e-ISSN: 2582-5208

International Research Journal of Modernization in Engineering Technology and Science

(Peer-Reviewed, Open Access, Fully Refereed International Journal)

Volume:04/Issue:04/April-2022 Impact Factor- 6.752 www.irjmets.com

DESIGN AND DEVELOP A SYSTEM FOR A DRIVERLESS CAR TO IDENTIFY HURDLES BY USING A CONVOLUTIONAL NEURAL NETWORK

Dr. Ravindra Jogekar*1, Ayush Kamdi*2, Pratik Maske*3, Aman Mishritkotkar*4, Hardik Barbatkar*5

*1Assistant Professor, Department Of Computer Science And Engineering Priyadarshini J.L College Of Engineering - Nagpur, Maharashtra, India.

*2,3,4,5Students, Department Of Computer Science And Engineering Priyadarshini J.L College Of Engineering - Nagpur, Maharashtra, India.

ABSTRACT

The evolution of Artificial Intelligence has catalyzed the field of technology. We can now develop things that were once just imagination. One such creation is the birth of the self-driving car. Days have come where one can do their work or even sleep in the car and without even touching the steering wheel, or accelerator you will still be able to reach your target destination safely. This paper proposes a working model of an autonomous car that's capable of driving from one location to another location or staying on different kinds of tracks such as straight tracks, curved tracks, and straight followed by curved tracks. A camera module is mounted over the top of the car with Raspberry Pi and sends the pictures from the real world to the CNN which then predicts one of the following directions. i.e. left, right, forward, and stop which is then followed by sending a signal from the Arduino to the controller of the car and as a result of it, the car moves in the desired direction without any human interaction.

Keywords: Self Driving Car, Convolution Neural Network, Object Detection, Image Recognition.

I. INTRODUCTION

Globally speaking, nearly 1.3 million people die in road crashes each year, on average 3,287 deaths a day [6]. And talking about India the number of people who were killed in a road accident in 2013 alone was 1, 37,000 [7]. Speeding, talking over the phone, drunk driving, and breaking traffic rules are the root causes behind these accidents and the statistics are rising day by day which is now becoming a major concern. No matter how hard we try to create awareness regarding safety and traffic rules that has to be followed while driving, accidents are still occurring and are not showing a sign to stop. Though human errors can never be eliminated, accidents can be stopped. And in this case, technology has surely come to our rescue. Starting from the very early radar-based collision detection to present-day technology, the advancement and improvement in this technology had seen exponential growth in recent years.

Self-driving cars are one of the most discussed technologies in the current scenario. What was once imagined is a reality now. The definition of a self-driving car is a car that promises to take the traveler to their destination with minimal human control while taking safety as its first priority. Many companies throughout the world are making serious and continuous efforts to make driving a safe and risk-free process and have started building prototypes for the same. Amongst these companies are Google, Tesla Mercedes, and many more who have built a successful and functioning prototype and are planning to release a model in the upcoming years.

Self-driving cars are expected to have faster reflexes than humans, and make more reliable judgments, thus avoiding mere faults which cause accidents in the first place. Apart from saving precious lives, another advantage this technology gives is better traffic flow regulation because unlike humans these cars ride with proper traffic rules, making rides smooth and congestion-free. Self-driving cars can also help in tackling parking space issues by allowing to create of a taxi/pooling service for the unused cars and by unused car we mean to say the car that is either staying for a few hours while the owner is at work or the car that is in the garage while the owner is out for a vacation. Thus we could make better use of land rather than using it for car parking space. The basic model of any Autopilot system involves a front-facing camera, radar, long-range ultrasonic sensors located around the car, and a digitally-controlled digital braking system. Radars enable the detection of vehicles and other moving objects around the car, front-facing camera helps to detect and recognize objects like cars, trees, driving lane, humans, traffic signals, and other important data. All this information is taken in a real-time environment and is fused into a learning network which then predicts the car's response accordingly.

