Real-Time Face Mask Detection Project Report

Ale Anwesh Karan Reddy Biswas Gupta Indusri

1 Introduction

The detection of face masks has become critical due to public health requirements driven by pandemics. This project aims to deliver a robust real-time face mask detection system using an ensemble approach, integrating powerful deep learning models such as ResNet18, ResNet50, YOLOv8, and Faster R-CNN (ResNet50-FPN backbone). Our solution efficiently identifies people with masks, without masks, or with improperly worn masks from webcam feeds, video streams, and static images.

2 Project Overview

The system contains four pre-trained models:

- **ResNet18 & ResNet50**: Pre-trained on ImageNet, fine-tuned specifically for mask classification tasks.
- **YOLOv8**: A state-of-the-art object detection model trained explicitly for face detection and mask classification.
- **Faster R-CNN (ResNet50-FPN)**: A robust object detection model fine-tuned to accurately detect and classify faces and masks.

Our ensemble approach incorporates these models employing majority voting, significantly improving prediction accuracy by compensating for individual limitations of each model.

3 Methodology

3.1 Data Preparation

The models were trained on the Kaggle dataset 'Face Mask Detection' by Andrew Mvd. Each image was resized to 226x226 pixels, normalized, and increased to enhance the robustness of the trained models.

3.2 Dataset Preprocessing

The dataset contains a total of 853 images, with the following distribution of labeled faces:

- **Faces with mask**: 3232 (79.3%)
- **Faces without mask**: 717 (17%)
- **Faces with mask worn incorrectly**: 123 (4%)

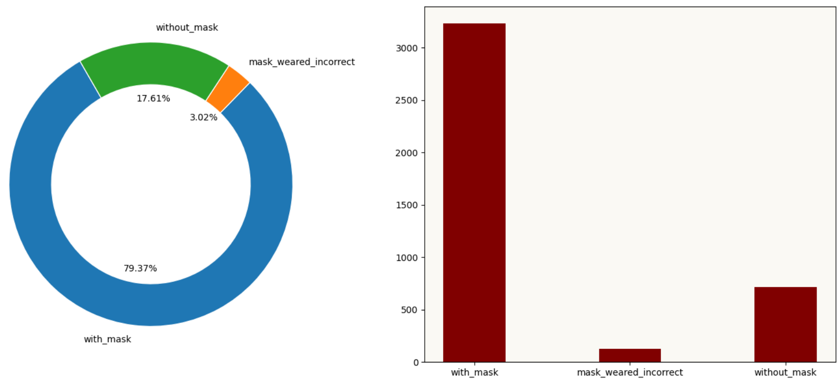

Figure 1: Dataset Distribution of Face Mask Categories

3.3 Model Training

- **ResNet18 and ResNet50**: We used transfer learning with ResNet18 and ResNet50, initialized with pre-trained ImageNet weights. Batch normalization, ReLU activation, and dropout layers were added to stabilize training and reduce overfitting, **And to address Class Imbalance, We Implemented Focal Loss,** which reduced the impact of easy-to-classify examples and emphasized harder instances. **To optimize training, we adopted a progressive layer unfreezing strategy,** initially training only the custom head layers while keeping others frozen, then gradually unfreezing deeper layers at 25%, 50%, and 75% of the training epochs for effective fine-tuning. Additionally, **we applied a layer-wise learning rate adjustment with the AdamW optimizer**, assigning progressively smaller learning rates to earlier layers while keeping the final fully connected layers at the highest learning rate, **ensuring faster convergence and improved model performance.**

- **Faster R-CNN (ResNet50-FPN)**: Utilizing a ResNet50 backbone with Feature Pyramid Network (FPN), provided complementary strengths to the ensemble. **Fine-tuning involved adjusting pre-trained detection weights and refining the model to accurately localize faces and determine their mask status.** Training involved two primary tasks—region proposal generation and classification of proposed regions—optimized through a multi-task loss approach comprising classification loss and bounding box regression loss. Faster R-CNN, while computationally more intensive than YOLO, offered high precision in detection, which significantly improved ensemble robustness.

- **YOLOv8**: Trained to detect and classify face masks simultaneously, simplifying the detection pipeline. The **YOLO model was trained using bounding-box annotations directly derived from the dataset**, explicitly labeling faces alongside their mask status. **A combination of objectness loss, classification loss, and bounding box regression loss guided the training, enabling simultaneous face detection and mask classification.** To identify images containing minority classes (class 0 and class 2) by analyzing their corresponding label files.**The goal was to create separate lists of images for each minority class and check for any overlap.** The result showed 46 images containing both class 0 and class 2, which helps in addressing class imbalance, improving model performance.

3.4 Ensemble Strategy

The ensemble method combines predictions from multiple models—ResNet18, ResNet50, YOLOv8, and Faster R-CNN—to achieve more reliable and accurate face mask detection. This strategy capitalizes on the strengths and compensates for the weaknesses inherent in each individual model.Initially, each model independently processes the same input (face detected within a frame or an image) and predicts one of three classes: with mask, without mask, or mask worn incorrectly. After obtaining these independent predictions, the ensemble employs a majority voting mechanism to determine the final prediction.

