

DEEPPFAKE DETECTION

(MAJOR PROJECT)

Prajwal Rajeev Hampannavar

Bachelor of Engineering in Computer Science Engineering (CGPA: 8.12)

KLE Dr. M. S. Sheshgiri College of Engineering and Technology





Table of Contents

<u>Introduction</u>	3
<u>Overview</u>	4
<u>Problem Statement</u>	6
<u>Methodology</u>	7
<u>Dataflow Diagram</u>	8
<u>Project workflow</u>	9
<u>Results</u>	11
<u>Conclusion</u>	15

Introduction

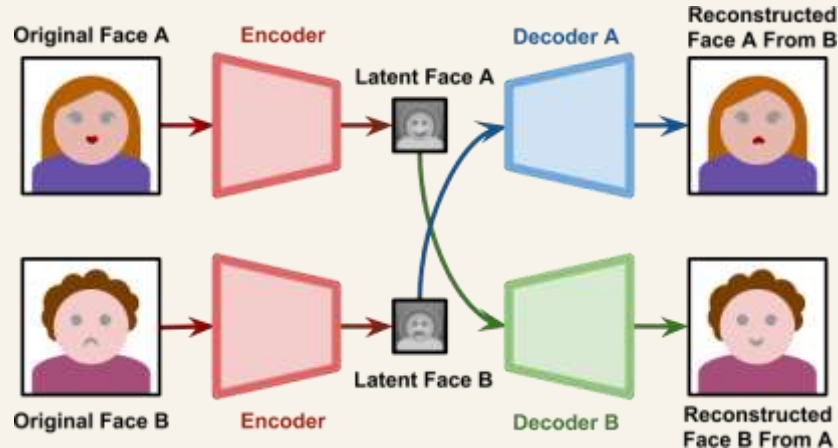
- Deep fake is a technique for human image synthesis based on artificial intelligence.
- Deep fakes are created by combing and superimposing existing images and videos onto source images or videos using a deep learning technique known as generative adversarial network.
- There are several cases where these realistic face swapped deepfakes are used to create political distress, fake terrorism events, revenge porn, blackmail peoples are easily visualized.
- It becomes veery important to spot the difference between the deepfake and original video. We are using AI to fight AI Deepfakes are related using tools like Face App and Face Swap, which using pre-trained neural networks like GAN or Auto encoders for these deepfakes creation.



How is Deepfake Created?

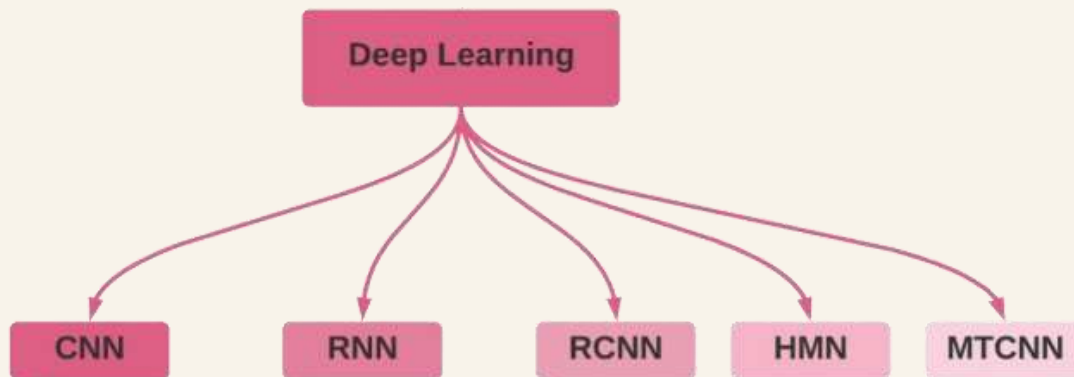
GAN(Generative Adversarial Networks)

A Generative Adversarial Network (GAN) consists of a generator (G) and a discriminator (D). During training, real images (x) from a dataset (X) are used, with the generator aiming to create realistic images ($G(z)$) from noise signals (z) while the discriminator distinguishes between real and generated images. The discriminator is trained to maximize the probability of correctly labeling both real and fake images, while the generator is trained to minimize the probability of its outputs being classified as fake.



How DeepLearning can help detect Deepfakes?

- There are some DeepLearning model which help to detect deepfakes.
- CNN and RNN are majorly used models as they are efficient and accurate.



