

FakeProof : A Dual Solution for Deepfake Creation and Detection

Project Title: Deepfake Generator and Detector Platform

Deepfake Video Detection Using CNNs and RNNs -

A deepfake generator combined with a deepfake detector in one web interface is the proposed project. The growth of AI-generated media has propelled deepfakes as one of the pressing issues that threaten personal privacy, security, and the authenticity of information. Our platform addresses creating and detecting deepfakes. The platform will allow the creation of AI-generated video and audio content through the deepfake generator, and testing for authenticity of such media through the deepfake detector. With this approach, we make use of Vultr's GPU-accelerated cloud services for an efficient, scalable deepfake generator and detector. All these will be integrated into a web page, so it is easy to dig in and experiment with deepfakes technologies; and at the same time, it contributes to furthering awareness of the ethical implications. Major users include media professionals and their content authors, security experts, and educators who need tools helping users to understand and handle deepfake technology.

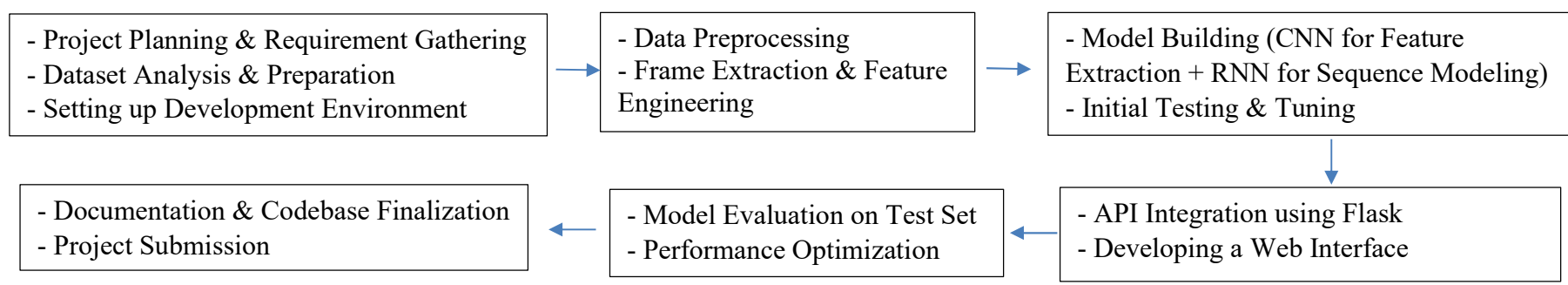
Detailed Project Plan-

Project Objectives -

The main objective of this project is to build a robust system that can detect DeepFake videos using a combination of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). The project focuses on:

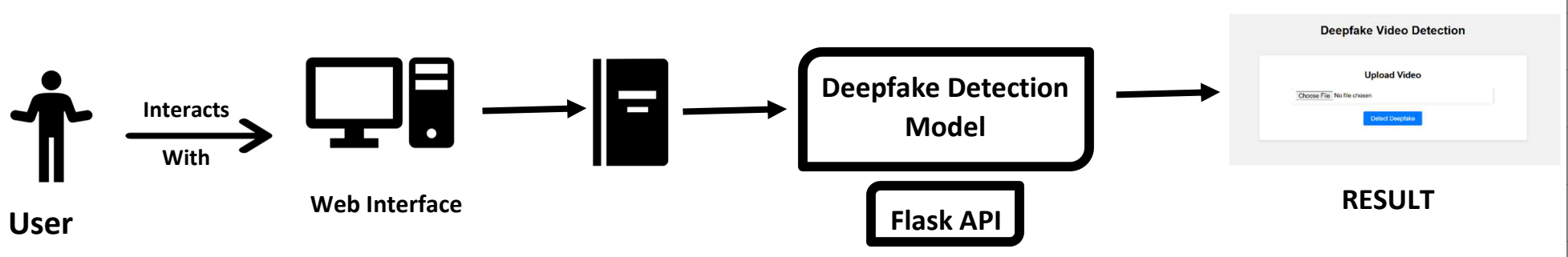
- Detecting manipulated or forged faces in videos.
- Classifying videos as "Real" or "Fake" based on video frames.
- Utilizing a pre-trained model (InceptionV3) for feature extraction and combining it with RNNs for sequential analysis.

Timeline with Milestones-



Deliverables-

- **Phase 1:** Dataset Preparation & Frame Extraction Code.
- **Phase 2:** CNN-RNN Model for DeepFake Detection.
- **Phase 3:** Flask API & Web Interface for video upload and detection.
- **Phase 4:** Final Documentation and GitHub Repository.





Detailed Component Diagram

1. Data Ingestion Layer:

- **Source:** DeepFake Video Dataset
- **Extraction:** Frames from videos

2. Model Layer:

- **CNN (InceptionV3)** -> Feature Extraction
- **RNN (LSTM/GRU)** -> Sequential Data Analysis

3. API Layer:

- Flask API to handle video uploads and processing

4. Web Interface:

- User Interface for uploading videos and displaying results

Technical Documentation

➤ System Architecture & Design

The project uses a hybrid CNN-RNN architecture to analyze videos:

- **CNN (InceptionV3):** Extracts spatial features from video frames.
- **RNN (LSTM):** Analyzes temporal patterns in the sequence of frames to detect inconsistencies indicative of DeepFakes.

➤ Key Components and Modules

- **Data Preprocessing:** Extracts frames from videos, normalizes them, and pads sequences.
- **Modeling:** Combines InceptionV3 for feature extraction with LSTM for sequence classification.
- **API Integration:** Flask API handles requests for video uploads and processes them for DeepFake detection.
- **Frontend:** A simple HTML interface for users to upload videos and view results.

➤ Setup and Usage Instructions

1. Clone the Repository:

```
git clone <GitHub Repo URL>
cd deepfake-detection
```

2. Install Dependencies:

```
pip install -r requirements.txt
```

3. Run the Flask Server:

```
python app.py
```

4. Access the Web Interface:

Open a browser and navigate to `http://localhost:5000`.