ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

**International Journal of Innovative Research in Computer and Communication Engineering**

*(A High Impact Factor, Monthly, Peer Reviewed Journal)*

*Website: www.ijircce.com*

**Vol. 6, Issue 6, June 2018**

# Autonomous RC Car Using Neural Network and Deep Learning

Amita Jajoo[1], Akarsh Kapasi[2], Anup Kalyanshetti[3], B Niket[4]

Assistant Professor, Department of Information Technology, D. Y. Patil college of Engineering, Akurdi-Pune, India

**ABSTRACT:** The project aims to build a monocular vision autonomous car prototype using Raspberry Pi as a processing chip and Arduino as a hardware interface. An HD camera coupled with an ultrasonic sensor is utilized to facilitate required data from the real world to the car. Avoiding possibilities of accidents and other human errors, the car can commute from one place to another in an intelligent and safe manner. Many existing algorithms like neural networking, deep learning and machine learning are combined together to provide the necessary control to the car.The main reason to use machine-learning algorithms in this project is to involve the basic functions and methodologies of AI. Here, literacy in AI and computer science will become as crucial as classic literacy such as reading or writing. By implementing parallelism with this process, we developed a new AI concept in order to uphold AI literacy. The concept comprises modules for different age groups on different educational levels. Fundamental AI/computer science topics addressed in each module are, amongst others, problem solving by search, sorting, graphs and data structures.

**KEYWORDS**: Raspberry PI, lane detection, obstacle detection, AI.

## I. INTRODUCTION

It is a common habit of people to multitask whenever needed and possible. For example, people often prepare mental lists of things to buy, points to cover in an important presentation or even mentally plan their entire day while driving. Thus, driver error is one of the most common cause of accidents. In addition to this, increased use of mobile phones, in-car entertainment systems, more traffic, and complicated town-planning, have increased the risk of accidents on road.
With a somewhat constant increase in number of accidents, it has become crucial to take over the human errors ensure safety of people in the car as well as of those on the road. This can be achieved with self-driving cars, which just need to know the destination and then let the passengers continue with their work. This will avoid accidents as well as will provide relief to people with busy schedules.
In this project, we try to tackle all the problems that are faced while driving a real car. The Remote Control (RC) car represents a smaller version of the real car with all its functions. Nowadays the research on automatic cars has taken a huge step forward with the work of Google and Tesla. Nevertheless, Mercedes with its Distronic did the first successful original work in 1999.
In recent years the development on autonomous cars have increased significantly due to automation in every field and the need of man to make every thing easier. We have developed several ways to automate cars and adding many unique features to the car according to the need but still haven't added the feature that the car navigates on its own. This features can be very helpful and probably one of the most important feature in future of car and car production management. This will indeed help countless amount of people if used cautiously.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

**International Journal of Innovative Research in Computer and Communication Engineering**

*(A High Impact Factor, Monthly, Peer Reviewed Journal)*

*Website: www.ijircce.com*

**Vol. 6, Issue 6, June 2018**

FIGURE1: MERCEDES DISTRONIC 1999

## II. RELATED WORK

There are mainly two papers related to this project:-

1) Control of autonomous cars for intelligent transportation system by Mohammad Abdul Qayuml, NafiulAlam Siddiqui, Mohammad Abtiqul Haque3, Abu Saleh Md. Tayeen Klipsch School of Electrical and Computer Engineering, New Mexico State University ->
This work emphasizes on founding a technology to develop novel approach for the incorporation of intelligent system control into transportation systems.
Implementation of an algorithm that controls a car to track a predefined track, which incorporates a human driver with computer control to increase human performance while reducing dependance on detailed driver attention is carried out in this work. Information received from the attatched camera about vehicle position and alignment provides the automatic decision making intelligence required to follow a virtual vehicle moving on track. The result shows good performance of this model independent algorithm with cheap RC cars.

