॥वसुधैव कुटुम्बकम्॥
SYMBIOSIS
INSTITUTE OF GEOINFORMATICS

Computer Vision: Project

<u>Face Mask Detector</u>

Name : Alviya Ali
PRN : 24070243005

• Introduction

The COVID-19 pandemic underscored the importance of wearing face masks to reduce the spread of airborne diseases. However, manual enforcement of mask-wearing in public places is inefficient and prone to human error. To address this, an automated Face Mask Detection System leveraging deep learning and computer vision techniques has been developed. This system utilizes Convolutional Neural Networks (CNNs) to classify individuals as wearing a mask or not. By integrating with real-time surveillance systems, it can assist in enforcing health regulations and improving public safety. The model is based on MobileNetV2, an optimized neural network for efficient image classification, making it suitable for real-time applications. The objective is to create a scalable and accurate face mask detection system that enhances safety measures in public spaces, ensuring compliance with mask mandates and reducing infection risks.

• Objective

The key objectives of this project include:

- Developing a deep learning model capable of distinguishing between images of people wearing and not wearing masks.
- Training the model using a dataset of labeled face images.
- Evaluating the model's performance using appropriate metrics.
- Deploying the model for real-time face mask detection using OpenCV.

- ## Tools and Libraries Used

  Python libraries used in this project for data preprocessing, model training, and real-time detection are :

- **TensorFlow/Keras:** Used to build and train the deep learning model (MobileNetV2).
- **OpenCV:** Used for real-time image processing and displaying predictions.
- **Matplotlib:** Used for visualizing training performance.
- **NumPy:** Used for array manipulations.
- **Scikit-learn:** Used for encoding labels and evaluating model performance.
- **Imutils:** Used for handling image paths.

- ## Dataset Description

  The dataset consists of images categorized into two classes:
    - With Mask
    - Without Mask
- **Preprocessing Steps:**
    1. **Loading Images:** Images were loaded from the dataset directory using OpenCV.
    2. **Resizing:** All images were resized to 224×224 pixels.
    3. Normalization: Pixel values were scaled using preprocess_input() from MobileNetV2.
    4. **Label Encoding:** The categorical labels ('with_mask' and 'without_mask') were converted into numerical values using LabelBinarizer() and to_categorical().
    5. **Train-Test Split**: The dataset was split into 80% training data and 20% testing data.
    6. **Data Augmentation:** Random transformations (rotation, zoom, shifts, flips) were applied to improve model generalization.

- ## Model Development and Training

  1). Model Architecture

  - The MobileNetV2 model was used as the base network due to its efficiency and high accuracy in image classification. The model includes:

- Feature Extraction: Pre-trained MobileNetV2 layers with frozen weights.

- Custom Fully Connected Layers: A pooling layer, followed by a dense layer with ReLU activation, dropout for regularization, and a softmax output layer.

2). Training Configuration

- Optimizer: Adam (learning_rate = 1e-4)

- Loss Function: Binary Crossentropy

- Batch Size: 32

- Epochs: 20

- Evaluation Metrics: Accuracy, Loss, and Classification Report

3). Training Process

- The dataset was fed into the model in batches.

- The training progress was monitored using accuracy and loss curves.

- Data augmentation was applied to enhance generalization.

