**Package Abstract**

**Deep Fake Detection using LSTM**

By 21pd01,21pd04 and 21pd34

**Aim:**

To develop a robust deepfake detection system using Long Short-Term Memory (LSTM) networks, capable of accurately identifying AI-generated forgeries in video content by capturing subtle temporal patterns and dynamics indicative of manipulation.

**Project Description**:

Deepfake technology has emerged as a significant threat to the authenticity and integrity of digital media content, raising concerns about misinformation, privacy infringement, and identity theft. Detecting these AI-generated forgeries has become a pressing challenge, demanding innovative solutions that can adapt to evolving deepfake techniques. This project proposes a novel approach to deepfake detection leveraging Long Short-Term Memory (LSTM) networks, a type of recurrent neural network (RNN) known for its ability to capture temporal dependencies in sequential data.

The proposed system will analyze video sequences frame by frame, extracting features that capture subtle temporal patterns indicative of deepfake manipulation. These features will be fed into an LSTM-based classifier trained on a dataset containing both authentic and deepfake videos. The LSTM network will learn to differentiate between genuine and manipulated content based on the temporal dynamics of facial expressions, movements, and contextual cues.

**Research Paper** :[Link](https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiDu4iZscSFAxUjSGwGHYM4Aw0QFnoECCAQAQ&url=https%3A%2F%2Fwww.politesi.polimi.it%2Fretrieve%2F05af0c90-00dd-4832-90fd-17236641d7b6%2Fthesis-final-version.pdf&usg=AOvVaw1JNYVkcgm8MtKzacf5gDg2&opi=89978449)

**Technologies and Libraries Used:**

**Python**: The primary programming language for implementing the deep learning model, data preprocessing, and evaluation scripts.

**Deep Learning Frameworks**: Libraries such as **TensorFlow** and **PyTorch** for building and training the LSTM network architecture.

**OpenCV:** Used for processing and analyzing video frames, including tasks like face detection, alignment, and feature extraction.

**Data Manipulation Libraries**: NumPy and Pandas for handling and manipulating the dataset during preprocessing and training phases.

**Machine Learning Libraries**: Scikit-learn for model evaluation, metrics calculation, and possibly for feature scaling or selection.

**GPU Acceleration**: If available, utilizing GPUs for faster training of deep learning models, which significantly speeds up experimentation and iteration.

**Development Environment**: Tools like Jupyter Notebooks or Google Colab for interactive development and experimentation with the code.

**Methodology:**

The methodology involves collecting a diverse dataset comprising authentic and deepfake videos covering various scenarios and facial expressions. Preprocessing steps include splitting videos into frames using OpenCV, detecting and aligning faces with algorithms like Haar cascades or MTCNN, and augmenting the dataset if necessary for improved generalization. Facial features such as landmarks and texture are extracted and prepared for input into the LSTM model. The model architecture is designed to capture temporal dependencies in these features, with experimentation on different LSTM configurations and additional layers for optimization.

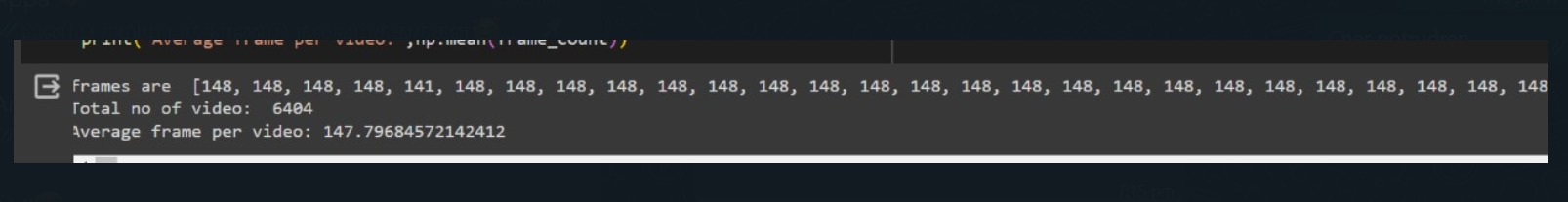
Following data splitting into training, validation, and test sets, the LSTM model is trained using techniques like early stopping and learning rate scheduling to prevent overfitting. Evaluation entails assessing model performance using metrics like accuracy, precision, recall, and F1-score, with analysis of misclassifications. Finally, the model is tested on unseen data for real-world performance and deployed for deepfake detection, ensuring scalability and efficiency in production environments.

