AUTHOR GUIDELINES FOR ICIP 2019 PROCEEDINGS AUTHORS

Renzo Teruya

The University of Texas at Austin

ABSTRACT

This paper details an approach to analyzing a physical chess board. The goal is with a picture of a chess board it can be reconstructed virtually. This can then be further polished. By converting it into FEN notation, the state of the board can be analyzed. The practical use for such a tool can be for self-analysis of a board or as a method to take a physical chess game to-go electronically. The approach and the model used are far from perfect and leave much room for further improvement. Cropping to the board is done through a multitude of edge detection techniques. The model used is a Keras image classification model and is trained using a small set of self-created data. Lastly, the suggested move is done using Stockfish. The overall purpose for this is to showcase image processing techniques learned throughout the span of the Fall 2020 Image Processing Class at U.T. Austin.

1. INTRODUCTION

The ancient game of chess is ever evolving. Especially with the exponential growth of computing, we learn more about the game every day. On websites such as Chess.com hundreds of millions of matches are played out and stored monthly. These games are crucial for computers to solve the game. Every match likely has a never seen position and thus there is something to learn from them. So, digitalizing physical games are very important.

It is also worth noting that computer analysis is essential to player development. What makes chess such an interesting game, is that it is impossible to play perfectly. Nobody plays optimally, that is why it is important to recognize when a move can be improved– to learn from one’s mistake.

That is what this project aims to solve. The image processing parts can really be broken down into two parts. First, the board recognition, how we located and capture the board. Second, there is recognizing the pieces. If these two tasks can be done, the board can be virtually reconstructed.

2. BOARD RECOGNITION

Capturing the board works very differently with different types of boards. That is why for the sake of this project, one board was used. So, the following section denotes how this particular board was framed. A cellphone was used as a network camera to capture the scene. Ideally, once the board is captured it can be cropped to solely show the 64 squares on the board. For consistency in this project, the board was captured bird’s eye-view with flash on.

The board is processed in grayscale first for the purpose of edge detection.

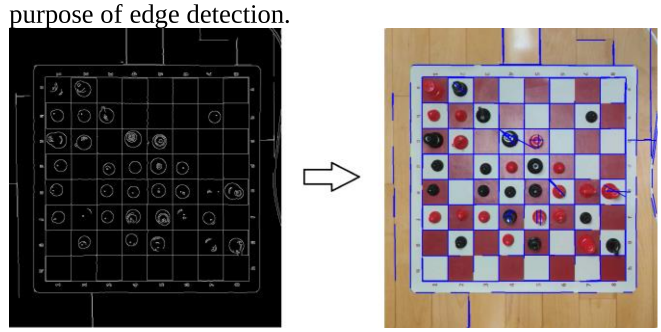

Figure 1. showcases the process used to capture the board. Through a series of Canny Edge detection and Hough Line Transformations, the original image is continually cropped to the largest rectangular object in the frame. This is done based on the intersection of extended Hough Lines.

It is worth noting, all images were taken with flash. A major reason for the use of flash in all pictures ultimately came down to the line detection. In certain environments, based on the amount of natural light or where the board was located, effected the canny edge detection. So, for the sake of consistency, flash was deemed essential.

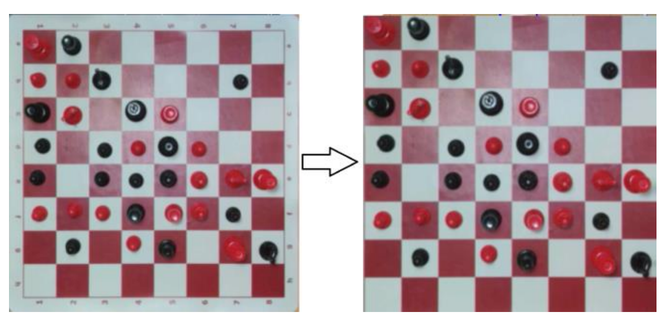

Figure 2. on the left represents the board after the first iteration of this cropping. The right represents the final product – the results of the second iteration. In between these two, note that the letters and numbers are lost. The orientation of the board is crucial in later steps (as it is passed through Stockfish). Thus, this program currently leaves the option to rotate the board later on.

