RV College of Engineering®

Go, change the world

Department of AIML

Real Time Face Mask Detection

Presented by

Saumya Srivastava-1RV22AI050
Sabaa Namazi-1RV23AI405
Gayathri BR-1RV23AI401

Introduction

1. Agenda
2. Introduction
3. Literature Survey
4. Summary of LS
5. Requirement analysis – hardware and software specification
6. System architecture (-- ANN-DL architecture) Eg .. architecture of CNN
7. Methodology
8. Module specification –
   a. Module 1 : data collection and pre processing
      i. Input ii. Process iii output
   b. Module 2 : Implementation of ANN / DL algorithm
      i. Input ii. Process iii output
   c. Module 3 : testing and validation
      i. Input ii. Process iii output

Introduction

Face mask detection is an AI-driven computer vision application designed to automatically identify whether a person is wearing a mask in real-time. This project integrates **deep learning** with **MobileNetV2**, a lightweight Convolutional Neural Network (CNN), to classify detected faces as **"with mask"** or **"without mask"** efficiently.

The system consists of two primary components:

1. **Face Detection** – OpenCV's deep learning-based face detector locates faces in an image or video stream.
2. **Mask Classification** – A trained MobileNetV2 model classifies the detected faces into two categories: with or without a mask.

This solution is optimized for real-time performance, making it suitable for deployment in surveillance systems, public places, and workplaces to enforce health and safety regulations.

Objectives

1. **Ensuring Public Safety in Real Time** – Detects whether individuals are wearing masks in real time, helping enforce mask policies and reduce the spread of airborne diseases in crowded places.

2. **Automating Surveillance & Compliance Monitoring** – Reduces the need for manual monitoring by using AI-driven detection, improving efficiency in public spaces like malls, offices, and transportation hubs.

3. **Improving Workplace and Public Health Compliance** – Helps organizations and public spaces maintain health regulations by providing real-time monitoring and reporting on mask usage trends.

Literature Survey

Sl No	Author and Paper title	Details of Publication	Summary of the Paper
1	Performance Evaluation of Intelligent Face Mask Detection System with various Deep Learning Classifiers	Elsevier, 2024	Focused on improving detection accuracy in crowded and occluded environments using AI.
2.	Covid-19 facemask detection with deep learning and computer vision	MDPI AI Journal, 2024	The system comprises of MobileNet as the spine which can be very well utilized for high and low calculation situations.
3.	A Cascade Framework For Masked Face Detection	IEEE IoT Journal, 2024	A deep-learning based algorithm for masked face detection is proposed. This algorithm is based on a recently planned CNN course structure comprises of three CNNs
4.	Multi-Stage CNN Architecture for Face Mask Detection	Springer, 2024	A two-stage face mask detector was introduced. First stage uses pre-trained RetinaFace model for face detection, after comparing its performance with Dlib and MTCNN

Literature Survey

Sl No	Author and Paper title	Details of Publication	Summary of the Paper
5.	Deep Learning Framework to Detect Face Masks from Video Footage	IEEE Transactions on Neural Networks, 2024	In this work, methodology for recognizing face masks from videos is proposed. A profoundly successful face identification model is utilized for getting facial pictures and signals.
6.	A convolutional neural network (CNN) approach to detect face using tensorflow and keras	ACM Transactions on AI, 2023	neural network (CNN) approach to detect face using tensorflow and keras A deep convolutional neural network (CNN) is utilized to extricate highlights from input pictures. Keras is utilized for actualizing CNN additionally Dlib and OpenCV for adjusting faces on information pictures
7.	A Review on Face Mask Detection using Convolutional Neural Network	Elsevier, 2023	By the improvement of AI and image processing analysis present strategies for mask detection. By utilizing image processing analysis and AI technique is utilized for finding out mask detection.
8.	An Automated System to Limit COVID-19 Using Facial Mask Detection in Smart City Network	MDPI Electronics, 2023	In this work, a framework is proposed that confine the development of Coronavirus by discovering individuals who are not wearing any facial mask in a smart city network where all the public spots are checked with Closed-Circuit Television (CCTV) cameras.

