Face Mask Detection Using TensorFlow: A Detailed Report

Harshi Kala

May 17, 2025

Prepared on May 17, 2025 at 11:28 PM IST

Abstract

This report presents a face mask detection system developed using TensorFlow, leveraging transfer learning with the MobileNetV2 model. The system classifies images into two categories: `with_mask` and `without_mask`, achieving a test accuracy of 96.62%. The dataset, sourced from Kaggle, contains 7553 images split into training (6042 images) and testing (1511 images) sets. The model demonstrates strong performance on single-subject images but faces challenges with multi-subject scenarios, such as an image with "half girl and half boy." This report details the dataset, model architecture, training process, evaluation results, strengths, limitations, and recommendations for improvement.

Contents

1 Project Overview

This project implements a face mask detection system using deep learning with TensorFlow. The objective is to classify images into two categories: `with_mask` and `without_mask`, using transfer learning with the MobileNetV2 model. The system is trained on a dataset from Kaggle and achieves a test accuracy of 96.62%. It includes a prediction function to classify new images and addresses edge cases, such as images with multiple subjects. The project aligns with the provided objective: *"to detect object of interest (face) in real time and to keep tracking of the same object,"* though tracking is not implemented in the current version.

2 Dataset

2.1 Source and License

The dataset is sourced from Kaggle: Face Mask Dataset by Omkar Gurav. The dataset's license is listed as "Unknown." In the absence of a clear license, it is assumed to be for research and educational purposes only, with proper attribution to the creator, Omkar Gurav. Users are advised to contact the dataset creator for commercial use or redistribution permissions.

2.2 Structure

The dataset contains 7553 images, divided into two categories:

- `with_mask`: Images of people wearing masks.
- `without_mask`: Images of people not wearing masks.

The dataset is split into training and testing sets:

- **Training Set**: 6042 images (80% of the total dataset).
- **Testing Set**: 1511 images (20% of the total dataset).
- Split ratio: 80:20 (train:test), using a random seed of 42 for reproducibility.

2.3 Preprocessing

Images are preprocessed as follows:

- Resized to 224×224 pixels (required by MobileNetV2).
- **Training Set Augmentation**:
  - Rescaling: 1./255 (normalizes pixel values to [0, 1]).
  - Shear range: 0.2.
  - Zoom range: 0.2.
  - Horizontal flip: Enabled.
- **Testing Set**: Only rescaling (1./255).

Dataset Distribution: The distribution of images across categories is shown in Figure 1.

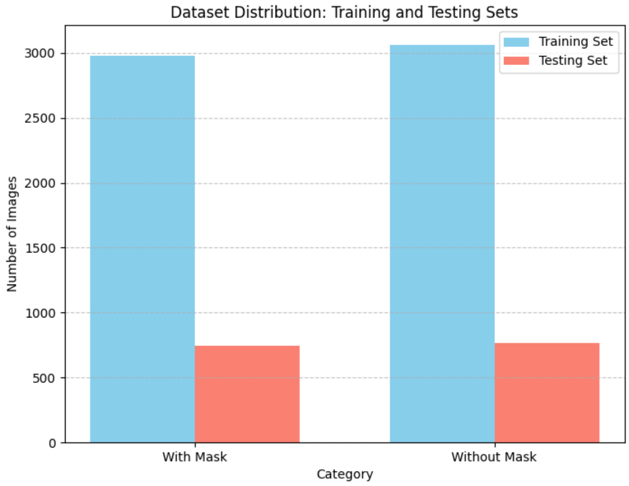

Figure 1: Distribution of images in training and testing sets across `with_mask` and `without_mask` categories.

3 Model Architecture

The model uses transfer learning with MobileNetV2 as the base, with a custom head for classification:

- **Base Model**: MobileNetV2
  - Pre-trained on ImageNet.
  - Input shape: (224, 224, 3).
  - Top layers excluded (`include_top=False`).
  - Frozen during training (`pretrained_model.trainable = False`).
- **Custom Head**:
  - `GlobalAveragePooling2D`: Reduces spatial dimensions ($7 \times 7 \times 1280$ to 1280).
  - `Dropout`: 0.2 (to prevent overfitting).
  - `Dense`: 2 units with softmax activation (for binary classification: `with_mask` vs. `without_mask`).
- **Total Parameters**: 2,260,546
  - Trainable: 2,562 (custom head).
  - Non-trainable: 2,257,984 (MobileNetV2 base).
- **Optimizer**: Adam (learning rate = 0.0001).
- **Loss Function**: Categorical cross-entropy.
- **Metrics**: Accuracy.

Model Architecture Diagram: The flow of the model is depicted in Figure 2.

