PROJECT REPORT ON

Face Mask Detection with Live Alert System

🔹 Introduction

The COVID-19 pandemic has emphasized the importance of wearing masks in public spaces. Manual monitoring is time-consuming and inefficient. This project introduces an AI-based Face Mask Detection system that automates this process using computer vision and deep learning.

🔹 Abstract

This project focuses on real-time face mask detection using a webcam. A convolutional neural network (CNN) model is trained to classify whether individuals are wearing a mask or not. The model is integrated with OpenCV for live video stream detection and alerts when no mask is detected. Haar Cascade classifiers are used for face detection, and Flask is optionally used for deployment. The project demonstrates the application of AI/ML in enhancing public safety.

🔹 Tools Used

- **Programming Language**: Python
- **Libraries**: OpenCV, TensorFlow/Keras, NumPy, Matplotlib
- **Face Detection**: Haar Cascade
- **Model Training**: CNN (Convolutional Neural Network)
- **Dataset**: Masked and unmasked face images (Kaggle Dataset)

🔹 Steps Involved in Building the Project

1. **Data Collection & Preprocessing**
    - Used a dataset of images categorized into with_mask and without_mask.
    - Images were resized to 128×128 and normalized.
2. **Model Building**
    - Built a CNN model using Keras with Conv2D, MaxPooling, and Dense layers.
    - Used Binary Crossentropy as the loss function and Adam optimizer.
3. **Model Training & Evaluation**
    - Trained the model on training data with validation.

- Achieved high accuracy in distinguishing between masked and unmasked faces.

4. **Face Detection**
    - Integrated OpenCV's haarcascade_frontalface_default.xml for detecting face regions in webcam frames.
5. **Mask Prediction on Real-Time Video**
    - Detected faces in real-time video.
    - Cropped and resized the face, then passed it to the trained model to predict mask status.
    - Displayed bounding boxes and labels like "Mask" or "No Mask".
6. **Live Alert System**
    - Alerts triggered visually (or optionally with sound) when a person without a mask is detected.
7. **Deployment**
    - Deployed a basic web interface using Flask for demo purposes.

🔹 Conclusion

The project successfully demonstrates how deep learning and computer vision can be used to automate face mask detection in real-time. It is fast, lightweight, and can be deployed on systems with minimal resources. This system can be extended with features like sound alerts, temperature checks, or integration with CCTV systems.