Case Study: Real-Time Face Mask Detection Using Deep Learning and OpenCV

---

1) Problem Statement and Objectives

Problem Statement:

During the COVID-19 pandemic, it became essential to ensure that individuals wear face masks properly in public areas to reduce virus transmission. Manual monitoring is inefficient and prone to human error.
This project aims to develop an **automated real-time face mask detection system** that identifies whether a person is wearing a mask, not wearing one, or wearing it improperly.

Objectives:

- Detect human faces in real-time using a webcam feed.
- Classify each detected face as **With Mask**, **Improper Mask**, or **Without Mask**.
- Provide **visual and sound alerts** for improper or missing masks.
- Develop a **lightweight deep learning model** that can run efficiently on CPU devices.
- Convert the model into **TensorFlow Lite** format for portable and mobile deployment.

---

2) Data Preprocessing

Dataset Overview:

The dataset consisted of three categories of images:

1. **With Mask**
2. **Without Mask**
3. **Improper Mask**

Each image represented a face captured in varying light, angles, and environments.

Preprocessing Steps:

1. **Image Resizing:** All images were resized to **224 × 224 pixels** to match the MobileNetV2 input requirement.
2. **Normalization:** Pixel values were scaled between -1 and 1 using preprocess_input() from TensorFlow.
3. **Augmentation:** To improve dataset diversity, the following augmentations were applied:
   - Rotation (±20°)
   - Horizontal Flip

    - Zoom (0.15x)
    - Width and Height Shifts (0.2)
    - Shear Transformations

4. **Train-Validation Split:**
   The dataset was divided into **80% training** and **20% validation** sets.

5. **Batch Generation:**
   The ImageDataGenerator was used to load and preprocess images efficiently during training.

These preprocessing steps ensured better generalization and reduced overfitting.

---

3) Model Selection and Development

Model Selection:

Based on research, **MobileNetV2** was chosen due to its:

- Lightweight architecture suitable for low-end systems.
- High accuracy with minimal computational cost.
- Proven effectiveness in image classification and real-time applications.

The model design was inspired by the paper
"Face Mask Detection Using MobileNetV2" (Loey et al., 2021, DOI: 10.1016/j.cviu.2021.103209).

Model Architecture:

- **Base Model:** MobileNetV2 (pretrained on ImageNet, with frozen layers)
- **Added Layers:**
    - GlobalAveragePooling2D
    - Dense (128, ReLU activation)
    - Dropout (0.3)
    - Output Layer (Softmax with 3 classes)
- **Optimizer:** Adam (learning rate = 0.0001)
- **Loss Function:** Categorical Crossentropy
- **Metrics:** Accuracy

Training:

- Epochs: 10 + 3 fine-tuning epochs
- Batch Size: 32
- Validation Split: 20%

- EarlyStopping and ModelCheckpoint callbacks were used to prevent overfitting and save the best model.

The model was then fine-tuned by unfreezing the last 30 layers of MobileNetV2 for higher accuracy.

---

4) Visualizations and Insights

Visualizations Generated:

1. **Training and Validation Accuracy/Loss Curves** – to monitor learning behavior.
2. **Confusion Matrix** – to analyze prediction performance across all classes.
3. **Augmented Image Samples** – to visualize image diversity after preprocessing.
4. **Real-Time Detection Window** – displaying colored bounding boxes and labels.

All visual outputs are stored in the **/artifacts** folder:

- training_curves.png
- confusion_matrix.png
- augmentation_samples.png

Insights:

- The model achieved **~87% overall accuracy** on validation data.
- “With Mask” detection performed best with **99% recall**.
- Lower accuracy for “Improper Mask” occurred due to fewer dataset samples.
- Real-time detection runs smoothly on CPU, showing responsive performance and stable frame rate.

---

5) Recommendations

1. **Increase Dataset Balance:** Add more samples for “Improper Mask” to improve detection accuracy.
2. **Custom Audio Alerts:** Use asynchronous sound playback to give voice warnings like “Please wear a mask properly.”
3. **Edge Deployment:** Deploy the .tflite model on Raspberry Pi or Android devices for real-time monitoring.
4. **Group Detection:** Extend system capability to handle multiple faces in a single frame.
5. **Cloud Analytics:** Connect detection logs to a cloud database to store violation data for further insights.

Summary

This project successfully demonstrates how **deep learning and computer vision** can be applied to promote public safety.
By combining **MobileNetV2**, **OpenCV**, and **TensorFlow Lite**, the system achieves reliable and fast detection while remaining lightweight enough to run without a GPU.

The model's **87% accuracy** proves that it performs effectively, and with further dataset improvements, it can reach production-level reliability for real-world mask compliance monitoring.