Face Mask Etiquette During a Pandemic - Training a Neural Network to Perform Object Detection

Yuanxi (Forrest) Li
Northeastern University
Vancouver, BC, Canada
li.yuanxi1@northeatstern.edu

***Abstract*—During COVID-19 pandemic, wearing a face mask outside is not uncommon for us. While people are aware of the importance of wearing them, not everyone wears them correctly. Sometimes we can see people pulling masks to one side or letting them rest on the chin below the nose. Some public places even mandate the use of wear face masks or coverings with securities ensuring people are entering while wearing face masks or covering correctly. In this document, we are trying to automate this process with the help of object detection models. We show how we apply a deep learning model on an open dataset of people wearing face masks to detect if people in a photo are wearing face masks or wearing them correctly.**

Keywords—object detection, yolov5, open data, face masks

I. INTRODUCTION

Since 2020 spring, COVID-19 has always been a trending topic worldwide in our lives. We are so used to wearing respiratory protection methods like face masks everyday. While people are aware of the importance of wearing them, not everyone knows or wears them correctly. Take wearing face masks as an example, despite being designed to completely cover the nose and mouth, it's not uncommon to see them being pulled to one side or resting on the chin below the nose or with baggy sides. [1] There are some public places mandating the use of face masks or coverings, human's work is involved to ensure people are entering wearing face masks or coverings correctly. [2] Computer vision techniques like object detection could be helpful to automate such monitoring process. In this document, we are exploring the possibility to use deep learning models to perform such object detection tasks. We are using YOLOv5 by Ultralytics [3] with an open dataset of people wearing face masks [4] to find out if people in a photo are wearing face masks or wearing them correctly. We will explain the process by introducing the dataset with selected images shown. A workflow of training a deep learning models from partitioning dataset, model configuring, model training, to performance evaluations will also be introduced.

II. DATASET INVESTIGATION

A. Data Source

We acquire the dataset "Mask Dataset" (e.g. Fig. 1) from an open dataset website. [4] It is labeled following the PASCAL VOC format (defined at section 8.3 of [5]). A total of 853 images are included. There are 3 classes defined as "with_mask", "without_mask", and "mask_weared_incorrect".