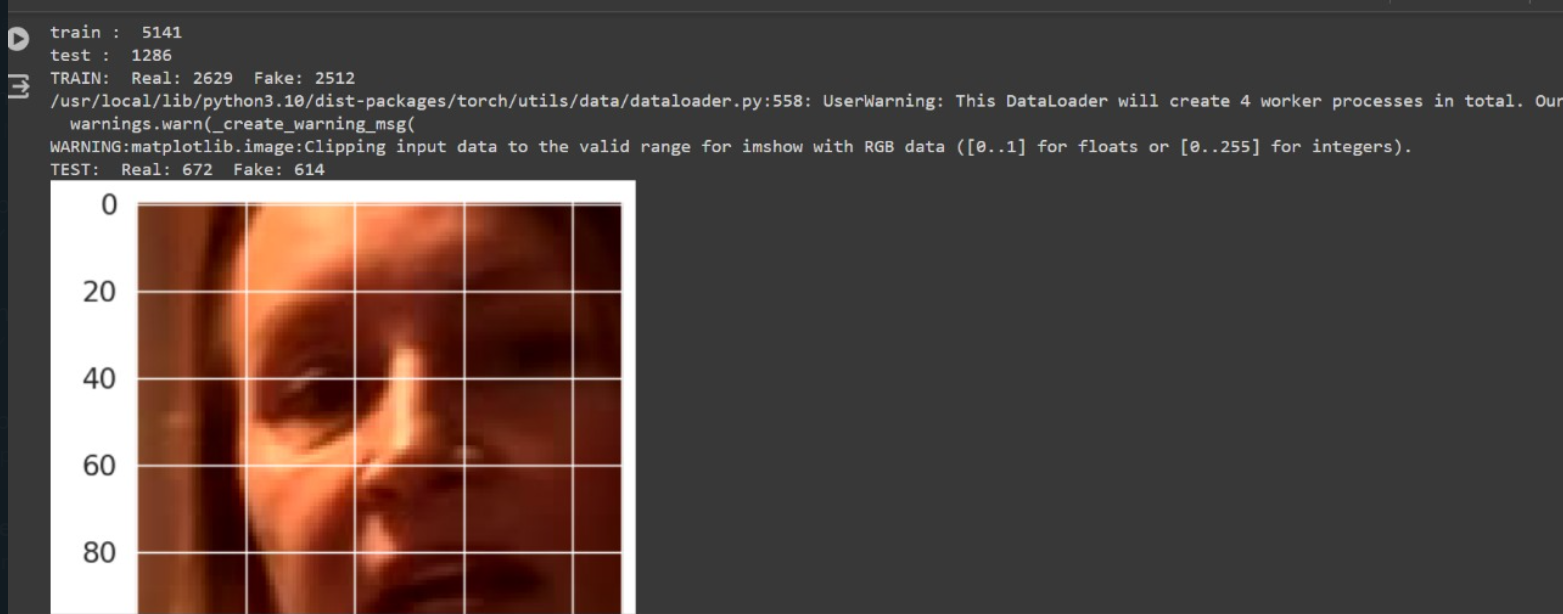
**Why LSTM?**

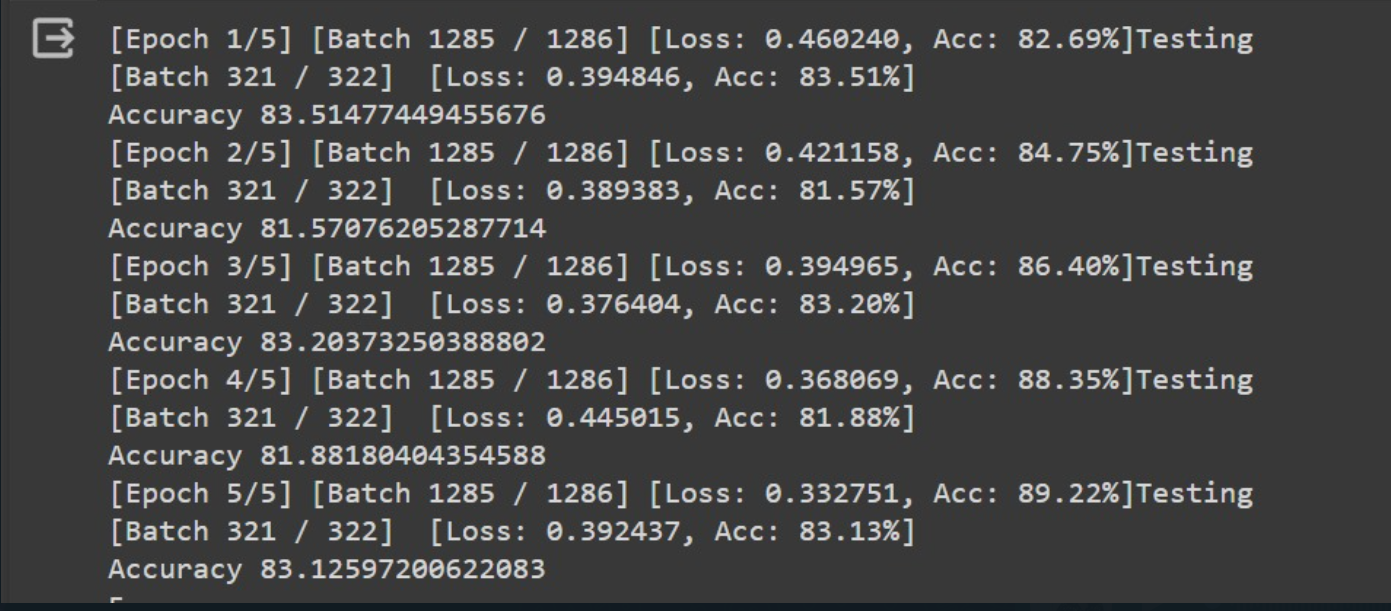
Deepfake videos often involve subtle temporal patterns and dynamics that are indicative of manipulation, such as inconsistent facial expressions or unnatural movements. LSTMs are designed to model sequential data and are capable of capturing long-range dependencies