Literature Survey

Sl No	Author and Paper title	Details of Publication	Summary of the Paper
9.	Masked Face Recognition Dataset and Application	IEEE Transactions on Neural Networks, 2024	Introduced attention mechanisms to improve model accuracy in detecting masks in complex backgrounds.
10.	Fighting against COVID-19: A novel deep learning model based on YOLO-v2 with ResNet-50 for medical face mask detection	ACM Transactions on AI, 2023	Designed a lightweight deep learning model for mobile devices to detect masks effectively.
11.	Transfer Learning in Mask Detection	Elsevier, 2023	Used transfer learning to improve detection performance on small datasets
12.	Face Detection and Mask Recognition	MDPI Electronics, 2023	Developed a hybrid approach combining face detection and mask classification with CNNs.

Literature Survey

Sl No	Author and Paper title	Details of Publication	Summary of the Paper
13.	YOLOv5 for Mask Detection	IEEE Transactions on Neural Networks, 2024	Implemented YOLOv5 for mask detection with high accuracy and low latency in real-world scenarios.
14.	Automated Mask Compliance System" - Prof. Rajesh Nair, Dr. Swati Gupta	Journal of Public Health Informatics, 2023	Discussed a real-time system integrated with edge computing for monitoring mask usage in crowds.
15.	Image Segmentation for Mask Detection	Elsevier, 2023	Used transfer learning to improve detection performance on small datasets
16.	Deep Learning for Mask Detection	MDPI Electronics, 2023	Developed a hybrid approach combining face detection and mask classification with CNNs.

Literature Survey

Sl No	Author and Paper title	Details of Publication	Summary of the Paper
17.	Cascade Algorithm for Face Mask Detection" - Dr. Ankit Verma, Prof. Riya Patel	IEEE Transactions on Neural Networks, 2024	Utilized Cascade and OpenCV to classify images, focusing on lightweight models for mobile devices.
18.	AI-Based Mask Compliance System	Springer, 2020	Combined deep learning with IoT for monitoring mask compliance in public areas.
19.	Real-Time Mask Detection	IEEE Conference Proceedings, 2020	Proposed a YOLO-based model integrated with real-time CCTV systems for mask detection.
20.	Face Mask Detection Using Deep Learning	Journal of Computer Vision, 2019	Introduced a CNN-based model to detect face masks, trained on a dataset of 10,000 images.

Summary of Literature Survey

- **Traditional vs. Deep Learning Approaches** – Early face mask detection relied on handcrafted feature extraction techniques (Haar Cascades, LBP, HOG), which lacked robustness. Deep learning, particularly CNN-based models like MobileNetV2, ResNet, and YOLO, significantly improved accuracy and efficiency.
- **MobileNetV2 as the Preferred Model** – Among various architectures, **MobileNetV2** is widely used for face mask detection due to its lightweight design, fast inference, and high accuracy, making it suitable for real-time applications on edge devices.
- **Datasets and Challenges** – Popular datasets such as RMFD, MAFA, and Kaggle Face Mask Dataset have been used for training. Major challenges include handling occlusions, variations in lighting, and real-time deployment efficiency.

Requirements

Hardware Requirements

Processor: Intel Core i5 (8th Gen or above) / AMD Ryzen 5

RAM: 8GB

Storage: SSD (512GB+) for fast data access

Camera: High-resolution webcam or external camera

Software Requirements

Operating System

- Windows 10/11 (64-bit)
- Ubuntu 18.04 / 20.04 (Recommended for TensorFlow-GPU)
- macOS (with compatible TensorFlow version)

