Project Assignment, Part I

AI Face Mask Detector

Submitted to:

Dr. René Witte

Due date:
Wednesday, June 8th

By

Team FS-03
Team Members :
Sherif Ghaffar (40058600 - GhaffarSherif - Data Specialist)
Arshia Hamidi (40068250 - ashthedaddy - Training Specialist)
Joseph Goldberger (21958283 - j-gold77 - Evaluation Specialist)
Ali Turkman (40111059 - yellowsub12 - Compliance Specialist)

Github Repo
https://github.com/yellowsub12/Artifical-Intelligence-Project

Submitted on 08/06/2022

Gina Cody School of Engineering and Computer Science

Concordia University

We certify that this submission is our original work and meets the faculty's expectations of originality.

Table of Contents

Dataset

For our dataset, we initially had a total of 4,969 images for our CNN to process, but our program managed to only detect 4,316 of them, which was a minor inconvenience but one that could be ignored seeing as we only needed a minimum of 1.2k training images and 200 testing images. We collected our images from a variety of sources, including Kaggle and Google Images, with the Kaggle datasets being cited below in the references. Among our 4,316 images, we decided to split them 75-25 for training-testing, seeing as training required significantly more images, and this resulted in 3237 images for training and 1079 for testing, as can be seen below.

```
#grabbing dataset
dataset = torchvision.datasets.ImageFolder('/Users/jgold/Documents/GitHub/Artifical-Intelligence-Project/Dataset',
                                           transform=transformation)

#randomly splitting for training and testing
training_set, testing_set = random_split(dataset, [3237, 1079])
```

This was done as per the TA's instruction to have all the images combined into a single dataset, and with the splitting being done by the program itself. Fortunately, we were able to gather only real images, none were fake or altered, and we think we had enough that we didn't need augmentation. The only issue was that the images were not spread evenly between our classes. For this project, we had 4 classes/categories of images, Cloth, N95, Surgical and NoMask. We found a Kaggle dataset with a large number of N95 masks and decided to use all of it, which significantly increased the size of the N95 mask data set compared to other datasets. The disparity was pretty pronounced, with 686 Cloth images, 1724 N95 images, 1274 Surgical images and 1012 NoMask images. Also, efforts were made to make sure that no duplicates were in the dataset, seeing as duplicates would bias the results if they were present in both training and testing sets. Finally, we also used transformation to resize all our images in 32 by 32 pixel formats inside the program, which is needed for the CNN up next.

CNN Architecture

Our CNN's architecture was inspired by the lab #07. The first part of our CNN has 4 convolutional layers, all of which are followed by batch normalisation, a leaky ReLU activation function and a 2D max pooling layer. This first part outputs a vector, which is then processed by the second part of our CNN, which consists of 3 fully connected layers with ReLU activations. This second part of the network also includes dropout layers to reduce overfitting. It ends with a softmax activation function. The output of this second layer should be a number representing how likely the CNN thinks each class is for the input image given to it. As the TA instructed, we also saved our trained model separately. We couldn't figure out how to label a confusion matrix with Scorch, so we just used PyTorch to evaluate data then used seaborn to build the confusion matrices with labels. As for the Convolutional Neural Network (CNN), ours is based off the one that was implemented in lab #07, with much of the same functions and techniques, but instead of the CIFAR-10 dataset, we used our own dataset with different categories. The dataset is initially loaded with a DataLoader PyTorch object.

Our training parameters are:

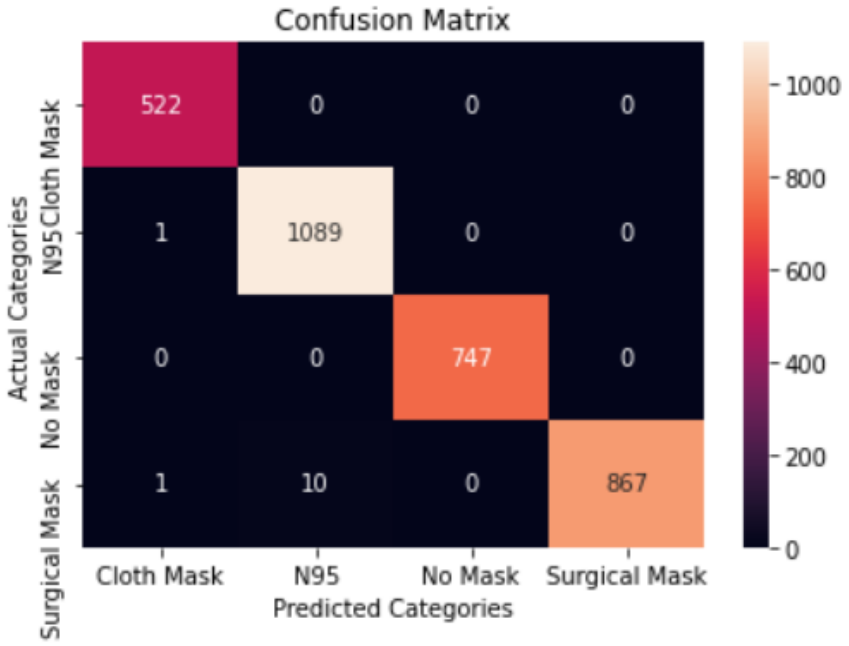
- 30 epochs
- Batch size of 100
- Adam as the optimization algorithm
    - Learning rate of 0.001
    - Momentum of 0.9

The reason why we picked 30 epochs is because we found out that the model would reach an accuracy of 97-100% around 30 epochs and a bit before. However, this accuracy is suspiciously high, which could mean that our neural network is overfitting and we might lower the number of epochs for Phase II of the project.

