



FACE MASK DETECTION

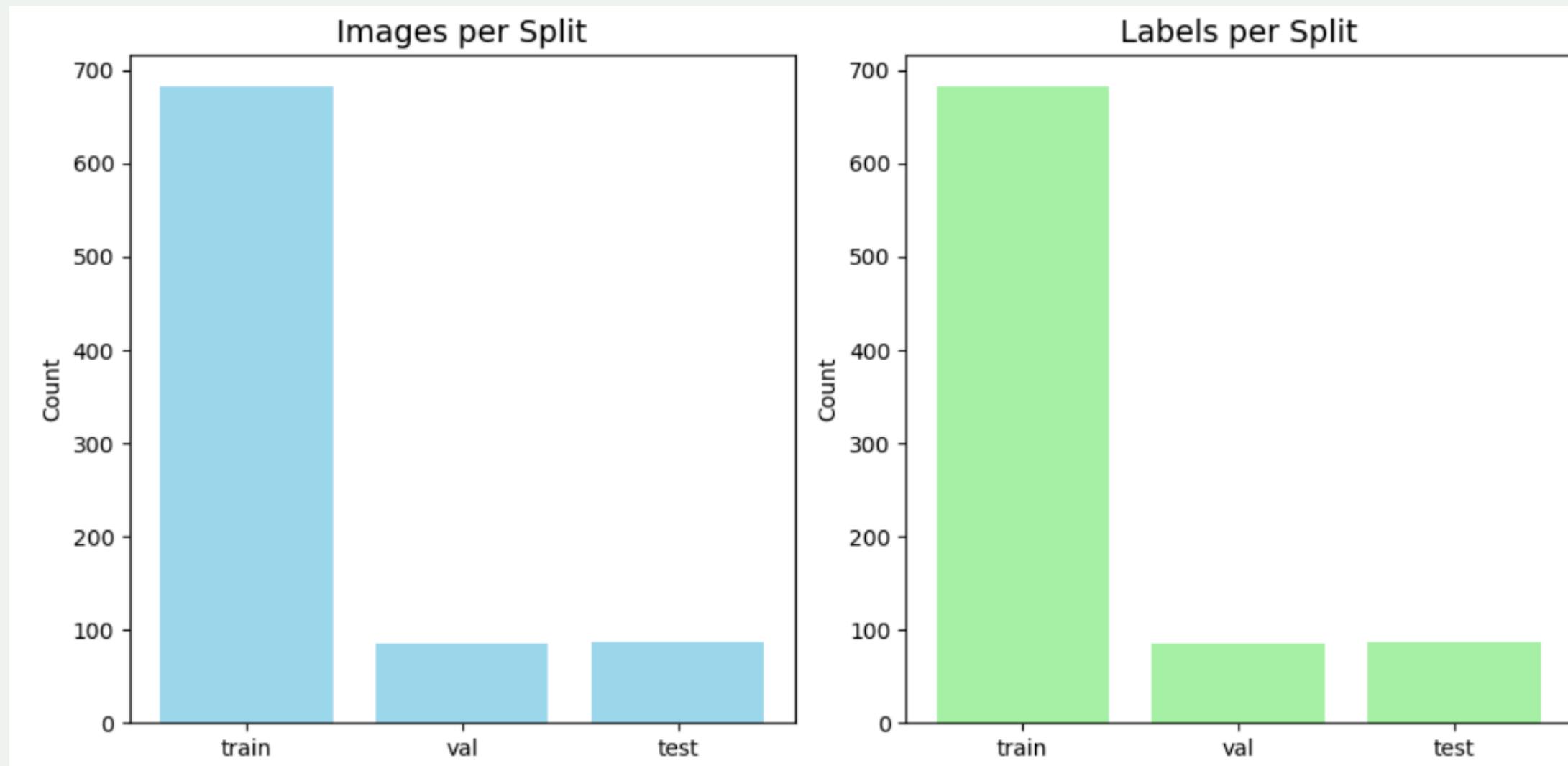
PRESENTED BY EIAMN ALRASHDI

Problem Statement

- Manual checking of face masks is slow and sometimes unreliable.
- We need an automatic system to detect if people wear masks correctly from images.
- Goal: classify each face as with mask, mask worn incorrectly, or without mask.

