

## 1) Introduction

The purpose of this project was to build upon the previous projects of solving 2x2 and 3x3 Raven's Progressive Matrix (RPM) problems using combinations of verbal and visual information. In this project 3x3 RPMs were solved using only visual information.

## 2) Theory of Operation

Unlike the past projects, no verbal information was available for this project. Fortunately, a visual approach was primarily used in the last two projects, so a completely visual approach was not difficult to implement.

Once more generate and test was used, and in some cases test and generate. In some cases, the problem would be tested for a certain pattern, and if the pattern matched, the answer was generated according to that pattern. In other cases, the answer was generated first and the provided answers were searched to see if the answer was present. The best match to this generated answer was found, and if the answer provided by the instructor and the generated answer matched closely enough, that answer was returned as the solution.

While a solely visual approach was taken with the 2x2 matrices, the behavior for this project was slightly different. To solve the 2x2 matrices, the relationships were divided into a set of simple image transformations which could be combined to form the answer. In this project the relationships could not be divided into such a set. The patterns were too varied to effectively abstract them, so the problems were handled on a case by case basis. With this structure the agent was able to answer problems it had encountered before, but it was not equipped solve new problems well unless these problems fit the same relationships it was trained to know.

Another difference between the 2x2 and 3x3 designs was how answers were selected. For the 2x2 matrices there were usually multiple ways to generate an answer that fit the pattern but would result in an incorrect answer. Some code was necessary therefore to handle cases where multiple generated guesses appeared in the provided answers. For the 3x3 matrices it was hypothesized that their complicated patterns meant having multiple guesses that matched an answer was very unlikely. Therefore the first guess that appeared in the provided answers (within a certain confidence range) was returned as the answer to the RPM.

While a visual approach was taken here, the way that the agent solves the RPMs visually was different from how the author solved the same problems visually. When initially solving the RPMs, the author found that he would derive a pattern and store this in some verbal form (i.e., shapes plus relationships between them). Then, when looking for an answer, a match to the verbal description would be sought, which entailed translating the visual representations of the answers into a verbal form. While all this happened rather quickly for the author, converting a visual scene to a verbal one was simply not within the scope of this project.

### 3) Implementation

Python 2.7 was used to create the agent for this project. Since a visual approach was used, the Pillow library was used to manipulate the images.

As described above, the approach for this project consisted of a series of cases for generating an answer, usually specific to only one or two problems from the in-sample problem set. If the “guess” image that was generated matched reasonably well to one of the provided answers, that answer was assumed to be correct and the answer number was returned along with a confidence rating. As a result, if a match was made none of the subsequent tests were performed.

Table 1 lists, in order, all the tests that the agent used to solve the RPMs for this project. It should be noted that some problems were deliberately not answered (though this doesn’t mean the agent responded with an “I don’t know”). Also, sometimes the agent had some luck and correctly answered a problem that it was not intended to solve. If the agent goes through all these tests and finds no match, it returns an “I don’t know.” No cases were specifically added for any of the Challenge problems.

*Table 1: Tests used by agent to solve RPMs*

Test Description	Solution	RPM(s) it was intended to solve
Images are all equal along row	Copy any image from last row	C-01, D-01
Difference between columns is the same along a row	Apply difference between E and F to image H	C-11, C-12
D and F are vertical reflections, B and H are horizontal reflections, C and G are 180° rotations	Rotate A 180°	C-07
B and D are 90° CW rotations, C and G are equal, F and H are 90° CCW rotations	Rotate F, logical OR with H	C-08
Pattern shifts right	Copy image A	D-02, D-03, D-11
Pattern along rows, separate pattern that shifts right	Combine patterns	D-06
Unique elements along row (XOR) $C = A \text{ XOR } B$	XOR images G and H	E-07, E-08
Similarity along a row. $C = A \text{ OR } B$	Find sameness between G and H. Answer = $G \text{ OR } H$	E-10, E-11
Unique elements along row, add center back in	Take unique elements to G and H, replace center from image H	E-06
Unique elements along column (XOR of column)	Take unique elements along last column	E-07
Logical AND of all RPM question images	Take logical AND of all RPM questions (A-H)	C-05, E-03, E-05
All the questions appear in the provided answers but one	Select the one answer that does not appear in the RPM questions	C-06, D-07, D-09

[Verbal] If the number of objects is the same among all the question figures AND there is a common object among all the figures	Find answer that has “huge” as an object size and that has the common object and that has the same number of objects	C-02
[Verbal] If all the objects in all the figures have the same shape AND the number of objects in the figure is a product of the number of objects in the first figure of that row	Find answer that has the proper number of the proper shapes	C-03, C-04
Pattern along row, pattern along columns	Combine row and column pattern	D-04, D-05
One pattern shifts left, another pattern shifts right	Combine right and left shift patterns	D-10
Along columns, last row is combination of one top half of first row and bottom half of second row	Combine top half of row 1 and bottom half of row 2 in last column	E-09
Fuzzier version of AND of all questions	Logically AND all the images A-H	E-02
Bisect image along vertical axis. Swap halves	Same as test	C-09

