RPM Milestone C Journal

Ayesha Ilyas
ailyas6@gatech.edu

1 INTRODUCTION

This is the journal submission for the RPM Project's Final Milestone. The Project's overall goal is to develop an AI agent that is able to solve the Raven Progressive Matrices intelligence test. The agent's primary goal is to identify patterns in the given matrix problems and select the correct answer. In the final milestone project report, I detail how the agent evolves to incorporate increasingly sophisticated strategies to handle more complex problem sets. The report covers strategies used to solve Problem Sets B, C, D, and E.

2 AGENT FUNCTIONALITY

While devising an implementation plan for milestone B, I researched two strategies. One that leveraged propositional information contained in images and creates candidate solutions, and the second, that uses visual heuristics (Joyner et al., 2015) and attempts to devise heuristics to vote for the most likely answer. Owing to the simplicity and ease of implementation I decided to use visual heuristic strategies (Dark Pixel Ratio (DPR), Intersection Pixel Ratio (IPR) and further add intelligence to the agent using production rules. I use DPR for 2x2 problems and as prdocution rules for 3x3 problems and use IPR for 3x3 problems.

2.1 2 x 2 Set B Problems

2.1.1 *Dark Pixel Ratio (DPR)*

The dark pixel ratio is "the difference in percentage of the number of dark-colored pixels with respect to the total number of pixels in the contiguous pixel sets of two matrix cells." (Joyner et al., 2015) This approach is also particularly appealing since our input images are black and white, and thus it is easy to compute the number of dark pixels.

For the 2x2 problems in this milestone containing images A, B, and C, the agent first tries to find the difference in the dark pixel ratio for the training images (A and B). The dark pixel ratio for an image is defined by the number of dark pixels in the image divided by the total number of pixels in the image.

In my implementation, the agent first computes the difference in the ratio of dark to white pixels in the two training images. Now, the agent will try to guess the answer by finding which answer choice matches this difference in ratio best. Hence, it will compute this ratio difference with image C and all of the answer choices and see which answer choice given is a ratio difference closest to the training set ratio.

As an example of our 2x2 problem, consider that we have given images A, B, and C and the goal is to find an image D such that the relationship between A and B is mapped to the relationship between C and D. Let DPR(X) be the dark pixel ratio of image X. As shown in Figure 1 (see Appendix Figure 1) we first find the difference in DPR between image A and image B (difference DPR (A,B). This value is what we are trying to get closest to for image C and one of the answer choices, here represented by the set X. Therefore, we find the difference in the DPR (C, X) where X belongs to the set of all the answer choices. Once we iterate over all answer choices, we find which answer choice gives a DPR difference close to Difference DPR(A,B) and vote this as the likely answer.

2.2 3 x 3 Set C Problems

To solve the 3 x 3 problems, the visual heuristic strategy Intersection Pixel Ratio is implemented to cast a vote for the likely answers.

2.2.1 *Intersection Pixel Ratio (IPR)*

The Intersection Pixel Ratio is "the difference in percentage of the number of dark-colored pixels present at the same coordinates with respect to the total number of dark-colored pixels in both matrix cells for a given set of contiguous pixels." (Joyner et al., 2015). To compute IPR between two images, the agent finds the number of dark pixels in both of the two input images, thus finding an intersection of the two images. To normalize, it takes a ratio of the number of intersection pixels and the total number of dark pixels in both images. This is the target ratio that it tries to match in candidate solutions (see the Appendix, Figure 2). The overall strategy is the same as that shared in milestone B.

Instead of generating an image and comparing it to the answer choices to see which choice the generated image matches closest to, this agent (based on visual heuristics strategies) tries to find the most likely answer from the given choices. Thus, it will not generate a solution image but compute a likelihood of which answer it thinks is correct or closest to the correct answer.

