FACIAL MASK DETECTION USING RESNET50

Gaon Choi
Department of Computer Science
Hanyang University
sites.google.com/view/gaonchoi

June 15, 2022

ABSTRACT

Since the outbreak of COVID-19, many countries by law still require wearing facial masks in public indoor places. There has been a need for the development of an automated kiosk that can check whether a person is wearing a mask before entering the building. The goal of the AI model embedded in the device is to classify each image whether a person in the images is wearing a mask or not. Thus, the given problem corresponds to a binary classification problem that classifies the data into two labels: non-masked and masked.

1 Introduction

Due to the outbreak of the COVID-19 virus, it has become mandatory to wear masks in various public facilities. In the recent trend, the government has eased distancing or lifted laws related to wearing masks outdoors, but the need for wearing masks is still high. Various prior studies can be found, such as the correlation between mask wearing and the degree of corona infection, and the development of mask wearing recognition devices. From the above research results and social trends, it can be inferred that wearing a mask will become an essential element in everyday life until the end of COVID-19. This study aims to build an artificial intelligence model that can detect whether a mask is worn by conducting transfer learning based on ResNet50. An accuracy of 95% or more could be obtained, and the processing speed was relatively sufficient to be used in an actual field.

2 Experiments

2.1 Synthetic Data Generation

Many public dataset exists for non-masked faces, but not so many for masked images, at leat not enough for training a deep network. Thus, we need to artificially generate realistic masked face images. We created masked images from non-masked images using MaskTheFace to generate facial-masked images. MaskTheFace is computer vision-based script to mask faces in images. It uses a dlib based face landmarks detector to identify the face tilt and six key features of the face necessary for applying mask. [1]

Figure 1: Synthetic Images generated by MaskTheFace

2.2 Data Augmentation

Data augmentation in data analysis are techniques used to increase the amount of data by adding slightly modified copies of already existing data or newly created synthetic data from existing data. It acts as a regularizer and helps reduce overfitting when training a machine learning model. We calculated mean and std of train set, to apply normalization with transforms.Normalize(). [2]

[1] https://github.com/aqeelanwar/MaskTheFace
[2] We did not use normalization. train mean: [0.5033, 0.4229, 0.3915], train std: [0.2870, 0.2535, 0.2684]

```
dataset_transform = torchvision.transforms.Compose([
        transforms.RandomHorizontalFlip(p=0.5),
        transforms.RandomVerticalFlip(p=0.5),
        transforms.RandomRotation((-135, 135)),
        transforms.RandomResizedCrop(size=(112,112), scale=(0.85, 1.0)),
        transforms.ToTensor(),
        # transforms.Normalize(mean=TRAIN_MEAN, std=TRAIN_STD)
])
```

2.3 Model Architecture

The author of ResNet50 noted the Vanishing Gradient problem, which moves away from optimization as the model gets deeper. By introducing Residual Block, which adds the input value as it is, the problem of forgetting the input was supplemented. ResNet50 has a total of 50 layers, and each layer has a different form of residual block. The figure below is a visual representation of the model structure.

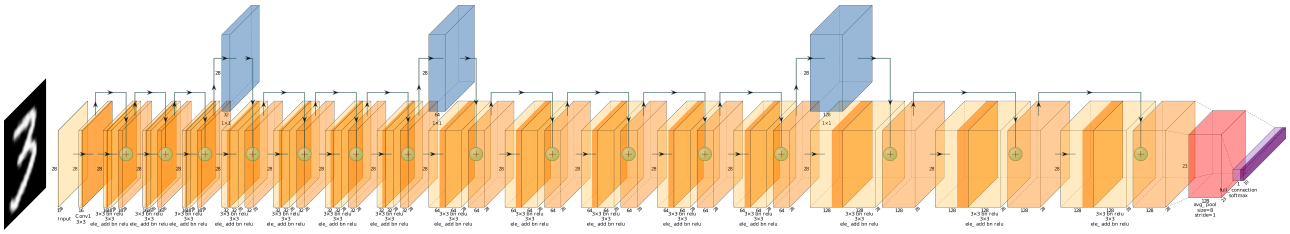


Figure 2: Model Structure of ResNet50

2.4 Model Training

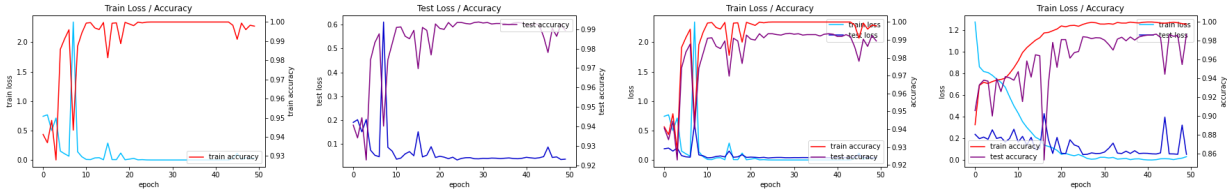
The hyperparameters applied when learning the model are as follows.