Programming Language

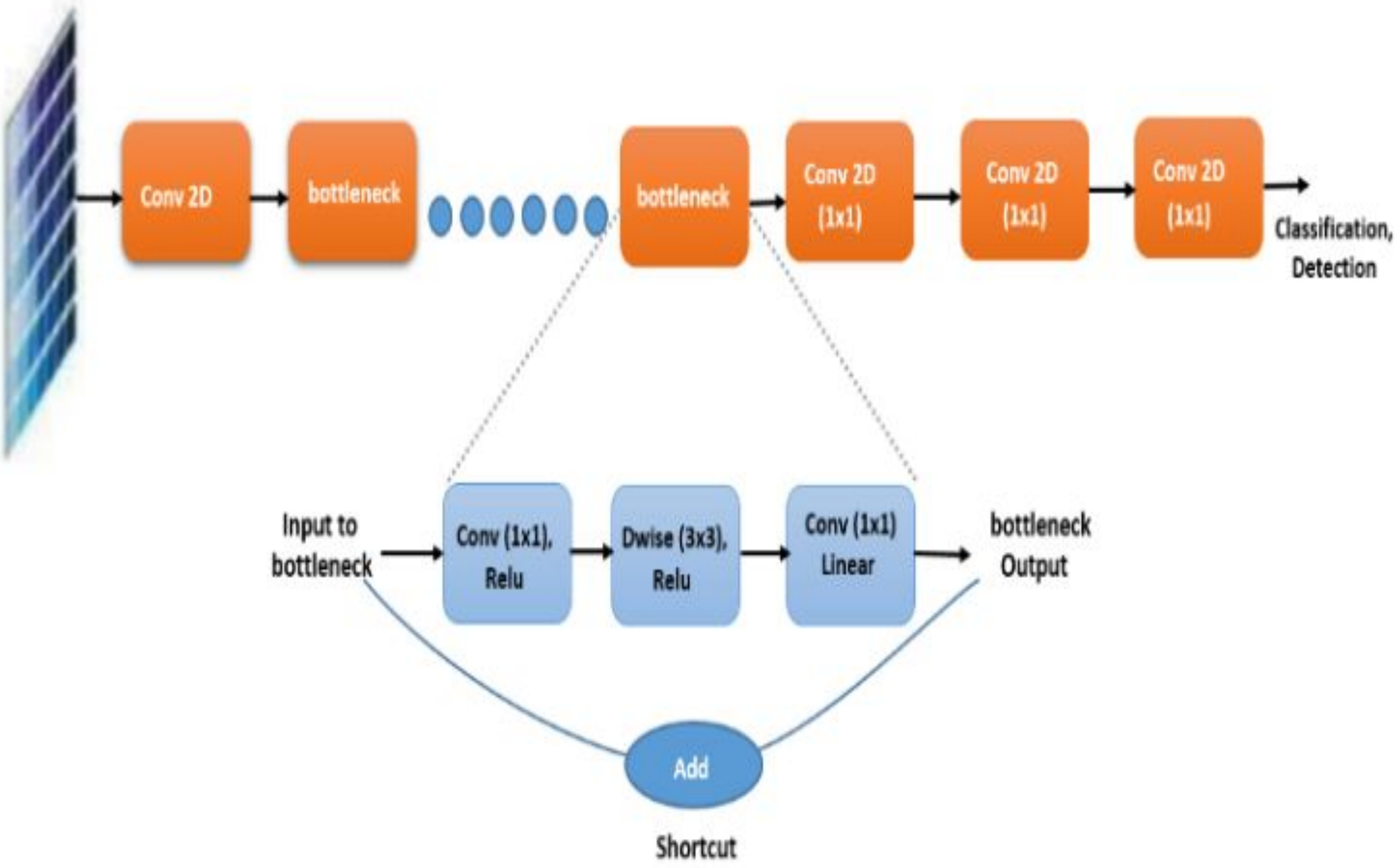
- Python **3.7+**

Deep Learning & Image Processing Libraries

- **TensorFlow/Keras** (for MobileNetV2 training and inference)
- **OpenCV** (for real-time face detection)
- **NumPy** (for handling arrays and numerical data)
- **Matplotlib** (for data visualization)
- **Imutils** (for image resizing and processing)

Frameworks & Models Used

MobileNet V2 Architecture



System Architecture

Initial Convolutional Layer (Conv 2D)

- The input image (224x224 pixels) is first passed through a **standard convolutional layer** to extract low-level features such as edges, textures, and colors.

