**Deepfake Detection**

**Submitted for**

**Artificial Intelligence and Machine Learning CSET301**

Submitted by:

**(E23CSEU0321) ANOUSHA SINGH**

**(E23CSEU0311) MEHAR SINGH**

Submitted to

**DR. YAJNASENI DASH**

**Jan-April 2025**

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING**

A close-up of a logo

Description automatically generated

**INDEX**

|  |  |  |
| --- | --- | --- |
| **Sr.No** | **Content** | **Page No** |
| 1. | Abstract | 3 |
| 2. | Introduction | 4 |
| 3. | Methodology | 5 |
| 4. | Hardware/Software Requirements | 6 |
| 5. | Experimental Result | 6 |
| 6. | Conclusion | 7 |
| 7. | Future scope | 8 |
| 8. | Github Link | 8 |

# Abstract

With the growing use of social media and the easy availability of various AI tools. It has become extremely easy to manipulate any video or photo of a person with the help of these tools. Deepfake is the manipulation of any digital content with the help of AI tools When disseminated across social media platforms, deepfakes have the potential to incite political unrest, spread misinformation, facilitate blackmail, and cause significant harm to the identity of individuals. In this report, we illustrate a new way to tackle this growing problem in our society. Our deepfake detection model known as ClarifAI, is capable of detecting deepfakes in any video with a proven accuracy. Our model uses Res-Next to extract frame wise features of the video to reduce noise and get a more focused video with just the deepfake affected part. Further, we train these features with the help of Long-Short Term Memory (LSTM) and Recurrent Neural Networks (RNN) to identify whether a video is compromised or not i.e., if a video has been altered with the help of deepfake or not. To train our model on real-time data and to establish that our model will work impeccably on any real time data provided, we extracted our dataset from Celeb-DF. Furthermore, with an interactive and easy-to-use interface, it would be very easy to navigate our website especially for new users. The video uploaded by user will be processed and the result will be rendered back to the user through the interface where the confidence level will also be visible. As stated before, our robust model gives accurate results and will definitely help anyone fight against the claws of deepfakes.

# Introduction

In the modern times, with internet users increasing ever second and more people engaging on various social media platforms, cybercrimes have also been on a rise with deepfakes being one of them. When disseminated across social media platforms, deepfakes have the potential to incite political unrest, spread misinformation, facilitate blackmail, and cause significant harm in various other ways. Deepfakes are fairly ease to create with the availability of various applications like FaceSwap etc. Our application ClaifAI, uses methods like LSTM and RNN to tackle this problem. With the help of Res-Next CNN, we have extracted features from the frames in order to reduce noise and to make our model predict better. For our model to giver accurate results for real time problems, we extracted our dataset from Celeb-DF, which provides real time deepfakes of various celebrities for our model to be trained on. Furthermore, to make our application more accessible and ready to use, we have also integrated an easy to use and navigate front end for our application in which the video which is uploaded by the user will be processed and the result will be rendered back to the user on the interface with confidence levels as well.

# Methodology

From gathering and cleaning data to building an interface for end users, each stage is crucial for creating a reliable spam detection system.

1. **Data Collection and Preprocessing**:  
   The first step involves gathering of Deepfake dataset with a mix of real and deepfake videos. Once we have this data, we need to make it ready for model training. This involves several preprocessing steps:

* **Text Cleaning**: The dataset is cleaned by removing any duplicate entries.
* **Framing:** Each video is converted to a sequence of frames out of which few are selected.
* **Cropping**: Each frame is now cropped to a level in which only the face is visible. This removes any background noise from the frame.

1. **Feature Extraction**:  
   After preprocessing, we need to convert the image data into a format that the machine learning model can work with. For this, we used feature extraction with a Residual Network (ResNet), which transforms the images into numerical feature vectors. ResNet helps by capturing important patterns and inconsistencies in facial images, giving more focus to features that are common in deepfake images but rare in real ones, helping the model focus on subtle differences that distinguish deepfakes from genuine videos.
2. **Model Selection and Ensemble Technique**:  
   To build an effective deepfake detection system, we experiment with different models and selected one that best captures both spatial and temporal patterns in video data. Since deepfake artifacts often become more apparent when analysing frame sequences, we used a model designed for sequential learning, namely:

* **LSTM (Long Short-Term Memory):** A type of recurrent neural network (RNN) that is well-suited for handling sequential data like video frames. LSTM networks can retain information across multiple time steps, allowing them to detect subtle temporal inconsistencies in facial movements, blinking patterns, or expressions that are often signs of deepfakes.

