# Sager Autonomous Robotics

**Prepared By:**

**Sanad Hamarneh**

**21/7/2024**

# 1.1Introduction

Masks play a crucial role in protecting the health of individuals against respiratory diseases, as it is one of the few precautions available for COVID-19 in the absence of immunization.We will create a model to detect people wearing masks, not wearing them, or wearing masks.

# 1.2 Dataset

## 1.2.1 Source

The images for this project have been sourced from Kaggle. The dataset contains 853 images, each annotated with bounding boxes in the PASCAL VOC format. The images are categorized into three classes:

- **With mask**
- **Without mask**
- **Mask worn incorrectly**

## 1.2.2 Description

The dataset used for training and evaluating the model includes:

- **Number of Images:** 125
- **Annotation Format:** YOLO (You Only Look Once) format, which is efficient for object detection tasks.
- **Class Distribution:**
    - **With mask:** Images where individuals are wearing masks correctly.
    - **Without mask:** Images where individuals are not wearing masks.

## 1.2.3 Annotation Process

The selected images were manually annotated to ensure accurate bounding boxes around the detected objects. This process involved:

- Using annotation tools(labelimg) to draw bounding boxes around individuals with and without masks.
- Converting the annotations into the YOLO format, which includes the class label and normalized coordinates of the bounding boxes.

## 1.2.4 Data Preprocessing

**Resize:**

- Resize all images to 640x640 pixels.
- Ensures consistency in input dimensions for the model.

**Normalize the images:**

- Normalize the pixel values to a range of [0, 1].
- Helps in faster convergence during training by maintaining uniform pixel value distribution.

**Random Cropping:**

- Randomly crop sections of the image to introduce variability and robustness.

**Horizontal Flipping:**

- Flip images horizontally to augment the dataset and improve generalization.

**Rotation:**

- Rotate images at random angles to help the model handle different orientations.

**Brightness and Contrast Adjustment:**

- Randomly adjust the brightness and contrast to simulate different lighting conditions.

**Scaling and Translation:**

- Apply scaling and translation transformations to vary the size and position of objects within the images.

# 1.3 Model Selection

## 1.3.1 Overview of YOLOv8

YOLOv8 is a state-of-the-art object detection framework known for its speed and accuracy. incorporating advancements in deep learning and computer vision to enhance detection performance while maintaining real-time processing capabilities. YOLOv8 offers various model sizes to balance between performance and computational efficiency.

## 1.3.2 YOLOv8n (Nano)

**YOLOv8n**, or the Nano version, is designed to be extremely lightweight, making it suitable for deployment on devices with limited computational resources, such as drones and mobile devices. Key features include:

- **Model Architecture:** Simplified architecture with fewer parameters and layers, enabling faster inference times.
- **Speed:** Optimized for real-time detection with minimal latency.
- **Applications:** Ideal for scenarios where speed is critical and computational resources are constrained, such as real-time monitoring and embedded systems.

## 1.3.3 YOLOv8s (Small)

**YOLOv8s**, or the Small version, offers a balanced trade-off between speed and accuracy. It is more robust than the Nano version while still being efficient enough for deployment on moderately powered devices. Key features include:

- **Model Architecture:** Enhanced architecture with more layers and parameters compared to YOLOv8n, providing better detection accuracy.
- **Performance:** Improved accuracy while maintaining relatively fast inference times.
- **Applications:** Suitable for applications requiring a balance between detection precision and speed, such as detailed surveillance and advanced monitoring systems.

# 1.4 Model Training

The training process involved experimenting with various configurations of the YOLOv8n and YOLOv8s models. The primary goal was to achieve a high level of accuracy while addressing class imbalance and optimizing the number of epochs. Below is a summary of each model's configuration and performance:

**Model 1: Initial Training with YOLOv8n**

- **Configuration:**
  - Model: YOLOv8n
  - Epochs: 15
  - Initial dataset without class imbalance adjustments
  - 125 Images
- **Results:The results will be in a separate file**
- **Challenges:** The initial model showed issues with class imbalance, leading to lower accuracy for the "Without mask" class.

**Model 2: Adjusted Data with YOLOv8n**

- **Configuration:**
  - Model: YOLOv8n
  - Epochs: 30
  - Initial dataset without class imbalance adjustments
- **Results: The results will be in a separate file**
- **Improvements:** The performance improved afterIncresing epochs

**Model 3: Extended Training with YOLOv8n**

- **Configuration:**
  - Model: YOLOv8n
  - Epochs: 30
  - Adjusted dataset with class balance
- **Results:The results will be in a separate file**
- **Observations:** The extended training period and balanced dataset resulted in better accuracy and more reliable detection, especially for the "Without mask" class.

**Model 4: Optimized Training with YOLOv8s**

- **Configuration:**
  - Model: YOLOv8s
  - Epochs: 150
  - Further optimized data augmentation techniques
- **Results:The results will be in a separate file**
- **Outcome:** The optimized model achieved the highest accuracy, with a significant reduction in false positives and false negatives.