

Daily Log

Tuesday October 15

Modified board segmentation script to allow me to label imgs quickly. After the input `wk`, a hypothetical square 15 of a board becomes `15_K.png`. A black king would be `bk` for `15_k.png`.

Wednesday October 16

Met with Kevin Chung to explain each others' code, chart path forward. He put his board-finder into a live feed. We decided to limit scope of angles final product will handle, based on limitations with piece detecting fully-occluded pieces.

Thursday October 17

Made board segmentation script accept directory of images, and create separate subfolders for each image in input directory. Used tripod to get higher-angle images of chessboards. Also took images with multiple chessboards and with a chessboard taking up very little space. Uploaded to Google Cloud.

Monday October 21

Reread paper cited in project proposal last year to figure out how to improve piece detection beyond throwing images into a CNN. Found paper used an "orthophoto" or top-down view.

Tuesday October 22

Implemented orthophoto (which turned out to be `findHomography()` + `warpPerspective()` in OpenCV). Drew visual representation of the areas of squares to be looked at for piece detection on orthophoto with OpenCV. Researched path forward: k-means of the colors for detecting pieces in regions of orthophoto.

Thursday October 24

Realized paper said it took canny edge points from regions rather than color matching; added canny and played with parameters to get black piece edges to show. Made region easier to resize by adding HTML-style margins. Separated pgn reader/writer methods into separate backend files for organization.

Timeline

Date	Goal	Met
Oct 7	Finish PGN writer	Done
Oct 14	Add end-of-game moves to PGN writer, modify partner's board segmentation script to tag images	Added as command-line arg, done
Oct 21	Gather and label images from chess videos with board-seg script	Looking into making labelling better with orthophoto, not started
Oct 28	Use perspective transform on board segmentation to identify pieces	Started, need to search graphically-drawn regions of orthophoto
Nov 4	Gather and label images with orthophoto guesses considered	Not started
Nov 11	Research and implement image augmentation on segmented images	Not started

Reflection

Last journal (two weeks ago), I wrote that my goal for this week was to make Kevin Chung's board-segmentation script more user-friendly, which I could then use to begin labelling images for CNN-based piece detection. I got the user-friendly part done—as the Daily Log says, the software now takes a directory of images, and allows me to label every individual square with a couple keystrokes.

However, on PSAT Wednesday, Kevin Chung and I realized that in low-angled cases, many of the chess pieces would be obscured. For example, a pawn right behind a queen of the same color has nearly no protruding edges and would be hard to see even for a human observer. So we resolved to use higher-angled images for AutoPGN: in a live example we'll have to raise the tripod, and it will restrict the chess videos we can parse, but otherwise this won't affect what we're accomplishing. We still aim to be able to handle a video feed of a FIDE-standard chess match and turn it into a list of moves automatically (end-of-game notwithstanding).

This shift in priority made me realize I could do more to detect pieces before throwing them at a CNN. One source Kevin Chung and I found last year for our proposal achieved CNN-like quality, but by using a technique called oriented chamfer matching. While ultimately I still plan to use a CNN to detect the pieces on the board, the chamfer matching forced the researchers to get creative with how they determined the locations of pieces on the board, as getting well-scaled images of squares with pieces on them was crucial for their technique. They did this by generating a top-down perspective of the board, then analyzing roughly the top half of each square in this "orthophoto" for canny-edge-detected pixels. Top-half regions with more than a certain number of edge pixels were deemed to have pieces. This got around the problem of tall pieces, like queens or kings, that covered more than one square on the regular-perspective photo.

So, I was a bit stalled on the image labelling. But hopefully once I finish implementing the orthophoto technique, I'll be able to 1) use it to label squares with pieces much quicker, since I won't have to label every empty square, 2) find difficult empty-square cases for the CNN, and 3) put this into the final piece detection software to make it more accurate than a raw CNN alone.

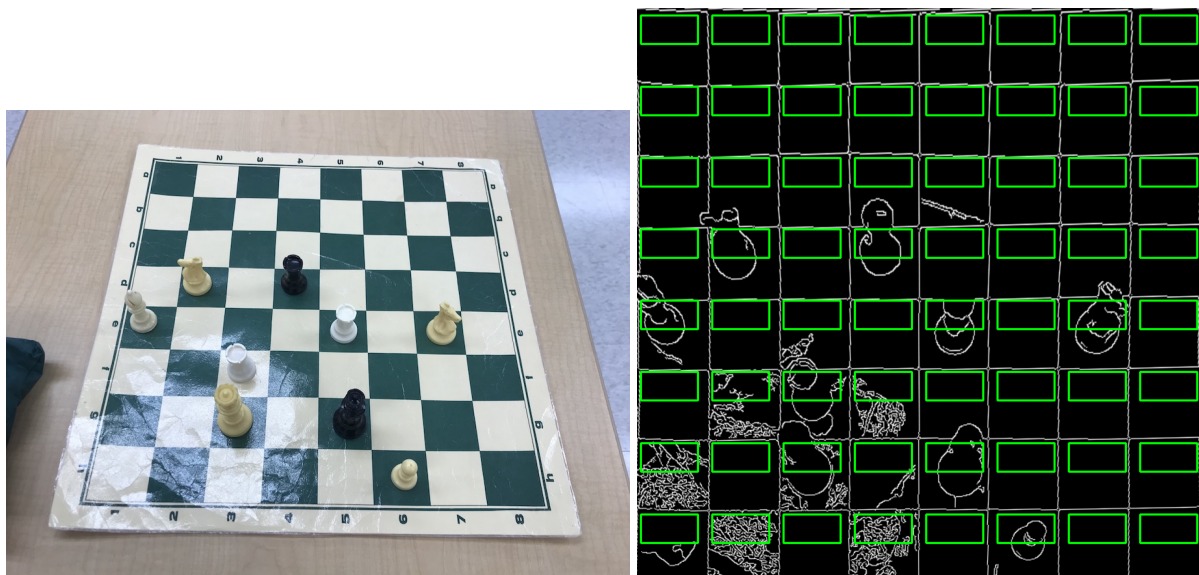


Figure 1: Regular color photo of chessboard (left) and corresponding orthophoto (right) with piece-search regions shown in green.