Raven's Progressive Matrices: A Visual AI Approach

Paul Johnston
pjohnston30@gatech.edu

Abstract—Raven's Progressive Matrices is a series of cognitive tests where a subject is given an incomplete matrix of images and is asked to determine which available answer image best completes the matrix pattern. Here we will explore solving these matrix problems using a visual AI approach that combines both the Affine and Set Transformation Induction (ASTI) model (Kunda et. al., 2013) as well as Dark Pixel Ratio and Intersection Pixel Ratio (DPR/IPR) model (Joyner et. al., 2015) with some modifications to both models.

1 AGENT

The agent described here is a hybrid of the ASTI and DPR/IPR models, with some modifications. The agent first runs the modified ASTI model on the problem, and if the certainty returned from the model is above a certain threshold, the agent selects the ASTI model's answer as the problem answer. If the threshold is not met for ASTI, the agent then tries the modified DPR/IPR model. If the DPR/IPR answer certainty is not above a certain threshold, the agent produces a last-ditch effort of guessing "5" as the answer image. Here we will describe in more detail the process of the modified ASTI and DPR/IPR models.

1.1 ASTI approach

In general terms, the ASTI model takes pairs of prompt images and tries to transform the first prompt image in such a way that it becomes the second prompt image. The various pairs and transformations for 2x2 problems can be seen in Figure 1 and additional pairs and transformations for 3x3 problems can be seen in Figure 2. For 3x3 problems, the agent supplements the training pairs with training triplets, and the transformations with AND, OR and XOR bitwise operators.

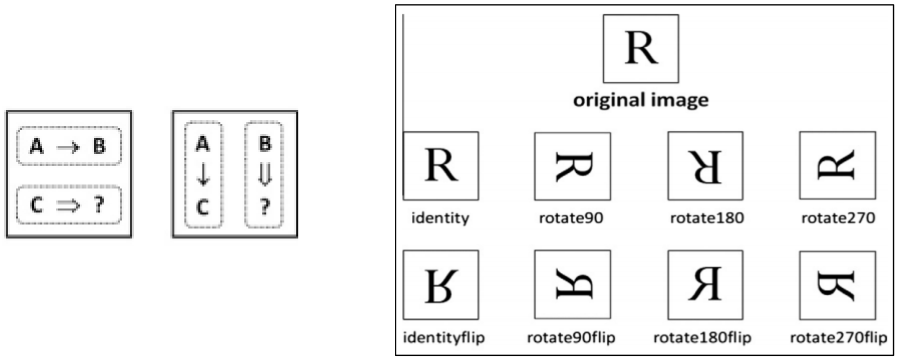


Figure 1 — ASTI training pairs and transformations for 2x2 problems (Kunda et. al., 2013).

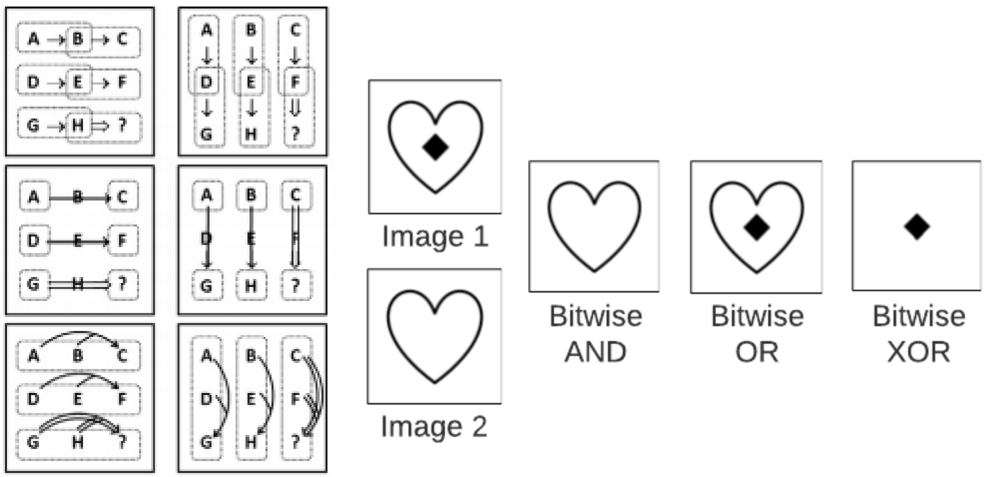


Figure 2 — ASTI supplementary training pairs, triplets, and bitwise operators for 3x3 problems (Kunda et.al., 2013).