The 3 x 3 case has nine images in total (see Appendix, Figure 4) and therefore a larger number of training pairs. To use the heuristic analysis, the agent breaks down the nine images of the 3 x 3 problem into nine 2 x 2 problems (with three inputs and one image to be selected from the set of answer choices). These nine 2 x 2 sub-problems try to capture the heuristic analysis for horizontal (items 1-5) , vertical (items 6-9) and diagonal (item 10) training pairs. Image X in the list below represents the set of all answer choices images provided.

1. Image A, Image B, Image H, Image X
2. Image B, Image C, Image H, Image X
3. Image D, Image E, Image H, Image X
4. Image E, Image F, Image H, Image X
5. Image G, Image H, Image H, Image X
6. Image A, Image D, Image F, Image X
7. Image D, Image G, Image F, Image X
8. Image B, Image E, Image F, Image X
9. Image E, Image H, Image F, Image X
10. Image A, Image E, Image E, Image X

Each of the 2 x 2 cases has four images in total (see Appendix, Figure 3) with image A and image B used as a training pair to find the threshold value for the heuristic (IPR for the 2 x 2 case). The agent then tries to find the answer choice that closely matches the IPR for image A and image B by computing the IPR between image C and all candidate solutions. Hence, given 3 input images and a list of candidate solutions, it votes for the fourth image from the candidate solutions that agent believes is the correct answer. Each of the nine 2 x 2 subproblems identify a candidate answer from the answer choices set. The most voted answer choice is selected as the final answer for the 3 x 3 problem.

2.3 3x3 Set D and E Problems

To solve the set D and E problems, I continue to use the IPR visual heuristic strategy. This acts as a catch all mechanism to solve the problems in these sets. Additional intelligence is provided to the agent by incorporating production rules. These are specific strategies that the agent uses once it encounters specific problems. Production rules are tailored to specific problems and they provide a precise sequence of steps to follow, ensuring a clear mapping between input patterns and output solutions. These rules often mimic how humans might approach solving RPM problems by observing symmetry, alignment, or geometric progression, such as based on recognizing specific transformations or relationships, such as rotations, reflections, additions, or subtractions of image components.

The production rule strategy is different from the visual heuristic strategy (DPR and IPR) in that here the agent generates a candidate image and then tries to find which of the answer choices matches closely to the generated image.

To find the answer choice that closely resembles this created shape, the agent computes the mean squared error between the generated image and all the answer choices. The answer choice with the lowest mean squared error is selected as the answer. The mean squared error measures the average squared difference between pixel values in two images.

Since due to noise created by image manipulation such as rotation etc., it is impossible to generate an image with zero mean squared error, hence I find the answer choice with the lowest value for this error and select as the answer.

2.3.1 *Basic Problem set E 01-04*

For the basic problems in set listed above, I am using a simple Bitwise xor operation between two images G and H in the problem set to generate a candidate solution and then using the similar technique described in section 2.3 find the image that matches closely with the generated image.

Consider the problem image for E-01 (see Appendix, Figure 5). If you explore the relationship row-wise, you will see that the third image in each row is an XOR of the previous two images. An XOR will return a dark pixel only where a pixel is dark in exactly one of the two images. Thus, a true value is returned if a pixel in image one is dark or a pixel in image two is dark. Thus, to find a candidate solution, I will perform an XOR operation on images G and H and the resultant solution will be image 1 and seen in figure 5 (see Appendix, Figure 5).

3 AGENT SUCCESS

I will demonstrate the agent's success for the cases it uses DPR, IPR and production rules to find a solution for a raven's problem.

3.1 Problem solved using DPR

The agent uses a DPR visual heuristic for the 2x2 set B problems. Consider Basic Problem B-05 (see the Appendix Figure 6). The agent calculates the dark pixel ratio between image A and image B (see the appendix Figure 6) and then tries to find the answer choice that will result in a similar dark pixel ratio when calculated against image C and all of the answer choices. Image A contains a triangle and image B a square; the dark pixel ratio calculated between these two images of a triangle and a square will rule out any answer choice containing a triangle. As image C is a triangle, the correct DPR matching that of A and B will be one of the images containing a square in the answer choice. Now, to select which image contains the square, the DPR ratio will again give us a clue. In image A, the triangle is on the lower right corner, and in image B the square is on the lower left side of the image square. The dark pixel ratio will capture this information and, given that in image C, the triangle is on the upper right corner, the answer image matching the dark pixel ratio between A and B will be answer choice 4 that has the rectangle in the upper left corner. A simple pixel-ratio heuristic captures this information and helps the agent to intelligently arrive at the answer solution.

