ISSN 2347 - 3983
Volume 10. No.4, April 2022
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter011042022.pdf
https://doi.org/10.30534/ijeter/2022/011042022

A brief study of operation autonomous vehicles

Ananth Raviraj Udupa, Kanisetty Venkata Hanush,Govarhdan.M, Ashik Kalagodu,V. Chayapathi.
RV College of Engineering, Bangalore, India

Received Date : March 1, 2022 Accepted Date : March 20, 2022 Published Date : April 07, 2022

ABSTRACT

Autonomous vehicles are also commonly known as self-driving vehicles, which do not require human involvement in order to operate or control themselves. In recent years, advancement in automotive thinking have improved but still require human input, depending on the degree of automation[1]. Experts expect vehicles to be able to drive themselves within 3-7 years. This paper describes the current state, the latest trends and the study of self-driving cars in the automotive industry[2]. A detailed analysis of the technology used by automotive vehicles to feel their position and the degree of automation in such vehicles is also included. As the widespread acceptance of self-driving vehicles is considered inevitable, so the need for specific technical and legal guidelines will be essential for a safe and conflict-free journey. Potential concerns about autonomous vehicles should be discarded with safe policies and technologies as discussed in the paper[3].

Key words: Autonomous Car, Artificial intelligence, Geo fence, Sensor technologies

1.INTRODUCTION

Autonomous vehicles: An autonomous vehicle is a vehicle that is capable of sensing its environment and moving safely with little or no human input[4].

2.LEVELSOF AUTONOMOUS VEHICLES

2.1 Level 0: The automated system issues warnings and may momentarily intervene but has no sustained vehicle control.

2.2 Level 1: The driver and the automated system share control of the vehicle. Examples are cruise control, parking assistance, lane-keeping assistance, etc.

2.3 Level 2: The automated system takes full control of the vehicle: accelerating, braking, and steering. The driver must monitor the driving and be prepared to intervene immediately at any time if the automated system fails to respond properly[5].

2.4 Level 3: The driver can safely turn their attention away from the driving tasks. The vehicle will handle situations that call for an immediate response, like emergency braking. The driver must still be prepared to intervene within some limited time, specified by the manufacturer when called upon by the vehicle to do so.

2.5 Level 4: As level 3, but no driver attention is ever required for safety, e.g., the driver may safely go to sleep or leave the driver's seat. However, self-driving is supported only in limited spatial areas (geofenced) or under special circumstances. An example would be a robotic taxi or a robotic delivery service that covers selected locations in an area, at a specific time, and in quantities.

2.6 Level 5: No human intervention is required at all. An example would be a robotic vehicle that works on all kinds of surfaces, all over the world, all year round, in all weather conditions[6].

3.AUTONOMOUSCAR

Autonomous driving is one of the key application areas of artificial intelligence (AI). Sensors in the vehicle generate a massive amount of data. To make sense of the data produced by these sensors,AVs need supercomputer-like, nearly instant processing capabilities. Companies developing AV systems rely heavily on AI, in the form of machine learning and deep learning, to process the vast amount of data efficiently and to train and validate their autonomous driving systems[7].

Artificial Intelligence seeks to create machines that seem to or have human intelligence. This includes broad, or general AI, and narrow AI, where the focus is on one task.

Machine Learning is a subset of artificial intelligence where we seek to give systems the ability to automatically learn and improve from experience without being explicitly programmed.

In machine learning, there are mainly three types. These are reinforcement learning, supervised learning, and unsupervised learning.