II. LITERATURE REVIEW

Autonomous cars consist of various sensors, they can be handled using several algorithms. These algorithms use various technologies. Currently, many countries are working on the development of autonomous cars. The growth of every country within the automobile industry is decided by advancements made within autonomous cars. Self-driving vehicles advancement involves different research and issues in the initial days[4]. Hence during this paper, we propose, information about what enhancement has done compared to the last century, new resource development for autonomous cars, and explains the technical and non-technical challenges and issues that autonomous car developers should face in the future.

Identification of Machine Learning Algorithm's on Self-Driving Car Navigation: Many statistical reports realized that over 94% of accident causes came from direct human causes like illegal overtaking, violating the speed limit, and suddenly cutting in. Therefore, autonomous vehicles were swiftly developed by starting a scaled RC-Car platform. During this research, they built a self-driving car for collecting data. Nvidia Jetson Nano is a small microprocessor board for developing and training models by using GPU 128-core Maxwell to rapidly process AI frameworks and models for applications like segmentation, object detection, and image classification [8]. For the training model, they used the three models, which are Artificial Neural Network Multilayer Perceptron (ANN-MLP), Support Vector Machine (SVM), and Convolution Neural Network - Long Short Term Memory (CNN-LSTM) [9] for comparison to find the simplest accuracy for self-driving car model (SDCM). The SVM can encourage both regression and classification problems, including linear and non-linear hyper- plane by using a kernel function to scale back complicated feature spaces. The ANN-MLP [9] is an artificial neural network and it's a nonparametric estimator to use for detecting and classifying objects. Convolution Neural Network Long Short-Term Memory (CNN-LSTM) is one of the models which is suitable for fixing classification problems, which consists of five main layers: fully connected stage, LSTM stage, Pooling stage, Detector stage, and Convolution stage [9]. They propose three scenarios and three-speed levels for comparing the accuracy of every algorithm: SVM, ANN-MLP, and CNN-LSTM. For the primary experiment, they set up the three-speed levels, which are 1, 2, and 3 km/h, respectively, without obstacles on the street.[1] According to the primary experiment, it will be seen that the percentage of the accuracy rate of the CNN-LSTM algorithm is the highest performance of all models at every speed level without obstacles on the street. As per the second experiment, although they add one condition to a scenario by adding an obstacle, CNN-LSTM is still the simplest accuracy algorithm. Within the final experiment, it's apparent that while we are including a few obstacles on the street, the accuracy of the Convolution Neural Network Long Short-Term Memory algorithm is higher than any algorithm within the experiment[5]. From the comparison algorithms of machine learning: CNN-LSTM, ANN-MLP, and SVM on different speed levels and different scenarios. From the paper, it can be seen that the percentage of the accuracy rate of the CNN-LSTM algorithm is the highest efficiency, not only with obstacles but also without obstacles.

III. METHODOLOGY

To enhance driver's safety at night time the algorithm includes lane detection with a vehicle recognition system. Lane detection helps to localize the markers in the order that the lane can be detected while vehicle recognition involves taillight extraction along with a taillight paring algorithm. An operation like canny edge extraction is done to extract an edge map to which a matching technique is applied followed by the selection of potentials edge points. Finally linking is done to localize the lane lines. Then flood-filling algorithm was applied to label the connected components. Then the biggest connected component is extracted which is called the lane. The model proposed by Gurjashan Singh Panna, Pritha Gupta, and Mohammad Dawud Ansari[10] in developing a prototype of the autonomous car involves the implementation of a lane detection algorithm along with obstacle detection. Their project aims to make a monocular vision autonomous car prototype that is capable of reaching a particular destination safely. The GPS steers the robot and is capable of reaching from one point to another without any human intervention. While in the former one with the help of the GSM system they promise to report a theft in the case there is any. An SMS alert is sent to the vehicle owner reporting the issue and as a result of it, the owner of the car can switch the ignition off in the latter one the project states that the vehicle can only be turned on if the authorized person sends a predefined location to the car. In [11] Dhanasingaraja R, Kalaimagal S, and Muralidharan G developed a system that takes the current position and gets the user destination. Then the system finds the shortest path to the destination and also extracts features like latitude, and longitude from the graph. So in an exceedingly nutshell it helps in navigation as well as monitoring the car. In the above-mentioned models, the art of taking decisions (left, right forward, or stop) was merely based on Image Processing techniques while some of them, it relies on a GPS to take the action. With all the work done before and after studying and analyzing thoroughly, the paper presents the latest model which involves Machine Learning as well as Image Processing. Image Processing helps in preparing the input image. The image is resized and converted into grayscale. Once it is completed the image is fed into the CNN[2]. The output of the Neural Network helps in taking the directional decision. Once it is done, the controller simply sends the corresponding signal, and therefore the car moves in that particular direction.