For 2x2 problems, the agent iterates through each training pair seen in Figure 1, and at each pair, it applies each transformation also seen in Figure 1 to the first image from the pair. The agent then takes this transformed result and "wiggles" it around in a 10x10 pixel window, calculating a similarity at each wiggle between the second training pair image. It then stores this similarity score along with the training pair, and transformation in a `CompositeTransformation` object.

With 3x3 problems, the agent will also iterate through the training triplets seen in Figure 2. With each training triplet, the agent performs each bitwise operator seen in Figure 2, to the first and second image of the triplet, "wiggling" each image to find the best translation. It then compares the result to the third image in the triplet, saving the similarity score, triplet, and bitwise operator in a `CompositeTransformation` object.

The similarity between two images in this agent is defined as the sum of the bitwise AND divided by the bitwise OR of the two images (Kunda, et. at., 2013). This always yields a similarity value between 0 and 1 inclusive.

"Wiggling" each image is a new technique developed with this agent and a main enhancement to the original ASTI model. With each comparison between two images, the agent will translate the first image in a 10x10 pixel window to find the best similarity value. It does not store this translation value like in the original ASTI model, instead it is meant to account for image jitters and usually results in a 1 or 2 pixel translation.

The agent then iterates through all the CompositeTransformation objects, finding the transformation with the best similarity score. If this were for example a 2x2 problem and the best transformation was rotating image A 90 degrees to fit image B, then the agent would rotate image C 90 degrees to produce a prediction image. The agent then compares this prediction image with the answer images, "wiggling" the prediction image and selecting the answer image that most closely resembles the prediction image according to the similarity value mentioned above.

1.2 ASTI modifications

The main modifications made here to the ASTI model in this agent are:

- No "pixel differences" (Kunda, et. al., 2013) are calculated.
- An additional diagonal training pair for 3x3 problems of A->E.
- AND, OR and XOR operators for 3x3 problems.
- "Wiggling" when calculating the similarity values between images.
- Using the similarity scores to decide if the ASTI answer is certain enough.

The "pixel differences" calculations were meant to account for image discrepancies as the images were scanned in by hand, however, with this agent the pixel differences were producing more problems later on with comparing the prediction image to the answer images. Instead "wiggling" was introduced to account for image jitters. Furthermore, the A->E training pair was added for 3x3 problems and the AND, OR, and XOR bitwise operators were added for 3x3 problems as well. The agent uses the similarity scores of both the CompositeTransformation object as well as the prediction with the answer image to determine if the ASTI answer is certain enough. If not, the agent moves on to the DPR/IPR agent.

2 DPR / IPR

The Dark Pixel Ratio / Intersection Pixel Ratio (DPR/IPR) model is one that solely utilizes pixel ratio comparisons between images in determining the answer image. No prediction image is produced, rather, votes are cast for the most likely answer image, where each vote is proportional to the pixel ratios.

2.1 DPR approach

DPR is defined as the difference of two ratios: the number of dark pixels from image 1 divided by the total number of pixels in image 1, and the number of dark pixels from image 2 divided by the total number of pixels in image 2. (Joyner, et. al., 2015). IPR is defined as first, taking the intersection of the two images. Then, finding the difference of two ratios: the number of dark pixels in the intersection image divided by the number of dark pixels in the first image, and the number of dark pixels in the intersection image divided by the number of dark pixels in the second image (Joyner, et. al., 2015).

