

FACE MASK DETECTION

Jaret Sanchez

ITAI 1378 – computer vision & AI

Tier 1

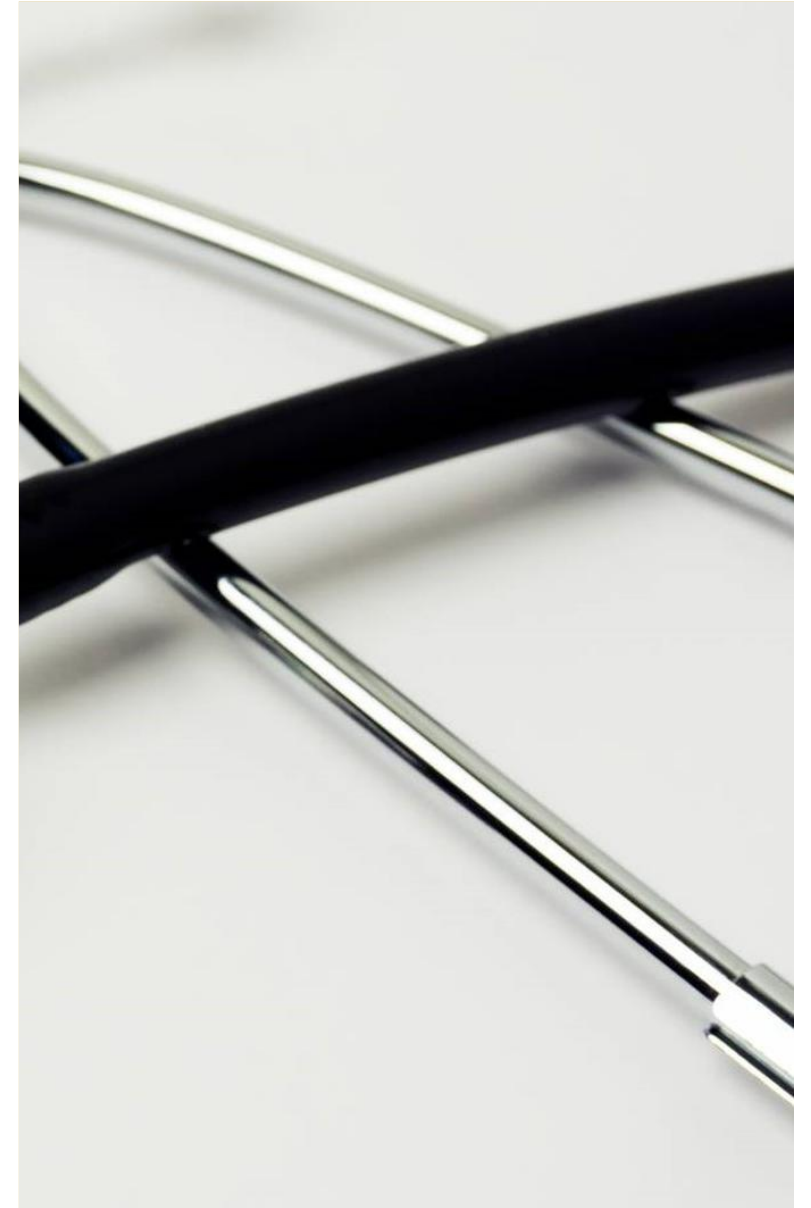


THE PROBLEM

During the public health crisis, the whole world had to face wearing a face mask was crucial to avoid exposure. It was challenging to ensure that people wear face masks properly in public spaces.

The places that require mask compliance include hospitals, schools, airports, and government facilities.

