CSCI 1430 Final Project Report:
Cover Your Nose

Samuel Wilkins, Shiqin Yan, Xiaoyan Zhao, Zhehui Liang.
Brown University

Abstract

The goal of this project is to create a web App identifying whether a person is wearing the mask in a proper way given an input image or video, especially having his/her nose covered. After trying and training various models, we adopted a PyTorch implementation of Single Shot MultiBox Detector. On top of this, we added Haar feature detectors to detect nose in the input image. We also created a website with our models running on the back end. And we made a video showcasing how our app can be used.

1. Introduction

Under the current Covid-19 pandemic, all people are required to wear masks to cut off all possible paths through which the virus might propagate. However, sometimes it turns out some people don't wear masks properly. Indeed, if the nose or mouth is exposed, one remains susceptible to the virus. So, we implemented this model to help people quickly determine whether they wear masks correctly.

The following is a flow-chart process:

1. Input images or video
2. Face Recognition in small boxes
3. Detection of mask in those boxes;
    - No Mask;
    - With mask, but also detecting a nose (Mask worn improperly);
    - With mask, no noses detected (Mask worn correctly)

2. Related Work

For the face detection part, we refer to the known Haar Cascades classifier, where this weblink provides some most common-used classifiers. You can also find a tutorial here as an alternative to HAAR.

For the mobilenet pipeline we added to the code in project 4, we refer to the paper here [1], though this was not included in our final pipeline.

First, we implemented a Fast-RCNN model. The original paper can be found here [2], and *the guidelines / tutorial that informed our implementation and training can be found here*.

We then focused on SSDs: [3] is the original paper underpinning the various implementations. First we explored, this Colab tutorial for a general (not mask-specific) SSD implementation that was informative but ultimately unused. There are also several other great SSD ports that are sources of inspiration that we list here but aren't listed in the reference part: Chainer, Keras, MXNET and Tensorflow. *The SSD we actually trained was adapted from this implementation*, and for the relatively light-weight pre-trained SSD mask detector *that we ultimately deployed after trying our own, we referred to the AIZOO mask model implementation over here*, which you can also try out independently on this demo website.

As for our dataset, we sourced images from the dataset associated with the AIZOO model (consisting of WIDER FACE Dataset and MAFA dataset) as well as the Dataset1 from Kaggle. So, aggregating the two, we get 6959 images (AIZOO's training set and Kaggle dataset) for training, and 1836 images (AIZOO's validation set) for validation.

Besides, here [4] also provides a good mask detector which was an early inspiration for our model.

3. Method

From the process stated in Introduction section, our pipeline would require a face / mask detector, a nose detector and ideally an eye detector for additional nose validation.

For the face / mask detection part, we implement three models listed as below:

1. The MobileNet model. This is a TensorFlow training pipeline made by adding new headers to the code from our project 4. It is included for thoroughness, but was never trained on a mask-related task.

2. A faster-RCNN-based mask detector. This model would intrinsically be more heavy weight than an SSD due to its ResNet 50 backbone and internal region proposal networks. After being trained on both the Kaggle1 and combined dataset, it achieved reasonable results on training images, but did not generalize well. Training on the combined dataset took drastically longer, so it was trained for fewer epochs, and thus actually performed worse (even on train images) than when trained on the smaller Kaggle1 dataset. See Section 3 for implementation reference links.

3. SSD mask detector implemented in PyTorch. This is more heavy weight than MobileNet, though less so than the Faster-RCNN, and would likely need to be served rather than deployed in the client. We will mainly discuss the preformance of SSD mask detector in the following “Results” section. Generally, it trained faster and performed better on the combined dataset than the RCNN.

Next, for the nose and eye detection functionality, we used the HAAR cascade classifiers in the OpenCV library here. Once we have detected a face, we ‘crop’ out regions of interest (ROI), or bounding boxes, for each face. We then run each ROI through the nose and eye detector, and record the results. This is done for efficiency, so that the HAAR detectors only process areas in which a face has been detected. For all HAAR results, we use single class non-maximum-suppression, and then return the largest nose. If the eye detection is successful, which it often is, we further validate the nose by selecting the largest nose below the pixel plane established by the eyes.

The following picture is a general example of a facial coordinate map that saved in CV2 library, which is used for facial landmark detection.

