CS7637: Knowledge-Based AI:
RPM Project: Milestone 4

Jing Gan
jgan34@gatech.edu

1 QUESTION 1

How does your agent currently function?

Please find more demonstrations in Question 3- "What problems does your agent perform well on?". I provide details in how my agents solve different questions.

Directions: My agent still views each question the same way as it does in the previous Milestone Assignment. As described in my Milestone 3,

> "In a 3 × 3 question, my agent mainly considers horizontal and vertical transformations. It does not examine the diagonal relation directly as I think it could be bypassed and compensated by relations in another two directions. Horizontally, the agent will check either 3 figures in 3 ways: 1) 2 figures as a pair – (A, B), (B, C), or 2) 3 figures as a whole – (A, B, C), or 3) addition and subtraction – A+B=C, A+C=B, B+C=A, etc. Similar patterns are checked vertically as well." (Gan, 2023)

Values: Several values are calculated by my agent and used to perform comparisons. As described in my Milestone 3, these values include "DPR (Dark Pixel Ratio), numbers of black (dark) pixels, and numbers of white pixels." In this assignment, I also add the IPR (Intersection Pixel Ratio) and make use of the *cv2.bitwise_and* function and the *cv2.bitwise_or* function to deal with the intersection and union questions that I fail to solve in the previous assignment. As a result, my agent is able to score 95/100 on Gradescope.

Methods: My agent mainly applies direct pixel comparisons (using the *cv2.absdiff* function). As described in the previous assignment,

> "There are 4 main methods: 1) just like Basic Problem C-01 where in each row the 3 figures are the same. The agent will compare the number of dark pixels pairwise to see if the figure remains the

> same and find an answer for Figure 9; 2) the number of dark elements in Row 2 is n times of Row 1, and the number in Row 3 is m times of Row 2. Thus, when the multipliers are found, by calculating and comparing the number of dark pixels the agent can find the answer for Figure 9; 3) each row contains 3 same figures but in a different order. In this case, the agent will calculate DPR for each row and each figure. If DPRs in Row 2 also show up in Row 1, the agent knows that Figure 9 needs a choice that has the missing DPR in Row 3. To deal with the noise, the agent allows a +/- 5% variation in ratios. Again, all such relations are checked vertically as well." (Gan, 2023)

Except for these three methods, this time I also make use of the OpenCV mask to transform the images. More specifically, I use it to rotate inner shapes while keeping the white background the same. After rotating the dark pixels (given shapes), I use mask to remove the original dark pixels and get the new and clean figure so that I can compare it to the next column or row.

Please find more details in Q3.

2 QUESTION 2

2.1 How well does your agent currently perform? How many problems does it get right on the Set D+E problems?

According to Gradescope, the best grade my agent can earn is 95/100 points.

In terms of all four sets of questions, my agent's performance is shown below:

Table 1 — Milestone 4 Gradescope Result

	Basic – D & E	**Test – D & E**	**Challenge D & E**	**Raven's D & E**
Correct	16	9	10	15 Pass
Incorrect	8	15	14	9 Fail

As shown in Table 1, although my agent can earn most of the points, I need to make more improvements especially on the Challenge set and the Test set. After

adding more details or improving the existing algorithms, my agent should be able to perform better on each question set.

3 QUESTION 3

3.1 What problems does your agent perform well on?

For Set D and Set E, my agent performs well on questions that involve the same three shapes but different orders such as Basic Problem D-02, Basic Problem D-03, etc.