3. PIECE SELECTION

This was arguably the hardest part of the project. The neural network ultimately worked with a reasonable degree of accuracy. In order to make this easier, the chess set selected was black and red with red and white squares. This is because, a decision had to be made. Either there are red pieces on a red square, black pieces on black squares, or white pieces on white squares. Visually, the one that appeared the most contrasting was the red – hence the choice.

Back to the board, this program has captured and cropped to the 64 squares. The difficult question is how to processes the individual pieces. A similar approach with the squares from the Hough Line Transformation could be used, but there was a significantly simpler solution. Since the board was captured using a bird's-eye view, the image can simply be sliced up to 64 squares.

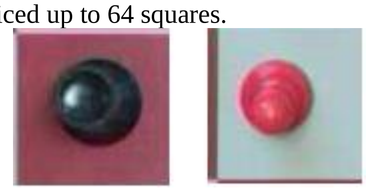

Figure 3. showcases the sliced individual pieces.

3.1. Data Set

A data set of certain chess pieces could have been pulled from the internet; the issue is that the pieces are very situational. Firstly, it is not the standard white and black, but rather red and black. The board is also red and white. There are many variations of chess sets so piece recognition is near impossible. Lastly, all images that need to be classified are top-down. Thus, it should be trained by similar images.

To do so, the pieces were captured the same as above. The board would be scrambled and then it would provide 64 images as part of the training dataset. These images were then rotated to account for different orientations. Sorting these images was done manually. Ultimately a composite of around 1600 images were used to train the model.

For this chess set, it was hard to discern the bishop with the pawns. Surely with more data, a better model can be formed. Yet, with the limited data it was passable.

4. OUTPUT

Forsyth-Edwards Notation or FEN for short is one of the outputs of this project. This is widely recognized standard for describing the board position of the chess game. The state of the board (properly orientated) along with the information of whose turn it is, and which sides can castle is everything needed to compress a board to a single string. The one line string can then be taken to be imported onto an online chess site or used for further analysis.

In order to do that further analysis Stockfish 12 was used. This is one of the strongest chess engines, hence the choice. By feeding it the FEN notation it can return the most optimal move. The purpose of this project at the end of the day, is to improve in chess. With a simple picture, board analysis on any state of the board can be done quickly.

5. FUTURE CHANGES

The result was far from perfect. There were certainly many shortcuts taken in order to simplify the project. This section will cover them and how they can be reapproached.

5.1. Board Recognition

It has already been mentioned that the images required flash and an over-head view. The over-head view can be addressed using series spatial image transformations. The issue with such an approach is that it can cause stretching in the pieces and thus a more robust dataset will be needed to train the model.

In order to shorten the scope of the project, all images were taken with certain levels of lighting. Different lighting caused inconsistencies with the Canny Edge Detection. Not to mention the fact that even if the board could be accurately captured, a more robust dataset would be needed to accurately classify the pieces.

The approach of finding a square and then finding the next square was effective in this environment. The issue arises if the picture was taken on say a square table. It would locate the table and then the outer edge of the board. An alternative would be using an external object recognition program to locate the board. Then doing the necessary processes in order to capture the playing squares. Another option would be to locate the playing grid and crop from there.

5.2. Piece Recognition

More than a larger dataset, there was a better approach. For one, the pieces were not often captured. Especially near the edges, since the board is diced in 64 equal parts, often parts would be cut off. This was especially true for taller pieces, such as the queen and king.

Figure 4. are kings that are essentially cut from their frame. This was mostly circumvented during this project by centering the pieces, yet as noticed it still occurred – especially near the edges. An approach to this could be when testing, take a larger frame

Worth mentioning, a way to increase accuracy - especially considering the limited sample size - would be separating the piece classification from the color. The dataset essentially doubles, and the piece can be predicted from a combination of these two models.

6. CONCLUSION

There is still much to be improved for the program to be practical. Again, like was iterated throughout a large part of this paper, the limiting factor to this program is data. With more data not only will the classifiers be more accurate, but equally importantly - more flexible.

The link to the code can be found here: https://github.com/Reto0/Chess_Scanner

7. REFERENCES

[1] "Forsyth-Edwards Notation." *Forsyth-Edwards Notation - Chessprogramming Wiki*, www.chessprogramming.org/Forsyth-Edwards_Notation.

[2] "Stockfish 12." *Stockfish*, 2020, stockfishchess.org/.