2) An Integrated Manual and Autonomous Driving Framework based on Driver Drowsiness Detection by Weihua Sheng, YongshengOu, Duy Tran, Eyosiyas Tadesse, Meiqin Liu, Gangfeng Yan ->
In this paper, a blueprint for automatic switching of manual driving and autonomous driving based on driver drowsiness detection is proposed and developed. The scale-down intelligent transportation system (ITS) testbed is presented first. This testbed has four main areas: an indoor localization system, an arena, automated RC cars, and roadside monitoring facilities. Secondly, the drowsiness detection algorithm, which integrates facial expression and racing wheel motion to recognize driver drowsiness is presented. Third, a hand-operated and autonomous driving switching mechanism is created, which is started by the detection of drowsiness. Finally, experiments are performed on the ITS testbed to demonstrate the effectiveness of the proposed framework.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 6, June 2018

## III. Ongoing Work

Currently four of the tech giants are working on the goal on creating an efficient autonomous car i.e. Tesla, Mercedes, Google and Uber with many other smaller organizations and university.

1. Tesla

In mid-October, 2015 Tesla Motors rolled out version 7 of their software in the U.S. that included Tesla Autopilot capability. On 9 January 2016, Tesla rolled out version 7.1 as an over-the-air update, adding a new "summon" feature that allows cars to self-park at parking locations without the driver in the car. Tesla's autonomous driving features can be classified as somewhere between level 2 and level 3 under the US Department of Transportation's National Highway Traffic Safety Administration (NHTSA) five levels of vehicle automation. At this level, the car can act independently but requires the full supervision of the driver, who must be prepared to take control at a moment. Autopilot should be used only on limited-access highways, and sometimes it will fail to detect lane markings and disengage itself. In urban driving, the system will not follow stop signs or even read traffic signals. In addition, the system also does not detect pedestrians or cyclists.

2. Google

In August 2012, Alphabet (then Google) announced that their vehicles, typically involving about a dozen cars on the road at any given point of time, had completed over 300,000 autonomous-driving miles (500,000 km) accident-free. They also stated that they were starting to test with single drivers instead of in pairs. In May 2014, Alphabet disclosed a new model that had no steering wheel, brake pedal, or gas pedal and was fully autonomous. In March 2016, Alphabet had test-driven their vehicles in autonomous mode a total of 1,500,000 mi (2,400,000 km). In December 2016, Alphabet Corporation announced that its technology would be derived to a new subsidiary called Waymo. Alphabet's test cars have been involved in 14 collisions, of which other drivers were at fault 13 out of 14 times, according to their reports regarding accidents. In 2016 the car's software caused a crash.

3. Mercedes

Due to Mercedes' past records of slowly implementing improvements of their autonomous driving characteristics that have been enormously tested, very few crashes that have been caused by it are known. One of the known crashed dates occurred in 2005, when German news magazine "Stern" was testing Mercedes' old Distronic system. During the test, the system sometimes was not able to manage to brake in time. Ulrich Mellinghoff was the Head of Safety, NVH, and Testing at the Mercedes-Benz Technology Centre. He stated that some of the tests failed because the vehicle was being tested inside a metallic hall, which created problems with the system's radar. Nowadays, upgraded radar and numerous other sensors, which are not exposed to a metallic environment anymore, are present in the iterations of the Distronic system. In 2008, it was discovered in a study conducted by Mercedes comparing the crash rates of their vehicles supplied with Distronic Plus and the vehicles without it. They concluded that those equipped with Distronic Plus have approximately 20 percent lower crash rate. In the year 2013, Mercedes went a step ahead in testing by inviting German Formula One driver Michael Schumacher to try to crash a Mercedes C-Class vehicle. This vehicle was well- equipped with all safety features that Mercedes offered for its production vehicles at the time. This features included the Stop and Go Pilot, Active Blind Spot Assist, Active Lane Keeping Assist, Brake Assist Plus, Collision Prevention Assist, Distronic Plus with Steering Assist, and Pre-Safe Brake. Due to the safety features, Schumacher was unable to crash the vehicle in practical circumstances.

