

IoT programming to Develop Self Driving Robotics Car using OpenCV and Other Emerging Technologies

Mohammed Maaz ¹ and Sabah Mohammed ²

¹Lakehead University

²Affiliation not available

October 30, 2023

Abstract

The advancement of Artificial Intelligence & Deep Learning has catalyzed the field of technology. The progression in these fields is exponentially increasing, and the discoveries which were once just an imagination are now changed into reality. The evolution of cars each year has made a lot of difference in people travelling from one place to another. One such reform involving Artificial Intelligence & Deep Learning is the birth of a self-driving car. The future is here where one can reach their destination hassle-free safely without the fear of accidents. This paper introduces a practical model of the self-driving robotics car, which can travel from one position to another on different types of tracks. A Pi-camera module is attached with the help of Raspberry Pi, which sends series of image frames to the Convolutional neural network, which then foretells the car to move in a specific direction, i.e. right, left, forward and reverse direction. The outcome is the robotics car, which travels in the desired direction without any individual effort.

IoT programming to Develop Self Driving Robotics Car using OpenCV and Other Emerging Technologies

Mohammed Maaz
Department of Computer Science
Lakehead University
Thunder Bay, Canada
mmoham10@lakeheadu.ca

Dr. Sabah Mohammed
COMP5800YB Supervisor
Lakehead University
Thunder Bay, Canada
mohammed@lakeheadu.ca

Abstract—The advancement of Artificial Intelligence & Deep Learning has catalyzed the field of technology. The progression in these fields is exponentially increasing, and the discoveries which were once just an imagination are now changed into reality. The evolution of cars each year has made a lot of difference in people travelling from one place to another. One such reform involving Artificial Intelligence & Deep Learning is the birth of a self-driving car. The future is here where one can reach their destination hassle-free safely without the fear of accidents. This paper introduces a practical model of the self-driving robotics car, which can travel from one position to another on different types of tracks. A Pi-camera module is attached with the help of Raspberry Pi, which sends series of image frames to the Convolutional neural network, which then foretells the car to move in a specific direction, i.e. right, left, forward and reverse direction. The outcome is the robotics car, which travels in the desired direction without any individual effort.

Keywords— *Artificial Intelligence (AI), Deep Learning (DL), Raspberry Pi, Robotics, Self-Driving, OpenCV*

I. INTRODUCTION

An autonomous vehicle, a self-driving car, is a connected automobile that relies on hardware, software and machine learning to travel in various conditions considering all the obstacles and road conditions utilizing real-time sensory data [1]. Today, Tesla, Google, Uber, and General Motors are all trying to create their self-driving cars that can run on real-world roads. Many analysts predict that within the next five years, we will start to have fully autonomous vehicles running in our cities. Within two to three decades, nearly all cars will be fully independent. There are various levels of different automation, according to SAE International [2]. Each level depends upon the human involvement associated with handling the automobile, which is as follows No Automation (level 0), Driver Assist (level 1), Partial Automation (level 2), Conditional Automation (level 3), High Automation (level 4), Full Automation (level 5). A lot of failures do exist in different models, but each day due to more enhanced techniques and algorithms, prospects to create high chances of yielding better results are more.

The technology inside self-driving cars considers many important factors like who monitors the road? Steering, acceleration, deceleration, monitoring surroundings, fallback for self-driving failures, automation taking full control. Motor vehicle accidents are the primary cause of death of teenagers in

the United States—with as many as 8,000 deaths per year have been caused by drivers aged between 16–20. Statistics in the US show the largest percentage of vehicle fatalities are by teens, and according to study, American teen drivers cost the Americans over \$34 billion annually. Many metropolitan cities, larger states, and federal agencies are actively encouraging autonomous vehicle deployment, but this is not universal; this is because the problem arises on trusting on an autonomous vehicle. A lot of challenges are faced in the self-driving industry each day, and many new issues arise after applying regressive testing methodologies, here are a few challenges faced namely,

- i. It is challenging to handle real-time obstacles for self-driving cars.
- ii. Driving requires many complex social interactions.
- iii. Bad weather makes everything trickier.
- iv. Regulator's approval is always a concern; designing regulations based on safety is very vital.
- v. Cybersecurity plays a critical role in self-driving cars so that they cannot be hacked, as vehicles get smarter, there are more ways to break through and disrupt what's going on.