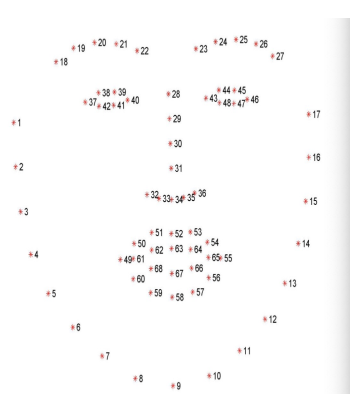


Figure 1: Facial coordinate map saved in CV2 library

4. Results

4.1. RCNN Mask detector

The following is a sample output for the RCNN mask detector on a training image.

Figure 2: RCNN mask detection result

4.2. SSD Pytorch mask detector

The following is a sample output for the trained PyTorch SSD mask detector on the combined dataset.

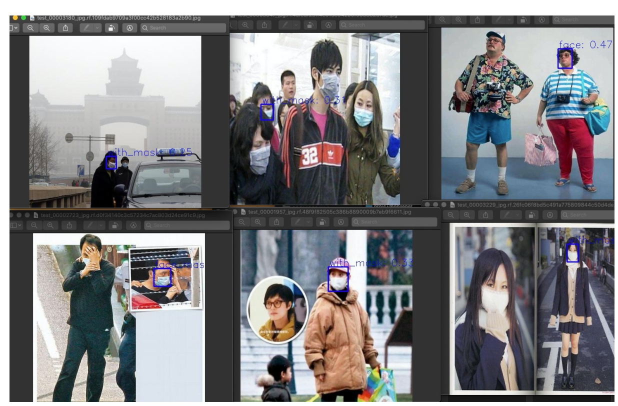


Figure 3: PyTorch SSD mask detection result

4.3. Combination of face / mask detector and nose detector

Here are some outputs from our full (pre-trained) pipeline for images with detectable noses.

(a) Nose picture1

(b) Nose picture 2

(c) Nose picture 3

And here are some outputs from our full (pre-trained) pipeline for images without detectable noses.

(a) No Nose picture 1

(b) No Nose picture 2

(c) No Nose picture 3

And the following is an output from running mask detection (from the pre-trained AIZOO PyTorch SSD, not the PyTorch SSD we trained) on multi-faces in a single image.

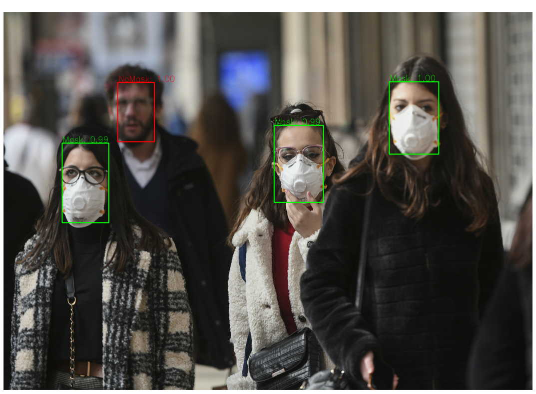


Figure 6: pre-trained mask detection on multi-faces

5. Discussion

5.1. Issues and interesting questions raised by our model training and exploration

Answers: The following section enumerates challenges that we solved in the course of this project, as well as those that could be addressed going forward.

Challenges overcome

- Because the Haar detectors can sometimes be overzealous in identifying noses, it forced us to come up with an additional solution. Ff we are able to detect eyes (which are often more stable), we filter out any 'noses' that are not spatially below the eyes detected in the image. Of course, if the eye detector is unable to locate any eyes, we lose the benefit of this spatial validation. Then, we pick the largest nose below the eyes as our validated nose. If none exists, then it's declared that no nose is detected.

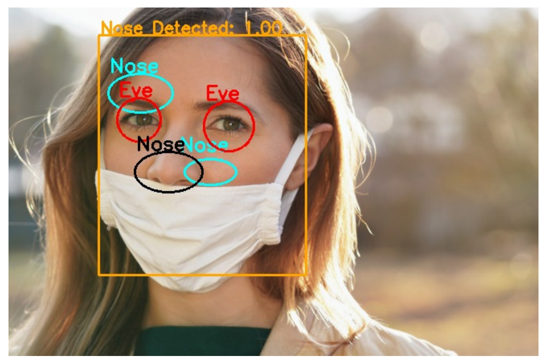


Figure 7: Eliminating false positive noses (blue) using detected eyes (red) and largest selection (black)

- We were able to reduce the number of random false positives for all HAAR features scattered throughout the image (away from actual faces) by first detecting facial regions of interest and then running those through the HAAR detectors.