4. Uber

In March 2017, an Uber test vehicle was involved in an accident in Arizona. The Uber vehicle was flipped as the other car failed to turn out.

## IV. Hardware Trends

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 6, June 2018

(a) Sensor Programming
The use of sensor is primarily used for navigation, based on the current systems and recommendations.

i. RADAR Sensor
FMCW radar is used in radar sensors for detecting moving or static targets, like cars, trains, cargo, etc in extreme weather conditions. FMCW is the short form for Frequency Modulated Continuous Wave. For preventing collisions, radar based sensors are best on board mobile equipment such as reach stackers, and mining vehicles. Port machineries like carriers, handlers, and shippers are also used.
• RTN7735PL transmitter
• RRN7745PL/46PL receiver
• RPN7720PL power amplifier in eWLB
• RCC1010 RADAR companion chip in TQFP-48 PGN

ii. LIDAR sensor

Lidar is a sensor which is also called as LIDAR, LADAR, and LiDAR. It is a surveying process. It measures distance from given point to a target by lightening up that target with the help of a pulsed laser light, and measuring the reflected pulses with a sensor. To create digital 3D-representations of the target, differences in laser return times and wavelengths can then be used. The name lidar, can be considered as an acronym of Light Detection and Ranging or Light Imaging, Detection, And Ranging, was initially a combination of words- light and radar.
• Velodyne HDL-64E LIDAR sensor (Google)
• Scanse Sweep 500Hz
• Luminar
• Strobe Cruise

## V. PROBLEM STATEMENT

The problem until now is that all the present successful autonomous cars use "Symbiotic Learning" for storing data on the server. This puts a huge pressure on server for the dependencies of its data set by test runs of the machine, that in-turn affects the performance of the car. In addition, the dependency of car on any one of the system for functioning or navigating affects the overall performance and increases the chances of accident in case of failures.
The main problems of the existing models are:-

- Dependency on single sensing (input) system.
- More requirement for test cases for training data.
- Complex in nature due to Symbiotic Learning.

## VI. SYSTEM FEATURES

The car will contain the following features:-
• Python based software base for stable navigation and more support by massive native libraries.
• Dual input system i.e by camera and ultrasonic sensor
• Image processing by neural network to successfully generate accurate measurements of track.
• Embedding concepts of Artificial Intelligence and Machine Learning for less training data and accurate testing results.

## VII. EXTERNAL INTERFACE REQUIREMENT

1. Hardware Requirements
   • Raspberry PI 3 B+ (1GB SDRAM, 1.2 Ghz Processor, Onchip Bluetooth and wifi)
   • Arduino UNO 3
   • PI Camera 5MP Camera
   • Ultrasonic sensor HC-SR04
   • RC Car

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

**International Journal of Innovative Research in Computer and Communication Engineering**

*(A High Impact Factor, Monthly, Peer Reviewed Journal)*

*Website: www.ijircce.com*

**Vol. 6, Issue 6, June 2018**

2. Software Requirements
   - Raspbian OS
   - Python
   - OpenCV Libraries 2.40.10.1
   - Numpy
   - PyGame
   - PySeries

