Project Report

Project Title: Face Mask Detection Using Machine Learning

Submitted To:
Dr. Waqar Malik

Submitted By:
Adhan Akram

Roll No: 20

Session: 2023–2027

Semester: 4th

Department: Artificial Intelligence

Faculty of Computing & Engineering
University of Kotli Azad Jammu & Kashmir

Contents

Abstract

The Face Mask Detection project aims to design a robust and efficient deep learning system capable of identifying whether a person is wearing a face mask or not. This system leverages transfer learning using the MobileNetV2 architecture for feature extraction and custom layers for binary classification. The model is trained on a dataset containing images of individuals with and without masks, with appropriate preprocessing and augmentation to improve generalization. The solution supports image-based testing and can be extended for real-time mask detection using a webcam. This project demonstrates the application of convolutional neural networks (CNNs) in practical computer vision problems relevant to public health monitoring.

1 Introduction

The global outbreak of infectious diseases has highlighted the importance of preventive measures, including wearing face masks in public spaces. Automated detection of face masks can assist authorities and organizations in enforcing mask compliance in crowded areas. This project focuses on developing a deep learning-based mask detection system that provides accurate predictions for uploaded images and can be extended to real-time video detection.

2 Objectives

- To implement a reliable face mask detection system using deep learning techniques.
- To utilize transfer learning with a pre-trained MobileNetV2 model for efficient feature extraction.
- To preprocess and augment the dataset to improve model generalization.
- To provide a user-friendly interface for testing images and visualizing results.
- To analyze training performance through metrics such as accuracy and loss.

3 Tools and Software Used

- **Programming Language:** Python 3.x
- **Libraries:** TensorFlow, Keras, OpenCV, PIL, NumPy, Matplotlib
- **Platform:** Google Colab
- **Dataset:** Images of people with and without masks from a publicly available GitHub repository
- **Additional Tools:** Google Colab file upload utility for testing images

4 Dataset Description

The dataset used in this project is structured into training and testing subsets:

- **train/** – Contains images of people with masks (`with_mask`) and without masks (`without_mask`) used for model training.
- **test/** – Contains images used for validation and testing.
- **Number of images:** Training set – 1101 images, Validation set – 275 images.
- **Image size:** All images are resized to 224x224 pixels for input to the CNN.
- **Class distribution:** Binary classification – Mask and No Mask.

5 Methodology

5.1 Data Preprocessing

Preprocessing is performed to ensure consistent input to the neural network:

- **Rescaling:** Pixel values are normalized to a range of 0–1.
- **Data augmentation:** Includes horizontal flips, zoom, and shear to improve the model's ability to generalize.
- **Validation split:** 20% of data is set aside for validation.

5.2 Model Architecture

The project uses transfer learning with MobileNetV2 as a base model:

- **Base model:** Pre-trained MobileNetV2 on ImageNet dataset.
- **Trainable layers:** Last ten layers of MobileNetV2 are unfrozen for fine-tuning.
- **Custom layers:**
  - Global Average Pooling for dimensionality reduction.
  - Dense layer with 128 neurons and ReLU activation.
  - Dropout layer with rate 0.5 to prevent overfitting.
  - Output layer with sigmoid activation for binary classification.

5.3 Model Compilation

- **Optimizer:** Adam with a learning rate of 0.0001
- **Loss function:** Binary cross-entropy
- **Evaluation metric:** Accuracy

5.4 Model Training

- Trained on the augmented dataset for 5 epochs to balance accuracy and training time.
- Batch size set to 32.
- Validation performance monitored at each epoch to detect overfitting or underfitting.

5.5 Performance Evaluation

- Training and validation accuracy and loss are plotted for visual assessment.
- Accuracy trends show improvement over epochs, indicating effective fine-tuning of the pre-trained model.

5.6 Testing and Deployment

- Users can upload an image to the system.
- Image preprocessing ensures consistency with training inputs.
- Model predicts mask presence and outputs either "Mask" or "No Mask".
- The system also displays the uploaded image with prediction results.
- Can be extended to real-time detection using a webcam with OpenCV.

6 Results

- Training accuracy reached approximately 90%, while validation accuracy varied across epochs due to limited dataset size.
- Loss decreased consistently during training, indicating proper learning.
- Uploaded test images were correctly classified, demonstrating the model's capability to generalize to unseen images.

6.1 Output

7 Conclusion

This project successfully implements a practical face mask detection system using deep learning. Transfer learning with MobileNetV2 enables fast and effective feature extraction, and fine-tuning improves accuracy. The system can classify uploaded images and can be extended for real-time detection, making it a suitable project for public safety applications.

8 References

1. Snehit Vaddi, *Face Mask Detection Using Deep Learning*, GitHub Repository. `https://github.com/snehitvaddi/FaceMask-Detection-using-Deeplearning`
2. TensorFlow Documentation – `https://www.tensorflow.org`
3. Keras Documentation – `https://keras.io`
4. OpenCV Documentation – `https://opencv.org`
5. Sandler, Howard, et al., *MobileNetV2: Inverted Residuals and Linear Bottlenecks*, CVPR 2018