Problem Statement

Developing effective deepfake detection software faces challenges due to the evolving sophistication of deepfake generation techniques. Current solutions often struggle to keep pace with the rapid advancements in deepfake technology, leading to a pressing need for more robust and adaptable detection algorithms. Additionally, the balance between accuracy and real-time processing speed remains a critical concern, as quick and reliable identification of deepfakes is crucial in various applications, including media forensics and online content moderation. Addressing these challenges requires innovative approaches to enhance detection accuracy, efficiency, and resilience against emerging deepfake strategies.



Methodologies

This method detects such artifacts by comparing the generated face areas and their surrounding regions by splitting the video into frames and extracting the features with a ResNext Convolutional Neural Network (CNN) and using the Recurrent Neural Network (RNN) with Long Short Term Memory(LSTM) capture the temporal inconsistencies between frames introduced by GAN during the reconstruction of the DeepFake.

Parameters Identified

1. Blinking of eyes
2. Wrinkles on face
3. Face angle
4. Facial Expressions



Dataflow Diagram

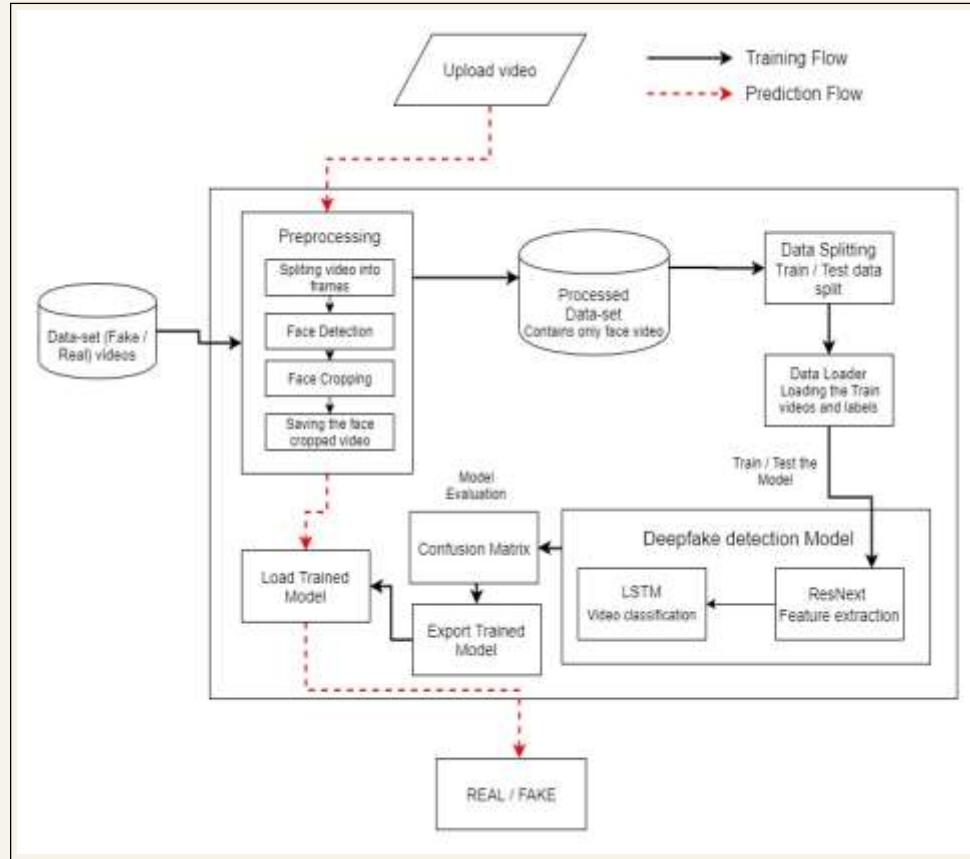



Figure: System Architecture

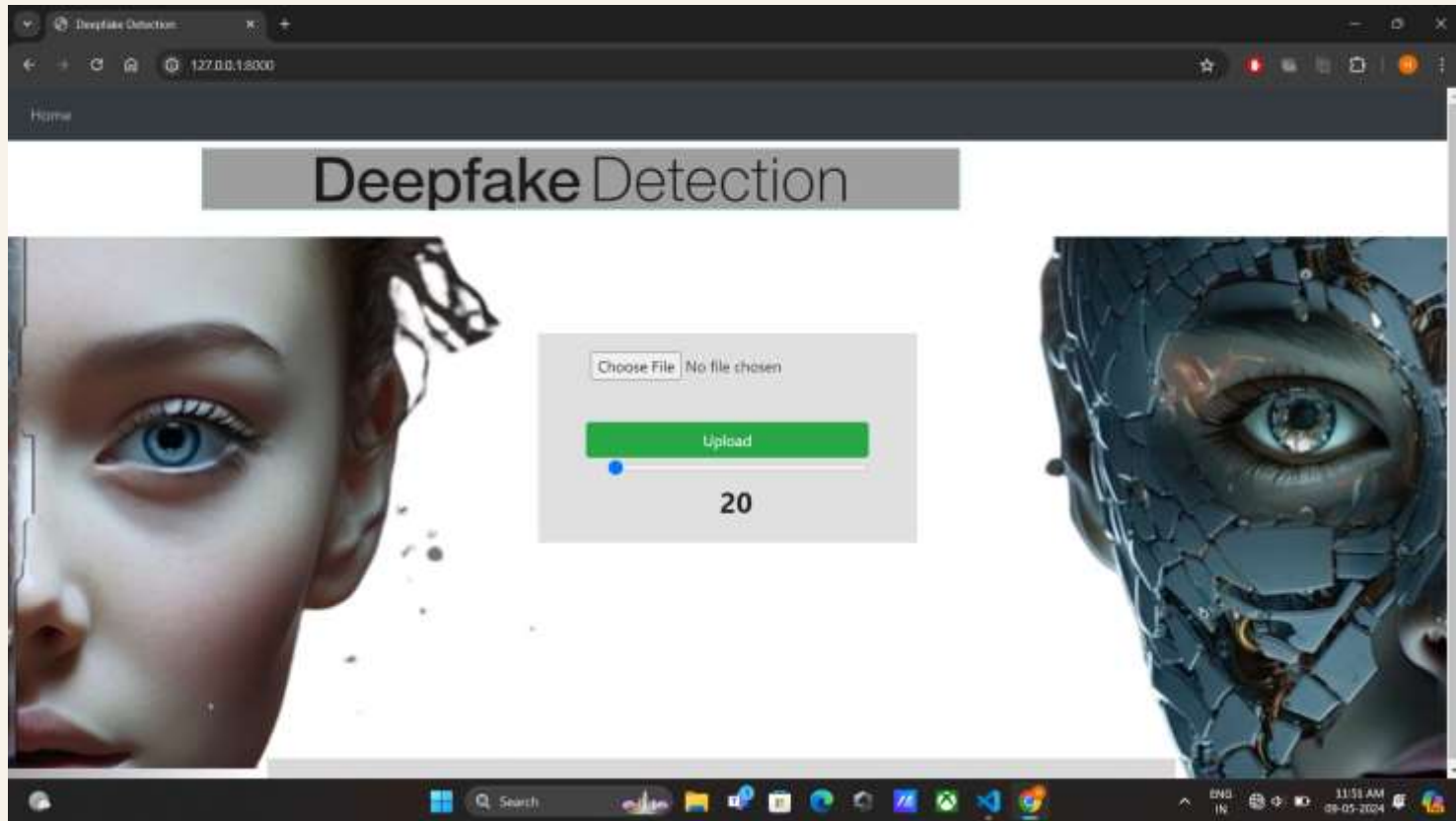
Project Workflow

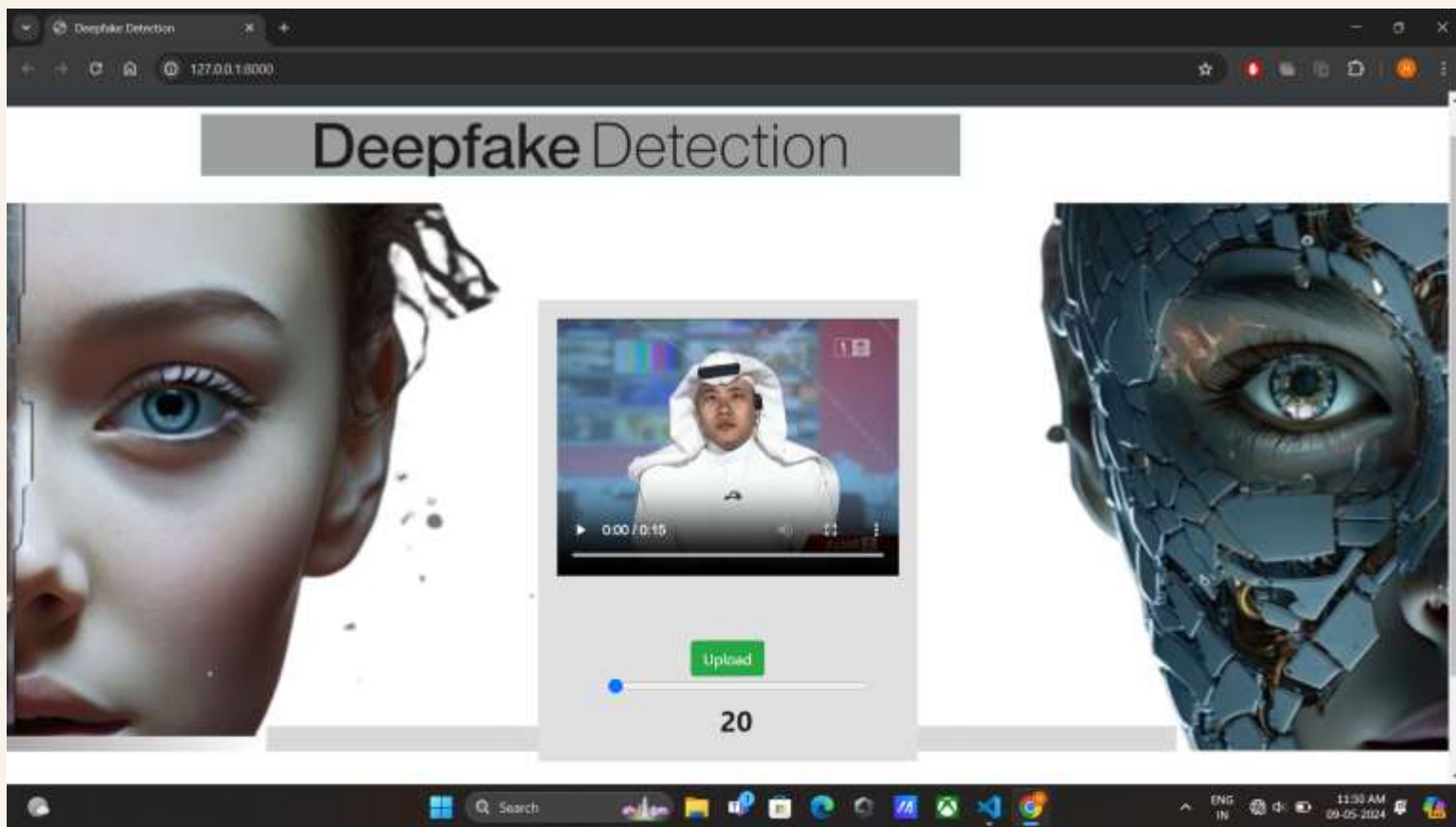
- **Data Collection and Preprocessing:** We start by collecting a dataset of videos. Using OpenCV, we extract frames from each video. For each frame, we use the `face_recognition` library to detect and crop faces, creating a new dataset that consists only of face-cropped frames.`
- **Model Training:** The dataset is split into 70% for training and 30% for testing. We use Google Colab for training due to its GPU support. During training, we pass the face-cropped frames through a pretrained ResNeXt model to extract high-level features. These features are then organized into sequences, which are used to train an LSTM model to capture temporal dependencies. The trained model is then saved and integrated into a Django project.
- **Evaluation:** To assess the model's performance, we use a confusion matrix during training and evaluation phases, analyzing metrics like accuracy, precision, recall, and F1-score. Based on these evaluations, we fine-tune the model and set an appropriate threshold for classification.

