International Journal of Grid and Distributed Computing
Vol. 13, No. 2 (2020), pp.77-88
http://dx.doi.org/10.33832/ijgdc.2020.13.2.06

SELF-DRIVING CAR USING LIDAR SENSING AND IMAGE PROCESSING

Sidra Memon[1], Muhammad Ahmed[1], Sanam Narejo[2], Umer Ahmed Baig[1], Bhawani Shankar Chowdry[1,4] and M. Rizwan Anjum[3]

[1]*Department of Electronics, MUET, Pakistan*
[2]*Department of Computer Systems, MUET, Pakistan*
[3]*Department of Electronic Engineering, IUB, Pakistan*
[4]*NCRA-CMS LAB MUET, Pakistan*
sidrajawaid2@gmail.com, mahmedmalik3@gmail.com,
sanam.narejo@faculty.muet.edu.pk, umar76574@gmail.com,
bhawani.chowdhry@faculty.muet.edu.pk, engr.rizwan@iub.edu.pk

***Abstract*— The self-driving vehicle reduces the driver's need and is subsequently suitable for people, such as older people, children, or individuals with disabilities, who are unable to drive. The main goal of this research is to develop a self-driving system by using the machine learning and deep learning approaches such as Convolutional Neural Network (CNN), Mask Regional Convolutional Neural Network (Mask RCNN). The developed system is capable of driving itself with minimal human input by using GPS while moving in particular lane by determining lane lines, detecting the obstacles in the path of it, recognize different objects and follow the road rules like traffic light and traffic signs and driving safely in different environmental conditions by avoiding accidents.**

***Keywords*— LIDAR, CNN (Convolutional Neural Network), GPS (Global Positioning System), Computer Vision (CV), Mask Regional Convolutional Neural Network (Mask-RCNN), Regional Proposal Network (RPN), Region of Interest**

1. INTRODUCTION

Transportation accidents are in an exceedingly one amongst the numerous causes of death in a world [1]. According to the report unconcealed by the “World Health Organization (WHO)”, over one million fatalities are caused due to road accidents and the numbers are even a lot of which incorporates very little or major injuries. Most of the time accidents happen because of human mistakes. Humans attempt mistakes in numerous ways, such as using mobile phones while driving, not following traffic rules, distracted through billboards and deficiency of sleep results in drowsiness generally while driving. Consequently, in response to the above-mentioned conditions accidents occur. Therefore, there is a need for a solution that helps humans in safe driving.

The tale of Self-driving automobiles began in the Nineteen Twenties. Once the first guided car was introduced, leading to more enhancement and improvement in cars. Further, the vision guided car [2] was introduced in 1988 with the use of LIDAR and computer vision for tracking and obstacle detection and prevention. This project was funded in the US by the DARPA using emerging technologies. For around 20 years, “Uber”, “tesla”, “google”, “Toyota” are some of the manufacturers that have been designing and testing these cars and they had achieved good results while moving towards complete automation.

Received: August 28, 2020
Reviewed: November 26, 2020
Accepted: December 7, 2020

ISSN: 2207-6379 IJGDC

A self-driving car can make more efficient use of car transportation and improve service to disabled individuals.

Self-driving cars have an ethical, legal, social, economic impact in the next years of evolution. Simultaneously, dealing with the issues of autonomy, privacy, liability, security, data protection, and safety [3].

Our developed prototype system focuses on majorly 5 features of self-driving cars such as lane detection, traffic sign recognition, traffic light recognition and classification, object detection and recognition and navigation. In general, there are five key elements of a self-driving car, Vision, Sensors data Integration, Navigation, Path planning, and control.

The motivation behind this research project is to build the Self-driving system prototyped on a remote-control Model car to maximize the safety and effectiveness of the vehicle. The other important things that give us the motivation are that it saves a lot of time by decreasing average travel time as the self-driving car could travel at a faster speed and it decreases a person's effort in driving,

The paper is organized as follows. Section 1 briefly explains the autonomous vehicles; Section 2 highlights the related research and past autonomous vehicle strategies that they follow. Section 3 describes the research methodology and it explains briefly about our implementation workflow, the different algorithms used for “obstacle detection and avoidance”, “lane detection”, “traffic light detection”, “traffic signs detection and their classification” and “navigation”; Section 4 discusses the results whereas the paper is concluding in Section 6.

