Journal Report 15

1/20/20-1/24/20
Kevin Fu
Computer Systems Research Lab
Period 1, White

Daily Log

Tuesday January 21

Finished training network from previous journal; achieved 98% validation accuracy after a total of 300 epochs. Tried different hyperparameters, but none surpassed this model.

Thursday January 23

Realized I mixed datasets when training prior model, retrained on correct dataset, found worse accuracy. Started writing script to get all possible future positions from current one. Currently, script can handle all moves except pawn captures, en passant, and castling. Created confusion matrix for Tues' model.

Tuesday January 28

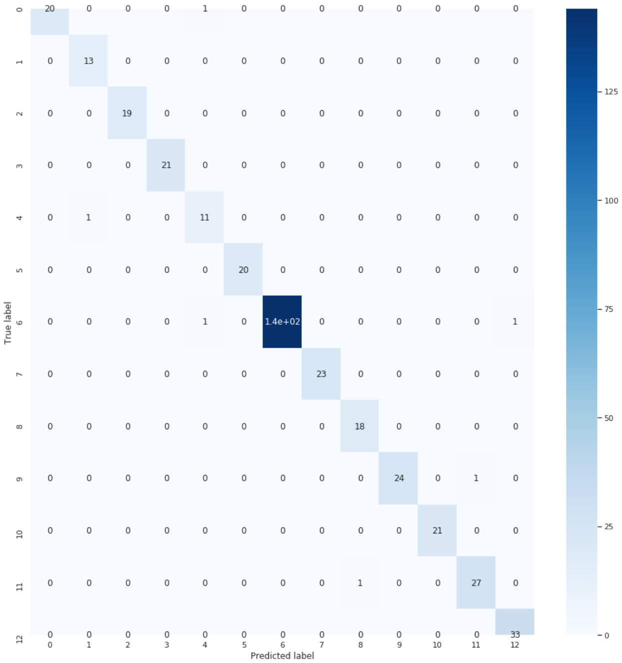
Met with Kevin Chung to start integrating our segments together before States in March. Decided to modify projected height threshold (more on that in Reflection).

Timeline

Date	Goal	Met
Jan 6	Reorganize and comment code, figure out how to load Keras models quicker, consider training empty/not-empty network	Code commented, waiting on hardware sticks, considered and decided against
Jan 13	Continue experimenting for better piece-recognition model	Done
Jan 20	Finish experimenting, work on reducing real-world runtimes by cutting down squares model looks at (chess logic)	Started
Jan 27	Separate input mechanism from rest of Winter Goal script, integrate new model/square-checking into Winter Goal script, continue chess-logic implementation	Done integrating, in progress
Feb 3	Adjust piece height thresholds, potentially retrain model	Not started

Reflection

This week, I trained a new ResNet model on an expanded dataset, and found a significant accuracy boost. The improved model produces this confusion matrix:



Though impressive, this confusion matrix highlights how imbalanced my dataset is (that dark square in the center is the empty class). When Kevin Chung and I met to test this on a real chessboard image, we found that the model still inaccurately classified a few pieces that seemed obvious to us.

To mitigate this, I will reduce the script's height projection from 2 vertical squares to 1.5, in hopes of cutting off pieces in the background. I might also try retraining the model on this new height-projection, but since this would mean relabelling all the images, I'm planning only to do that as a last resort. For now, I'll continue to work on reducing the number of squares the neural network has to look at by adding chess logic.