Face Mask Detection using YoloV3

Project report

Sana Moin & Navneet Singh Arora

06.08.2021

1 Introduction

The recent outbreak of Coronavirus disease (COVID-19) severely affected the entire world infecting millions of people across multiple phases. As it can be spread by coughing or sneezing, masks have become a necessity and hence mask detection is very important [1].
Using Object Detection technique of Computer Vision, we try to detect if a person is wearing a mask correctly, incorrectly, or not at all. We use YoloV3, a single-shot detection algorithm, wherein the model will have the entire context of the input image because it only looks at an image once [2]. This fascinating computer vision and image processing technique detects and defines objects such as persons, vehicles, and animals from digital images and videos.

2 System Description

The system is based on Real-Time Face Mask Detection Method Based on YOLOv3 using Properly Wearing Masked Face Detection Dataset (PWMFD). In this paper, SE (Squeeze and Excitation Module) has been added by the authors to further improve YoloV3 Performance as compared to YOLOv2. It is an Attention Mechanism Module [1]. It is an open-source neural network framework written in C and CUDA.

3 Modifications

Following changes are made before the execution:

- New utility py file to convert the annotation files (.xml) in the required Yolo format (.txt with normalization).
- Modified Yolo configuration file classes and filter parameter. Filter size is calculated using: number of anchors * (5+ number of classes). In our case, 3*(5+3) = 24.
- Number of batches, epochs, and subdivisions are modified due to GPU memory constraints.
- For executing the model, Pretrained convolutional layer embeddings are downloaded and the darknet repository is cloned, along with few modifications in Makefile to make use of GPU, CUDA, and OPENCV.
- A specific folder structure for data is used(images and labels folder under data with the list of files under train.txt and val.txt for training and validation respectively) as per Yolo's execution constraints.
- Finally, a script is created to collect the final outputs on the detection set and create the required visualizations as well for the error analysis.

4 Evaluation and Experimentation Details

4.1 Metrics

For evaluation purposes, multiple metrics are used: Three categories of loss functions are used(Bounding Box Loss, Object Loss, and Classification Loss), Precision, Recall, F1 Score and Mean Average Precision (mAP).

4.2 Loss Function

The loss function used for this experimentation is the Mean Squared Error Loss.

4.3 Epochs

The number of epochs for this experiment were 10. This is computed using number of classes, number of images in the dataset and the number of Yolo filters.

4.4 Other Parameters

Parameter Name	Value
Batch Size	64
Filters (before Yolo Layer)	24
Step Size	4800

Parameter Name	Value
Learning Rate	0.001
Decay	0.0005
Momentum	0.9

5 Results

Parameter	Value	Parameter	Value
Precision	91.62%	mAP_0.5:0.95	57.28%
Recall	85.08%	Val Bounding box loss	0.0210
F1-Score	88.23%	Val Object loss	0.0058
mAP_0.5	90.44%	Val Classification loss	0.0016

The visualizations for all these can be seen in the Appendix section.

6 Conclusion

The YoloV3 based model is implemented using the PWMFD dataset for detecting Mask/No-Mask/Incorrect-Mask objects in the images. The model performs well with 88.23% F1 score. Even though the model works with speed and accuracy and performs well but there are still limitations as multiple actions, like coughing or sneezing with no mask, can still spread the Covid disease. Hence, future work involves extending the model to action detection that can help avoid high rise in cases in the future.

References

[1] Xinbei Jiang et al. "Real-Time Face Mask Detection Method Based on YOLOv3". In: *Electronics* 10.7 (2021), p. 837.

[2] Joseph Redmon and Ali Farhadi. "Yolov3: An incremental improvement". In: *arXiv preprint arXiv:1804.0276* (2018).

Appendices

Imbalanced Data Statistics

As can be seen from the figure, the data across classes is imbalanced having 'Without Mask' objects with maximum instances, followed by 'With Mask' object and at last 'Incorrect Mask' object which are negligible compared to the other two classes.