3.2 Problem solved using IPR

The agent uses a DPR visual heuristic for the 3x3 problems. Consider Basic Problem C-01 (see the Appendix Figure 4). The agent breaks down this problem into nine 2x2 problems and lets each sub-problem vote for the final answer (see section 2.2.1). Each of these 2 x 2 sub-problem computes the IPR between a training pair and tries to find the answer choice that matches this intersection pixel ratio between an image from the problem set and one of the answer choices best. The nine training pairs aim to capture the relationships horizontally, vertically and diagonally. The horizontal relationship across images in this example is that they are identical, this means that exactly the same pixels are dark in both images, a candidate solution that thus matches closest to the image from the problem will be most voted. For example, consider the sub-problem image A, image B and image H, image X: here, X is a set of all answer choices. Image A and image B are identical and hence exactly the same pixels are dark in both images. This is the relationship that we will look for among the candidate solutions, hence an image that has the exact same pixels dark as those in image H. This sub problem will therefore vote for answer choice 3. Similarly, vertical pairs such as image A, image D and image F and image X, will try to find the answer choice that matches the IPR between A and D. I find the intersecting pixels and divide by the total number of dark pixels. Here, intersecting pixels are inside square in image D and square in image A. When voting for a candidate solution matching IPR with F, answer choice 3 will be voted again as the inside two squares in answer choice 3 intersect with the two squares in image F and then there is an outer square contributing to the total number of dark pixels.

3.3 Problem solved using Production Rules

Consider basic problem D-06 (see Appendix, Figure 7) where I am adding a production rule to use the DPR heuristic in a specific way so as to solve this problem. As mentioned, I am using my human intuition to provide a strategy for the agent to solve this problem. The initial strategy of breaking down this 3x3 problem into a 2x2 problem was not working, so I explored if there is a candidate 2x2 sub-problem that is representative of the overall problem and can be used to confidently find the answer. I concluded that this representative 2x2 problem is image D, image E, image H and image X. I am using dark-pixel ratio to find the solution to this sub-problem and extrapolating this answer to be the same for the 3x3 problem. The DPR for this subproblem captures the triangle changing to a circle and the outer shape remaining the same, hence answer choice 1 is selected.

4 AGENT PERFORMANCE

The agent answers a total of 42 correct basic problems and 26 test problems, achieving a total score of 66. The production rules are added to target specific basic problems and improve the agent's score. However, the agent struggles to generalize these production rules to the corresponding test problems.

The agent struggles most with the Challenge problems. For set B, it answers only 3 correctly and achieves a 100% accuracy. For Challenge sets D and E, it answers only one question correctly.

The agent does slightly better but still struggles for the Raven problem where for Raven B it answers 6 correct, for Raven C only 2, for raven D only 3 and for Raven E problems the agent gets 4 correct answers.

Table 1—Agent Performance

Problem Set	Total	Correct	Incorrect	Accuracy
Basic B	12	12	12	100%
Challenge B	12	3	9	25%
Raven's B	12	6	6	50%
Test B	12	8	4	66.7%
Basic C	12	9	3	75%
Challenge C	12	1	11	8.3%
Raven's C	12	2	10	16.7%
Test C	12	5	7	41.7%
Basic D	12	9	3	75%
Challenge D	12	1	11	8.3%
Raven's D	12	3	9	25%
Test D	12	3	9	25%
Basic E	12	12	12	100%
Challenge E	12	1	11	8.3%
Raven's E	12	4	8	66.7%
Test E	12	8	4	66.7%

5 AGENT STRUGGLES

The agent struggles to translate the production rules created for specific basic problems to the corresponding test problem. The production rules are specific and do not adapt to a similar problem.