## VIII. SYSTEM ARCHITECTURE

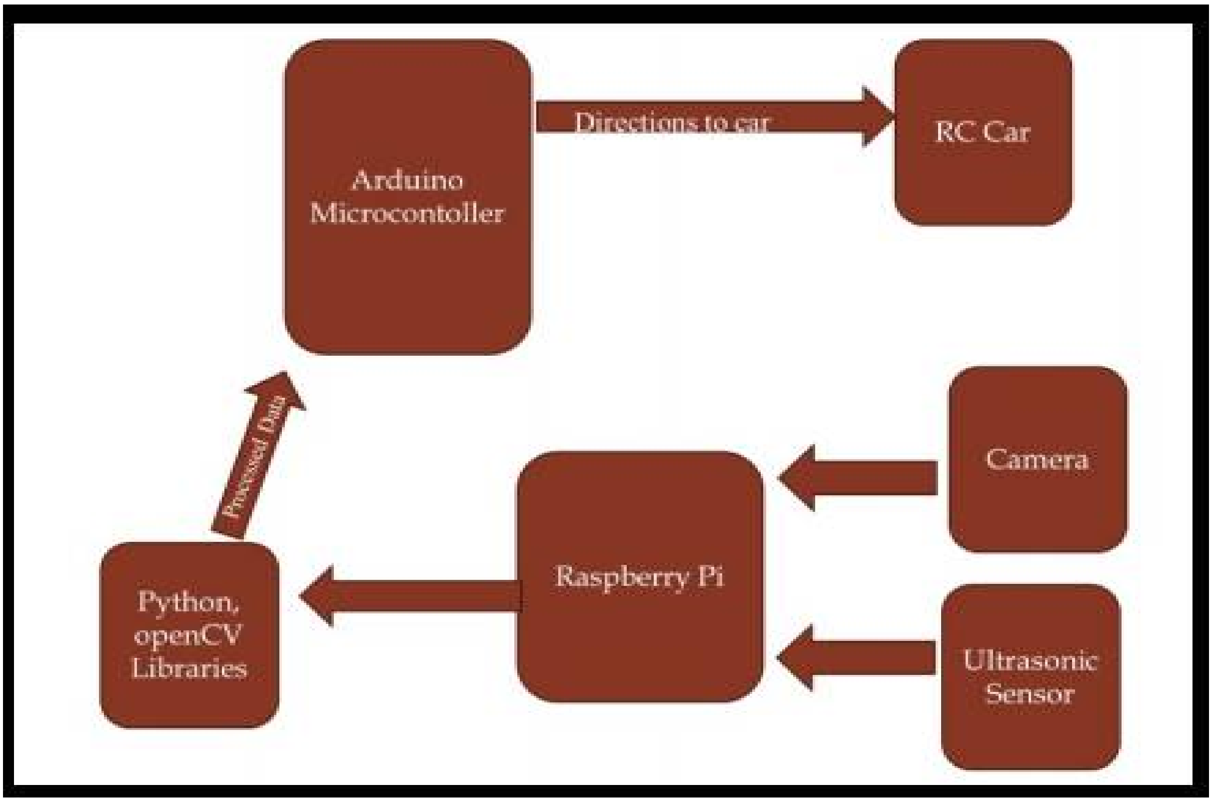


Figure 2: System Architecture.

## IX. PROPOSED SYSTEM

The project will work on a simple RC car and with many modification make it an autonomous car model. The input should be received by Raspberry PI through camera (visual) and sensors (physical).

**Important Modules and Algorithms**

1. Modules.

- Raspberry PI 3 model B+
- Arduino UNO MEGA 3
- PI Camera
- System

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(A High Impact Factor, Monthly, Peer Reviewed Journal)

Website: www.ijircce.com

Vol. 6, Issue 6, June 2018

2. Algorithm.

(a) Cascade Classifier (HaaropenCV)

Object recognition consists of digital image features that are also called Haarlike features.The features get their name from their intuitive similarity with Haar wavelets and were used in the first real-time face detector.

A collaboration with an alternate feature set based on Haar wavelets instead of the usual image intensities was stated in the publication by Papageorgiou et al.Viola and Jones adapted the idea of using Haar wavelets and developedthisalgorithm. This algorithm mentioned above deals with adjacent rectangular areas at a certainlocation in a detection window. It adds up the pixel intensities in eacharea and calculates the difference between these sums. This difference is then used to differentiate subsections of an image. For example, suppose say we have an image database with human faces. It is a general observation that in all the given faces, the area of the eyes is darker than the area of the cheeks. Hence, face detection consists of a regular Haar feature which is a set of two adjacent rectangles that are positioned above the eye and the cheek area. The rectangles are in a position with respect to the detection window that acts like a bounding box to the target object. In this case the target object is the face.

During detection phase of the Viola Jones object detection framework, a window of the size of the target is moved over the input image. For each subsection of the image, the Haar-like feature is calculated. This difference is then tallied with a learned threshold that differentiates non-objects from objects.

