International Journal of Recent Research and Review, Special Issues- 2025
ISSN 2277 – 8322

COMPUTER VISION FOR AUTONOMOUS DRIVING

Dr. Ganesh Dixit[1], Shubham Sharma[2], Tanishk[3], Himanshu Sharma[4], Parth Pareek[5]
[1]Head Of Department,[2,3,4,5] Research scholar
[1,2,3,4,5]Department of Artificial Intelligence & Data Science
Arya College of Engineering, Jaipur, Rajasthan

Abstract— Research and development of futuristic vehicles that can reduce traffic accidents and contribute to a safer driving environment can be facilitated by computer vision in self- driving cars. Riders can be driven to their destinations by self- driving technology, which eliminates the need for human interaction. However, the technology for self-driving cars is still in its infancy. Deploying such autonomous systems is difficult in crowded places like cities since even a tiny amount of data can result in a serious accident. Deep learning and computer vision techniques are frequently employed to improve autonomous driving situations. The primary areas of interest for computer vision-based techniques are identifying roadblocks and assessing the flow of traffic. Additionally, a lot of researchers are improving the circumstances for autonomous driving by utilizing deep learning-based techniques like convolutional neural networks. The assessment of computer vision in self- driving cars was the main topic of this study.

Keywords— Autonomous Driving, Deep Learning, CNN, Computer vision

I. Introduction

These days, autonomous vehicles, often known as self-driving cars, are a hot issue. By detecting their surroundings and making the appropriate judgments without human assistance, self-driving automobiles are able to operate the vehicle. The majority of autonomous car systems rely on sensors, actuators, deep learning algorithms, machine learning, and other technologies.

Taking into account the current body of research on autonomous driving According to Prabhakar et al. [1], the primary part of autonomous cars is the collision avoidance system. To do it, the surrounding environment is measured and mapped using a variety of sensors. Both active and passive sensors are used in autonomous cars. Active sensors include radars, millimetres-wave radars, and lidars; passive sensors include optical sensors that resemble standard cameras. Nevertheless, those active sensors have drawbacks including slower scanning speeds, reduced spatial resolution, and interference from other sensors of the same sort. However, there is a budget for the active sensors. Additionally, visual information with additional variables, such as classes, is provided via the optical sensor.

It raises the challenge in the process of detecting obstacles. According to Tamika et al., one may utilize environmental sensing to detect neighboring cars and obstructions, comprehend traffic signs, and determine the traffic signal countdowns and status. Decision- making duties continue in accordance with the measurements gathered. Because of the vast amount of data gathered by the sensors, making decisions in real- time becomes difficult, and each circumstance is different [2]. According to Jeong et al., AI-based real- time image processing outperforms traditional image processing techniques like SURF and SHIFT for object recognition.

The scenario is analyzed using a rule-based system and sensor data that is assigned into discrete mathematical form based on the probability distribution. The rule-based system reliability is broken down into a variety of intricate scenarios when compared to existing systems [3]. The current AI-based scenario recognition and object recognition systems function as separate frameworks. There will be certain mistakes made while using two or more separate frameworks, which will decrease the system's flexibility and make operations more complicated. Johari and Swami employed a variety of computer vision algorithms, deep learning techniques, and tools to identify objects in situations involving autonomous vehicles.

In order to determine the object identification performance under various terrestrial and meteorological situations, they concentrated on CNN, SSD, R-CNN, and R-FCN. They point out that CNN learns to extract the features on its own, whereas the traditional approach splits raw images into segments for extraction [4]. Li et al' research indicates that the Tencent Autonomous Driving 16K (TAD16K) benchmark enhanced the current TT100K benchmark. Following that, findings are evaluated using the SSD and DetectNet object detection algorithms. Next, a real-world application is used to test the trained network [5]. Darapaneni et al. concentrated their research efforts on developing a straightforward autonomous driving model that could be implemented more affordably and with less complexity. Through this model, can analyses the road obstacle images can be analyzed, segmented the objects can be segmented, and assist to move more safely [6].

In the upcoming sections through this research paper, we are going to review the existing research works under section II.

The selected research works results elaborate under section III. Finally, section IV will explain the conclusions of the selected research works with their strengths and gaps.

