**Real-Time Face Mask Detection using Deep Learning by Pranad Munjal et al. is acomparative study between two convolutional neural networks (CNNs): MobileNetV2 and VGG16**

Submitted by:

ABHINAND P SREENIVASAN (243301) DAGIS

ARUN P U (243304) DAGIS

ASWATHI B (243306) DAGIS

In partial fulfillment of the requirements for the award of Master’s program in

Of



**KERALA UNIVERSITY OF DIGITAL SCIENCES, INNOVATION, AND TECHNOLOGY**

**(DIGITAL UNIVERSITY KERALA)**

TECHNOPARK PHASE-IV, THIRUVANANTHAPURAM

May 2025

**Table of Contents**

1. Aim
2. Methodology
   1. Image Preprocessing
   2. Data Normalization and Augmentation
   3. Data Generator Setup
3. Model Training and Evaluation
   1. Model Architecture Using Transfer Learning
   2. Model Training
   3. Model Evaluation
   4. Performance Visualization

4.Result

5. Conclusion

6.GitHub Link

Aim

The primary aim of this study is to develop a reliable and efficient deep learning-based system for real-time face mask detection, which is crucial for public safety during health crises such as the COVID-19 pandemic. The project focuses on comparing two prominent convolutional neural network (CNN) architectures—**MobileNetV2** and **VGG16**—to evaluate their effectiveness in classifying whether individuals in images or video streams are wearing face masks. By analyzing these models in terms of accuracy, training time, computational efficiency, and real-world applicability, the goal is to identify the most suitable architecture for deployment in environments such as surveillance systems and public transport hubs. The study aims to strike a balance between high detection accuracy and low resource consumption to enable fast, scalable, and practical implementation of face mask monitoring in real time.

Methodology

The methodology for this image classification project is divided into two main phases: Phase 1 – Image Preprocessing and Phase 2 – Model Training and Evaluation. These phases ensure a structured pipeline for efficient implementation and evaluation of deep learning models using transfer learning.

Phase 1: Image Preprocessing

1.1 Dataset Extraction and Structure

The dataset is initially compressed in a ZIP file, which is extracted using Python’s zipfile module. The extracted directory is organized into a format suitable for training with Keras, where each subfolder represents a distinct class of images.

1.2 Data Normalization and Augmentation

The ImageDataGenerator class from Keras is used to normalize image pixel values and split the dataset into training and validation subsets. All images are resized to 224 × 224 pixels, and pixel values are rescaled to the [0, 1] range. An 80:20 split is used for training and validation, respectively.

1.3 Data Generator Setup

Training and validation generators are created using flow\_from\_directory(), which automatically labels the data based on folder names and generates

Phase 2: Model Training and Evaluation

2.1 Model Architecture Using Transfer Learning

Two pre-trained convolutional neural networks—VGG16 and MobileNetV2—are used as base models. These are imported from tensorflow.keras.applications, excluding their top layers (include\_top=False) to allow the addition of a custom classifier head.

The custom top includes:

A Global Average Pooling layer,

A Dropout layer (rate = 0.3),

A Dense output layer with softmax activation for multi-class classification.

All layers in the base models are frozen to preserve their learned features from ImageNet.

2.2 Model Training

Both models are trained using the same training and validation data. Each model is trained for 5 epochs with a batch size of 32 using the fit() method. The training history is stored for later analysis.

2.3 Model Evaluation

Validation accuracy is calculated using the evaluate() method to assess how well each model generalizes to unseen data.

2.4 Performance Visualization

Validation accuracy over each epoch is plotted using Matplotlib to visually compare the learning curves of VGG16 and MobileNetV2.

Results

Conclusion

This study successfully implemented and compared two convolutional neural network models—**MobileNetV2** and **VGG16**—for the task of face mask detection. Both models were trained on a balanced dataset containing images of individuals with and without masks. The results demonstrated that **MobileNetV2 outperformed VGG16**, achieving a **higher validation accuracy of 99.2%**, compared to **98.6%** for VGG16, while also exhibiting **lower training loss and faster convergence**. The lightweight architecture of MobileNetV2 makes it more suitable for **real-time deployment**, especially in resource-constrained environments such as surveillance systems and embedded devices. The project highlights the practical value of deep learning in enhancing public health monitoring during pandemics through automated, real-time face mask detection.

GitHub Link: