Project 2(Summer 2020)

Anshuta Awasthi
aawasthi32@gatech.edu

1 INTRODUCTION:

As part of Project2 we are trying to solve the Problem Set C of RPM problems, which consists of 3x3 matrices. The implementation of project 2 is built upon the project 1 which was solving only 2x2 matrices.I decided to go with the pure visual implementation, as it may be helpful for the next project.

In the matrix, 8 out of 9 images are known and we have to design and implement the agent to predict the 9th image by selecting the most optimal answer from the 8 choices given. The implementation strategy is to first solve as many test cases as possible by applying similar logic with DPR (Dark Pixel Ratio) as described in (Joyner, 2015).

Later few Affine transformations have been implemented to solve more problems.

Apart from some common transformations implemented in project 1, I tried to generate the transformed image by reversing /flipping the pixels horizontally and vertically and then tested the generated image similarity with the given options to solve some tricky problems.

Initially, I started with considering all the 3 cells(either horizontally or vertically) to establish the pattern and then apply transformation, but later it was clear that few problems can be solved by just considering the transformation of edge rows or columns and middle cells can be ignored as shown for Basic problem-07 in figure 1 below, it is sufficient to just consider A and C cells and establish the pattern as mirror image. I applied this trick to few other problems too and it improved the efficiency of my agent significantly.

Mean value of pixel by pixel difference between the two images is used as a **"similarity measure"** for finding the similarity or differences between the images.

The production system is built as series of nested if /else rules.

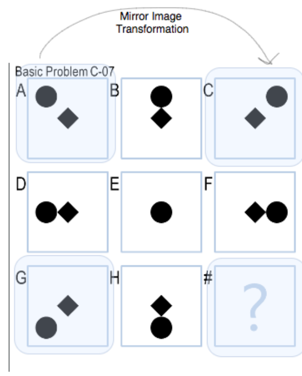

Figure 1—Basic Problem C-07

2 JOURNAL ENTRIES

2.1 Submission 1:

Date of Submission: 2020-06-24 00:04:03 UTC
Changes in this version:
I started to build my Agent on the similar lines of the DPR(Dark Pixel Ratio) algorithm described in the paper(Joyner, 2015). I converted the images to binary 1D numpy arrays for simplicity of operations. This helped me calculate the number of dark pixels in each image.
In this iteration, the production system compares the dark pixel ratios horizontally between the cells C and B , verified if the same ratio exists between F and E , in case it satisfies the horizontal pattern , the agent further tries to find the option which has the same dark pixel ratio with image in cell H. If it fails to find a horizontal pattern, it checks for vertical patterns in a similar manner . Ratio between vertical cells(G:D) is taken first and then verified if the same ratio exists in the other column too (E:H). At last, the option which gives the same ratio with image F is selected as the answer . This helped me to establish the baseline of the system
Agent Vs. Human Approach: Since the Agent is approaching the problem by comparing and calculating the number of dark pixels in each image, this is slightly different than the way humans think about it. For humans, it is easy to identify shapes, number of shapes,size and compare there alignment in the image. While the Agent's approach is very quantitative , since its calculating ratios and difference of pixel values. Also Agent does not have any understanding of spatial distribution of pixels. Although from different perspective, both Agent and

humans are comparing white and black portions among images and trying to establish the pattern.
Outcome ,Efficiency and Performance:
Execution Time : 9.529 Secs

Table 1—Autograder Results

Basic	Raven's	Test	Challenge
2/12	3/12	4/12	2/12

Initially,there was some discrepancy in the results obtained locally on my computer and Bonnie. On changing the python version, the results started to match I was hoping for the better response. To my surprise, it did pass 4 Test problems though.
This result gave me a hint that I am heading in right direction for generalized approach and need to refine my code in order to improve Basic problem score.

2.2 Submission 2:

Date of Submission: 2020-06-24 22:54:37 UTC
Changes in this version:
On the analysis, I found that more than 1 answer options were satisfying the criteria of near dark pixel ratio and hence my Agent was failing. There was a need of threshold refinement. I made the threshold range more restrictive in order to filter out the options which were producing dark pixel ratios very close to the desired one, but they are not the best choice.
I also tried to implement few basic Affine transformations in this iteration.
Agent Vs. Human Approach:
In this Iteration, Affine transformations are also implemented in addition to DPR. While with DPR, Agent is solving problem with very quantitative and calculative way . Affine transformation is very similar to how humans try to establish patterns, by checking rotations , horizontal flip , vertical flip etc.
Outcome ,Efficiency and Performance:
Execution Time : 6.943 Secs
The resulting Agent improved on basic problem result but 1 test problem which was passing before is failing in this submission. This has happened because of more restrictive threshold values. Although basic Affine transformations don't seem to be working in any of the problems.

