# Face Mask Detection Using YOLO for Multi-Face Object Detection

Project Midterm Report

*Team Members: Gabriel Ferreira*

**Abstract**

This project aims to develop a computer vision system capable of detecting face masks, a crucial task during the COVID-19 pandemic and in post-pandemic setting where mask-wearing remains relevant. This project aims to implement the system using YOLO (You Only Look Once) and explore its extension to real-time detection using OpenCV. The Face Mask Detection Dataset from Kaggle is used for training and evaluation. Initially, a Convolutional Neural Network (CNN) was used as a baseline to classify images as "with mask", "without mask", or "with mask incorrectly". However, through this initial exploration phase, it became clear that a classification model like CNN, which outputs one label per picture, is limited in detecting multiple individuals in the same picture. As a result, I transitioned to using YOLO, a state-of-the-art object detection model, which allows for detecting multiple faces in an image and classifying each face as either one of the labels I just mentioned above.

## 1 Introduction

Face mask detection has become a crucial task during the COVID-19 pandemic, especially in public spaces where compliance with mask-wearing protocols is essential for preventing the spread of the virus. Beyond the pandemic, monitoring mask usage remains relevant in various sectors such as healthcare, public transportation, and crowded environments where safety measures are required. To address this need, computer vision and machine learning models can be leveraged to automate the detection of face masks in real-time.

Initially, this project explored the use of a CNN for classifying images into three categories: "with mask," "without mask," and "with mask incorrectly." The CNN provided a good starting point, achieving reasonable accuracy in single-label classification tasks. However, during this phase, a key limitation of CNNs became apparent: they can only classify one label per image. In scenarios where multiple people are present in a single frame, this restriction significantly limits the model's ability to accurately detect and classify all individuals in the image.

To overcome this challenge, I transitioned to using YOLO, as mentioned before, a state-of-the-art object detection model. Unlike CNNs, YOLO is designed to detect multiple objects within an image and classify each one independently. This feature makes YOLO a good fit for face mask detection in real-world environments, where multiple faces may need to be detected and classified simultaneously. YOLO's efficiency and speed are particularly valuable for real-time applications, where quick decisions are needed, such as in video surveillance systems. The integration of YOLO with OpenCV for real-time detection is a tentative step in this project, as I aim to explore its capabilities for live video analysis.

The dataset used for this project is the Face Mask Detection Dataset from Kaggle. It consists of a diverse set of images featuring individuals with and without masks, labeled into three categories. The dataset includes various lighting conditions, angles, and face orientations, providing a robust foundation for training the model. The initial experiments with CNN allowed us to gain insights into the dataset and refine our approach for more complex tasks like multi-object detection.

By leveraging YOLO, the goal is to develop a model that can not only detect whether individuals are wearing face masks but also handle multiple detections within the same frame. The expected outcome is a real-time, high-accuracy system that can be deployed in environments where monitoring mask usage is critical. Future work will focus on fine-tuning the model for real-time performance and testing its capabilities in dynamic settings.

## 2 Related Work

YOLO has been widely applied in various object detection tasks due to its real-time detection capabilities. One notable example is its application in automatic license plate recognition, as described in a recent study (Chen, J., & Su, Y. (2020)). In that project, YOLO was used to detect and extract license plates from images in a fast and efficient manner, even in challenging conditions like low lighting or occlusions. The model's ability to handle these variations and its speed made it an ideal choice for a task where real-time performance is critical.

In the license plate recognition use case, YOLO's detection framework allowed the model to both localize the plates and classify them in a single pass through the network. This is particularly useful in real-world environments where vehicles may be moving, and the system needs to process frames quickly to capture the license plates in different positions and lighting conditions. The YOLO model used in this project was able to recognize license plates with high accuracy and speed, making it an optimal solution for traffic monitoring systems.

Given YOLO's success in detecting and classifying objects in real-time in the license plate recognition task, it is a strong candidate for the face mask detection problem. Similar to license plates, faces in public spaces may appear in various orientations, positions, and lighting conditions. YOLO's ability to process images efficiently and detect multiple objects, faces in our case, in a single frame makes it well-suited for our project, where detecting and classifying the mask status of multiple individuals is critical.

## 3 Methodology

Our approach includes the following key steps:

### 3.1 Dataset Collection

This project uses the Face Mask Detection Dataset from Kaggle, which contains images labeled as "with mask," "without mask," and "with mask incorrectly." The dataset is pre-split into training and testing sets. It provides a diverse range of images featuring people from various backgrounds, angles, and lighting conditions, ensuring a robust model.

