Mask Wearing Classification

Chanwut Kittivorawong*
Paul G. Allen School of Computer Science and Engineering
University of Washington, Seattle
Seattle, WA 98105
chanwutk@cs.washington.edu

Louis Maliyam
Paul G. Allen School of Computer Science and Engineering
University of Washington, Seattle
Seattle, WA 98105
maliyp@cs.washington.edu

Abstract

As of the beginning of 2021, most restaurants, grocery stores, healthcare, and other public spaces are open. However, people are required to wear a mask to enter those places. In this paper, we create a camera application that determines whether a human face is wearing a mask. The application classifies human faces using a convolutional neural network model. This application can reduce the work of staff and increase higher protections for the sake of common benefits of everyone sharing the spaces.

1 Introduction

In the world of pandemics, it's important to protect each other to slow down the spread of COVID-19. According to Cheng et al. [1], wearing a mask is one of the most efficient ways to not pass on or to not receive COVID-19 from respiratory droplets. Therefore, people are required to wear masks in many indoor facilities. With this enforcement, each facility needs extra resources to monitor this activity. For instance, staffs in a restaurant need to make sure that all customers wear masks. We want to minimize the resources uses in this enforcement. Therefore, we solved this problem by creating a camera application that detects whether a person is wearing a mask or not. The application uses a convolutional neural network model for classification. Then, we created an application that makes use of the neural network model by filming a person and detecting whether the person is wearing a mask or not, frame by frame and in real-time. In the end, we evaluated our model performance, and we found that our model had a 92.16% accuracy when testing with our dataset. With some incorrect predictions, we wanted to find out what causes the model to incorrectly classify some images. After the model was trained, we extracted the output of network layers from our model and compared the outputs from correctly classified images and incorrectly classified images.

2 Datasets

To train our model, we use Face Mask Detection Dataset owned by Gurav [2]. The datasets include 3,725 images of human faces wearing masks and 3,828 images of human faces without masks. Then, we broke them down into two datasets–training dataset and test dataset. See Table 1.

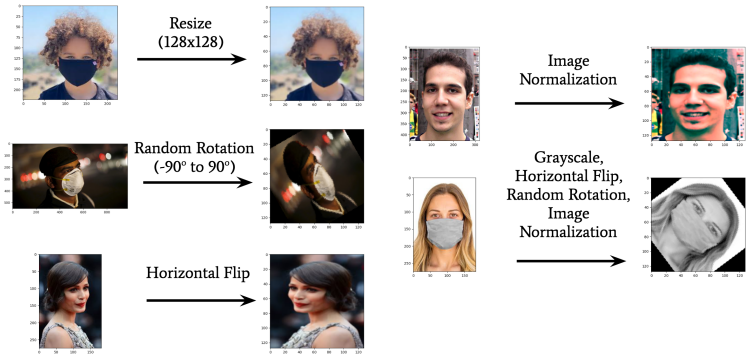
*https://chanwutk.github.io

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

Table 1: The number of images being used to train/test for each class

Classes	Training Dataset	Testing Dataset
with mask	3,354	371
without mask	3,446	382

Figure 1: example of images that were transformed. We always resize the image to 128 pixels x pixels, apply grayscale and image normalization on them. We also perform random rotation and random horizontal flip on the images to generalize the model.



2.1 Data Preparation

To ensure the consistency of input size, we resized every image into a square image with size 128 pixels by 128 pixels. In addition, we prepared the data before training by applying these transformations to our training data (Figure 1): grayscale-filter, random rotation (-90 degrees to 90 degrees), random horizontal flip, and image normalization. We applied the grayscale-filter because different cameras might have different color profiles. The random rotation and random horizontal flip were applied so that our model did not overfit the training dataset. The image normalization was applied so that the mean of each image was at the same point and distributed similarly.

3 Neural Network Model

Since this problem is to classify images, we chose to use a deep neural network to efficiently and accurately solve the problem. Specifically, we were using a convolutional neural network to detect whether people in the images were wearing masks.

3.1 Model Structure

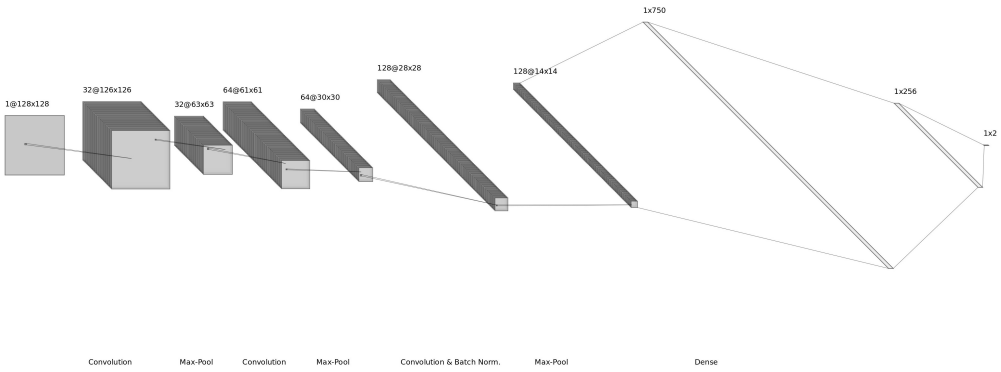
The model receives a batch of grayscale images with the size of 128 pixels by 128 pixels as an input. In order to predict the output, the model makes use of several hidden layers to come up with probability distribution (or predictions).