The agent iterates through the training pairs seen in Figure 3. At each training pair, the agent takes two metrics : DPR and IPR. With each metric, the agent compares the value with the DPR and IPR respectively of the training pair and compares it to that of the target image (seen in blue) with the "?" image. The "?" image is filled in with each answer choice, and the agent votes on each answer choice only if the DPR and IPR metrics are within a threshold. The vote the agent casts is also proportional to the DPR and IPR metrics. For example, if the agent found a difference of DPRs of 0.3 for answer choice 5, then the agent would cast a vote of 0.7, (1 – 0.3), for answer choice 5.

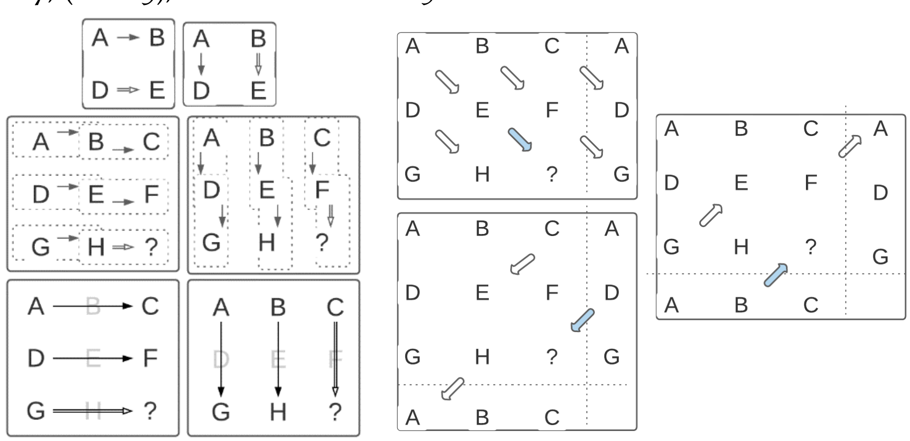


Figure 3 — DPR/IPR training pairs used in this agent.

2.2 DPR/IPR modifications

The main modifications to the DPR/IPR model in this agent are as follows:

- Addition of diagonal training pairs.
- The use of the second-best vote to determine the certainty of the answer.

For example, if the first-best vote value is 10 and the second-best vote value is 2, then the vote ratio (second-best / best) is 0.2, low, meaning the agent is sure. If the first-best vote value is 10 for example, and the second-best vote value is 9.5, then the vote value is 0.95, meaning the agent is unsure, and a guess of "5" is made.

3 PERFORMANCE

This agent was tested on 16 problem sets, where each problem set contained 12 problems, resulting in a total of 192 problems. Of these 192 problems, the agent is able to get 104 correct. The performance by set is in Table 1.

Set	Score	Set	Score	Set	Score	Set	Score
Basic Set B	12/12	Test Set B	10/12	Ravens Set B	8/12	Challenge Set B	4/12
Basic Set C	7/12	Test Set C	9/12	Ravens Set C	5/12	Challenge Set C	3/12
Basic Set D	9/12	Test Set D	6/12	Ravens Set D	5/12	Challenge Set D	1/12
Basic Set E	8/12	Test Set E	7/12	Ravens Set E	5/12	Challenge Set E	5/12

Table 1 — Problem set performance for this agent.

Here we will walk through three example problems the agent solves correctly. The first will be using ASTI traditional transformations, the second with ASTI AND, OR and XOR operators and finally the DPR/IPR method.

3.1 ASTI tradition transformation successful solve

Figure 4 contains a problem the agent is able to solve correctly using the traditional ASTI transformation method of rotating/flipping images to produce a prediction image.

