A Review Paper on Computer Architecture for Autonomous Driving

Mrinal Paliwal

SOEIT, Sanskriti University, Mathura, Uttar Pradesh, India

Email Id- mrinalpaliwal.cse@sanskriti.edu.in

ABSTRACT: Idea mapping is a graphical learning approach that can be useful for concept linkage and organising in the classroom. Concept maps are methods for organising and displaying information that give an elegant, readily understood representation of an expert's domain knowledge. These techniques have been utilised in educational settings to improve the connections between theory and practise, as well as between other topics taught in a course. They also assist students in forming connections between prior knowledge and newly given concepts, promoting meaningful rather than mechanical learning. The general interconnections between hardware, computer fundamentals, computer operations, and so on used to be simple and clear enough to comprehend computer systems. Nonetheless, Morden computer technologies have gotten increasingly sophisticated, making understanding the entire computer system extremely challenging. This research examines the value of idea maps in a Computer Architecture and Organization course (CAO). Students were encouraged to create idea maps over the course of a semester, which they were then permitted to utilise when preparing for their final test. The students' exam performance was then assessed, as well as their views regarding the course, the concept maps, and the questions on them. The findings led to the conclusion that concept maps not only contributed to the students' overall performance in the course, but also helped them prepare for their exams.

KEYWORD: Architecture, Autonomous Driving, Computer, Design, Vehicle.

1. INTRODUCTION

Computer architecture is concerned with balancing a computer quality of the results, efficiency, cost, and dependability. The ratio of these conflicting forces may be shown using the example of code set architecture. Because a single instruction can convey some higher-level abstraction, more complicated instruction sets let programmers to design more space-efficient programmes (such as the x86 Loop instruction). Longer and more complicated instructions, on the other hand, take longer for the CPU to decode and might be more expensive to implement. When instructions interact in unanticipated ways, the greater complexity of a big instruction set also generates more opportunity for unreliability[1].

In fact, the results of the surveys show that the actual planning of the design maps enhanced their achievement even more. The students had no idea that the questions they had answered quickly (and mostly correctly) were actually questions about concept maps[2]. As a result, one of the study's goals was met: strengthening the links between ideas and reducing conceptual uncertainty, as well as accurately acquiring concepts and knowledge with content maps. They also assist students in forming connections between prior knowledge and newly given concepts, promoting meaningful rather than mechanical learning. The general interconnections between hardware, computer fundamentals, computer operations, and so on used to be simple and clear enough to comprehend computer systems[3].

An autonomous vehicle must be able to sense its surroundings and navigate safely without the assistance of humans. Indeed, the National Highway Traffic Safety Administration (NHTSA) of the United States Department of Transportation has legally established five categories of autonomous driving[4].

- Level 0:

The operator maintains total control of the car at all times; the transport is not at all independent.

- Level 1:

Semi-autonomous; the driver controls most operations, but the vehicle can do certain functions autonomously, such as stopping.

- Level 2:

The driver is physically removed from driving the car by not touching the steering wheel or foot pedals. At least two functions, cruise control and lane centring, are automated because of this.

- Level 3:

A driver may still entirely transfer safety-critical duties to the vehicles and is not needed to monitor the scenario as closely as the reduced ranks.

- Level 4:

During the whole trip, the vehicle performs all safety-critical duties, and the driver is not expected to manage the car at any moment since the vehicle would control itself.

Levels 3 and 4 autonomous cars must use a variety of sensors to perceive their environment, including LiDAR, GPS, IMU, cameras, and others. They must be able to locate themselves based on sensor inputs and make real-time judgments about how to travel within the observed environment. Autonomous driving poses exceptionally high demands on processing resources and electrical power consumption because to the massive volume of sensor data and the high complexity of the calculation pipeline[5]. Existing designs frequently call for numerous servers, each with multiple high-end CPUs and GPUs, to be installed in an autonomous vehicle. These ideas have a number of flaws. For starters, the expenses are too expensive, making autonomy unattainable to the average people. Second, because this configuration uses thousands of Watts, power supply and heat dissipation become an issue, putting significant demands on the vehicle's power system[6].

