Sean Yang

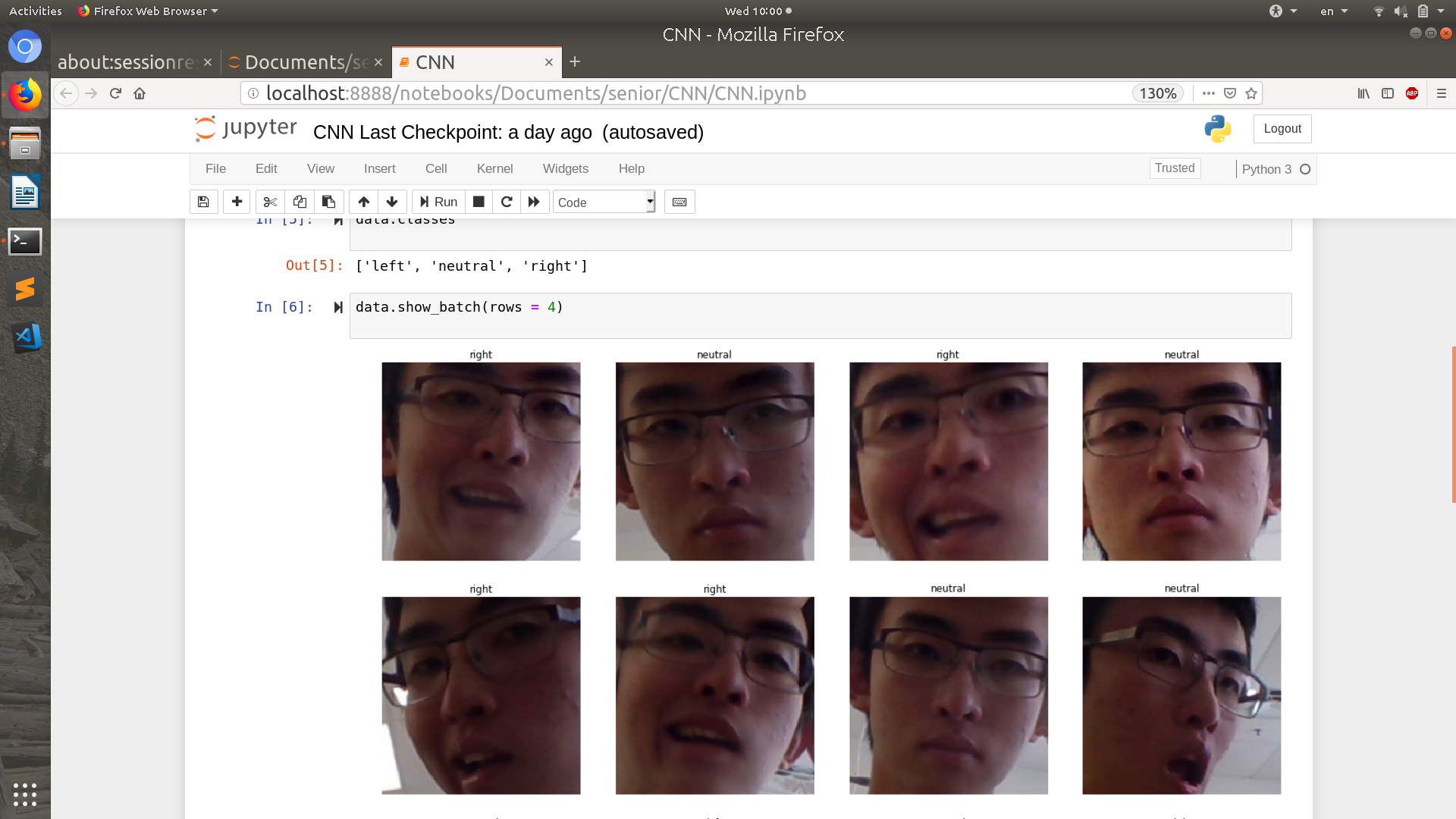
Pd 4

5/1/19

Journal report 26

I continued to explore the CNN approach this week. I successfully read in the data using the fastai library. It’s quite simple and convenient. I tried to follow the instructions from an example code for pet classification from the class. However, the API documentation seems to change quite often. I had to search up things and look at the documentation to fix the errors. For example, a class that was called ConvLearner is called create\_cnn now.

After loading the data, I performed data augmentation to generate some extra data for training. There are multiple options for transforms such as warp, change lighting, rotate, and flip. I chose to disable flip and warp because they would distort the image by a large margin and it would make the detection inaccurate. I also limited rotation to 15 degrees just to be safe.



My current approach is to do transfer learning by fine-tuning resnet. The library has a fit\_one\_cycle function which is based on a paper published in 2018. I’m not sure exactly how many cycles to fit. I need to read the paper and look at people’s discussions about it online.

Currently, the accuracy is ok, hovering around 60 for the validation set. The training loss was going down for both the training and the validation set. It’s because I don’t have much training data. When I recorded the gestures, around 30 image for each gesture was plenty. However, I recorded over 220 images for CNN. This data is probably too little for it. However, recording more would be annoying. This also proves why using KNN for feature point would make sense over CNN. CNN is not great with a small dataset and it would be hard to tell users the exact amount of data needed to get a good model. I also noticed that my validation loss is always lower than my training loss, which is quite unusual. Right now I’m training on my cpu. I will try to save time by training on duke or google colab with a GPU next time. I will also try to play around with different learning rate and stochastic optimization algorithms.

