**DeepFake Detection - Simplified!  
============================**

**Project Overview**

The DeepFake Detection Web Application is a powerful tool designed to identify whether a video is real or fake using advanced deep learning technique. The system integrates a trained model with a user-friendly interface, making it possible to detect DeepFake videos with confidence scores.

**Key Features**

* Utilizes Deep Learning techniques, including ResNext and LSTM.
* Provides confidence ratios for video authenticity.
* Backend built using Flask.
* Frontend built using ReactJS.

**Project Structure**

DeepFake-Detection

|--- DeepFake\_Detection

|--- Implementation Video

|--- Project-Setup.txt

|--- Requirements.txt

**Directory Explanation**

1. **DeepFake\_Detection:** The root folder containing the primary project files.
2. **Implementation Video:** Demonstrates the project’s functionality.
3. **Project-Setup.txt:** Contains all necessary steps to run the project.
4. **Requirements.txt:** Lists the Python libraries required for the project.

**Important Notes**

1. In the root folder, create a new directory called **"Uploaded\_Files"** to store user-uploaded videos.
2. Create another directory called **"model"** in the root folder and place the trained model file in it.

**Model Performance**

The DeepFake Detection model was trained and tested with notable accuracy.

**Key Metrics:**

* **Model Accuracy:** High accuracy achieved during testing.
* **Training and Validation Accuracy:** [Training and Validation Accuracy Graph]
* **Training and Validation Loss:** [Training and Validation Loss Graph]
* **Confusion Matrix:** [Confusion Matrix Visualization]

**Results Summary**

* The system effectively identifies DeepFake videos and outputs a confidence ratio indicating the likelihood of a video being fake or real.

**Advantages and Disadvantages of Models**

**ResNext**

**=======**

**Advantages:**

* **Efficient Feature Extraction:** Handles complex patterns in video frames effectively.
* **Improved Accuracy:** Outperforms traditional convolutional models.
* **Parallel Training:** Easier to optimize and train.

**Disadvantages:**

* **Computationally Intensive:** Requires high processing power.
* **Model Complexity:** Challenging to fine-tune and deploy on edge devices.

**LSTM (Long Short-Term Memory)**

**===========================**

**Advantages:**

* **Temporal Pattern Detection:** Ideal for capturing sequential dependencies in video frames.
* **Memory Retention:** Effective at handling long sequences.

**Disadvantages:**

* **Slow Training:** Computationally expensive due to sequential processing.
* **Vanishing Gradient Issues:** Despite improvements, still susceptible to learning inefficiencies.

**Conclusion**

This project demonstrates the effective use of deep learning techniques to combat the spread of DeepFake videos. The user-friendly interface, coupled with a robust backend, makes it a valuable tool for video verification.