There are many other points taken into consideration based on autonomous driving level, which is capable of handling vehicle performance being driven by a highly sophisticated system of sensors and actuators like Navigation & guidance, Driving & safety, Road conditions, Weather conditions, Traffic conditions, Radar interference. This paper helps in overcoming these challenges faced in self-driving using OpenCV and IoT networking using different technologies like Raspberry Pi, Radar, Lidar, and various sophisticated sensors and actuators. Radar is usually used for long-range object detection and also tracking velocities of pedestrians and other vehicles. Lidar acts as an eye, which provides 360 deg view surrounding the car. Using different sensors, the data is taken in real-time and is fed in the convolutional neural network medium neural network. Once trained it yields the desired result to move the car.

II. METHODS FOR DEVELOPING IOT BASED SELF-DRIVING ROBOTS

The development of a self-driving robotics car depends upon the level of automation based on human interaction and control of the vehicle. There are different levels of automation, which are as follows, No automation (Level 0), Drivers assistance (Level 1), Level 2 & 3 requires a human driver to co-share the driving effort, it refers to ADAS (Advanced Driver-Assistance Systems), High automation (Level 4), Full Automation (Level 5).

- i. Level 0 – A car with absolutely no automation is termed as level 0.
- ii. Level 1 – Drivers Assistance, the features here include rear camera, lane discipline assistance which tells driver to follow certain traffic laws and adds a few safety features.
- iii. Level 2- The automated system takes control on driving aspects such as steering, acceleration and deceleration when approached a pedestrian or an obstacle. The driver however can manage everything. Ex. Tesla's with autopilot.
- iv. Level 3 - Uber's self-driving car is an example of Level 3 because while the carrier controls most of the navigation, but still the driver is required.
- v. Level 4 - Level 4 is defined as high automation. Google's Waymo the autonomous vehicle company comes under Level 4 autonomy.
- vi. Level 5 is termed as full automation; the self-driving car has complete control over all systems.

For partial and high-level automation i.e. Level 4 and Level 5 self-driving autonomous vehicles, the AI system will have full control of all the systems. In theory, the AI system will not get drunk and will not get exhausted while driving a car and is considered as justification for pursuing actual self-driving cars. No more intoxicated drivers because of sophisticated AI-driven self-driving cars. No more human faulty chauffeurs that snap and opt to drive into a roadside fence. Human decisions and individual exemptions will go out the window, and driverless cars will not get angry.

This paper introduces partially level 4 and level 5 based automation, as it has features of both high and full automation. The self-driving car system consists of three subsystems: input system (camera, ultrasonic sensor), processing unit (computer) and RC car control unit.

A. Input Unit

A Raspberry Pi board (model B+), connected with a Pi camera module and an HC-SR04 ultrasonic sensor is used to obtain input data. Two python programs run on Raspberry Pi for streaming colour video and ultrasonic sensors to send data to the computer via a network, which is a local Wi-Fi connection, connected to the same network. In order to produce low latency video streaming, video resolution is reduced or scaled down to QVGA (320×240) resolution.

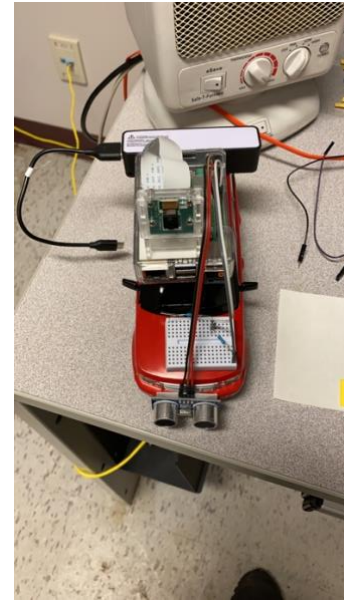


Fig. 1. RC car mounted with Raspberry Pi along with Pi Camera, HC-SR04 sensor.

B. Processing Unit

The processing unit (computer) manages various tasks: receiving data from Raspberry Pi, neural network training and prediction (steering), object detection (stop sign), distance measurement (using the ultrasonic sensor), and sending directions to Arduino through the USB connection.

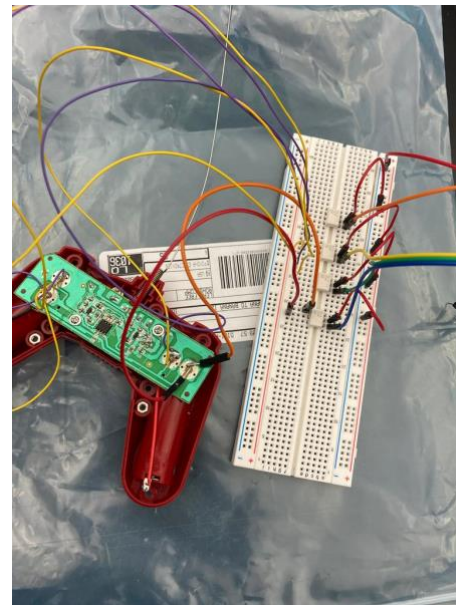


Fig. 2. RC controlled remote controller unit connected to Arduino UNO

TCP Server

A TCP server program runs on the computer to receive the video, which converts the video into streamed image frames and also collects ultrasonic sensor data from the Raspberry Pi.