1.1 Tasks in Autonomous Driving:

Automatic driving is a complicated system that entails a variety of responsibilities. As illustrated in Figure 1, multiple real systems, including sensor processing, perception, localization, planning, and control, must work together to achieve autonomous operation in metropolitan environments with unpredictable traffic. It is worth noting that most effective autonomous driving solutions rely primarily on Laser for mapping, localization, and obstacle avoidance, with additional sensors utilised for peripheral purposes[7].

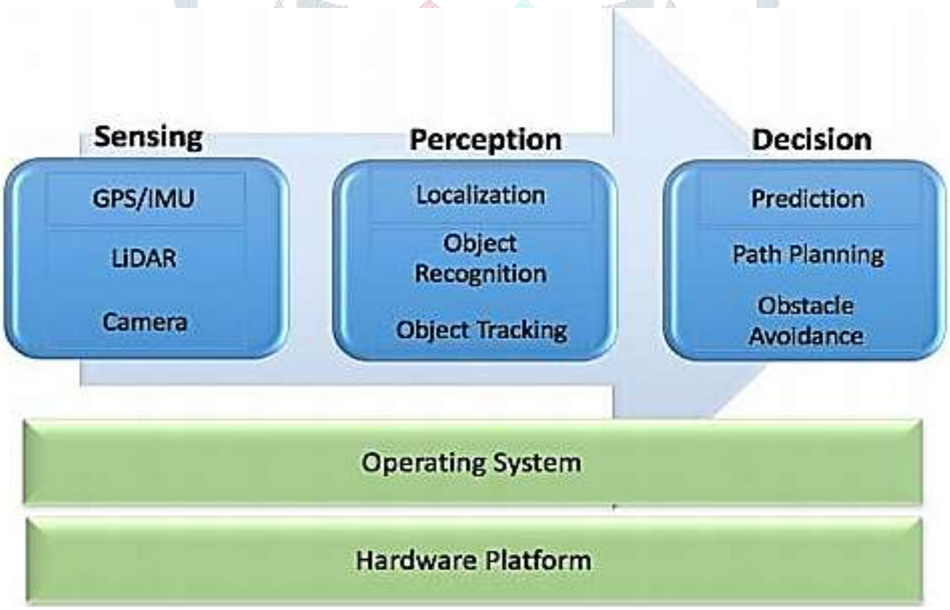

Figure 1: The above figure shows the Autonomous vehicle consisting of three main stages[8].

1.1.1 Sensing:

A self-driving car usually has many primary sensors. Indeed, because each type of sensor has advantages and disadvantages, data from numerous sensors must be integrated in autonomous cars for greater dependability and safety[9].

They may consist of the following:

1.1.1.1 GPS and Inertial Measurement Unit (IMU):

By transmitting both inertial updates and a global location estimate at a rapid pace, the GPS/IMU system aids the autonomous vehicle in locating itself. Although GPS is a reasonably precise localization sensor, its update rate is sluggish (about 10 Hz); therefore, it can provide real-time updates. An IMU's precision, on the other hand, decreases with time and hence can be depended on to give accurate position updates over lengthy periods. An IMU, on the other hand, may give updates at or above 200 Hz to meet the real-time requirement.

In the case of a car driving at 60 km / hr, the distance travelled between two position updates is less than 0.2 metres[10].

1.1.1.2 LiDAR:

For mapping, localization, and obstacle avoidance, LiDAR is employed. It operates by reflecting a laser beam off objects and measuring the time, it takes the reflection to determine distance. It is utilised as the primary sensor in most autonomous vehicle applications due of its excellent precision. LiDAR may be used to create high maps and to pinpoint the location of a moving vehicle. To identify obstructions ahead, etc. versus high-definition maps.

