Face Mask Detection in Crowd Settings

Vinay Garimella
UVA School of Engineering
85 Engineers Way
Charlottesville, VA 22903
vg5ak@virginia.edu

William Peterson
UVA School of Engineering
85 Engineers Way
Charlottesville, VA 22903
wcp7cp@virginia.edu

Anthony Taylor
UVA School of Engineering
85 Engineers Way
Charlottesville, VA 22903
at8gb@virginia.edu

1. MOTIVATION

To help mitigate the spread of COVID-19, masks have become an integral part of one's daily wear. Globally, governments and authorities have put in place a series of measures to ensure that students are adhering to safe social distancing guidelines, including mask wearing. We want to use deep learning models to approximate mask compliance, as well as uncover social behaviors that may indicate the relative compliance. For example, understanding how one individual's adherence to mask wearing affect another's

2. Background

Object detection allows us to detect human figures in crowd settings. The YOLO model is a state-of-the-art object detection model that is fast and generalizable to a large set of classes, but requires a lot of computational power because of real-time capability (though there exists a few variations of YOLO that trade object detection performance for less computational demand) [5].

Another state-of-the-art model is the Faster R-CNN model. Faster R-CNN works by defining a set of anchor boxes at set locations in the training images. For each anchor box, the scores of the predefined classes are found by feeding the bounding box through a convolutional neural network. Unlike YOLO, Faster R-CNN performs both bounding box regression and classification for each anchor box. This is a large contributor to its slower prediction time. Faster R-CNN, however, does perform better than its predecessors by eliminating the use of selective search in finding the region proposals [6].

Other object detection and classification approaches have relied on traditional computer vision techniques to extract features (SIFT, HOG) to feed into simpler machine learning models. Pros for these models include simpler training, while cons include a significant decrease in performance, and less robust to high variance from noisy or poor input/camera quality.

3. Claim / Target Task

Our primary goal is to approximate mask compliance by applying deep learning models to detect frequencies of individuals wearing masks. Furthermore, our targeted environment is one of high crowd density, as opposed to separated, and smaller individual groups.

With an appropriately designed model, we wish to use predicted mask compliance to better understand how social behaviors may dictate mask compliance. For example, whether one individual's decision to wear a mask influences the remainder of their localized group's mask compliance.

The model should be able to approximate a crowd's mask compliance given current social distancing measures. With such a model, one can approximate an environment's mask wearing compliance by testing the model on a sample of images from the environment, recording frequencies of those wearing masks vs those who are not.

4. FIGURES

Figure A below provides the relative compliance to mask wearing by age, based upon a sample of individuals. While our project may not align exactly with this figure's sampling, it provides a reference for which we can compare the results of our model's predicted mask wearing compliance.

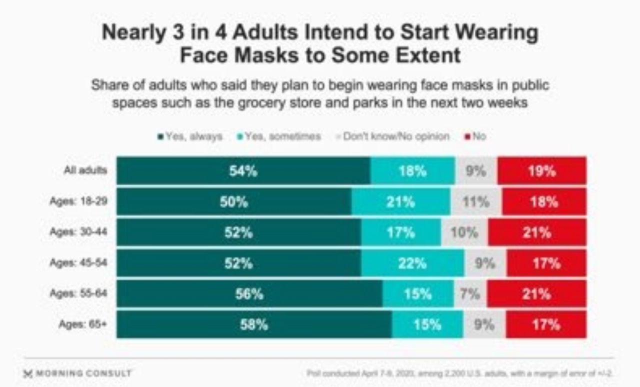
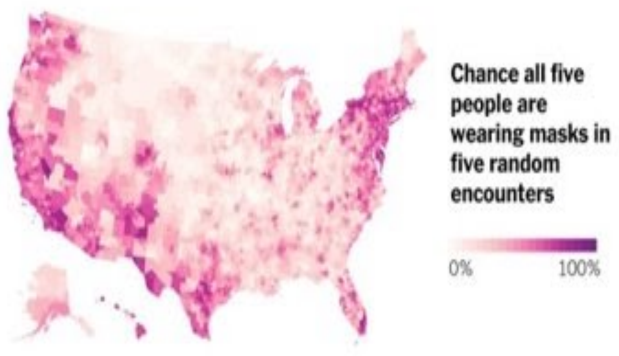


Figure B provides another reference point to compare the predictions of our model and its approximation of mask wearing compliance in the University
community.


