Visual Heuristics for Image Pattern Recognition

Thomas J. Barnes
tbarnes@gatech.edu

Abstract. The purpose of this project submission is to program an intelligent agent that passes Raven's Progressive Matrices problems with accuracy surpassing a human guessing (at least 7/12 problems correct on Basic Problem Sets D and E and Test Problem Sets D and E) via computer vision algorithms given images as inputs. Python 3.4.3 and the PIL library were used to complete this project.

Introduction

My thought process behind solving the Raven's Matrices problems in each of the projects throughout this course has always been visual. I had an inkling that with the difficulty increasing with each project going forward, I had to leverage a strictly visual approach in designing the intelligent agent's image comparison algorithms. I employed an iterative test harness where I printed the problem information before each visual heuristic is applied to the given problem images to the console, as well as debug values for troubleshooting during the runtime of said heuristics, and concluding with printing the resulting scores and execution times of the visual heuristics run on that problem. This allowed me to see where the agent was performing as expected, not as expected, or if I have made a mistake in the calculations. I will continue to build off of what I had in Projects 1 and 2.

Visual Heuristics

My visual heuristics are combinations of basic functions provided by the Pillow library that essentially represent a logical bit-wise AND, OR, and XOR between two images. **AND_images** will result in an image with the dark pixels that two images have in common and light pixels everywhere else. **OR_images** will make every instance of a dark pixel between the two images present in the resulting image. **XOR_images** will create an image with dark pixels only where there were differing locations of dark pixels between the two images.

Matrix Image Relationship Types

With these building blocks, I had some tools to tackle the several problems in this project. I started by tackling the horizontal and vertical relationships of the 3x3 matrices and later began to tackle diagonal relationships (see Figure 1).

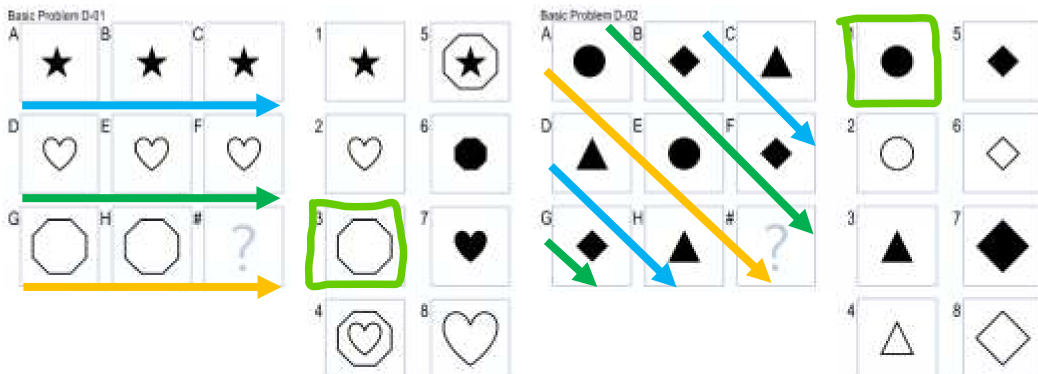

Figure 1. Two examples of the 3x3 matrices required for solving in Problem Set D of Project 3. The first (Basic Problem D-01) is an example of one solved by utilizing a horizontal relationship while the second (Basic Problem D-02) is solved with a diagonal relationship. Correct answers are boxed in green.

Scoring

For each non-diagonal heuristic run, the first course of action is to determine whether the relationship using the heuristic is stronger horizontally or vertically. This comes down to scoring, so usually I would perform something like:

Score(Heuristic(Image_A, Image_B), Image_C)

To translate that pseudocode into plain English, I would run the visual heuristic on the first two images in the first two columns and rows of a problem and compare their resulting scores against one another. Going into this project, the scoring resulted in a floating-point number between 0 and 100, calculated by comparing two lists of binary values associated with each pixel of each image to see where these values (0 for black, 1 for white) intersect, dividing that count by the total number of pixels in each image to develop an intersection ratio, and finally multiplying that ratio by 100 to result in a percentage. If both of the scores associated with the horizontal or vertical relationship were greater than the other, then that would be the assigned relationship going forward with the heuristic. After this process, we check if both of the scores associated with that relationship (either **Score_ABC** and **Score_DEF** for horizontal or **Score_ADG** and **Score_BEH** for vertical) exceed the score threshold associated with that heuristic to even be a valid basis for a solution. If so, potential **Image_I**s are then scored with the result of a heuristic on either **Image_G** and **Image_H** for the horizontal relationship or **Image_C** and **Image_F** for the vertical relationship.

