Journal Report 10

11/18/19-11/23/19
Kevin Fu
Computer Systems Research Lab
Period 1, White

Daily Log

Monday November 18

Finished labelling dataset from last week for 795 labelled squares in total.

Tuesday November 19

Out sick.

Thursday November 21

Out for band trip.

Friday November 22

Found and labelled squares of lower-angle chessboard images. 585 new labelled squares added to dataset (though some obscured pieces were labelled with a ?). Created script to count stats of images gathered: each type of piece, total number of each color, etc.

Saturday November 23

Added validation script that allows me to rename mislabelled pieces. Decided on image augmentation parameters. Reorganized code into new folders, commented all of my code from past few weeks.

Timeline

Date	Goal	Met
Nov 11	Label roughly 30 more chessboards for 1000 training photos, and research and implement image augmentation	Done
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, started
Dec 2	Modify ResNet architecture for data then begin training	Not started
Dec 9	Either split CNN into three pieces (detailed below) or improve piece-recognition CNN to at least 75% accuracy	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)	Not done

Reflection

Though I was out for two of three Syslab days this week, I'm proud to report I'm ahead of schedule. Keras' `ImageDataGenerator` function is phenomenally simple, and my partner's success with a high-accuracy lattice-point-detecting CNN trained on 50 images is encouraging, since with my low-angle images added, I have at least 50 images of each piece type (though not 50 of each piece and color, as the statistics below show). As such, rather than add more images to my dataset, I'll try augmenting the data I have and throwing it at ResNet. It might be more effective than I hoped for.

If not, my plan is to try breaking the problem into smaller ResNet-based CNNs. First, a CNN to differentiate between white pieces, black pieces, and empty squares, then two separate CNNs to differentiate between pieces: one for black pieces and another for white pieces, used as necessary according to the first CNN. In the event that throwing everything at one CNN doesn't work, this will hopefully amplify the effective size of my small dataset.

Output of image counting script:

```
piece types:

black bishop: 29
black king: 26
black knight: 32
black pawn: 164
black queen: 26
black rook: 52
white bishop: 42
white king: 26
```

```
white knight: 47
white pawn: 171
white queen: 28
white rook: 63

?: 20
[...]
---------------
totals:

black: 329
white: 377
empty: 1256

pieces: 706
squares: 1962
```

Sample image augmentations on a white knight: