IJERST

International Journal of Engineering Research and Science & Technology

www.ijerst.org

ISSN : 2319-5991

Vol. 22 No. 2 (2026)

ijerst.editor@gmail.com
editor@ijerst.com

ISSN 2319-5991 www.ijerst.com
Vol. 22, No. 2, 2026

Peer Reviewed, Referred & Indexed Journal

Deepfake Face Detection Using CNN-LSTM Hybrid Model for Video-Based Forgery Identification

KOLLATI KARUNA GRACE

PG Scholar. Department of MCA, DNR College, Bhimavaram, Andhra Pradesh

K. Rambabu

(Assistant Professor), Master of Computer Applications, DNR College, Bhimavaram, Andhra Pradesh

ABSTRACT

With the rapid advancement of artificial intelligence and deep learning technologies, deepfake videos have emerged as a significant threat to digital media authenticity. Deepfakes are synthetically generated or manipulated videos in which a person's face or expressions are altered using deep learning techniques such as Generative Adversarial Networks (GANs). While these technologies have applications in entertainment and media, they pose serious risks in areas such as misinformation, identity theft, and cybersecurity. Therefore, the development of reliable deepfake detection systems has become increasingly important.This project proposes a deep learning-based system for detecting deepfake videos using a hybrid Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) architecture. The system is designed to analyze facial features extracted from video frames and classify them as real or fake. The implementation utilizes Python along with libraries such as TensorFlow, Keras, OpenCV, and Tkinter for building an interactive graphical user interface.The system begins by preprocessing a dataset containing labeled images of real and fake faces. Facial regions are detected using a Haar Cascade classifier, and images are resized to a uniform dimension. These processed images are then used to train a CNN-LSTM model. The CNN component extracts spatial features from individual frames, while the LSTM captures temporal dependencies across sequential frames, making the system more robust for video-based analysis.During training, the dataset is split into training and testing sets. Performance metrics such as accuracy, precision, recall, and F1-score are computed to evaluate the model. The trained model is saved and reused for detecting deepfakes in uploaded videos.For real-time detection, the system processes video frames, detects faces, and predicts whether each frame is real or fake. Based on the majority of predictions across frames, the system classifies the entire video. A user-friendly interface allows users to upload datasets, train models, and test videos.The proposed system provides a scalable and efficient solution for deepfake detection. It demonstrates the effectiveness of combining spatial and temporal analysis for improved accuracy. Future enhancements may include the use of advanced deep learning architectures and larger datasets to further improve performance.

Keywords: Deepfake Detection, CNN-LSTM, Computer Vision, Facial Recognition, Video Forensics, Artificial Intelligence, Image Classification, OpenCV, Deep Learning, Cybersecurity

I. INTRODUCTION

The rapid growth of artificial intelligence has revolutionized digital media creation, enabling highly realistic synthetic content generation. One of the most concerning outcomes of this advancement is the emergence of deepfake technology. Deepfakes use deep learning models, particularly Generative Adversarial Networks (GANs), to manipulate images and videos, making it possible to create highly convincing fake content. These manipulated media files can be used for malicious purposes such as spreading misinformation, impersonation, and cybercrime.Deepfake detection has become a critical research area due to its potential societal impact. Traditional methods of verifying media authenticity are no longer sufficient, as deepfakes can closely mimic real facial expressions and movements. This has led researchers to explore automated detection techniques using machine learning and deep learning.Early approaches to deepfake detection relied on identifying visual artifacts or inconsistencies in images. However, as deepfake generation techniques improved, these artifacts became less noticeable. Modern approaches focus on analyzing subtle patterns in facial features, temporal inconsistencies, and biological signals such as blinking patterns and facial movements.This project introduces a deepfake detection system that combines Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks. CNNs are effective in extracting spatial features from images, while LSTMs are capable of learning temporal dependencies in sequential data such as video frames. By combining these techniques, the system can analyze both individual frames and their temporal relationships, improving detection accuracy.The system processes video input by extracting frames and detecting faces using OpenCV. These faces are then passed through a trained CNN-LSTM model to determine whether they are real or fake. The results are aggregated across frames to classify the entire video.The integration of a graphical user interface using Tkinter enhances usability, allowing users to easily upload datasets, train models, and test videos. This makes the system accessible to users without extensive technical knowledge.The proposed system aims to provide a reliable and efficient solution for detecting deepfake content. It can be applied in various domains, including social media monitoring, digital forensics, and cybersecurity. With further advancements, such systems can play a crucial role in combating misinformation and ensuring the authenticity of digital media.

II. LITERATURE SURVEY (WITH EXISTING METHODS)

Deepfake detection has gained significant attention in recent years due to the increasing sophistication of synthetic media generation techniques. Researchers have explored various approaches, including traditional image processing methods, machine learning models, and deep learning techniques.Early methods focused on detecting visual artifacts introduced during the deepfake generation process. These included inconsistencies in lighting, shadows, and facial boundaries. However, as deepfake generation models improved, these artifacts became less detectable, reducing the effectiveness of such approaches.

