A State-of-the-Art Approach to Face Mask Detection in Crowd Settings

Vinay Garimella | Will Peterson | Anthony Taylor

Motivation

- To help mitigate the risk of COVID-19 spread, 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

Background

YOLO:

1. Pros: Fast and generalizable to a large set of classes, can perform classification and bounding box detection at the same time/in the same step
2. Cons: Requires a lot of computational power because of real-time capability (though there exist a few variations of YOLO that trade object detection performance for less computational demand)

Traditional Approaches:

1. 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
    a. Pros: Simpler training
    b. Cons: Significant decrease in performance, less robust to high variance from noisy or poor input/camera quality

Background

Faster R-CNN: Define a set of anchor boxes for set locations in the training images. Each of the boxes from the anchor points correspond to bounding boxes. The scores of each bounding box are calculated by passing the box through a CNN. Faster R-CNN is often slow because of this split process of calculating boxes and relying on a CNN prediction for each box

a. Pros: Fewer region proposals compared to previous R-CNN, does not require selective search to find region proposals
b. Cons: Much *slower* than YOLO, cannot perform classification and bounding box detection at the same time (bounding box first, then regions are classified)

Target Task

- Develop a model that can not only detect and segment humans from images but develop a model applicable for real-time crowd video-monitoring
- Try to reveal certain patterns in mask wearing to better understand how crowds comply to social distancing measures
- Predict an approximate level of compliance by testing samples of images and recording the frequencies of those wearing masks vs. those who are not

Overview

Approximate mask compliance by applying deep learning models to detect frequencies of individuals wearing masks vs. not wearing masks.

Claim

- Our primary goal is to create a model that will accurately detect mask wearing in a crowd setting
- If contextual information is provided, we aim to better uncover how individual mask compliance may influence the entirety of the crowd's compliance

An Intuitive Figure Showing WHY needed

An Intuitive Figure Showing WHY needed

Proposed Solution

- Originally, our proposed solution involved the detection faces to be then fed into a model that would classify whether the face contained a mask or not. We later abandoned this approach and have since adopted a new strategy:
  - Mask Detection: Rather than having two separate steps for face detection and mask classification, we thought to combine these steps into one. Our new proposed solution involves the use of a state-of-the-art detection model trained on a *mask-detection* dataset, consisting of class and bounding box information for each image.
  - For our model, we plan to investigate the use of Faster R-CNN and YOLO as potential candidates for training on the Mask Detection dataset.
  - We will also explore the use of pre-trained weights in the training of these models and merely finetune the classification layers of the models

Implementation

The implementation involved:

- Transfer Learning with Faster R-CNN – loading old weights and training
- Reusing the Pre-Trained YoloV3 Weights and Config

Non-Maximum Suppression: In calculating the bounding boxes, an important step is to eliminate certain overlaps between boxes so that duplicates do not arise

- For this, we can use Non-Maximum Suppression. Non-max Suppression works by grabbing the bounding box with the highest confidence score that has not already been used. With this bounding box, we calculate the IOU (intersection/overlap area divided by union area) for each box that crosses our bounding box's border. If the calculated IOU is *greater* than our NMS-defined threshold, we discard it, and if below, we include it in our prediction. We continue this process until we have examined all boxes *or* the next maximum scored bounding box is less than our defined threshold

Implementation

Faster R-CNN:

- We chose to use Faster R-CNN in training on the Mask Detection dataset. The main reason being the extensive documentation from PyTorch and its assisting Faster R-CNN model class
- Training / Testing:
  - **Learning Rate:** 0.0045, **Epochs:** 20, **Batch Size:** 4, **Training Size:** 800, **Number of Classes:** 4
  - Thought of augmenting images in hopes to increase training set size and possibly improve generalizability – however did not use this approach because of fears of bounding boxes no longer matching up correctly.
  - Trained on Google Colab Pro's GPU services – sped up training tremendously
  - Tested on own data using mere observation

Implementation

YOLO Model:

- The YOLOv3 weights and configuration was downloaded from https://pjreddie.com/darknet/yolo/
  - The weights and configuration were based off YOLOv3's training on the COCO dataset (80 classes)
- OpenCV's DNN module was used to load the YOLO's weights and model configuration
- The output from the YOLO model was split into its bounding boxes and classes

a. The person score for each detection was found by indexing the predicted class list with the index of “person” in our predefined coco.names file. The bounding box associated with the person score that met the threshold standard (0.70) was included in the Non-Maximum Suppression step

- YOLO was used to detect human figures from crowds
- We did make an attempt to stack the Faster R-CNN and YOLO models together, where the YOLO would segment human figures from the crowd and each human figure was fed into the Faster R-CNN model for mask detection – This proved to be a slow and at times, worse performing approach

Implementation

- **Instead of stacking the models, each was used independently**
  - The Faster R-CNN was used for mask detection
  - The YOLO model for human detection
  - We can, however, use each to define a “range” of mask compliance with the predictions of each