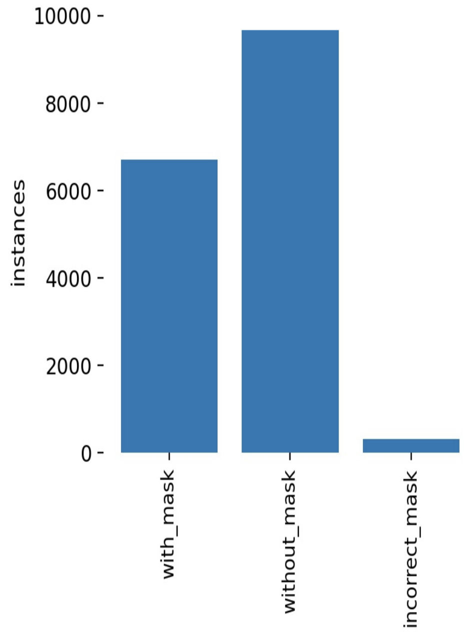

Figure 1: Imbalanced Data Statistics

Result across all classes for Imbalanced Data

Figure 1 shows the True Positives, True Negatives, False Positives and False Negatives per class and across classes. This is relevant as the data is imbalanced across classes.

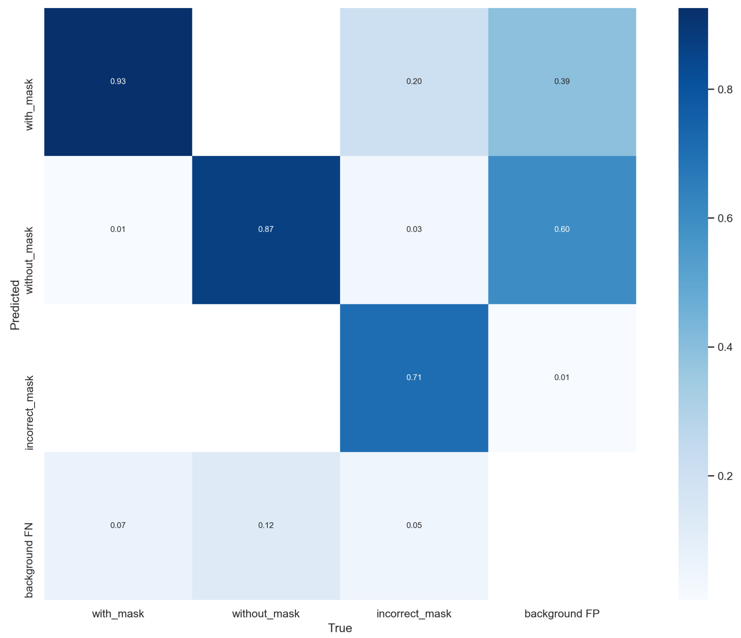

Figure 2: Confusion Matrix across all classes

Mean Average Precision (mAP)

Figure 2 shows the mAP using threshold 0.5 and 0.95.

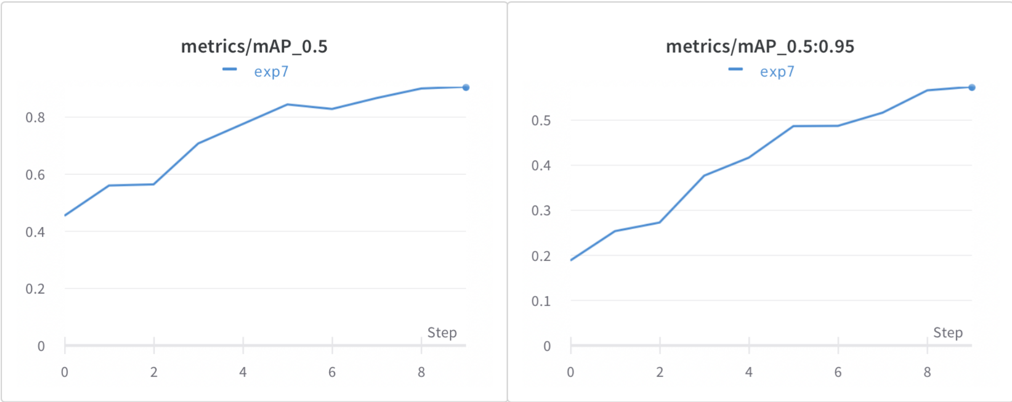

Figure 3: Mean Average Precision with different Thresholds

Precision and Recall

This is computed for Multi-class Classification considering Imbalanced data.