II. REVIEWING THE EXISTING WORKS

CNN is a multilayer neural network that is specifically designed for analyzing 2D data such as videos or images. CNN can use fewer data processing requirements in the raw data input and extract the expected features. CNN has a hierarchical structure, and the collected information passes through the particular layers of the network. Each respective layer performed some digital filtering to obtain the features. A feature map is contained in the initial layer of the network. After that, it becomes a subsampling process, which helps to reduce the dimensionality. After that, trainable bias transfers to the activation function. Finally, the activation function outputs transfer to the feed-forward system to collect the outcome. The weights will be updated connecting, a mean of the backpropagation function with the loss function.

Vision-based algorithms for detecting road objects are not very accurate, claim Prabhakar et al. The aforementioned issue is brought on by ill-lit roads, various vehicle forms, and situational illumination. However, they highlight the methods for object detection that are based on deep learning. Convolutional neural networks (CNN) can employ vision- based techniques, and powerful GPUs can be employed to speed up processing. To identify and categorize the on-road impediments, it employed region-based convolutional neural networks (RCNN). They preprocessed their PASCAL VOC 2012 dataset using a pre-trained model named ZF Net. This system's output made direct use of the motion planning procedure. CNN can be used to improve mean average precision in object classification, as he mentioned, but it takes a long time for object detection. As a result, they employed Region-Based CNN (R- CNN), a modified form of CNN. The independent region proposals are produced by the first of R- CNN's three modules. The linear support vector machines are found in the third module, while the fixed-length feature vector is extracted from region suggestions in the second module. However, the R-CNN also requires a lot of processing power. Thus, they suggested a quicker RCNN as well as a fast RCNN. Region of Interest (ROI) is used in the rapid RCNN to extract the feature vectors from the feature map. Object detection performance is improved with faster RCNNs. The Area Proposal Network (RPN), a distinct CNN, was utilized to create the region proposals in the quicker RCNN approach. RCNN was used in the implementation of the on-road obstacle detection and categorization system. They used ImageNet and the ZF net network model to initialize the weights. They restricted their object classes to no more than 20 classes and used the PASCAL VOC 2012 dataset. There are numerous non-road objects in the PASCAL VOC 2012 training set. It is necessary to retrain the network model using on-road objects. Instead of retraining the model, they employed masking [1] to improve the detection process' efficiency.

They captured the real-time video and divided it into image frames. Each image frame is sent to the trained R-CNN for processing. In this process, the identified objects will be annotated with bounding boxes. The bounded box results are sent to the tracking module to take the automated motion actions.

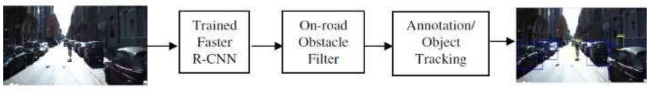

Fig. 1. System diagram of obstacle detection and classification using deep learning [1].

According to Jeong et al. [3], YOLO is the most effective AI- based object recognition method out of those like SSD and R- CNN. The key benefit of YOLO is that it can identify things 1000 times quicker than other algorithms and can identify objects that are similar to R-CNN. Jeong suggested a system that can simultaneously detect objects and recognize situations. Additionally, it will be beneficial to obtain dependable learning using that one framework. According to Jeong's research, RCNN outperforms CNN in terms of recognition rate. However, they noted that RCNN can become sluggish. Due to its speed advantage over RCNN, Single Shot Detector (SSD) can be utilized. However, the accuracy of object detection decreases due to the limitations imposed by a restricted number of anchor boxes. Therefore, they took into consideration You Only Look Once (YOLO), a more precise object detecting algorithm. Similar to SSD, it will split an image into grid cells; however, it also applies hard negative, removes the FC layer, and transforms the image into a fully convolutional model. They want to apply a hybrid YOLO LSTM model with this technique. YOLO will use real-time image frame inputs to determine the object's size, kind, and location. To find the artifacts, they employed YOLO as a detector. The situation was identified using LSTM.

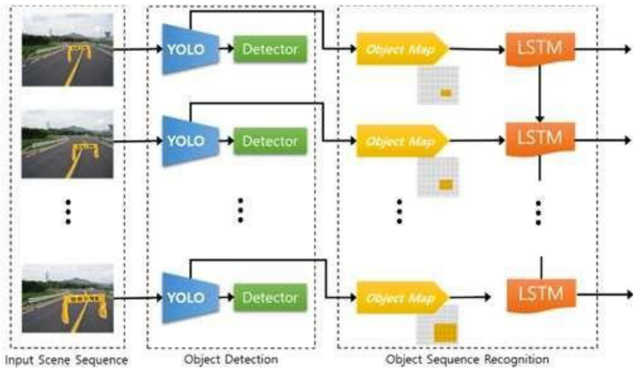

Fig. 2. System overview of driving scene understanding using hybrid deep neural network [3].