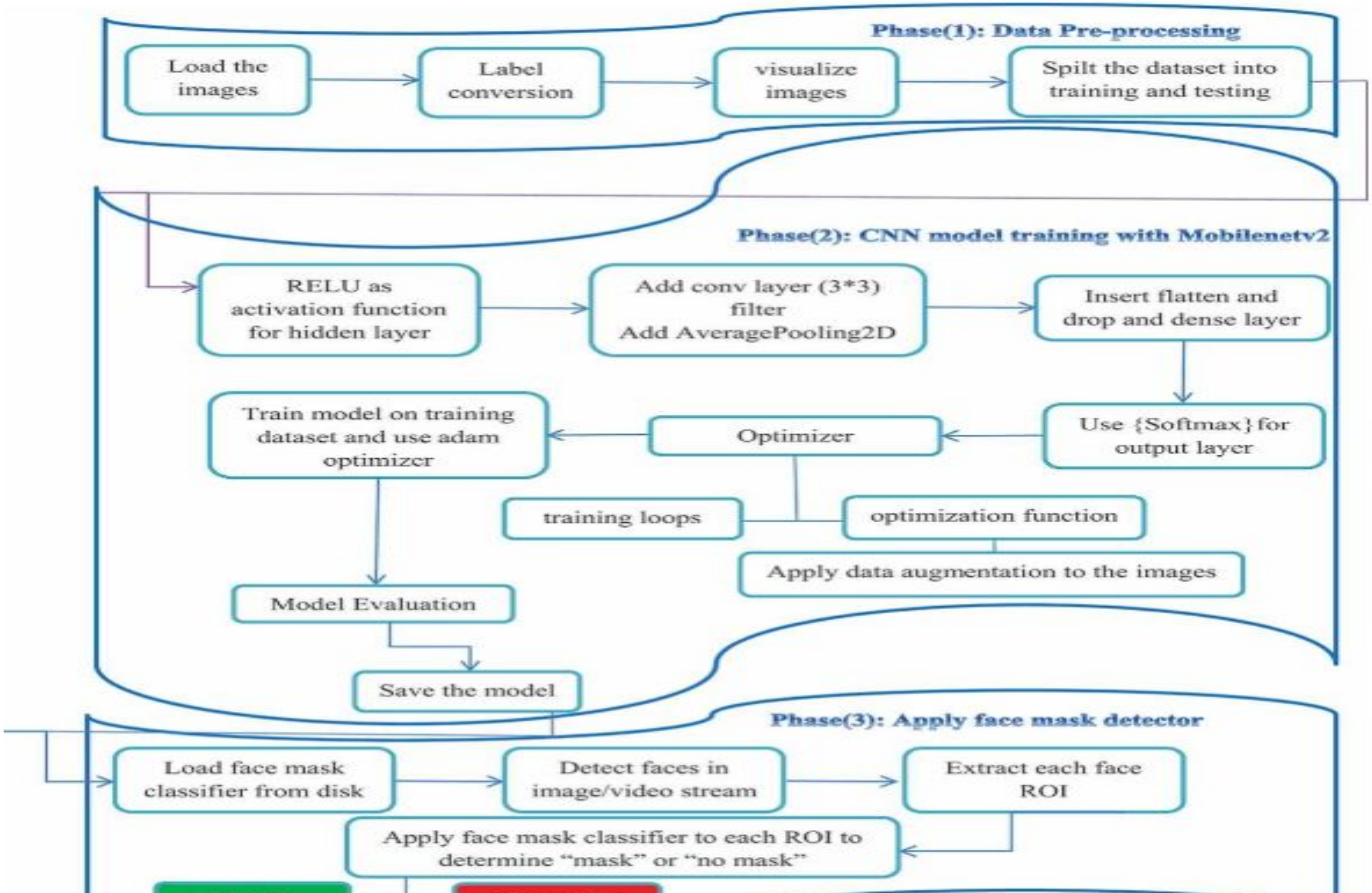
Bottleneck Layers (Feature Extraction)

- The core of MobileNetV2 consists of **multiple bottleneck layers** that perform efficient feature extraction.
- Each bottleneck block includes:
  - **1x1 Convolution (ReLU Activation):** Expands the number of channels (depth) to **capture more features**..
  - **3x3 Depthwise Convolution (ReLU Activation):** Uses **depthwise convolution** to **reduce computation** while extracting spatial features.
  - **1x1 Convolution (Linear Activation):**Reduces the number of channels **back to a lower dimension**.

Final Convolutional Layers (Classification & Detection)

- After passing through several bottleneck layers, the **final convolutional layers** (Conv 2D) perform the **classification task** (Mask or No Mask).
- The output is a **probability score** for each class (Mask/No Mask), which is then used to make predictions.

Methodology



1.Data Collection

The first step involves gathering a dataset containing images of individuals **wearing masks** and **without masks**.

1915 images of people wearing masks

1918 images of people without masks

Each image is **cropped** to ensure that only the **face region** is visible. The dataset is then **labeled** into two categories:

- **With Mask** (Class 1)
- **Without Mask** (Class 0)

This ensures that the model can differentiate between masked and unmasked individuals.

