Journal Report 9
11/11/19-11/15/19
Kevin Fu
Computer Systems Research Lab
Period 1, White

Daily Log

Monday November 11

Took roughly 40 frames from chess videos for piece labelling. (Only ended up using about half since rotated boards won't work with partner's current board-segmentation script.) Added flag to piece labelling script to display pngs at the proper size.

Tuesday November 12

Downsized input images to piece labeller that had a resolution too large to fit on my screen. Changed piece height to 1.75 square lengths, then back to 2. Labelled 760 individual squares from roughly 20 board images. Found a couple boards where Canny-based piece detect failed on black pieces.

Thursday November 14

Fixed Canny-based piece detect by removing Gaussian blur, making for a sharper image. Updated GitHub README and changed remote to put code on GitHub classroom. Researched solutions to problem of differently sized input images. Started redoing all board image labels with new Canny settings. Researched tips for working with small datasets.

Timeline

Date	Goal	Met
Nov 4	Gather and label squares with orthophoto guesses considered	Done
Nov 11	Label roughly 30 more chessboards for 1000 training photos, and research and implement image augmentation	Square labelling half done, image augmentation looked into, not implemented
Nov 18	Finish labelling squares, count square types, add more square labels if needed	Not done
Nov 25	Implement image augmentations	Not started
Dec 2	Modify ResNet architecture then begin training	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

This week, I labelled a full dataset's worth of images before realizing that some of my boards were missing the black chess pieces entirely. I tried to fix this by experimenting with the Canny thresholds and brightness and contrast changes to the original chessboard images. Eventually I found that removing the preprocessing Gaussian blur created enough contrast to find the black pieces in every image, without creating too many false positives for my piece recognition system. I'm currently roughly halfway through relabelling the chessboard images with the no-Gaussian Canny orthophoto in place.

After creating my dataset, I will analyze how many images of each piece type and color I have. However, compared to other ResNet-based image classifiers, my dataset is tiny, especially for a 13 class problem (empty square, 6 white & 6 black pieces). I think it might be smart to split the piece-recognition into several small CNNs to get around this: first, square with piece vs empty square; then if piece detected, black vs white piece; then pawn vs other piece; then if needed, differentiate between other pieces. However, I may just add some more images to my dataset and try throwing it all at a ResNet-style CNN first.

I also did research on image augmentation to expand my dataset. I'll have to crop the images because currently they're all different sizes, but they need to be the same size for a CNN. As such, I plan to crop random sections of the images, expanding the dataset at the same time. I also plan to apply randomized rotation, width shift, height shift, brightness, shear, zoom, channel shift, and/or a horizontal flip to make my dataset larger and more robust.

After all this preprocessing, I will hopefully be well on the way to my winter goal. For said presentation, I'll click on the four corners of a chessboard image, and the script should output what pieces it thinks are on the board. I'm a little afraid that 75% accuracy is too high, because I just picked a random percentage when writing my goal, but hopefully combining a large, augmented dataset with a smart CNN architecture will be able to achieve this.