Journal Report 3

9/16/19-9/20/19
Addison Phelps
Computer Systems Research Lab
Period 2, White

Daily Log

Monday September 9

I worked on being able to train the ResNet50 network and editing the structure to improve accuracy and training times. I created the image data generators and tried training for a couple epochs to ensure everything was still working. After ensuring that everything was working, I trained the model for 30 epochs and achieved about 90 percent accuracy on the training data (not testing).

Tuesday September 10

Worked on the splitting the training and testing data to ensure overfitting was not happening in future training sessions. Also worked on making the prediction function to predict the output of specified images by the user. And added a line at the ending to save the model weights after training. I also cleaned up (removed) some of the images with barely recognizable logos.

Thursday September 12

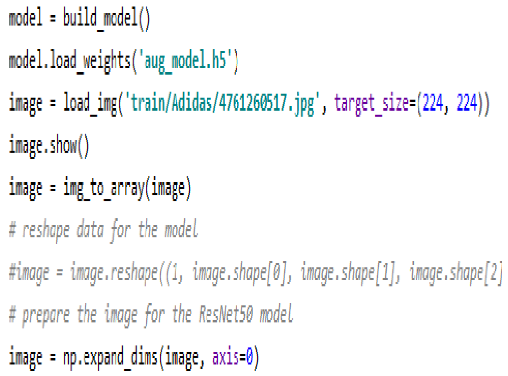
Used the parse predictions function created last week to work on outputting the percent confidence the model had in its prediction, the logo name with the highest confidence, and the name of the logo. Originally, the output prediction would have an array of the probabilities that a specified image was one of the logos for every given logo type (which there are 27).

Timeline

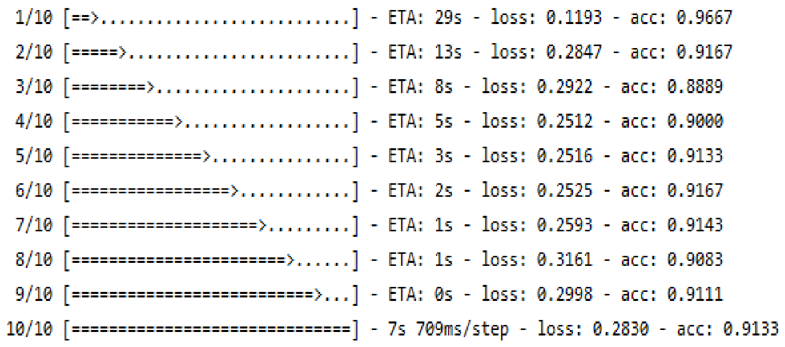
Date	Goal	Met
9/3	organize data/processing	yes - organized the data into folders/augmentation
9/9	create CNN/start training	yes - built the network and ready to train
9/16	reach acceptable accuracy (over 95 percent) on original training data	no - not yet, got to 90 percent (w/o validation set however)
9/23	reach acceptable accuracy (over 95 percent) on original training data with a validation set	no - not yet
9/30	reach acceptable accuracy (over 95 percent) on data with augmentations	no - not yet

Reflection

While I didn't achieve exactly what I wanted this week, I still think I made good progress and it should be easy to achieve my next goals. At this point, I'm about done with setting up the training and testing with the model, and all that's left is training. However, I did notice that training took very long, so I will have to probably use one of the gpus in the syslab, and familiarize myself with connecting to them and using them. I don't expect many challenges to come up during training besides slight optimizations, but if training works well for the un-augmented data, training for the augmented data should be a breeze (but I'll probably have to train for a longer period of time). I think if I can optimize how I train testing different classification models for improved accuracy (like VGG or Inception) would be useful as well.



```
model = build_model()
model.load_weights('aug_model.h5')
image = load_img('train/Adidas/4761260517.jpg', target_size=(224, 224))
image.show()
image = img_to_array(image)
# reshape data for the model
#image = image.reshape((1, image.shape[0], image.shape[1], image.shape[2]
# prepare the image for the ResNet50 model
image = np.expand_dims(image, axis=0)
```



```
 1/10 [==>...........................] - ETA: 29s - loss: 0.1193 - acc: 0.9667
 2/10 [=====>........................] - ETA: 13s - loss: 0.2847 - acc: 0.9167
 3/10 [========>.....................] - ETA: 8s - loss: 0.2922 - acc: 0.8889
 4/10 [===========>..................] - ETA: 5s - loss: 0.2512 - acc: 0.9000
 5/10 [==============>...............] - ETA: 3s - loss: 0.2516 - acc: 0.9133
 6/10 [=================>............] - ETA: 2s - loss: 0.2525 - acc: 0.9167
 7/10 [====================>.........] - ETA: 1s - loss: 0.2593 - acc: 0.9143
 8/10 [=======================>......] - ETA: 1s - loss: 0.3161 - acc: 0.9083
 9/10 [==========================>...] - ETA: 0s - loss: 0.2998 - acc: 0.9111
10/10 [==============================] - 7s 709ms/step - loss: 0.2830 - acc: 0.9133
```