Consider Basic Problem C-05 (see Appendix, Figure 8). While running the agent locally, it incorrectly identifies option 1 as the answer, whereas the correct answer is 3. I believe this is an instance where the nine 2x2 sub-problems that I break down this problem into are not voting for the correct answer. This is because not all problems will be solvable by breaking down the 3x3 problem into nine sub-problems. It may be the case that only exploring horizontal relationship should be enough, or only diagonal or only vertical. For some production rules, I experimented and found one single 2x2 sub-problem that is representative of the overall problem and solving that particular 2x2 sub-problem answer the 3x3 problem.

Next, consider Basic Problem D-07 (see Appendix, Figure 9), in my local run, the agent marks 2 as the answer whereas the correct answer is 1

6 AGENT DEVELOPMENT

From the research that I have been able to do, there are two main strategies for developing an agent that solves the RPM test. One approach is to use the propositional information contained in the images, and the other is to use visual heuristics so that the agent "reduces the input space to sets of contiguous nonwhite pixels".

As far as I have been able to understand, strategies using propositional information are similar to the method described in the lectures where to identify the relationships between two images (A and B), we first identify the components in both image A, and B, secondly we match each component in image A to a component in image B and then try to determine the transformation that the component in image A underwent so that it matches the paired component in image B. In my first report, I had planned to use this method of representing knowledge via a semantic network and identifying the components and their relationships to design the agent.

However, owing to the lack of my familiarity with OpenCV and knowledge of computer vision strategies, I decided to investigate if there may be an easier way to design an agent that meets the criteria for this milestone. This is when I encountered the Visual Heuristic strategies (Joyner et al., 2015), particularly interesting as they appeared to do a very basic comparison using pixel ratios and had a decent success rate. Instead of generating an image and comparing it to the answer choices to see which choice the generated image matches closest to, this agent (based on visual heuristics strategies) tries to find the most likely answer from the given choices. Thus, it will not generate a solution image but compute a likelihood of which answer it thinks is correct or closest to the correct answer.

The visual heuristic strategies, though easy to implement, did not improve accuracy of my agent beyond a point, even though I find tuned the voting mechanisms considerable. This is when I decided to incorporate production rules to assist the agent in solving the basic problems that are known beforehand.

6.1 Human Comparison

My agent is different in its technique from how a human would solve Raven's problem. The human approach is more propositional where we try to identify components of the images and try to gauge how these components are transforming across images to come up with a relationship between images. This is where a human differs from my agent as this agent does not use any propositional information rather use visual heuristics. These heuristics do not directly ascertain how components are transitioning between images, rather what sort of pixel-based relationship exists between images.

Another area of difference is that humans generate a final solution image and then match the answer to the given choices. The visual heuristic portion of my agent does not do this; instead, it calculates a likelihood of correctness for each answer choice. However, for some of the production rules that I implemented, I am creating an answer image and comparing it to the choices. I am using a mean square error to find the most likely answer choice matching with the generated image.

In contrast, human beings do not use pixel ratio analysis to find candidate solutions.

7 REFERENCES

1. Kunda, M., McGreggor, K., & Goel, A. (2011). Two Visual Strategies for Solving the Raven's Progressive Matrices Intelligence Test. In Proceedings of the twenty-fifth AAAI Conference on Artificial Intelligence (pp. 1555–1558). Menlo Park, CA; AAAI Press.
2. Joyner, D. A., Bedwell, D., Graham, C., Lemmon, W., Martinez, O., & Goel, A. K. (2015). Using Human Computation to Acquire Novel Methods for Addressing Visual Analogy Problems on Intelligence Tests. In Proceedings of the Sixth International Conference on Computational Creativity June 2015 (pp. 23–30).