OpenCV is the biggest open-source library for machine learning, computer vision, and image processing and presently it plays an important role in real-time operation which is extremely important in current systems. By using it, one can process videos and images to identify the faces, objects, or even handwriting of a person. When it's integrated with different libraries, for example, NumPy, python is capable of processing the OpenCV array structure for analysis. To Identify picture patterns and their different features we use vector space and perform mathematical operations on these features. The first OpenCV version was 1.0. OpenCV is released under a BSD license and hence it is free for both **commercial** and **academic** use. It has C++, C, Python, and Java interfaces and supports Windows, Linux, Mac OS, iOS, and Android. When OpenCV was designed it mainly focuses on real-time applications for computational efficiency. All things are written in optimized C or C++ to take benefit of multi-core processing. Look at the following images :

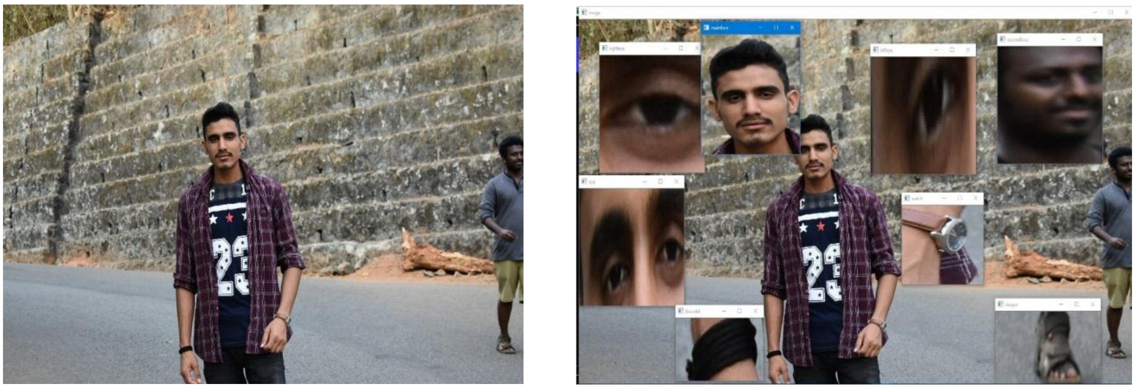


Fig 1: Image recognition process [12]

From the above original image, many pieces of data that are present within the original image can be obtained. Like in the above image there are two faces available and the person(I) in the image wearing a watch, bracelet, etc. So with the help of OpenCV, we can get all these types of information from the original image [12].

Raspberry Pi Operating System is a Debian-based operating system for Raspberry Pi. Since 2013, it has been officially provided by the Raspberry Pi Foundation as the primary OS for the Raspberry Pi group of compact single-board computers. Raspberry Pi Operating System is a desktop environment that looks similar to many common desktops like Microsoft Windows and macOS and is based on LXDE. The menu bar is positioned at the top and contains an application menu and shortcuts to File Manager, Terminal, and Chromium. On the right are a Bluetooth menu, volume control, a Wi-Fi menu, and a digital clock [13].