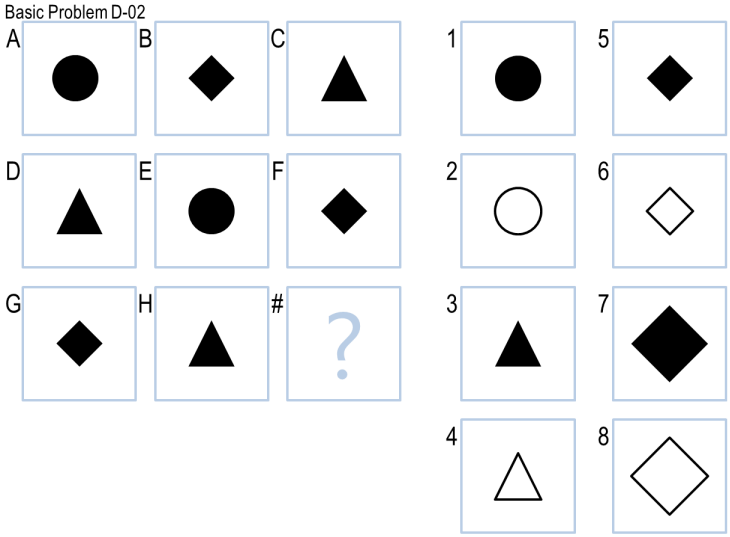

Figure 1— Basic Problem D-02

For example, in Basic Problem D-02 above, it is easy to notice that each row has three shapes: circle, diamond, and triangle. From top to bottom, the last shape is moved to the front when moving to the next row. Therefore, it is expected that the circle (option 1) should be our answer. To find the answer, my agent will first record the DPR for each element Row 1 and Row 2; second, it compares the 3 ratios to those from Row 2 and see if the two sets of ratios are the same or could be treated as the same when the difference is below a threshold I set up for the agent; if yes, the agent will use the ratio set to check the DPRs of Image G and Image H and the unmatched DPR will belong to the answer image. Finally, the agent matches this remaining DPR to each choice and returns the closest image as the answer.

My agent also performs well on questions involving addition and subtraction of specific patterns such as Basic Problem E-02, Basic Problem E-06, etc.

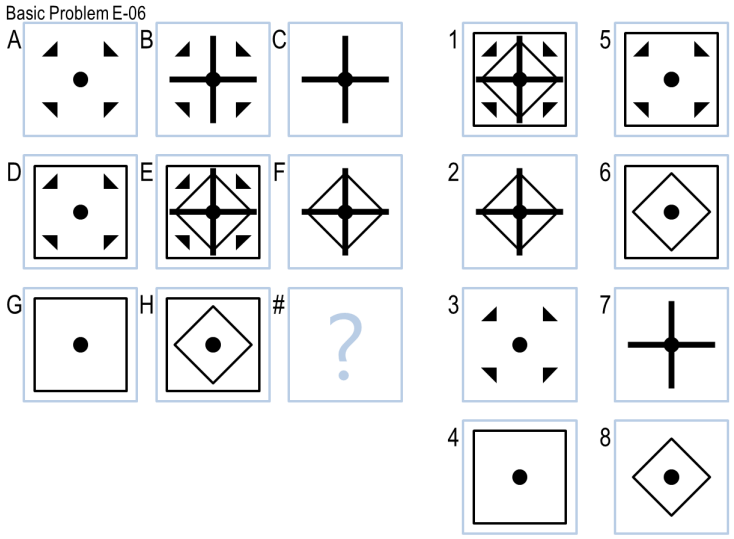

Figure 2 — Basic Problem E-06

For example, in Basic Problem E-06 above, in each row the last image is generated by subtracting the first column from the second column while keeping the centered dot. By following this logic, we expect the answer is to remove the square frame from Image H. As a result, the agent will choose Choice 8. To achieve this, I mainly use the cv2.absdiff() function to find that Column 1 is indeed the difference between Column 2 and Column 3. My agent would ignore the dot for simplicity as it is of low DPR and plays a minor role in the whole image.

My agent also performs well on questions involving the union or intersection of figures in Column 1 and Column 2 such as Basic Problem E-04, Basic Problem E-10, etc.