Image frames are transformed into grayscale and are decoded into numpy arrays, which are then fed into the neural network.

Neural Network

One benefit of using a neural network is that once that it is trained, and it only needs to load trained parameters later. Once trained, prediction can be very fast, and only one part of the image frame is used, which is the lower half of the input image frame for training and then predicts direction to turn. There are approximately 38,000 nodes in the input layer, and then only 32 nodes are used in the hidden layer, and these are selected relatively arbitrary. The output layer constitutes four layers where every node assigns steering control instructions: left, right, forward and reverse sequentially.

Object Detection

This paper adopted the shape-based approach and uses Haar feature-based cascade classifier algorithm for object detection. Considering every object needs its own classifier and understands the same process in training and detection, this paper is only centered on stop sign detection. OpenCV offers a trainer as well as a detector. Positive samples (containing target object) were acquired using Pi camera, and the captured video is converted to image frames and cropped to get the desired object which are visible. Negative samples (without target object), are obtained without using obstacles. Using the cv2 package, which supports OpenCV in python, the frames are used to apply different image processing techniques to get the desired object. These are fed into the network for training and get desired results.

III. TECHNOLOGY – SELF DRIVING CAR

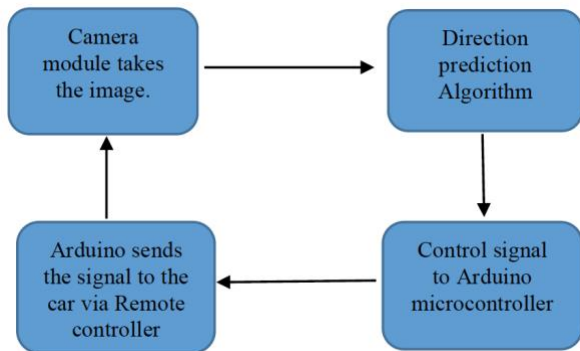
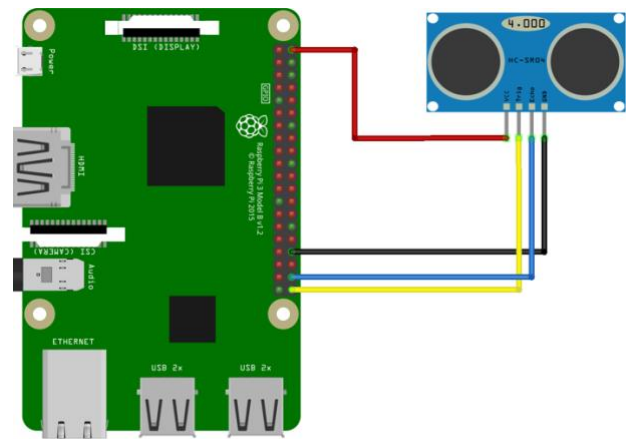


Fig. 3. Concept of execution of autonomous car block diagram

In this paper, the execution is distributed into different steps: the pi camera module, which is connected to Raspberry Pi captures the video, this captured video is converted into image frames which are fed into the convolutional neural network. Various algorithms are applied into the network for several epochs. The algorithm predicts the direction and conveys the control to Arduino UNO. Finally, the signal received triggers the controller to move the car in specified direction.



fritzing

Fig. 4. Fritzing Diagram of HC-SR04 sensor connected to Raspberry Pi

The HC-SR04 ultrasonic sensor is connected to Raspberry Pi using a small piece of breadboard and some jumper cables. The sensor and Pi are mounted on the car, the ultrasonic module is used to constantly measure distances. The Trigger accepts a 3.3V output from the Pi but the Echo pin must be converted to 3.3V from 5V before it reaches the GPIO input. The Pi Camera is also connected to Pi, it is used to capture the video and convert the video into image frames and pass in onto CNN.