• Results

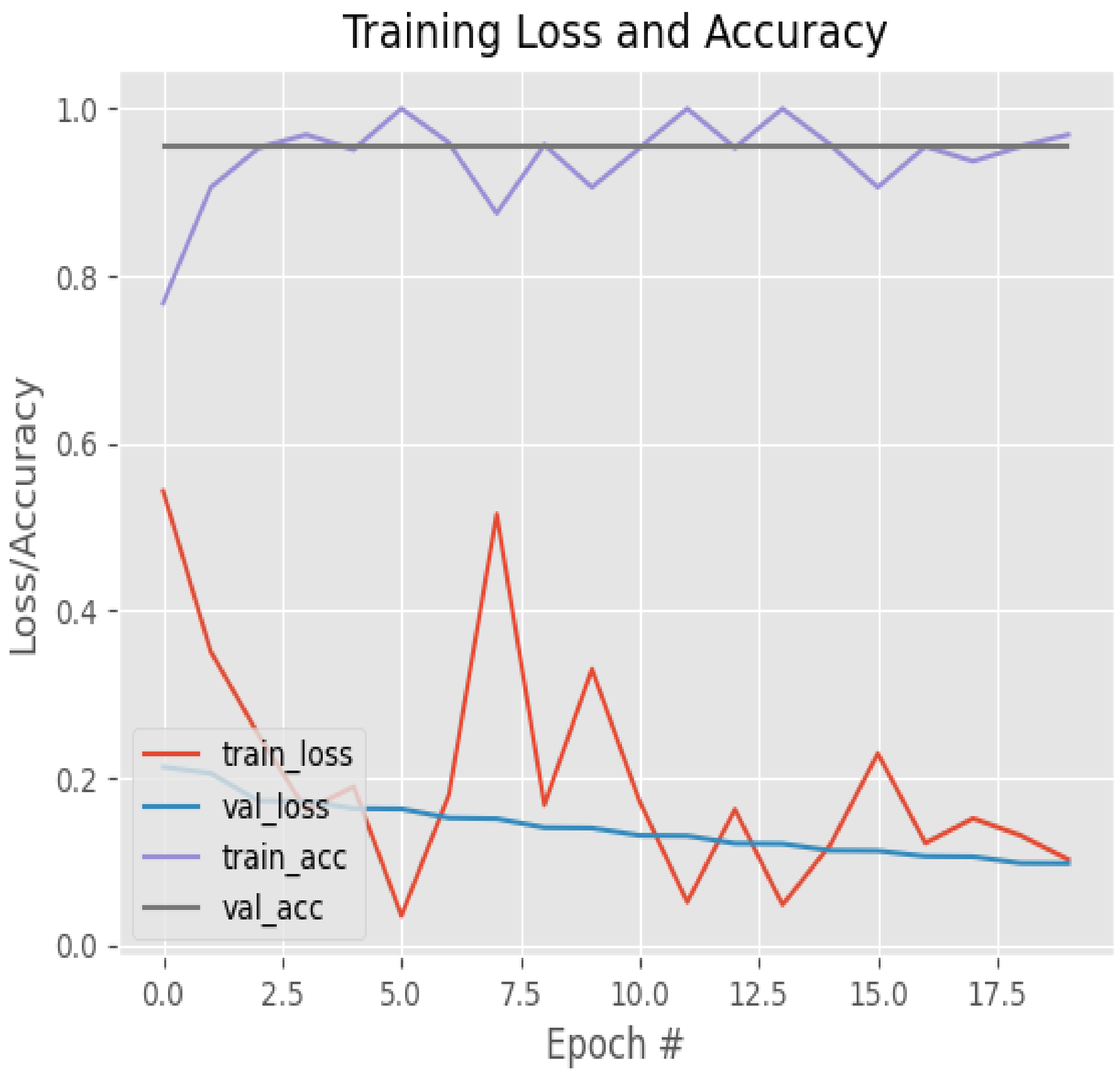
1). Model Performance

- The model successfully classified images into "With Mask" and "Without Mask" categories.
- The classification report showed high accuracy in both training and validation datasets.
- The training and validation loss/accuracy curves indicated minimal overfitting.

2). Training Loss and Accuracy Analysis

The training process was monitored using a plot of training loss, validation loss, training accuracy, and validation accuracy over the course of 20 epochs.