For the last layer, the model has employed the YOLO identification. LSTM requires object type and position in order to determine the situation over time. However, the YOLO detector output cannot be used directly by LSTM. To do so, they connected the object map using a unique matrix architecture. Since RNN performs worse on long sequence data, they chose to employ the LSTM structure instead. LSTM uses cell state structure to solve the RNN challenge.

When considering Johari and Swami's research work [4], they mentioned that deep learning-based visual odometry can be used for object detection processes. For Situations,

More labeling based on pixels is needed for recognition. By determining the driving state, planning the motion of the vehicle, and operating the vehicle, they break down the driving task into three parts. They proposed a deep learning method for self-driving car scenarios using the Keras and Tensorflow frameworks. The outputs of their model vehicle's sensors were relocated to dynamic and static object detection and localization. After that, forward to the perceptual and environment mapping components. After that, motion planning and controlling are done by the system supervisor. The appropriate vehicle actuators receive the output.

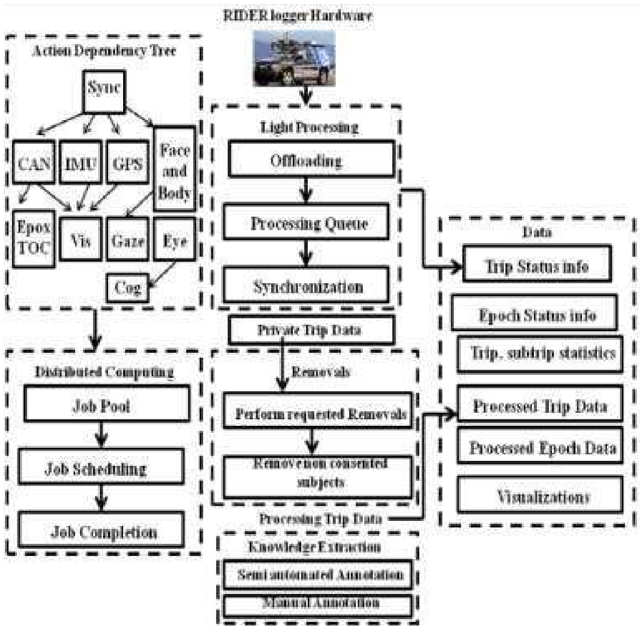

Fig. 3. MIT self-driving cars software stack [4].

Li et al. [5] stated that convolutional benchmarks like as Microsoft COCO, Pascal VOC, and ImageNet ILSVRC are utilized for object detection; nevertheless, their focus is on general-purpose detection tasks. They stated that their application in autonomous driving is not appropriate. As a result, they developed the TAD16K benchmark and used the object detection algorithms DetectNet and Single Shot MultiBox Detector (SSD) for training. By utilizing the GoogLeNet model, DetectNet improved accuracy while cutting down on training time. CNN then uses the input images of different sizes to forecast the bounding boxes and convergence map. The SSD was initialized using a pre-trained VGG-16 model that took feature map sizes and aspect ratios into account. Use the bounding box changes to improve the object's shape, and create scores for every object category.

In order to segment the images by matching the ground truth, Darapaneni et al. use a sequential image collection that is taken from car cameras. Next, use the UNET and FCN algorithms to build a model. After that, validate the model, increase its accuracy, and deploy it to make judgments. In contrast to past model research projects, they created the simulation environment and gathered the required data using the CARLA UE4 client. The resulting data is difficult to visualize because it comprises twelve bands. Thus, the data is transformed into two dimensions using the dimensional reduction algorithm. Then, by evaluating the similarities between the photographs, eliminate the unnecessary information. Considering the

class frequency, they reduced the multiclass imbalance by adding a personalized weight and loss function. To identify the roadblocks, they develop a model. However, as CNN image classifiers only provide a limited feature map and down sample each layer, they did not use them. In order to rebuild images, they employ transpose convolutions [6].

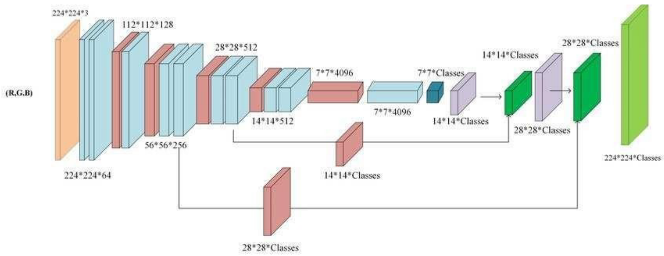

Fig. 4. UNet architecture [7].