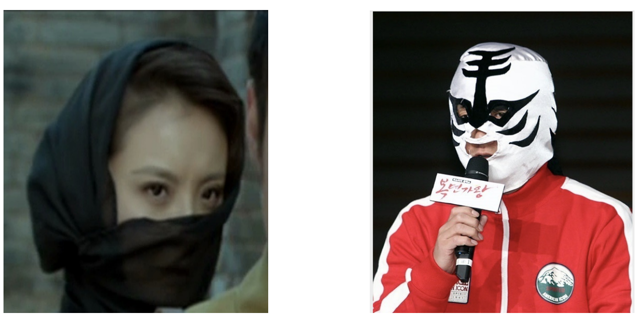
Challenges to address:

- Different kinds of masks look totally different. The following picture from the Kaggle training dataset shows the contrast between a common surgical mask and a medical-used mask. While in our model, almost all of the images are just similar to the one on the right side of the image, and thus may not generalize well to other forms of PPE.

Figure 8: Different types of masks might look absolutely different!

To counteract this bias, we might need a specific dataset which comes from a specialized hospital or clinic, that focuses the left-hand-side type masks.

- One of our dataset sources consists of three categories: No mask, true mask and fake mask. Here ”fake masks” means that the noses and mouths are covered by something else rather than a mask. The following are some samples from this dataset.



(a) Nose and mouth covered by veils

(b) Whole face covered by masks

So far, while some of our models do account for a third class label, others only return two categories with a certain probability, and thus these don’t have the capacity to predict whether a person is wearing a ”fake mask”. As you can see from the picture, some of those ”fake masks” are really deceiving (the black veils look really similar to masks), thus training on those images might lead to a misleading result if they cannot be properly categorized. This is also a field that might be improved.

- Next, training the RCNN was glacial on the full dataset, and thus additional experimentation with downsizing input images before detection without sacrificing performance could have yielded significant improvements in training speed
- Finally, not all forms of data augmentation were easy or even reasonable to implement: because localized bounding boxes for each face were also part of the label for each training image, a shear transformation, a horizontal flip (easier) or other spatial changes would require a corresponding manipulation of the label bounding box locations to avoid corrupting the training process.

5.2. Societal Discussion

Please respond to the following questions. Different projects will have different scales and qualities of impact; we ask you to think creatively and consider the broader implications of your project rather than just the more narrow current technical capability. Responses should together take up roughly one page in your final report.

1. Describe the socio-historical context of your project to identify three broad societal factors that could affect your data, goal, and/or hypothesis. These factors might include current or historical policies, events, social conditions, and larger societal systems. Cite at least one outside source.
   Answer: There are several big events occurred last year and this year that might affect our dataset or relate to awareness regarding mask wearing. The following are the listings:
   - Political rallies and mass protests. A lot of people going to many of these kinds of events do not and did not wear masks to protect themselves and others, which exemplifies the importance of correct mask wearing to minimize the spread of Covid-19 virus. Besides, the six-feet distance policy is also often ignored. As an example, some news commentary downplaying the relationship between BLM and virus spreading (like this one) is perhaps deceptive.
   - Some early policies when the virus spread out in Mar-April 2020. Some policies launched by Trump were really misguided. He urged all facilities and residents to go outdoor without masks and leave masks to medical use in hospital. A lot of people didn’t wear masks at that time, or instead, they replace masks by other shelters like veils, napkins, etc. Such actions correspond to the ”fake masks” in our dataset.
   - The launch of vaccination. Some people believe that they don’t need masks anymore after taking vaccination, but it’s probably wrong. This essay explains the reason why we still need masks after being vaccinated (and thus why this project remains relevant).

Hence as the pandemic continues, and people relax after vaccine administration rates have risen, we might expect more and more no-mask input images or "fake mask" images on our front end.

2. Who are the major stakeholders in this project? What is your relationship to these stakeholders?
Stakeholders are those who may be affected by or have an effect on your project topic. Some examples of stakeholders are a particular demographic group, residents of a particular geographic area, and people experiencing or at risk for a particular problem.
Consider the following questions to help identify stakeholders:

    - Who does this project topic currently affect?
    - Who might be harmed by your research findings?
    - Who might benefit from your research findings?

    Answers: Here, the topic of our project affects the general public. In other words, all people must be conscientious of their mask wearing and all people can use our deployed website here to find out whether themselves have worn masks in the correct way.

    Besides from ourselves, the project model can also been implemented in public places. For instance, the model can be applied, perhaps with ethical repercussions, to surveillance cameras located at the entrance of, say, a supermarket to ensure individuals wear masks properly.

    So far, we have not really identified any groups that might be harmed by our models, at least given how our pipeline is currently deployed.