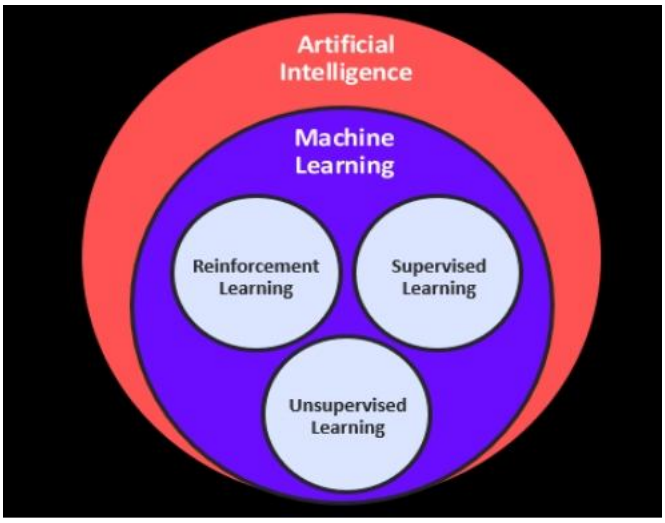

Figure 1: Shows Branches of artificial intelligence

Reinforcement learning (RL) is an area of machine learning concerned with how intelligent agents ought to take actions in an environment to maximize the notion of cumulative reward.

Reinforcement learning has been applied successfully in several practical use cases some recent examples include autonomous cars, fleet logistics, financial trading, data center cooling, hatchback systems, and game development. Games being a closed environment with a discrete set of possible actions make for a great research platform for RL, you have probably heard of deep mine with Alpha Go but, in the world's, best Go player an open AI managing to hold its own in a multiplayer drag competition with some of the world's best players both of these successes were achieved using reinforcement learning.

Supervised learning is machine learning which uses labeled datasets to train algorithms to classify data or predict outcomes accurately. Supervised learning requires a large amount of curated training data consisting of label and data pairs to learn models learn to accomplish a well-defined single task and don't scale easily to other tasks or subtasks without new data[8].

Unsupervised learning uses machine learning algorithms to analyze and cluster unlabeled datasets. These algorithms discover hidden patterns or data groupings without the need for human intervention. Its ability to discover similarities and differences in information makes it the ideal solution for exploratory data analysis, cross-selling strategies, customer segmentation, and image recognition.

4.AUTONOMOUS CAR MODEL

4.1 COMPONENTS PRESENT IN THE MODEL

CAR	18^{th}scale 4WD with monster truck chassis
CPU	Intel Atom Processor
MEMORY	4 GB RAM
STORAGE	32 GB (expandable)
WI-FI	802.11ac
CAMERA	4 MP camera with MJPEG
DRIVE BATTERY	1000mAh lithium polymer
COMPUTE BATTERY	13600 mAh USB-C
SENSORS	Integrated accelerometer and gyroscope
PORTS	4x USB-A, 1x USB-C, 1x Micro-USB, 1x HDMI
SOFTWARE	Ubuntu OS 16.04.3 LTS, Intel Open VINO toolkit, ROS Kinetic

5.TERMS RELATED TO THEM ODEL

5.1Agent: A piece of software that acts autonomously in a given environment to reach a specified goal in a given direction[9].

5.2Environment: The environment with which our agent interacts.

5.3State: The current state of the environment that is visible, or known, to the agent and upon which it needs to act.

5.4 Action: Given the current state of the agent, it needs to take any action to try and achieve its goal. Action is taken based on exploring, or exploiting what the agent has learned.

5.5 Reward: If the chosen action gets the agent closer to the goal, then the agent gets reinforced with a positive point, else it gets a negative point.

5.6 Episode: Each iteration where an agent goes from the start position to a termination state (crashes off track or finishes track)

The model uses reinforcement learning to train to self-drive around the track. An important part of reinforcement learning is the reward function. A reward function is a function, which helps the agent to determine if the action it just took was good or bad, and how good or bad, based on the outcome of the action. An example of a reward function is as follows

Reward function

```
def reward_function(params):
    '''
    Example of rewarding the agent to stay inside the two borders of the track
    '''

    # Read input parameters
    all_wheels_on_track = params['all_wheels_on_track']
    distance_from_center = params['distance_from_center']
    track_width = params['track_width']

    # Give a very low reward by default
    reward = 1e-3

    # Give a high reward if no wheels go off the track and
    # the car is somewhere in between the track borders
    if all_wheels_on_track and (0.5*track_width - distance_from_center) >= 0.05:
        reward = 1.0

    # Always return a float value
    return float(reward)
```

Figure 2 : Shows Reward function code

The above reward function reinforces the agent if all the wheels of the model are within the track and penalize it if the wheels are off track.