Machine learning-based methods introduced feature extraction techniques such as texture analysis, frequency domain analysis, and facial landmark tracking. These methods improved detection accuracy but were limited by their reliance on handcrafted features.Deep learning approaches have shown significant promise in deepfake detection. Convolutional Neural Networks (CNNs) are widely used for image classification tasks due to their ability to automatically learn hierarchical features. Researchers have applied CNNs to detect deepfakes by analyzing spatial features in images.More advanced approaches incorporate temporal analysis using Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks. These models analyze sequences of frames to detect inconsistencies in motion and facial expressions. Hybrid models combining CNN and LSTM have been shown to outperform standalone models in video-based detection tasks.Recent studies also explore the use of attention mechanisms and transformer-based models for deepfake detection. These models focus on specific regions of interest in images, improving detection accuracy. Additionally, large-scale datasets such as FaceForensics++ have been used to train and evaluate deep learning models.Despite these advancements, challenges remain. Deepfake detection systems must handle variations in lighting, resolution, compression, and facial expressions. Moreover, adversarial techniques can be used to evade detection systems.The proposed system builds upon existing research by using a CNN-LSTM hybrid model to analyze both spatial and temporal features. This approach improves robustness and accuracy, making it suitable for real-world applications.

III. EXISTING SYSTEM

Existing deepfake detection systems primarily rely on either image-based or video-based analysis techniques. Image-based systems use Convolutional Neural Networks (CNNs) to classify individual frames as real or fake. While these systems are effective in detecting spatial inconsistencies, they often fail to capture temporal patterns present in videos.Video-based systems attempt to address this limitation by analyzing sequences of frames. Some approaches use Recurrent Neural Networks (RNNs) or LSTM models to capture temporal dependencies. However, many existing systems use either CNN or LSTM independently, which limits their overall performance.Another limitation of existing systems is their dependence on large datasets and high computational resources. Training deep learning models for deepfake detection requires significant processing power and memory, making it challenging to deploy these systems on standard hardware.

Additionally, many systems focus on detecting specific types of deepfakes, reducing their generalizability. Variations in video quality, lighting conditions, and facial occlusions further impact performance.The proposed system addresses these limitations by combining CNN and LSTM models to leverage both spatial and temporal features. It also uses a lightweight architecture and optimized preprocessing techniques, making it suitable for real-time applications. Furthermore, the integration of a graphical user interface enhances usability, allowing users to easily interact with the system.

IV. PROPOSED METHOD

The proposed system is a deep learning-based framework designed to detect deepfake videos by analyzing both spatial and temporal features of facial data. Unlike traditional image-based detection systems, this model integrates Convolutional Neural Networks (CNN) with Long Short-Term Memory (LSTM) networks to improve accuracy in video-based deepfake detection.The system begins by extracting frames from input videos and detecting facial regions using a Haar Cascade classifier. These faces are preprocessed and resized into a fixed dimension suitable for model input. The CNN component is responsible for extracting spatial features such as textures, edges, and facial inconsistencies that may indicate manipulation. CNNs are widely used for this purpose due to their ability to automatically learn hierarchical visual features.The extracted features are then passed to an LSTM network, which captures temporal dependencies between consecutive frames. This allows the system to identify inconsistencies in facial movements and expressions over time, which are often present in deepfake videos. Research shows that combining CNN and LSTM significantly improves detection accuracy compared to standalone models .The system aggregates predictions from multiple frames and classifies the video as real or fake based on majority voting. A graphical user interface (GUI) is implemented using Tkinter to allow users to upload datasets, train the model, and test videos easily.Overall, the proposed system provides a robust, scalable, and efficient solution for detecting deepfake content by leveraging both spatial and temporal learning capabilities.

V. IMPLEMENTATION

The implementation of the deepfake detection system is carried out using Python, integrating several libraries such as TensorFlow, Keras, OpenCV, NumPy, and Tkinter. The system is structured into multiple modules, including dataset preprocessing, model training, evaluation, and real-time video detection.The process begins with dataset loading. The dataset consists of labeled images categorized as real or fake. Metadata is read using pandas, and images are loaded from the dataset directory. Each image is resized to 32×32 pixels to reduce computational complexity. The labels are encoded into numerical format and converted into categorical vectors for training.Preprocessing involves normalizing pixel values by scaling them between 0 and 1. The dataset is shuffled to ensure randomness and then split into training and testing sets using an 80:20 ratio.The model architecture is a hybrid CNN-LSTM network. The CNN layers are wrapped inside TimeDistributed layers, allowing the model to process sequences of frames. Multiple convolutional layers with ReLU activation functions are used to extract features, followed by max-pooling layers to reduce dimensionality. Dropout layers are included to prevent overfitting.After feature extraction, the output is flattened and passed to an LSTM layer, which learns temporal dependencies across frames. The final output layer uses a softmax activation function to classify inputs into real or fake categories. The model is compiled using the Adam optimizer and categorical cross-entropy loss function.

