Computer Vision PS

We hope you enjoyed learning Computer Vision in the Winter School of AI & Robotics. Now, it's time to use all that you have learned. The complete Winter School Problem Statement comprises two tasks, which focus on showing how the software of a very simple autonomous vehicle is designed.

Task-1

This task comprises three parts:

1. In the first part, using the probability of 0.2~0.3, you have to generate a maze. The maze will be such that each pixel or group of pixels in the image will either be a black pixel or a white pixel. Now, for this, initially, make the image of a small size and then resize it to the required size so that instead of getting a pixelated image, we will get chunks of black pixels. Also, take care to mark two points on the maze with a grey-coloured pixel, one of which will be the starting point and the other will be the ending point. The black pixel is synonymous with an obstacle, and the white pixel is synonymous with an open path. After completing the maze generation, the task is to start at one of the grey points and traverse through the maze to the other grey point. This should be done using a variety of Path-finding Algorithms(A*, BFS, DFS, Dijkstra should be compulsorily used while using more path-finding Graph/Tree Algorithms will fetch more points). Finally, document the time taken for path-finding and the distance of path found for all the Algorithms for a variety of Randomly-generated images.
2. In the second part, there is a map provided to you(Attached in mail). The map has been pre-processed for you. It has a red pixel at (880, 145) and a green pixel at (359, 563). The red pixel is the start point, and the green pixel is the endpoint. You have to work on this map and get the shortest path from the start to the endpoint using the algorithms used in the first part of Task-1.
3. The third part is pretty straightforward. You have to get images of your locality from Google Maps. Set a start point and an endpoint manually and apply the algorithms after doing the required pre-processing.

Task-2

This task comprises two main parts and one bonus part:

1. In the first part, you have to generate a 1-2 minute long video using 12-15 (can be more depending on you) images. These images will be of traffic lights(it will be two images, with green light and red light), Stop signals, speed signals(2-3 different speeds), turn signals, etc. The mentioned ones are compulsory to use in generating the video. Along with it, use more images to complete the 12-15 images requirement.
2. In the second part, the video generated in the first part will be run and depending on the different road signals, the program will generate different outputs and store them in a text file. For example, let's say the output stores the speeds of the left and right motors, so upon showing the green signal, both the motors will keep running. If a red sign pops up, the motors will stop unless and until a new image of the green sign comes up. Some things have to be taken care of, like, if a red sign has come up, or a Stop signal has come up, then immediately after it, a green sign should come, or a Go signal should come.
3. This is the bonus part. It comprises three parts:
    a. The first part of the bonus task is a parallel implementation of video generation and storing the output in the text file.
    b. The second part is the hardware implementation of the whole thing. This means that the output produced in the text file should be read and sent to the Arduino, where it will drive the motors depending on the output coming to it.
    c. In the third part, implement everything in sync i.e. in parallel.

Submission:

These tasks are to be done in **teams of 1-3 members** i.e. the maximum team size can be 3, and the minimum team size can be 1. You will get a Google Form where you have to submit your team names and other details of the members of the team. One of the team members should make a folder on google drive and upload all your files. Necessary files include input images, codes, output images, etc. Try to organise stuff to make it easy for us to get your code up and running on our systems. Also, make a small document explaining how to get your code up and running. Upload this as well on google drive. Later on, you will get two Google Forms links, one on **9th April for mid-evaluations** and the other on **16th April for final submission**. Therein, you have to submit the link to your google drive.

Along with the code, you have to make a presentation documenting all of the functions, algorithms, e.t.c. that you have used in your code and why you used it. It should also contain photos and videos of your code's outputs. You will have to present it later on.

Timeline:

The timeline is as follows:

1. Problem Statement Release - 4th April
2. Mid-Evaluations - 10th April
3. Final Submissions - 17th April
4. Team Presentations - The week post 17th April

Note:

Both the tasks are **completely open-ended**. You are free to make any constructive changes to the problem statements. However, you will have to explain them in your presentations, mention them in a document briefly, and upload it in the drive.