Table 2—Autograder Results

Basic	Raven's	Test	Challenge
3/12	3/12	3/12	2/12

With Basic , Test and Raven scores all set to 3, I can say that the Agent may not be very efficient, but so far it is working almost equally in all the sets.

2.3 Submission 3:

Date of Submission: 2020-06-25 21:16:17 UTC
Changes in this version:
So far, I was working on dark pixel ratios only , but I realized that in some problems it is not the ratio but the dark pixel difference which will work. For ex. in the problem shown below(figure 2), each image cell is having 1 more black diamond than it's previous cell(horizontally or vertically).The difference between B and C is addition of 1 more black diamond. Same with E and F cells. With the number of dark pixels of image H known , this problem could be solved easily. Since we are calculating similarity measure between 2 images as mean of pixel wise difference(i.e. comparing pixels at the same position in 2 images and count how many pixels match between them). It is not only taking into account the total number of dark pixels but there spatial distribution too.
Agent Vs. Human Approach:

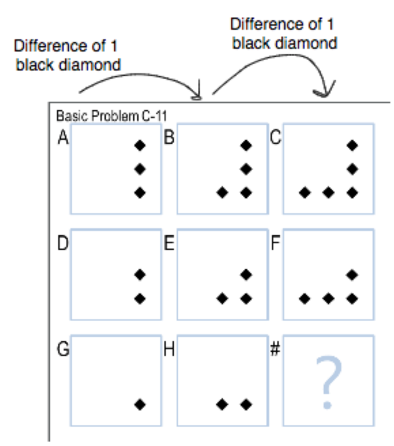

Figure 2—Basic Problem C-11

Again, if we leave the quantitative aspect of Dark pixels calculation, this ap-

proach is very similar to the way humans will perceive this. The above described process is almost same for humans too.
Outcome ,Efficiency and Performance:
Execution Time : 10.723 Secs

Table 3—Autograder Results

Basic	Raven's	Test	Challenge
4/12	6/12	7/12	3/12

The simple change boosted the score to 7 for test problems.It improved the score on raven's and challenge set also. However, for Basic Problem set, I am still getting poor results. Looks like the model of Agent is generalized well, but basic set problems are too specific as compared to invisible test set. Efficiency wise agent's performance is good.

2.4 Submission 4:

Date of Submission: 2020-06-26 12:52:29 UTC
Changes in this version:
For the iterations prior to this, I was considering all the 3 cells in horizontal and vertical rows, to establish pattern. That was making my model more complex for problems like Basic Problem-10 I found it more convenient to ignore middle column and just replicate the transformation between 2 edge cells in a parallel row for example in the figure 3 shown below , the shape in middle of cell A is doubling and overlapping in cell B and then again moving apart in cell C.
If we consider it as one operation of just "doubling of shape" from A to C directly, the logic becomes easy to program.
This way it can be worked more like a 2x2 matrix. This simplified the agent design significantly. However I did not notice any reduction in execution time.
Agent Vs. Human Approach:
The Agent design is simplified in this iteration and only significant cells it is learning from are A,C and G to predict the image in last cell of the matrix. We humans are quick in seeing the transformations such as shapes overlapping , shapes moving apart etc,which may be difficult for Agent to consider. The Agent has made a clever choice to leave the complex transformation.On the other hand, Agent is behaving just like humans in analyzing transformations.
Outcome ,Efficiency and Performance:
Execution Time : 10.837 Secs

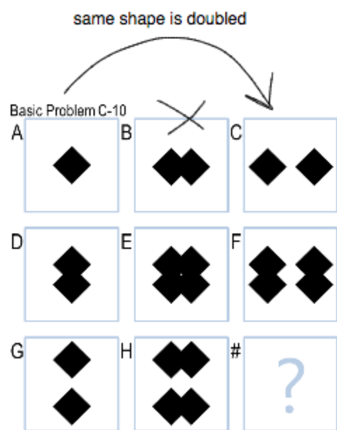

Figure 3—Transformation : Basic Problem C-10

Table 4—Autograder Results

Basic	Raven's	Test	Challenge
7/12	4/12	6/12	4/12

The change helped me achieve the magic figure of 7 passed cases in Basic Problem Set. On the other hand, it reduced the passed cases count for Test and challenge sets. The design of Agent is simplified, I did not see any significant improvement in execution time.