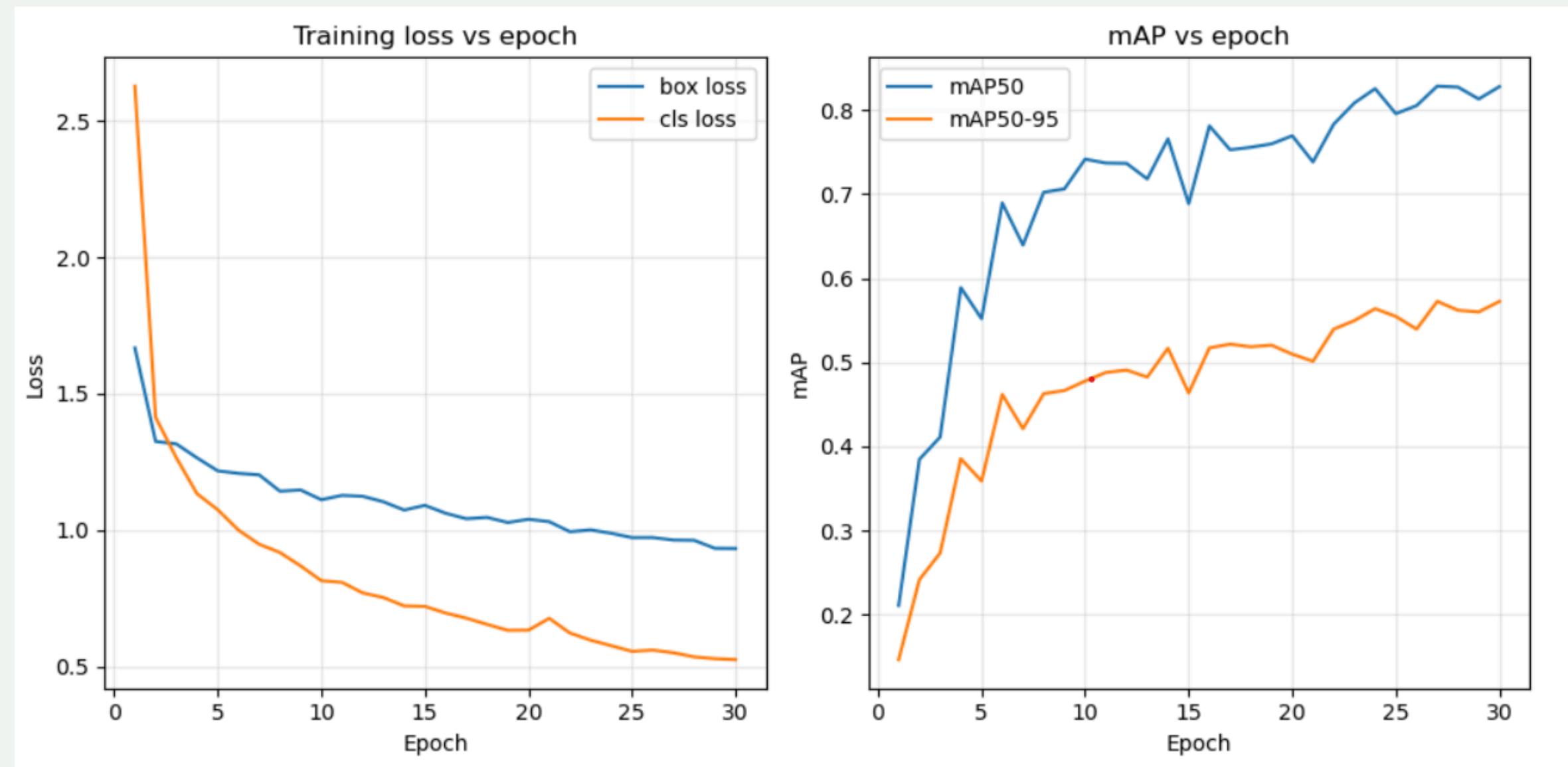
DATASET STATISTICS

- Total images: 853 (annotations converted from XML to YOLO format).
- Split: Train 682 (80%), Validation 85 (10%), Test 86 (10%).
- Classes: With Mask (correct), Mask Incorrect, Without Mask.
- The bar charts show how many images and labels are in each split, confirming that every image has a label.



Model and Training

- Detection model: YOLOv8 from Ultralytics.
- Trained with standard YOLO settings and an optimizer such as SGD..
- Training monitored using loss and mAP curves.