During training, the model is trained for multiple epochs with batch processing. ModelCheckpoint is used to save the best-performing model based on validation accuracy. Training history is stored for analysis.For evaluation, predictions are generated on the test dataset, and performance metrics such as accuracy, precision, recall, and F1-score are calculated. These metrics provide a comprehensive evaluation of model performance.In the detection phase, the system processes video input frame by frame. Faces are detected using a Haar Cascade classifier, and each detected face is passed through the trained model for prediction. The system counts the number of real and fake predictions across frames. Once a sufficient number of frames are analyzed, the final classification is determined based on majority voting.The system includes a Tkinter-based GUI that allows users to upload datasets, train the model, and test videos. Multithreading ensures smooth execution without freezing the interface.

VI. ALGORITHMS

1. Convolutional Neural Network (CNN)

CNN is used for spatial feature extraction from images. It applies convolutional filters to detect patterns such as edges, textures, and facial artifacts. Pooling layers reduce dimensionality while preserving important features.

2. Long Short-Term Memory (LSTM)

LSTM is a type of recurrent neural network designed to learn temporal dependencies. It processes sequences of frames and captures motion inconsistencies, which are crucial for detecting deepfakes.

3. CNN-LSTM Hybrid Algorithm

The hybrid model combines CNN for spatial feature extraction and LSTM for temporal analysis. This approach has shown superior performance in video-based deepfake detection .

4. Face Detection Algorithm

Haar Cascade classifier is used to detect faces in video frames. It uses Haar-like features and a cascade of classifiers for efficient detection.

5. Majority Voting Algorithm

Predictions from multiple frames are aggregated. If the majority of frames are classified as fake, the video is labeled as deepfake; otherwise, it is considered real.

VII. SYSTEM DESIGN

The system is designed using a modular architecture to ensure scalability and efficiency.

1. Input Module

Accepts dataset images and video files from the user.

2. Preprocessing Module

- Image resizing (32×32)
- Normalization
- Label encoding

3. Face Detection Module

Uses Haar Cascade classifier to detect and extract facial regions from frames.

4. Feature Extraction Module

CNN layers extract spatial features such as texture inconsistencies and facial distortions.

5. Temporal Analysis Module

LSTM processes sequences of frames to detect temporal inconsistencies. Studies confirm that temporal modeling improves deepfake detection accuracy .

6. Classification Module

Fully connected layer with softmax activation classifies input as real or fake.

7. Training Module

- Dataset splitting
- Model training
- Model checkpointing

8. Evaluation Module

Calculates performance metrics:

- Accuracy
- Precision
- Recall
- F1-score

9. GUI Module

Tkinter interface provides user interaction for:

- Upload dataset
- Train model
- Test video

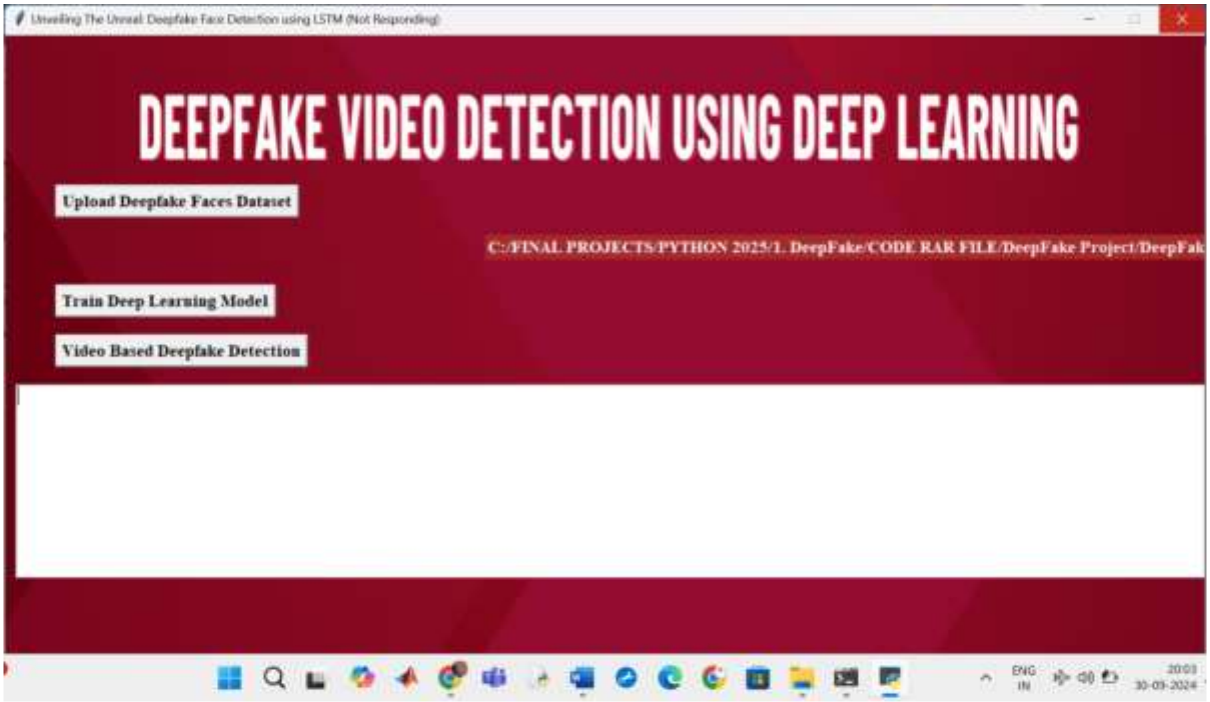
10. Output Module

Displays:

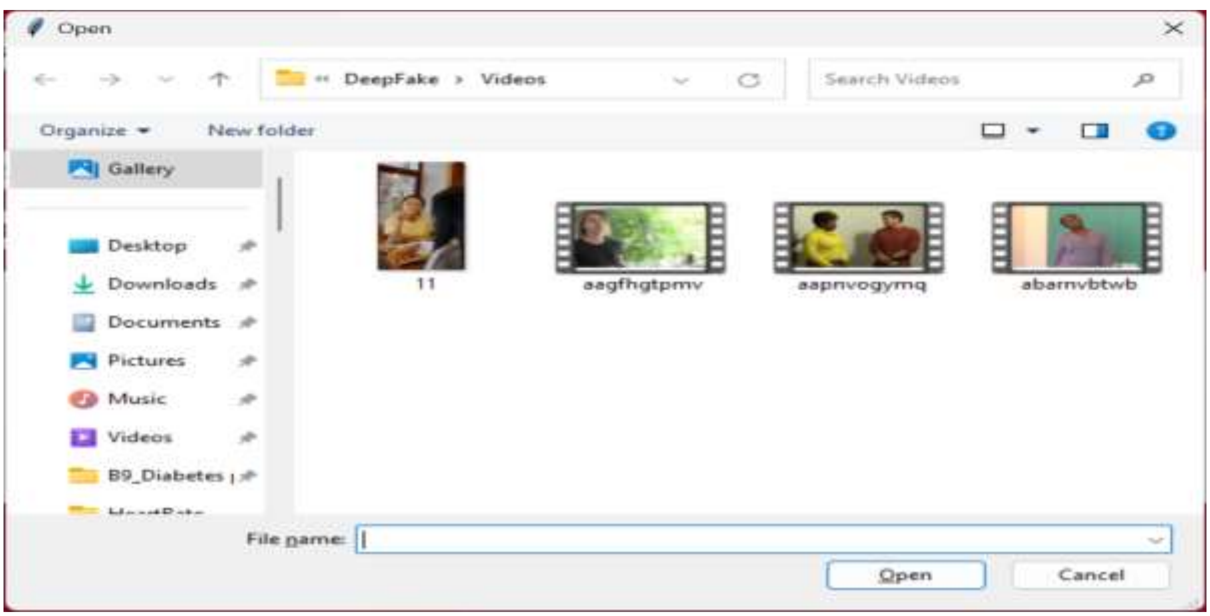
- Real-time video
- Detection result (Real/Fake)