2. RELATED WORK

In the year 1977, “Japan's Tsukuba Mechanical Engineering Laboratory" made the first truly robotic vehicle with two cameras for processing and have a speed of up to 30 kilometers per hour. The long - haul competition for Self - Driving Cars starts with DARPA (Defense Advanced Research Projects Agency) [4]. In 2014, Google unveiled a model of its unmanned car, which had no controlling wheel, brake or accelerator, gas pedal, simply has buttons to start, stop, and a PC screen to demonstrate the path. Through GPS and Google maps to navigate [5]. By the end of 2014; this project was tested on 8 self-driving vehicles for more than 700,000 km and there was no mishap.

In three levels, namely perception, planning, and control, the core competencies of an unmanned vehicle system. Perception alludes to an autonomous system's capability to gather data and to obtain relevant environmental knowledge [6]. The raw sensor data is directly used for further processing in point cloud-based methods. This method gives a better environmental representation, but at the cost of increased processing time and decreased memory efficiency. To mitigate this, the raw point cloud usually uses a voxel-based filtering mechanism to decrease the number of points, as proposed in [7, 8].

In self-drivingcar environmental perception, the vision system usually includes lane detection, object detection, and traffic sign detection. Primarily the lane detection is the key component of the vision of smart cars-based driver assistance systems. The paper [9] proposed a lane finding algorithm based on the comparison of colored lines, and another algorithm which is based on representative line extraction. A Median filter is applied for image noise reduction and for lane detection. In self-driving systems, the recognition of traffic signs play an important role. In [10], the author proposed a novel approach to the recognition and detection of traffic signs in real-time situations by using the “Support Vector Machine (SVM) algorithm” for detection and “convolutional neural networks” to classify traffic signs in their subclasses. The paper [11] proposed an algorithm for traffic sign detection which is based on deep learning. The “RGB normalization-based color finding algorithm”, and regional feature decision criteria are used to automatically recognize the multi-sign interconnection candidate regions and perform edge smoothing and contour tracking for the extracted target regions.

Detection of objects is a critical task for self-driving cars besides high accuracy for the protection. In paper [12], the author proposed Squeeze-Det, a complete CNN network for the recognition of objects aimed at meeting all the above limitations at the same time. However, the particular position of a vehicle is often difficult to determine and thus the problem of localization is often defined as a problem of assessment [13]. The GPS signal and the car's dead calculation odometry are authentic, stable, and costly, and high accuracy sensors are required. In [14, 15], Road-matching algorithms utilized in addition to GPS and INS to modernize the vehicle's localization computation with a precedent road map to restrain movement. In [16], For a self-driving car, correct and precise self-localization is compulsory. This paper depicts the vehicle localization method by using two different formats of the map, 1st one depends on 2D vector map of initiating footprints, representing the borders of buildings using vector lines, 2nd one based on the planar surface map of building and ground. It is also to note that the Hybrid Cloud Computing model has been growing extensively due to its Infrastructure as a Service (IaaS) architecture, customization and cost benefits[17].

This research paper proposes a straightforward and less expensive self-driving automotive model that fulfills all the requirements of automation or self-driving systems.

3. RESEARCH METHODOLOGY

Our goal is to develop a low-cost self-driving car with the simplest techniques and sensors available in the market for third world countries such as Pakistan. Our proposed self-driving car prototype is built based on the following workflow diagram, Figure 1. summarizes the flow of our research methodology. It encompasses the entire decision-making and execution process and steps followed to make autonomy possible. The steps are in such a way that firstly it takes raw data from the sensors and then which is being processed by an algorithm and gives the perception of objects, traffic signs, and traffic lights. Afterward, through these perceptions, track planning is made, and controls are taken to make the ride safe and easier. The steps are further explained briefly in the subsequent sections.

The initial step of this system is to perceive and gather the necessary data from its environment and road surroundings. The approach is to extract data from four types of sensors: LIDAR, Ultrasonic sensor, GPS, and a camera. The camera is employed in detecting lane lines, traffic light detection and their classification, vehicles, and pedestrians. The LIDAR and the Ultrasonic sensor are utilized for obstacle evasion. Localization/Navigation is done with the GPS sensor. Data from these sensors flows into several components of the perception step.

