

Motor AI Technical Challenge

Introduction:

Please read this section carefully. As per your preference, you have decided to apply for the open position in the perception team here at Motor AI. The challenge consists of one task which will be explained in detail below. If you have any questions or difficulties, please do not hesitate to reach out to us.

Some important points:

1. If any code is not self-developed but from a source such as stackoverflow, tensorflow, or pytorch, please include the link as a comment near the piece of code that you have copied.
2. The task can be completed without a GPU so we hope that you should not have any computational issues with that task. If you have a GPU you should be able to finish the task early.
3. Please follow the standard coding guidelines for the respective programming language (including variable naming, code readability, understandability, project structure, code optimization, reusability, etc.).
4. Prepare a concise but thorough report of your approach to each step in the task. Content to include in the report will be outlined in the task description but you can organize it as you like and add any extra information you would like.
5. Create a github private repository with all of your work and send the github link as the final submission. Share the repository with username **Porosh014** and **ArvindChandran**

A helpful tip would be to read the instructions carefully, cover all the points mentioned in the tasks and reach out to us in case you have any questions. :)

All the best and happy coding! :)

Task 1: Control the Mountain-Car-v0 Gym env using your webcam

OpenAI Gym environments are used to train Reinforcement Learning algorithms. If you have not worked with them before, you can read up on them in the following resources.

1. Open AI Gym Homepage: <https://gym.openai.com/>
2. The environment you will be using: <https://gym.openai.com/envs/MountainCar-v0/>
3. Installation and other documentation: <https://gym.openai.com/docs/>
4. The task description below includes general reinforcement learning terms such as environment, agent, rewards, states and actions. If you are not familiar with these terms, you will need to familiarize yourself with them before continuing.

OpenAI Gym is a lightweight library which you can easily install, and integrate with the task.

Your task is to control the Mountain-Car using your webcam in your laptop/PC. How you want to control the car is left up to you. You can choose what gestures and modeling choices you would like to use. Key steps to this task include the following:

1. You are not to use existing datasets or pre-trained models for hand/face/or any other type of gesture detection. This means we expect you to create your own dataset for training, validation, and testing. We will evaluate the design choices you make in creating the dataset so please include explanations on the points below within your report.
 - a. What gestures you used and why you chose those specific gestures
 - b. Explain the design strategies you used to create the dataset
 - c. What risks are you aware of in your dataset: where it might fail and why?
 - d. What were the difficulties you faced and how did you overcome them? (If there were any difficulties)
 - e. Describe the properties and statistics about the dataset
2. The next step in the task is to use the dataset to train a deep learning model which will give the output action based on the gesture you do in front of your webcam. This output action will be sent to the Mountain Car Env and it will return the new state and reward and the cycle continues until the episode ends. Some key points to help you here:
 - a. You are free to use pytorch/tensorflow.
 - b. Study the environment before creating the dataset and the model
 - c. Create the most optimized model possible as your inference needs to be real-time to keep up with the environment
 - d. You are allowed to use custom models or even pre-trained ones such as VGG/ResNets etc. which are available in Pytorch/Tensorflow and trained on ImageNet, but you should be able to explain your design choices.
3. Evaluation strategies/metrics used for both the model and the task.
4. Explain your code flow and the results within the report.
5. Evaluate your code in real-time and explain if your model works or not. If it works why it works, if it does not then why it doesn't.
6. Explain within the report, your failed approaches in brief, what you learned from them and how this led to a better model or dataset.
7. Submit the code and the documentation (**pdf or github readme**) on **github** by creating a **private repository** and share this repository with us. Share the repository with username **Porosh014**. Also, upload a video of your live webcam feed using the gestures to control the car, the mountain-car visualization and the action given to the environment at each timestep simultaneously. **Please record a single video with at least 5 consecutive successes of being able to control the car and explore all actions in the demo. It is okay if this video is shared via GoogleDrive/OneDrive.**
8. Even if you fail to successfully control the car, please submit the code, the documentation, and the video. Explain why it fails in the documentation.
9. Your video will not be shared with anyone other than the technical team and will be deleted once the interview process is over.

10. Just a reminder in case you forgot, **follow coding guidelines and prepare concise and in-depth documentation.**
11. **Please do not fake results or do not plagiarize as this will be apparent and unhelpful in getting to know your coding style :)**

Additional Task (Optional):

1. Introspection to check what the model looks at when it is predicting the action. You can write custom code but please spend only 4 hours max, or you can use the tf-keras-vis library. (<https://github.com/keisen/tf-keras-vis>)
2. Explain why you used a particular introspection model and interpret the results

Deadline:

The information about the deadline is included in the email.

Good Luck !