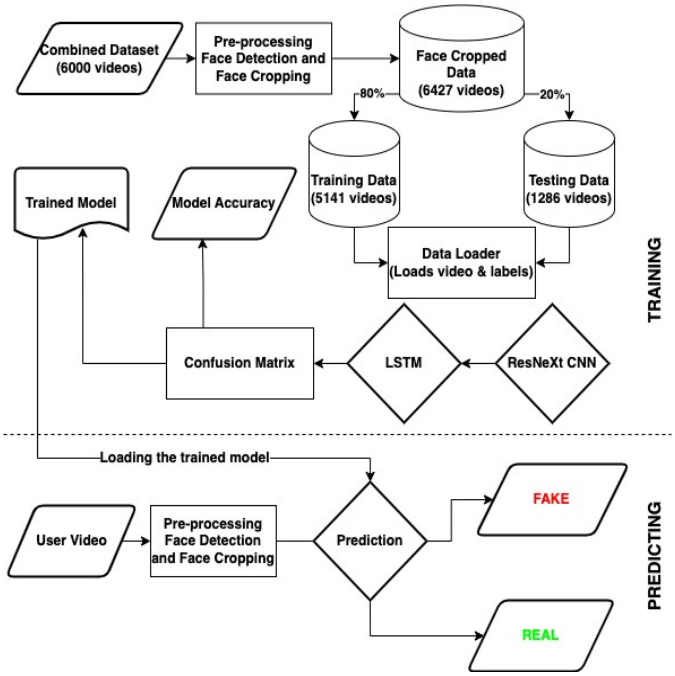


**Fig. 4:** Project Workflow

The two components of the project workflow are training and predicting. The combined dataset, which undergoes face detection and cropping preprocessing, is used for training. The preprocessed data is then divided into a train and test set with an 80:20 ratio. The data loader uses this to load the labels and video after that. The model utilised here is resnext50 for feature extraction, followed by one LSTM layer for sequential video frame processing. This produces an actual vs. expected label confusion matrix. A trained model file and model accuracy are the end results of the training procedure.
The prediction phase comes next, in which a user-submitted video and the previously trained model file are sent for preprocessing. The prediction model's output indicates whether the input video is genuine or fake.

## V. Results


**Fig. 5:** Model Predictions

The model's performance was interpreted based on a variety of performance metrics including the Testing Accuracy, Precision, Recall, and F1-score. The testing results achieved have been tabulated in Table 1. Figure 6 shows the Confusion Matrix plotted based on the model's testing results.

| Performance Metrics | Score |
|---|---|
| Testing Accuracy | 84.29 |
| Precision | 0.876 |
| Recall | 0.813 |
| F1-Score | 0.843 |

## VI. Conclusion

In conclusion, we proposed a neural network-based method for detecting deep fake videos so as to curb hazards of such videos on modern-day social media platforms. For this purpose, we used a pre-trained ResNext Convolutional neural network model to extract frame-level features in a video and an LSTM(Long Short-Term Memory) for temporal sequence processing to detect differences between the t and t-1 frames. With a significant level of accuracy and precision, our model is capable of predicting the output by processing 1 second of a video, which is equivalent to processing 10 frames per second.

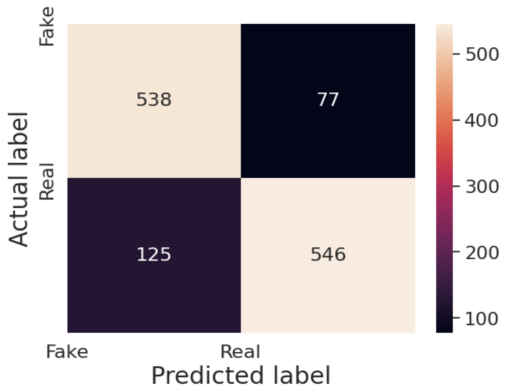


**Fig. 6:** Confusion Matrix

## VII. Future Scope

With the increase in the use of AI-based approaches in the today's world, there is always a scope for enhancements in any developed system, especially one that is so relevant in today's time. The future scope of this project could include working on an enhanced algorithm that could potentially detect full body deep fakes also, though right now, only faces are supported. Furthermore, web based platforms could be converted into mobile apps for the ease of use, as well as for the automatic detection and prevention of Deep fake videos while uploading content on social media.

## References


[1] Yuezun Li and Siwei Lyu. Exposingdf videos by detecting face warping artifacts. *arXiv preprint arXiv:1811.00656v3*, 2018.

[2] Yuezun Li, Ming-Ching Chang, and Siwei Lyu. Exposing ai created fake videos by detecting eye blinking. *arXiv preprint arXiv:*, 2020.

[3] Huy H. Nguyen, Junichi Yamagishi, and Isao Echizen. Using capsule networks to detect forged images and videos. 2019.

[4] Hyeongwoo Kim, Pablo Garrido, Ayush Tewari, and Weipeng Xu. Deep video portraits. *arXiv preprint arXiv:1901.02212v2*, 2019.

[5] Umur Aybars Ciftci, Ilke Demir, and Lijun Yin. Detection of synthetic portrait videos using biological signals. *arXiv preprint arXiv:1901.02212v2*, 2019.

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, 2014.

[7] David G¨uera and Edward J Delp. Deepfake video detection using recurrent neural networks. In *AVSS*, 2018.