Deepfake Detection Web App – Project Report

Kishore B
GitHub Repository: https://github.com/kishoreb023/DeepfakeDetectionWeb

Project Overview

This project is a web-based application for detecting deepfake images using a **custom-trained Convolutional Neural Network (CNN)**. The system is designed to operate efficiently on CPU-based systems without requiring any GPU support. Users can upload an image and instantly receive prediction results (Real or Fake), confidence levels, and visual indicators such as a dynamic bar chart and color-coded labels – all integrated in a modern and responsive user interface.

Objective

- Build a reliable and lightweight deepfake detection system
- Provide binary classification: **Real or Fake**
- Display confidence scores visually
- Enable smooth and fast predictions on CPU
- Deliver a modern and engaging frontend interface
- Maintain low latency and high performance
- Ensure simplicity for end-users through one-step prediction
- Visualize prediction results with clarity
- Optimize for real-time interaction in web environments

Model Details

- **Model Type:** Custom-trained CNN
- **Input Shape:** 224x224 RGB Images
- **Framework:** TensorFlow / Keras
- **Output:** Binary Classification (Real / Fake)
- **Training Dataset:** Synthetic and real-world face images
- **Accuracy Achieved:** 98%+ on validation set
- **Preprocessing:** Resizing, normalization, data augmentation

Tech Stack Used

- **Backend:** Python, Flask
- **Frontend:** HTML, CSS, JavaScript
- **Model Inference:** TensorFlow / Keras
- **Visualization:** Matplotlib (Bar Chart)
- **Styling:** Bootstrap & Custom CSS

Key Features

- Upload an image to instantly detect whether it is Real or Fake
- Real-time confidence score visualization using bar chart
- Intuitive UI design with color-coded feedback
  - **Blue = Real**
  - **Red = Fake**
- Fully optimized for CPU usage
- Result display appears instantly after upload (no extra click needed)
- Prediction results displayed beside uploaded image for clarity
- Secure file handling and automatic deletion after prediction

---

Results & Performance

The application demonstrates excellent performance even on low-spec machines. The custom model loads quickly, and predictions are returned within seconds. The visual bar chart, along with image preview and clearly labeled prediction results, ensures an effective user experience. Testing on a variety of deepfake and real images showed high accuracy and strong generalization of the model.

Internship Acknowledgment

This project was developed as part of the **Artificial Intelligence Internship** program at **Navodita Infotech**.

- **Internship Duration:** 5th June 2025 – 4th July 2025
- **Internship Position:** Artificial Intelligence Intern

Special thanks to **Navodita Infotech** for the opportunity, mentorship, and support throughout this project. This internship provided real-world experience in AI-based web deployment, teamwork, and applied machine learning practices.