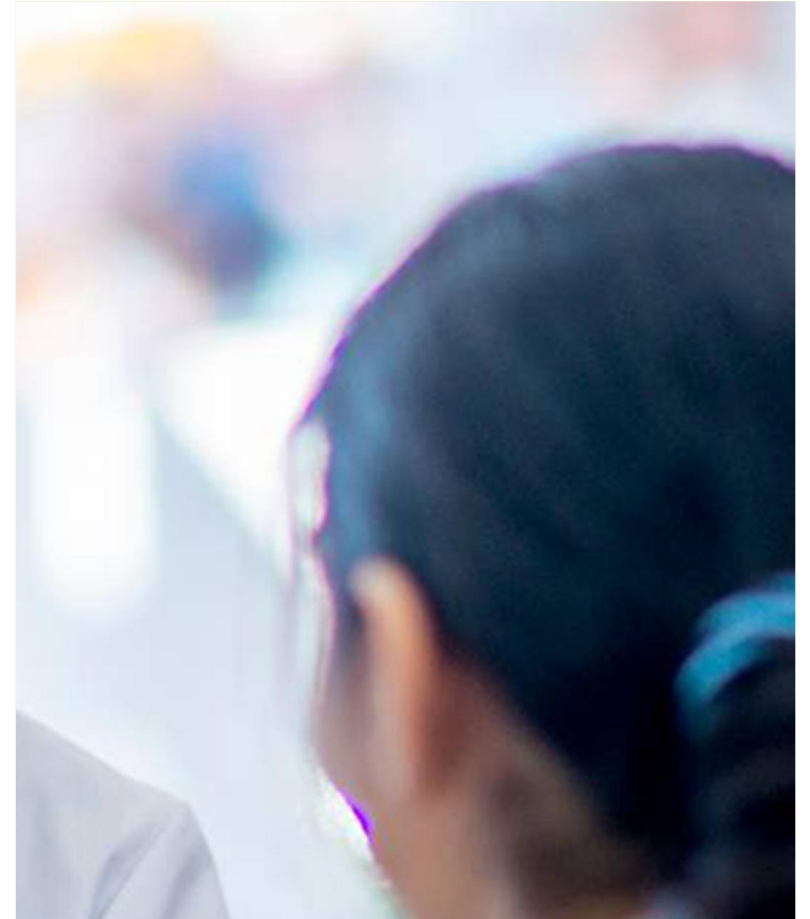
This issue is important because manual monitoring is prone to human error; automated detection will improve face mask compliance and safety.



PROPOSED SOLUTION

My system will automatically detect if a person is wearing a mask correctly, not wearing a mask, and if the mask is being worn incorrectly.

Automated face mask detection will improve compliance and safety in public environments.



TECHNICAL APPROACH

Technique: Object Detection

Model: YOLOv8

Framework: PyTorch

The reason why I chose to use YOLOv8 is because this real-time object detection model is the best option for detecting multiple faces and mask types in one frame.



