Real-Time Deepfake Detection for Video Streaming Platforms

Project Domain / Category

AI Web Application / Machine Learning

Abstract / Introduction

With the rise of deepfake technology, it has become increasingly easy to manipulate videos, posing risks to media authenticity, personal privacy, and public trust. In live streaming and video conferencing platforms, deepfake videos can be used for impersonation and misinformation, creating significant challenges for ensuring authenticity.

The objective of this project is to develop a real-time deepfake detection system that can analyze video streams and identify manipulated content. The application will detect deepfakes by analyzing each video frame using advanced machine learning models. Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks will be employed to detect inconsistencies across frames, ensuring the detection is fast and reliable.

The system will be designed for use on streaming platforms, helping institutions, organizations, and individuals identify manipulated content during live sessions. This project will allow students to explore the latest deep learning techniques in multimedia security while focusing on a practical and impactful solution.

Functional Requirements:

1. **User Authentication:**
   - Users will register and log in to access the detection system.
   - Admin-level authentication for platform owners.

2. **Real-Time Video Input:**
   - The system will support live video streaming or video conferencing platforms (e.g., Zoom or WebRTC-based platforms).
   - Users can also upload recorded videos for analysis.

3. **Deepfake Detection:**
   - Use CNN-LSTM-based models to process video frames in real-time and detect deepfake content.
   - Incorporate facial landmarks analysis and detect inconsistencies in eye movement, lip-syncing, or lighting patterns.

4. **Visualization of Results:**
   - The system will highlight detected frames with a confidence score indicating the likelihood of the video being a deepfake.

- Alerts will be provided during live streams when a deepfake is detected.

5. **User Feedback:**

   Users can provide feedback on detection accuracy to improve future models.

6. **Error Handling:**

   Proper messages will be shown in case of processing delays, network issues, or unsupported video formats.

7. **Privacy and Data Handling:**

   - The system will process video data locally or ensure end-to-end encryption to maintain user privacy.
   - No video data will be stored without user consent.

Tools:

Development Environments / IDEs:

- **Backend Development:** Python with Flask or Django.
- **Frontend Development:** HTML, CSS, and JavaScript (or React.js for a dynamic interface).

Libraries and Frameworks:

1. **Deep Learning Models:** TensorFlow or PyTorch for building and training CNN-LSTM-based models.
2. **Video Processing:** OpenCV for extracting and processing video frames.
3. **Facial Detection:** Dlib or MTCNN for detecting facial landmarks.
4. **Model Deployment:** ONNX or TensorFlow Lite for deploying models on the web for real-time performance.
5. **Security Tools:** JWT or OAuth for user authentication.

Hardware Requirements:

- **GPU Support:** A system with GPU acceleration is recommended for real-time video processing or can use cloud based GPUs (e.g, Google Colab, Google Cloud Platform).
- **Web Server:** Use cloud-based servers (e.g., AWS, Azure) to deploy the application.

Supervisor:

Name: Sonia Salman
Email ID: Sonia.salman@vu.edu.pk
Skype ID: sonia_salman