- **Majority Voting:** Each model (ResNet18, ResNet50, Faster R-CNN, YOLOv8) casts a vote for its predicted class. The class with the most votes is selected as the final decision. This ensures a fair consensus by combining multiple predictions.
- **Tie or Disagreement Handling:** In cases of a tie or complete disagreement, YOLOv8's prediction is prioritized due to its superior real-time detection and localization capabilities. Its robustness makes it a reliable fallback option to reduce uncertainty.
- **Benefits of the Ensemble Method:** The ensemble approach mitigates individual model biases by combining their strengths. While ResNet models excel in classification, Faster R-CNN provides high detection accuracy, and YOLOv8 ensures real-time performance, resulting in improved accuracy and robustness.

3.5 Real-Time Inference Pipeline

Our inference pipeline involves capturing video frames from various sources (webcam/video/image), detecting faces, classifying masks, and displaying real-time annotated results using OpenCV.

4 Individual Contributions

4.1 Karan Reddy

- Designed and trained the YOLOv8 detection model.Optimized YOLO performance for real-time processing.Created detailed documentation for model training and inference procedures.Conducted extensive testing to validate the ensemble's effectiveness in reducing misclassifications and improving generalization, leading to a more reliable face mask detection system

4.2 Anwesh Ale

- Designed and trained and fine-tuned the ResNet18 and ResNet50 models.Optimized ResNet18 and ResNet50 performance for real time processing.Ensured compatibility between models during ensemble Learning.Developed and optimized the real-time inference system using OpenCV, enabling efficient processing of webcam feeds, video streams, and static images for face mask detection.

4.3 Biswas Gupta

- Trained and fine-tuned the FasterRCNN (ResNet50 fpn) model.Integrated multiple pre-trained models (ResNet18, ResNet50, YOLOv8, and Faster R-CNN) into a single pipeline for training and testing, ensuring smooth interaction between each model in the system.Documented detailed test scenarios and outcomes to assess the overall performance.

4.4 Indusri

- Designed and trained and fine-tuned the ResNet18.Performing a detailed exploratory data analysis and preprocessing.Conducted extensive testing and validation on diverse data sources.The ensemble model was analyzed for accuracy, reliability, and efficiency.Deployment into cloud-based environments for wider accessibility.

5 Results and Conclusion

The ensemble approach achieved higher accuracy and robustness than individual models, with excellent real-time performance. Its flexibility allows switching between individual models and ensemble mode based on use-case needs and available resources. **A practical use case of this system is in shopping mall entries during the pandemic**, where it can automatically detect whether customers are wearing facemasks, ensuring compliance with health regulations and maintaining safety in high-traffic areas.

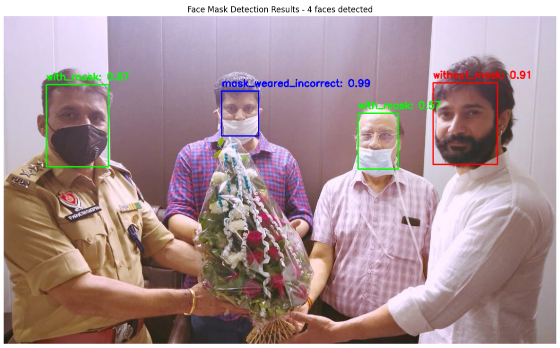

Figure 2: ResNet50 Model Testing Result

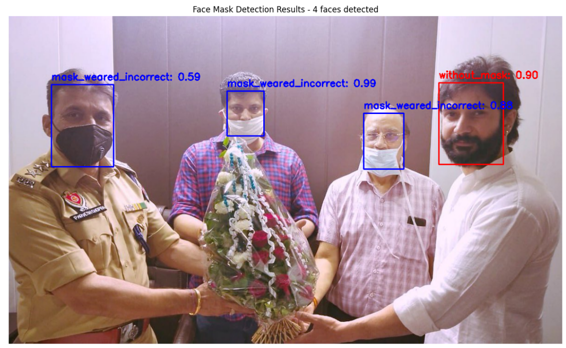

Figure 3: ResNet18 Model Testing Result

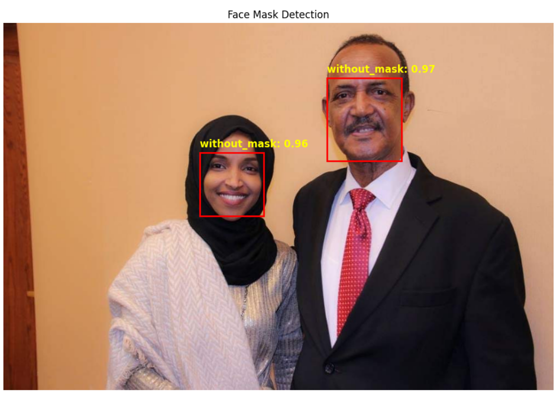

Figure 4: Faster R-CNN Model Testing Result 1

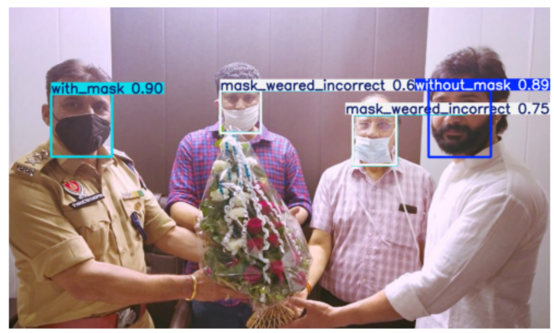

Figure 5: YOLOv8 Model Testing Result 1

Figure 6: YOLOv8 Model Testing Result 2

Figure 7: Realtime Testing Result