DATA PLAN

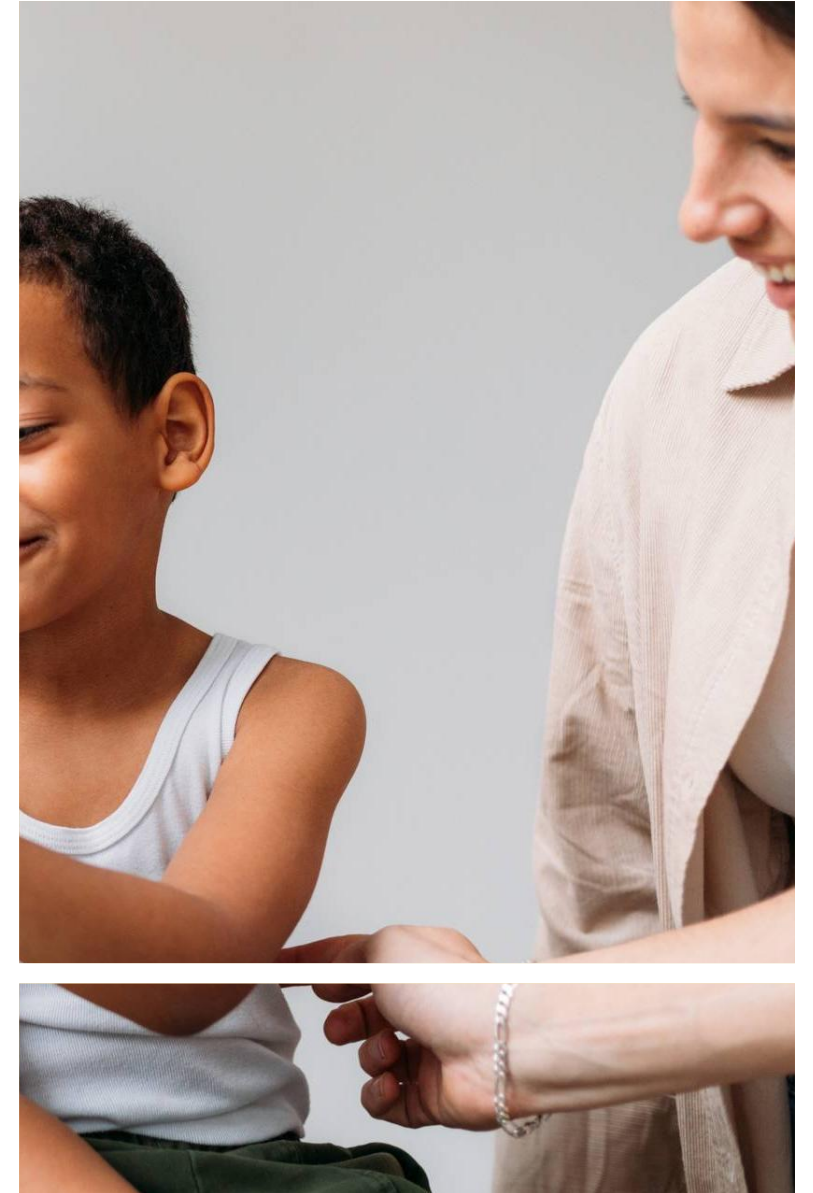
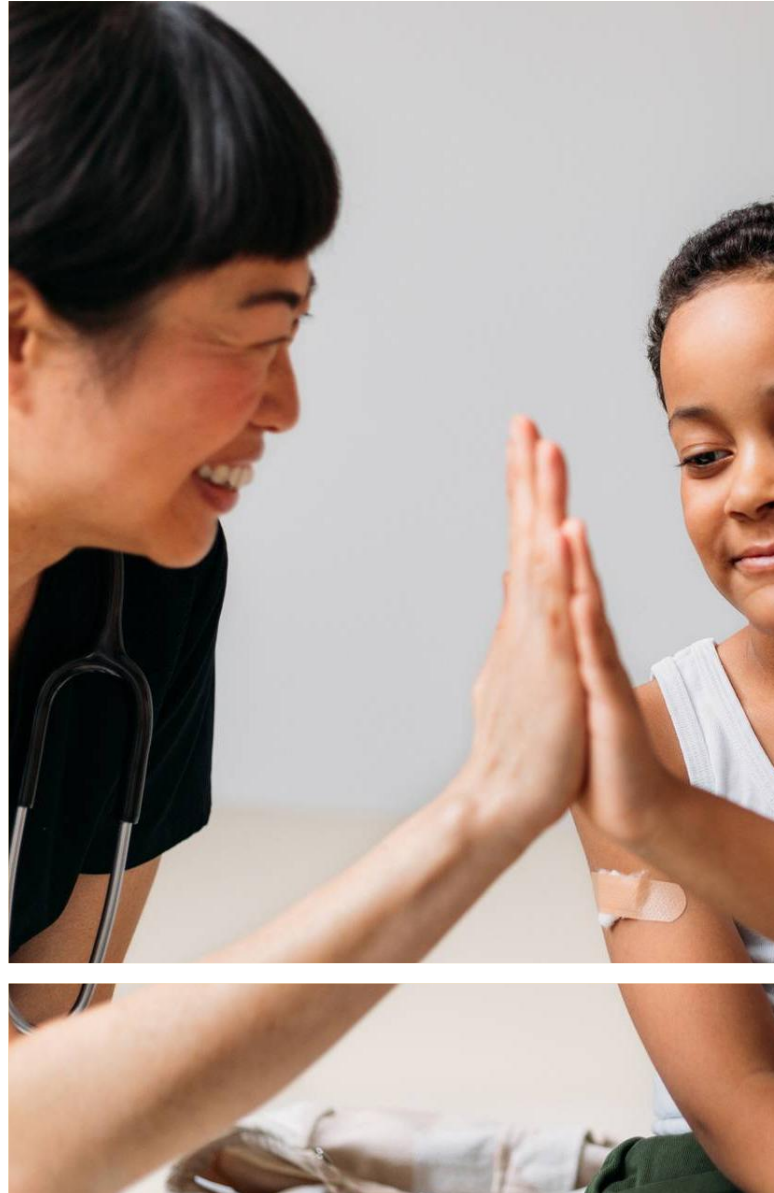
Source: Kaggle – Face Mask Detection Dataset