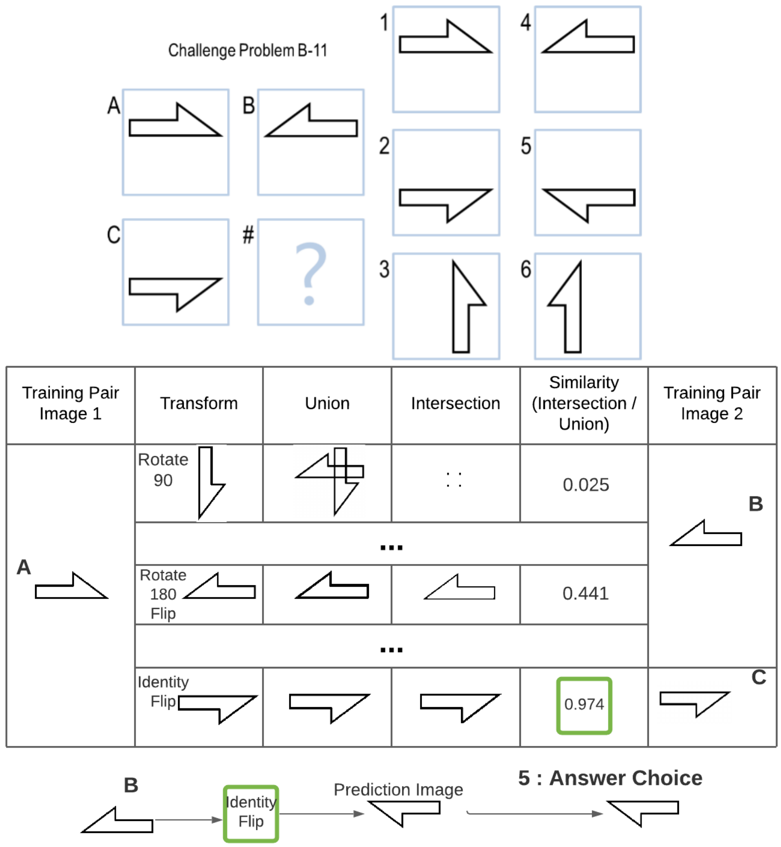


Figure 4 — Tradition ASTI transformation used to correctly solve a problem.

The reasoning process is as follows: The agent iterates through the training pairs, and executes the transforms seen in Figure 1. The agent finds that flipping A horizontally most matches it to C. It then flips image B horizontally to produce the prediction image. It then finds the similarity score between this prediction image and all the images, finding the highest to be with image 5 of 0.974. The agent then finds that this similarity score of 0.974 is higher than the threshold (0.94) to select the ASTI model's guess, so the agent uses the ASTI guess of 5, which is correct.

3.2 ASTI bitwise operators successful solve

With the bitwise operators, the agent uses a training triplet as seen in Figure 5.

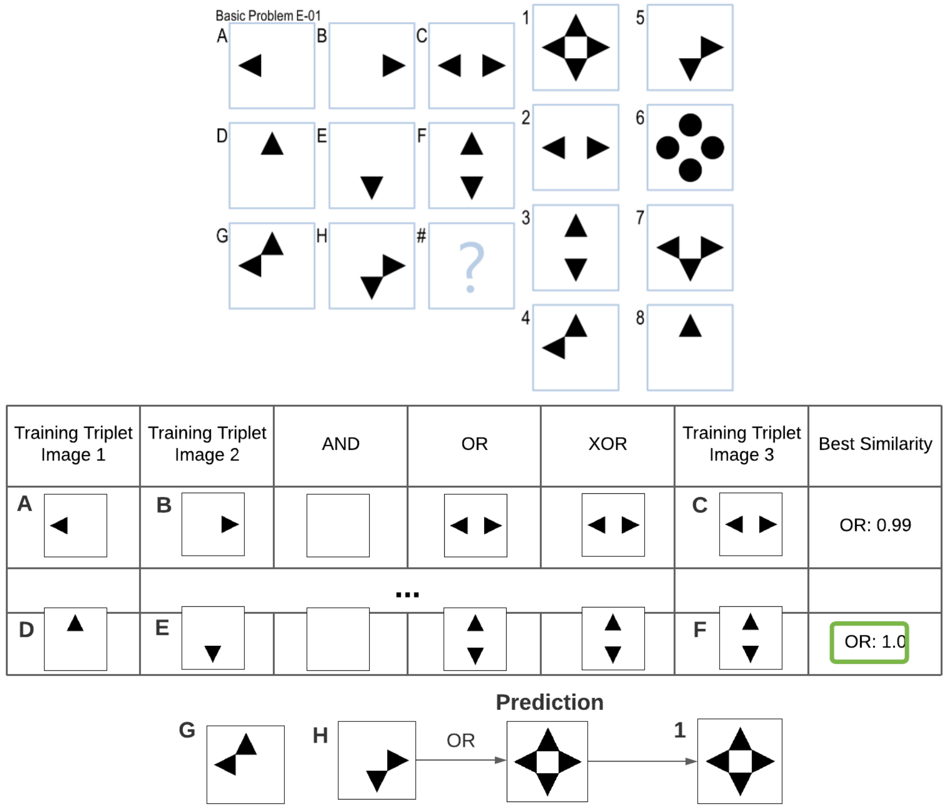
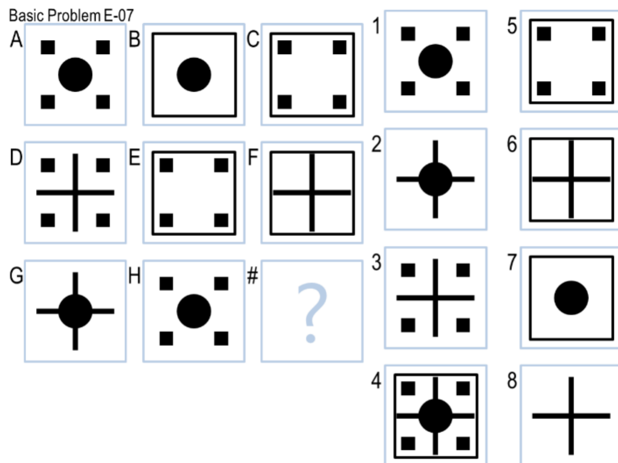