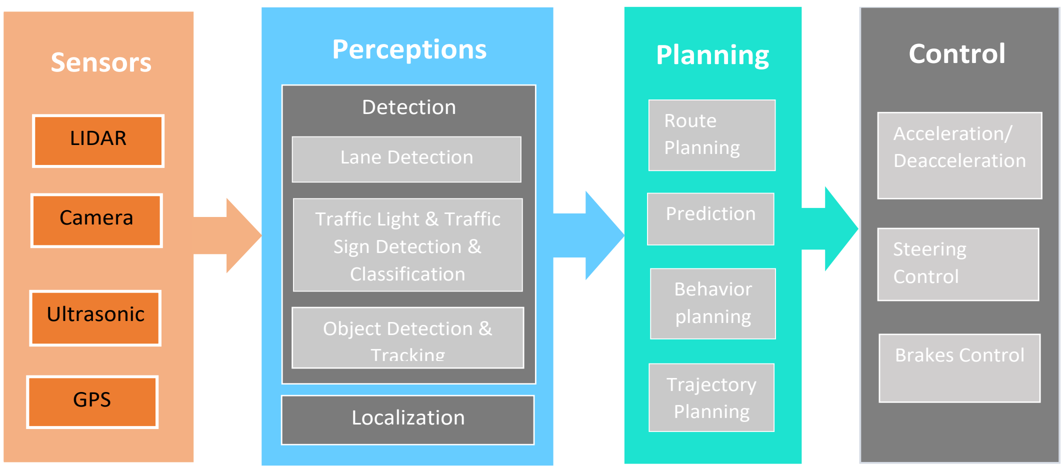

Fig. 1 Workflow Diagram

The initial step of this system is to perceive and gather the necessary data from its environment and road surroundings. The approach is to extract data from four types of sensors: LIDAR, Ultrasonic sensor, GPS, and a camera. The camera is employed in detecting lane lines, traffic light detection and their classification, vehicles, and pedestrians. The LIDAR and the Ultrasonic sensor are utilized for obstacle evasion. Localization/Navigation is done with the GPS sensor. Data from these sensors flows into several components of the perception step.

The second step is to perform the required algorithms on the drawn-out data, for detection and localization purposes. The perception step is divided into detection and localization parts. The detection block uses this data to detect entities outside the vehicle i.e. lanes, other vehicles, obstacles, traffic lights, traffic signs, and pedestrians. Computer Vision CV and Machine learning algorithms are labored to achieve each individual perception task. All of them use the same visual data from the same sensor.

In this work, we have used ‘LIDAR’ and three ‘Ultrasonic sensors’ to detect the obstacles in the path by measuring the distance between the car and obstacle and it is controlled by ‘Arduino UNO’ micro-controller. Ultrasonic sensors are mounted on the bumper of the car. When the sensor receives data from the surrounding area then it sends the information to the controller to decide the motion of the car wheel to choose a path free of obstacles. When the car starts, both the car motors will run, and the car moves forward. The ultrasonic sensor constantly calculates the distance between the car and the reflective surface during this time. The time required by the beam to return is saved in a variable and converted to distance using suitable measurements as shown in the form of an equation (1)

$$\text{Distance} = (\text{Time} * \text{Speed of Sound in Air } (343 \text{ m/s}))/2 \qquad \text{Equation (1)}$$

The micro-controller processes this data. If the range between the car and the obstacle is less than 200 centimeters. Later, the car gets to stop and scan for new distances in both the left and right directions. The car will take a left turn if the range to the right side is more than the left side and vice versa. But this can be done for low range distance. For high range distance, LIDAR does the same thing. A LIDAR is placed at the top of the car. It will take the 3D image of the car’s surroundings and calculate the distance to an obstacle by illuminating laser light pulses and then differences in wavelengths and return times of laser can be used to create the 3-D representations and then detect the obstacle. Figure 2 is the circuit diagram that shows the connections between the motors of the car and the ultrasonic sensor with the Arduino.