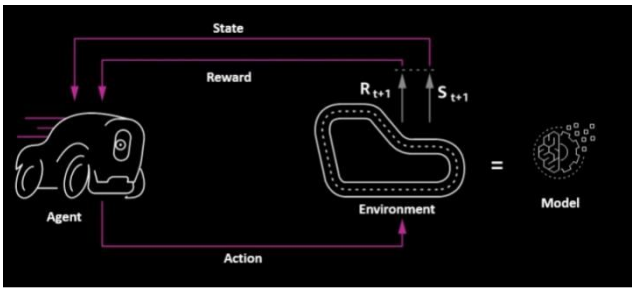

Figure 3: Shows Agent action and reward procedure

The learning of the model begins with the exploration of the environment. The model explores the track until it moves out of bounds or reaches the destination before it starts again. It is learning through iteration. As it drives around, the vehicle accumulates rewards for the action it takes. Best actions lead to positive rewards and bad actions lead to negative rewards. It records each step (state it started in, the action it took, the reward is received, and the next state) in memory. All steps from start to go off track or the finish is called an Episode. As the agent gains more and more experience through the exploration and iteration, it starts learning where it repeatedly gets higher rewards. It then starts exploring less and moving on to exploiting what it has learned. Convergence happens when a model starts repeatedly picking specific actions depending on the state it is in. As the model continues training the actions from each state don't change any more. The model is optimizing for expected cumulative return and model performance will be the same repeatedly, with subsequent updates to the model not changing the model behavior. There is a trade-off between exploration and exploitation, which you determine with a hyper parameter. If it explores too much your model may take a very long time to converge. If it exploits too soon, your model may not find the best driving behavior and potentially also fail to converge. Hence the hyper parameters have to be optimized to get the best model. Then the model repeats the training to drive on its own the entire track.

RL ALGORITHMS: **Proximal policy optimization (PPO)**

The proximal policy optimization involves collecting a small batch of experiences interacting with the environment and using that batch to update its decision-making policy.

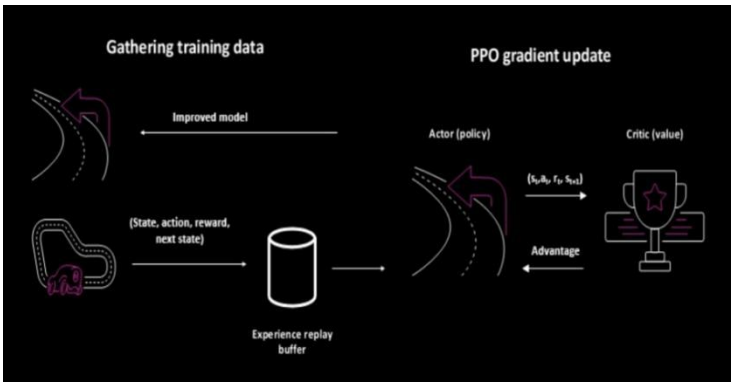

Figure 4 : Shows Proximal policy optimization

On the left, we have our simulator using the latest policy (model) to get new experiences.

The experience is fed into an experience replay buffer which feeds our Proximal Policy Optimization algorithm once we have the set number of episodes. On the right, we update our model using PPO.

While PPO is a policy optimization method, it uses an actor-critic method to learn.

Actor Updates Policy using Clipped updates: - Compute PPO gradient and shift policy in the direction where we get the highest value. We clip updates to prevent the policy from being updated too much, a common issue with policy optimization methods. Typically, we keep the ratio of new and old policy in [0.8, 1.2].

The critic tells the actor how good the action taken was, and how the actor should adjust its network.

