Report of Question 4

Zekun Zhang (zz364)

zekunzhang.1996@gmail.com

Of course, I would like to choose the **SheepdogBot**. Because it really takes me so much time in the Question 1 and I won't like to control dogs again, and just make them control themselves.

The sheepdog bot should have the ability to 'feel' and 'react', just like 'eyes', 'ears' and 'legs'. So, what I need is cameras, microphones and wheels or mechanical legs. Moreover, I still need the 'dogs' to have a smart brain, then maybe GPS chip for collecting location and sensor-computer pairs for the humidity, light and temperature.

Governing and protecting the sheep are the major job for a sheepdog, so I can first model the tasks the sheepdog should finish.

1. Observe the sheep and circumstance. Identify the sheep belongs to it and make sure every sheep is in the flock. And identify the unsafe objects just like wolfs or thieves around the sheep.

2. Remember the location of its home and the meadow. Lead the sheep to the meadow and get them back at a regular time

3. Identify the unsafe objects like wolfs or thieves and protect the sheep by setting an alarm or making a call.

1. Observe the sheep and make sure every sheep is in the flock:

For this part, in my view it can be achieved by using a clustering algorithm. Just as the algorithm I used in the Final Question 2, I would use K-Nearest Neighbors (KNN) as a basic case.

1. Make sure that our knowledge base is the location of each sheep can be captured from the camera and collect all the locations information of the sheep.

2. Then calculate the location and transform it into a 2-dimensional coordinate.

3. Next calculate the Euclidean distances between each sheep and all other sheep and take a threshold K as the number of samples.

4. Comparing the results, find out the specific sheep with the largest distance value from others. I can first pre-define a threshold value as a range of the flock. If the distance of the specific sheep is out of range, it can be considered as the one detached from the flock, and the remaining work is what I've done in the Final Question 1 --- the bot needs to chase after it towards the flock and make it back.

2. Find way connected home and meadow:

This question can be converted to the shortest path problem or the Maze problem I've done in the Assignment 1. So, it's really clear that model the path first and then utilize an algorithm like A* to find the shortest path.

In the real situation, the location of meadow is changed time by time and the exact location during the shepherd time is also slightly changed due to the random move of the sheep. To make sure to escort the sheep to the meadow safely, the bot would take the shortest path to make the escorting process as short as possible, since it's really unsafe during this process. I may assume that there's a map with weight stored in the memory of the bot and the weight represents that the safe factor (more dangerous with a higher factor value) of different type of terrain. Maybe there's some place like gas station or some supply station, the weight shall be negative because it's much safer and would have some fuel or battery supply.

Then I can create a weighted graph with the information offered and next design our A* algorithm part.

For the A* algorithm, as we all know the function:

$$f(n) = g(n) + h(n)$$

where the f (n) is the estimated cost from the start to the destination by walking through Node n. g(n) is the cost taken from the start to the Node n and h(n) is the estimated cost of the best path from Node n to the destination. By adjusting the heuristic function, the path can be dynamically changed to achieve other purposes in path choosing.

3. How to identify an enemy:

For this part, what the bot should do is to identify the unsafe like a threatening human/animal. For example, when a wolf or a bear which is extremely dangerous is towards to the sheep and the bot should set an alarm or call the master and lead the sheep to fallback. So, the question is converted to a recognization problem. A well-trained Convolutional Neural Network (CNN) would be a possible method and worth to try. In this way, I can utilize the camera that can capture images in a real-time manner. Then it's a learning process and before the bot is allowed to work, first use the pictures of wolves or bears these dangerous animals to label them as danger. As for some animals which aren't such aggressive, label these animals as safe ones. Then pretrain the CNN model.

As far as I'm concerned, the most difficult part should be the process of learning human beings, because some people like the master or some workers or friends of the master would safe ones, however, the thieves should be labeled as dangerous ones. But the key problem is how to recognize these bad people.

In my views, it really needs a long way to go in this part, but I can design a simple one by recognizing some people who has stayed with the safe ones as safe. Just like a friend of the master, if he stays around the master for several times, then label this guy as a safe one.

Then go to the training part. Before the training process, the bot would take the picture of the face of a human and resize the picture to a standard size, which is the input of the convolutional neural network, and changed into a grayscale image, just as what I've done in the Assignment 4. Then regularize the value of each pixel to make the model more compatible and easier to converge. For safe labels, they should be processed as one class using a one-hot key from. The structure of the CNN may vary but mainly is made up of the input layer, convolutional layer, pooling layer, dropout layer, fully connected layer, and the output layer. Then the activation function can be chosen as Rectified Linear Unit (ReLU) for its outstanding performance in the convolutional computation and non-linearity. The optimizer can be an Adam Optimizer, which is based on the theory of Gradient Descent, but the difference is that Adam Optimizer can automatically adjust its learning rate, which is easier to get an ideal result with less human interference. A well-trained CNN model can tell the classification of any creature appeared in the camera, and the prediction result can let the bot know if this is a dangerous one or a harmless one.

Bonus:

1. After finishing the course, one thing in the Assignment 1 I would like to redo and improve in the Harden Maze part. For the first time, my team use the flip way to make the maze harder which for now I could consider it as a random way, because we just try to use the Genetic Algorithm to keep some valid flip. After I've learned Neural Network , it comes out that we could find out patterns of the hard maze and then it would much easier for us to detect the complexity of the maze. Then it would take us much less time to do this harden process. That's just an idea.

2. Actually, I don't remember quite clearly, is it on a superman-like costume?

3.