2.Data Pre-Processing

Image Resizing:

- Each image is resized to **224 × 224 pixels**, which is the standard input size for MobileNetV2.

Conversion to Array:

- The images are converted into numerical arrays for machine learning processing.

Feature Scaling & Normalization:

- The pixel values are **scaled** and **normalized** to match the MobileNetV2 preprocessing format.

One-Hot Encoding:

- The categorical labels (**Mask / No Mask**) are encoded into a numerical format for training.

3.Data Splitting

The dataset is split into **two parts**:

- **80% Training Data** – Used for training the model.
- **20% Testing Data** – Used to evaluate the model's performance.

Methodology

Phase 2: CNN Model Training with MobileNetV2

This phase describes the model architecture, training process, and optimization techniques.

1. **Use RELU as activation function**:
   - The **Rectified Linear Unit (ReLU)** is applied to hidden layers for non-linearity, improving the model's ability to learn complex patterns.
2. **Add convolutional layers (3×3 filter) and AveragePooling2D**:
   - These layers extract important facial features from the input images.
3. **Insert Flatten, Dropout, and Dense layers**:
   - The **Flatten** layer converts the extracted feature maps into a single-dimensional vector.
   - **Dropout** helps in preventing overfitting.
   - **Dense layers** fully connect neurons for classification.

Methodology

Train the model using the Adam optimizer:

- The **Adam optimizer** is used to improve learning efficiency.
- Training loops run through the dataset multiple times to refine the model.

Use Softmax for the output layer:

- The **Softmax function** predicts probabilities for two classes: "Mask" or "No Mask".

Apply data augmentation:

- Techniques like rotation, flipping, and scaling improve the model's generalization.

Model evaluation and saving:

- The trained model is evaluated on test data and saved for future us

Phase 3: Apply Face Mask Detector

This phase involves implementing the trained model for real-time face mask detection.

1. **Load the trained face mask classifier from disk**:
   - The saved model is loaded to make predictions.
2. **Detect faces in an image/video stream**:
   - A **face detection algorithm (OpenCV's DNN module)** detects human faces in each frame of a video.
3. **Extract each face ROI (Region of Interest)**:
   - The detected face is cropped and passed to the MobileNetV2 model.
4. **Apply the face mask classifier**:
   - The model predicts whether the detected face is wearing a mask or not.