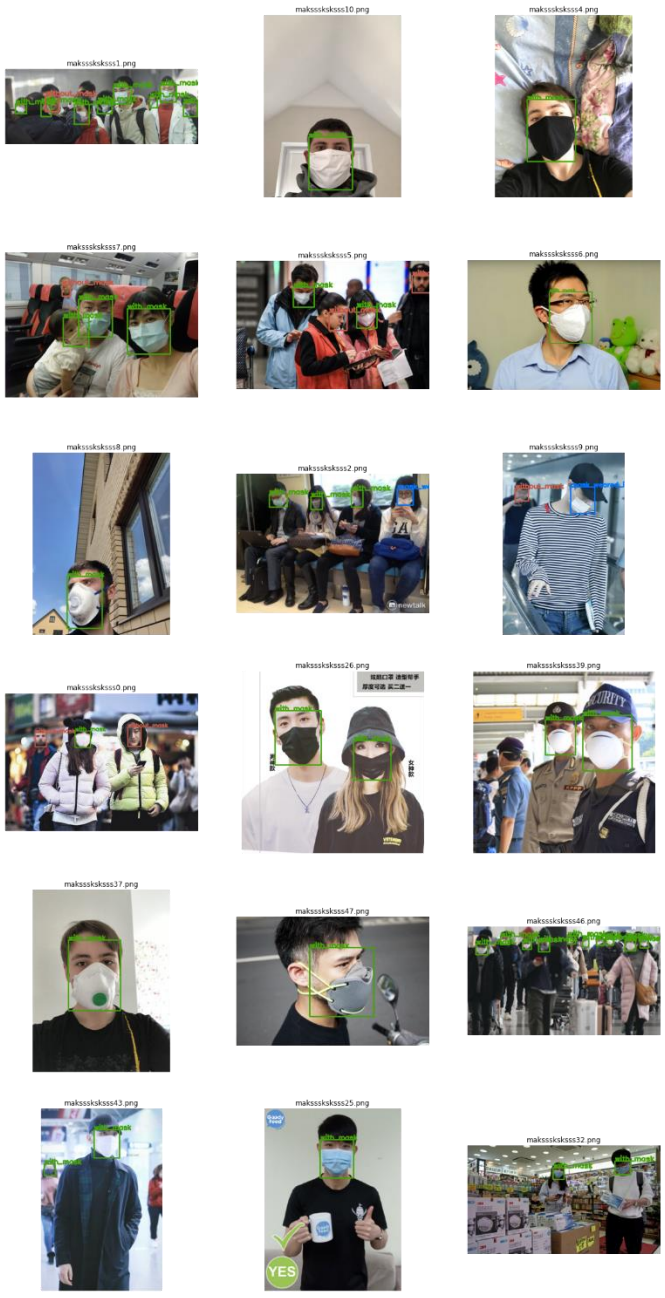


Fig. 1. Examples of images in dataset.

B. Discussions of Data

Among all images in dataset, most of them are showing east and south-east Asian people. There are also many images with the same person (e.g. the 2nd and 3rd image in the 1st row, Fig. 1). This might make the trained model biased to perform better on east & south-east Asian people & those frequently appeared people.

The dataset is called "Mask Dataset". According to some resources (e.g. US FDA [6]), respiratory protection methods include face/surgical masks, face coverings, and respirators. However, various types of protection methods more than face or surgical masks appear in this dataset. Due to the various looks of these types of protection methods, a potentially better dataset might differentiate these methods. In this way, the later trained model might be better capable of finding features and distinguishing the different characteristics among them. Image Segmentation Methods

We applied 4 popular traditional image segmentation methods to do the task of estimating area of leaves with pre-processed images as explained before.

C. Removing Images from Dataset

After manually checking all images in the dataset, we decided to remove some images from the dataset. The main reasons of discarding them are: (*image/annotation ids*)

- Tiny OR ambiguous labeled images: 3, 11, 36, 52, 64, 89, 91, 159, 194, 221, 255, 277, 294, 382, 486, 719, 724, 754, 823
- Debatable image: 432
- Wrongly labeled image: 634
- Repeated images (*more images still missing*): 354, 431, 466, 604, 624, 690

Take image “maksssksksss64.png” with its annotation “maksssksksss64.xml” (Fig. 2) as an example. Fig. 2-b is the annotated image with green rectangles marking the boundaries of each objects. There are some very small bounding boxes in the background that are difficult to tell what are them. Fig. 2-a is a zoomed area of the background. It is hard to tell, even in human eyes, if these people are wearing masks or correctly wearing masks. Labelling those objects as “with_mask” would be misleading to the models. As the image would further downscale to smaller sizes, each of these people’s faces with or without masks might be represented by one or two pixels and would be not practical in our context.

Another example is image “maksssksksss432.png” with its annotation “maksssksksss432.xml” (Fig. 3). We removed this image from the dataset because it would be debatable to tell if the person covering faces with hands is using a face mask or covering.

The last example to show is image “maksssksksss634.png” with its annotation “maksssksksss634.xml” (Fig. 4). This is a wrongly labelled image with the bounding box not covering an object.

There are also some repeated images and we decide to remove them from the dataset.