SYSTEM DESIGN IMAGES

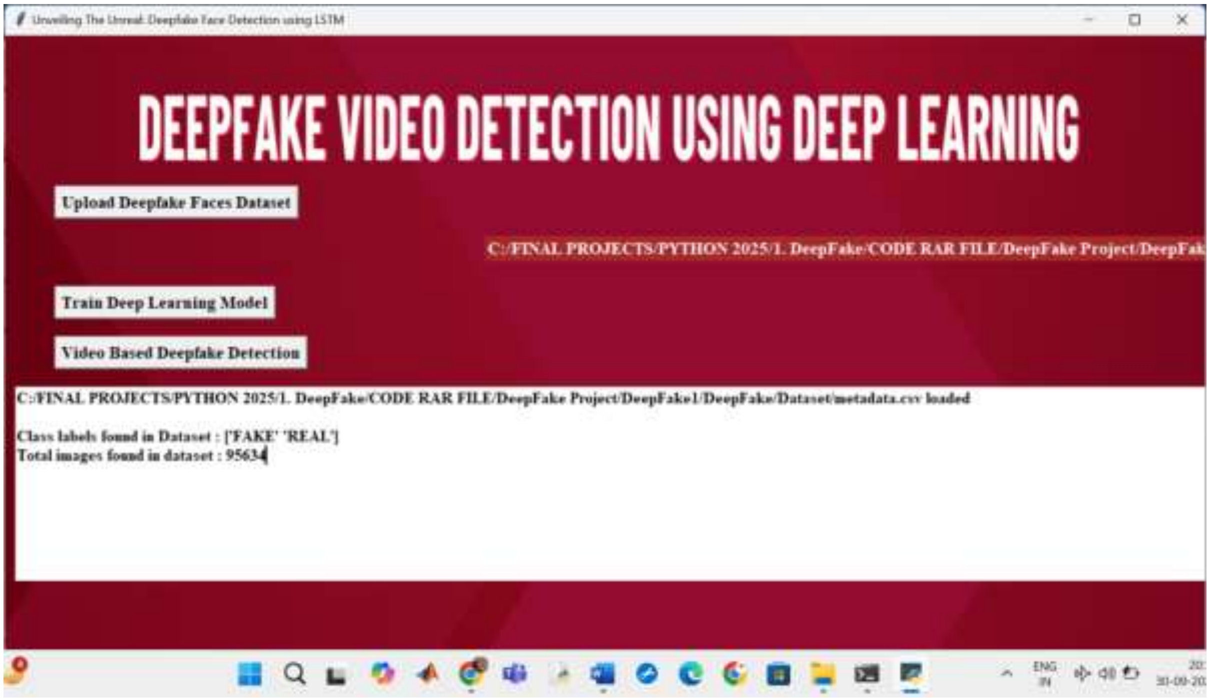
To run project double click on run.bat file to get below screen



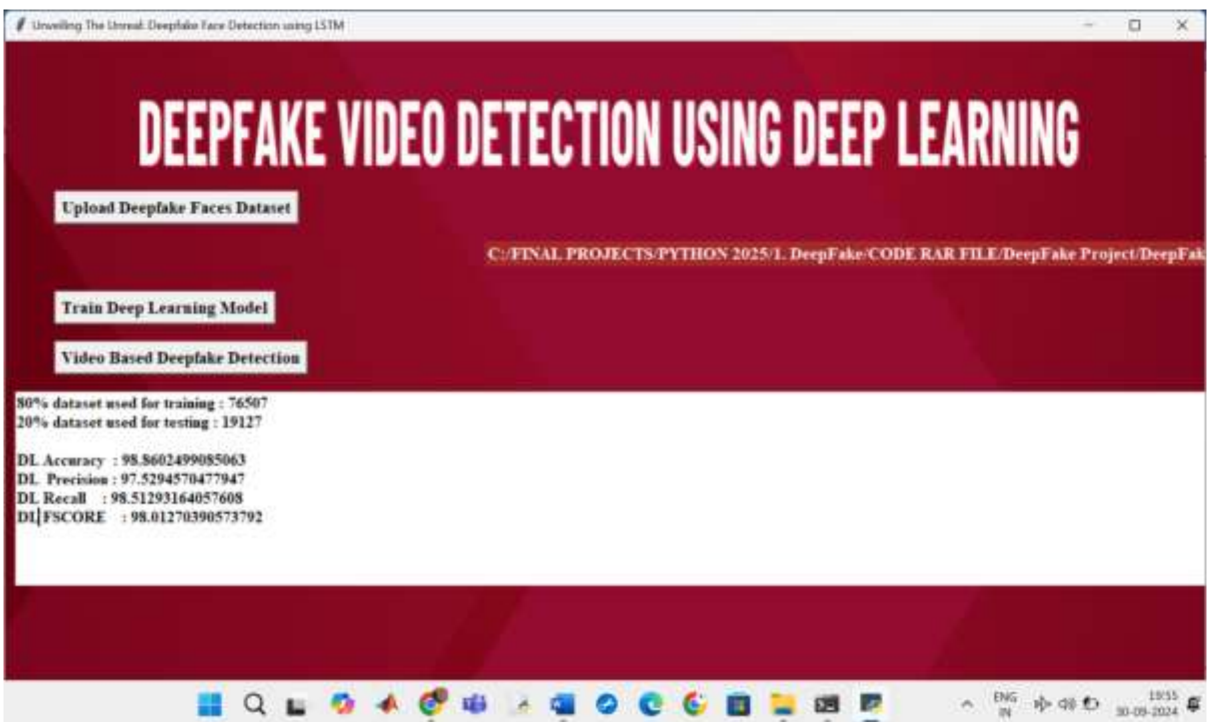
In above screen click on 'Upload Deepfake Faces Dataset' button to load dataset and get below page