8 APPENDICES

$$\text{DPR(A)} = \frac{\text{Count\_Dark\_Pixels(A)}}{\text{Total\_Pixels(A)}}$$

$$\text{DPR(B)} = \frac{\text{Count\_Dark\_Pixels(B)}}{\text{Total\_Pixels(B)}}$$

$$\text{Difference DPR(A,B)} = \text{abs(DPR(A) - DPR(B))}$$

$$\text{DPR(C)} = \frac{\text{Count\_Dark\_Pixels(C)}}{\text{Total\_Pixels(C)}}$$

$$\text{DPR(X)} = \frac{\text{Count\_Dark\_Pixels(C)}}{\text{Total\_Pixels(C)}}$$

X = {set of Answer choices}

$$\text{Difference DPR(C,X)} = \text{abs(DPR(C) - DPR(X))}$$

Figure 1—Finding the likely candidate solution using DPR

$$\text{IPR(A,B)} = \frac{\text{Intersection\_Dark\_Pixels(A,B)}}{\text{Total\_Dark\_Pixels(A+B)}}$$

Intersection_Dark_Pixels(A,B)= Number of pixels that are dark in both image A and image B

Total_Dark_Pixels(A + B)= Number of dark pixels in A + Number of dark pixels in B

$$\text{IPR(C,X)} = \frac{\text{Intersection\_Dark\_Pixels(C,X)}}{\text{Total\_Dark\_Pixels(C+X)}}$$

X = {set of Answer choices}

Figure 2—Finding the likely candidate solution using IPR

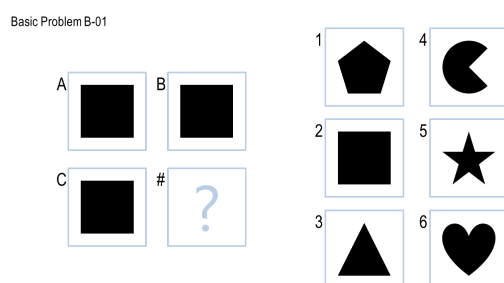

Figure 3—IPR heuristic used for 2 x 2 case

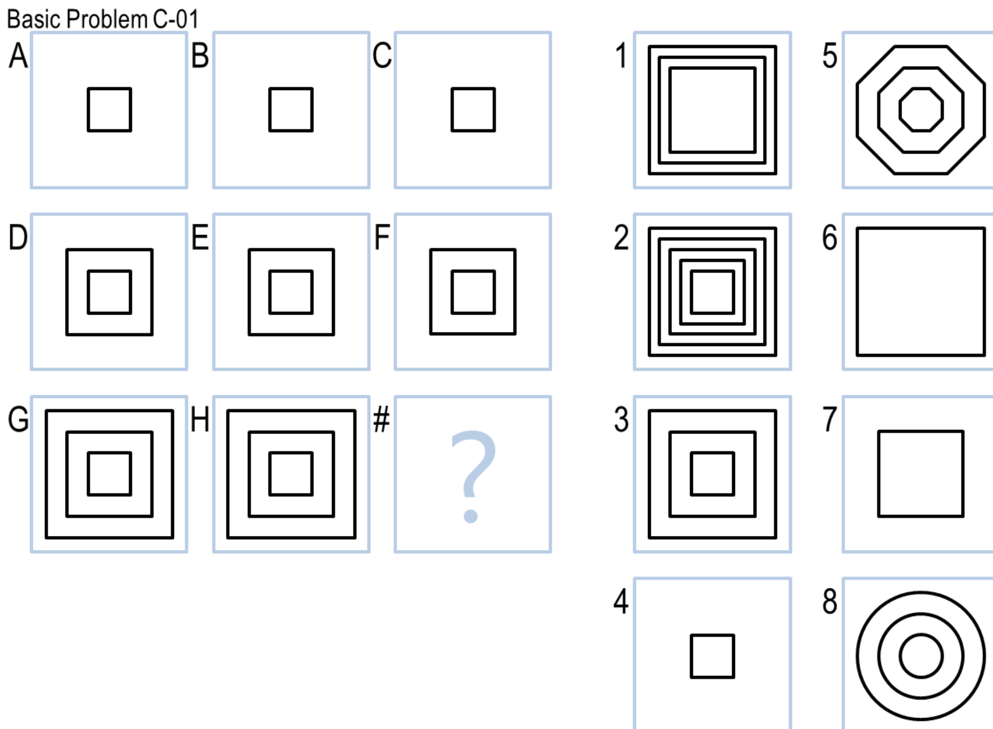

Figure 4—Break the 3 x 3 problem into 9 2 x 2 sub-problems for IPR heuristic analysis