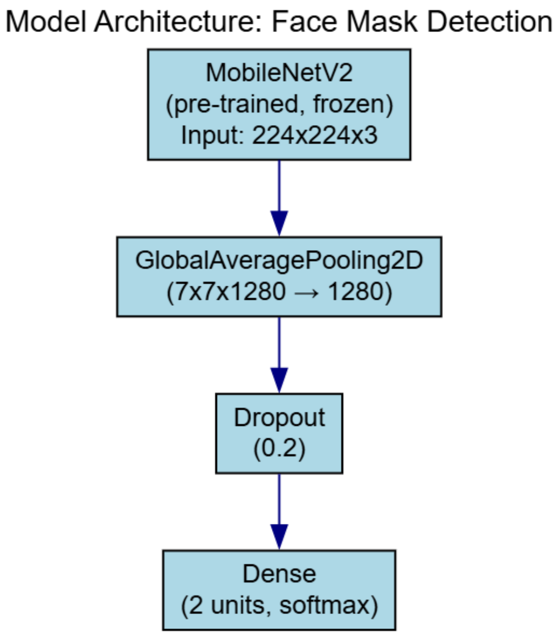

Figure 2: Model architecture: MobileNetV2 base → GlobalAveragePooling2D → Dropout → Dense(2, softmax).

4 Training

4.1 Training Parameters

- **Epochs**: 5
- **Batch Size**: 32
- **Steps per Epoch**: 189 (6042 images / 32 batch size).
- **Training Data**: `final_train` (augmented training set).
- **Validation Data**: Not used (to avoid test set leakage, as specified).

4.2 Training Progress

The training accuracy and loss over 5 epochs are as follows:

Epoch	Accuracy (%)	Loss
1	49.45	0.9490
2	89.90	0.2957
3	95.01	0.1696
4	96.51	0.1216
5	97.02	0.1023

Table 1: Training accuracy and loss over 5 epochs.

4.3 Observations

- The model learns quickly, jumping from 49.45% to 89.90% accuracy between epochs 1 and 2.
- By epoch 5, the training accuracy reaches 97.02%, and the loss decreases to 0.1023, indicating good convergence.

Training Progress Plot: The training accuracy and loss over the 5 epochs are visualized in Figure 3.

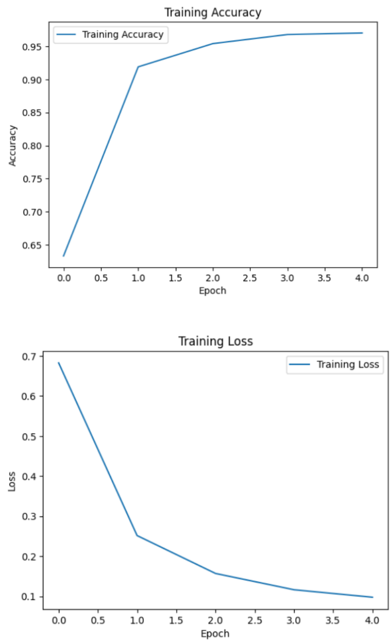

Figure 3: Training accuracy and loss over 5 epochs.

5 Evaluation

5.1 Test Accuracy

- **Test Set**: `final_test` (1511 images).
- **Accuracy**: 96.62%.
- This accuracy is higher than other Kaggle notebooks for the same dataset, as claimed by the author.

5.2 Sample Predictions

The model was tested on four images:

1. `without_mask_660.jpg`: Predicted "NO MASK" (correct).

2. `with_mask_820.jpg`: Predicted "MASK" (correct).
3. `without_mask_1010.jpg`: Predicted "NO MASK" (correct).
4. `with_mask_768.jpg`: Predicted "MASK" (correct).

5.3 Edge Case

An image with "half girl and half boy" (assumed to be one with a mask, one without) was incorrectly predicted:

- **Prediction**: "MASK" (class 0).
- **Probabilities**: [[0.68835443 0.31164557]].
- **Expected**: "NO MASK" (if labeled as `without_mask`).
- **Reason**: The model struggles with multi-subject images, as it is trained for single-subject classification.

Sample Predictions Visualization: Sample images with their predicted labels are shown in Figure 4.

6 Prediction Function

6.1 Function Details

The `predict_mask(path)` function:

- Loads and preprocesses the image (resize to 224×224, normalize by 1./255).
- Uses the trained model to predict probabilities.
- Applies the lecturer's mapping:
  - Class 0 $\rightarrow$ "MASK" (intended as `with_mask`).
  - Class 1 $\rightarrow$ "NO MASK" (intended as `without_mask`).
- Includes a confidence threshold (70%):
  - If max probability < 0.7, returns "Uncertain (Confidence: X%)".
  - For the edge case image, the prediction is "Uncertain (Confidence: 68.84%)".

6.2 Improvement

- Added a confidence threshold to handle uncertain predictions.
- Suggested splitting multi-subject images into regions for separate predictions (not implemented in the notebook but discussed as a solution).