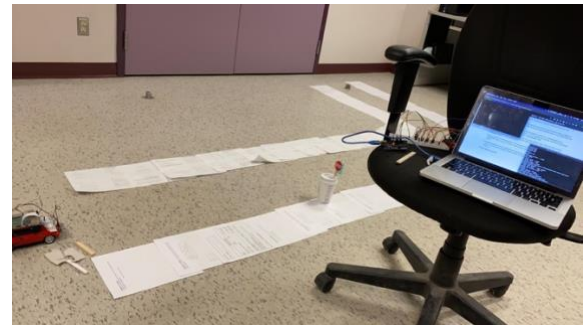


Fig. 5. Sample images of the track on which the car is trained.

The car was trained on different combinations of tracks like straight, curved. Training the vehicle on the straight track was much more comfortable as the image frames were much easier to obtain and train the model, a total of 15-20 videos were recorded to obtain better results, the images were re-sized to 320*240 on which the network was trained. The Convolutional Neural Network had 128 input nodes, two hidden layers of 32 nodes each and finally, the output layer consisting of 4 nodes for each of the four output.

IV. AUTONOMOUS CAR MODEL DESIGN

The autonomous car hardware specification used in this paper are Raspberry Pi, Pi camera module, Arduino UNO and remote-controlled car. Different types of algorithms are used in self-driving car models which are as follows: Haar Cascade

Algorithm, Regression Algorithm, Decision Algorithm, Pattern Recognition Algorithm.

A. Haar Cascade Algorithm

Haar Cascade Classifier is basically termed as object detection approach, it uses classifier generated during training the network. It is based on machine learning approach where in a cascade function is trained with a number of positive and negative images. Based on training of positive and negative images it detects the object in the other captured image frames. The algorithm uses Haar Cascade features to identify the patterns in the frames, its features very much resemble to convolutional kernel. A convolutional kernel is a matrix which is used to carry out different operations on the image frames such as edge blurring, detection, sharpening, reshaping the image etc. The 3 main features used are blurring, edge detection and sharpening.

B. Regression Algorithm

This algorithm is used for object localization, predicting movement and also object detection. The working of regression algorithm compares variables to predict an objects position and distance. There are different kinds of regression algorithm used in self-driving cars like decision forest regression, neural network regression and Bayesian regression. A combination of decision forest and Bayesian is best to use as it is good in predicting events and provides fast online detection.

C. Decision Algorithm

The decision matrix algorithm analyzes a set of data and its corresponding values and then identifies and evaluates the performance of the associated data. As the name suggests, they are used for decision making. Whether a vehicle demands to brake or initiates to move in a particular direction based on the level of confidence, these algorithms recognize and predict the next movement of objects. The decision matrix algorithms have several decision models trained independently, and these models predict to make an overall prediction while minimizing the chances of errors in decision making.

D. Pattern Recognition Algorithm

The images captured through a sensed camera in self-driving autonomous, which includes an Advanced Driver Assistance Systems (ADAS) consists of all sets of environmental data and refining of the image frames, discovers the instances of an object category and rules out irrelevant data in the image frame. Before analyzing the objects, the identification of patterns is an essential step in a dataset. These kinds of algorithms are termed as data reduction algorithms.

In this paper all types of algorithms were used one at a time on the self-driving autonomous car to check how it recognizes the image frames and train the model. Haar's cascade classifier was easy to implement on the cost-effective hardware, it also yielded better result with ease of execution. Other 3 techniques took lot of time and process to run on the board

had look of issues due to multiple reasons like development of the machine learning was difficult on the hardware. All these algorithms can be applied but would rather require additional hardware development kits like Google coral, intended to bolster the development of machine learning models which are the edge. The accuracy obtained using Haar's cascade classifier was close to 60% on training samples and 80+ % on testing samples. In fig 6, the accuracy on training samples can be seen.

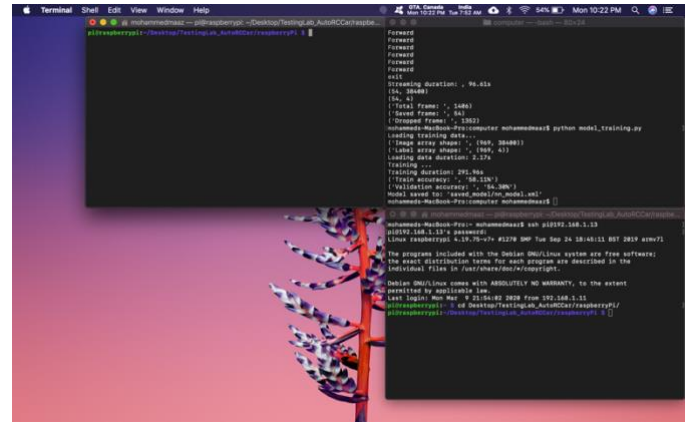


Fig. 6. Accuracy of the trained autonomous car model using Haar's Cascade Classifier.