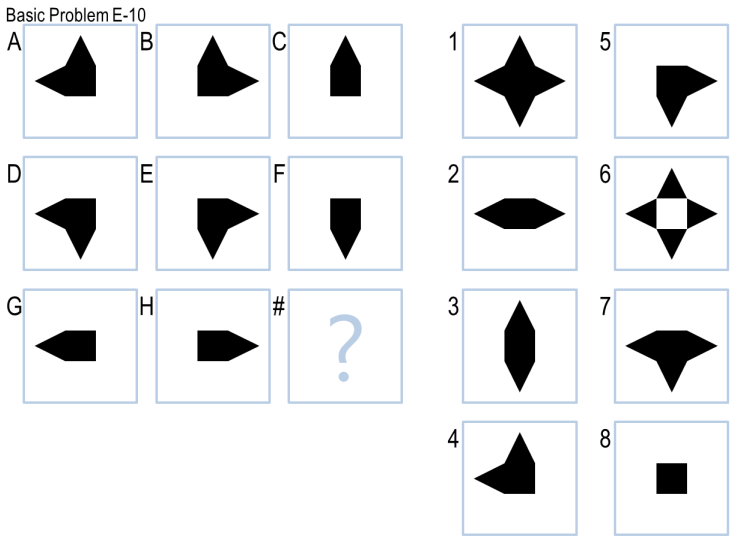

Figure 3— **Basic Problem E-10**

For example, in Basic Problem E-10 above, figures in the last column can be viewed as the intersection of the first two columns. By following this logic, we can expect that the answer is simply a square in the center – Choice 8. My agent would first flip the black and white color of each image, and then it applies the *cv2.bitwise_and* function to find the intersection where both images are white (black in the original images). Finally, it compares the result to Column 3. If both Row 1 and Row 2 follow such a pattern, it deals with Image G and H in the same way and returns the flipped-color Choice 8.

3.2 What problems (if any) does it struggle on? Why does it struggle?

It struggles with problems such as Basic Problem E-07, Basic Problem D-06, etc.

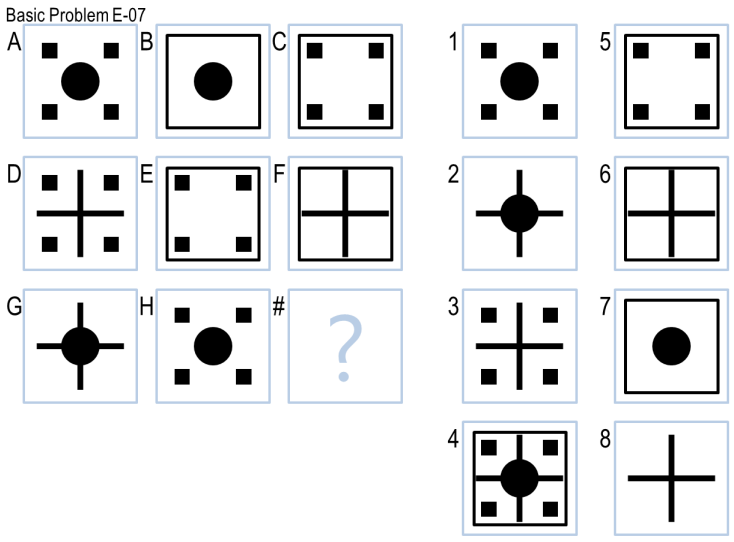

Figure 4— **Basic Problem E-07**

In such questions, it even takes a while for a human to understand the pattern. They are hard for the agent to detect because each image contains more than 2 or even 3 contours. And each contour transforms independently when moving to the next column or the next row. My agent finds it hard to deal with multiple independent contours at the same time and aggregate them as a whole to generate an answer in the last row.

My agent also struggles in questions such as Challenge Problem E-11 and Challenge Problem E-12.

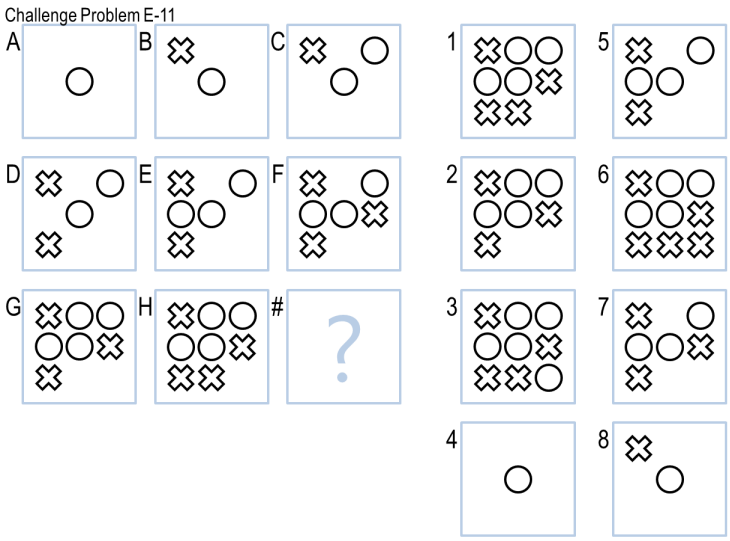

Figure 5— Challenge Problem E-11

For Challenge Problem E-11, my agent needs to deal with not only the cross and circle but also their locations and numbers. The main challenge for my agent here is to deal with the locations and I need to explore a certain algorithm for this.