Once the policy is updated the new model is sent to get more experience. Hyperparameters that feed into the reinforcement learning algorithm. They control various aspects of the algorithm, such as how quickly it should learn, how often we should train the network.

The network is the "model" that when we give it an image or state as input, will return the action that should maximize the cumulative rewards from all future states? Weights are the "coefficients" in our neural network that help determine where we "look" in each picture to select the action that will maximize our expected cumulative rewards.

Some PPO Parameters are

Learning rate: - Controls how big the updates are to your network weights. If you're learning rate is big the model will train fast but may not converge

Batch: - When we update the network, we don't typically use all the experience at once. We carve up experience into batch sizes and use each batch in turn to update the weights. Thus, the network is updated one batch at a time.

Epochs: - 1 epoch means update the network only once, which is run through all batches once. 2 epochs mean run through all batches twice, so will train through all batches, and then retrain through all batches

Discount factor: - This specifies how much future rewards contribute to the expected reward. The larger the discount factor the farther out the rewards that the model will consider, and slow down the training. With a discount of 0.9, the vehicle includes rewards from an order of 10 future steps. With a discount of 0.999, the vehicle considers rewards from an order of 1000 future steps. We recommend 0.99, 0.999, and 0.9999

Episodes between training: - Specifies how much experience to obtain before training the model.

Graphical representation of reinforcement learning of the model

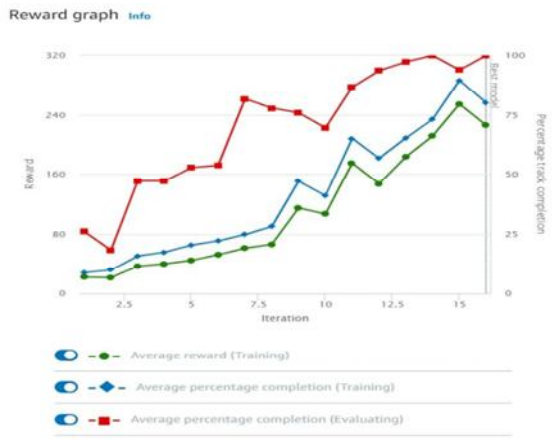

Figure 5 Shows Graphical representation of reward points of the agent

In the graph as we move from left to right the time of training increases.

From the graph, it is clear that the model at the beginning is exploring the environment; hence it doesn't complete the track. Also, the rewards gained by the model are less. This part of learning is called Exploration where the model explores the track. Here it takes an action that results in a higher reward. Finally, it explores the entire track and completes the track without getting out. Then it tries to exploit the environment to optimize the rewards and hyper parameters of the model. Hence the model trains to self-drive through reinforcement learning.

6.SOFTWAREARCHITECTURE

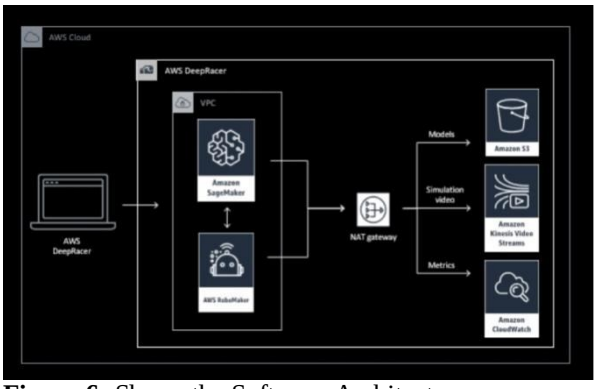

Figure 6 : Shows the Software Architecture

The model uses Amazon Sage Maker to train the RL models, AWS Robemaker's to provide the simulation environment, Amazon S3 to store models, Amazon Cloud Watch to store logs, and Amazon Kinesis Video Stream to display the video in the console. The service is built on top of other AWS services.

After the training of the model, it is evaluated and tested. If it passes the test then the model has been successfully trained toself-drive. Else it is trained with a new set of reward functions and hyper parameters.