Figure 5 — A successful AND, OR, XOR operator problem for ASTI.

The reasoning process is as follows: First the agent tries ANDing image 1 (A) and image 2 (B) of the training triplet, it then compares the result to C yielding a similarity of 0.0 since the AND result of A and B is a blank image. The agent next tries OR and XOR, favoring OR if the XOR and OR similarity results are the same, as XOR is more involved and complicated in most cases for prediction. It then repeats this process for the training triplets seen in Figure 2, finding the OR operator of D and E to be the best. It then applies the OR operator to G and H, yielding a prediction that looks very close to answer choice 1.

3.3 DPR/IPR successful solve

The reasoning process for a successful DPR/IPR solve can be seen in Figure 6. First the agent tries ASTI, and the similarity is 0.7 from doing XOR of A and B to yield C. This however does not meet the agent's allowable ASTI threshold of 0.94 so the agent continues on to DPR/IPR. The agent iterates through the training pairs of Figure 3 calculating DPR/IPR values for each iteration, finding answer choice 3 to be the highest voted, with a vote value of 13.8. Here is where the second-best vote value mentioned in section 2.2 comes into play. The second-best vote value in this case is 12.63, yielding a vote ration of 0.91 which is allowable according to the agent, the cutoff being <0.93. The agent continues with the DPR/IPR selection and returns 3 as the correct answer for the problem.



Training Pair Image 1	Training Pair Image 2	DPR	Target imageDPR with 3	+Vote for 3	...	Target imageDPR with 8	+Vote for 8
A	B	-0.013	0.012	0	...	0.088	0
			...				
B	H	0.014	0.014	0.999	...	0.091	0

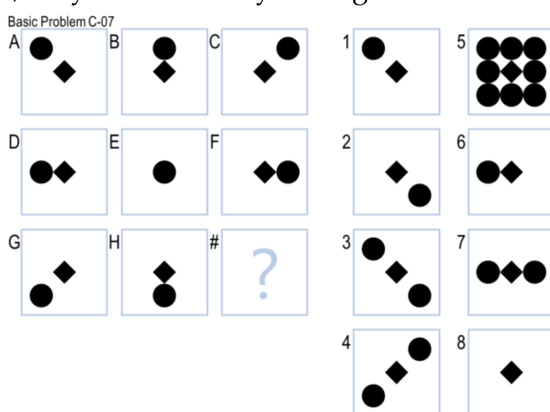
Figure 6 — A successful DPR solve reasoning process. The process here is repeated for IPR.

