Homework 4

Note: I had to comment out the spheremarker code in order for the simlation to run at a reasonable speed on my system.

Problem 2

For my ImgProcessingActionPredictor I used the same algorithm specified in the handout. I thresholded the grayscale img to identify the white line. Then I found the midpoint of the whiteline. If the whiteline midpoint index was greater than the (width of the robot camera/2) then turn right otherwise turn left. If the whiteline midpoint index with between (width of the robot camera/2)-(the width of the white-line/3) and (width of the robot camera/2)+(the width of the whiteline/3) the move forward. I found this limit through trial and error after trying to fix the left-right loop that would sometimes leave the robot stuck.
Performance of algorithm
maps/test/map1: 8/8 landmarks
maps/test/map2: 6/6 landmarks
note: the algorithm also achieved 15/15 landmarks on maps/test/map3 when I the left action was uncommented. However, I have commented this piece of code out just in case the unseen map would cause the algorithm to leave the robot in a left-right loop.

0.1 Extra Credit

I tried many different things to get the model to work for left turn maps as well. However, often times it would leave the robot stuck in a left-right loop even on test/map1 and test/map2 giving worse performance than my problem 3 model. I realized that the dataset collection was very important here to allow the robot to better distinguish left and right turns against moving forward. I then went on to move around and delete dataset images if they were too similar and only leave in the more extreme examples for turning left, right and moving forward (I did not collect data for moving backwards). However, still the performance of the model was not great even though the model had 85%++ accuracy on the validation set (sometimes reaching 92%++). Therefore, more work could be done in curating a more appropriate dataset.