Mini Project: Face Mask Detection Using Computer Vision

1. Objective

Develop a computer vision system that automatically detects whether a person is wearing a face mask or not in real-time using a webcam or video feed.

2. Tools & Technologies

- Programming Language: Python

- Libraries/Frameworks:

  * TensorFlow / Keras (for model training)

  * OpenCV (for image capture and processing)

  * NumPy, Matplotlib (for data manipulation and visualization)

- Dataset: Kaggle Face Mask Detection Dataset or MaskedFace-Net

3. Project Workflow

Step 1: Data Collection & Preprocessing

Download a dataset of labeled images (With Mask, Without Mask).

Preprocess images: Resize, Normalize, and Split (Train/Test).

Step 2: Model Design

Use a Convolutional Neural Network (CNN). Example in Keras:

```
model = Sequential([
    Conv2D(32, (3,3), activation='relu', input_shape=(128, 128, 3)),
    MaxPooling2D(2,2),
    Conv2D(64, (3,3), activation='relu'),
    MaxPooling2D(2,2),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(2, activation='softmax')
])
```

Step 3: Model Training

Compile and train the model with augmented data.

```
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

Step 4: Real-Time Detection with OpenCV

Load model, use OpenCV for video capture and face detection. Predict mask status for each face

and display with bounding boxes.

4. Skills Learned

- Image classification

- Object detection

- Model training and evaluation

- Real-time system integration

5. Real-World Applications

Used in public safety monitoring, automated alerts, and access control in places requiring masks.

6. Evaluation Metrics

Accuracy, Precision, Recall, F1-score, Confusion matrix, Detection latency.

7. Optional Enhancements

- Deploy using Flask/Streamlit

- Use MobileNetV2/YOLOv5 for fast inference

- Add voice alerts or notifications