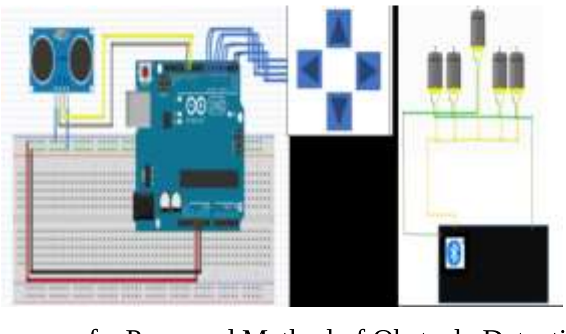

Fig. 2 Circuit Diagram of a Proposed Method of Obstacle Detection and Avoidance

Figure 3 shows the resulting output video frame from the entire process for lane detection. A video of the road with a mobile camera fixed onto the car is taken and is then passed through several steps. The first step in finding lanes lines is the distortion correction. Incoming light is distorted by camera lenses to concentrate on the camera sensor. They often end up mildly inaccurately distorting light. However, by converting the ‘RGB’ image into ‘Grayscale’, then to obtain the camera matrix and distortion coefficients. and then, to correct the remainder of the input data. “Perspective Warp” is used in the camera video in the second step to getting a bird's eye view of the lanes so that the car would be able to detect curved lanes easily without any errors.

Typically, the lane lines have a strong road contrast, so “Sobel Filtering” is applied after getting the bird’s eye view of the lanes to filter out the road image. ‘Sobel Operator’ gives the image function's gradient and detect high contrast regions of the road to filter lane markings and disregard the road. In addition to this, **HSL** Color Space (Hue, Saturation, Lightness is an alternative representation of the RGB color model) is also used to identify saturation and lightness variations.

In the next step “Histogram Peak Detection” is employed. If it begins at a place where lane pixels are present, it performs well. Each part of the histogram shows the number of white pixels in each image column. Then taking each side of the image's highest peaks, one for each lane line. Thereafter, the “Sliding Window” algorithm is used to distinguish between the boundaries of the left and right lane so that we can fit two different lane boundaries. A sign is provided to the pixels that fall inside the windows. Then, we apply polynomial regression individually to the selected pixels. Finally, we just create an overlay that fills the lane’s detected portion and then it can be applied to video.