For questions such as Challenge Problem E-12,

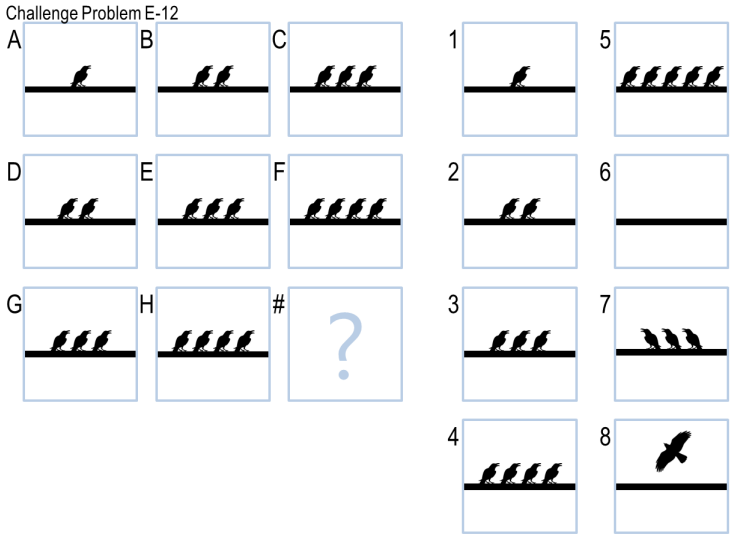

Figure 6— Basic Problem E-12

My agent fails to deal with images with large "noise". In this question, it is obvious that in each next column, a new bird is added. It is intuitive for my agent to subtract Column 1 from Column 2 and Column 2 from Column 3 to find that the pixel difference is the same. However, it seems that the given images are of great noises in dark pixels and the result shows that the pixel difference between Choice 3 and H and Choice 4 and H is even smaller than Choice 5 and H. In this case, although my agent can capture the pattern, it cannot find the correct answer.

4 QUESTION 4

How efficient is your agent? Does it take a long time to run? Does it slow down significantly on certain kinds of problems?

Since I lack the knowledge of Big O notation as I am not a CS major and haven't taken related courses, I would like to describe the efficiency in terms of correctness and running time.

It is very efficient in terms of solving questions quickly. When I test my agent locally, it provides answers in a few seconds. And on Gradescope when it is fed with more questions of more complex transformations, the running time increases significantly. I usually need to wait about 10-15 seconds for the result. However, I see no timeout warning.

In terms of correctness, since I have no access to the test questions on Gradescope, I cannot tell which types of questions are the most difficult to solve (probably fail all the time). I do expect that it would fail on most of the tests that I mentioned in the previous question. And as I wrote in Milestone report 3, "there may be other problems that involve even more complex transformations such as filling, changing both the frames and numbers of certain shapes at the same time."

5 QUESTION 5

How do you plan to improve your agent's performance on these problems before the final project submission?

I would like to improve my agent in two aspects: contours and noise-bearing.

For contours, I would like to explore functions in OpenCV such as *drawContours*, approxPolyDP, and contourArea to deal with several contours as well as their

transformations. I have read some hints from other students in Ed but I think I need more practice to set up my agent.

For noise-solving issues, my agent already uses grayscale to decrease the noises, but it turns out in some questions this is not enough. I would like to discuss with classmates and search online for better solutions. Currently, I have no idea what specific functions I should look into.

6 QUESTION 6

What feedback would you hope to get from classmates about how your agent could do better? What challenges do you think could benefit from someone else's feedback?

As I mentioned in earlier reports, "in fact, any suggestions and feedback are greatly welcomed." If you have more ideas on dealing with contour problems, please let me know!