IV. PROPOSED MODEL

The proposed model takes an image with the help of a Pi cam attached with Raspberry Pi on the car. The laptop and the Raspberry Pi are connected to the same network, the Raspberry Pi sends the image captured which serves as the input image to the CNN. The image is gray-scaled before passing it to CNN. Upon prediction, the model gives one of the four outputs i.e. right, left, forward, or stop[3]. When the result's predicted corresponding Arduino signal is triggered it successively helps the car to move in a specific direction with the assistance of its controller. The Hardware involved in this project is a Raspberry Pi, camera module, Arduino microcontroller, and of course a JC kit (car). Let's discuss each hardware in brief.

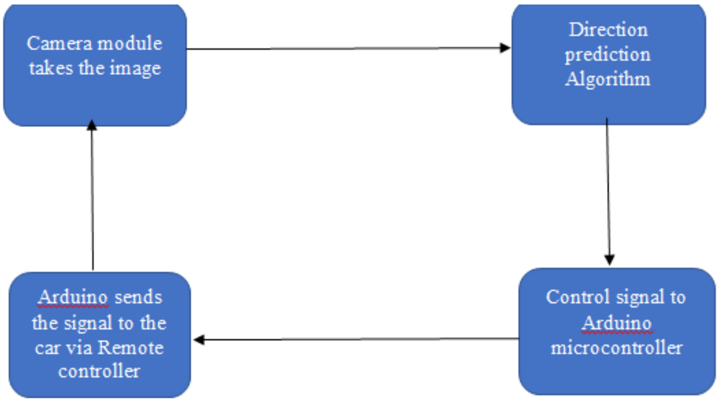

Camera module takes the image
Direction prediction Algorithm
Control signal to Arduino microcontroller
Arduino sends the signal to the car via Remote controller

Fig 2: Proposed self-driving model

Raspberry Pi :

The Raspberry Pi is a small low-cost single board computer having a processor speed ranging from 700 MHz to 1.2 GHz the Pi 3. The on-board memory ranges from 256 MB to 1 GB of random access memory. The board supports up to 4 USB ports along with an HDMI port. Along with all this, it has a number of GPIO pins that support protocols like I^{2}C. Moreover, it also supports Wi-Fi and Bluetooth facility which makes the device very compatible with other devices. It supports Scratch and Python programming languages [14]. It supports many operating systems like Ubuntu MATE, Snappy Ubuntu, Pidora, Linutop, and many more out of which Raspbian is specifically designed to support Raspberry Pi's hardware [15].

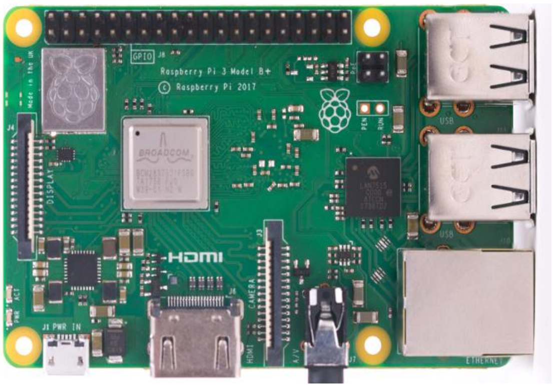


Fig 3: Raspberry Pi [20]

Camera Module:

The camera module is a great gadget to capture time-lapse, slow motion with great video clarity. The dimensions of the camera are 25mm to 24mm by 9mm, which connects to Raspberry Pi via a flexible elastic cord that supports the serial interface. The camera image sensor has a resolution of five megapixels and has a focused lens. The camera provides great support for security purposes. Various characteristics of the camera are it supports a 5MP sensor, Wide image, capable of 2592x1944 stills, 1080p30 video on Camera module v1 [16].