- **We define *predicted* minimum and maximum mask wearing compliance as:**
  - Maximum Mask Compliance = $$\frac{\text{\# of Masks Detected by Faster RCNN}}{min(\text{\# of people detected by Faster RCNN}, \text{\# of people detected by YOLO})}$$
  - Minimum Mask Compliance = $$\frac{\text{\# of Masks Detected by Faster RCNN}}{max(\text{\# of people detected by Faster RCNN}, \text{\# of people detected by YOLO})}$$

Data Summary

Three datasets were used, summarized in the next two slides:

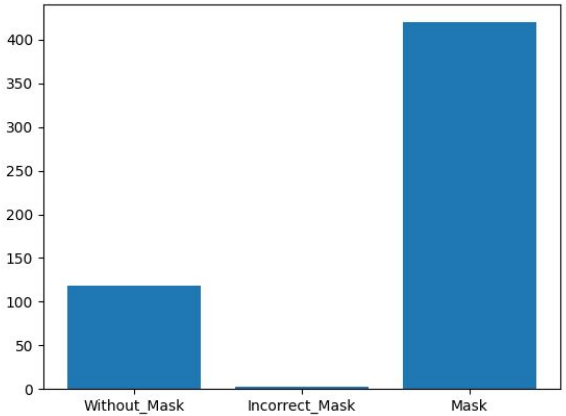
1. https://www.kaggle.com/andrewmvd/face-mask-detection
    a. This dataset consisted of 853 images of crowds either wearing or not wearing masks (or both!)
        i. 800 of the images were used in training the Faster R-CNN, the remaining 53 were used for manual testing and validation
        ii. Annotations were provided, listing the class (“wearing mask”, “mask is being worn incorrectly”, or “no mask”)
            1. The model was trained on 4 classes: # of classes above + 1 class (for background)
        iii. Later, all images were used to measuring the approx. mask compliance by predicting number of people wearing masks out of all persons detected

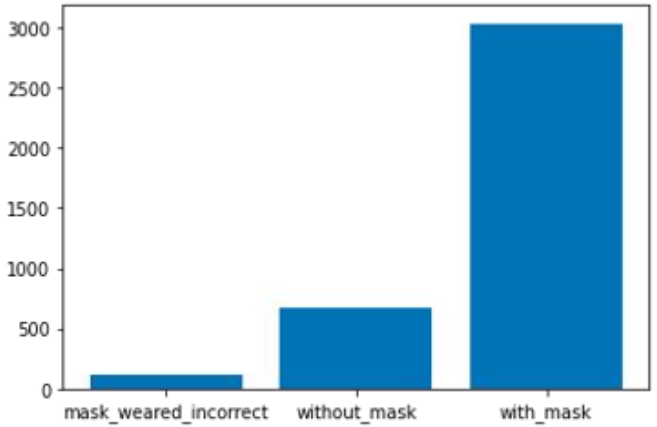
Data Summary (cont.)

2. https://www.kaggle.com/prithwirajmitra/covid-face-mask-detection-dataset
    a. 300 images were taken from this dataset and used in testing the Faster R-CNN and YOLO models
        i. The YOLO model being tested on its ability to detect humans
        ii. The Faster R-CNN being tested on its ability to detect masks and used the counts for each class as approximation for total person count
3. Personal collection from scraping Google
    a. Found 50 images for use in testing the Faster R-CNN and YOLO in measuring mask compliance

Data Summary (cont.)

- The predicted class frequencies matches the data it was trained on
- This could likely be the result of two occurrences: (1) The model classifies mask wearing in the same distribution it was learnt from *OR (2)* the more plausible assumption would be that both datasets contain similar distributions, as both are focused on mask wearing
- ***Note**: The order of the different classes is not the same in each figure.*





Experimental Results

	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

Experimental Results

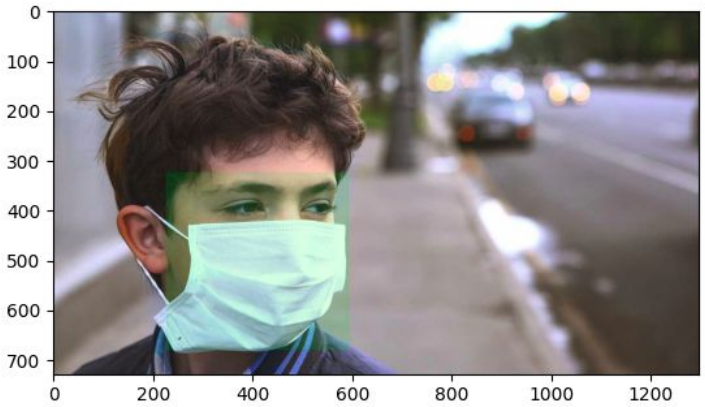
- Comparing our model against the ground truths, we find that the Faster R-CNN performed quite well in detecting masks. It did, however, suffer from a decreased performance in detecting absence of masks and masks being worn incorrectly. This is largely due to the lack of data provided in these two classes, as images of individuals wearing masks remained a dominant class throughout the training of the model.
    - 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.
    - A Non-max threshold that was too low suffers in high crowd density environments by underestimating persons