Chance all five people are wearing masks in five random encounters
0%
100%

5. Proposed Solution

Originally, we thought to devise an approach that would require stacking models together. The original approach involved the use of a model in detecting faces, segmenting the bounding boxes, and feeding the segmented faces into a mask classification model. This, however, became problematic in that, while face detection appears to be a relatively simple problem, we could not trust the performance of such face detection models in scenarios where a mask was present. Therefore, we simplified the problem into a single step, relying on the use of state-of-the-art models for mask detection, as opposed to the original design of separating face detection and mask classification. For mask detection, we plan to explore multiple datasets where images, alongside their class and bounding box annotations, are provided [3]. The two candidate models for mask detection are YOLO and Faster R-CNN, both highly respected models in the field of object detection. To ease training, we may leverage pre-trained weights in the fine-tuning of these models; thereby, taking advantage of common transfer learning methods to expedite training.

6. Contributions

This model contributes a simpler way to detect face mask compliance and can be applied to social distancing policies, such as determining safety by group size and density of the crowd. To implement this, there needs to be better video and image data so as to accurately detect compliance.

7. Implementation

Our implementation, inline with our proposed solution, leveraged a YOLOv3 configuration and its pre-trained weights, as well as training the classification layers of a pre-trained Faster R-CNN model (provided by PyTorch's FasterRCNN module).

The Faster R-CNN was trained on a Mask Detection dataset, consisting of 853 images, each providing corresponding bounding box coordinates and class (see Data Summary) [1]. Finding a generalizable fit for this dataset, multiple training parameters were fine tuned. The optimal parameters returned are summarized as:

Parameter	**Value**
Learning Rate	**0.0045**
Epochs	**20**
Batch Size	**4**
Training Size	**800**
Number of Classes	**4**

The number of classes was found by counting the number of unique classes provided by the dataset and adding one for the background class. The classes provided by the dataset are summarized as: "Wearing mask", "Incorrectly wearing mask", "Not wearing mask" [1].

The Faster R-CNN was used to train in mask detection and the YOLO model to train in human detection. To calculate the bounding box, which points to the faces in the images, we used Non-Maximum Suppression (NMS) to make sure that there were no duplicates. This works by counting images that do not overlap too much, while discarding images that do overlap. This will be used for each image until all the faces are tested. We had our own datasets for masked and unmasked people respectively. The R-CNN model was used due to its extensive documentation, and was run on Google Colab, which sped up training tremendously [4].

While we tried to train the model with both the R-CNN and YOLO together, it proved too difficult to get accurate results while stacking the models. In separating the models, we used the YOLO model as a means of counting crowds and the Faster R-CNN to count the number of masked vs. unmasked individuals. We used two formulas to detect the minimum and maximum number of people complying with face-mask regulations. The minimum and maximum mask compliance is defined below:

Maximum Mask Compliance =

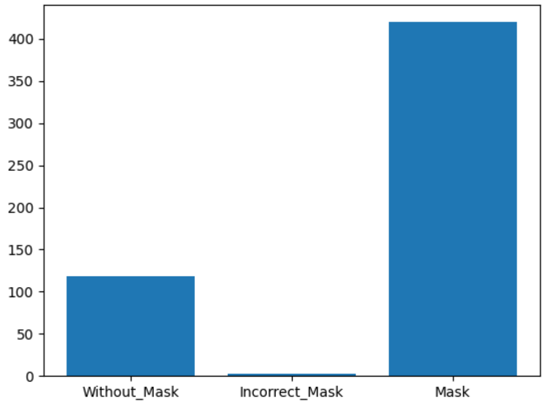
$$\frac{\text{\# of Masks Detected by Faster RCNN}}{min(\text{\# of people detected by Faster RCNN,\# of people detected by YOLO})}$$