Size: ~7,000 labeled images

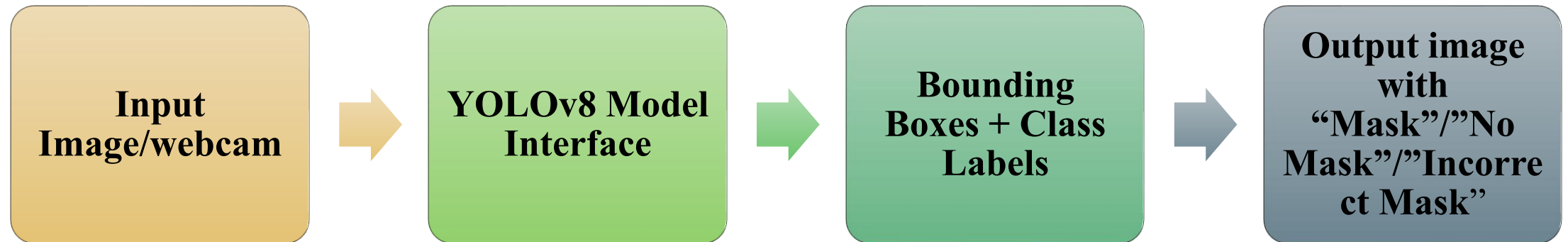
Labels: With Mask, Without Mask, Incorrectly Worn Mask

Preparation:

- Download and verify dataset
- Split into training, validation, and testing sets
- Convert annotations into YOLO format if needed



SYSTEM DIAGRAM





SUCCESS METRICS

Metric	Target
Detection Accuracy	$\geq 90\%$
Inference Speed	< 1 second per image

WEEK-BY-WEEK PLAN

Week	Task	Milestone
I0 (Oct 30)	Get dataset, set up Colab & YOLOv8	Dataset ready
I1 (Nov 6)	Fine-tune YOLOv8 on mask dataset	Model training
I2 (Nov 13)	Test and improve accuracy	Model optimized
I3 (Nov 20)	Create real-time demo	Demo ready
I4 (Nov 27)	Final evaluation and documentation	Project ready
I5 (Dec 4)	Present project	Presentation day



CHALLENGES & BACKUP PLANS

Risk	Solution
Dataset too small	Use additional Roboflow datasets
Model accuracy below target	Add data augmentation
GPU limits in Colab	Use local runtime
Label imbalance	Apply class weighting

RESOURCES NEEDED

Resource	Options
Compute	Google Colab
Frameworks	PyTorch + Ultralytics YOLOv8
Dataset	Kaggle Face Mask Detection
Estimated Cost	\$0
Tools	OpenCV, Matplotlib