over time, making them effective for analyzing the temporal aspects of video frames.

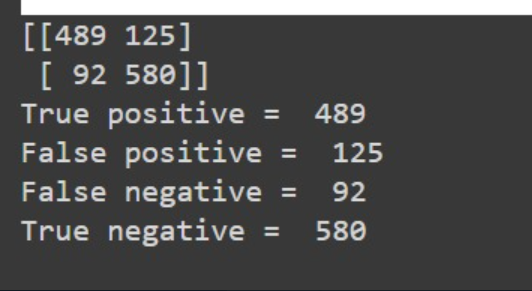
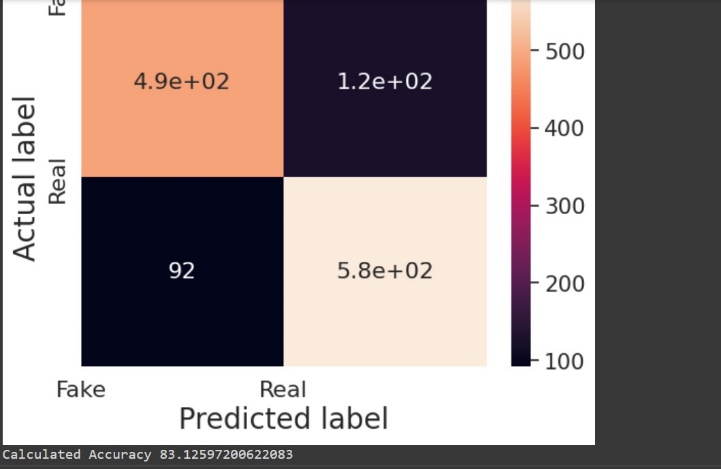
**Results:**

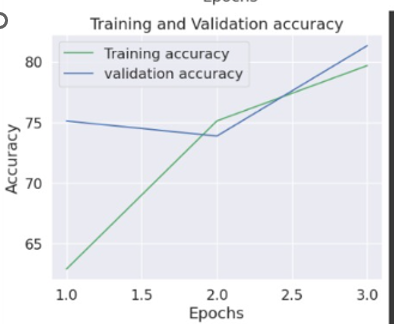
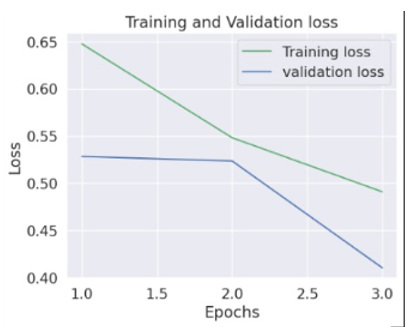
The accuracy that we achieved was 83.3 for the validation set.

****

****

****

****

****