1. **User Interface Development**:  
   To make the deepfake detection system easy to use, we developed a web interface using Next.js for the frontend and FastAPI for the backend. The interface is user-friendly and interactive, allowing users to upload a video file and receive an instant classification of whether the video is real or deepfake with a single click. FastAPI handles the backend logic, where the trained model processes the input video—extracting faces, running feature extraction with ResNet, and analyzing temporal patterns using LSTM. Next.js provides a responsive and intuitive frontend, making the tool accessible and suitable for both personal use and integration into larger platforms.

# Hardware/Software Required

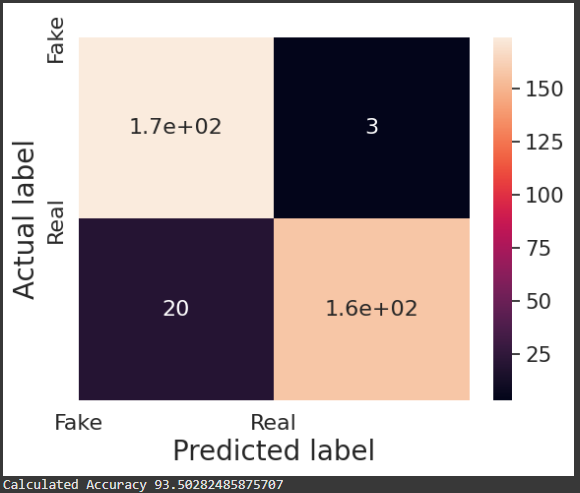
To run this project, a system with at least 8GB of RAM and a 64-bit processor is required to run the model. A NVIDIA GPU would be preferred but not necessary.

For software, the latest version of Python should be installed along with pip to manage dependencies. Essential libraries like NumPy, Pandas, Matplotlib, face-recognition, pytorch, gdown, fastapi, etc must also be installed to run the application smoothly. For frontend, npm should be installed along with nodejs, nextjs, tailwind css, emailjs, etc

An IDE such as VS Code is recommended for writing and executing the code, while a modern browser like Chrome or Edge is needed to run the application and access its interface. This setup ensures that the project runs efficiently and provides an optimal user experience.

# Experimental Results

The classifier was trained on a dataset of real and deepfake videos, achieving an accuracy of near about 93%. The models demonstrated high training and testing accuracy, effectively identifying deepfakes while minimizing false negatives. It performed well in classifying real videos, ensuring reliable detection of deepfakes. The vercel deployed webapp provides quick and accurate results, validating the system's efficiency and ease of use.



# Conclusion

The deepfake detection system successfully addresses the challenge of identifying manipulated video content with high accuracy and reliability. By combining ResNet for spatial feature extraction and LSTM for temporal analysis, the model effectively captures subtle inconsistencies in facial expressions and motion across video frames. This hybrid approach has proven effective in distinguishing real videos from deepfakes, even when the manipulations are visually convincing. With a user-friendly web interface built using Next.js and FastAPI, the system is both accessible and practical for real-world use. This project highlights the potential of deep learning in combating misinformation, ensuring content authenticity, and enhancing digital media security.

# Future Scope

The future scope of this deepfake detection project includes enhancing model accuracy through larger and more diverse datasets, incorporating audio analysis for multimodal detection, and enabling real-time video analysis for use in live platforms. Expanding the system to support explainability features and deploying it as a cloud-based API or browser extension can make it more accessible and practical for media platforms, content creators, and the general public, helping combat the growing threat of misinformation and digital manipulation.

# GitHub Link

<https://github.com/Anousha-Singh/ClarifAI>