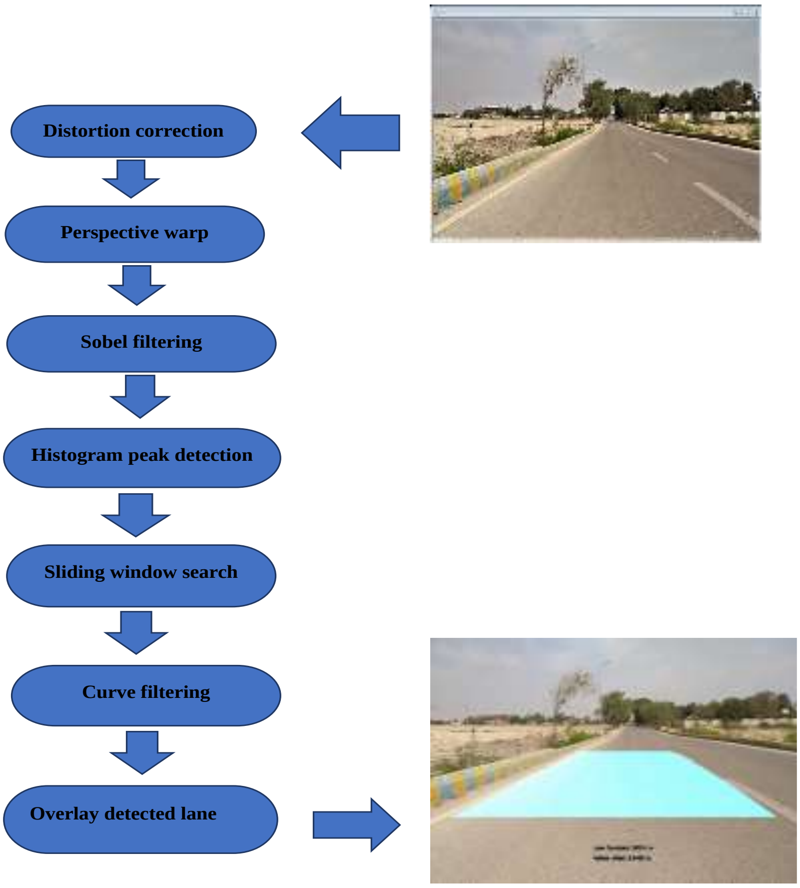

Fig. 3 Lane Detection Flow Chart

The detection of objects mainly related to classes of vehicles and pedestrians. A new detection algorithm base on machine learning has been proposed in this paper called “Mask-RCNN” to improve the precision and robustness of object recognition under harsh conditions like shadows, different weather conditions, and light changes. This algorithm can perform both; **Object Detection**, which provides the coordinates (x, y) bounding box for every individual object in an image and **Instance Segmentation**, allows us to obtain a '**pixel-wise mask'** for every object in an image, even if the objects are of the same class label.

The architecture of Mask R-CNN uses a 'Region Proposal Network (RPN)' to produce regions of an image that have an object. Each of these areas is rated on the basis of their "objectness rating" and then the most confident object regions of the top N are retained. We set N=300 each of the 300 chosen ROIs pass through three parallel branches of the network: Label prediction, Bounding box prediction, Mask prediction. Each of the 300 ROIs goes through non-maximum suppression during prediction and the top 100 detection boxes are maintained, this results in a 4D 100 * L * 15 * 15 tensor where L is the number of class labels in the dataset and 15 * 15 is the dimension of each L mask. Our model suggests pictures in RGB format, so we use a technique to exchange the color channels (as opposed to the standard ordering of OpenCV's BGR color channel) and then move the picture forward through the network to create both object detection and pixel-wise mask predictions. After that, we create bounding boxes and score texts plus class label for each individual object in the image and then the image is converted back to BGR. For the object, we take out the pixel-wise segmentation and resize it to the initial size of the image. Then we finally limit the mask because it is a binary array/image and extract the ROI where the object is located.

Object detectors like "YOLO", "Faster R-CNNs", and "Single Shot Detectors (SSDs)" produce sets of (x, y) coordinates representing an object's bounding box in a picture. Obtaining an object's bounding boxes is useful, but the bounding box itself does not show us that which pixels belong to the foreground object and which pixels related to the background.

Due to the number of problems, robust and accurate detection of traffic signs is a challenge, such as shadows, several weather conditions, and partial obstruction that often occur in real traffic videos. This paper proposed a deep learning algorithm called "Convolutional Neural Network (CNN)" with "Keras" for the detection of traffic signs and their classification. Traffic signs detection and classification is difficult because it has a complex and large dataset. To visualize the traffic signs, we assign different 'ClassID' for each sign. Before the design and train our model, the 'RGB' image first initially converted into 'Grayscale' and then preprocessing our data. The pre-processing of the image target to speed up image processing, filter the noise and image interference and boost the accuracy of the identification. Some images are bright, and some are dim after histogram equalization these images have similar lighting effect.

We then use them to train our model. Firstly, we added convolutional layer 'Conv2D' that consist of filters that recognize various features within the image. As 'CNN' consists of filters, we specify 30 filters and set the size of the filters 5 * 5. Then specify the shape of each input image being fed into the neural network and it is 32 * 32 * 1 where 1 is the single channel of pixel intensity values. Then use a technique for the activation of nodes. After the convolutional layer, 'Maxpooling2D' layer is used to decrease the picture spatial size, parameters, and computation in the network. The pooling layer works on each individual feature map and enables the features to be recognized. Then we set the pooling size 2 * 2 which scaled-down all the feature maps from the convolutional layer. Thereafter, we added the second convolutional layer but this time used 15 filters to minimize the computational power and use the size 3 * 3 of the filter and add the pooling layer for the second time having the same size and after that 'Flatten Layer' is used to take all the multi-layered image levels down to one plane.

After that "dense layer" is added, every node in the subsequent layer is connected to every preceding layer and by setting the amount of node to 500 it increases the network size and the number of parameters which means the network fit the data arbitrary well, below that gives less accuracy and more than 500 requires more computing power. Then we use the "Dropout layer" to drop half of the input nodes. Finally, the technique is used so that the layers effectively classifies between 43 classes.

