Face Mask Detection, Classification, and Segmentation

T2-24-25 - AIM 825 - Visual Recognition - Sec A

March 2025

Authors:
Ayush Kashyap (IMT2022129)
Rishit Mane (IMT2022564)
Rohan Rajesh (IMT2022575)

1 Introduction

1.1 Background and Motivation

Face mask detection is an essential task in modern-day computer vision, particularly for public health applications. With the rise of infectious diseases like COVID-19, ensuring compliance with mask-wearing protocols has become critical in various settings, including airports, shopping malls, offices, and public transport.

Automated face mask detection systems provide an efficient solution to monitor and enforce mask-wearing policies. By leveraging computer vision techniques, these systems can classify whether an individual is wearing a mask and segment the masked regions in an image.

1.2 Objective

The main goal of this project is to develop a robust model for detecting and classifying face masks using:

- **Traditional Machine Learning Approaches:** Using handcrafted features and classifiers like SVM and MLP.
- **Deep Learning Approaches:** Training Convolutional Neural Networks (CNNs) for improved classification and using U-Net for segmentation.

1.3 Challenges

Several challenges arise in face mask detection:

- Variability in lighting, background, and facial expressions.
- Partial occlusion due to different types of masks.
- Low-resolution and noisy images.

2 Dataset

2.1 Description

The dataset consists of labeled images of individuals with and without face masks. It is divided into two main parts:

- **Face Mask Classification Dataset:** Contains images categorized into two classes (mask/no mask). Available at: `https://github.com/chandrikadeb7/Face-Mask-Detection/tree/master/dataset`.
- **Masked Face Segmentation Dataset:** Includes images with corresponding pixel-wise segmentation masks. Available at: `https://github.com/sadjadrz/MFSD`.

2.2 Data Preprocessing

- Images are resized to 128×128 pixels for uniformity.
- Applied grayscale conversion and normalization.
- Augmentation techniques such as rotation, flipping, and brightness adjustments were used to improve model generalization.

3 Methodology

3.1 Task A: Classification Using Machine Learning (4 Marks)

We implemented two traditional classifiers using handcrafted features.

3.1.1 Feature Extraction

- **HOG (Histogram of Oriented Gradients)**: Captures texture and gradient-based features.
- **Sobel Edge Detection**: Highlights edges to distinguish masks from facial features.
- **Canny Edge Detection**: Detects strong and weak edges.

3.1.2 Machine Learning Models

- **Support Vector Machine (SVM)**: A supervised model that finds an optimal hyperplane for classification.
- **Multi-Layer Perceptron (MLP)**: A feedforward artificial neural network trained using backpropagation.

3.1.3 Performance Metrics

- Accuracy
- Precision
- Recall
- F1-score

3.1.4 Results

Model	Validation Accuracy
SVM	85.83%
MLP	90.59%

Table 1: Performance of Machine Learning Classifiers

3.2 Task B: Classification Using CNN (3 Marks)

We implemented a deep learning-based CNN classifier to improve accuracy.

3.2.1 CNN Architecture

- **Convolutional Layers**: Extract spatial features.
- **Pooling Layers**: Reduce dimensionality.
- **Fully Connected Layers**: Perform final classification.

3.2.2 Hyperparameter Tuning

- Learning rates: 0.001, 0.0005
- Optimizers: Adam, SGD
- Dropout: 0.3, 0.5

3.2.3 Results

Optimizer	Dropout	Validation Accuracy
Adam	0.3	95.2%
Adam	0.5	94.7%
SGD	0.3	91.5%
SGD	0.5	90.2%

Table 2: Performance of CNN with Different Hyperparameters

4 Face Mask Segmentation (3 Marks)

To further enhance face mask detection, we implemented a U-Net model for segmentation.

4.1 U-Net Architecture

- Encoder: Uses CNN layers to extract features.
- Decoder: Upsamples features to generate a segmentation mask.

4.2 Performance Metrics

- IoU (Intersection over Union)
- Dice Coefficient

4.3 Results

Metric	**Score**
IoU	0.87
Dice Coefficient	0.89

Table 3: Performance of U-Net Model

5 Conclusion

- Traditional ML models performed well, but CNN achieved higher accuracy.
- Adam optimizer with 0.3 dropout provided the best CNN performance.
- U-Net was effective in segmenting face masks.

6 GitHub Repository

The full implementation is available at:

`https://github.com/ayusharyakashyap/VR_Project1`