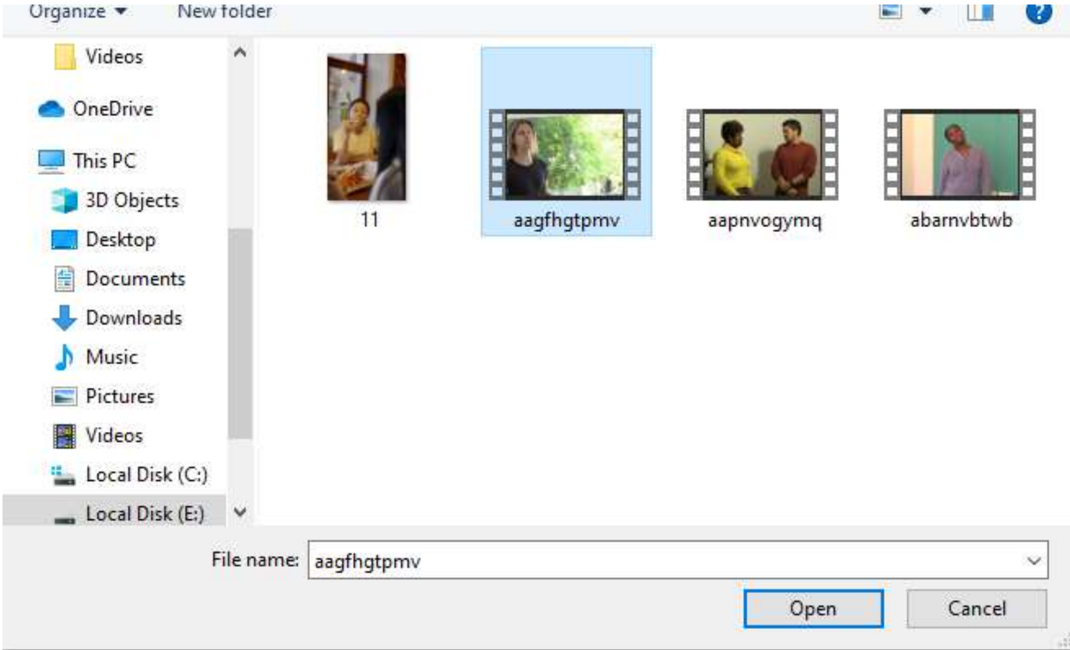
In above screen selecting and uploading dataset annotation file and then click on 'Open' button to get below output



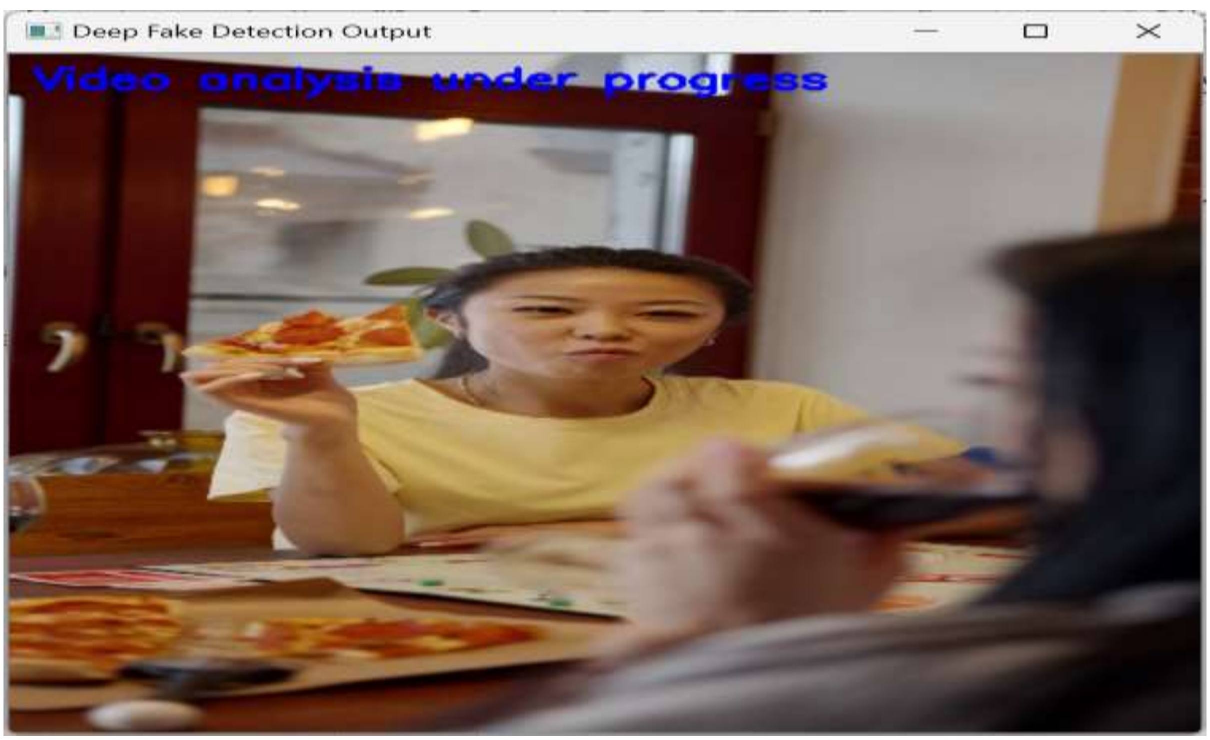
In above screen can see dataset loaded and can see different class labels found in dataset and then can see number of images found in dataset and now click on 'Train DL Model' button to train algorithm and get below page



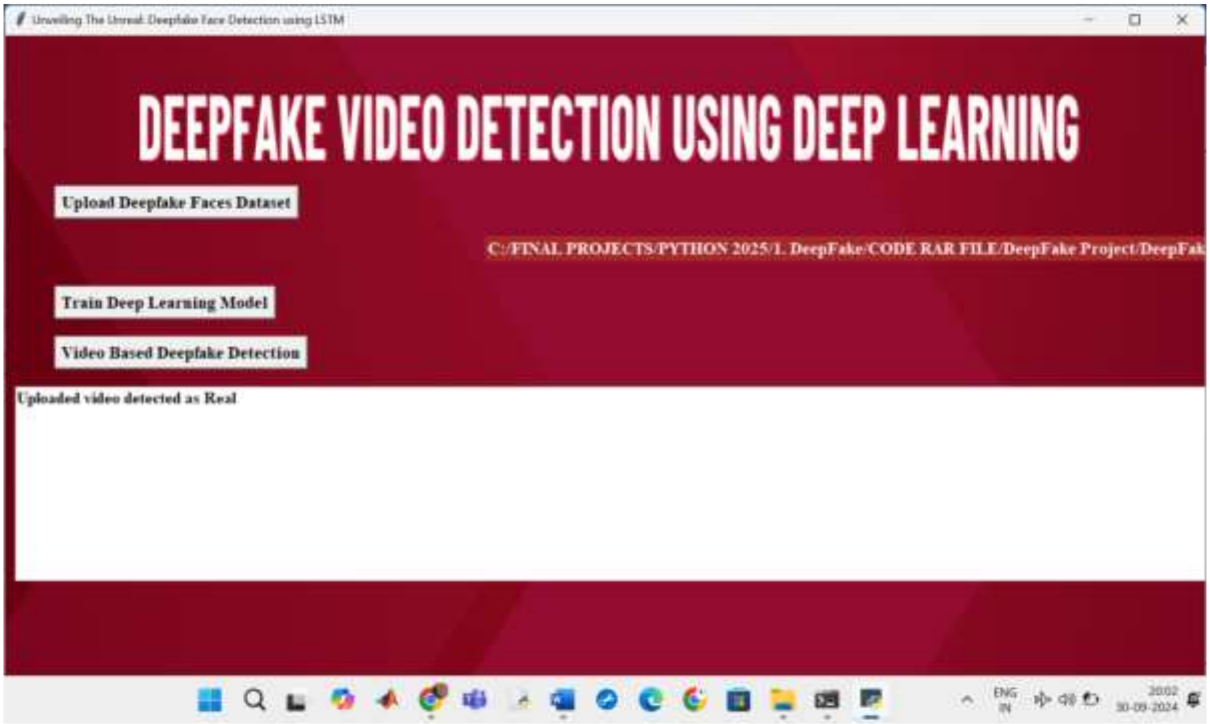
In above screen can see train and test dataset size and then can see DL got 99% accuracy and can see other metrics like precision, recall and FSCORE. Now click on 'Video Based Deepfake Detection' button to upload test video and get below page