This type of Haar-like feature is just a weak learner or classifier which means that its detection quality is just a little better than random guessing. Therefore, a huge quantity of Haar-like features are needed to describe an object with sufficient accuracy. The Haar-like features are hence, arranged in a classifier cascade in the Viola Jones object detection framework in order to generate a strong learner or classifier. The key plus point of a Haar-like feature over most other features is its calculation speed. A Haar-like feature of any size can be calculated in constant time (2-rectangle feature requires roughly 60 microprocessor instructions) due to the usage of integral images.

One of the contributions of Viola and Jones was the usage summed-area tables, which were known as integral images. Integral images can be stated as 2-D lookup tables in similar a matrix with size equal to the original image. Every element of the integral image contains the addition of all pixels positioned on the up-left region of the original image (in relation to the element’s position). This arrangement allows computing addition of rectangular areas in the image, at any location or scale, using only four lookups: sum = I(C) + I(A) I(N) I(D) where points A, B, C and D are of integral image as shown in figure below:

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

**International Journal of Innovative Research in Computer and Communication Engineering**

*(A High Impact Factor, Monthly, Peer Reviewed Journal)*

*Website: www.ijircce.com*

**Vol. 6, Issue 6, June 2018**

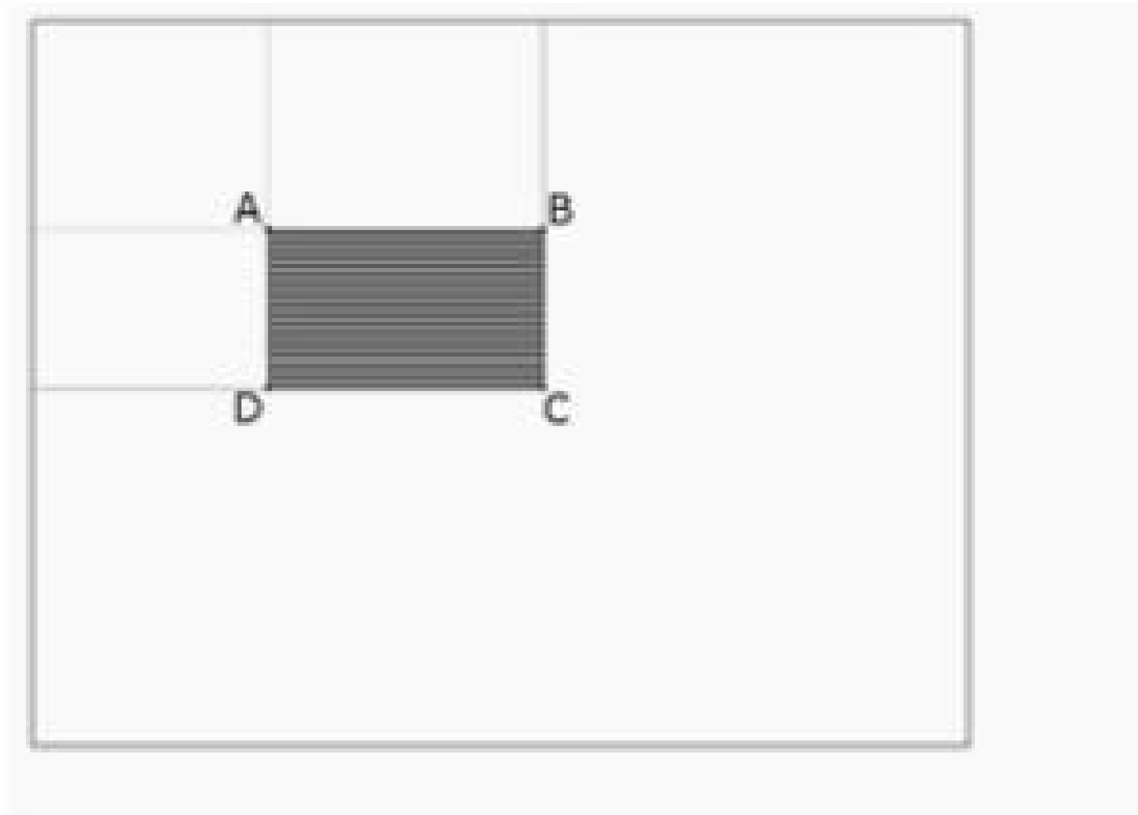


Figure 3: internal image with window of target's size.

