RPM Milestone 3:
CS7637

Josh Adams
jadams334@gatech.edu

Abstract— The RPM Project: Milestone 3 journal is a place to reflect on the obstacles faced while improving the agent's performance. In the second milestone I discussed the improvements I made to make an agent which would be able to solve the RPM problem. Some of those methods were utilized and others were just attempted.

1 CURRENT FUNCTIONALITY

My agent has been improved considerably from milestone 2 where it was able to barely get full credit for the assignment. To answering almost all problems for milestone 3 correctly.

1.1 How the Agent works

My agent works by progressively filtering out possible answer choices. Within the agent is a tiered filtering architecture which get progressively more complex, the further down the architecture a problem reaches. Within each filtering level it also gets progressively more specific until it either reaches an answer or runs through its various checks. At the top you have the most basic such as comparing the number of white and black pixels between images. At this filtering tier the agent will compare all images and look for consistencies and remove those answer which do not align. If an answer is found it will stop and return the answer, otherwise it goes into more specific filtering within that tier. Previously it compared all images, next it may check all images in a given row to all images in another row. It will then compare columns, then diagonals, then various conditional operations on the images between rows, columns, and diagonals, such as Row 1 may be an AND operation between Row 0 and Row 2.

If the problem continues it moves to singular image comparisons. Such as images above, below and beside the image, using various operations. An example would be looking at Basic Problem C-12. Image E is a cv2.bitwise_and operation between image D and B. Once all pixel-based filtering has finished, typically many of the possible answers have been removed but generally not enough to solve the

problem. The problem progresses to more complex filtering. Pixel filtering to union-based comparisons to dark pixel ratio(DPR) then inter-pixel ratio (IPR). Then comes translation-based filtering such as various rotations, or image flips. Then independent contour filtering, shape filtering.

The final tier is a multi-faceted approach. This approach uses IPR, DPR, contour detection, template matching, line, and edge detection. It uses contour detection to extract possible templates within an image. Since this is a multi-faceted approach many things will be compared before removing a possible answer choice. Template matching between problem images and possible answers, as well as line angles. Such as finding 90-degree lines in all problem images but certain possible answers, no 90-degree lines are being detected. This could be a problem with the parameters used for the line detection, which is why I combine many other techniques to be more precise in removal. If a problem has reached this tier, it typically has its possible answers reduced significantly and removal of a possible answer can be problematic as these are the most likely solutions to the problem.

2 AGENT PERFORMANCE AND EFFICIENCY

Overall, I believe my agents performance and efficiency is acceptable but far from optimal. My agent does struggle on some problems in terms of runtime. The reason is that as the agent progresses in the problem, more complex strategies are used. This typically happens when a problem has many small white areas throughout the images.

2.1 Agent Performance

My agent is able to meet the requirements of this milestone. I have made modifications which get all of the Basic correct but consequently, the agent misses one more in the Test set. Overall, my agent performs well but there is absolutely room for improvement. On problems which my agent answers incorrectly, it is because it returns to first element in its processed set of possible answers. My agent gets 11 out of 12 correct for the basic problems correct and 10 out of 12 for the test problems.

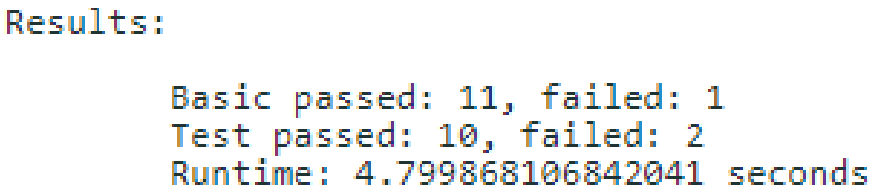

Figure 1 — Agent performance on milestone 3.

While my agent currently does not produce a correct answer on all problems, it is very successful in reducing possible answer choices. My agent is able to reduce the possible answer choices by half on every problem, most problems are reduced down to two or three possible answers. I am still working through methods to select the best answer choices when finished and more than one possible answer remains. Problems my agent performs well on are those which have large, consistent similarities between all images. An example of this is Basic Problem C-01. All of the images share a white square which is located in the center of each image. Each row also shares features with its adjacent images. Problems where my agent struggles would be those where it must begin comparing inverses of images. An example of this is Challenge Problem C-04. My agent is able to get the correct answer but is clearly spending much more time on that problem than others as shown in figure 3.

2.2 Agent Efficiency

In terms of efficiency, my agent relatively quick. It is able to solve each problem in less than half of a second. I definitely could make my agent much quicker by nearly 60% if I had a more selective setup process. Currently my agent does all processing needed by any problem within the setup function.