Code Walkthrough

Faster R-CNN & YOLO Walkthrough:
https://colab.research.google.com/drive/1KJ-ScEVDvn0oOugBA-nSA5ArzoryfbQe#scrollTo=8dpqR8v9zyrj

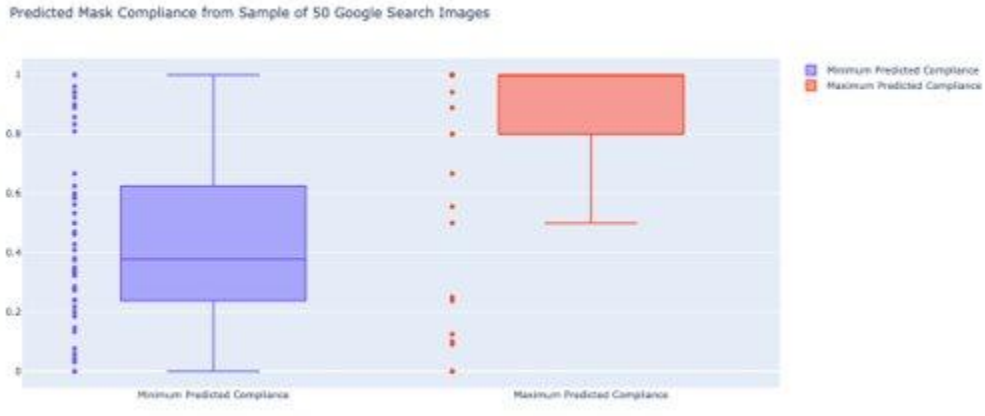
https://github.com/willyptrain/cs4774-mask-detection/blob/new_master/yolo_and_rcnn_example.ipynb

Example Model Predictions



High vs. Low Crowd Density

Effect of Setting: Mask Protest likely implies little to no mask wearing compliance



Experimental Analysis

Approximation of Mask Compliance on two sets of data, using the previously defined Minimum and Maximum Mask Compliance formulas

Experimental Analysis

Issues affecting results:

- A large contributor to error in estimating mask compliance was the way in which we defined mask compliance
  - Number of people detected was often lower than truth
  - Increasing NMS threshold helped in increasing person count detected by YOLO model
  - Could maybe provide a level of confidence based upon scores recorded for each bounding box and label
- Domain Generalization:
  - One dataset (see slide 16, #2) was not consistent with the type of images the Faster R-CNN model was trained on. This may have been a likely contributor to any drop in performance as the model was accustomed to crowd images and not single face closeups
  - Domain Generalization, where data is gathered from more than one source and differs in either setting or style (crowd vs. single closeup, etc.) may provide a more generalized model that is able to generalize to a wide variety of datasets

Experimental Analysis

- Faster R-CNN is difficult to apply to live streams as predictions are much slower to more commonly used models for live detection (i.e. YOLO)
    - A potential improvement in speed may come from translating the code from Python to C++ as the latter is often much faster
- Increasing the size of the dataset, and more specifically the number of samples where persons are either wearing a mask incorrectly or not at all are necessary. Current mask detection datasets are often poorly documented or lack bounding box information.
    - Data that provides contextual background behind each image: Date, Context, Time of Day, Location, etc.
- Faster R-CNN: Increasing the threshold did decrease the number of mask detections, *but* it also limited the number of false positives
- YOLO: Increasing the NMS threshold worked well in crowd settings as bounding boxes are expected to overlap, however the NMS threshold may be decreased in settings where crowd density is lower as overlapping bounding boxes are less likely

Contributions

- Provides a mean of approximating level of compliance to mask wearing policies in a large crowd settings
    - Performance optimizations should be made before applying the model to real-time video monitoring
- Serves as a baseline for object detection models in detecting masked vs. unmasked individuals
- Our evident need for improved datasets will hopefully promote greater contribution in this area, largely for simply increasing the number of images and improving distribution of classes

Conclusion

- 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.

Conclusion and Future Work

- When stacking the YOLO and Faster R-CNN models together, we'd be curious to understand 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 be very interested in understanding how the context from the image's setting affects mask compliance. 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

References

1. https://github.com/willyptrain/cs4774-mask-detection
2. https://www.kaggle.com/prithwirajmitra/covid-face-mask-detection-dataset
3. https://www.kaggle.com/andrewmvd/face-mask-detection
4. https://pytorch.org/tutorials/intermediate/torchvision_tutorial.html
5. https://pjreddie.com/darknet/yolo/
6. https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e
7. https://opencv-tutorial.readthedocs.io/en/latest/yolo/yolo.html