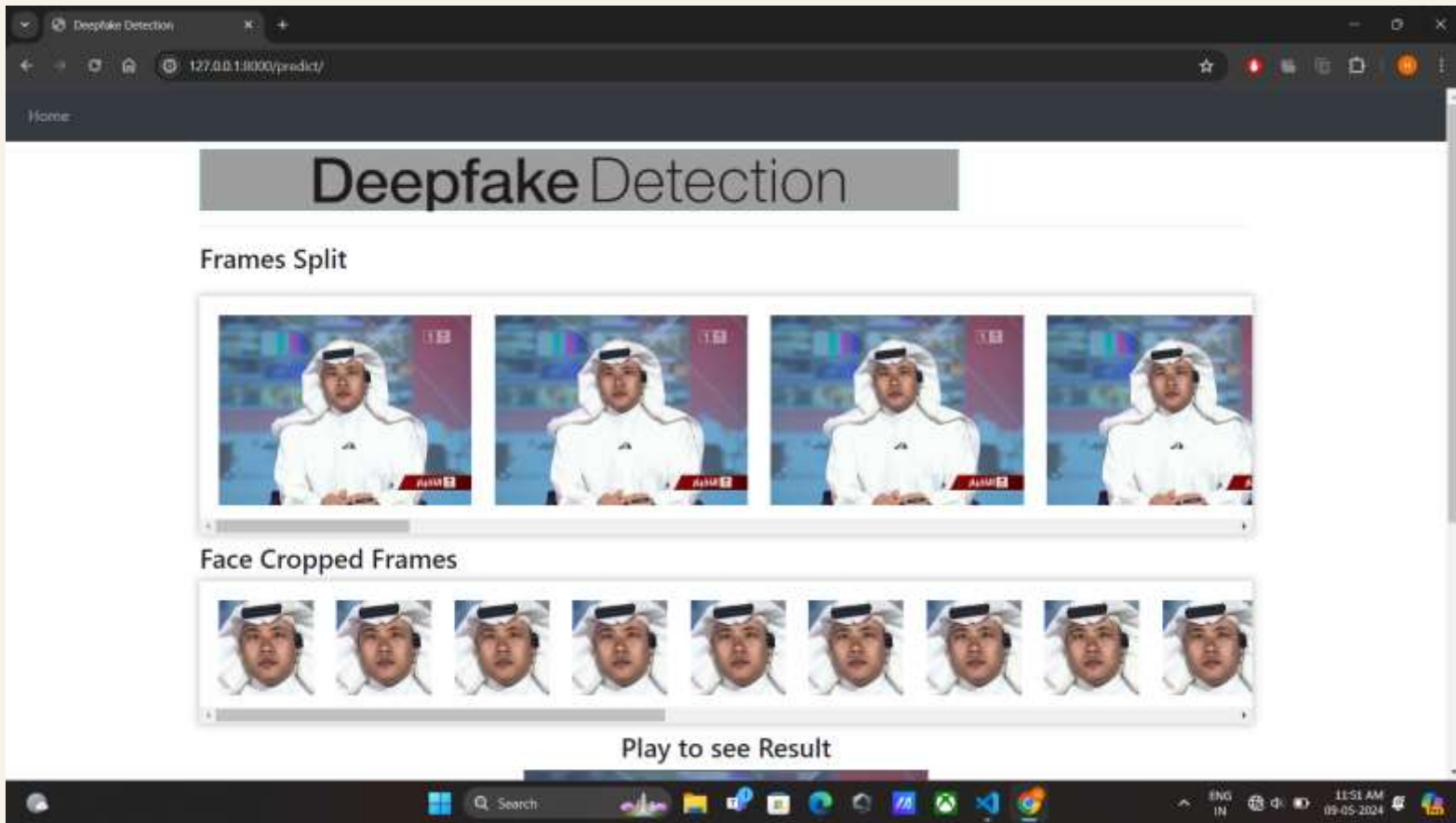


- 
- **Prediction Process:** When a user uploads a video through the web application, we divide the video into frames using OpenCV. We detect and crop faces in each frame and extract features using the pretrained ResNeXt model. These features form a sequence that is passed through the trained LSTM model to obtain raw prediction scores (logits). We then apply the SoftMax function to convert these logits into probabilities.
 - **Classification:** The probabilities are compared to a predefined threshold. If the probability of the video being real is above the threshold, it is classified as real; otherwise, it is classified as fake. The result, along with the confidence level, is displayed to the user.