For model evaluation and image segmentation used mean pixel accuracy and mean intersection over the union.

For self-driving cars, Yu-Ho Tseng and Shau-Shiun Jan presented a network model that included semantic segmentation with object detection. The model in this project was constructed using FCNs and SSD, and SSD7 is capable of detecting both autos and pedestrians. The base network, segmentation decoder, and detection decoder are the three components of their suggested networked paradigm. The basic network architecture has seven layers and uses 256x256 pictures as inputs. Each layer in this project is made up of max pooling, convolution, and batch normalization. They have utilized MLIND-Capstone, a segmentation decoder created by Github user Virgo. Convolutional networks serve as the foundation for semantic segmentation. Three nonsampling layers, seven transposed convolution layers, and seven convolutional layers with these max-pooling layers made up this network [8].

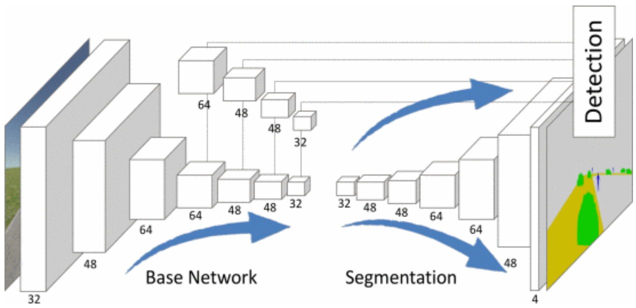

Fig. 5. Proposed network architecture for computer vision detection and segmentation for autonomous driving [8].

For the ETRI, Min et al. researched SEA level 3 autonomous driving technology. This research project includes modeling, mapping, machine learning, and basic software. The basic software includes planning, controlling, perception, and localization. Perception software was used to differentiate between moving objects and traffic lights using a camera and a lidar sensor. Global path planning, task planning, and risk assessment were all done with planning software.

In this project, three computers are used for perception, planning, and control, along with cameras and a lidar sensor for autonomous driving. They used a map database to create driving plans and localize the geoplace. Sensor-based and leader sensor-based perceptions are the two types of perception systems used in this project. A camera sensor was used to implement traffic signal detection and location. Alidar sensors and fusion with a camera sensor are used to identify moving objects. In order to find the vehicle and organize the driving task and course, the mapping technology in use creates HD map data. They have developed a vision-based mapping method that estimates motion and creates depth maps using stereo cameras. In order to detect map features and reconstruct 3D roads, machine learning is used [9].

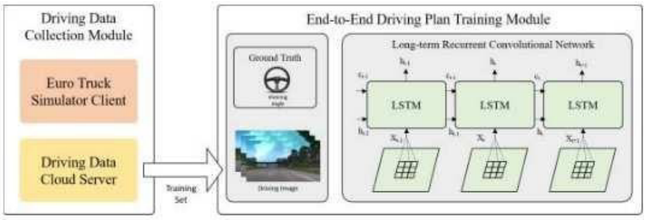

Fig. 6. System architecture [10].

The driver is in charge of controlling the vehicle's motion and knowing when to apply the brake and acceleration pedals. Using CNN, the self-driving system can identify a range of objects and determine their associations to create the driving scenario. However, there is a problem with this strategy. The driver looks at the time-series vision data instead than the single-vision data to identify the issue. CNN has limitations when it comes to time-series analysis, but it excels at spatially analyzing individual images using three-dimensional vectors. Therefore, CNN by itself won't solve the issue. Myoung-Jae Lee and Young-guk Ha suggested a solution to get around this issue [10].

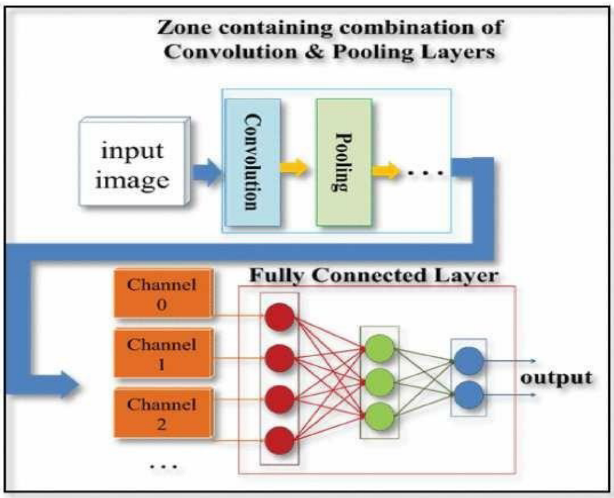

Fig. 7. Classifying images using CNN [11].