Throughout this development period I made several changes to how my agent scores its answers. What I just described, I later found out to be called the **Intersection Pixel Ratio (IPR)** method. I discovered an alternative method called the **Dark Pixel Ratio (DPR)** method later, in which the difference in percentage of the number of dark pixels with respect to the total number of pixels in each image is compared with an optimally near-zero value to represent similarity. I ended up using a combination of both: DPR first, and then IPR when no answer was chosen from the DPR heuristics.

Guessing

If all else fails and a valid answer is not found with the threshold constraints set for each heuristic, I created a **Get_Guess** function that will generate a random integer between 1 and 8 as a guess answer for each 3x3 matrix problem. This gives me a 12.5% chance of a correct answer when this happens (better than skipping).

Journal Entry #1 (2019-04-11 23:10:44 UTC)

Changes

None. This was the first submission I did without changing any code from Project 2. Just wanted to see where I stood grade-wise before changing things.

Comparison to Human Cognition

The agent checks to see if there is a shape similarity that runs horizontally or vertically without knowing the shapes themselves. It simply checks to see that there is a pattern of similarity and then executes that model upon the final row or column to solve for the answer with the best score with this method. As I went through a lot of the problems in Basic Problems D, I found myself scratching my head at a few of them at first glance and actually scanning the rows or columns just like the agent would to find a pattern. The only difference being, when I found the pattern it was easy to point out the answer using semantic knowledge of geometry rather than mathematical scoring.

Results

The problems that did not incorporate diagonal patterns or extra visual relationships were solved well by the agent (e.g. Basic E-01, Basic E-02, Basic E-3). It struggled with everything else.

Average Execution Time per Problem	Total Execution Time	Basic Problems D Correct	Test Problems D Correct	Basic Problems E Correct	Test Problems E Correct
720 ms	138.189 sec	2/12	2/12	8/12	2/12

Journal Entry #2 (2019-04-18 23:02:14 UTC)

Changes

This submission I added support for the diagonal relationships, which I observed come in two patterns with three groups:

```
# Checking for Diagonal Pattern 1
# [A] (B) {C}
# {D} [E] (F)
# (G) {H} [I]
```

```
# Checking for Diagonal Pattern 2
# {A} (B) [C]
# (D) [E] {F}
# [G] {H} (I)
```

Figure 2. Taken straight from my Python code, these comments are diagrams of the two different diagonal patterns I observed between the two problem sets assigned in Project 3. They are horizontal mirrors of one another.

I was targeting Basic Problems D for this effort and tried to cover at least the basic diagonal problems. During this development process, I developed a helper function called **Similarity_Check_3x3** that takes in three images to simplify the comparison process between three images. It will AND the first two and compare that image to the third image them via the IPR method to return a score. The first step in establishing diagonal similarity is seeing if the relationship exists via similar methods as described in the Introduction for horizontal or vertical relationships, however, instead of the first two rows or columns as an initial scoring hurdle, I would use the two groups that did not include **Image_I** in Diagonal Pattern 1 and then in Diagonal Pattern 2.

This resulted in a boost in number of problems correct at a slight detriment to performance.

Comparison to Human Cognition

A human would be able to see diagonal problems and establish the grouping of shapes naturally, however my intelligent agent has to first establish that there are groups of similar shapes to speak of (not knowing what these shapes are, just that they occupy roughly the same space in each image) and then based on those groups, use the information to find an answer that most matches the AND of the two images of the incomplete group.

Results

The agent did well on simple diagonal pattern problems like Basic D-02, Basic D-03, and Basic D-11. It struggles with diagonal problems with noise like Basic D-06 and D-07.

Average Execution Time per Problem	Total Execution Time	Basic Problems D Correct	Test Problems D Correct	Basic Problems E Correct	Test Problems E Correct
829 ms	159.134 sec	6/12	4/12	8/12	4/12

Journal Entry #3 (2019-04-19 07:55:45 UTC)

Changes

Originally, I was simply using the images themselves to establish diagonal similarity, but a lot of the diagonal pattern groups actually had a secondary visual pattern compounded on top (see Figure 3), so I took the distinct three AND products of each diagonal pattern group and compared those for similarity first to establish that a shape similarity exists regardless of the noise around them. Even then, I'm finding that multiple similar answers are being produced because of the secondary relationship and the incorrect answer may be getting chosen because the agent has no basis on the second relationship. To mitigate this, I develop a "dual pattern image" based on the AND of the OR between **Image_A**, **Image_B**, **Image_D**, and **Image_E** for Diagonal Pattern 1 to incorporate only the similar parts of both patterns in one comparison image against the chosen subset of answers and use that for a second round of scoring. Not knowingly factually why, just based on a hunch of incorporating more visual data as comparison to the already chosen few, this actually produced a correct result for a few tricky problems (e.g. Basic Problem D-07).