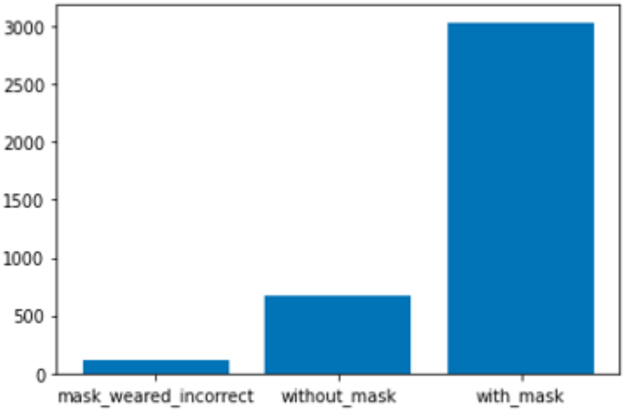
Minimum Mask Compliance =

$$\frac{\text{\# of Masks Detected by Faster RCNN}}{max(\text{\# of people detected by Faster RCNN,\# of people detected by YOLO})}$$

8. Data Summary

We used three different datasets, the first two datasets were from kaggle, and the last dataset was from a personal collection of images from Google Images. The first kaggle dataset contained 853 images of crowds wearing or not wearing masks, these images were used for training the data, with 53 of them being used for testing [1]. This dataset provided images alongside annotations to be able to train our model to detect masked and unmasked individuals, as opposed to mere classification. The second and third datasets consist of 200 and 50 images and will be used in testing both the R-CNN and the YOLO models. Note how the predicted class frequencies match the data it was trained on. This could likely be the result of two occurrences: (i)The model classifies mask wearing in the same distribution it was learnt from or (ii) the more plausible assumption would be that both datasets contain similar distributions, as both are focused on mask wearing.





Figures 4 & 5: Note*: The order of the different classes is not the same in each figure.*

9. Experimental Results + Baselines (state-of-the-art)

Comparing our model against the ground truths (see Fig. 5), we find that the Faster R-CNN performed quite well in detecting masks, returning an average recall of 0.969 from each tested threshold for mask detection. It did, however, suffer from a decreased performance in detecting absence of masks and masks being worn incorrectly, its average recall in detecting absence of masks being 0.735. This is largely due to the lack of data provided in these two classes, as images of individuals wearing masks remained a dominant class throughout the training of the model [1]. Therefore, a larger dataset and one in which each class has a uniform distribution could likely improve the model's performance.

With our YOLO model, we found that it was able to detect individuals very well, in both high and low crowd density settings. A key component in the YOLO's performance was a suitable Non-max Suppression threshold. A Non-max threshold too high proved to overestimate the number of individuals in an image as fewer bounding boxes were being discarded despite overlapping borders. However, a Non-max threshold that was too low would not work well in a crowded environment where overlapping bounding boxes are to be expected, consequently underestimating the number of individuals.

	Mask Recall	**Non-Mask Recall**	**Human Recall (Masked Individuals**	**Human Recall (Non-Masked Individuals)**
Faster R-CNN (threshold = 0.75)	0.979	0.69	0.925	0.949
Faster R-CNN (threshold = 0.9)	0.959	0.78	0.906	0.949
YOLO (NMS threshold = 0.1)	—	—	0.84	0.925
YOLO (NMS threshold = 0.9)	—	—	0.44	0.60

Figure 5: This figure shows the recalls for both models on our second dataset [3]

10. Experimental Analysis

A couple issues that we had estimating mask compliance is that the number of people detected by the YOLO model was always less than the total number of people in the image. To better detect people we increased the NMS threshold, this gave more room for the bounding boxes to overlap.

There was one dataset [3] which was inconsistent with the original crowd density of the first dataset, of which the R-CNN model was trained on. This might have contributed to a drop in performance due to the different styles of images, i.e. crowd images vs close ups. An increase in images as well as better distribution of masked vs. unmasked individuals would allow for a more generalizable model in our opinion.