Okuyama et al have researched the autonomous driving system based on Deep Q learning. They proposed a simulation study of an autonomous agent learning to drive in a simplified setting for this investigation that consisted solely of static items and traffic signs. Then concentrated their research on learning to drive by identifying road objects using just one camera, and they trained the agent in the simulated environment using a deep Q network. After extracting visual features, the CNN's several layers eventually learn to identify the images. In the simulation scenario, obstacles are positioned at random intervals of 30 meters, while foot pathways encircle straight roads on both sides. The impediments are blocks, cars, and human figures. A pool of pre-made objects is used to randomly generate these static elements. This simulation environment was constructed using Unity, and the Unity-to-Python conversion modules make use of the interface between Unity and Python applications [11].

III. CONCLUSION

Researchers' findings throughout the aforementioned five years showed that applying deep learning and machine learning techniques significantly improved autonomous driving scenarios. However, in order to improve accuracy, life-critical methods like autonomous driving must be improved. Because RNNs can carry long sequence data, they have a gradient vanishing problem when identifying with the Prabhakar model [1], which lowers performance. After reviewing other research articles, we discovered that LSTM may be used in place of RNN and that it performs better [3]. Additionally, the use of simulation environments [6, 9, 11] offers additional advantages for simple data gathering and testing. However, using and testing in real-world situations is essential. Some models lack ground truth objects, which leads to limited results [5]. We discovered that it was brought on by the data sets' reduced availability of data. But as contemporary technology have improved, so too have the methods for gathering data sets become more sophisticated and precise.

REFERENCES

[1] "Obstacle Detection and Classification using Deep Learning for Tracking in High-Speed Autonomous Driving," by G. Prabhakar, B. Kailath, S. Natarajan, and R. Kumar, 2017 IEEE Region 10 Symposium (TENSYMP), pp. 1-6, 2017.

[2] The paper "Deep Learning for Vision and Decision Making in Self Driving Cars-Challenges with Ethical Decision Making," by B. Asmika, G. Mounika, and P. S. Rani, was presented at the 2021 International Conference on Intelligent Technologies (CONIT) and covered pages 1–5.

[3] "Driving Scene Understanding Using Hybrid Deep Neural Network," 2019 IEEE International Conference on Big Data and Smart Computing (BigComp), pp. 1– 4, by H.-j. Jeong, S.-y. Choi, S.-s. Jang, and Y.-g. Ha.

[4] "Comparing Autonomy and Studying Deep Learning Tools for Object Detection in Autonomous Self Driving Vehicles," 2nd International Conference on Data, Engineering and Applications (IDEA), pp. 1-6, 2020, by A. Johari and P. D. Swami.

[5] "Autonomous Car Driving Using Deep Learning," 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC), 2021, pp. 29–33, N. Darapaneni, P. Raj, A. R. Paduri, E. Anand, K. Rajarathinam, P. T. Eapen, K. Sethuraman, and S. Krishnamurthy.

[6] "Supervised Classification of Multisensor Remotely Sensed Images Using a Deep Learning Framework," Remote Sensing, vol. 10, p. 1429, 2018, P. Saber, W. Schwartzkopf, F. Koehler, and S. Piramanayagam.

[7] "Combination of computer vision detection and segmentation for autonomous driving," by Y.-H. Tseng and S.-S. Jan, IEEE/ION Position, Location and Navigation Symposium (PLANS), 2018, pp. 1047-1052.

[8] "SAE Level 3 Autonomous Driving Technology of the ETRI," by K. Min, S. Han, D. Lee, D. Choi, K. Sung, and J. Choi, 2019 International Conference on Information and Communication Technology Convergence (ICTC), pp. 464-466, 2019.

[9] In their 2020 IEEE International Conference on Big Data and Smart Computing (BigComp), M.-j. Lee and Y.-g. Ha discuss "Autonomous Driving Control Using End-to-End Deep Learning," pp. 470–473.

[10] "Autonomous Driving System based on Deep Q Learnig," by T. Okuyama, T. Gonsalves, and J. Upadhay, 2018 International Conference on Intelligent Autonomous Systems (ICoIAS), pp. 201-205, 2018.