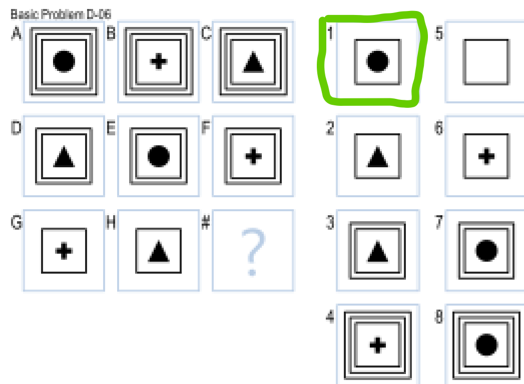

Figure 3. An example of a problem with diagonal relationships that also had compounded secondary relationships on top. You can see one group contains circles, another contains triangles, and the last contains plus shapes.

Comparison to Human Cognition

A human could see Figure 3 and not have to erase the outer squares to see that the inner shapes are almost identical and follow a diagonal pattern. My agent could not do the latter without the former. And even then, it has a hard time choosing between answers 1,7, and 8 because it never develops information for the secondary relationship.

Results

Definitive improvement for Basic Problems D (got Basic D-07 correct this time at the cost of Basic D-06), but took a slight setback in Basic Problems E. It was probably caused by the tweaking of thresholds or that the diagonal dual pattern image tiebreaker doesn't always work as intended.

Average Execution	**Total Execution Time**	**Basic Problems D Correct**	**Test Problems D Correct**	**Basic Problems E Correct**	**Test Problems E Correct**

Time per Problem					
1.189 sec	228.296 sec	7/12	2/12	7/12	5/12

Journal Entry #4 (2019-04-21 06:47:25 UTC)

Changes

Completely refactored scoring to include both the scoring and difference metrics of IPR and DPR, respectively. I also made several quality of life improvements to the heuristics by both splitting every heuristic to have a an IPR or DPR implementation and making sure each heuristic is actually competing with each other for the best answer choosing algorithm (max for IPR, min for DPR)—prior to this submission, certain heuristics like the diagonal similarity comparison were being done after all of the other ones instead of being compared to them in tandem. This was both to make sure that it occurred if a simpler approach did not work and because I was not sure if its thresholds were competitive with the other heuristics. Now every heuristic is on even footing from a comparison perspective (except guessing of course). I noticed the DPR methods complete a lot faster than the IPR methods, perhaps because the comparison is simply a difference of sums of dark pixels made into ratios instead of checking for intersection at every possible pixel location. Added a Subtract_Image heuristic that removes the common dark pixels between two images and compares it to the potential answers. This was added in response to Basic Problem E-04, which would work with XOR_images for IPR scoring *if* the positions of the images were consistent.

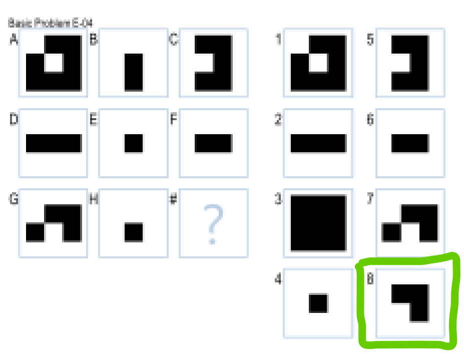

Figure 4. Basic Problem E-04, which has a pattern of removing areas of dark pixels both horizontally and vertically. However, the piece that is removed is moved to be inconsistent with where it would be removed from the first image (e.g. from Image_C to Image_F, the dark pixel block is centered). This makes XOR_images not a valid solution.

Comparison to Human Cognition

The main difference here is obviously that humans do not calculate exact numeric ratios for comparison between two images, but I do think DPR is actually a *more human* approach to cognition because there is a sense, when I do the scanning at least, of comparing the area of dark pixels when comparing the shapes in each image of the matrices. A human could visibly see in Basic E-04 that the dark area that needs to be removed was centered without having to count the dark pixel ratio to check.