Evaluation

Using the *convolutional neural network (CNN),* after training the model on the training set, our model's training accuracy was measured by splitting the 1079 images into 11 batches. During our 30 epochs on the training data, by epoch 19 we started seeing some epochs with 100% accuracy. This accuracy would fluctuate between 95-100% on the remaining 11 epochs.

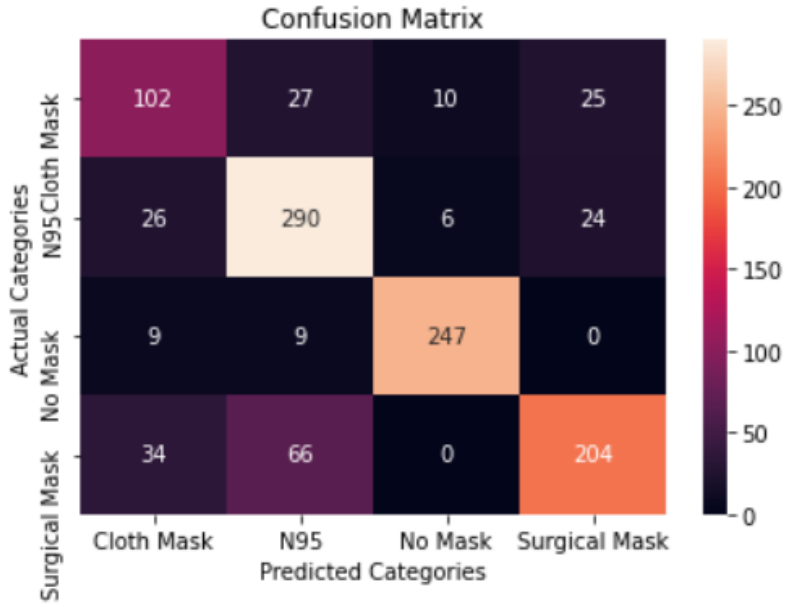
```
Time to evaluate our data.
First will be the training data.
```



	precision	recall	f1-score	support
0	1.00	1.00	1.00	524
1	1.00	0.99	0.99	1099
2	1.00	1.00	1.00	747
3	0.99	1.00	0.99	867
accuracy			1.00	3237
macro avg	1.00	1.00	1.00	3237
weighted avg	1.00	1.00	1.00	3237

After saving the model, we were ready to calculate the accuracy, recall, f1 score and plot the confusion matrix on our testing data.

```
Now to evaluate the testing data.
```



	precision	recall	f1-score	support
0	0.62	0.60	0.61	171
1	0.84	0.74	0.79	392
2	0.93	0.94	0.94	263
3	0.67	0.81	0.73	253
accuracy			0.78	1079
macro avg	0.77	0.77	0.77	1079
weighted avg	0.79	0.78	0.78	1079

Note*: We couldn't figure out how to anchor the y axis labels so they are overlapping a little

Our NN performance was great on our training data. Our best precision, recall and accuracy using f1 score were in the no mask category. Our neural network was 94% precise when labelling images with no mask. In this category it got confused most with cloth masks. It also

had 90% recall and 92% f1 score in this category, so it performed above average. N95 and surgical masks were next in the rankings, scoring 80% and 77% respectively in precision. The NN on N95s got confused with both surgical and cloth masks equally whereas the surgical masks were believed to be N95s mostly. Lastly we had cloth masks.

On the test dataset, the NN performance metrics were not quite as good. The overall accuracy was around 78%. Our precision, recall and f1-score were the best for the no mask class. The class that had the worst metrics was the cloth mask class.

We believe there is definitely room for improvement, especially on the cloth masks. The first thing we must do is balance our datasets better. All 4 of our classes should be representing a closer to equal amount of images. The cloth masks class has the lowest amount of images, which showed in the performance metrics for that class. Second, we believe that we overtrained the model, because by the 19th epoch on the training set we were getting 100% accuracy scores, but we continued for 11 more epochs, 33% of the total amount. This overtraining may have led to overfitting and thus our NN had trouble actually distinguishing differences on the testing data.

References

Datasets :

https://www.kaggle.com/datasets/coffee124/facemaskn95

https://www.kaggle.com/prithwirajmitra/covid-face-mask-detection-dataset

https://www.kaggle.com/datasets/ashishjangra27/face-mask-12k-images-dataset

https://www.kaggle.com/datasets/dhruvmak/face-mask-detection

Lab #07 :

https://moodle.concordia.ca/moodle/pluginfile.php/5430653/mod_resource/content/5/lab07-q.pdf