Self-Driving Car is a smart vehicle that can detect traffic lights and follow them according to road rules. This enables the car to move in an unpredictable environment by avoiding accidents. “TensorFlow Object Detection API” is used in this paper in the task of traffic light detection.

For training the model first we accumulate our data than manually annotate the pictures for the network. The annotation tool generates a file. In the third step, we convert our information into the ‘TFRecord’ format to train the model with the API. This format takes the pictures and the file and merges them into one and that has been provided as an input for training. Then we set up a pipeline for detection. This paper uses two models for training, “SSD-Inception” and “Faster-RCNN”. The number of classes are adjusted to 4 and the path is also set for the model checkpoint, the train and test data files. We also decreased the number of regional proposals from 300, ‘Faster RCNN’ to 10 and ‘SSD Inception’ from 100 to 50. Then we also create a label map for each class. So, before training our model the Image dataset, TensorFlow model API, ‘Label map’ file, ‘TFRecord’ file that created earlier, ‘COCO’ pre-trained network model is required. CPU is used for training a total of four different models, two models with ‘Faster R-CNN’ (one for simulator images and real images each) and two with ‘Single Shot Detector Inception’.

The identification and classification were great with both models, although the model trained by SSD inception produced a few minor errors but was properly classified by the Faster R-CNN model. However, the SSD Inception model's plus point is that it ran nearly 3 times faster on simulator than the Faster R-CNN model and nearly 5-6 times faster on the real images.

The localization component in the perception block uses the GPS to decide the location of the car on the map. The instruction is regenerated each time the self-driving car moves to enable it to know its new position each time so it can verify that planned path is being followed. The third and most important step is for the system to be conclusive enough to plan its action on a real-time basis. With this step, it is decided how to treat the surroundings and position on the map. This step is divided into trajectory planning, prediction, behavior planning, and motion planning.

Behavior planning uses information from the lane detection block of the previous step to consider different locations as options, to reach will perform an action e.g. overtaking a vehicle, changing lanes. The trajectory planning block is fed output from the obstacle and object detection blocks from the previous step. Trajectory planning is computing the path or trajectory to be followed to reach the considered location from the options. Each of the path is evaluated to decide which path is most suitable for reaching a potential location is the most time, energy and cost-effective. Trajectory planning also uses prediction to decide the best trajectory, whether the predicted trajectory of another vehicle, pedestrian or obstacle collides with the vehicles’ own.

The fourth step is to take over car controls and drive through, without any human intervention. The plans made for self-driving cars motion and behavior are translated onto the brake, steering, and accelerator with the control step which actuates the car by sending messages for acceleration, brake, and steering. The waypoints to target locations and definite velocities generated by the planning step are taken as messages which are processed through a mechanical-control algorithm, calculating just how much to steer, accelerate, or throttle, in order to complete the intended task.

The RC model car used, is directly controllable with a remote control, which has all the steer, speed, and brake controls to operate the car. Therefore, all the messages that are sent from the planning subsystem are translated onto the remote’s controls. This is accomplished through a manipulating circuitry connected to the remote’s interior circuitry.

4. RESULTS AND DISCUSSION

The results obtained from previous investigations [14, 15] using a lane detection algorithm using TensorFlow and the real-time traffic sign detection and recognition model are with low accuracy and precision. In our model we are fusing ultrasonic sensor and LiDAR for the obstacle detection and avoidance which gives 3D representation of the surrounding. The visualization in Figure 4 indicates that the car is moving in a defined path without striking an obstacle.

Fig. 4 Obstacle Detection & Avoidance

We are using a computer vision algorithm for lane detection in which we used the perspective warp function that gives the bird eye view of the lanes on the road so that we are able to detect the curved or straight lane lines easily with high accuracy in shadows and light changing areas as shown in Figure 5.

Fig. 5 Lane Detection using the Perspective Wrap Function

The object recognition is done through the machine learning algorithm named Mask-RCNN which includes the objection recognition that bounding box at every object in the frame and instance segmentation that gives a pixel-wise mask to that object which makes the recognition highly accurate as shown in Figure 6.

