Face-Mask-Detection-with-Live-Alert-System

1. Introduction: -

The COVID-19 pandemic has heightened the importance of personal protective measures such as wearing face masks. This project aims to address the need for a real-time face mask detection system to promote safety and compliance. The system utilizes computer vision and deep learning techniques to detect whether individuals in a video feed are wearing masks and provides live alerts when violations are detected.

2. Abstract :-

This project implements a real-time face mask detection application using a webcam feed. Leveraging pre-trained deep learning models and OpenCV, the system detects faces and classifies them into "Mask" or "No Mask" categories. If a person is detected without a mask, a visual alert is triggered along with a printed warning in the console. This solution is highly relevant for public spaces, offices, and health-critical environments to automate compliance monitoring.

3. Tools Used :-

- **Programming Language**: Python
- **Libraries/Frameworks**:
  - TensorFlow / Keras (for model training and loading)
  - OpenCV (for image processing and video stream)
  - NumPy (for numerical operations)
  - imutils (for video stream manipulation)
- **Model**: A pre-trained Keras CNN model saved as mask_detector_model.h5

4. Steps Involved in Building the Project:-

1. **Face Detection**:
   - Used OpenCV's Haar Cascade classifier to detect faces in a live webcam stream.
   - Alternatively, detect_mask_video.py used a Caffe-based DNN face detector for better accuracy.

2. **Model Integration**:

   o Loaded a pre-trained Keras model (mask_detector_model.h5) that was trained to classify faces into "Mask" and "No Mask" categories.

3. **Image Preprocessing**:

   o Detected faces are cropped, resized (100x100 or 224x224 depending on script), normalized, and reshaped to match model input.

4. **Prediction and Classification**:

   o For each detected face, predictions are made using the model.

   o Class labels are determined and visualized on the screen with bounding boxes and probability.

5. **Live Alert System**:

   o In case of "No Mask", the system prints a console alert: 🚨 ALERT: No Mask Detected!.

   o The interface displays red rectangles for "No Mask" and green for "Mask".

6. **Deployment**:

   o Run via terminal using: python detect_mask_live.py (or detect_mask_video.py if DNN-based).

   o Press 'q' to quit the live feed.

---

5. Conclusion :-

This project successfully demonstrates the integration of machine learning and computer vision to solve a practical, health-oriented problem. The face mask detection system performs efficiently in real-time and can be enhanced further with features like sound alerts, mask compliance logs, and deployment over surveillance systems. It serves as a foundational step toward smart health monitoring applications using AI.