The model consists of three convolutional layers with a max-pooling layer after each of them. We performed a batch normalization at the last convolutional neural network to make the model learns better. Moreover, we flattened the last max-pooling layer out to input into two fully connected layers which eventually predicted the output of the model. The model can classify images into two classes–with mask and without mask. See Figure 2.

3.2 Loss Function and Optimizer

We used softmax and cross-entropy as our loss functions. We used softmax at the end of our prediction because the softmax function simulated the probability of an input image being one of the two classes. And, we used the cross-entropy function at the end because we would like to amplify the loss value

Figure 2: the architecture of the convolutional neural network, which consists of three convolutional layers with a max-pooling layer after each of them. The output from the last layer then being inputted into the last two fully connected layers which predict the final output.



to penalize the model when it predicted incorrectly. For an optimizer for our training, we used SGD because a model trained with SGD is less likely to overfit the dataset as it chooses data points at random.

4 Results

4.1 Training and Model Performance

We trained the model for 100 epochs using a learning rate of 0.01, a momentum of 0.9, and a weight decay of 0.0005. The final model yields 92.16% accuracy on the test dataset.

4.2 Prediction Examples

Figure 3 shows some examples of images that were being tested and the output that the model predicted. The first row shows the images that are correctly predicted. The second row are images with people wearing masks, but our model falsely predicted that they were not wearing masks. The third row are images of people not wearing masks. However, our model incorrectly predicted that they were wearing masks.

Based on the examples here, we noticed that the images that were correctly predicted tend to have the following characteristics: (1) the human faces are large in the frame, (2) the face part and mask part are easily distinguishable, and (3) there is not much going on in the background.

On the opposite side, images that were incorrectly predicted might contain some of these characteristics: (1) there are variations of color and pattern on the masks, (2) images contain accessories and/or facial hair, which make the model misassociate them with the masks, and (3) the mask is not easily distinguishable from the face.

4.3 Output Layer Visualization

As the model incorrectly predicted some images, we were curious about the cause of these incorrect predictions. So, we extracted and visualized the output of the first ReLU layer to see the pattern of what pixels were activated. In other words, we wanted to know what features our model captured from the images. Starting from a correctly predicted image (Figure 4), we can see that some channels either activate the face part or the mask part. So, the model can distinguish between the two parts. Furthermore, we can see that our model can capture the eyes of the humans in the images. On the other hand, our model cannot capture some of these features in the incorrectly classified images.

Figure 3: examples of images that were used to test the model. The first row are images that the model predicted correctly. The second row and the third row are images that the model incorrectly predicted.

In the first incorrectly classified example (Figure 5), our model could not find the clear boundaries between face and mask. In the second incorrectly classified example (Figure 6), our model could not capture the eyes of the humans in the image. In the third incorrectly classified example (Figure 7), our model activated the parts of the image that were not the human face.

4.4 Camera Application

Our camera application captures video of human faces. It then processes the video in real-time, frame by frame, by classifying if a human in the frame is wearing a mask or not. Our camera application does well when the human face is close to the camera (Figure 8). However, when the human face is far away or when the background of the video has too many objects (Figure 9), our model becomes inaccurate. One future improvement in response to this problem is that we would like to cut out only the human face part of the image before classifying it. With this improvement, the background of the video becomes irrelevant and will improve the accuracy of our application.

Broader Impact

In this paper, we created a camera application that classifies human faces if they are wearing masks or not. This application can make a big impact in the world of pandemics where people are required to wear masks inside indoor facilities. With this application, these facilities can reduce the number of staff to monitor this enforcement and use machines, instead. However, there is a downside to this application, as well. This camera application could raise concerns in privacy as it records human faces. In the real-world use, we would need to inform the visitors of these indoor facilities that our application only uses but not keep the video of anyone. We also explored the problem behind the inaccuracy of our application by visualizing the layer outputs of our model. We found that many incorrect classifications come from the fact that these images have many objects in the background and confuse our model. In future work, we would like to improve our application by adding another neural network that detects faces. Then, we can crop only the face part to be classified using our current model.

Acknowledgments and Disclosure of Funding

We thank all the UW CSE 573 staffs for their helps in this class.