## 4) Results and Discussion

The results of testing can be seen in Figure 1 and Figure 2. As expected, the agent performs reasonably well on the in-sample problems that were used to build the test cases above (Basic Problem Sets). Performance is also reasonable, although poorer, on the out-of-sample test cases (Test Problem Sets). When all graded test cases were taken together, the agent had an accuracy of about 71%. However, the agent’s accuracy drops significantly when the Challenge and Ravens problems were considered. This was expected, though, since the agent was not given the knowledge to solve these problems.

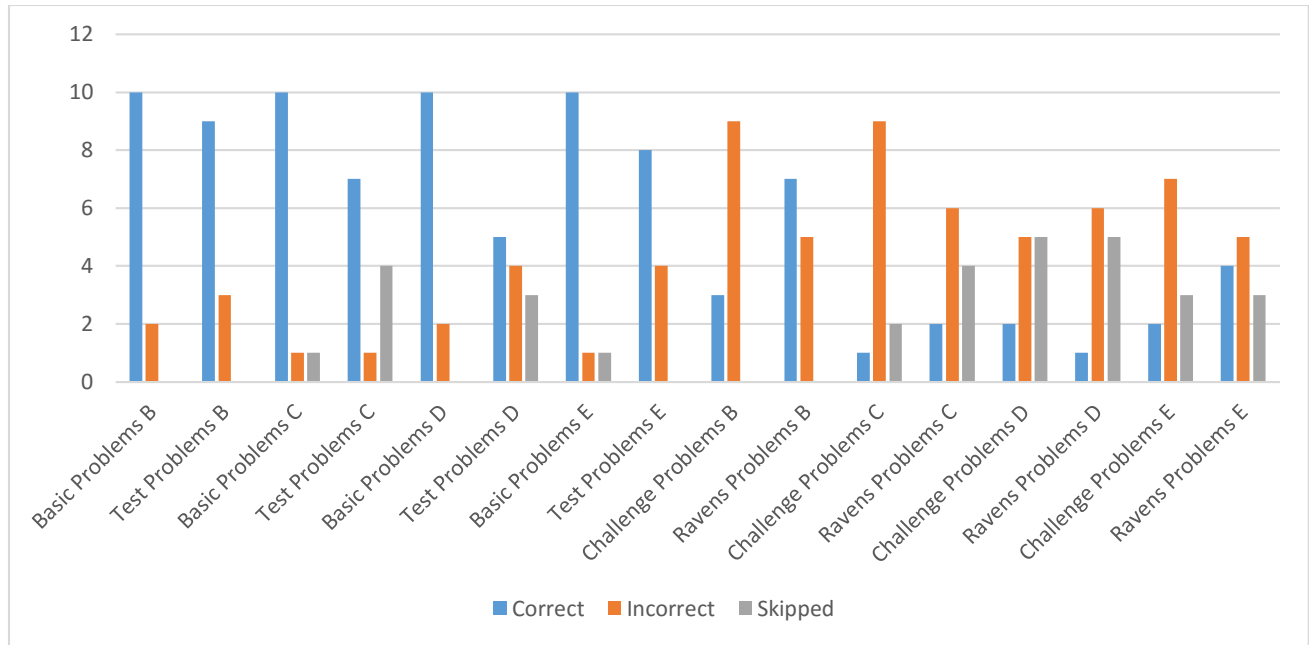


Figure 1: Performance of agent across all test sets

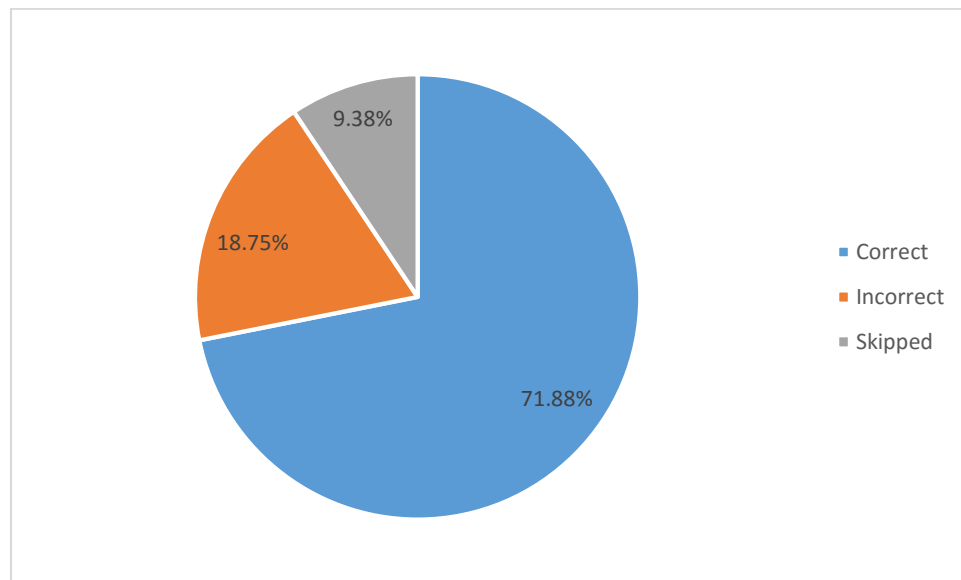


Figure 2: Accuracy of agent across Basic Problem Sets B, C, D, and E

Regarding execution speed, no specific tests were performed. The author did note that the agent seemed slightly slower solving the 3x3 problems than when solving 2x2 problems. However this was expected since the 3x3 problems were larger and would require more time to solve. Furthermore, this project involved answering twice as many problems as the last project, so that increased the execution time as well. The agent took nowhere near the allotted time of 90 minutes to solve all the problems.

It should be noted that solving the problems visually seemed to be far less computationally complex than solving verbally. When solving verbally the overhead of searching through the data structure for a pattern seemed to be much more complicated than applying a few image manipulation commands and comparing a set of images. While no tests were conducted, the author postulates that solving the RPMs visually is at least more computationally feasible and at most more computationally efficient than a verbal solution.

The assumption that false positives would be rare did not hold for this implementation. Due to either small features or some of the imperfections between images, incorrect answers were quite common during the early stages of development. Fortunately an ordering of the tests was found so that the tests answered the specified problems correctly.

Part of the reason the false positives were so common was because the agent did not operate at the proper level of abstraction. It reasoned over pixels while humans reason over shapes. This was how the author reasoned as he solved the RPM. The placement of individual pixels did not matter nearly as much as the relative location of pixels (as they form shapes). The agent was not able to capture the relative location of the pixels and relied on a “dumb” testing mechanism to determine pixel-by-pixel how many pixels were the same between two images. As such this testing mechanism was highly susceptible to minor perturbations between images. Take for example Figure 3 and Figure 4 below. To the average human these two figures are identical, and for the purposes of answering an RPM they are. However, the circles are slightly offset, and when they are compared, as in Figure 5, there are some minor differences. (Note: Black denotes commonality, white denotes difference.) These minor differences were at times enough to confuse the agent as to what the correct answer was.

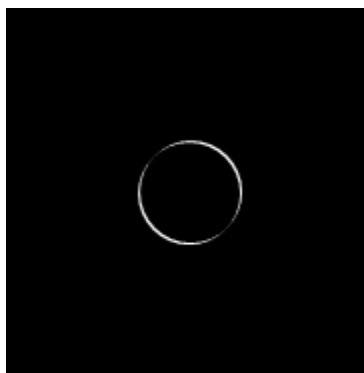
In examining some other work in this area, a weighted image difference is one potential solution. For this, the difference of two images is the sum of the difference of two images multiplied by the weight of the respective pixels. There are many ways the weights of the pixels can be assigned: radially from the center, sum of the pixel location ( $x+y$ ),  $x$  value,  $y$  value, etc. Going further, the comparison could be a two-step process. First the raw image difference is found. Next, a signature is created for the image that is a weighted sum of all the pixels, something similar to a moment of inertia. These signatures could then be compared. Unfortunately, there was not sufficient time to complete either of these methods for this project.



*Figure 3: Basic Problem D-03 Image A*



*Figure 4: Basic Problem D-03 Image E*



*Figure 5: Difference between Basic Problem D-03 images A and E*

## 5) Conclusions and Future Work

In this project an agent to solve 3x3 RPMs was planned, developed, and tested. The performance of the agent depended mostly on the knowledge it was imparted with during development. For problems it was designed to answer the performance was acceptable, but for problems outside of that domain, the accuracy dropped dramatically. This was primarily due to the difficulty in generalizing the 3x3 RPMs into a set of relationships that the agent could then use to solve new problems. Almost every RPM had a different relationship amongst the images, and creating a case for every possible combination simply was not feasible.

The largest shortcoming in the agent was the poor technique for comparing images. This was a simple pixel-by-pixel comparison that was susceptible to noise. A new method needs developed that can determine the relationship between pixels in order to develop the concept of a shape, and then reasoning can be done over shapes rather than over pixels. Weighted image difference was offered as a promising next step.