Simulation-to-real domain transfer

After the model is trained successfully in the simulators, its data is transferred to the real model to self-drive on a physical track.

Challenges faced during sim-to-real domain transfer are that the model is trained using simulated images, which is a bit different from real images. Hence the model might get confused.

7. ABBREVIATION

AI: artificial intelligence, AV: autonomous vehicle, AWS: amazon web service, PPO: proximal policy optimization, RAM: random access memory, RL: reinforcement learning, USB: Universal serial bus

8.CONCLUSION

An autonomous vehicle is one use of artificial intelligence technology that has a huge impact on current and future society. As the use of AI will vary greatly, and it is widely available, there will be a great need to understand the different aspects of the decision-making process. This means that to accurately draft the context of each unique decision the technology must be taken at a real value and create an indirect relationship model of segmentation ideas. That is to say, we should not try to get the general idea that one size fits all of the intelligence-making apps as no single framework can adequately represent the many possibilities of machine decision applications. The formation of the mind is an important part of the epistemological process that underpins governance and control. Given the nature of the strategy focused on a series of definitions that promote public debate, such a framework must be open to investigation. Overall, this paper examines how a key conceptual process has received little attention in academics in the area of expected governance. The main argument is that the conceptual framework has the following implications in terms of debates about the management of automotive vehicles. It is a nuanced argument because the concept of a single concept is not set for each session. Instead, the acceptance of the limitations of the current debate is acknowledged, accompanied by calls for more precision in the construction of those structures, and perhaps strangely, to explore the number of ways to use the mind in the construction of the mind. It is a natural self-deception to emphasize a complete and final set of ideas. Instead, the conceptual framework should be a more flexible practice with a recurring component, as it is not just a matter of the conceptual accuracy used, but rather the fulfillment of the impact of that important framework. Since both security arguments and debates on the ethics of AVs present the discourse in a certain way, there is a need to revisit the original draft and adopt a mixed view in terms of that conceptual framework. Such is the complexity and intensification of debates about the social impact of AV to such an extent that there is a tendency to reverse the debate for greater use. Although this is common in any field, especially in the area of emerging technologies. For example, the debates about the risks posed by nanotechnology show similar weaknesses regarding the original conceptual framework, and the consequences have been controlling failures and widespread confusion among participants.

REFERENCES

[1]AWS deep racer developer guide.
[2]M. Weber, “Where to? A History of Autonomous Vehicles,” 2014. [Online]. Available: http://www.computerhistory.org/atchm/where-to-ahistory-of-autonomous-vehicles/
[3]“Details - google car,” http://googlecarjames.weebly.com/details.html, (Visited on 09/08/2016)
[4]P. Svec, A. Thakur, E. Raboin, B. C. Shah, and S. K. Gupta, “Target ˇ following with motion prediction for unmanned surface vehicle operating in cluttered environments,” Autonomous Robots, vol. 36, no. 4, pp. 383–405, apr 2013.
[5] P. Jaroszek and M. Trojnacki, “Localization of the Wheeled Mobile Robot Based on Multi-Sensor Data Fusion,” Journal of Automation, Mobile Robotics & Intelligent Systems, vol. 9, no. 3, pp. 73–84, jul 2015.
[6]“Official google blog: Green lights for our self-driving vehicle prototypes,” https://googleblog.blogspot.nl/2015/05/self-driving-vehicleprototypes-on-road.html, (Visited on 02/16/2016).
[7]Cardew, K. H. F. "The Automatic Steering of Vehicles: An Experimental System Fitted to a DS 19 Citroen Car." RRL report; LR 340 (1970).
[8]Jacqueline LeMoigne, and Phillip Veatch. "Vision-based navigation: A status report." In Proceedings of Image Understanding Workshop, pp. 153-169. 1987
[9]Shladover, Steven E. "Lane assist systems for bus rapid transit, Volume I:Technology Assessment." Publication RTA 65A0160, US Department of Transportation, Washington, DC (2007)