3. Research or journalism on your broader project topic may have already been conducted. What was the societal impact of existing research? Discuss the implication of this research on your project and consider the following questions to help identify at least one implication. It may affect:

    - How you should frame your goal,
    - How you should design your algorithm,
    - How you should analyze your data,
    - How you should interpret your findings, and
    - How you should present your results.

    Answer: A prominent existing paper in the domain of face mask detection is RetinaFaceMask, which leverages a fairly complex architecture to set the standard for state of the art mask detection on a large public mask dataset. The authors note that they believe their recent efforts to constitute "one of the first dedicated face mask detectors," supporting our conclusion that there is not *much* existing literature on this topic. Thus, this paper could be an imporant benchmark for future mask detection efforts. This model also informed how we designed our algorithm, as well as provided pointers for potential improvement: it relied on a ResNet as a feature extraction backbone, and this inspired us to look into other general detection models like the RCNN that do the same. Additionally, if we had more time to explore implementations, it seems that using a feature pyramid network (FPN) as a neck on top of a ResNet backbone like RetinaFaceMask might improve our accuracy by better capturing "high-level semantic information".

4. How could an individual or particular community's civil rights or civil liberties (such as privacy) be affected by your project?
Answers: On the current stage, information of face landmarks are private information which can be used as payment methods such as Alipay, Apple Pay. Thus when people upload their photos to check whether they wear masks correctly, they are taking risks on the possibilities that their facial information being stolen by hackers. The risk can be reduced by installing a Firewall on our website.

5. If you are using data, what kind of biases might this data contain? Do any of these represent underlying historical or societal biases? How can this bias be mitigated?
Consider the following questions to help you:

    - Were the systems and processes used to collect the data biased against any groups?
    - Is the data being used in a manner agreed to by the individuals who provided the data?

    Answer:

    (a) The WIDER FACE dataset is built up by Multimedia Laboratory, Department of Information Engineering, The Chinese University of Hong Kong, which is also open to public
    (b) The MAFA dataset owned by Rahul Mangalampalli and provided by Kaggle, is also open to public
    (c) COCO dataset is a widely-used large-scale object-detection, segmentation and captioning dataset, including 80 object categories, built up by 13 collaborators, sponsored by CVDF, Microsoft, facebook and Mighty Ai. In addition, the dataset is open to public. Though we did not use it directly, it was used to train the ResNet 50 backbone of the RCNN, for example.

Thus all datasets we use are fairly ethnically diverse (by design) and do not require any API requests. Thus, while there's always room for improvement, it doesn't seem that there's particular bias against any groups or individual parties. However, it is difficult to say for certain that all photos of individuals reside in these datasets with the express permission of all of the subjects.

6. Conclusion

We have implemented a model to help people identify whether they wear masks properly, and people can check on the website here whether the masks ae worn correctly. The topic matters significantly in the current stage of the pandemic as it can help you prevent exposure to virus by engaging in best-practices for mask wearing. We hope that the pandemic will be gone sooner or later following sufficient vaccine distribution, and that, in the meantime, everyone around us remains aware of protecting themselves and others.

References

[1] Andrew G.Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets:efficient convolutional neural networks for mobile vision applications. *arXiv:1704.04861 [cs.CV] 17,Apr 2017*.

[2] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv: 1506.01497 [cs.CV] 6 Jan 2016*.

[3] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Cheng-Yang Fu Scott Reed, and Alexander C.Berg. Ssd: Single shot multibox detector. *arXiv:1512.02325v5 [cs.CV] 29 Dec 2016*.

[4] Xinqi Fan Mingjie Jiang and Hong Yan. Retina face mask: A face mask detector. *arXiv:2005.03950v2 [cs.CV] 8 Jun 2020*.

Appendix

Team contributions

Please describe in one paragraph per team member what each of you contributed to the project.

Samuel Wilkins Researched and trained the RCNN model, explored first dead-end SSD model, implemented bounding box utility class and contributed to the design of the pre-trained pipeline deployed on the app, first draft of eye-nose validation. Edited final report.

Shiqin Yan Explored various methods to create a nose detector; Build the nose and eyes detectors with HAAR feature-based cascade classifiers; Trained the SSD mask detector on our dataset; Developed the final app with Flask and deployed it to cloud with Docker;

Xiaoyan Zhao Final app front-end implementation, backend testing and style adjustment. Contributed to deploy the model and tested the nose and eye detectors. Made scripts and slides for the final presentation.

Zhehui Liang Searching and collecting data from various sources like Kaggle, WIDER FACE dataset and MAFA dataset. Complete the main part writings for the progress report, final report and presentation ppt on Overleaf.