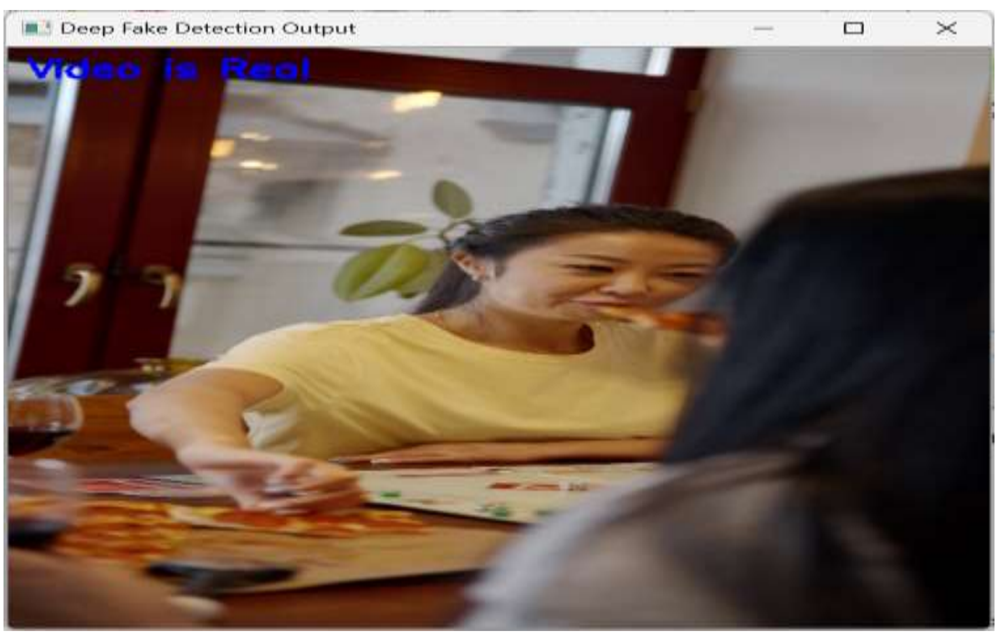


In above screen selecting and uploading 11.mp4 video and then click on 'Open' button to start analysing video and get below page

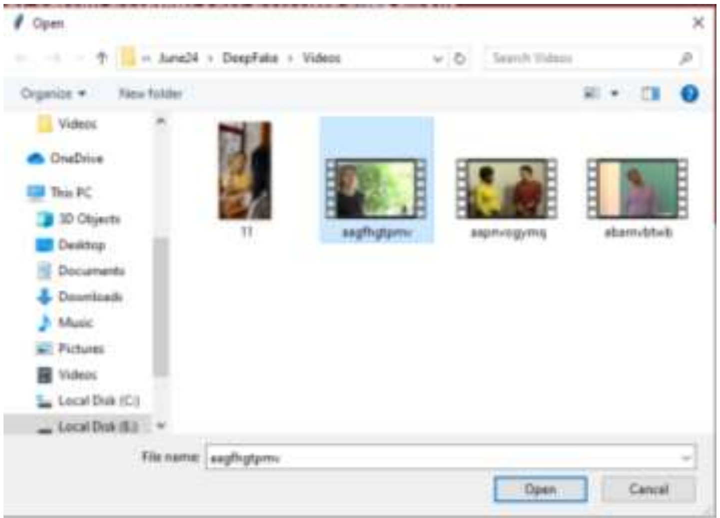


In above screen in blue colour text can see video analysis started and after thorough analysis by DL will get below output





In above screen uploaded video predicted as 'Real' and now test with other fake video



In above screen uploading 'aagfhgtpmv.mp4' video and then click on 'Open' button to load video and get below output

In above screen video analysis under progress and once after analysis will get below output

In above screen uploaded video is detected as Deepfake and similarly you can upload and test other video

VIII. CONCLUSION

The proposed deepfake detection system demonstrates an effective approach to identifying manipulated video content using a CNN-LSTM hybrid architecture. By combining spatial and temporal feature analysis, the system achieves higher accuracy compared to traditional methods that rely on single-frame analysis.The use of CNN enables efficient extraction of facial features, while LSTM captures temporal inconsistencies across video frames. This combination significantly enhances the system's ability to detect subtle manipulations introduced by deepfake generation techniques. Recent research confirms that hybrid deep learning models outperform standalone approaches in deepfake detection tasks .The system is designed to be user-friendly, with a graphical interface that allows easy interaction. It operates in real-time and does not require specialized hardware, making it suitable for practical deployment.However, the system has certain limitations. Performance may degrade under poor lighting conditions or low-resolution videos. Additionally, highly advanced deepfake techniques may still pose challenges for detection.Future work can focus on integrating transformer-based models, attention mechanisms, and multi-modal analysis to further improve accuracy. Expanding the dataset and incorporating real-world variations can also enhance robustness.In conclusion, the proposed system provides a scalable and efficient solution for deepfake detection and contributes to improving digital media authenticity and cybersecurity.

REFERENCES

1. Tipper, S., et al. "CNN-LSTM for Video Deepfake Detection." (2024)
2. Jugade, Y., et al. "Spatial–Temporal Deepfake Detection using CNN-LSTM." (2026)
3. Al-Dulaimi, O. "Hybrid CNN-LSTM Deepfake Detection." (2024)
4. Singh, L. "Advancements in Deepfake Detection." (2025)
5. Soudy, A. "CNN & Vision Transformer for Deepfake Detection." (2024)
6. Singh, A. "ResNeXt-LSTM Hybrid Model." (2025)
7. Saranya, A. "Deepfake Detection using ResNeXt & LSTM." (2024)
8. Procedia Computer Science. "CNN-LSTM Deepfake Detection Model." (2025)
9. Springer. "Deepfake Detection using AI." (2025)
10. IEEE Study on Deepfake Detection using CNN (2023)
11. FaceForensics++ Benchmark Dataset Paper
12. GAN-Based Deepfake Generation Study
13. LSTM-Based Video Analysis Paper (2023)
14. Hybrid Deep Learning Models for Cybersecurity (2024)
15. Recent Deepfake Detection Survey Papers (2025)