3.4 ASTI unsuccessful solve

The ASTI model fails sometimes like on problems seen in Appendix 7.1: An ASTI unsuccessful solve. The agent performs the identity transform on image B and compares the result to image H. It is a perfect match. The agent now thinks that it should identity transform image C. It does so and finds that a perfect match is available as one of the options, answer 4 which is incorrect. The correct answer is 6. This problem also tricks the agent's mentality to try DPR/IPR because the similarity scores are so high for the identity transform, so DPR/IPR is not run on this problem.

3.5 DPR/IPR unsuccessful solve

In Basic Problem C-07 seen in Figure 7, the ASTI model first thinks that rotating D 180 degrees is the best transform, to get F (which technically, it is correct, rotating D 180 degrees yields F). However, the ASTI model then rotates G 180 degrees which does not equal 2, the correct answer choice. The similarity score is not very close anyways for ASTI, so the agent forgoes ASTI and tries DPR/IPR. With DPR/IPR, the agent thinks that 6 is the correct answer from comparing D and F again. This seems to be a trap for both models, as they can not distinguish between that although the shapes and dark pixel ratios are equivalent between many combinations, only a select few yield a good answer.



Training Pair Image 1	Training Pair Image 2	DPR	Target imageDPR with 2	+Vote for 2	...	Target imageDPR with 6	+Vote for 6
D	F	0	0.001	0	...	0	1
...							
A	D	0	0	1	...	-0.001	0

Figure 7— A failed DPR/IPR problem for the agent.

4 OVERALL APPROACH

This agent's approach can generally characterized as, first try ASTI and if the certainty of the answer is high enough, no need to continue with models. If the certainty if not high enough, try DPR/IPR, and again if the certainty is not high enough, just guess 5. It would most closely align with the ever-expanding set of heuristics, as first ASTI was the only model developed, then DPR/IPR in addition to that when meeting tougher problem sets like D and E. Also, the addition of bitwise operators was a result of Basic Set E. This pattern suggests that this agent is more of a conglomerate of models rather than a highly tuned single one.

It was almost the case that I threw out ASTI completely and soley relied on DPR/IPR but many problems from basic set B and C were lending to ASTI rather than DPR/IPR.

5 HUMAN COMPARISON

This agent is similar to a human reasoning approach to solving these problems in that humans tend to look for rotation and image flip patterns when doing these tests. It is very common to see problems like Figure 4 and think "oh, I need to flip one of these to get the answer". The ASTI portion of the agent is most akin to that thinking. The iterative approach however in the agent is unlike a human's thinking. Humans don't tend to preset a certain number of flips/rotations and go through each one seeing if it matches an answer. It seems to be almost a jump to the best one with humans which is fascinating.

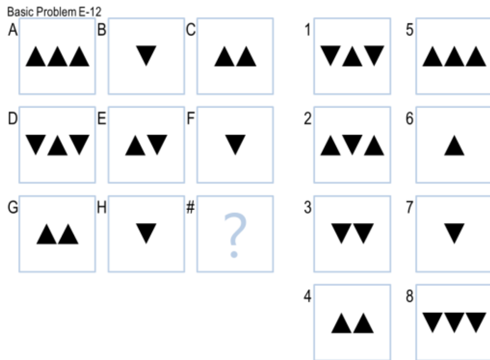
With DPR/IPR, I feel that it is very unlike a human's reasoning. Mainly because humans don't have that acute sense of pixel-counting that computers do. 100 pixels off for a human might not be noticeable but to the agent it would be enough to choose one model over another.

6 REFERENCES

1. Joyner, D., 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. *Udacity*.
2. Kunda, M., McGreggor, K., & Goel, A. K. (2013). A computational model for solving problems from the RPM intelligence test using iconic visual representations. *Cognitive Systems Research*.

7 APPENDICES

7.1 An ASTI unsuccessful solve



Training Pair Image 1	Transform	Union	Intersection	Similarity (Intersection / Union)	Training Pair Image 2
B	Identity			1.0	H