Results

Average Execution Time per Problem	**Total Execution Time**	**Basic Problems D Correct**	**Test Problems D Correct**	**Basic Problems E Correct**	**Test Problems E Correct**
1.658 sec	318.412 sec	7/12	5/12	8/12	6/12

Journal Entry #5 (2019-04-21 06:55:00 UTC)

Changes

A lot of very critical parts of the code were written incorrectly. For instance, my RGB_to_Binary function that converts an RGB tuple into a binary decimal for white and black was converting the colors incorrectly. In other words, I was comparing white pixels instead of black for DPR calculations! Luckily it has been incorrect consistently for all images so IPR scoring worked fine until now, but I noticed very strange results for DPR difference scoring. I would have never known. I had an epiphany of using XOR throughout the diagonal pattern problems just like the normal horizontal, vertical pattern ones and it looks like that actually had a positive effect. I saw no real uses for AND and OR though. Additionally, several of the basic functions used in the IPR and DPR conversion process were still there in error, so I made a lot of fixes there.

Comparison to Human Cognition

I would imagine seeing black and white the correct way would make a large difference in ways other than discerning shape similarity. Diagonally though, being able to make more nuanced relationships with XOR does make a difference. I was able to do this without really thinking about it.

Results

Average Execution Time per Problem	Total Execution Time	Basic Problems D Correct	Test Problems D Correct	Basic Problems E Correct	Test Problems E Correct
1.674 sec	321.429 sec	7/12	5/12	8/12	6/12

Journal Entry #6 (2019-04-21 08:06:42 UTC)

Changes

None. This was a re-attempt to try to coax out another point in Test Set E via guess probability.

Results

Average Execution Time per Problem	Total Execution Time	Basic Problems D Correct	Test Problems D Correct	Basic Problems E Correct	Test Problems E Correct
1.455 sec	279.324 sec	7/12	5/12	8/12	6/12

Journal Entry #7 (2019-04-21 08:12:54 UTC)

Changes

Lowered some of the IPR thresholds (XOR from 96 to 94, Diagonal Similarity from 97 to 95) to see if it would affect any of the problems on Test Set E.

Results

Average Execution Time per Problem	Total Execution Time	Basic Problems D Correct	Test Problems D Correct	Basic Problems E Correct	Test Problems E Correct
1.455 ms	279.278 sec	8/12	5/12	10/12	6/12

Journal Entry #8 (2019-04-21 08:22:13 UTC)

Changes

None. I submitted one more time to see if guess probability would be on my side. It worked out in Test Problems E but not in Test Problems D.

Results

Average Execution Time per Problem	Total Execution Time	Basic Problems D Correct	Test Problems D Correct	Basic Problems E Correct	Test Problems E Correct
1.517 ms	291.291 sec	7/12	6/12	10/12	7/12

Journal Entry #9 (2019-04-21 16:32:28 UTC)

Changes

None. I submitted one more time to see if guess probability would be on my side. Fortunately for me, it worked out in both Test Problems D and E this time!

Results

Average Execution Time per Problem	Total Execution Time	Basic Problems D Correct	Test Problems D Correct	Basic Problems E Correct	Test Problems E Correct
1.454 ms	279.242 sec	7/12	7/12	11/12	7/12

Journal Entry #10 (2019-04-21 16:46:17 UTC)

Changes

None. This submission was for more pages!

Results

Average Execution Time per Problem	Total Execution Time	Basic Problems D Correct	Test Problems D Correct	Basic Problems E Correct	Test Problems E Correct
1.485 ms	285.16 sec	7/12	7/12	10/12	7/12

Conclusion

This project was a daunting task to say the least. I had little confidence in reaching the implementation requirement this time around, but I was happy with the results despite the somewhat brute-force means towards the end. I focused on targeting one type of problem at a time this project (starting with simple diagonal patterns, then more complex ones, and finally adding in DPR scoring to refactor how certain problems are scored). Having the variety between IPR and DPR is why I was able to squeak by this time around. The difference between how my final agent solves these problems and how a human would largely stem from its inability to determine movement of shapes across images and discern multiple distinct shapes. A human could discern when shapes are translated over at a certain offset distance, adding or removing shapes each time. Because of this cognitive gap, the likelihood of false positive is higher and utilizes much more computation than a human simply realizing the absence or addition of shapes moving. If I had more time, I would try to tackle the problems with more concise differences (tiny shapes being added on or compounded relationships) by caching shapes as arbitrary profiles to make it easier to determine similarity without having to do expensive comparisons each time around.

References

- Velez, F. (2019, April 20). Basic Problems D & E. Retrieved April 20, 2019, from https://piazza.com/class/jqfcveblst2hc?cid=1510
- Joyner, D., Bedwell, D., Graham, C., Lemmon, W., Martinez, O., & Goel, A. (2015, May). Using Human Computation to Acquire Novel Methods for ... Retrieved from http://www.davidjoyner.net/blog/wp-content/uploads/2015/05/JoynerBedwellGrahamLemmonMartinezGoel-ICCC2015-Distribution.pdf