2.5 Submission 5:

Date of Submission: 2020-06-26 21:14:12 UTC
Changes in this version:
Once the Agent successfully passed 7 basic and 7 Test cases, I switched my approach from finding generalized solution to handling each case at a time from this iteration. The method of Affine transformation implemented in submission 2 to only apply the transformation on the corner cells and leave the middle one(row or column). The same transformation was not working in Submission2 , since the agent was looking for 2 transitions - A–>B and then A–>C. Here we are verifying transformation from A to C and then applying it on G to get the resulting image as described in the introduction section.
Agent Vs. Human Approach:
With 8 basic cases passed, Agent is still not compatible to humans.Agent is failing on the problems which could have been solved by individuals easily. The

cognitive connection of agent is same as described before and it has not change in this iteration since we have just extended the previously described approach.
Outcome ,Efficiency and Performance:
Execution Time : 11.014 Secs

Table 5—Autograder Results

Basic	Raven's	Test	Challenge
8/12	4/12	6/12	4/12

Although basic set has 8 cases passing, the change in this submission could not improve the test or challenge score. The test set still has only 6 passing cases now. While earlier it reached till 7 cases and started failing in 1 of the cases after changes were implemented. Now the Agent is focusing on the previously failed cases and trying to solve them one by one.

2.6 Submission 6:

Date of Submission: 2020-06-26 23:07:43 UTC
Changes in this version:
The main focus of this version is to get the Basic Problem C-09 solved. Here the challenge is to address the problem where different shapes in the image have different transformations. The 2 shapes in the image are rotated in different directions.Since the shapes still are not changing , No. of dark pixels in images A to C are not changing at all.Hence the DPR technique will fail. Also all the affine transformations done before are applicable to image as a whole.They don't apply on the individual shapes within an image.
This problem is solved by numpy image slicing and then moving the pixel positions horizontally from the left side of the centre to the right side.Similarly the pixels from right half are moved to left half horizontally. It is implemented under nested if /else rules that performs several checks before establishing the confidence in transformation.
Only transformation from the cell A to C has been verified and the same transformation is then applied to image G in order to obtain the resulting image.
Agent Vs. Human Approach:
I do not think that my agent is smart enough to tackle similar kind of other problems, where the rotation of different shapes are in different directions. The approach and test implemented is very specific to basic problem C-09. As a human, i can visualize the transformation as shapes moving and changing their

positions and rotations easily. While agent is not as efficient as humans in applying rotation or transformations in separate shapes inside an image.
Outcome ,Efficiency and Performance:
Execution Time :12.117 Secs

Table 6—Autograder Results

Basic	Raven's	Test	Challenge
10/12	4/12	6/12	4/12

The performance on basic set is good, but the agent is not generic enough with only 6 cases from Test set are passing. There is definitely need of having more generic solution in order to improve the score on unseen problems. Slicing, Indexing operations on numpy are fast. That is the reason , the execution time has not changed much in this version.

2.7 Submission 7:

Date of Submission: 2020-06-27 01:05:26 UTC
Changes in this version: The strategy in this iteration is to look into some of the problems in challenge set, which are similar to basic set problems, but still failing. One such problem is Challenge Problem C-06, on debugging it was clear that the DPR method implmented before had a flaw and the code block for vertical (columnwise) implementation of DPR was not reachable and not being utilized for any case.
A small fix where the similarity in spatial distribution of pixels in horizontally located cells(similarity between A and C) is compared with vertically located cells(similarity between A and G). The direction with greater similarity in image is chosen as the basis of ration calculations.
Agent Vs. Human Approach:
The agent is not capable of identifying different shapes in the images , hence depends on very quantitative method of pixel comparison between 2 images.While such kind of problems are very easy to spot by humans by just identifying same or different shapes.
However when it comes to ratio, humans also tend to think in the similar lines .For example the ratio between number of triangles in cells A and C is 1:3. If we assume the same ratio between cell C and "unknown" . It gives us the number of triangles in unknown cell as 9. The agent is calculating the ratio of black pixels, which is the same logic.

Outcome ,Efficiency and Performance:
Execution Time: 13.069 Secs

Table 7—Autograder Results

Basic	Raven's	Test	Challenge
10/12	6/12	7/12	5/12

Change in this version has increased the score of Test and Challenge set each by 1. Also the Raven's set score is improved by 2. That shows the changes in this iteration has made our agent perform better on unseen problems. Hence the overfitting is reduced to some extent and agent is more generalized.

