CSCI 1430 Final Project Report: Go Board Reader

Team name: Jikai Zhang.
Brown University

Abstract

This document is a template for your final project reports, presented in a conference-paper style. It is sightly-more complicated LaTeX, but not much more complex than the earlier project reports. This document, along with your code, any supplemental material, and your 2-minute presentation, are qualitatively what determines your grade.

1. Proposal Notes

1. Overriding principle: show us your effort.
2. If you wish us to consider any aspect of your project, it should be presented here.
3. Please include a problem statement, related work, your method, your results (figures! tables!), any comparison to existing techniques, and references.
4. If you made something work - great! If you didn't quite make it work - tell us about the problems, why you think it didn't work, and what you would do to fix it.
5. Length: Approximately four pages for techical description and one page for societal implications.
6. Please include an appendix to the report which details what each team member contributed to the project. One paragraph max each.
7. Any other materials should go in an appendix.

2. Introduction

In this project, I am trying to recognize a Go game from an image containing a Go board. Go is an Asian board game with 19 * 19 crossing points, with two players alternately placing black and white stones on the crossing points. The goal of this project is to obtain the stone position information from an image a user takes.

The project can be roughly divided into two tasks: locating the board and recognizing the stones, each has its own difficulty.

To locate the board, there are two approaches: find the four edges or the four corners. One difficulty of locating the board is: there are too many edges and too many corners both inside the board and, if there are other objects in the image, outside the board. Another obstacle is, sometimes stones may lay on the edges or the corners, making the edge or corner irregular.

To recognize the stones, a most significant challenge is to determine the threshold of black and white. Sometimes when there are strong glares on the board, black stones can become brighter than white stones in the shadow. Another challenge is, sometimes the stones can be irregular (broken stones), and sometimes the size varies on the same board (when the user run out of stones, (s)he need to borrow other stones.

For board locating, I tried Canny Edge Detection with OpenCV's findContours() and perspective transform. For stone recognition, I tried Hough circles and CLAHE (Contrast Limited Adaptive Histogram Equalization).

At the end of a Go game, player usually have to calculate the number of crossing points each one owns to determine the winner. This calculation can be boring and time-costing, and miscalculation happens if someone is careless. If the problem is solved, it will make it much easier and reliable to determine the winner.

3. Related Work

I did not refer to papers for this project, but referred to several Stack Overflow posts for some features:

Four point transformation, Average RGB of circles, Bright reflection reduction

Software Used: Python 3.8, OpenCV, numpy, matplotlib, pygame, tkinter (These are the major ones, you can refer to the requirements.txt file in my GitHub repo)

4. Method

What was your approach? Walk us through what you did. Include diagrams if it helps understanding. For instance, if you used a CNN, what was the architecture? If you changed WebGazer's processes, how does the new system flow look

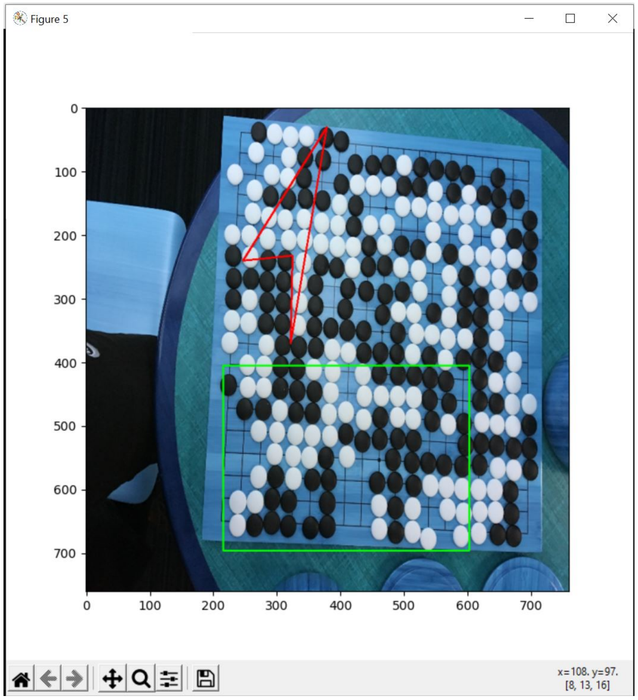

Figure 1. The area bounded by the red lines is the largest quadrilateral my model found.

