CV - Project Report

Owen Attard (xattaro00)

December 2024

https://github.com/OwenAttard22/CV-Project

1 Introduction

Face masks are essential for protecting the individual and their peers as they reduce the transmission of infected respiratory particles. A study found that risk is reduced by 70% for those who always wear a mask outdoors [1].

However, monitoring the compliance of face masks is a laborious and challenging task, especially in crowded and large areas. To address this, automated systems powered by computer vision have emerged as prominent solutions [2, 3]. In this paper, we aim to implement and evaluate two popular object detection frameworks for detecting and classifying proper face mask usage.

The following objectives have been defined to achieve this aim:

1. Define a dataset and pre-process it as necessary.
2. Implement a Yolo object detection model on this dataset and evaluate it using mean average precision.
3. Implement a Faster R-CNN model on this dataset and evaluate it using mean average precision.

2 Methodology

2.1 Objective 1: Dataset and Pre-Processing

An initial dataset was built using a publicly available dataset on Kaggle about face mask detection that includes 853 images, each labelled with one or more persons inside as either wearing a face mask improperly or not wearing one at all, as shown in Figure 1 [4].

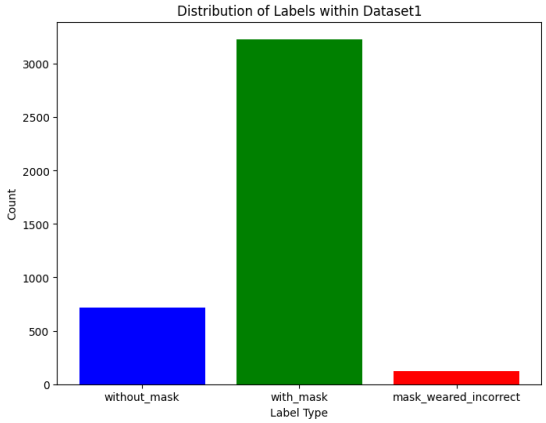

Figure 1: Initial Dataset Distribution

Using another Kaggle dataset that focuses on face detection, the original dataset was extended by adding 998 images of faces labelled as not wearing a mask to balance the main two labels of the dataset [5]. The extended and final dataset distribution can be seen in Figure 2.

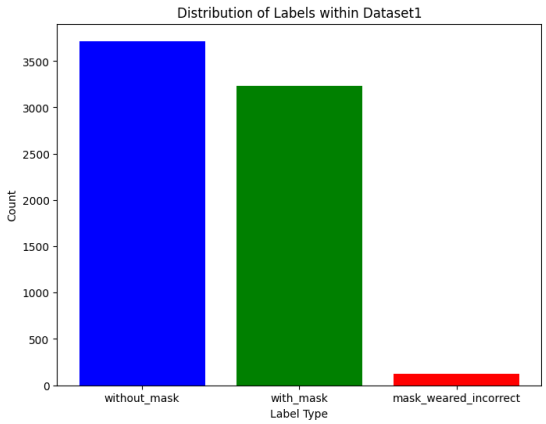

Figure 2: Final Dataset Distribution

The face mask dataset was labelled using Yolo format, and the face detection dataset was labelled in Pascal VOC format. Initially, both formats were considered a requrement for the proposed models, pre-processing involved converting one format into the other and vice-versa. Later, the introduction of the detectron2 library required the conversion of the Pascal VOC labels to COCO format. The dataset was prepared for k-fold cross-validation, where $k = 5$. This process splits the data into k equal parts, using $k - 1$ parts for training and the

remaining part for validation in each iteration.

2.2 Objective 2: YOLO Model

YOLOv11 was used to implement the Yolo model as it is the latest release by Ultralytics in September 2024. Specifically, the nano variant of YOLOv11 was implemented due to its efficiency in training. The five folds were trained for 100 epochs each, with a batch size of 15 and an image size of 640px.

Fold	mAP50	mAP50-95
1	0.801	0.537
2	0.870	0.606
3	0.875	0.604
4	**0.878**	**0.624**
5	0.881	0.612

Table 1: YOLO Model: mAP50 and mAP50-95 Results Across 5 Folds

The fourth fold achieved the highest overall performance, with a mAP50-95 of 0.624. To test its practical capabilities, the model was used to infer predictions on a video of a crowded environment. Qualitatively the infered video demonstrated promising accuracy in detecting proper and improper mask usage, showcasing the potential for real-world application.

Figure 3: Inference Video Image

2.3 Objective 3: Faster R-CNN Model

Meta's Detectron2 library was used to implement a Faster R-CNN model, specifically the Faster R-CNN R-50-FPN 3x variant [6]. Training was performed using

a learning rate of 0.0025 for 10,000 iterations, which was determined based on the training time per fold of approximately 1.5 hours.

Fold	**mAP50-95**
1	0.537
2	0.454
3	0.501
4	0.468
5	0.483

Table 2: Faster R-CNN: mAP50-95 Results Across 5 Folds

While the Faster R-CNN model achieved reasonable results, it was outperformed by the YOLO model in terms of mAP50-95.

2.4 Hardware Limitations and Resolution

A key limitation faced during the project was the hardware capability of local systems, which lacked sufficient resources to train the models efficiently leading to 14 hour training times per Yolo fold. This limitation was resolved by utilizing Google's Colab platform, which provided access to GPU resources. All committed model folds were trained using Colab, ensuring consistent training conditions and reduced training time.

3 Conclusion

This study looked at the implementation of two popular object detection frameworks for face mask usage detection where we compared YOLOv11's nano variant against detectron2's Faster R-CNN 50 variant. The results showed that for training of similar time periods the Yolo model provided the best mean average precision within the IoU range of 50 to 95. A qualitative evaluation of Yolo's best fold's inference on a video of a crowd displayed promising results.

The inference video can be found within the github repository or more specifically click here..

References

[1] J. Howard, A. Huang, Z. Li, Z. Tufekci, V. Zdimal, H.-M. Van Der Westhuizen, A. Von Delft, A. Price, L. Fridman, L.-H. Tang, *et al.*, "An evidence review of face masks against covid-19," *Proceedings of the National Academy of Sciences*, vol. 118, no. 4, p. e2014564118, 2021.

[2] J. Ieamsaard, S. N. Charoensook, and S. Yammen, "Deep learning-based face mask detection using yolov5," in *2021 9th International Electrical Engineering Congress (iEECON)*, pp. 428–431, 2021.

[3] R. Liu and Z. Ren, "Application of yolo on mask detection task," in *2021 IEEE 13th International Conference on Computer Research and Development (ICCRD)*, pp. 130–136, 2021.

[4] A. MVD, "Face mask detection dataset," 2021. Accessed: December 2024.

[5] F. E. Menshawi, "Face detection dataset," 2021. Accessed: December 2024.

[6] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2." `https://github.com/facebookresearch/detectron2`, 2019.