2.8 Submission 8:

Date of Submission: 2020-06-27 23:21:36 UTC
Changes in this version:
The aim of this version is to solve 2 basic problems which were failing before. Basic Problem C-08 and Basic Problem C-12 has been solved in this submission. Though the pattern in Basic Problem C-12 is quite clear that vertically or horizontally - one square on the top row gets filled on each cell as we either move from left to right or top to bottom. But we need to convert it to some easy transformation for the Agent.
Here is the logic implemented for Basic Problem C-12 -
If the cell has an adjacent cell above it and an adjacent cell to its left , then the cell image will be result of "OR" operation of both the adjacent cells(top of it and left to it) i.e. image E is the result of "OR" operation between image D and image B. Similarly the resulting image should be OR operation of image H and image F.
Logic implemented for Basic Problem C-08 - 1).Horizontally 2 new vertical blocks are added in first transition i.e. In A to B 2 back vertical blocks are added to the right half of image B
2).In second transition say from B to C , newly added blocks in B(the blocks in right half of B) are replicated to the left half of image resulting in image C.
Note : when the blocks are added they are overwriting the existing blocks. The logic is verified in the second horizontal row and implemented for the third horizontal row.
Agent Vs. Human Approach:
The method to solve both of these problems is very specific to these particular

problem. For Basic Problem C-12 , the way agent is solving the problem is very different from how humans visualize it. For humans it is so easy to see that moving left to right(i.e C–>B –>A) each time a square in the top row gets filled. But Agent is performing OR operation between the two cells which are adjacent (top and left) to it. It is unlikely for human to think like this when he /she looks at the problem first time.

The problem C-08 is the trickiest one in the set, It took me a while to figure out the pattern described above. Hence it is not easy even for humans to solve this. However after implementation, the agent is more likely to solve it the same way as I solved it first time.

Outcome ,Efficiency and Performance:

Execution Time: 12.457 Secs

Table 8—Autograder Results

Basic	**Raven's**	**Test**	**Challenge**
12/12	6/12	8/12	5/12

This gives me the perfect score in Basic problem set. Test and Raven's score are also above average. That makes the agent efficient enough to give good performance on unseen data. With the implementation of last few cases being so particular and specific to the problem itself, It is evident that Agent is suffering from some amount of overfitting to basic problem set.

3 CONCLUSION:

a).Implementation Approach : Starting from the first version to last submission, the Agent has improved incrementally through several iterations of refinement and learning. Several approaches like Dark Pixel based algorithms, Affine Transformation, Image operations using slicing and indexing of numpy arrays have been applied through the process. Agent has been developed as a set of Production system rules with incremental if/else statements. Initially agent takes generic approach to solve the problems through **"Commonsense Reasoning"** with basic operations like identical, expansion, flip, increasing/decreasing no. of shapes ,different rotations etc. and patterns are checked horizontally , vertically or diagonally. Later Agent tries to solve one specific problem at a time and keeps building on the previous iteration by making small changes in the generic approach . That shows that the agent is adapting to **"Case Based Reasoning"**.

b). Cognitive Connection : Agent's design and development through the series of iterations is intimately connected to human cognition ability of learning one example at a time.This behaviour simulates the incremental concept learning.
Almost in all the problems we solved, the transformation is derived from the analogical reasoning by finding conceptual and logical analogies between the images. "Analogical Reasoning" is the core of human cognition too.
Apart from this, we already discussed about how basic transformations we applied are exactly similar to how humans perceive the problems. But when we work on images as array of pixels the agent's approach is little different than humans. It is very quantitative and calculative.
However, at the high level this approach also simulates the human cognition ability to differentiate between the changes in size, color, rotation and movement of shapes in the image very easily.

c).Further Improvements: The final agent is not perfect, but it is performing above average in all 4 problem sets. Also, Agent has improved a lot from previous project. Having said that, there is still lot of scope of improvement. Here, I have only tried algorithms based on Dark Pixels,there are other visual heuristics which could have been implemented such as IPR mentioned in (Joyner, 2015)
Also, Agent is inefficient in finding different shapes in the image, which is why it failed on many challenge problems. Agent is unable to determine the number of sides in a https://www.overleaf.com/project/5edae4558722220001602557shape based on visual method. Also it fails to recognize shapes and generic relationship between them. It also failed on challenge problems which have different shapes in the image filled with stripes. So far agent relies on number of pixels based heuristics. In next project, I am planning to use pixel distance based heuristics too.

4 REFERENCES

[1] Joyner, D.A. (2015). "Using Human Computation to Acquire Novel Methods for Addressing Visual Analogy Problems on Intelligence Tests". In: *In Proceedings of the Sixth International Conference on Computational Creativity.* Provo, Utah.