# Real-time Face mask detection using YOLO algorithm

**Wei-Lun-Hsu,** University of Maryland, College Park, hsu118@umd.edu

**Kevin Chou,** University of Maryland, College Park, kchou1@umd.edu

**Pranav Mare**, University of Maryland, College Park, pmare@umd.edu

## Abstract

Effective strategies to restrain COVID-19 pandemic need high attention to mitigate negatively impacted communal health and the global economy, with the brim-full horizon yet to unfold. In the absence of effective antiviral and limited medical resources, many measures are recommended by WHO to control the infection rate and avoid exhausting the limited medical resources. Wearing a mask is among the non-pharmaceutical intervention measures that can be used to cut the primary source of SARS-CoV2 droplets expelled by an infected individual. Regardless of discourse on medical resources and diversities in masks, all countries are mandating coverings over the nose and mouth in public. To contribute towards communal health, in the project, we will implement real-time object detection toward mask-wearing. We aim to use the project to save human resources in checking if people are wearing masks properly and develop a counter to keep track of the number of people not wearing masks.

## 2. Introduction

The 209th report of the world health organization (WHO) published on 16th August 2020 reported that coronavirus disease (COVID-19) caused by acute respiratory syndrome (SARS-CoV2) has globally infected more than 6 Million people and caused over 379,941 deaths worldwide [1]. According to Carissa F. Etienne, Director, Pan American Health Organization (PAHO), the key to control COVID-19 pandemic is to maintain social distance, improving surveillance and strengthening health systems. Recently, a study on understanding measures to tackle COVID-19 pandemic carried by the researchers at the University of Edinburgh reveals that wearing a face mask or other covering over the nose and mouth cuts the risk of Coronavirus spread by avoiding forward distance travelled by a person's exhaled breath by more than 90% [2]. Steffen et al. also carried out an exhaustive study to compute the community-wide impact of mask use in the general public, a portion of which may be asymptomatically infectious in New York and Washington. The findings reveal that near universal adoption (80%) of even weak masks (20% effective) could prevent 17–45% of projected deaths over two months in New Work and reduce the peak daily death rate by 34–58% [3]. Their results strongly recommend the use of face masks in the general public to curtail the spread of Coronavirus. Further, with the reopening of countries from COVID-19 lockdown, Government and Public health agencies are recommending face masks as essential measures to keep us safe when venturing into public. To mandate the use of facemasks, it becomes essential to devise some techniques that force individuals to apply a mask before exposure to public places. This project will be detecting the people without/with masks in real-time using convolution neural networks with a newer version of the YOLO algorithm.

## 2. Related Work

A modern detector is usually composed of two parts, a backbone which is pre-trained on ImageNet and a head which is used to predict classes and bounding boxes of objects. For those detectors running on GPU platform, their backbone could be VGG [4], ResNet [5], ResNet[6], or DenseNet [7]. For those detectors running on CPU platform, their backbone could be SqueezeNet [8], MobileNet, or ShuffleNet . As to the head part, it is usually categorized into two kinds, i.e., one-stage object detector and two-stage object detector. The most representative two-stage object detector is the R-CNN [9] series, including fast R-CNN [10], faster R-CNN [11]. It is also possible to make a twostage object detector an anchor-free object detector, such as RepPoints. As for one-stage object detector, the most representative models are YOLO [11,12,13], SSD [14], and RetinaNet [15].

Faster RCNN is composed of a Regions Proposal Network(RPN) and the image classifier. Detecting. To perform detention, Faster RCNN utilized the RPN to come up with a list of region proposals then passed it to the second stage to detect and refine the bounding box. Whereas Yolo divided the image to grids and then passed to the detection network, which was only presented in a single stage.

The Single Shot Detector (SDD), it first runs a convolution network to compute a feature map then pass them to the next layer to depict the bounding boxes. Different layers are designed to deal with objects with different scales. However, in Yolo, the network will predict on different grid region proposals.