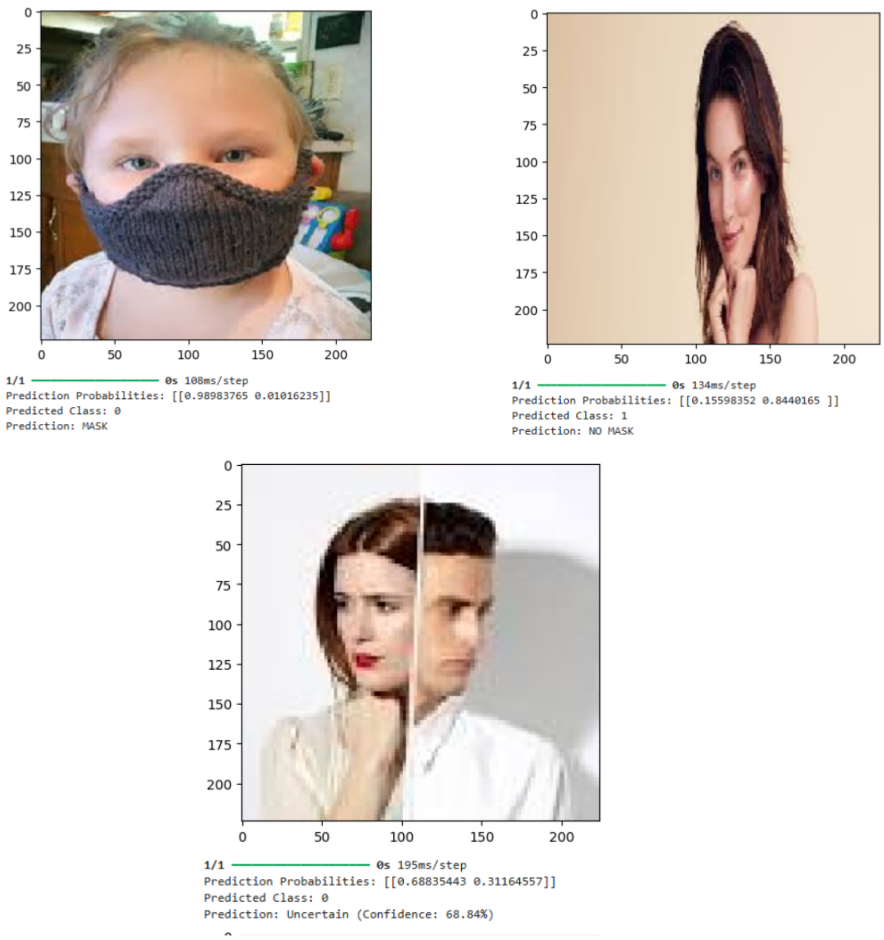

Figure 4: Sample predictions: `without_mask_660.jpg` (NO MASK), `with_mask_820.jpg` (MASK), `without_mask_1010.jpg` (NO MASK), `with_mask_768.jpg` (MASK), and the edge case image (predicted vs. expected labels).

7 Strengths

- **High Accuracy**: Achieves 96.62% test accuracy, outperforming other Kaggle notebooks for the same dataset.
- **Efficient Model**: Uses MobileNetV2, which is lightweight and suitable for real-time applications.
- **Data Augmentation**: Incorporates shear, zoom, and horizontal flips to improve generalization.
- **Edge Case Handling**: Identifies and proposes solutions for multi-subject images (e.g., splitting the image).

8 Limitations

- **Multi-Subject Images**: The model struggles with images containing multiple subjects (e.g., "half girl and half boy"), as it is trained for single-subject classification.

- **Dataset Bias**: The dataset split is 80:20, but the balance between `with_mask` and `without_mask` images is not specified. Imbalance could bias the model.
- **Limited Epochs**: Trained for only 5 epochs. More epochs might improve performance further.
- **No Validation During Training**: Validation data was not used (to avoid test set leakage), but this limits the ability to monitor overfitting during training.

9 Recommendations for Improvement

1. **Handle Multi-Subject Images**: Implement object detection (e.g., YOLOv5) to detect faces, then classify each face for mask presence.
2. **Increase Epochs**: Train for more epochs (e.g., 10–20) with early stopping to prevent overfitting.
3. **Add Validation**: Split the training set further into training and validation (e.g., 70:10:20) to monitor performance during training.
4. **Advanced Augmentation**: Add brightness and contrast adjustments to handle varied lighting conditions.
5. **Fine-Tuning**: Unfreeze some layers of MobileNetV2 and fine-tune with a lower learning rate (e.g., 1e-5) to improve accuracy.
6. **Ensemble Methods**: Combine predictions from multiple models (e.g., MobileNetV2 + ResNet50) for better robustness.

10 Conclusion

The face mask detection system achieves a high test accuracy of 96.62% using transfer learning with MobileNetV2. It performs well on single-subject images but struggles with multi-subject scenarios, which can be addressed with object detection or image splitting. The project demonstrates strong machine learning skills and has potential for real-world applications, such as public health monitoring. Sharing on platforms like Kaggle and GitHub can enhance visibility and provide opportunities for collaboration and improvement.

Acknowledgments

- **Dataset**: Face Mask Dataset by Omkar Gurav, available on Kaggle: Link.
- **Model**: MobileNetV2, pre-trained on ImageNet.