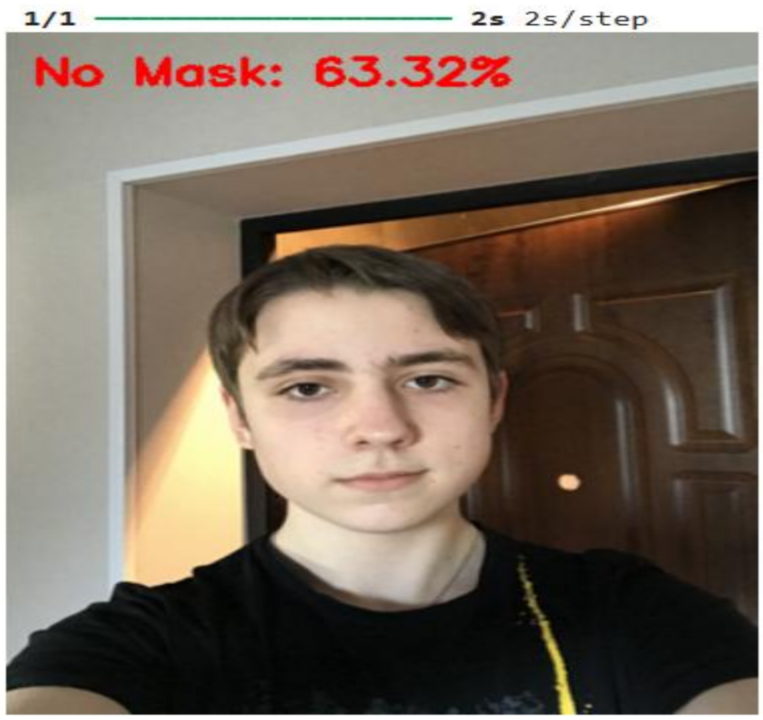
- **Loss Trends**: Training loss (red line) shows a decreasing trend, indicating proper learning. Validation loss (blue line) remains stable, suggesting the model generalizes well.
- **Accuracy Trends**: Training accuracy (black line) is consistently high, while validation accuracy (purple line) is also high, indicating minimal overfitting.
- **Key Observations**: The fluctuations in training loss suggest variations in learning, which could be further smoothed by fine-tuning hyperparameters.



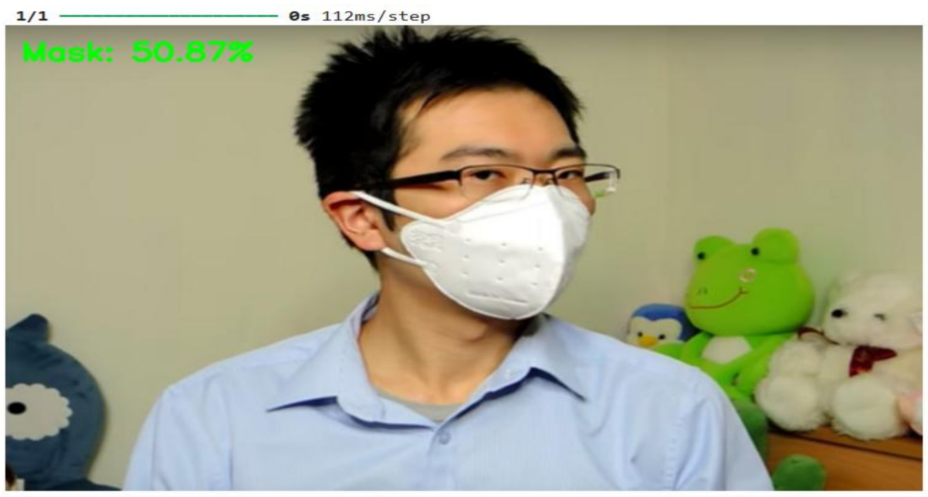
3). Sample Predictions

The trained model was used to predict face mask status on sample images. The results included:

- Predicted label: Mask / No Mask
- Confidence Score: Probability of the prediction being correct
- Image Display: The model annotated the image with the prediction label and confidence percentage.



```
1/1 ━━━━━━━━━━━━━━━━━━━━ 2s 2s/step
```



```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 112ms/step
```

- **Deployment and Real-Time Implementation**

  1). Loading the Model
  - The trained model was saved as mask_detector.keras and loaded for real-time detection.

  2). Face Mask Detection Using OpenCV

  - The OpenCV library was used to read and process images.

  - The model classified each image and overlaid the predicted label and confidence score on the image.

  - The final output image was displayed using cv2_imshow().

- **Conclusion**

- A deep learning model was successfully developed and trained for Face Mask Detection.
- The MobileNetV2 model demonstrated high accuracy in classifying masked and unmasked faces.
- The model was tested on various images and achieved reliable predictions with high confidence scores.
- Real-time implementation was achieved using OpenCV, making it suitable for surveillance applications.