Result shows Faster RCNN has the least performance of all, and SDD works fastest. And when it comes to accuracy, in the MS COCO dataset, VOC2007 and VOC 2012, the Faster RCNN has the best accuracy then the other. To get better accuracy and performance, we decided to use YOLO for our network.

| Method | mAP | FPS |
|---|---|---|
| Faster R-CNN (VGG16) | 73.2 | 7 |
| Fast YOLO | 52.7 | 155 |
| YOLO (VGG16) | 66.4 | 21 |
| SSD300 | 74.3 | 46 |
| SSD512 | 76.8 | 19 |
| SSD300 | 74.3 | 59 |
| SSD512 | 76.8 | 22 |

Figure1: Comparing other algorithms

## 3. Data

Initially, we used the Face Mask Detection data from Kaggle. It's a small data set containing 800 hundred labeled well pictures. However, we observe that someone tries to use their cloth or hands to cover their face in the real world. It is one of the disadvantages of this data set. We can't detect the hand or cloth covering the face correctly using this data set. Also, this data set lacks the profile and obscure face, which causes the weight training from this dataset to have little resistance in real-world detection. Therefore, we decided to customize our data set from WIDER Face dataset, MAFA dataset, and RMFD since there are different advantages to these three datasets. We use without mask pictures from WIDER Face dataset, extract the hand or cloth covering image from MAFA dataset, and download masking people pictures from RMFD dataset. Combining features from these three dataset could help us dispose of every situation including hand covering, cloth covering, profile, obscure input(e.g. closed-circuit television or moving people). After ensuring the data source, we used the labeling to add or modify labels in some pictures manually. The label format in YOLO is:

**Class_id, x, y, w, h**

x: The x coordinate of the center point of the target (horizontal) / the total width of the picture
y: The y coordinate (vertical) of the center of the target / the total height of the picture
w: the width of the target frame / total width of the picture
h: height of target frame / total height of picture

For example the size of this picture is 600 * 600
x = x_center / width = 250 / 600 = 0.416
y = y_center / height = 285 / 600 = 0.475
w = (xmax - xmin) / width
= (390 - 116) / 600 = 0.456
h = (ymax - ymin) / height
= (410 - 136) / 600 = 0.456
If label 1 = M
Label document is:
1 0.416 0.475 0.456 0.456

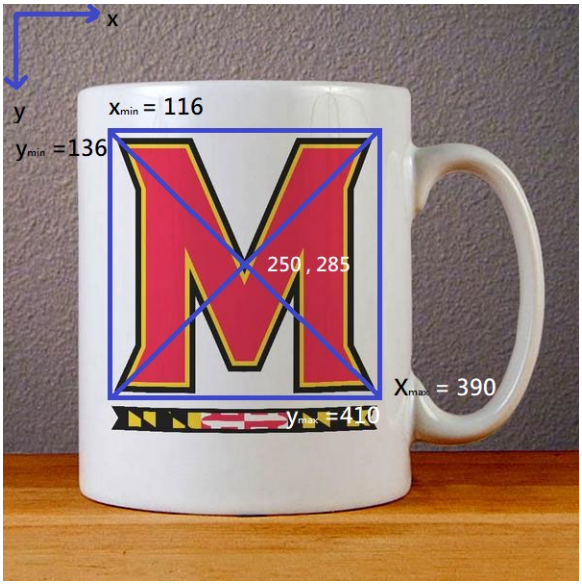


Figure 2: Data labelling

Use the BECLogits loss function to calculate the loss of the objectness score, the class probability score uses the cross-entropy loss function (BCEclsloss), and the bounding box uses GIOU Loss.