Both loss curves decrease and become stable, which shows the model learned good weights.

mAP keeps increasing and reaches strong values (about 0.8 at IoU 0.5), so detection accuracy improves with more epochs

final Model Results

- Final model shows high detection performance on test set.
- Strong on with mask and without mask classes.
- Main weakness: confusion between with mask and mask worn incorrectly.

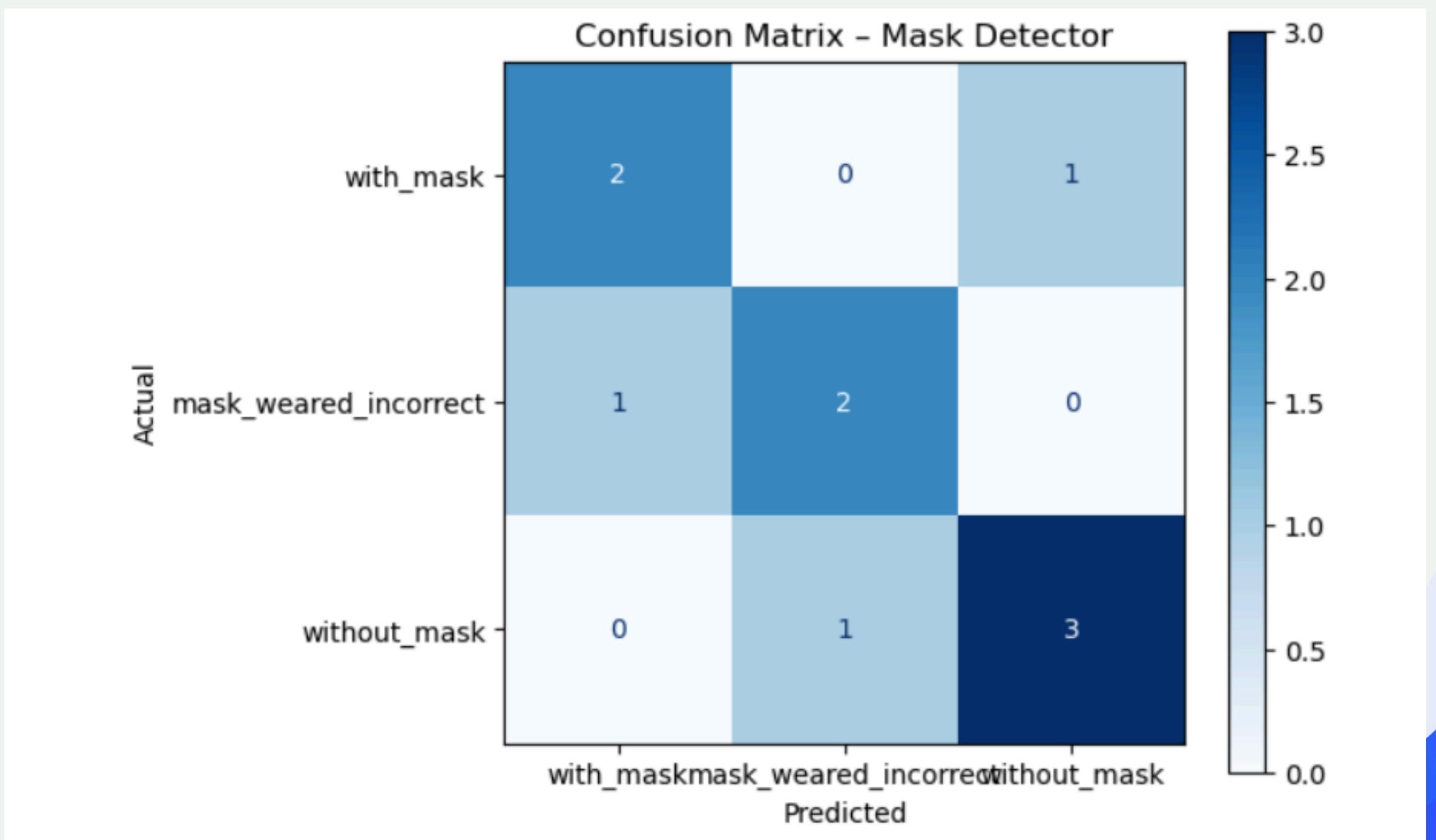
YOUR FINAL RESULTS:

mAP@0.5:	0.749
mAP@0.5:0.95:	0.533
Precision:	0.819
Recall:	0.713
F1-Score:	0.913