Finding the addition of the shaded rectangular area, every Haar-like feature might require more than four lookups, considering how it was defined. Viola and Jones contains 2-rectangle, 3-rectangle and 4-rectangle features that require six lookups, eight lookups, and nine lookups respectively.

(b) Chess Board Camera Calibration

A traditional problem in computer vision is 3D modelling, where a 3D structure about a scene is to be deduced from its 2D images. In real life, cameras are complicated devices and, it is essential to use a camera to photograph and measure distance between objects present in the real 3D world. A conventional pinhole camera model is used to shape the relation between 3D co-ordinates of the real world and pixel co-ordinates with the help of perspective transformation.

$$\mathbf{x} = K[R \quad t]\mathbf{X} \quad , \quad \mathbf{x} \in \mathbb{P}^2 \quad , \quad \mathbf{X} \in \mathbb{P}^3,$$

where $\mathbb{P}^n$ is the projective space of dimension $n$.

In such situations, it is necessary to gauge or estimate different parameters of the given perspective model. This is achieved by camera calibration.

$$3 \times 4 \text{ matrix } M = K[R \quad t]$$

Calibration of camera is very crucial with respect to computer vision pipeline as majority of successive algorithms need information of parameters of the camera as input. For accurate camera gradation or tuning, chessboards are implemented. This is because, a chessboard structure is simple. The grids present can be assigned for naturally marking required points in an image. Following techniques of traditional calibration use chessboards:

1. Multiplane Calibration:

   Multiplane Calibration enables the computation of camera constants from multiple views of a single plane.

2. Direct Linear Transformation:

   The DLT gradation or calibration applies correlation between real world points (3D co-ordinates) and image pixel points to gauge camera constants.

## X. CONCLUSION

In this project, a method to make a self driving robot car is presented. The different hardware components and their assembly are clearly described. A noble method to determine the uneven, marked or unmarked track edges is explained in details relying upon OpenCV. Using ultrasonic sensors, the collisions with obstacles is avoided. The implementation of the given algorithm has been successfully carried out on a small autonomous car.

## REFERENCES

1) OpenCV 2.40.10.1 Libraries Documentation
2) Numpy documentation
3) PyGame and PySeries Documentation
4) Agarwal and Duchi, 2011,A. Agarwal, J.C. Duchi **Distributed delayed stochastic optimization** Advances in Neural Information Processing Systems (2011), pp. 873-881
5) Agarwal et al., 2014
6) A. Agarwal, O. Chapelle, M. Dudík, J. Langford**A reliable effective terascale linear learning system** J. Mach. Learn. Res., 15 (1) (2014), pp. 1111-1133
7) Avidan and Butman, 2006 S. Avidan, M. Butman **Blind vision** European Conference on Computer Vision (2006), pp. 1-13
8) Bishop, C. M. (2006) Pattern Recognition and Machine Learning. Chapter 5: Neural Networks.
9) Schmidhuber, J. (2015). Deep Learning in Neural Networks: An Overview. Neural Networks 61: 85-117.
10) Bengio, Y., LeCun, Y., Hinton, G. (2015). Deep Learning. Nature 521: 436-44.
11) Goodfellow, I., Bengio, Y. and Courville, A. (2016) Deep Learning. MIT Press.
12) **Dropout: a simple way to prevent neural networks from overfitting**, by Hinton, G.E., Krizhevsky, A., Srivastava, N., Sutskever, I., & Salakhutdinov, R. (2014). Journal of Machine Learning Research, 15, 1929-1958. (cited 2084 times, HIC: 142 , CV: 536).