Journal Report 11

11/25/19-12/6/19
Kevin Fu
Computer Systems Research Lab
Period 1, White

Daily Log

Tuesday November 26

Reorganized GitHub repo to match format specified by Mr. White. Learned how to scp folders and ssh into TJ CSL. Also learned how to run jupyter notebooks remotely that I can open locally.

Monday December 2

Started writing CNN preprocessing code in jupyter notebook, found that Keras' flowFromDirectory method takes a different input file structure than what I had.

Tuesday December 3

Wrote script to reformat input folder for flowFromDirectory, putting images into subfolders by class. Experimented with top, center, and bottom cropping of images, decided on width-only cropping. Also restricted dataset to high-angle imgs only. Scp'd dataset into snowy and loaded it into Keras, found that Keras can't split dataset into train/validation sets without augmenting both.

Thursday December 5

Wrote script to split directory into train/validation sets randomly given a split % (chose 20%). Copied new train/valid dirs over. Found blog post that detailed transfer learning with Keras' ResNet50, copy-pasted code and tweaked hyperparams slightly. Went from 65% to 85% validation accuracy.

Timeline

Date	Goal	Met
Nov 18	Finish labelling squares, count square types, add more square labels if needed	Done, deciding to not add squares for now
Nov 25	Implement image augmentations, reorganize GitHub repo	Done
Dec 2	Modify ResNet architecture for data then begin training	Done
Dec 9	Either split CNN into three pieces (detailed in Journal 10) or improve piece-recognition CNN to at least 75% accuracy	Done (at 85% accuracy, no tricks needed)
Dec 16	Write prediction-only script, combine with board-segmentation script to fulfill winter goal	Not started
Winter Goal (Dec 19)	Have a script that converts a chessboard image to a digital array using a piece-recognition CNN (75% accuracy)	Piece-recognition CNN done

Reflection

I spent a significant amount of time working outside of class this week. The Keras ImageDataGenerator for data augmentation was less plug-and-play than I'd hoped, so I was afraid I wouldn't meet my Winter Goal. I had to crop the images in my dataset, then split it into training/validation sets outside of Keras before I could load them into Keras. I chose to crop the width only, as the unique features of chess pieces are generally at the top and bottom, and I found that center, top, or bottom cropping the height removed this information. I also chose to restrict the data to the high_angle photos only, because I knew introducing the low_angle photos would make the dataset unnecessarily challenging. (Many of the photos in the low_angle dataset are blurry and hard to identify for me.) Going forward, I'll consider using the full dataset I gathered.

Once I formatted the dataset properly, I followed this blog post's ResNet-based architecture (https://jkjung-avt.github.io/keras-image-cropping/) almost to the letter. I had to solve some of the bugs caused by running tensorflow-gpu remotely on a jupyter notebook. I also chose to stretch the images out rather than compress them all to the smallest image in my dataset, as the original ResNet was trained on (224, 224) images. I also increased the batch size and epoch count from the blog post, since TJ's GPU's are much more powerful than what the blog author had availible.

With these tweaks and bugfixes on the blog post, I was able to achieve an 85.50% validation accuracy on the partial dataset. This surprised me; I thought I would have to adjust more for more than 75% accuracy. My next steps will be to write a jupyter notebook to visualize an arbitrary loaded model's predictions on a single image, then combine this with the board segmentation script to fulfill my Winter Goal. I'll do this with Keras' load_model method, so when I retrain the network I can just load new weights in and nothing will break. Afterwards, I'll look at precision and recall metrics, to see how I can improve my network with better metrics than just accuracy.