Face Mask Detection with Live Alert System

🔍 Project Overview

The **Face Mask Detection with Live Alert System** is a real-time computer vision web application built using Python, OpenCV, Keras (TensorFlow), and Flask. It uses a pre-trained deep learning model to detect whether a person is wearing a mask or not via a webcam and displays a live video feed with appropriate labels.

📁 Project Structure

php

CopyEdit

FaceMaskDetector/

├── app.py # Flask web application

├── detect_mask_video.py # Real-time mask detection script

├── mask_detector.model # Trained CNN model

├── haarcascade_frontalface_default.xml # Face detector

├── static/

| └── style.css # Web styling

├── templates/

| └── index.html # Web interface

└── README.md # Project instructions

⚙️ Technologies Used

- Python
- OpenCV
- TensorFlow / Keras
- Flask
- HTML/CSS (for UI)

🧠 Model

The model used is a Convolutional Neural Network (CNN) trained on a dataset of masked and unmasked face images. It was saved in .model format using Keras.

🚀 How to Run the Web App

1. **Install dependencies** (if not already):

bash

CopyEdit

pip install opencv-python tensorflow flask

2. **Ensure the following files are present in your project folder**:
    - mask_detector.model
    - haarcascade_frontalface_default.xml
    - app.py
    - templates/index.html
    - static/style.css
3. **Run the Flask app**:

bash

CopyEdit

python app.py

4. Open your browser and go to:
   http://127.0.0.1:5000

📸 Live Detection Features

- Real-time detection using webcam.
- Color-coded labels:
    - ✅ Green: Mask
    - ❌ Red: No Mask

📝 Notes

- Ensure webcam access is granted.
- For Windows, run the project from a folder without spaces in the path (e.g., avoid C:\Users\Your Name\Desktop\...).
- Model accuracy depends on lighting and camera clarity.

🙌 Acknowledgments

- Haar Cascade from OpenCV.
- Dataset used to train the model: Kaggle Face Mask Dataset