Journal Report 21

3/9/20-3/13/20

Kevin Fu
Computer Systems Research Lab
Period 1, White

Daily Log

Monday March 9

Added piece detection model from last week to data collection process. Now labelling is just correcting the model's predictions. Tried training on ResNet50 for 40 epochs, 98.33% accuracy.

Tuesday March 10

Labelled 90 images in roughly half an hour. Merged new images with old dataset, capped each class at 1000 images. Copied to snowy.

Thursday March 12

New dataset performs worse than old one, despite being more balanced. Copied full (un-capped) dataset to find similar accuracy levels. After 20 epochs, ResNet152 achieves 98.46% accuracy.

Timeline

Date	Goal	Met
Feb 24	Finalize orthophoto tweaks, begin piece labelling (March 6th)	Done
Mar 2	Finish piece labelling, train new model for States (March 6th)	Done
Mar 9	Gather more data, play with ResNet types	Done
Mar 16	Label more data, add additional chess logic, increase data augmentations	Not started
Mar 23	Try on Google Coral, start hand occlusion detection	Not started

Reflection

After trying smaller ResNet variants, I've decided to stick with the 152-layer model. The results from the 50-layer version weren't bad, but Kevin Chung tells me that on his GPU-enabled laptop, piece detection takes only half a second to run, even with all the shearing and homography transforming I do before it gets to the neural network. Thus, I think going to a smaller model would needlessly sacrifice accuracy for speed.

I've decided that I need to add more images of varying camera angles and brightnesses to my dataset, and I'll also increase the related data augmentations (channel shift and brightness) to assist with that. It's going to be strange working from home, but on the bright side, I'll have much more time to do the grunt work of labelling images. This is now much quicker and easier, thanks to the addition of the piece-detection model from last week into the data collection workflow.

Updated confusion matrix from this week's training:

```
Confusion Matrix
[[ 88   0   0   2   0   0   1   0   0   0   0   0   0]
 [  0  80   0   1   0   0   0   0   0   0   0   0   0]
 [  2   0  51   1   0   0   0   0   0   0   0   0   0]
 [  0   0   0 398   0   0   1   0   0   0   0   0   0]
 [  1   2   0   1  35   0   0   0   0   0   0   0   0]
 [  0   1   0   2   0  85   2   0   0   0   0   0   0]
 [  1   0   0   0   0   0 585   0   0   1   0   0   0]
 [  0   0   0   0   0   0   1  83   0   0   2   0   1]
 [  0   0   0   0   0   0   0   1  82   0   0   0   0]
 [  0   0   0   0   0   0   0   0   1  56   0   0   0]
 [  0   0   0   0   0   0   1   2   0   0 374   0   0]
 [  0   0   0   0   0   0   0   0   1   0   0  45   0]
 [  0   0   0   0   0   0   0   1   0   0   2   0  94]]
```