Some issues with the results are that the R-CNN is difficult to apply to livestreams due to being extremely slow compared to models for live detection. This could be improved by more processing power, parallelization, or changing to code to C++. There also needs to be some data which provides context for each image, such as date, time of day, and location. Increasing the threshold decreased the number of mask detections but also decreased the number of false positives, and increasing the NMS threshold worked well for a large overlap of people, but should be decreased for a lower density.

11. Conclusion and Future Work

We set out to use deep learning models to approximate mask compliance and uncover social behaviors that may indicate the relative compliance within a given social setting. Our initial approach of stacking models together via (i) modeling face detection, (ii) segmenting bounding boxes, and (iii) then feeding the segmented faces into a mask classification model lead to unstable face detection performance. To mitigate performance instabilities, we leveraged existing state-of-the-art models for mask detection. Our departure from separating face detection and mask classification allowed us to analyze three independent datasets: two datasets from Kaggle [1,3], and a third proprietary collection of images amassed from Google Images.

The experimental results of the 2nd dataset illustrate that, within the 0.1 NMS threshold, our YOLO model was able to identify a mean individual count with ~88.2% accuracy (see Fig. 5 & 6). These findings were apparent in both high and low crowd density settings. YOLO's moderate Non-max Suppression threshold was the key contributor to the model's performance.

Additionally, the experimental results indicate that our Faster R-CNN mask detection model has ~96% recall given a 0.9 threshold (see Fig. 5 & 6). It did, however, suffer from a decreased performance in detecting the absence of masks and the presence of masks being worn incorrectly. This is a consequence of the training set's dominance of individuals wearing masks. A larger and uniformly distributed dataset could likely improve the model's performance.

Our findings lead to the following inquiries into candidate future works:

Further inquiry into understanding how the Non-max Suppression threshold for the YOLO model interplays with the Faster R-CNN's Non-max Suppression threshold and the resulting performance would perhaps yield better model performance when stacking the YOLO and Faster R-CNN models together.

Further research into how the context from the image's setting affects mask compliance could lead to increasing understanding of mask compliance trends. For example, mask protests would often imply worse mask compliance compared to the average sample image. Such a project would involve classifying environments into a series of categories and predicting their associated mask compliance to get a better understanding of the relationship between setting and mask compliance. Could using such an approach, where setting/environment is included in the mask compliance prediction, improve results?

Would be interested how predicted mask compliance would change if given images from multiple time periods (early March vs. October, for example). This would require much more extensive datasets, including metadata on the time frame for each image.

12. REFERENCES

[1] Andrewmvd. 2020. Face Mask Detection. (June 2020). Retrieved December 6, 2020 from https://www.kaggle.com/andrewmvd/face-mask-detection

[2] Joseph Redmon, Ali Farhadi. 2018. YOLOv3: An Incremental Improvement. arXiv:1804.02767. Retrieved from https://arxiv.org/abs/1804.02767

[3] Prithwiraj Mithra. 2020. Covid Face Mask Detection Dataset. (July 2020). Retrieved December 6, 2020 from https://www.kaggle.com/prithwirajmitra/covid-face-mask-detection-dataset

[4] PyTorch. 2017. Torchvision Object Detection Finetuning Tutorial. Retrieved December 6, 2020 from https://pytorch.org/tutorials/intermediate/torchvision_tutorial.html

[5] Raphael Holzer. 2019 YOLO – object detection. Retrieved December 6, 2020 from https://opencv-tutorial.readthedocs.io/en/latest/yolo/yolo.html

[6] Rohith Gandhi. 2018. R-CNN, Fast R-CNN, Faster R-CNN, YOLO — Object Detection Algorithms. (July 2018). Retrieved December 6, 2020 from https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e

[7] Will Peterson, Vinay Garimella, Anthony Taylor. 2020. Faster R-CNN and YOLO modelling. (March 2001). Retrieved December 6, 2020 from https://github.com/willyptrain/cs4774-mask-detection