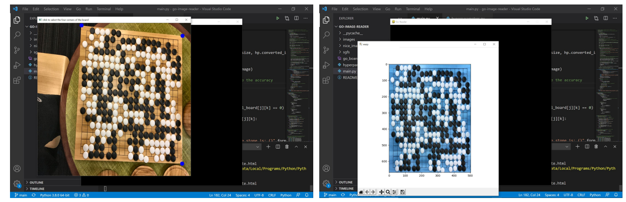
Figure 2. Original image VS cropped image.

vs. the old system flow? Include equations as necessary, e.g., Pythagoras' theorem (Eq. **??**):

As said in the Introduction section, the project is roughly split into two parts.

To locate the board, I first tried using Canny Edge Detection with OpenCV's findContours() function. Before the Canny Edge Detection, I reduce the noise in the original image by applying a Gaussian filter. Then the edge detection helps to find all edges. Next, using OpenCV's findContours() function to find the quadrilateral with the largest area. In theory, this would likely be the board. However, in most circumstances the model fails to identify the board edge as a single connected line (Figure 1). It may be considered multiple dashed lines, none of which can be identified as a quadrilateral or a shape with area.

As an alternative, I decide to let users mouse click on the four corners of the board. I rearrange the 4 positions in the order that starts at the top-left, and goes clockwise. Then I perform a perspective transform to "crop" the board and fit in in a rectangle.

To recognize the stones, I first tried a very naive approach: select four points around each crossing point on the board and check if one of them is above a white threshold or below a black threshold. Such a method reached an accuracy that is not so bad, but setting a hard-coded threshold for black and white, as well as assuming the four points I selected will fall on the stone, are all questionable.

Realizing the above problems, I used OpenCV's HoughCircles() function, which is super helpful in finding circles in an image. After a few experiments, I figured out the parameters with relatively high accuracy. For a 760 * 760 board image (i.e. 40 * 40 grid size for 19 * 19 board), circles with radius between 2 and 45, and minimum distance of 15 has the best accuracy out of all the numbers I tried.

After finding the stones, the next task is to detect its color. I first locate each stone (circle) in the board image using OpenCV's circle() function and compute the average RGB value of the stone. To determine its color, the first approach I tried is very simple: set a pixel threshold for black and white (the threshold with the highest accuracy is 120). But I suddenly realized a problem: the brightness between two images can vary significantly. So I set the average RGB value of all stones on the same board as the threshold.

A very interesting yet difficult problem with color detection is the bright glares, which may happen if there are strong lights over the board or the stones. This brightness difference within the same board increases the error rate of color determination. I added two features to reduce such effect. First, I used CLAHE (Contrast Limited Adaptive Histogram Equalization), which reduces the contrast caused by the strong light. Then I only take 40% radius of the stones when computing the average RGB value, because glares tend to not happen near the center of the stones.

5. Results

Asking the users to click on the four corners of the board gives us a very good result in cropping the board, as shown in Figure 2.

For the stone detection, there are two important accuracy information that we care the most: the accuracy of determining whether there is a stone on a crossing point or not (we notate it as **Accuracy 1**), and the accuracy of determining both the existence and the color of the stone (we notate it as **Accuracy 2**). Note that $Accuracy2 <= Accuracy1$ because if a stone fails test 1, it cannot pass test 2.

At first, I used the naive stone detection and simple color detection method I mentioned in the Method section, this method achieved an accuracy of 88.60% for Accuracy 1, and 87.01% for Accuracy 2.

Then, I improved it by applying HoughCircles() for stone detection, and setting the threshold between black and white as the mean RGB value of all stones on the board. With

	Accuracy 1 (existence)	Accuracy 2 (existence + color)
Naive stone detection + Naive color	88.60%	87.01%
HoughCircles + mean color threshold	95.06%	94.42%
Add CLAHE feature	96.18%	95.55%
Calculate stone color on smaller radius	96.18%	95.89%

Table 1. Accuracy comparison between different approaches

this change, Accuracy 1 jumped to 95.06%, and Accuracy 2 jumped to 94.42%.