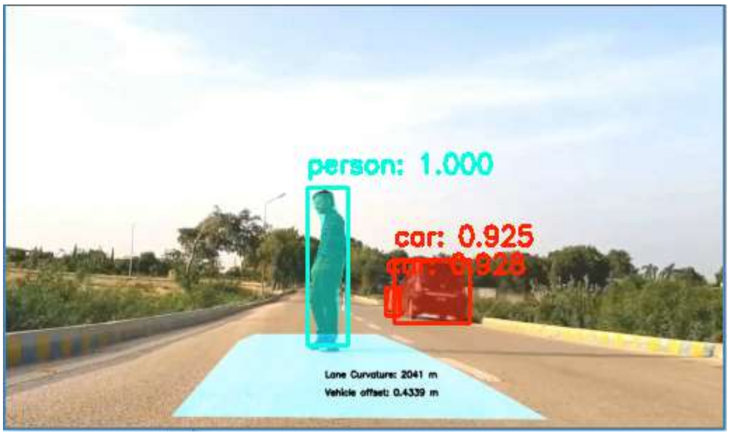

Fig. 6 Object Recognition through Mask RCNN

Traffic sign detection and recognition model uses the technique of deep learning which uses multi-layer CNN and to visualize the traffic signs, we assign different 'ClassID' for each sign which increases precision in detection and recognition of traffic signs as shown in Figure 7.

Fig. 7 Traffic Sign Recognition

"TensorFlow Object Detection API" is used in the task of traffic light detection which makes the RCNN 3 times faster. "SSD-Inception" and "Faster-RCNN" is used to train the model. It makes the traffic light detection easier in a sunny environment as shown in Figure 8 and 9.

Fig. 8 (a) and (b) Traffic Light Detection through Tensorflow in a Sunny Environment

Fig. 9 (a) and (b) Traffic Light Detection through Tensorflow in Night

The GPS system, owing the function to store the present position data, can be applied to running positioning and to record the continuous tracing as shown in Figure 10.

Fig. 10 Navigation by GPS System

5. CONCLUSION

The over-all project "self-driving car using lidar sensing and image processing technology" is based on the Sensors (such as Ultrasonic, LIDAR, GPS module), Controller, Processor, Machine Learning and Deep Learning approaches. The model detected obstacles precisely using 3 ultrasonic sensors and one lidar sensor and the data of the sensors is processed by 'Arduino' micro-controller. Arduino controls the movement of the wheel motors and avoids the obstacles by deviates the path. Lane detection is done using mobile which serves as a camera and "OpenCV" technique for the processing of camera images to detect the lane lines. The camera images pass through several processes like perspective warp, Sobel filtering, sliding window search, curve fitting and in the last, we just create an overlay that fills the lane's detected portion. For object recognition, we use the deep learning algorithm called "Mask R-CNN" which provides the coordinates (x, y)-bounding box for each individual object in an image and it also gives us the '**pixel-wise mask'** for any single object in an image. By using this algorithm, the car not only detects the object but also precisely classify that from which class the object belongs and where the object is located.

Traffic signs detection and classification is also a difficult task because it has a complex dataset. We use limited dataset having different classes, "Keras" and "Convolutional Neural Network (CNN)" a machine learning technique for the classification of traffic signs. We assign different 'ClassID' for each sign and every sign passes through the convolutional layers and pooling layers and the car detects and recognize each sign successfully with good accuracy.

We build self-driving system on prototype remote-control car that can perform several driving tasks like lane detection, obstacle detection and avoidance, traffic light detection, navigation, traffic sign detection and object recognition with greater accuracy but due to the insufficient amount of resources and time we are unable to build the self-driving system on a real-time car because it requires years of research and testing. “Google”, “Tesla”, “Audi”, “Toyota”, “Uber”, “Nissan” are some organizations that are developing and testing these cars for more than 20 years.

In the future, our goal is to implement this system on a real car with larger datasets, more accurate sensors, and powerful processor to handle all the incoming data, and enhance the capabilities of a car by improving algorithms used to increase systems road awareness. Also, the system would train itself with the passage of time making it ahead closer to the perfect self-driving system. Also, the dependency on a single main sensor would be minimized by increasing the number of secondary sensors. Not only this, but the system with minimal changes can also be designed for public transport vehicles such as buses and trucks.

