Real Time Mask Detection from Video for COVID-19 Safety

Brian Fogelson, Max Froehlich, John Hutchinson, Scott Smith
Electrical and Computer Engineering
University of Michigan, Ann Arbor
{bfogels, mfroehli, htchnsn, ssscrazy}@umich.edu

EXECUTIVE OVERVIEW

COVID-19 has made a significant impact on our world, and mask usage has been shown to reduce the likelihood of transmission and thus save lives. Ensuring people follow proper mask usage and social distancing policies can help mitigate the spread and reduce the loss of life from this virus. In this paper, we discuss the implementation of a real-time mask detection system for use on video feeds. Our system can detect whether or not an individual is properly wearing a mask. Using transfer learning, we were able to take a pre-trained ResNet-50 model and retrain it to differentiate between masked and unmasked people in images. We developed a simple application which uses this model and can detect whether an individual is wearing a mask or not in real-time from a computer webcam. This application has strong potential for use in public places to help ensure safety guidelines are met.

Keywords

Computer Vision; Classification; Video; Mask Detection; Safety; Real Time

1. BACKGROUND AND IMPACT

COVID-19 has dramatically altered the lives of most Americans, and many individuals worldwide. As disease numbers continue to rise, it is imperative that we follow proper masking and social distancing guidelines. Mask usage has been shown to significantly reduce the likelihood of getting or transmitting COVID-19.

According to the CDC, multi-layer cloth masks block 70-80% of fine droplets and particles when worn properly. Better, masks such as N95 block 95% of particulates [1]. By wearing masks, airborne transmission is cut drastically, especially in public or densely packed areas where social distancing may not be possible. However, masks are most effective when worn properly and continuously. CDC guidelines explain that masks must be put over the nose and under the chin and fit snugly against the sides of the face [2]. This is to ensure that droplets cannot bypass the mask. As a result, it is beneficial to inform and encourage proper mask usage to reduce the risk of transmission and infection.

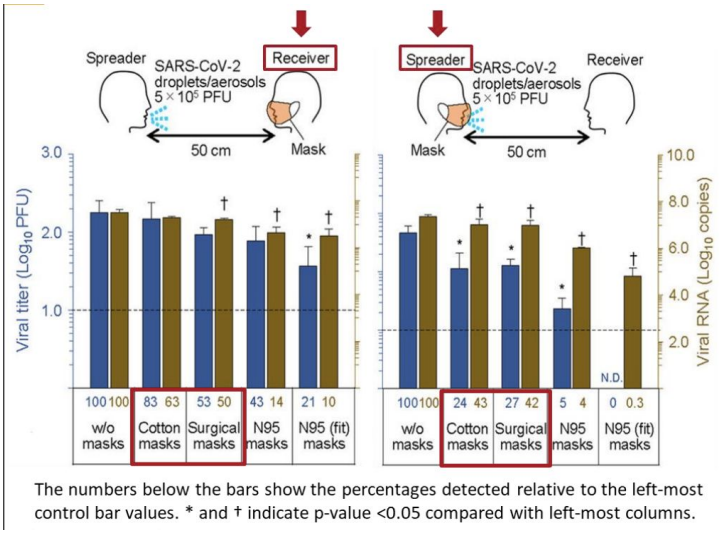

Figure 1. Wearing a mask properly lowers the likelihood of transmission of COVID-19.

We propose an automated detection system which can identify individuals who are not wearing their masks.

2. METHOD

Recognition is performed using a ResNet50 neural network [4]. The network was pre-trained on ImageNet, available through PyTorch, meaning that it was already able to label images into one of 1000 categories with superhuman performance [5]. We utilized transfer learning, starting with this network, and replacing the last layer with our own classification layers to perform classification of masked or not on an image. In general, starting with a pre-trained net leads to much faster training [6].

Our classification layers included a linear layer, a ReLU layer, and a dropout layer before making the final classification with another linear layer. At the beginning of training, the weights of the network were frozen except for the new classification layers and the learning rate started at the Karpathy constant of 3e-4 [7]. Once reasonable training accuracy was achieved of about 85%, the rest of layers were unfrozen, and training continued with a lower learning rate.

Initially, the network was able to quickly converge to 100% accuracy on the training data, but worked poorly in practice. To help remedy this, data augmentation was introduced. Random crops were taken from the training images, then a random affine transformation was applied to these random crops, and color jitter was applied to help make the data more robust to noise, and to prevent the network from overfitting on training data [7].

To see if performance could be increased during test time, the script was updated to find all the faces in an image, and then perform the classification on just the faces. Unfortunately, using the haar cascade from open cv2 to pick out faces in images did not work when someone is wearing a mask. Therefore, this did not help with performance even after trying to tune hyperparameters to get the face detection to work with masks [8].

In the end, the entire video capture was used for detecting masks. This video capture was scaled down to 256x256, and then normalized by the mean and standard deviation from the ImageNet dataset. Additionally, precautions were taken since the ImageNet model included in PyTorch expects RGB inputs, whereas opencv reads in BGR images from the webcam [9].

Once the model was trained, its weights were exported to a file, which were then read in when the real-time video mask detector was run. The real-time detector took each frame of a video, performed pre-processing on it to match the inputs of the neural network, and gave an output score for each category. The label was then created by seeing which output class had the highest score.