## 4. Methods

We firstly use the tool RoboFlow to transfer labels from XML to Yolo format, then resize them to a unique size(416*416). We then split data to 70% as a training set and 20% as validation set and 10% as testing set. In order to increase the accuracy of the network, we imply image flipping, cropping, cutout and rotation. Moreover, since our network serves the purpose of detecting bounding boxes, we also apply Image flipping, Rotation, and Cropping in bounding boxes. To reach the goal of real-time mask detection, we implemented the newest YOLOv5 and did transfer learning on the default Yolo network. We also use both Wandb and TensorBoard to keep track of the training of our network.After training, we examine F1-curve ,PR-curve, confusion matrix and the loss history of both bounding box and object to improve our network.
Lastly, we implement functions in the detection code in Yolov5[2] to implement the counter of each class.

## 5. Experiments

To perform transfer learning, we used weights and layers from Yolo with 270 layers and SGD gradient descent.

First, we tried to plot out the distribution of each class(Figure 3), the plot shows the ratio of data without a mask and with a mask is 2:1, which will not cause data imbalance(Tabel1).

Then we looked at the loss function of the training in 30(Figure4), 50(Figure5),and 100 epochs(Figure6) to determine how many epochs we use as our condition to prevent overfitting. In the final result, we decided to train the model for 80. Based on the (Figure7), we decide to stop training to prevent overfitting. We also used the PR curve(Figure8), and F1 curve(Figure9) to evaluate our model. By comparing the value of PR curve and F1 value, we could observe that the performance of 80 epochs is better than others no matter in the accuracy or threshold.

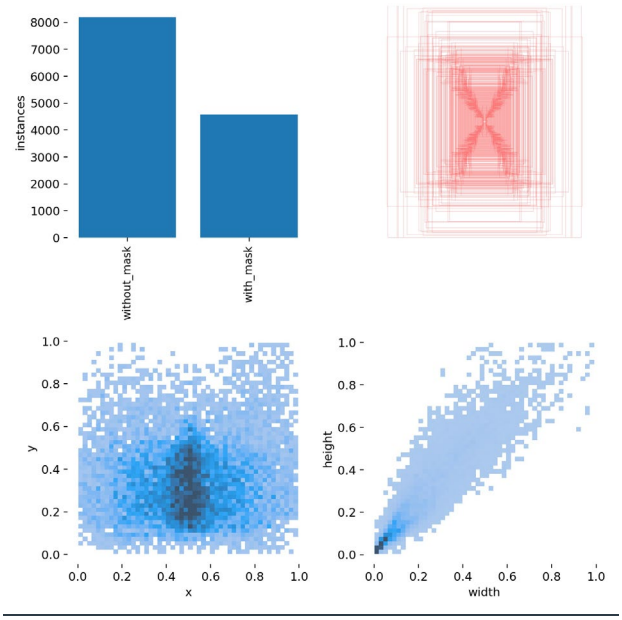


Figure 3. Data distribution

| Degree of imbalance | The proportion of Minority Class |
|---|---|
| Mild | 20-40% of the data set |
| Moderate | 1-20% of the data set |
| Extreme | <1% of the data set |