Results







Deepfake Detection

127.0.0.1:8000/predict/

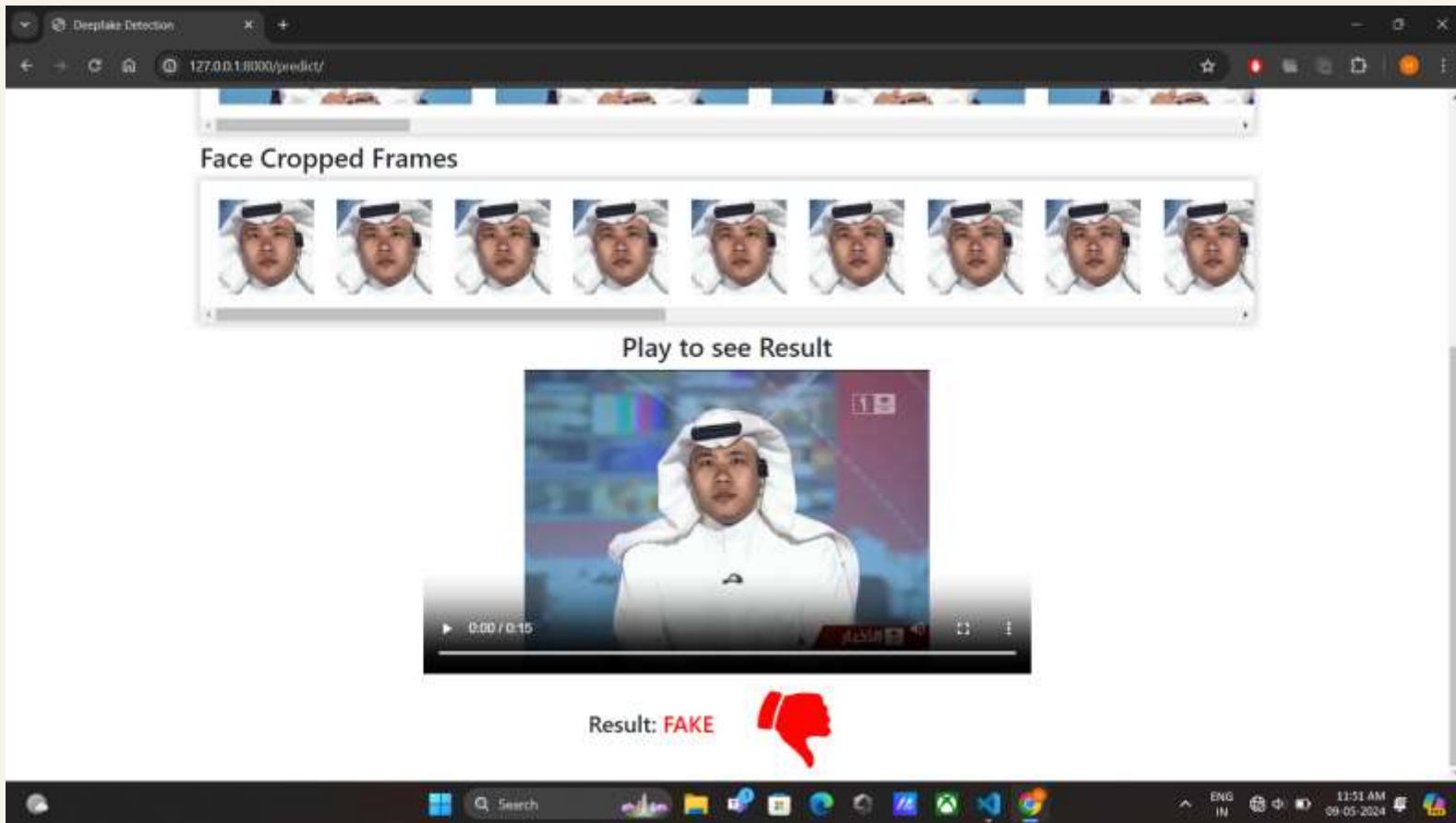
Face Cropped Frames

Play to see Result

0:00 / 0:15

Result: **FAKE**

ENG IN 11:51 AM 09-05-2024



Conclusion

With the increase in the use of Deepfake videos around the world, it is very necessary to detect such videos before they could cause some sort of harm. Various Machine Learning and Deep Learning-based techniques along with the different features are used to classify the videos as fake or real. Among all the different techniques used, the one that uses CNN and LSTM has proved to be more accurate in the classification of the videos. Here, various datasets that contain several real and fake videos have been used for the classification. We implemented the model by using pre-trained ResNext CNNmodel to extract the frame level features and LSTM for temporal sequence processing to spot the changes between frame.





Thank You