

Daily Log

Monday March 23

Started making graphical display of nnet's confidence intervals.

Wednesday March 25

Finished graphical display of confidence intervals.

Thursday March 26

Split the 500-line identify_pieces.py into two separate scripts: one for the nnet-independent pre-processing, and the other for the nnet-dependent piece classification.

Friday March 27

Continued improving readability of the two new scripts. Kevin Chung suggested I change data labelling to take pgn files and assume the images are in the sequence of a game.

Saturday March 28

After getting over how much time I would've saved had I done this from the start, changed data labelling to take a pgn file and assume the images of an input directory are in sequence.

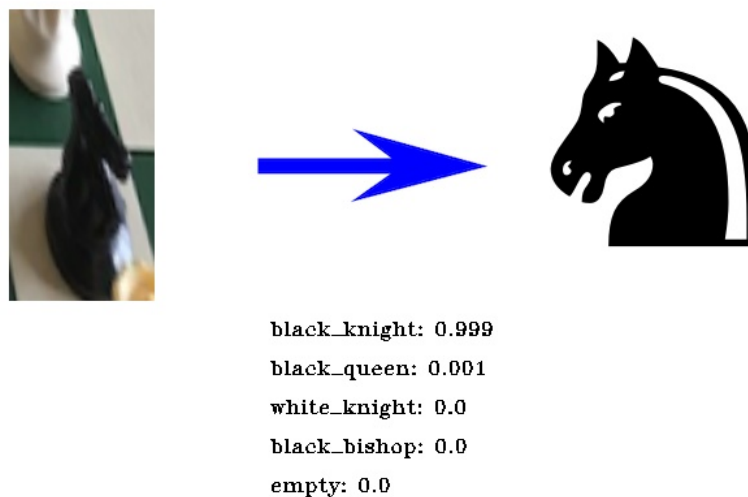
Timeline

Date	Goal	Met
Mar 9	Gather more data, play with ResNet types	Done
Mar 16	Label more data, add additional chess logic, increase data augmentations	Done except additional chess logic
Mar 23	Make UI graphics, see if hand occlusion is feasible	Done, not started
Mar 30	Add additional chess logic, consider ensemble learning	Not started
Apr 6	Spring Break	NA

Reflection

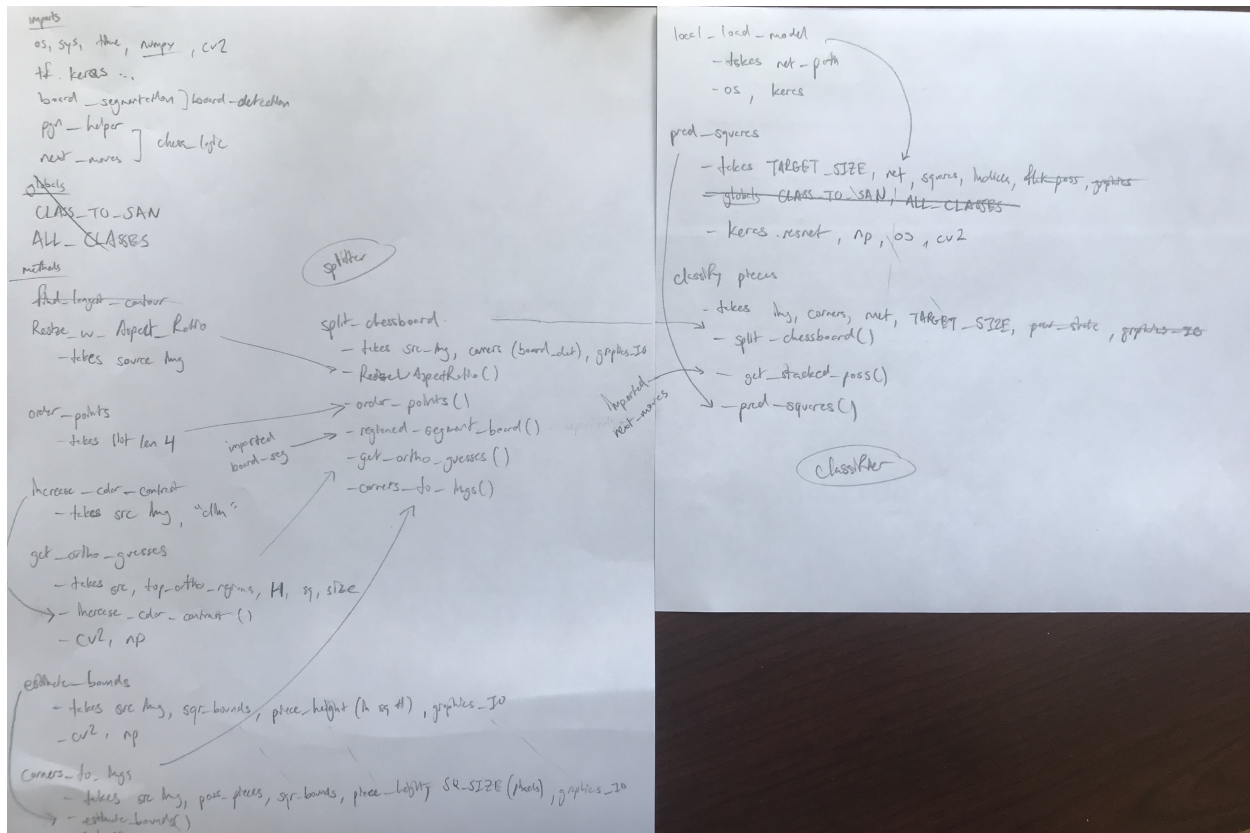
Got a bit off-track this week, but I think the time I spent working on improving the readability of the piece detection code and the time I spent speeding up the data collection process will pay off. Kevin Chung and I both agreed to try and finish this project, despite school being announced closed for the year on Monday. I'm planning to try ensemble learning, or averaging the confidences from multiple neural networks (likely three for speed), as the current ResNet model is incredibly sensitive to lighting changes. I'll also add chess logic to the board predictions: ie, one king necessary for each side, max eight pawns, etc.

Two photos of my work this week. First is the UI graphic I produced to show what the nnet's predictions of a given image are:



Second is the paper/pencil diagram I used to figure out how best to split the piece detection script into two. (I eliminated some unnecessary globals and method parameters by doing this as

well.)



Also, Kevin Chung added a live video mode to the UI. It processes quickly enough to track a real game, although it doesn't transcribe pgn yet because we don't check for bad frames (ie, hand-occluded ones). I think an easy solution would be to run a legality check: if there's a frame where an impossible move is played, discard it. This assumes a perfectly functioning piece detection step, however.