Kevin Chung

Period 5

2/3/2020 - 2/7/2020

Journal 17

This week, I talked to Dr. Gabor about plans for the upcoming month, since we want to be able to showcase our project at chess states. The biggest thing was creating a user interface that allows users to easily interact and understand the process behind the recognition. I’m currently working on the user interface along with improving the board detection.

For the user interface, I decided on a rather simple GUI using Tkinter. The user will be able to specify what kind of input they would like to use, and also view each individual step of the process. They will also be able to view an animation of images showing each step of the process in order.

On another note, I successfully combined both Kevin Fu’s and my module to pipeline the whole process together. We now have code that is able to take an image or a single frame from live input, and output an estimate of the current board state. After lots of testing, I was able to figure out areas of improvement:

* Depending on the number of straight lines in the image, the image processing can take anywhere from 8-25 seconds on my personal machine, which is much slower than we want it to be. We will test our software on the GPU cluster at TJ, but it is difficult because only certain libraries are available on the GPUs.
* Board detection is often faulty at lower camera angles. This is most likely due to pieces obscuring the intersections on the board, which my algorithm uses to determine a decent board frame. One way we could address this is to run the board algorithm more than once to narrow down the exact frame of the board. However, this would add significant time to the process.
* Piece detection is near perfect when there are isolated pieces on the board (a piece is surrounded by empty squares). However, when, for example, a pawn is in front of a queen, the network gets quite confused. We believe that this is due to the training dataset being heavily biased towards empty squares, meaning the network is less trained on the classification of actual pieces.
* The most common misclassifications for pieces are a pawn as a bishop or knight, a queen as a king, or a king as a queen. It almost never gets the color incorrect. We will try to clean this up by implementing chess logic into our algorithm (i.e. one side can only have two bishops max.)