COMP9444 Neural Networks and Deep Learning

Group Project – Face Mask Recognition
Neural Nerds Project Summary

Term 2, 2022

z5196145 Abhyudit Gupta
z5268980 Harshwardhan Kothari
z5238561 Arpit Singh Rulania
z5260447 Zane Vom
z5137192 Prompong Yutasane

I. Introduction

The aim of the paper is to provide a brief and concise summary of an application of AI in the task of face mask detection. The problem we are trying to solve is the determination of a person wearing a face mask in a given photo or even a live video stream. This is important because since the emergence of the COVID-19 pandemic in early 2020, all sorts of measures have been put into place by governments across the world to reduce and mitigate the spread of infection. One method that has shown to be extremely effective in this regard, especially when social distancing is virtually impossible, and that is the mandating of face masks. This allowed people to lower the risk of transmission and therefore the rate of hospitalization as well. As such, it is of the upmost importance that people wear them in times of need, even when COVID-19 has a smaller presence in the community, lest another outbreak ensues. This technique will ensure the appropriate agencies can monitor the situation and apply changes or penalties where necessary. Our contribution towards improving current practice using the proposed solution is while there are numerous code bases trained for the purposes of face mask detection, there are very few that distinguish the type of mask the person is wearing.

II. Methods

We chose a 0.0001 learning rate because it speeds up the accuracy in a lower epoch, and we need 20 epochs because that is what we must come to as the ideal value after multiple trials. Our program only does two classifications, hence the input on the last layer is two. To load the images, we use a load image function from the Keras preprocessing library to load an image path and transform the image to a 224 x 224 size, so that our model will be perfect in that shape. To split the testing and training data, we have a function named train_t_test_split(), which has a test size of 20% of the dataset for testing, and the other 80% for training.

Here we use MobileNetsV2 instead of the traditional CNN layer because it is faster in processing, and it also uses less parameters. However, because of these advantages, it's one major disadvantage is that it is less accurate. By using MobileNetV2 we are generating two types of models: the head and base models, and with regards to the base model, there are some pretrained models specifically for images by having the weights parameter equal to "imagenet" which initializes the weights for us, which in turn gives us better results. The next parameter, include_top was made false, where include_top is a Boolean value which is to say whether the fully connected layer is on the top of the network, and the third and last parameter is input_tensor, which is nothing but the shape of the image going through. The shape includes three channels in the image, which means it takes in colour images in the form of RGB, and the first two numbers indicates the target size of the input image. In terms of the head model, we created a head model object in which we're passing the base model output as the first parameter, and then we created the pooling, which is a size of 6 x 6. Next, we flattened these layers and added a dense layer with 128 neurons. Our activation layer here is "relu", which is basically the go-to activation function for non-linear use cases, and we use Dropout to prevent overfitting of our models. The output is 5 layers, for all types of 4 masks and no mask. We're giving the activation function of "softmax" because it is based on giving 1s and 0s values, which suits our binary classification purposes quite well. Once all that has gone through, the model function is called in which it accepts two parameters, one is inputs and the other is outputs. The inputs will be the base model and the outputs will be the head model. Initially, we need to freeze the layers in the base model so that they won't be updated during the first training process because they are just a replacement for the Convolutional Neural Networks.

III. Experimental Setup

We acquired the dataset from an open-source Machine Learning website called Kaggle for this project and the URL for the source is as follows:

https://www.kaggle.com/datasets/bahadoreizadkhah/face-mask-types-dataset

The refined dataset we ended up going along with is summarized in Table 1 below, and through our research, we found out that 100 images for each category was the bare minimum. Thusly, we were able to overcome this issue by applying data augmentation, specifically by way of applying a random rotation to some photos, to simulate the possible angles at which the person's head may be at. One thing to note is that 17% of the total 5000 images in the dataset was allocated for testing purposes. We've also broken down the Jupyter Notebook files into two files: 01-Mask-Detection-Train-Model.ipynb and 02-Live-Video-Mask-Detection.ipynb. The former trains the model with the dataset and the latter is a program that is a live video reading from the model after it's trained to detect whether the mask is cloth or surgical.

