Obstacle Detection Using a Stereo Vision of a Car

M.S. Khatib[1], Saqib Saifee[2], Muzaffar Turak[3], Charandeep Singh[4], Nihar Meshram[4], Pratik Jambulkar[5]

[1]Associate Professor & HOD, Dept. of Computer Science and Engineering, Anjuman College of Engineering and Technology, Nagpur, Maharashtra, India. [2,3,4,5,6]Student of Graduation, Dept. of Computer Science and Engineering, Anjuman College of Engineering and Technology, Nagpur, Maharashtra, India.

---***---

Abstract - *We have created a Deep Learning Q-network (DQN) agent to build a self-driving car from raw sensory inputs. We evaluated our agent's performance against several standard agents in a simulation environment. Our results show that our agent is capable of successfully controlling a car to navigate inside a simulation environment.*

***Key Words*: Deep Q Learning, Agent, Sensory, Inputs, Simulation.**

1.INTRODUCTION

We know that Computer vision is one of the toughest problems in AI has been challenging researchers and engineers for many years. This problem of extracting useful information from images and videos finds application in an exceeding sort of fields like robotics, remote sensing, computer game, industrial automation, etc. The concept of where cars drive by themselves has gained immense popularity today during this new world, mainly motivated by the number of accidents that occur thanks to errors by drivers. Understanding the environment accurately and quickly is one of the foremost essential and challenging tasks for self-driving systems like self-driving cars. The self-driving supported deep reinforcement learning is that the most vital application of AI that has become a well-liked topic. Most of these autonomous methods concentrate on a way to directly learn the end-to-end self-driving control strategy from the raw data. In other words, this control strategy is considered a distinct sort of mapping between images and driving behavior, which usually faces an issue of low generalization ability. to boost the generalization ability for driving behavior, the reinforcement learning method requires extrinsic reward from the $64000 environment, which can damage the car. to get a stable and safe virtual simulation environment that will be constructed employing a different driving scene. A theoretically conceptualized model is established and analyzed within the virtual simulation environment, and it's trained by double Deep Q-network.

We will take revolution into the globe of AI and build our self-driving car. this is often visiting be a modeled version of a car (so it won't be driving on the streets of real cities) but still - it'll learn the way to drive itself. The keyword here is to learn because the car won't incline any rules on the way to operate within the environment beforehand - it'll figure everything out on its own. to attain this, we'll be using Deep Q-Learning. Deep Q-Learning is the result of combining Q-Learning with a synthetic Neural Network. The states of the environment are encoded by a vector which is passed as input into the Neural Network. Then the Neural Network will try and predict which action should be played, by returning as outputs a Q-value for each of the possible actions. In the end, the simplest action to play is chosen by taking the one that has the best Q-value.

2. Method and Material

- Reinforcement learning
- Bellman Equation
- Live Penalty Reinforcement
- The Induced Plan
- Markov Decision Process
- Addition of Live Penalty

3. Reinforcement learning: -

Imagine a robot (Also referred to as an agent) is trying to select up a pen, and it fails. It tries again, fails. After repeating this process several times, it finally gave a positive response. The agent has now learned the way to finish its task, this is often Reinforcement Learning, in short, it's lots like how living creatures learn.

The thing is we already understand how we, humans, learn. We understand the concepts of intrinsic and extrinsic rewards that guide us to become better at things, for example, if you're playing bowling - you recognize that you simply have to hit the pins and a strike could be a perfect shot. we all know this because we are rewarded with points for hitting the pins down and that we are "punished" with no points when your ball ends within the ditch. So, your brain projects those conditions of the environment onto your actions and that is how you recognize after you do good and when - not such a lot, and that is how we learn.

But how can we explain that to an AI?

We will specialize in a sort of reinforcement learning called Q-Learning.

We will derive this Self-Driving Car with PyTorch, a highly advanced Deep Learning & Artificial Intelligence platform.

A. Proposed System

First, we will build the environment containing the map, the car and all the features that go with it.

Then, we will build the AI, which will be the Deep Q-Learning model.

And eventually, we will have an exciting demo.

We will implement the Self-Driving Car with PyTorch, a highly advanced Deep Learning & Artificial Intelligence platform.

B. Implementation and Discussion

The proposed system will be implemented as follows:

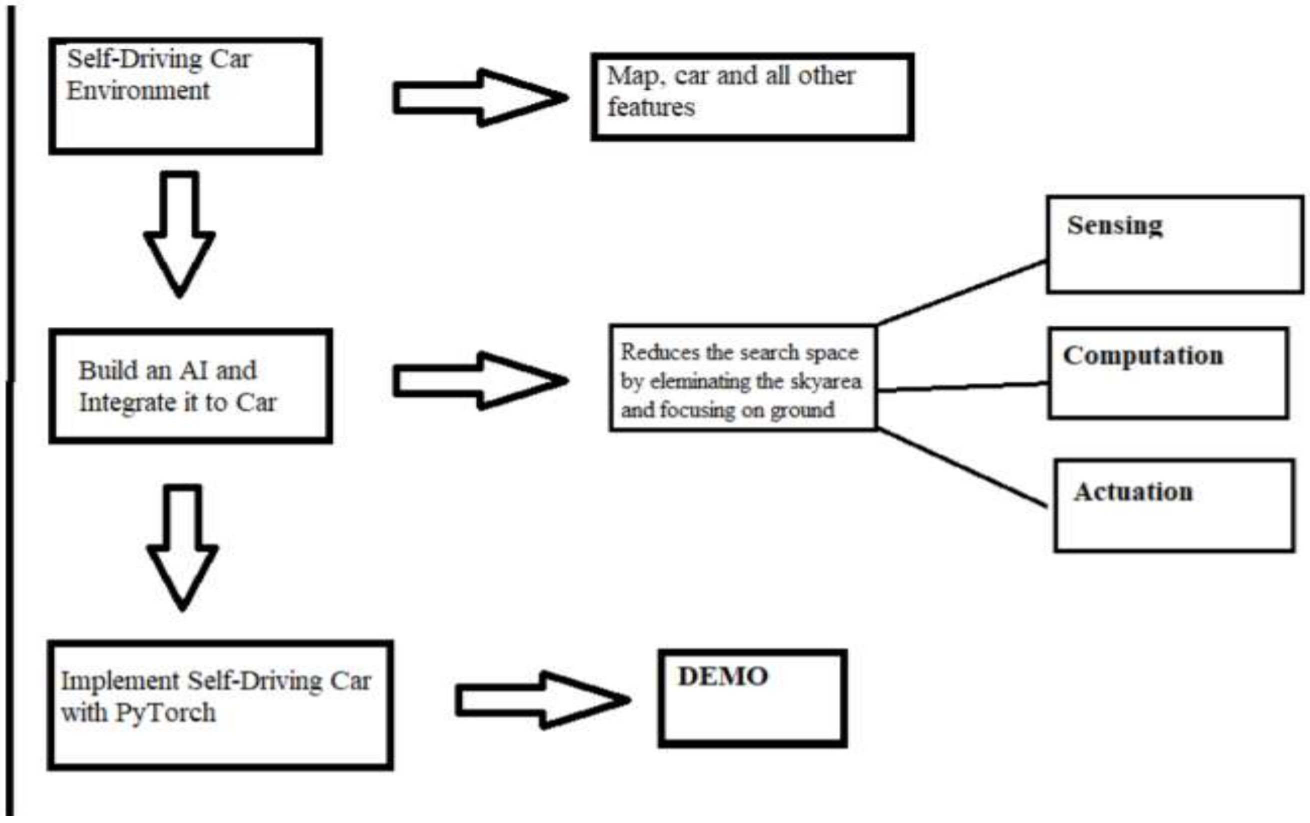
Self-Driving Car Environment
Map, car and all other features
Sensing
Build an AI and Integrate it to Car
Reduces the search space by eleminating the skyarea and focusing on ground
Computation
Actuation
Implement Self-Driving Car with PyTorch
DEMO

Figure 1: Flow of the work

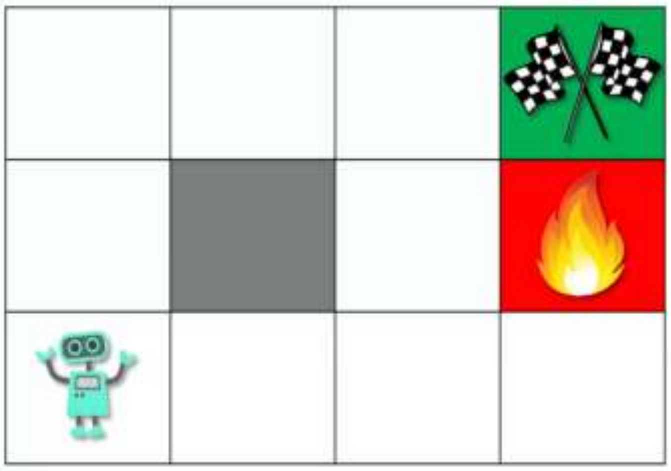

Figure 2: An environment where the agent or in our case a car is going to be trained

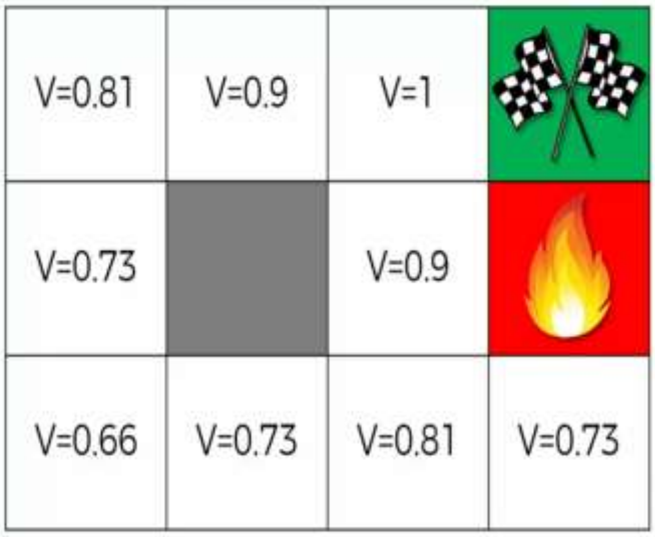

Figure 3: Q-values Demonstration

4. CONCLUSIONS

Throughout this project, we aim to import the use of reinforced learning to make a car learn how to detect obstacles in its path and during its journey learn the most optimal way to reach its destination. The learning will be punishment and reward-based learning where the car will be punished for each wrong step taken.

REFERENCES

[1] Python Cookbook: Recipes for Mastering Python 3 **Book By** Brian K. Jones and David M. Beazley

[2] To Complete Python Course: Beginner to Advance. **Course By**: Kawser Hamid, Pentester

[3] Artificial Intelligence Foundation – 101. **Course By:** Aditya Sharma.

[4] Reinforcement Learning I: Introduction. **Paper By:** Richard S. Sutton and Andrew G. Barto.