In practice, lighting and background played a role in the classification of each frame; however, the mask score almost always increased when a mask was put on, and the non-masked score almost always increased when the mask was taken off. As a result, there is a constant in the code that adjusts the baseline score in practice. A value of .07 was added to the non-mask score and subtracted from the masked score, and this value works well for most backgrounds and lighting environments.

3. PROTOTYPE

In order to evaluate our design, we implemented a simple prototype. Our prototype consists of two sequential parts - the training of the model and the usage of that model on a live video feed from a webcam.

3.1 MODEL TRAINING

The first portion of our prototype involved retraining an existing PyTorch ResNet-50 model to work for our system [3]. The model training algorithm was developed in Python and can be found on our GitHub [12].

In order to train our classifier, we needed thousands of images from three sets: proper

mask, improper mask, and no mask. The proper mask set implies that an individual is wearing a mask that covers their mouth and nose completely. The images for people without masks were taken from the CelebA dataset [10]. This dataset contains thousands of cropped photos of the faces of famous individuals.

For the improper mask and correct mask images, we used the MaskedFace-Net dataset. This data set contains thousands of images of individuals wearing masks properly and wearing masks improperly [11]. We fed these two data sets through our training algorithm and created a few models.

3.2 REAL-TIME VIDEO

The second portion of our prototype involved the development of a Python program that utilizes a computer's webcam. This program analyzes webcam footage and feeds it through the model generated from the previous step. The program grabs individual frames from the video and performs its analysis on the frames. The program can be found on our GitHub [12].

We tried on multiple computers of varying processing power, and were able to achieve real-time processing of the data. The extra frame rate depended on many circumstances, but analysis of the video footage was easily feasible on a modern personal computer.

We started by classifying between all three categories: no mask, improper mask, and correct mask. However, this did not perform well in practice, so we changed this program to only distinguish between no mask and correct mask.

4. RESULTS

Figure 2. Masked and unmasked classification.

We were able to successfully distinguish between mask and no mask, but the distinction between mask and incorrect mask was often inconsistent or inaccurate. To address this, given the limitations of the project logistics, we merged the correct and incorrect mask groupings. The final version of the classifier assigned the same label to images that matched the correct and incorrect groupings. This gave us satisfactory results, even in the inconsistent environment of live video. Beyond some expected wavering back and forth while putting on and removing a mask, the algorithm proved to be relatively accurate and consistent.

Figure 3. An example of an improperly worn mask from our dataset, which proved more difficult than expected to differentiate from correct mask usage [11].

Our algorithm was effective across different kinds of masks that we had available for testing. We tested with cloth masks (Fig. 2), as well as different styles of disposable masks (Fig. 4). This variety also covered different colors of masks, which was a concern due to the lack of diversity of mask color in the training data used.

Figure 4. An example of masked classifications with two different styles and colors of masks.

5. ACKNOWLEDGMENTS

We would like to offer special thanks to Professor Jason Corso, Parker Alexander Koch, and OSCAR IGNACIO DE LIMA for their support this semester.

6. REFERENCES

[1] "Scientific Brief: Community Use of Cloth Masks to Control the Spread of SARS-CoV-2." *Cdc.gov*, 20 Nov. 2020, www.cdc.gov/coronavirus/2019-ncov/more/masking-science-sars-cov2.html.

[2] "How to Safely Wear and Take Off a Cloth Face Covering." *Centers for Disease Control and Prevention*, Centers for Disease Control and Prevention, 28 Nov. 2020, www.cdc.gov/coronavirus/2019-ncov/prevent-getting-sick/how-to-wear-cloth-face-coverings.html.

[3] Pytorch Team. *PyTorch*, pytorch.org/hub/pytorch_vision_resnet/.

[4] He, Kaiming, et al. "Deep Residual Learning for Image Recognition." *ArXiv.org*, 10 Dec. 2015, arxiv.org/abs/1512.03385.

[5] Pytorch Team. *PyTorch*, pytorch.org/hub/pytorch_vision_resnet/.

[6] Li, Fei-Fei, et al. *CS231n Convolutional Neural Networks for Visual Recognition*, Stanford University, cs231n.github.io/transfer-learning/.

[7] Pointer, Ian. "Programming PyTorch for Deep Learning." *O'Reilly Online Learning*, O'Reilly Media, Inc., www.oreilly.com/library/view/programming-pytorch-for/9781492045342/ch04.html.

[8] "Face Detection Using Haar Cascades." *OpenCV*, opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_objdetect/py_face_detection/py_face_detection.html.

[9] "Changing Colorspaces." *OpenCV*, OpenCV, docs.opencv.org/master/db/d64/tutorial_js_colorspaces.html.

[10] Luo, Ping, et al. *Large-Scale CelebFaces Attributes (CelebA) Dataset*, mmlab.ie.cuhk.edu.hk/projects/CelebA.html.

[11] Cabani, Adnane, and Karim Hammoudi. "Cabani/MaskedFace-Net." *GitHub*, github.com/cabani/MaskedFace-Net.

[12] Smith, Scott, et al. "Ssscrazy/Masks-R-Us." *GitHub*, 2020, github.com/ssscrazy/Masks-R-Us.