Table 1: Summarised Dataset

Mask Type	**Number of Images**
Cloth Mask	1000
Surgical Mask	1000
N95 Mask	1000
N95 Valve Mask	1000
No Mask	1000
Total	5000

IV. Results

The main findings from the evaluation are that for the first epoch, the loss was high and accuracy was expectedly low, but then as the epochs went on, the loss was decreasing and the accuracy only kept increasing, as can be seen in Figure 1 in the appendix.
After the training process was completed, the resulting table showed that the accuracy ended up being at around 90%, which is to be expected given the lower accuracy of MobileNetV2, but still accurate enough for our purposes. Refer to Figure 2 in the appendix.
The plot created from running the code was also able to better clearly show these results. Refer to Figure 3 in the appendix.

V. Conclusions

The key strengths of the proposed solution are that the program can distinguish between a cloth and surgical mask on the video stream. If we had more time for the project, we could improve this by adding more types of masks to the dataset.

VI. Appendix

Figure 1: Epochs and their resulting accuracy

```
Epoch 1/20
129/129 [==============================] - 52s 377ms/step - loss: 0.5248 - accuracy: 0.3692 - val_loss: 0.3574 - val_accuracy: 0.6580
Epoch 2/20
129/129 [==============================] - 49s 381ms/step - loss: 0.3724 - accuracy: 0.5934 - val_loss: 0.2892 - val_accuracy: 0.7203
Epoch 3/20
129/129 [==============================] - 56s 438ms/step - loss: 0.3230 - accuracy: 0.6524 - val_loss: 0.2515 - val_accuracy: 0.7709
Epoch 4/20
129/129 [==============================] - 53s 407ms/step - loss: 0.2965 - accuracy: 0.7022 - val_loss: 0.2332 - val_accuracy: 0.7826
Epoch 5/20
129/129 [==============================] - 54s 418ms/step - loss: 0.2731 - accuracy: 0.7284 - val_loss: 0.2186 - val_accuracy: 0.7932
Epoch 6/20
129/129 [==============================] - 53s 410ms/step - loss: 0.2551 - accuracy: 0.7462 - val_loss: 0.2079 - val_accuracy: 0.7991
Epoch 7/20
129/129 [==============================] - 51s 395ms/step - loss: 0.2444 - accuracy: 0.7600 - val_loss: 0.1981 - val_accuracy: 0.8132
Epoch 8/20
129/129 [==============================] - 48s 372ms/step - loss: 0.2347 - accuracy: 0.7683 - val_loss: 0.1915 - val_accuracy: 0.8179
Epoch 9/20
129/129 [==============================] - 47s 365ms/step - loss: 0.2188 - accuracy: 0.7914 - val_loss: 0.1843 - val_accuracy: 0.8190
Epoch 10/20
129/129 [==============================] - 47s 361ms/step - loss: 0.2159 - accuracy: 0.7858 - val_loss: 0.1810 - val_accuracy: 0.8273
Epoch 11/20
129/129 [==============================] - 50s 390ms/step - loss: 0.2093 - accuracy: 0.7972 - val_loss: 0.1782 - val_accuracy: 0.8331
...
    accuracy                           0.86       851
   macro avg       0.86      0.86      0.86       851
weighted avg       0.86      0.86      0.86       851
```

Figure 2: Overall evaluation

```
[92m[COMPLETE] Training head of network
[93m[IN PROGRESS] Evaluating network and making predictions on testing set
27/27 [==============================] - 5s 171ms/step
[92m[COMPLETE] Evaluating network and making predictions on testing set

              precision    recall  f1-score   support

         N95       0.81      0.72      0.76       170
   N95_valve       0.78      0.89      0.83       171
       cloth       0.88      0.80      0.84       170
     no_mask       0.98      1.00      0.99       170
    surgical       0.85      0.87      0.86       170

    accuracy                           0.86       851
   macro avg       0.86      0.86      0.86       851
weighted avg       0.86      0.86      0.86       851
```

Figure 3: Training loss and accuracy plot