A LiDAR device, such as the Velodyne 64-beam laser, spins at 10 Hz and captures around 1.3 million readings per second in normal operation.

The difficulty with LiDAR is that it has two major flaws:

- Measurements may be highly noisy when there are many suspended particles in the air, such as rain droplets and dust.
- A 64-beam LiDAR device is quite expensive.

1.1.1.3 Camera:

Cameras are commonly used for lanes detection, traffic signal detection, and pedestrian identification, among other duties. Existing systems often attach eight or more 1080p cameras around the car to detect, recognise, and track things in front of, behind, and on both sides of the vehicle to improve autonomous vehicle safety. These cameras typically operate at 60 frames per second, generating about 1.8 GB of raw data per second when combined.

1.1.1.4 Radar and Sonar:

In obstacle avoidance, radar and sonar systems are often utilised as the final line of defence. The distance to the nearest item in front of the vehicle's route is calculated using radar and sonar data. When we detect that an item is approaching from behind and that there is a risk of collision, the autonomous car should apply the brakes or turn to avoid the impediment. As a result, the data provided by radar and sonar requires little processing and is typically sent straight to the control processor, bypassing the main computing pipeline, in order to execute such "urgent" operations as swerving, applying the brakes, or pre-tensioning the seatbelts[11].

1.1.2 Perception:

After gathering sensor data, we input it into the perception step to gain a better understanding of the vehicle's surroundings. Localization, object detection, and object tracking are the three primary challenges of autonomous driving perception.

1.1.2.1 Localization:

A high infrared reflectance ground map may be generated using GPS/IMU and LiDAR data as part of the localization process. We might use a particle filter technique to link the LiDAR readings with the map to locate a moving vehicle relative to these maps. The particle filter technique has been shown to be successful in urban areas and to accomplish real-time localisation with 10-centimeter precision. The expensive expense of LiDAR, on the other hand, may limit its widespread use.

1.1.2.2 Object Detection:

However, in recent decades, vision-based Deep Learning technology has advanced rapidly, achieving substantial object detection and tracking accuracy. The Convolution Neural Network (CNN) is a Deep Neural Network (DNN) that is frequently utilised in object recognition applications.

The following layers often make up a general CNN assessment pipeline:

- The Convolution Layer, which includes several filters for extracting various characteristics from the input picture. Each filter has a set of "learnable" parameters that will be determined when the training step is completed.
- The Activation Layer, which determines whether the target neuron should be activated.

- The Pooling Layer, which decreases the representation's spatial dimension in order to reduce the number of parameters and, as a result, the network's calculation.
- The Fully Connected Layer, in which neurons are fully connected to all preceding layer activations. In a CNN, the convolution layer is frequently the most computationally demanding layer.

1.1.2.3 Object Tracking:

The automated estimate of an item's trajectory as it travels is referred to as object tracking. The objective of object tracking is to autonomously monitor the trajectory of the item after it has been detected using object identification algorithms. This technology may be used to track nearby moving cars and pedestrians to guarantee that the present vehicle does not crash with them. Deep learning algorithms have shown improvements in object tracking over traditional computer vision techniques in recent years. A stacked Auto-Encoder may be taught offline to learn generic picture characteristics that are more resilient against fluctuations in perspectives and vehicle locations by using supplementary natural photos. Then, for online tracking, the offline-trained model may be used.

1.1.3 Discussion:

The decision phase can produce a secure and effective action plan in real-time depending on an assessment of the vehicle's surroundings. Probabilistic processes and Markov chains are commonly used in the decision step.

1.1.3.1 Prediction:

When traveling through traffic, one of the most difficult problems for human drivers is coping with the potential behaviours of other vehicles, which might have a direct impact on their own driving strategy. This is especially true when the road has many lanes or when the car is approaching a traffic light. The decision unit makes predictions about surrounding vehicles and decides on an action plan based on these predictions to ensure that the vehicle drives safely in these situations. To forecast the behaviour of other cars, create a stochastic model of the other traffic participants' accessible position sets and correlate these reachable sets with probability distributions.