RESULTS

In this paper, after training the model several times with 32 videos of different tracks with many failed attempts, the model was successfully able to run the car on the track as desired in forward, right, left directions. Stopping the vehicle if we find an obstacle or seeing a stop sign wasn't that accurate compared to moving the car in a specified direction, but if tried out using different algorithms in the future, it would yield better results.

CONCLUSION

In this paper, the overall design of the autonomous car model presented was cost-effective and was done using the OpenCV package in python. In the field of an autonomous self-driving car, the proposed work yields better hassle-free transport system. The real time capabilities can be extended to real world so that the travel is safe and less drunk drivers and lesser accidents, also the concern on auto insurance based on driver and demerit system would be lesser risk in insurance practice. Vulnerable testing of autonomous car is currently better, but machines cannot be trusted with 100% automation when human life is vital here, so universally it is still considered a low based industry, whereas in the future it may play a very important role in transportation.

LIMITATION IN THE CURRENT SYSTEM

Hardware and Software learn and react from different situations by machine learning. Few situations are like unpredictable agent behaviour, complexity in perception self-driving security, visibility occlusion from weather, optical

illusions like a mirror reflection. Most of the real-world scenarios aren't considered, so extending the model by trying out different sensors and actuators would enhance the system and would also increase the accuracy and safety features of the autonomous car, which is very vital in the self-driving automotive industry.

CONTRIBUTION AND FUTURE WORK

In advancement to the current model, the system can be optimized by using more data and train model for a longer number of epochs so that it yields better results in terms of self-driving. The car moves slightly out of the track, as lane discipline is an important rule in traffic laws. If we current this in the future, it would make system more efficient in terms of traffic laws, and it will also be added as a safety feature. In the future, more enhanced algorithms and sensors can be applied to yield better results.

REFERENCES

- [1] Zhao, Xin, Nan Wu, Yunfeng Lou, and Harutoshi Ogai. "Development of automatic driving system by robocar." In *SICE Annual Conference 2011*, pp. 2170-2173. IEEE, 2011.
- [2] Inagaki, Toshiyuki, and Thomas B. Sheridan. "A critique of the SAE conditional driving automation definition, and analyses of options for improvement." *Cognition, technology & work* 21, no. 4 (2019): 569-578.
- [3] Sun, Yuxiang, Nan Wu, Shigeyuki Tateno, and Harutoshi Ogai. "Development of driving support system for electric vehicle by using image processing technology." In *2012 12th International Conference on Control, Automation and Systems*, pp. 1965-1968. IEEE, 2012.
- [4] Hong, Tian, Wang Lixia, and Ding Xiaoqing. "Pedestrian detection based on merged cascade classifier." In *2011 International Conference of Soft Computing and Pattern Recognition (SoCPaR)*, pp. 237-241. IEEE, 2011.
- [5] Hasan, Y., Arif, M.U., Asif, A. and Raza, R.H., 2016, December. Comparative analysis of vehicle detection in urban traffic environment using Haar cascaded classifiers and blob statistics. In *2016 Future Technologies Conference (FTC)* (pp. 547-552). IEEE.
- [6] Janahiraman, Tiagrajah V., and Mohamed Shahrul Mohamed Subuhan. "Traffic Light Detection Using Tensorflow Object Detection Framework." In *2019 IEEE 9th International Conference on System Engineering and Technology (ICSET)*, pp. 108-113. IEEE, 2019.
- [7] Rahul, Y., and Binoy B. Nair. "Camera-Based Object Detection, Identification and Distance Estimation." In *2018 2nd International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE)*, pp. 203-205. IEEE, 2018.
- [8] Aziz, Mochamad Vicky Ghani, Ary Setijadi Prihatmanto, and Hilwadi Hindersah. "Implementation of lane detection algorithm for self-driving car on toll road cipularang using Python language." In *2017 4th International Conference on Electric Vehicular Technology (ICEVT)*, pp. 144-148. IEEE, 2017.
- [9] Kulkarni, Ruturaj, Shruti Dhavalikar, and Sonal Bangar. "Traffic Light Detection and Recognition for Self Driving Cars Using Deep Learning." In *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, pp. 1-4. IEEE, 2018.
- [10] Lin, Wen-Yen, Wang-Hsin Hsu, and Yi-Yuan Chiang. "A Combination of Feedback Control and Vision-Based Deep Learning Mechanism for Guiding Self-Driving Cars." In *2018 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*, pp. 262-266. IEEE, 2018.