Annaliese Muth

Professor Daniel O'Connor

MATH 373

5/17/2021

Facial Key Point Detection Report

For our final project, we were given a dataset consisting of images of people's faces, along with key points—such as their eyes, nose, etc. In this project, we were tasked with writing a convolutional neural network, and then a pretrained convolutional neural network, using any architecture of our choosing.

In my GitHub python file, I have created 5 different models, each with the architecture listed below:

1) A simple neural network model with 2 layers, without using any convolutions.

2) A convolutional neural network with 8 layers, using this architecture for each layer:

```
model.add(Conv2D(36, (5, 5)))
model.add(Activation('relu'))
model.add(MaxPooling2D(pool_size=(2, 2), strides=(2, 2)))
model.add(Dropout(0.2))
```

3) A convolutional neural network with 8 layers, using a pretrained network via ResNet, using the same architecture as model 2.

4) A convolutional neural network with 8 layers, using data augmentation—specifically increasing the brightness of each image, using the same architecture as model 2.

5) A convolutional neural network with 8 layers, using data augmentation (increasing the brightness of each image) and using a pretrained model via Resnet, using the same architecture as model 2.

Since the first model—the simple model—was not part of the assignment, but rather for my personal understanding and progress for the project, I will focus on only models 2 through 5.

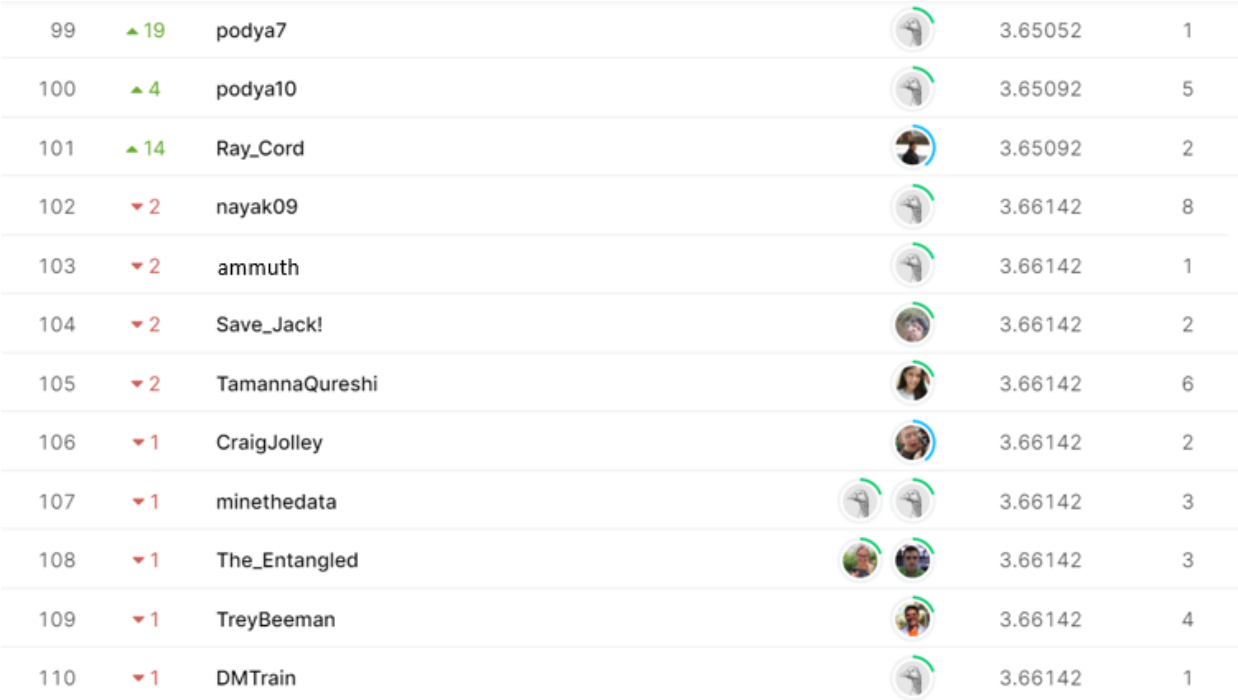
After I got the simple neural network to work, I tried 4 to 8 layers, but I decided to go with 8 layers because it yielded higher accuracy results. However, I did not decide to use more

than 8 layers because my models took so long to fit anyways. In addition, I used 200 epochs for the second model, and then I used 50 epochs for the rest of the models—due to the fact that it took hours to fit the later models. However, I do acknowledge that increasing the number of epochs would increase the model's accuracy.

Overall, I would argue that of the models I included, model 5—using a convolutional neural network with data augmentation and ResNet. My reasoning to this is because our model's accuracy reached a higher percentage than the rest of the models. I did try using other data augmentation originally—such as cropping or flipping each image, but I found that increasing the brightness yielded better results.

The learning rate I used for my models was .01.

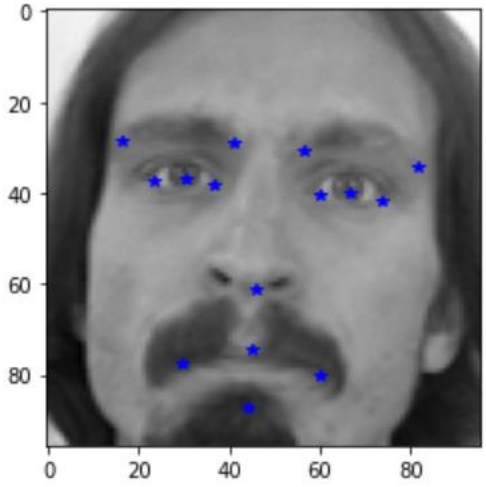
My Kaggle score was 3.66142.



99	▲ 19	podya7		3.65052	1
100	▲ 4	podya10		3.65092	5
101	▲ 14	Ray_Cord		3.65092	2
102	▼ 2	nayak09		3.66142	8
103	▼ 2	ammuth		3.66142	1
104	▼ 2	Save_Jack!		3.66142	2
105	▼ 2	TamannaQureshi		3.66142	6
106	▼ 1	CraigJolley		3.66142	2
107	▼ 1	minethedata		3.66142	3
108	▼ 1	The_Entangled		3.66142	3
109	▼ 1	TreyBeeman		3.66142	4
110	▼ 1	DMTrain		3.66142	1

For optimization algorithms, tried both Adam and SGD, and ended up using SGD because it yielded better results—slightly.

In my predictions, most of the images overlayed with the key points were accurate. Some could've been more accurate, such as the first image below.



Below, are some other images with the prediction key points—we notice that those which aren't very accurate are those whose faces are turned from the camera slightly, instead of facing the camera head on. While we do have this issue, overall the key points are pretty accurate. For example, image 2x5 and image

I have included the GitHub link to my code here: Final Project MATH373.