References

[1] Cheng, V., Wong, S., Chuang, V., So, S., Chen, J., Sridhar, S., To, K., Chan, J., Hung, I., Ho, P. & Yuen, K. (2020) *The role of community-wide wearing of face mask for control of coro-*

navirus disease 2019 (COVID-19) epidemic due to SARS-CoV-2 https://www.sciencedirect.com/science/article/abs/pii/S0163445320302358?casa_token=BI_okioGo04AAAAA:FB5PL1IYJ3KPUCDHp83V6BywFXXEnifn5VSOTYZ9koedSRdJLWJ6iBJn69nXXG57GR1mTiyL-OY

[2] Gurav, O. (2020) *Face Mask Detection Dataset*. https://www.kaggle.com/omkargurav/face-mask-dataset/metadata

Appendix

Figure 4: Output tensor of the first ReLU layer of a correctly classified image

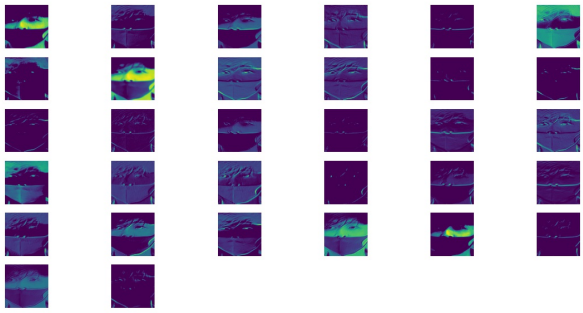


Figure 5: Output tensor of the first ReLU layer of an incorrectly classified image

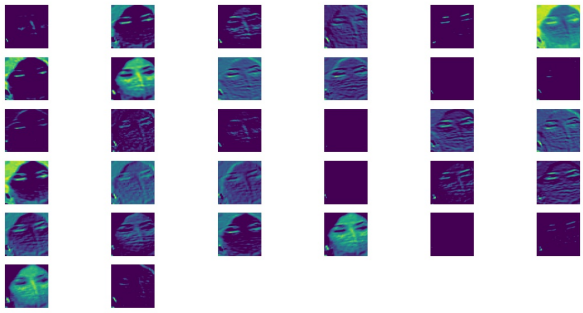


Figure 6: Output tensor of the first ReLU layer of an incorrectly classified image

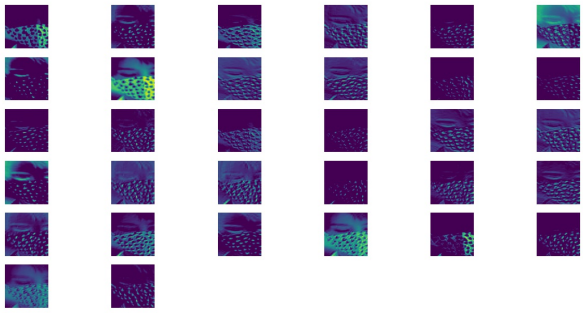


Figure 7: Output tensor of the first ReLU layer of an incorrectly classified image

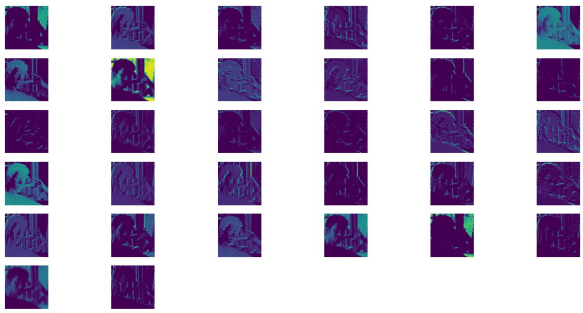


Figure 8: Camera application when correctly classified a face

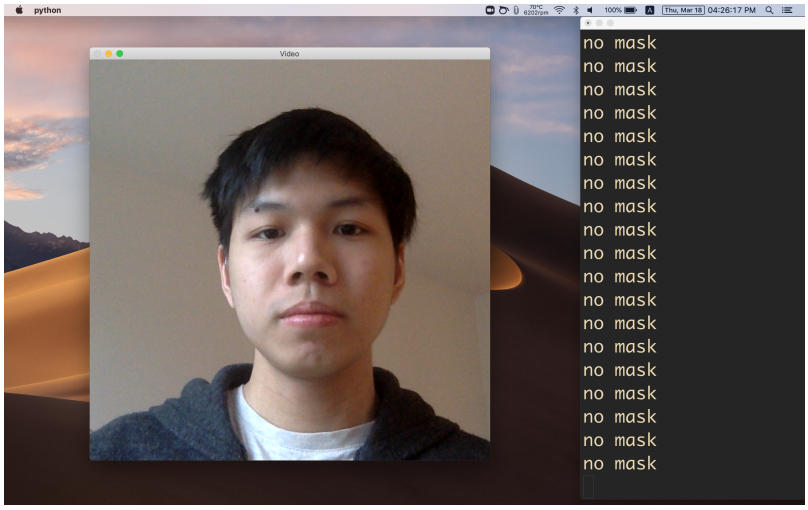


Figure 9: Camera application when incorrectly classified a face