1.1.3.2 Path Planning:

Planning the course of an autonomous, nimble vehicle in a dynamic environment is a difficult problem to solve, especially when the vehicle's full manoeuvring capabilities are required. A brute-force method would be to explore all potential pathways and use a cost function to determine which option is the best. The brute force technique, on the other hand, would need a lot of computing power and would not be able to produce navigation plans in real time. Probabilistic planners have been used to offer effective real-time path planning by avoiding the computing cost of deterministic, complete methods.

1.1.3.3 Obstacle Avoidance:

Since protection is the most important consideration in automated cars, at least two tiers of obstacle avoidance systems must be used to guarantee the vehicle does not crash with obstructions. The first is proactive, and it is based on traffic forecasts. The traffic prediction system generates metrics such as time to collision or projected minimum distance at runtime, and the obstacle avoidance mechanism is activated to execute local path re-planning based on this information. If the proactive system fails, the reactive method, which is based on radar data, will take over. When the radar identifies an impediment, it will override the existing control and avoid it.

2. DISCUSSION

The author has discussed about the Computer Architecture on Autonomous Driving, An autonomous vehicle must be able to sense its surroundings and navigate safely without the assistance of humans. Indeed, the National Highway Traffic Safety Administration (NHTSA) of the United States Department of Transportation has legally established five categories of autonomous driving. In fact, the results of the surveys show that the actual planning of the design maps enhanced their achievement even more. The students had no idea that the questions they had answered quickly (and mostly correctly) were actually questions about concept maps. As a result, one of the study's goals was met: strengthening the links between ideas and reducing conceptual uncertainty, as well as accurately acquiring concepts and knowledge with content maps. They also assist students in forming connections between prior knowledge and newly given concepts, promoting meaningful rather than mechanical learning. The general interconnections between hardware, computer fundamentals, computer operations, and so on used to be simple and clear enough to comprehend computer systems. Nonetheless, Morden computer technologies have gotten increasingly sophisticated, making understanding the entire computer system extremely challenging. This research examines the value of idea maps in a Computer Architecture and Organization course (CAO). Students were encouraged to create idea maps over the course of a semester, which they were then permitted to utilise when preparing for their final test. Figure 2 shows the design of computer architecture.