a

b

Fig. 2. maksssksksss64.png

Fig. 3. Maksssksksss432.png

Fig. 4. maksssksksss634.png

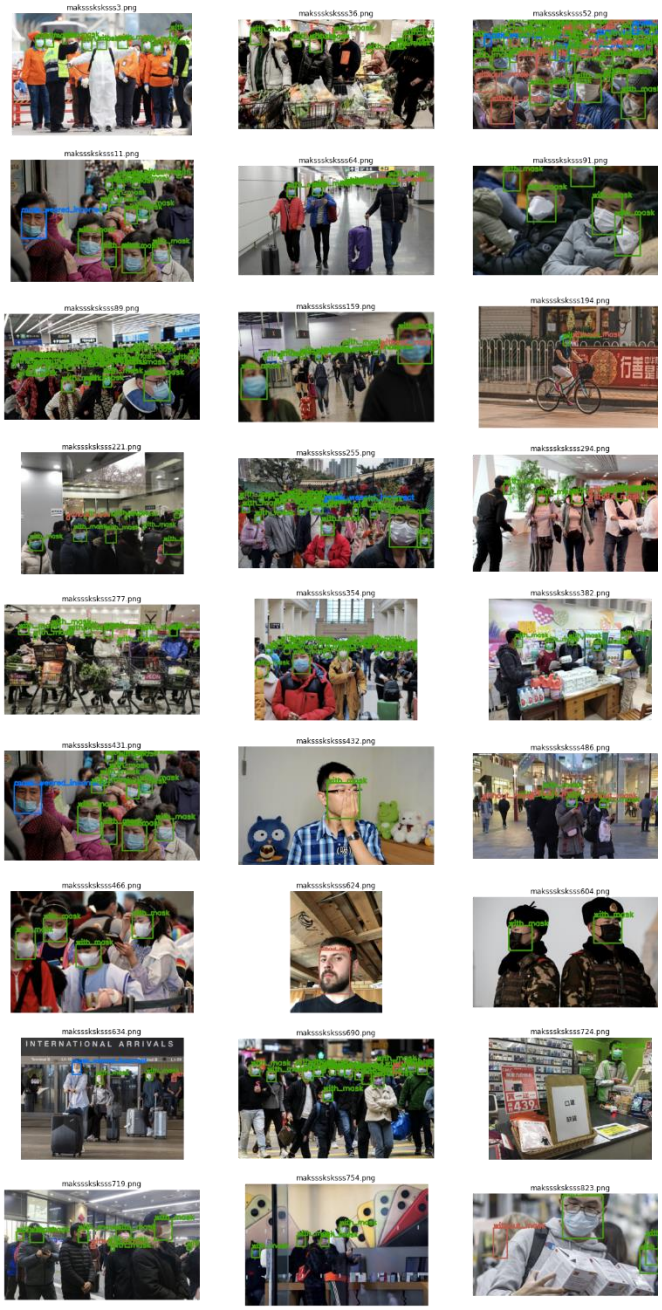


Fig. 5. All discarded images in dataset.

III. Data Pre-Processing

A. Annotations Reformatting

Provided dataset [4] only provides PASCAL VOC format annotations. However, the selected model is YOLOv5 by Ultralytics requiring another format of annotations:

```
<class> < x_center> < y_center> <width> <height>
```

where such one row represents one object. Above coordinates must be in normalized xywh format (from 0 - 1). (refers to section 1.2 of [7])

For this step of reformatting, we define a function to perform this transformation across all annotations in dataset.

B. Dataset Partition

Next is the partition step where we split both images and annotations into separate training, validation, and testing sets. Number of images assigned to each set is specified as below:

- Training set (~70% of the images)
- Validation set (~15% of the images)
- Testing set (~15% of the images)

C. Data Augmentation

When performing computer vision tasks with deep learning neural network models, a large number of training data enables models to effectively learn. However, it is often expensive and laborious to collect more data. Recently there has been more extensive use of generic data augmentation to improve performance of deep learning models. [8]

In our task, we apply some common data augmentation techniques under color space transformations, geometric transformations, and random erasing these 3 categories [9] on our datasets.

Color space transformations:

- Image HSV-Hue augmentation – config. value: hsv_h: 0.015
- Image HSV-Saturation augmentation – config. value: hsv_s: 0.7
- Image HSV-Value augmentation – config. value: hsv_v: 0.4

Geometric transformations:

- Image translation (+/- fraction) – config. value: translate: 0.1
- Image scaling (+/- gain) – config. value: scale: 0.5
- Flip image to left or right (probability) – config. value: fliplr: 0.5

Random erasing:

- Image mosaic (probability) – config. value: mosaic: 1.0

IV. Model Training & Performance

We choose YOLOv5 [3] object detection model to train on our processed dataset. Despite the controversy behind YOLOv5 is just due to its choice of name, it does not take away the fact that this is after all a great YOLO object detection model ported on PyTorch. For our task, YOLOv5 has its great advantage due to its ease of use and fast developing cycle. [10]

A. *Configurations*

We perform training of the YOLOv5 model 2 times with or without discarding images. Model training configurations are shown as below:

```
--img 416
--batch 32
--epochs 100
--cfg yolov5s.yaml
--data "/content/drive/MyDrive/CV5330/L4/MaskPascalVOC/data.yaml"
--hyp "/content/drive/MyDrive/CV5330/L4/MaskPascalVOC/hyp.scratch.yaml"
--weights yolov5s.pt
--cache
```

B. *Model Performance*

Result of the 2 runs are shown as below:

1) *Run 1: use all images from dataset:*
   a) *Metrics during the training-Fig. 6-a*
   b) *Training performance-Table I*
   c) *Testing set result-Fig. 7-a-1 & Fig. 7-b-1*
2) *Run 2: discard all mentioned images in section II.-B.:*
   a) *Metrics during the training-Fig. 6-b*
   b) *Training performance -Table II*
   c) *Testing set result-Fig. 7-a-2 & Fig. 7-b-2*

V. Conclusion & Future Work

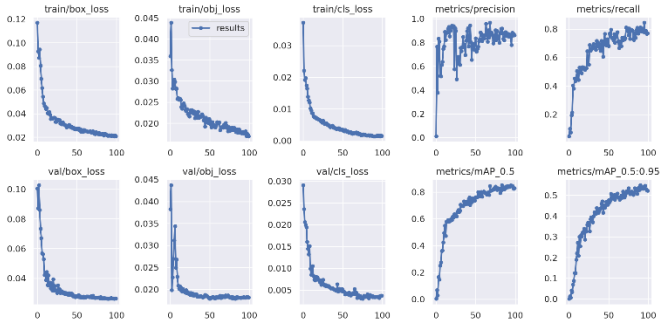
This project's results demonstrate that deep learning neural networks like YOLOv5 are effective to detect if people are wearing face masks or correctly wearing them.

Preserving all images in dataset rather than removing some has better performances. This shows our removal of images has counter effects in training the model. Since the scale of our chosen dataset is too small – there are only fewer than 900 images in the dataset. Collecting more data and label them correctly would be the next step to achieve better performance.

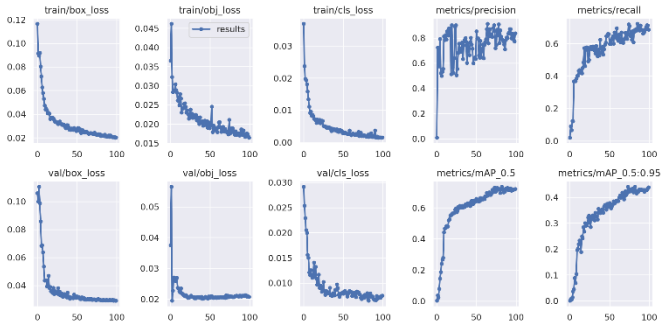
Fig. 8 shows an imbalanced labels distribution across all annotations in dataset. A specific data augmentation for less common labels [11] could be applied to further enhance our trained model.

From Fig. 9, we can see the model having a high tendency to detect "weared_mask_incorrect" and background as "with_mask". This may be caused by inaccurate or ambiguous data annotations as shown in section II-B. Future works needs to ensure a clearer criterion when labelling images.

As explained in section IV., we selected YOLOv5 with its ease of developing advantage. A comparison of other models would be helpful to determine which model can better achieve our goal of detecting people wearing face masks and coverings.



a. Run 1



b. Run 2

Fig. 6. Model training metrics diagrams.

TABLE I. Run 1 Validation Results

YOLOv5s summary: 213 layers, 7018216 parameters, 0 gradients, 15.8 GFLOPs						
Class	Images	Labels	P	R	**mAP@.5**	mAP@.5:.95
all	128	645	0.859	0.811	**0.846**	0.548
without_mask	128	151	0.873	0.821	**0.854**	0.495
with_mask	128	477	0.91	0.931	**0.948**	0.646
mask_weared_incorrect	128	17	0.794	0.682	**0.737**	0.504

TABLE II. Run 2 Validation Results

YOLOv5s summary: 213 layers, 7018216 parameters, 0 gradients, 15.8 GFLOPs						
Class	Images	Labels	P	R	**mAP@.5**	mAP@.5:.95
all	124	777	0.787	0.685	**0.736**	0.441
without_mask	124	205	0.724	0.707	**0.749**	0.383

with_mask	124	545	0.837	0.905	**0.91**	0.583
mask_weared_incorrect	124	27	0.8	0.444	**0.55**	0.357

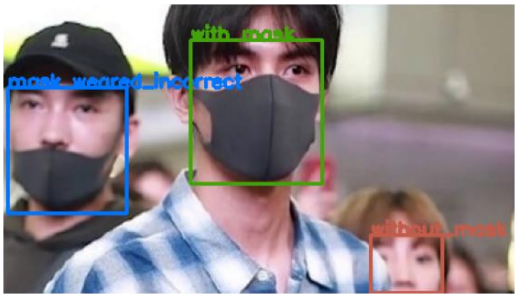
a-1. Model detection-1st run-maksssksksss829.png

a-2. Model detection-2nd run-maksssksksss829.png

a-3. Ground truth-maksssksksss829.png

b-1. Model detection-1st run-maksssksksss307.png

b-2. Model detection-2nd run-maksssksksss307.png



b-3. Ground truth-maksssksksss307.png

Fig. 7. Model weights detection images.