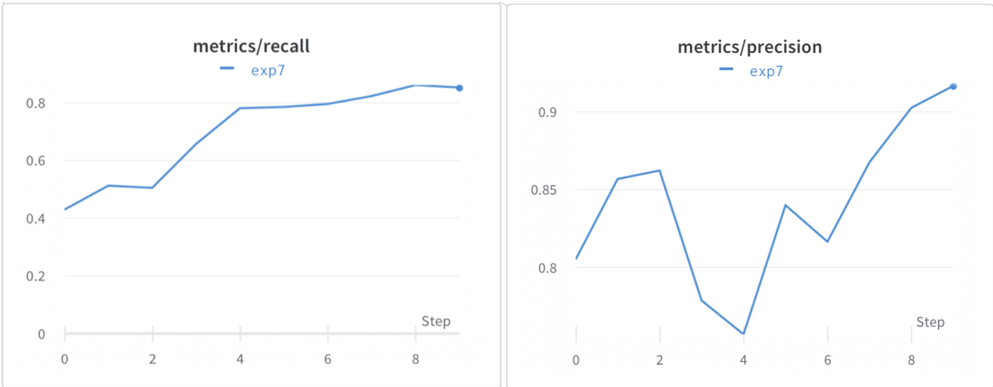

Figure 4: Recall and Precision Across all Classes

Loss Functions

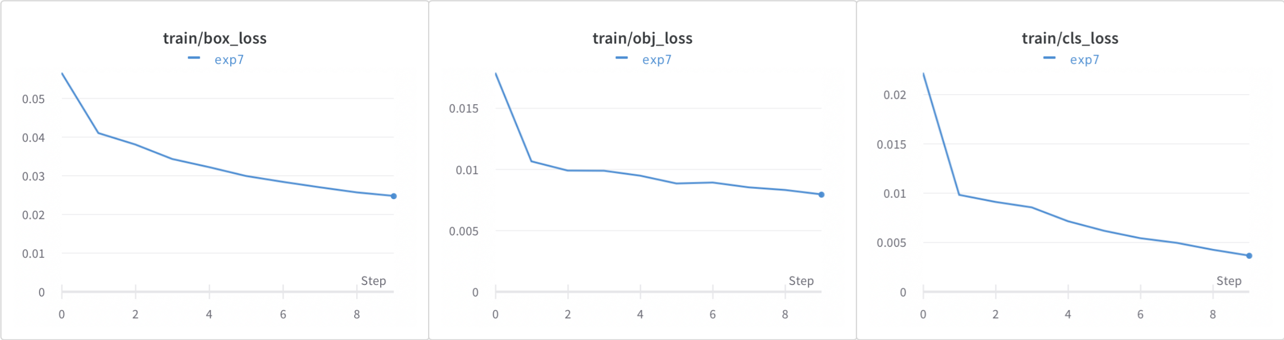

Figure 5: Training Loss

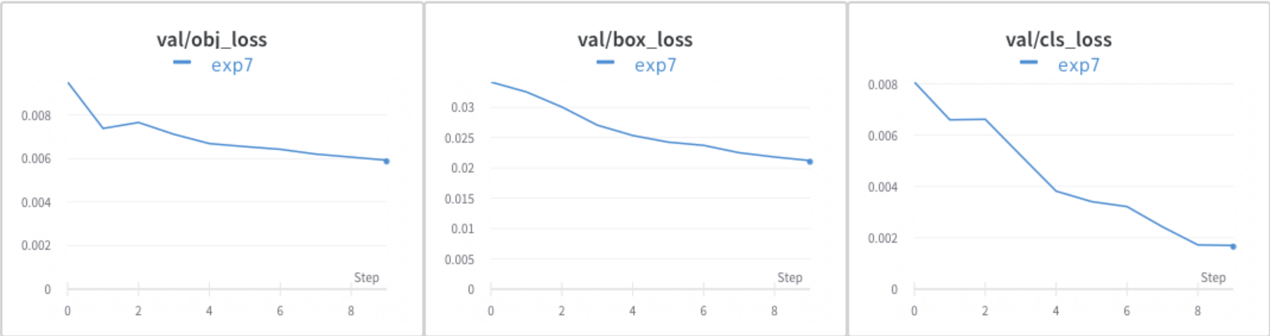

Figure 6: Validation Loss

Prediction Images

Finally, this section shows the output (prediction) images with class labels. Where the model is not 100% confident, in that case multiple labels along with the detection percentage is displayed otherwise only a single class is shown.