(1) input size: $3 \times 112 \times 112$ (RGB image), output format: $\mathbb{R} \in (0, 1)$
(2) learning rate: 0.01
(3) the number of epochs: 50
(4) train data: 15,862, test data: 3,245
(5) batch size: 64
(6) optimizer: SGD with momentum = 0.9, weight decay = $5e^{-4}$
(7) loss function: torch.nn.BCEWithLogitsLoss

$$l_n = -w_n[y_n \cdot \log \sigma(x_n) + (1 - y_n) \cdot \log(1 - \sigma(x_n))] \tag{1}$$

(8) modification of model structure: ResNet50.fc = Linear(2048, 1)

3 Results



(a) Train Loss, Accuracy (b) Test Loss, Accuracy (c) Overall Loss, Accuracy (d) Without Momentum

Figure 3: Model Train/Test Performance Plot. horizontal axis: epoch, vertical axis: accuracy(percentage), loss

4 Conclusions

In this project, transfer learning was conducted with ResNet50, and a model was constructed to detect whether a mask is worn in a given image. Based on the test set, the accuracy was more than 99%, and it was confirmed that the model properly classified most pictures even when the actual image was uploaded and tested.

Metrics-1	Loss	Metrics-2	Accuracy(%)
Train Loss Validation Loss	0.02047936655046101 0.03538672445137081	Train Accuracy Validation Accuracy	99.82978186861682 99.24242424242424
Metrics-3	Time Cost		
Elapsed Time in Training	1 minute/epoch (approximation)		

Table 1: Model Performance

5 Discussions

5.1 Regularization technique - Early Stopping

In machine learning, early stopping is a form of regularization used to avoid overfitting when training a learner with an iterative method, such as gradient descent. [3]

This technique was used to train our model. Compared to the test loss value in the previous epoch for each iteration, the model parameter was stored only when the test value was smaller than before. Therefore, if the overfitting occurs and the test loss is larger than before, the result is not reflected, indirectly applying early stopping technique.

5.2 Momentum

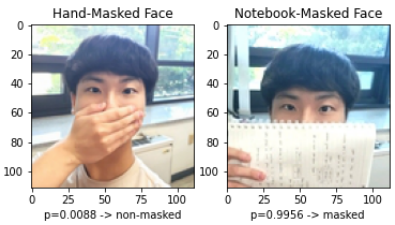
Gradient descent can oscillate when it enters a narrow valley with large learning rate. We can enforce a bit smoother trajectory by adding an additional term to gradient descent called "momentum". It prevents oscillative movement in a valley. The presence of momentum in the learning process had a great influence on the degree of improvement in the local minima. It can be seen from Figures 3-(c) and (d) that there is a large difference in terms of loss and accuracy.

5.3 Limitations of the current model

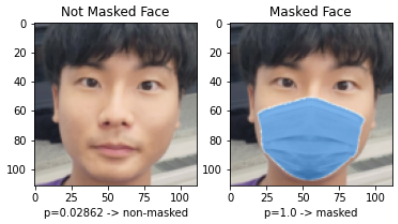
The ResNet50 determines the image based on the texture shown in the image. The characteristics of the model become a limitation when measuring whether or not a mask is actually worn. This experiment was done with the pretrained ResNet-50 architecture to examine its limitations. Both of the pictures below are non-masked. The image covered with hands was judged to be non-masked because the textures of the face and hands were similar, but when the face was covered with a notebook rather than a hand, it was judged to be masked because the textures were clearly different. In addition, when the model was trained without any pretrained parameters, the model only detected the correct mask types which is in the MaskTheFace dataset. To compensate for these drawbacks, it is necessary to establish a model capable of separately recognizing objects for masks in future studies.

6 Qualitative Evaluation

In order to flexibly determine the performance of the model, the author's mask-wearing photo and mask-free photo were arbitrarily converted to 112 $\times$ 112 size and entered. The trained model returns one real number between 0 and 1 for each image, since the last layer consists of a sigmoid. With 0.5 as the boundary value, the closer to 0, the less the mask was worn, and the closer to 1, the more the mask was worn. The model returned a value of 0.02862 for the first picture(non-masked) and 1.0 for the second picture(masked). Thus, the trained model properly classifies whether or not to wear a mask in the input picture.



(a) Limitations of ResNet50-based Model



(b) Model Qualification with real images

[3] https://en.wikipedia.org/wiki/Early_stopping