Confusion Matrix Results

- Correct predictions: 10 (77%)
- Misclassifications: 3 (23%)

Actual Class	Predicted Correctly	Misclassified As
with_mask	2	1 as without_mask
mask_weared_incorrect	2	1 as with_mask
without_mask	3	1 as mask_weared_incorrect



bounding box sample

```
#3. Sample Annotations Visualization
# Actual bounding boxes on sample image
import cv2
import matplotlib.patches as patches

# Show 1 image with bounding boxes
img_path = 'dataset/images/train/makssssksksss0.png' # First image
img = cv2.imread(img_path)
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
label_path = img_path.replace('images', 'labels').replace('.png', '.txt')

fig, ax = plt.subplots(1, 1, figsize=(12,8))
ax.imshow(img)

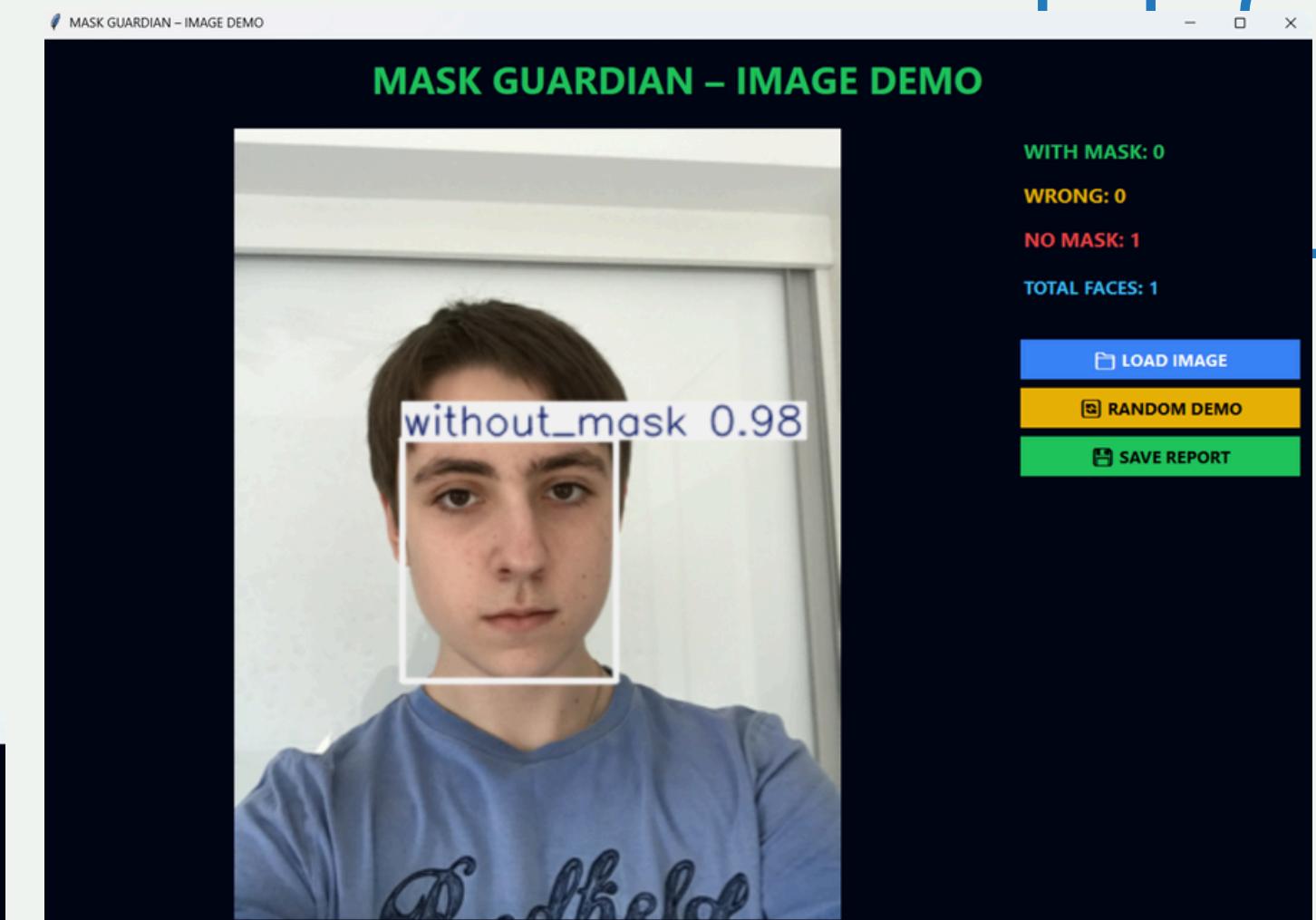
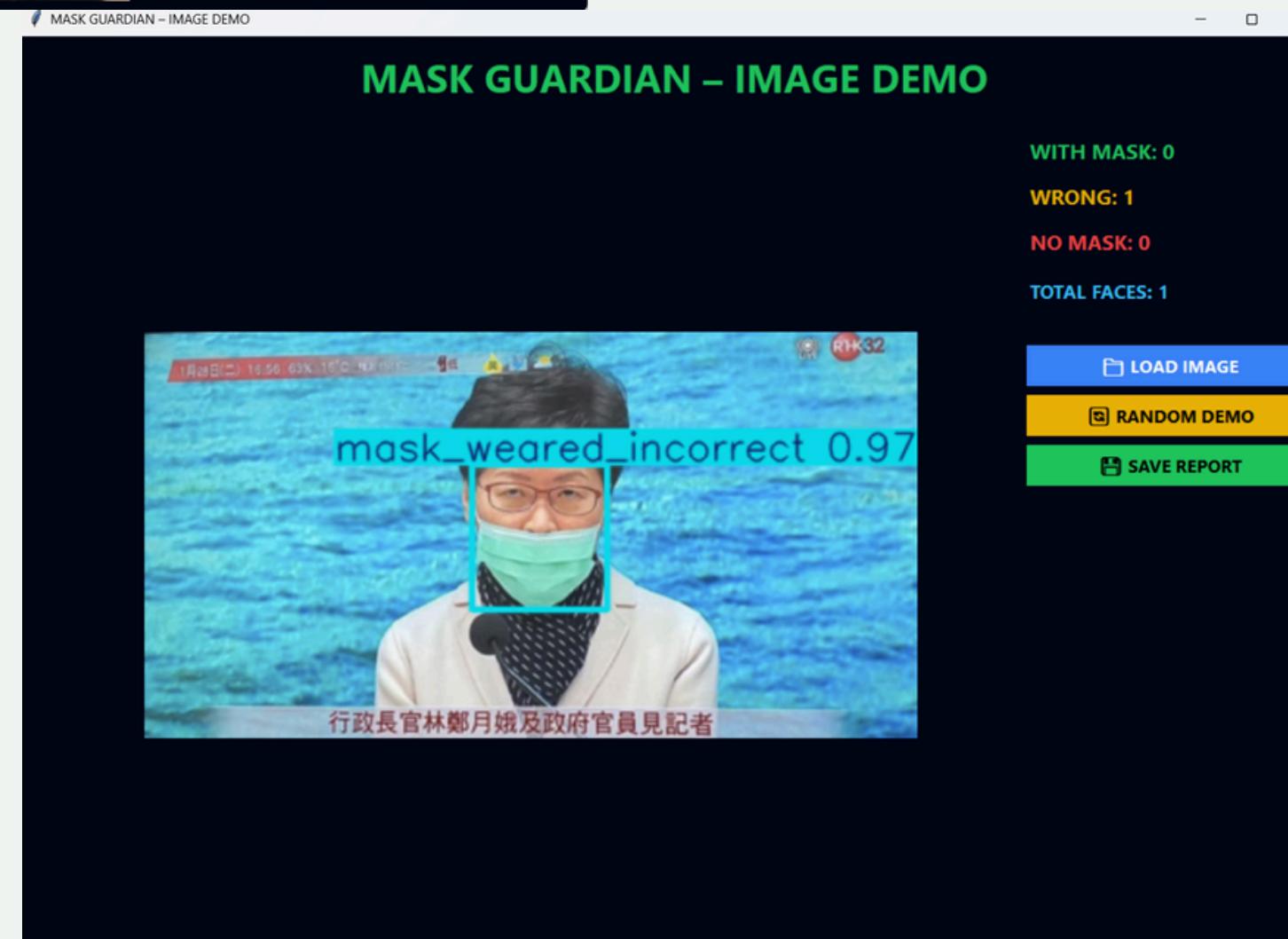
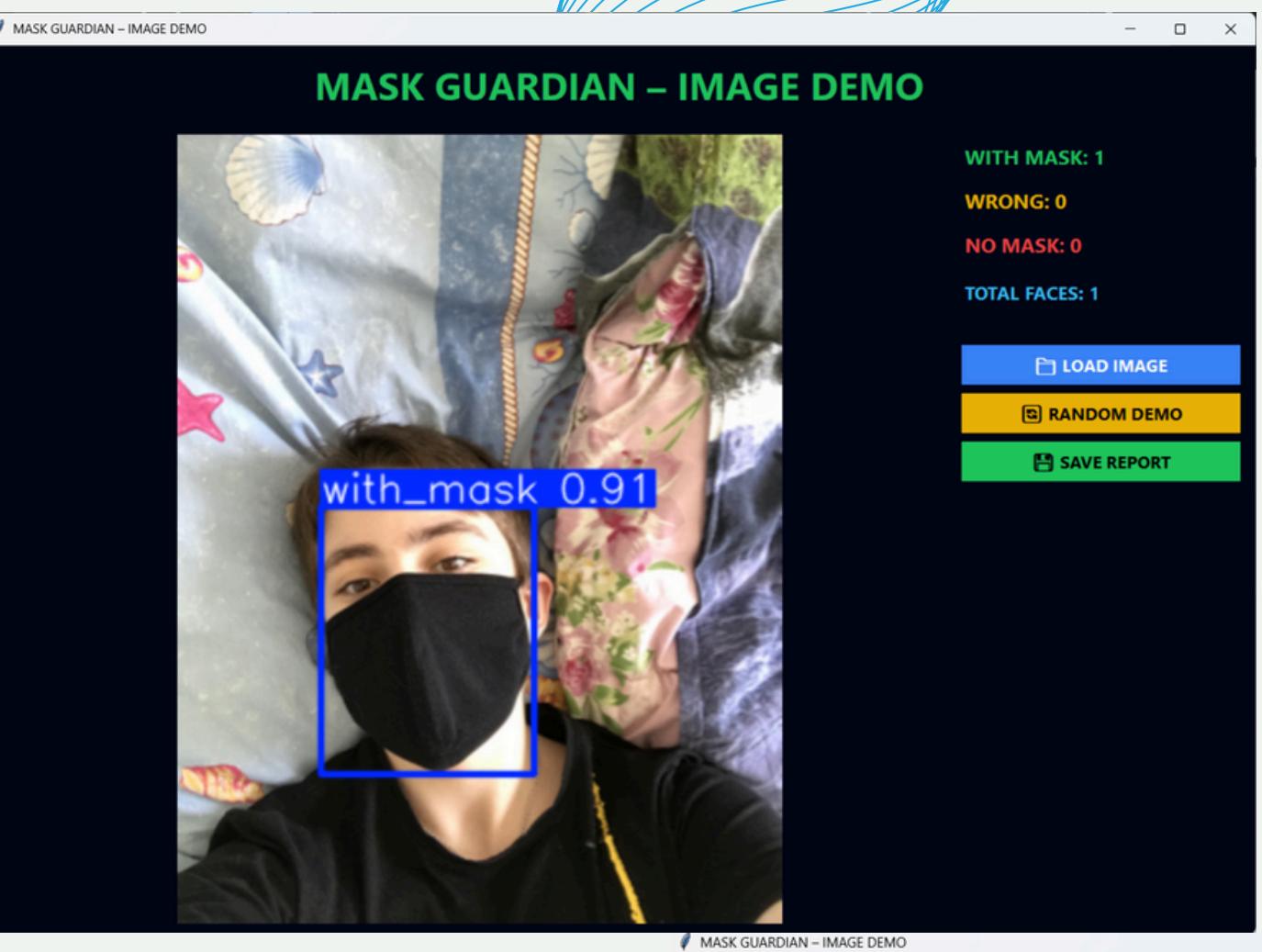
if os.path.exists(label_path):
    with open(label_path, 'r') as f:
        for line in f:
            cls, x, y, w, h = map(float, line.split())
            x1 = (x - w/2) * img.shape[1]
            y1 = (y - h/2) * img.shape[0]
            x2 = (x + w/2) * img.shape[1]
            y2 = (y + h/2) * img.shape[0]

            rect = patches.Rectangle((x1,y1), x2-x1, y2-y1, linewidth=2,
                                    edgecolor='red', facecolor='none')
            ax.add_patch(rect)

ax.set_title('Sample Image with YOLO Annotations')
plt.savefig('sample_annotations.png', dpi=300, bbox_inches='tight')
plt.show()
```