Table 1: Imbalance Data Set


Figure 4. Loss of 30 epoch


Figure 5. Loss of 50 epoch


Figure 6. Loss of 100 epoch


Figure 7. Loss of 100 epoch

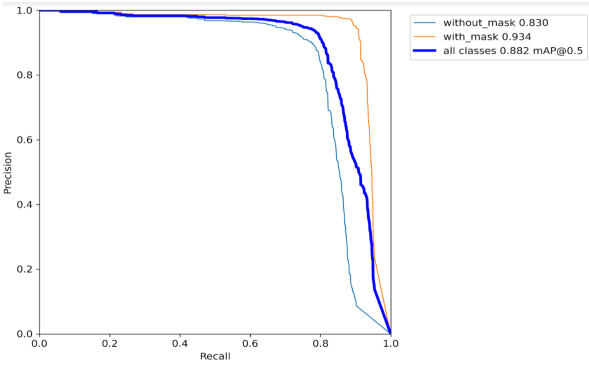


Figure 8. PR_curve of 80 epoch

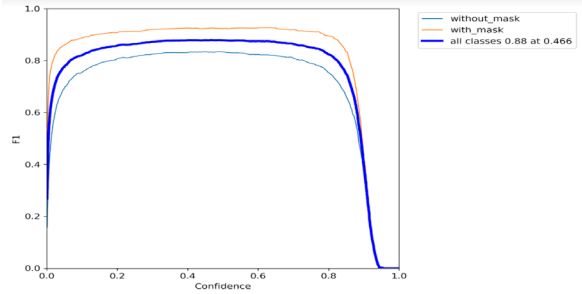


Figure 9. F1_curve of 80 epoch

| condition | PR_curve | F1_curve |
|---|---|---|
| 100 | 0.88 mAP @0.5 | 0.88 at 0.53 |
| 80 | 0.882 mAP@0.5 | 0.88 at 0.466 |
| 50 | 0.88 mAP @0.5 | 0.87 at 0.47 |
| 30 | 0.877 mAp@0.5 | 0.87 at 0.524 |

Tabel 2. PR and L1 curve value

## 6. Limitations

To increase the performance of real-time detection, the YOLOv5 only looks once at the image, consequently, the accuracy will be less than Faster-RCNN and other networks.

In addition, due to the imbalance of the data, people that wear masks improperly have less accuracy. Also if the person is wearing irregular masks, the accuracy will drop. And if the people aren't facing directly to the camera, it will be harder to detect.