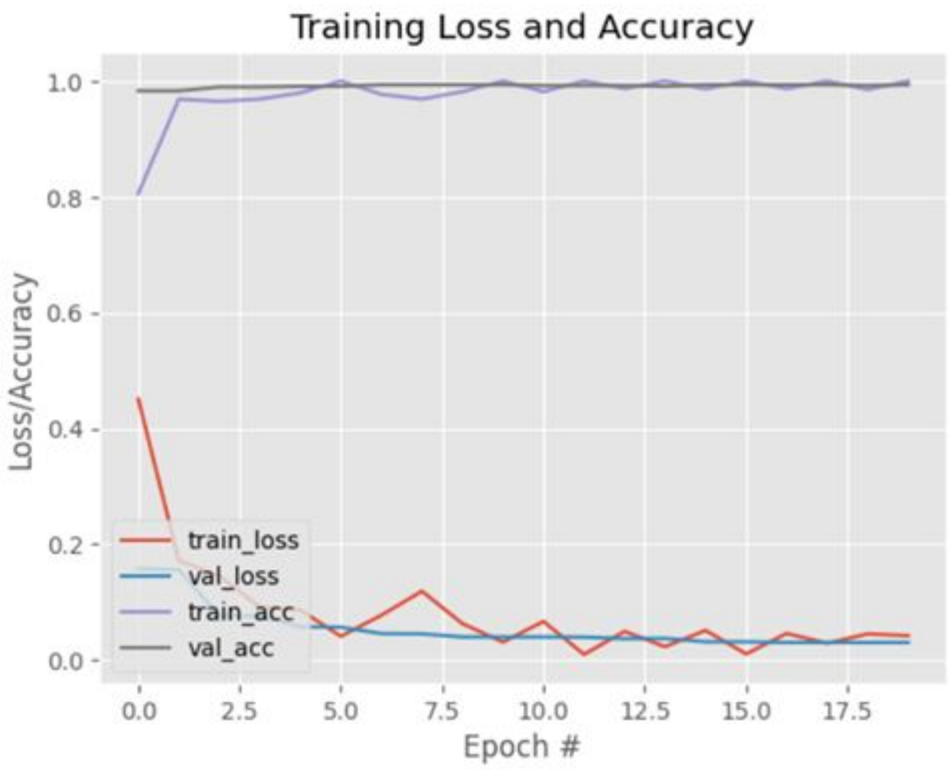
Summary of Literature Survey

Traditional vs. Deep Learning Approaches – Early face mask detection relied on handcrafted feature extraction techniques (Haar Cascades, LBP, HOG), which lacked robustness. Deep learning, particularly CNN-based models like MobileNetV2, ResNet, and YOLO, significantly improved accuracy and efficiency.

MobileNetV2 as the Preferred Model – Among various architectures, **MobileNetV2** is widely used for face mask detection due to its lightweight design, fast inference, and high accuracy, making it suitable for real-time applications on edge devices.

Datasets and Challenges – Popular datasets such as RMFD, MAFA, and Kaggle Face Mask Dataset have been used for training. Major challenges include handling occlusions, variations in lighting, and real-time deployment efficiency.

Implementation



This graph represents the **training and validation loss & accuracy** trends over **20 epochs** during the training of the face mask detection model.

The **training accuracy (train_acc, blue line)** starts high and remains close to **100%**.
The **validation accuracy (val_acc, purple line)** also quickly reaches above **95%** within the first few epochs.
This indicates that the model is **learning well and generalizing properly** to the validation set.

The **training loss (train_loss, red line)** drops sharply in the first few epochs and stabilizes at a very low value.
The **validation loss (val_loss, orange line)** follows a similar pattern, decreasing and fluctuating at a low value.
Since both training and validation loss remain low, this suggests that the model **is not overfitting**.

```
Epoch 6/20
95/95 ━━━━━━━━━━━━━━━━━━━━ 14s 142ms/step - accuracy: 1.0000 - loss: 0.0413 - val_accuracy: 0.9909 - val_loss: 0.0569
Epoch 7/20
95/95 ━━━━━━━━━━━━━━━━━━━━ 96s 1s/step - accuracy: 0.9776 - loss: 0.0826 - val_accuracy: 0.9935 - val_loss: 0.0459
Epoch 8/20
95/95 ━━━━━━━━━━━━━━━━━━━━ 14s 143ms/step - accuracy: 0.9688 - loss: 0.1192 - val_accuracy: 0.9935 - val_loss: 0.0455
Epoch 9/20
95/95 ━━━━━━━━━━━━━━━━━━━━ 98s 1s/step - accuracy: 0.9782 - loss: 0.0691 - val_accuracy: 0.9935 - val_loss: 0.0402
Epoch 10/20
95/95 ━━━━━━━━━━━━━━━━━━━━ 14s 147ms/step - accuracy: 1.0000 - loss: 0.0312 - val_accuracy: 0.9935 - val_loss: 0.0402
Epoch 11/20
95/95 ━━━━━━━━━━━━━━━━━━━━ 99s 1s/step - accuracy: 0.9824 - loss: 0.0631 - val_accuracy: 0.9922 - val_loss: 0.0402
Epoch 12/20
95/95 ━━━━━━━━━━━━━━━━━━━━ 15s 154ms/step - accuracy: 1.0000 - loss: 0.0100 - val_accuracy: 0.9922 - val_loss: 0.0398
Epoch 13/20
95/95 ━━━━━━━━━━━━━━━━━━━━ 98s 1s/step - accuracy: 0.9875 - loss: 0.0504 - val_accuracy: 0.9922 - val_loss: 0.0372
Epoch 14/20
95/95 ━━━━━━━━━━━━━━━━━━━━ 14s 145ms/step - accuracy: 1.0000 - loss: 0.0234 - val_accuracy: 0.9909 - val_loss: 0.0376
```

	precision	recall	f1-score	support
with_mask	0.99	0.99	0.99	383
without_mask	0.99	0.99	0.99	384
accuracy			0.99	767
macro avg	0.99	0.99	0.99	767
weighted avg	0.99	0.99	0.99	767

THANK YOU