### 3.2 Initial Exploration with CNN

Initially, a CNN was used as a baseline to classify images into the three categories. The CNN architecture consisted of multiple convolutional layers followed by max pooling and dense layers. While the CNN achieved reasonable accuracy in classifying images, it became evident that the CNN could only output one label per image, making it unsuitable for detecting multiple individuals in a single image. This learning informed the decision to move towards a more suitable model for object detection.

### 3.3 Transition to YOLO for Object Detection

YOLO was chosen as the main model for this project due to its ability to detect multiple objects (faces) within an image and classify each face individually. YOLO v5/v8 is particularly well-suited for this task due to its high speed and efficiency. I plan to fine-tune a pretrained YOLO v5 or v8 model on our dataset to detect faces and classify them into the three labels. The bounding boxes provided by YOLO will allow for the detection of multiple faces in each image, which was a limitation of the initial CNN approach.

### 3.4 Data Preprocessing

Data preprocessing involves resizing images, normalizing pixel values, and applying data augmentation techniques such as rotation, flipping, and zooming. Augmentation is used to artificially increase the

dataset size and improve model generalization.

### 3.5 Training the YOLO Model

The YOLO v5 or v8 model will be retrained using transfer learning. I will utilize the pretrained weights and fine-tune the model on our face mask detection dataset. The training process will involve optimizing the model's learning rate, batch size, and other hyper-parameters to achieve the best performance.

### 3.6 Real-Time Detection (Tentative)

Once the YOLO model has been trained, I plan to explore its integration with OpenCV to enable real-time detection. This will involve loading the trained YOLO model and applying it to video frames captured by a webcam, with the goal of detecting and classifying multiple faces in real-time. I will use OpenCV functions to process the video and apply the model's predictions on a frame-by-frame basis.

### 3.7 Evaluation Metrics

To evaluate the performance of YOLO, I will use mean Average Precision (mAP), a standard metric for object detection models. I will also evaluate the accuracy, precision, recall, and F1-score of the model for each class ("with mask," "without mask," and "with mask incorrectly"). Comparisons will be made between YOLO and the baseline CNN model to highlight the benefits of using an object detection approach over a classification model.

## 4 Preliminary Results

**Initial CNN results** indicated that the model could classify individual images with reasonable accuracy. Specifically, the CNN was trained to distinguish between three classes: "with mask," "without mask," and "with mask incorrectly." However, a significant limitation of the CNN became apparent: the model could only classify one label per image. In real-world scenarios where multiple people may be present in a single image, this approach is insufficient, as it fails to detect and classify multiple faces individually.

The confusion matrix from the CNN baseline model showed some misclassifications, particularly between "with mask" and "with mask incorrectly." Overall, the CNN achieved an accuracy of 75%, which was promising for a baseline model but still insufficient for detecting multiple people in a single frame.

**Transition to YOLO:** To address this limitation, I plan to transition to YOLO for object detection as previously mentioned. YOLO's ability to detect multiple faces and classify each face individually will overcome the constraint of the CNN model. However, YOLO has not yet been implemented in this phase of the project. Future work will involve retraining YOLO to detect and classify multiple faces in a single image and to improve overall performance.

Below is a summary of the CNN baseline results:

| Model | Accuracy | Capability |
|---|---|---|
| CNN (baseline) | 75% | Can only classify one label per image |
| YOLO (planned) | N/A | Detects multiple faces and classifies each face separately |

Table 1: Summary of CNN baseline results and future YOLO implementation.

## 5 Future Plan

Moving forward, the primary goal is to retrain the YOLO v5 or v8 model to detect multiple individuals in images and classify their mask-wearing status. After successfully training the model, I will focus on implementing real-time detection using OpenCV and testing its performance in dynamic environments. **This real-time application is tentative but is seen as a key extension of the project.** Optimization for speed and efficiency will be critical, especially when handling live video feeds. The final phase may include exploring deployment options for web or mobile applications to provide real-world utility.

## 6 References


1. Kaggle, "Face Mask Detection Dataset," available at: https://www.kaggle.com/datasets/andrewmvd/face-mask-detection/code.

2. Chen, J., & Su, Y. (2020). YOLO-Based License Plate Recognition Algorithm. Journal of Computer Engineering and Applications, 45(2).