Figure 7: Image 1 with False Negative and False Positive Predictions

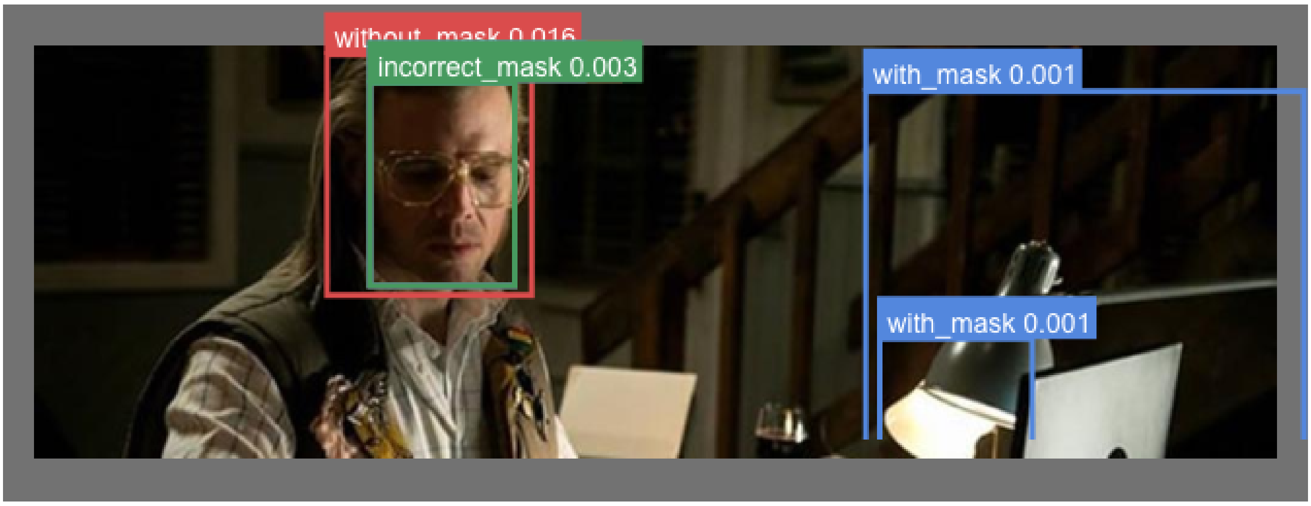

Figure 8: Image 2 with False Negative and False Positive Predictions

Figure 9: Image 3 with Accurate Predictions but Low Recall

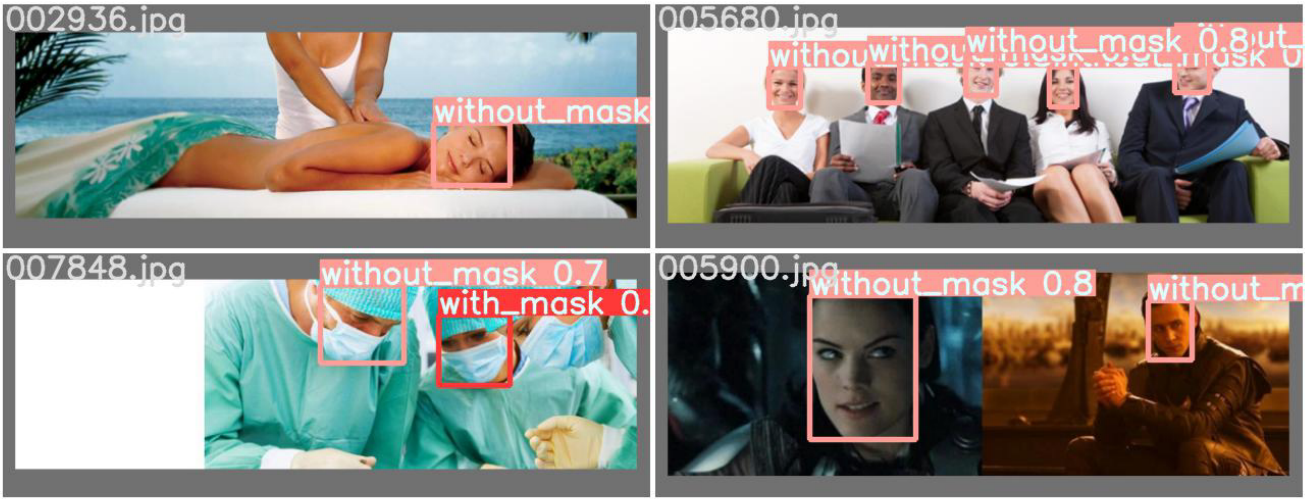

Figure 10: Image 4 with Accurate Prediction

Figure 11: Image 5 with False Positives but High Recall