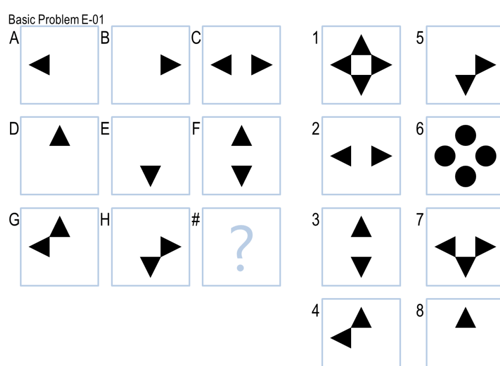

Figure 5—Basic Problem E-01

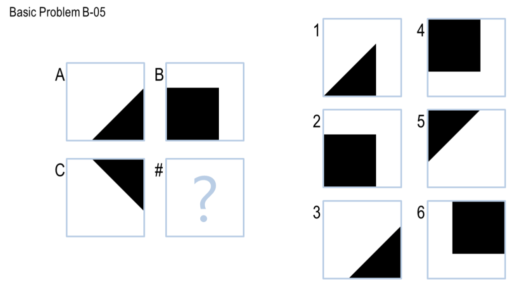

Figure 6—Basic Problem B-05 solves using DPR heuristic

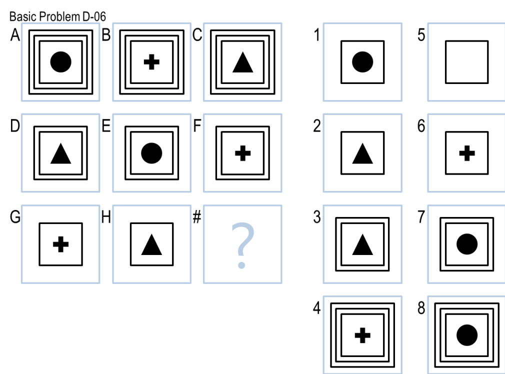

Figure 7—Basic Problem D-06 Production Rule

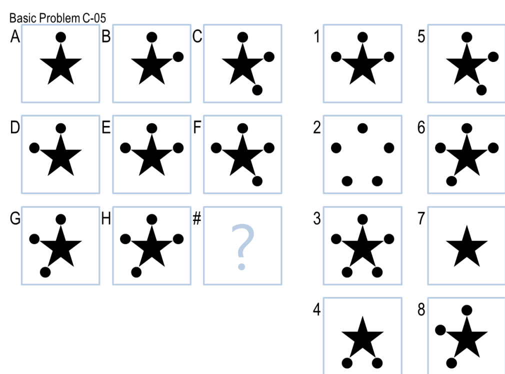

Figure 8—Basic Problem C-05 incorrect answer

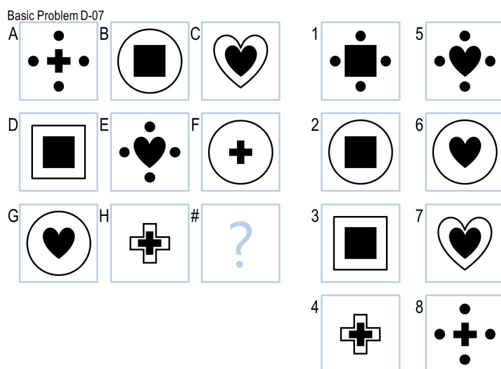

Figure 9—Basic Proble D-07