Next, to reduce the effect of bright glares in stone detection, I first applied CLAHE. CLAHE helped my model to reach 96.18% accuracy for Accuracy 1, and 95.55% for Accuracy 2.

Finally, to reduce the effect of bright glares in stone color detection, I only compute the middle part of the stone (0.4r radius). And Accuracy 2 improved to 95.89%.

To make it easier to understand, I made a table above (Table 1).

5.1. Technical Discussion

I think there are nothing particularly interesting I discovered with my methods. One problem with my project is, although I am gradually increasing the accuracy, it is not practical as a product since the accuracy is not 100% in most cases. This means that the product is not very reliable, the users may still need to verify and modify the result we output to match the real board, which also makes the product inefficient.

5.2. Societal Discussion

1. Describe the socio-historical context of your project to identify three broad societal factors that could affect your data, goal, and/or hypothesis. These factors might include current or historical policies, events, social conditions, and larger societal systems. Cite at least one outside source.

   A problem of my project is, it is hard to apply for official Go matches. After the emergence of AlphaGo, mobile phones and computers are prohibited during the match (because players have tried to cheat). If the match lasts for two days, then the players have to hand in all their electronic devices, and cannot communicate with anyone in case of cheating. If we allow referees to use our project for judging the winner, it is possible a player pays someone to edit the program and cheat.

   Even if there is no one cheating, it is hard to make sure the model reaches 100% accuracy in all circumstances, it is not worthwhile to take this risk in official matches.

   Thirdly, Go is a popular game in East Asian for two thousand years, it is not only a game, but also a cultural symbol. Matches played by top player is crucial in cultural transmission. Therefore, the calculation of the area in the end of the game, as part of the culture, should not be replaced in official matches.

2. Who are the major stakeholders in this project? What is your relationship to these stakeholders?
   Stakeholders are those who may be affected by or have an effect on your project topic. Some examples of stakeholders are a particular demographic group, residents of a particular geographic area, and people experiencing or at risk for a particular problem.
   Consider the following questions to help identify stakeholders:

   - Who does this project topic currently affect?
   - Who might be harmed by your research findings?
   - Who might benefit from your research findings?

   This project affects Go players. It is expected to make the winner determination more efficient and eliminate the possible errors. So in ideal conditions, it should be beneficial to all Go players using the product. However, since my project did not achieve a perfect accuracy (100%), so people using my project may actually be harmed by this inpreciseness.

3. Research or journalism on your broader project topic may have already been conducted. What was the societal impact of existing research? Discuss the implication of this research on your project and consider the following questions to help identify at least one implication. It may affect:

   - How you should frame your goal,
   - How you should design your algorithm,
   - How you should analyze your data,
   - How you should interpret your findings, and
   - How you should present your results.

   Because Go only has popularity among a minority of people, so there was not significant social impact, even though there are very reliable products in the market addressing the problem.

   However, as Go players, when I play Go with my friends on a real board, we rely very much on the auto-calculation apps after they came out. They made the calculation extremely simple: just take a photo, it will immediately show you the result.

   With these successful products in mind, I set my goal very high. I also learned that I should present the board that my model interprets to the user, that is why I made a simple GUI to visualize the result board.

4. How could an individual or particular community's civil rights or civil liberties (such as privacy) be affected by your project?

   Nobody's civil rights will be jeopardized by my project. However, it is possible that hackers can retrieve some personal information from the photos that users take. So it is important that I secure the user photos. (In fact, in my project, there is no database to store the photos, so all the photos that users take are temporary).

5. If you are using data, what kind of biases might this data contain? Do any of these represent underlying historical or societal biases? How can this bias be mitigated? Consider the following questions to help you:

   - Were the systems and processes used to collect the data biased against any groups?
   - Is the data being used in a manner agreed to by the individuals who provided the data?

   My project do not use data. The only input is a user-taken photo, which should not contain biases.

6. Conclusion

What you did, why it matters, what the impact is going forward.

In this project, I designed a model that interprets the Go board information in a user-taken photo with a relatively high accuracy. It can be applied as an interface, with which users can use to analyze their games on an electronic device, or determine the winner.