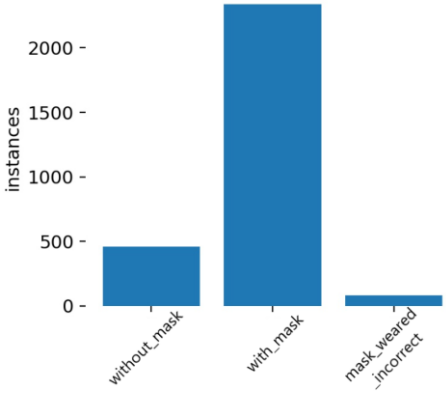


Fig. 8. Dataset labels distribution.

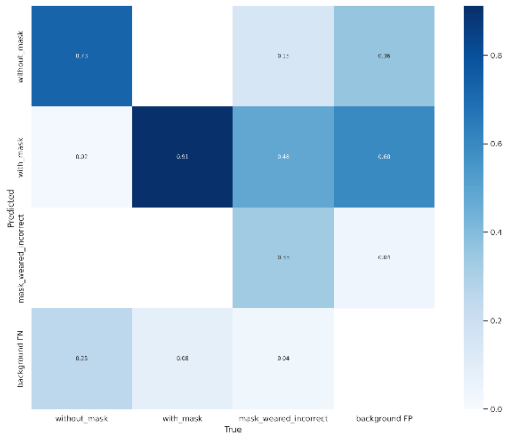


Fig. 9. Model training confusion matrix-2nd run.

[1] Guardian News and Media. (2020, October 2). The most common ways we're wearing face masks incorrectly. The Guardian. Retrieved March 31, 2022, from https://www.theguardian.com/world/2020/oct/02/the-most-common-ways-were-wearing-face-masks-incorrectly

[2] Yang, C.-W., Phung, T. H., Shuai, H.-H., & Cheng, W.-H. (2022). Mask or Non-Mask? Robust Face Mask Detector via Triplet-Consistency Representation Learning. ACM Transactions on Multimedia Computing, Communications, and Applications, 18(1s), 1–20. https://doi.org/10.1145/3472623

[3] ultralytics/yolov5. (2020, August 21). GitHub. https://github.com/ultralytics/yolov5

[4] Mask Dataset | MakeML - Create Neural Network with ease. (n.d.). Makeml.app. Retrieved March 31, 2022, from https://makeml.app/datasets/mask

[5] Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2009). The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, *88*(2), 303–338. https://doi.org/10.1007/s11263-009-0275-4

[6] Health, C. for D. and R. (2021). N95 Respirators, Surgical Masks, Face Masks, and Barrier Face Coverings. *FDA*. https://www.fda.gov/medical-devices/personal-protective-equipment-infection-control/n95-respirators-surgical-masks-face-masks-and-barrier-face-coverings

[7] *Train Custom Data · ultralytics/yolov5 Wiki*. (n.d.). GitHub. Retrieved April 1, 2022, from https://github.com/ultralytics/yolov5/wiki/Train-Custom-Data#12-create-labels-1

[8] Taylor, L., & Nitschke, G. (2018, November 1). *Improving Deep Learning with Generic Data Augmentation*. IEEE Xplore. https://doi.org/10.1109/SSCI.2018.8628742

[9] Keita, Z. (2021, April 13). *Simple Image Data Augmentation Technics to Mitigate Overfitting In Computer Vision*. Medium. https://towardsdatascience.com/simple-image-data-augmentation-technics-to-mitigate-overfitting-in-computer-vision-2a6966f51af4

[10] Maindola, G. (2021, June 20). Introduction to YOLOv5 Object Detection with Tutorial. MLK - Machine Learning Knowledge. https://machinelearningknowledge.ai/introduction-to-yolov5-object-detection-with-tutorial/

[11] Saini, M., & Susan, S. (2020). Deep transfer with minority data augmentation for imbalanced breast cancer dataset. Applied Soft Computing, 97, 106759. https://doi.org/10.1016/j.asoc.2020.106759