ACKNOWLEDGMENTS

We acknowledge the support of the ‘Haptics, Human Robotics, and Condition Monitoring Lab’ established in Mehran University of Engineering and Technology, Jamshoro under the umbrella of National Center of Robotics and Automation funded by the Higher Education Commission (HEC), Pakistan

REFERENCES

[1] Developments in the Last Century, the Present Scenario, and the Expected Future of Autonomous Vehicle Technology. Proceedings of the 12th International Conference on Informatics in Control, Automation and Robotics. doi: 10.5220/0005540501910198.
[2] Thorpe, C., Hebert, M., Kanade, T. and Shafer, S., “Vision and Navigation for the Carnegie Mellon Navlab”, High Precision Navigation, pp. 97-122, doi:10.1007/978-3-642-74585-0_6.
[3] Ryan, M., “The Future of Transportation: Ethical, Legal, Social and Economic Impacts of Self-driving Vehicles in the Year 2025”, Science and Engineering Ethics, (2019), doi: 10.1007/s11948-019-00130-2.
[4] Levinson, J., Askeland, J., Becker, J., Dolson, J., Held, D., Kammel, S. and Thrun, S., “Towards fully autonomous driving: Systems and algorithms”, 2011 IEEE Intelligent Vehicles Symposium (IV), (2011), doi:10.1109/ivs.2011.5940562.
[5] Dethe, S. N., Shevatkar, V. S. and Bijwe, R. P., “Google Driverless Car”, International Journal of Scientific Research in Science, Engineering and Technology, (2016).
[6] Asvadi, A., Premebida, C., Peixoto, P. and Nunes, U., “3D Lidar-based static and moving obstacle detection in driving environments: An approach based on voxels and multi-region ground planes”, Robot, Auton. Syst., vol. 83, (2016), pp. 299-311.
[7] Zhao, G., Xiao, X., Yuan, J. and Ng, G. W., “Fusion of 3D-LIDAR and camera data for scene parsing”, J. Vis. Commun. Image Represent, vol. 25, (2014), pp. 165-183.
[8] Moosmann, F., Pink, O. and Stiller, C., “Segmentation of 3D lidar data in non-flat urban environments using a local convexity criterion”, In Proceedings of the 2009 IEEE Intelligent Vehicles Symposium, Xi’an, China, (2009) June 3-5, pp. 215-220.
[9] Park, H., “Implementation of Lane Detection Algorithm for Self-driving Vehicles Using Tensor Flow”, In book: Innovative Mobile and Internet Services in Ubiquitous Computing, pp. 438-447, DOI: 10.1007/978-3-319-93554-6_42
[10] Shao, F., Wang, X., Meng, F., Rui, T., Wang, D. and Tang, J., “Real-Time Traffic Sign Detection and Recognition Method Based on Simplified Gabor Wavelets and CNNs. Sensors”, (2018), Doi: 10.3390/s18103192.
[11] Research on traffic sign detection algorithm based on deep learning [Quan Wang Weiping Fu First published: 03 August 2018] https://doi.org/10.1002/cpe.4675.
[12] Wu, B., Iandola, F., Jin, P. H. and Keutzer, K., “SqueezeDet: Unified, Small, Low Power Fully Convolutional Neural networks for Real-Time Object Detection for Autonomous Driving”, IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), (2017), DOI: 10.1109/CVPRW.2017.60.
[13] Kelly, A., “Mobile Robotics”, Cambridge University Press: New York, NY, USA, (2013).

[14] Najjar, M. E. E., "A Road-Matching Method for Precise Vehicle Localization Using Belief Theory and Kalman Filtering", Auton. Robots, vol. 19, (2005), pp. 173-191.
[15] Guivant, J. and Katz, R., "Global urban localization based on road maps", In Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS), San Diego, CA, USA, (2007) October 29-November 2.
[16] Javanmardi, E., Gu, Y., Javanmardi, M. and Kamijo, S., "Autonomous vehicle self-localization based on abstract map and multi-channel LiDAR in urban area", IATSS Research, (2018), doi:10.1016/j.iatssr.2018.05.001.
[17] Dhirani, L. L., Newe, T. and Nizamani, S., "Hybrid Cloud Computing QoS Glitches", 2018 5th International Multi-Topic ICT Conference (IMTIC), Jamshoro, (2018), pp. 1-6, doi: 10.1109/IMTIC.2018.8467224.