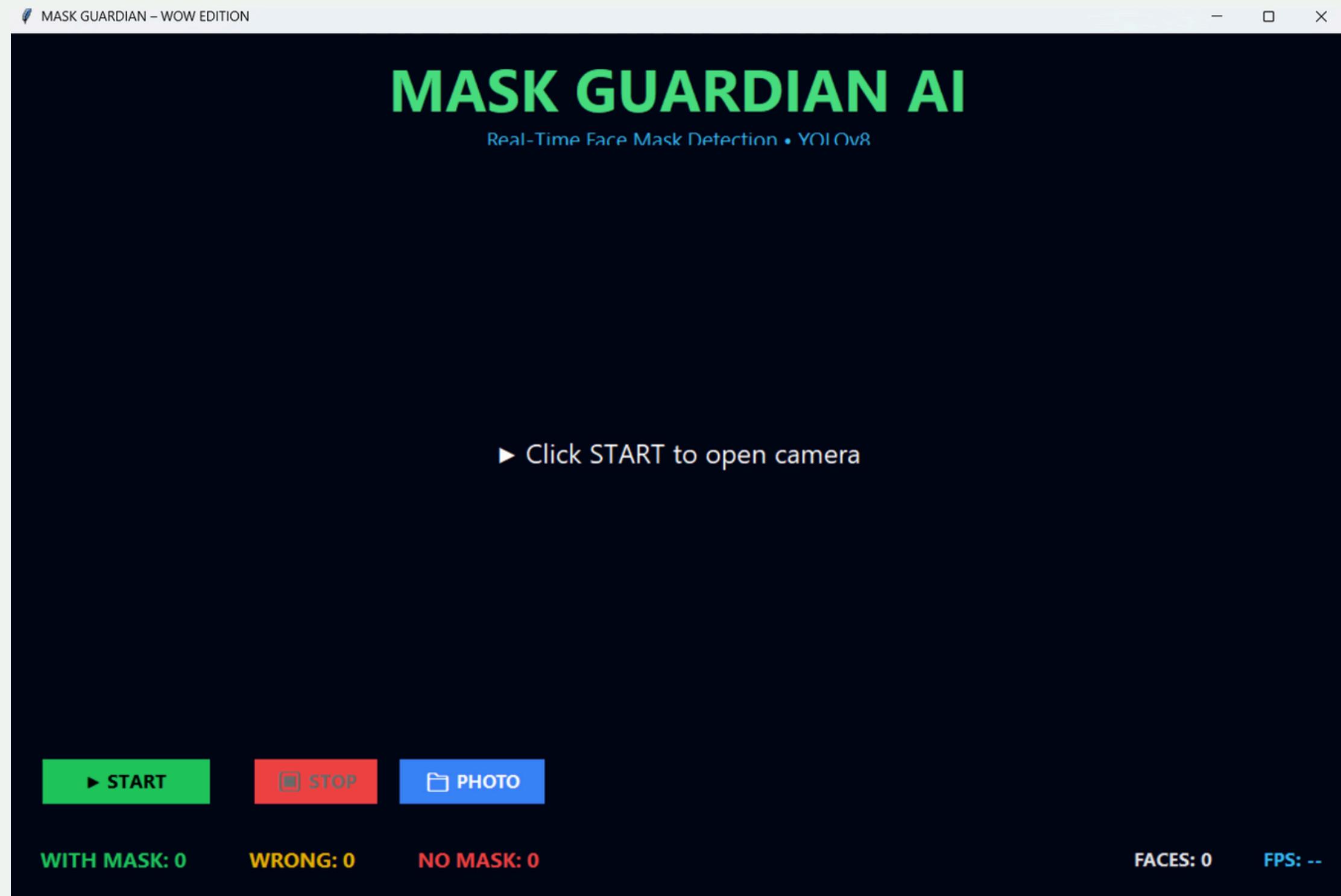
GUI Application



GUI Application



real time face detection



Questions

- Q1: Why do we need an automatic face-mask detection system instead of only human monitoring?
- Q2: Where could this system be integrated in the future?



Thank You!!