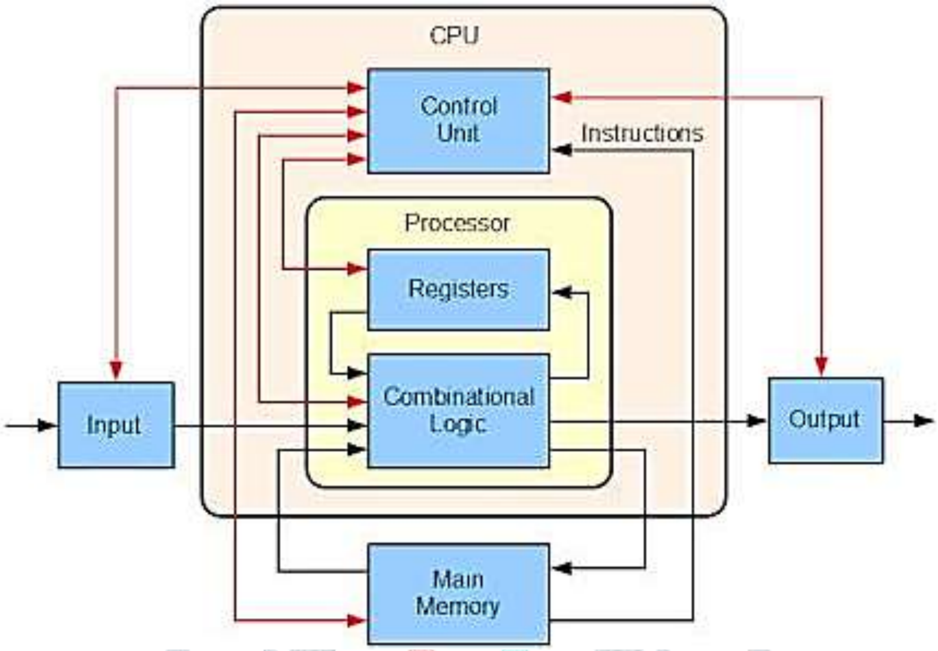

Figure 2: The above figure shows the design of computer architecture [wikipedia].

3. CONCLUSION

The author had concluded about the Computer Architecture on autonomous driving, without the help of people, an autonomous vehicle must be able to perceive its surrounds and manoeuvre safely. Indeed, the US Department of Transportation's National Highway Traffic Safety Administration (NHTSA) has created five categories of autonomous driving. One of the outcomes of the study was the students' belief that if they had been provided topic maps prior to the mid-term, they would have earned higher grades, as well as their desire to utilise content maps when studying for examinations in other courses. The content maps contributed positively not only with the teaching of the course, but also to the students' success rate when they were utilised for subsequent tests, according to this study. Existing designs frequently call for numerous servers, each with multiple high-end CPUs and GPUs, to be installed in an autonomous vehicle. These ideas have a number of flaws. For starters, the expenses are too expensive, making autonomy unattainable to the average people. Second, because this configuration uses thousands of Watts, power supply and heat dissipation become an issue, putting significant demands on the vehicle's power system. Because content maps are an effective teaching technique, it can be assumed that using this method in course teaching and studying will be helpful.

REFERENCES

[1] M. S. Thottethodi and S. Mukherjee, "Top picks from the 2013 computer architecture conferences," *IEEE Micro*. 2014, doi: 10.1109/MM.2014.53.

[2] N. Topalog Lu, S. Sahin, and Ç. Uluyol, "Comparing the effectiveness of a microprocessor training system and a simulator to teach computer architecture," *Int. J. Eng. Educ.*, 2018.

[3] Z. Radivojevic, Z. Stanisavljevic, and M. Punt, "Configurable simulator for computer architecture and organization," *Comput. Appl. Eng. Educ.*, 2018, doi: 10.1002/cae.22034.

[4] G. C. Fox and P. C. Messina, "ADVANCED COMPUTER ARCHITECTURES.," *Sci. Am.*, 1987, doi: 10.1038/scientificamerican1087-66.

[5] T. Aydogan and S. Ergun, "A study to determine the contribution made by concept maps to a computer architecture and organization course," *Eur. J. Contemp. Educ.*, 2016, doi: 10.13187/ejced.2016.15.76.

[6] B. Falsafi and G. H. Loh, "Top picks from the 2012 computer architecture conferences," *IEEE Micro*. 2013, doi: 10.1109/MM.2013.65.

[7] M. Shute, "Computer architecture: a quantitative approach," *Microelectronics J.*, 1993, doi: 10.1016/0026-2692(93)90111-q.

[8] S. Liu, J. Tang, Z. Zhang, and J. Gaudiot, "CAAD : Computer Architecture for Autonomous Driving," 2017.

[9] S. Liu, J. Tang, Z. Zhang, and J. L. Gaudiot, "Computer Architectures for Autonomous Driving," *Computer (Long. Beach. Calif).*, 2017, doi: 10.1109/MC.2017.3001256.

[10] D. Nemirovsky *et al.*, "A general guide to applying machine learning to computer architecture," *Supercomput. Front. Innov.*, 2018, doi: 10.14529/js?180106.

[11] D. Kehagias, "A survey of assignments in undergraduate computer architecture courses," *Int. J. Emerg. Technol. Learn.*, 2016, doi: 10.3991/ijet.v11i06.5776.