## 7. Results

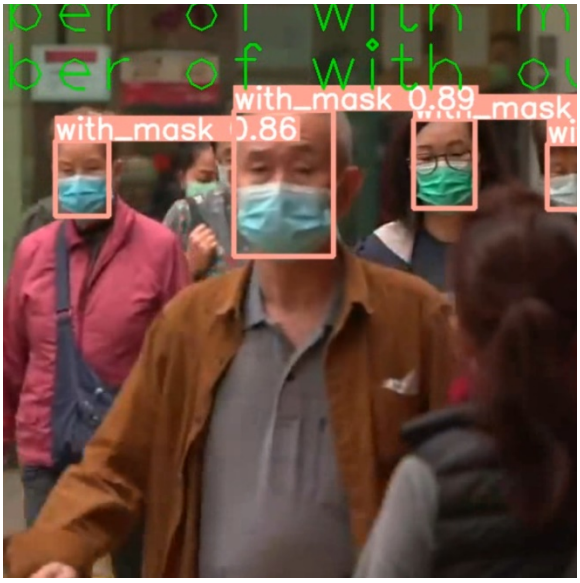


Figure 10. People with Mask detected

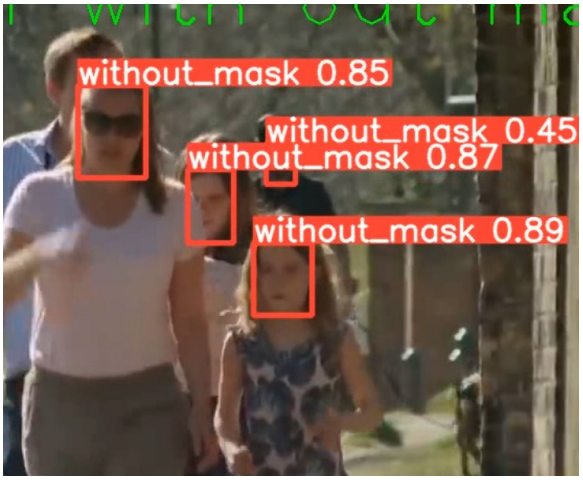


Figure 11. People without mask detected

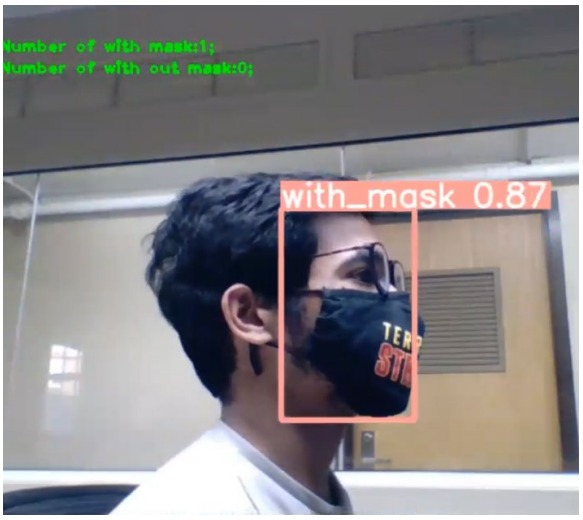


Figure: 12 Correctly detecting mask sideways. Moreover, displaying the count of mask/without mask people on the top left corner.

## 8. Conclusion

According to our customized dataset, our model could discriminate against people who use the clothes to cover his/her face or use their hands or arms to cover their face in real time detection compared to the public dataset.

We can discriminate against people without masks with 85 to 95% accuracy and less than 0.1 seconds. We believe the model with the counter can help monitor the campus by recording student's habits of wearing masks in real-time. During the training phase, we learned a lot from data loading, network training, and evaluating models.

In the future, we can also implement time counters to record the time of people wearing masks in order to prevent spreading disease.

## 9. References

[1] World Health Organization et al. Coronavirus disease 2019 (covid-19): situation report, 96. 2020. - Google Search. (n.d.). https://www.who.int/docs/default-sour ce/coronaviruse/situation-reports/20200816-covid-19-sitrep-209.pdf?sfvrsn=5 dde1ca2_2. .

[2] L.R. Garcia Godoy, et al., Facial protection for healthcare workers during pandemics: a scoping review, BMJ, Glob. Heal. 5 (5) (2020), e002553, https://doi.org/10.1136/bmjgh-2020-002553.

[3] S.E. Eikenberry, et al., To mask or not to mask: Modeling the potential for face mask use by the general public to curtail the COVID-19 pandemic, Infect. Dis. Model. 5 (2020) 293–308, https://doi.org/10.1016/j.idm.2020.04.001. [5] Wearing surgical masks in public could help slow COVID-19 pandemic's advance: Masks may limit the spread diseases including influenza, rhinoviruses and coronaviruses – ScienceDaily. (n.d.). https://www.sciencedaily.com/releases/2020/04/200403132345.htm

[4] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image

recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 770–778, 2016Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 770–778, 2016

[6]Saining Xie, Ross Girshick, Piotr Dollar, Zhuowen Tu, and ´Kaiming He. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1492–1500, 2017.

[7]Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 4700–4708, 2017

[8] Forrest N Iandola, Song Han, Matthew W Moskewicz,Khalid Ashraf, William J Dally, and Kurt Keutzer. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and¡ 0.5 MB model size. arXiv preprint

[9] Ross Girshick. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pages 1440–1448, 2015.

[10] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 580–587, 2014

[11] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In Advances in Neural Information Processing Systems (NIPS), pages 91–99, 2015.

[11] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 779–88, 2016.

[12] Joseph Redmon and Ali Farhadi. YOLO9000: better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 7263–7271, 2` 017.

[13] Joseph Redmon and Ali Farhadi. YOLOv3: An incremental improvement. arXiv preprint arXiv:1804.02767, 2018.

[14] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision (ECCV), pages 21–37, 2016.

[15] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In ´ Proceedings of the IEEE International Conference on Computer Vision (ICCV), pages 2980–2988, 2017.

[16] Roboflow
https://roboflow.com/
[17] Yolov5
https://github.com/ultralytics/yolov5
[18] Wandb
https://wandb.ai/site
[19]TensorBoard
https://www.tensorflow.org/tensorboard
[Tabel 1]
https://developers.google.com/machine-learning/data-prep/construct/sampling-splitting/imbalanced-data