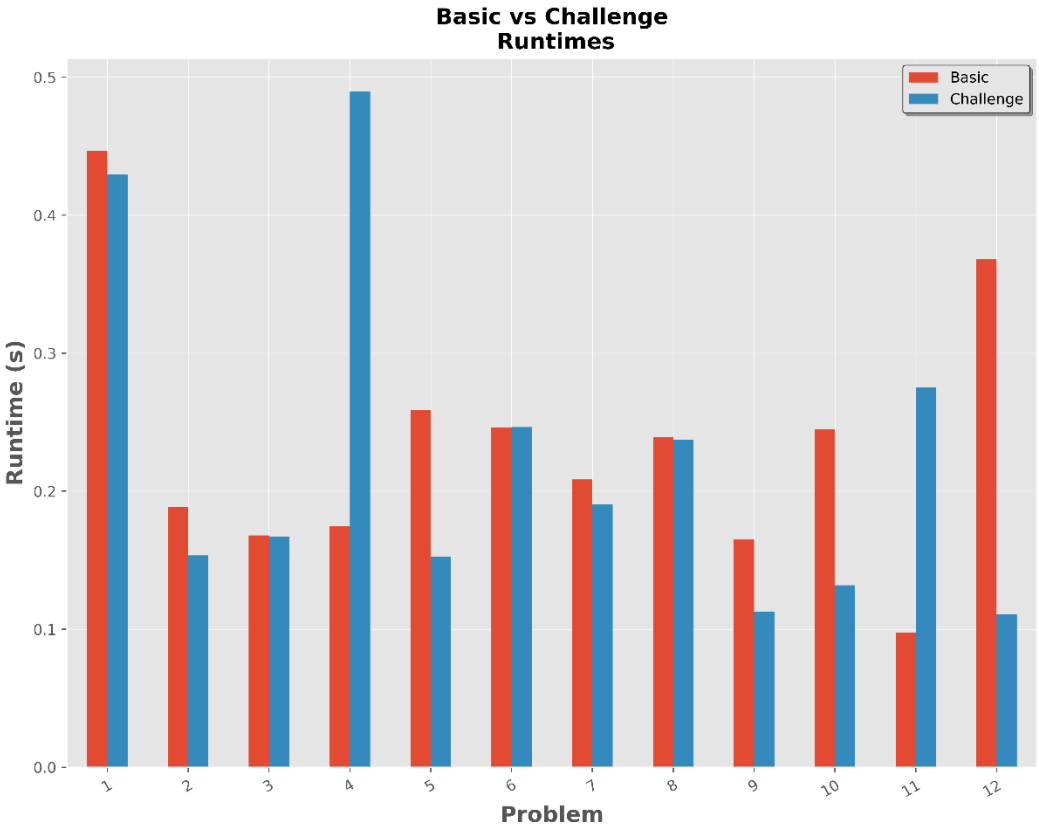

Figure 2 — Runtimes for Basic and Challenge problems required for milestone 3.

If I could determine the processing needed by a problem prior to the setup then I could selectively do only the needed processing. In most cases a very small subset of all the processing is actually needed for any given problem. My agent currently performs in worst case $O(N^2)$ but this is rarely the case and would only occur when my agent fails to remove any possible answer choices. My agent works by reducing possible answers and one a possible answer choice is removed, that answer choice is never evaluated again and this reduces the problem space.

3 IMPROVEMENT PLAN AND FEEDBACK

3.1 Improvement Plan

My plans to improve my agent consist of more rule-based filtering such as adjacent image augmentation. As wells comparisons between many more

combinations of images. For example, my agent will start to compare A and C with different operations, previously would only compare A and B or B and C.

3.2 Problems with future sets

As of right now I would expect many of the problems in the future sets to be difficult for my agent to answer correctly.

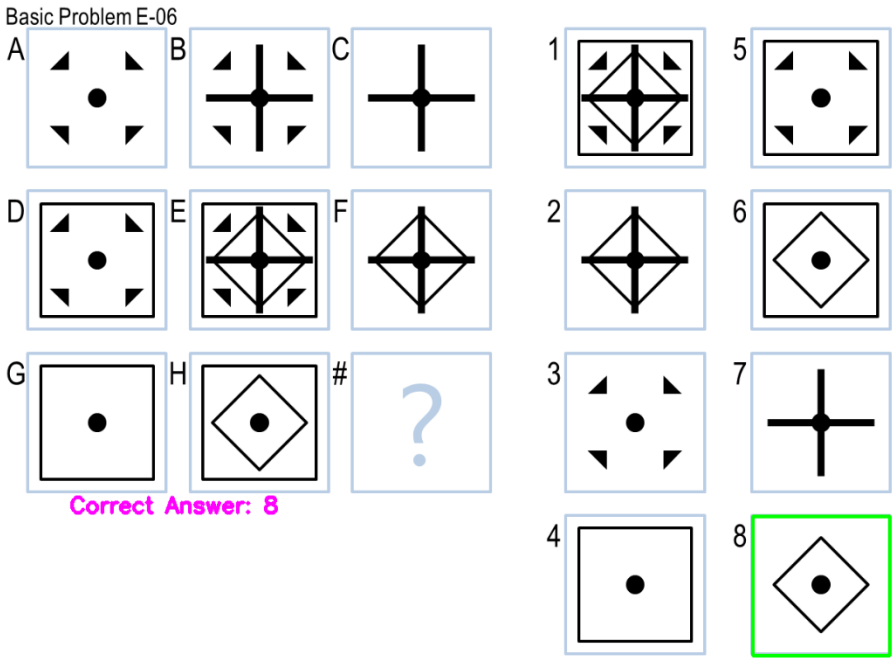

Figure 3—Seemingly difficult problem in future problem sets.
Source: My code generated the image but used images provided.

One of the reasons my agent would perform poorly on this problem is that my agent requires image relationships to be consistent between its neighbors. If image B was a continuation of A, then C would need to be a continuation of B. In this problem it could be seen as B is an cv.bitwise_and between A and C. This would be something which I will implement for the next milestone.

3.3 Requested Feedback

Feedback which I would benefit the most from would be better ways to solve these problems. I have problems selecting the correct answer from the reduced answer choices. My agent is able to reduce practically all problems to four or less

possible answers, depending on difficulty. I am not sure what other means I could use to solve these problems.

4 REFERENCES

1. OpenCv https://opencv.org/
2. IPR was described in Ed Discussion.
3. DPR was described in Ed Discussion.