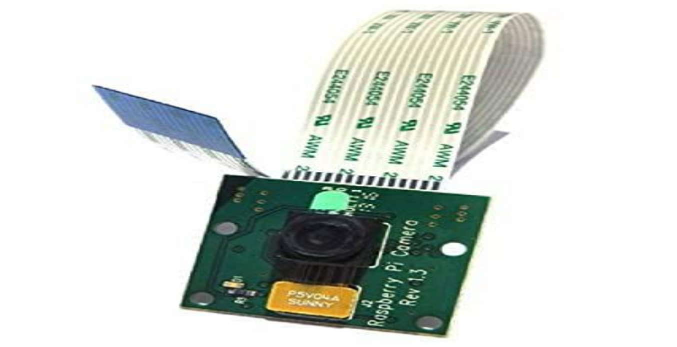


Fig 4: Pi Camera [21]

Arduino Microcontroller :

This microcontroller is based on ATmega329P. There are 14 digital input/output pins available out of which 6 can be used as PWM outputs. It also supports 6 analog inputs. It has a 16 MHz quartz crystal, a USB connection, an ICSP header, a power jack, and a reset button. It has 32 kb of flash memory and 2 kb of SRAM and weighs around 25g [17]. Apart from all these features Arduino IDE is very user-friendly and uses basic c as its programming language.

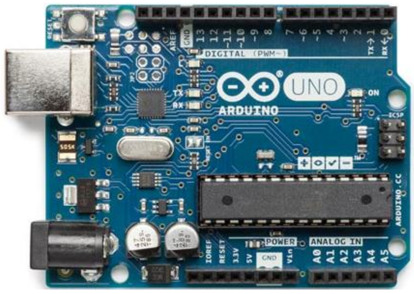


Fig 5: Arduino Uno [22]

After attaching this hardware to the car and connecting Arduino with the controller. The setup looks something like that shown in the figure.

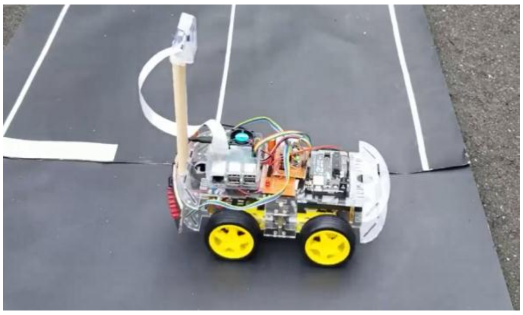
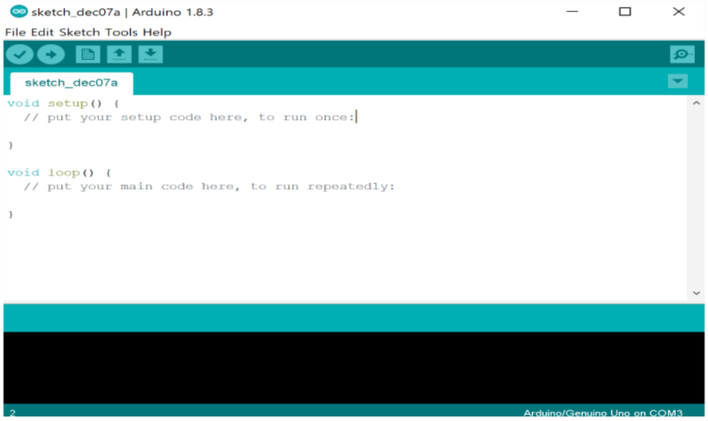


Fig 6: Overall hardware setup

These were all about the hardware needed to build a working prototype. Along with this hardware the software was used in this project. Are Raspberry Pi cam interface, Arduino IDE, OpenCV, and Spyder environment for developing machine learning model. Let's discuss each of these briefly.

Arduino IDE :

Arduino IDE is the platform where the programs are written for the Arduino board. It has compile button which helps in compiling the code along with the upload tab which helps to upload the code on the board. Programs written on Arduino IDE are often called Sketches and are saved as .ino extensions. The editor has numerous other features like verifying, saving, and uploading, including a library and serial monitor. Apart from this, the developers have made easy-to-use functions, which makes coding easy and fun. Moreover, there are several examples provided for every interface which helps the user learn more about functions and hardware as well.



sketch_dec07a | Arduino 1.8.3
File Edit Sketch Tools Help
sketch_dec07a
2
Arduino/Genuino Uno on COM3

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

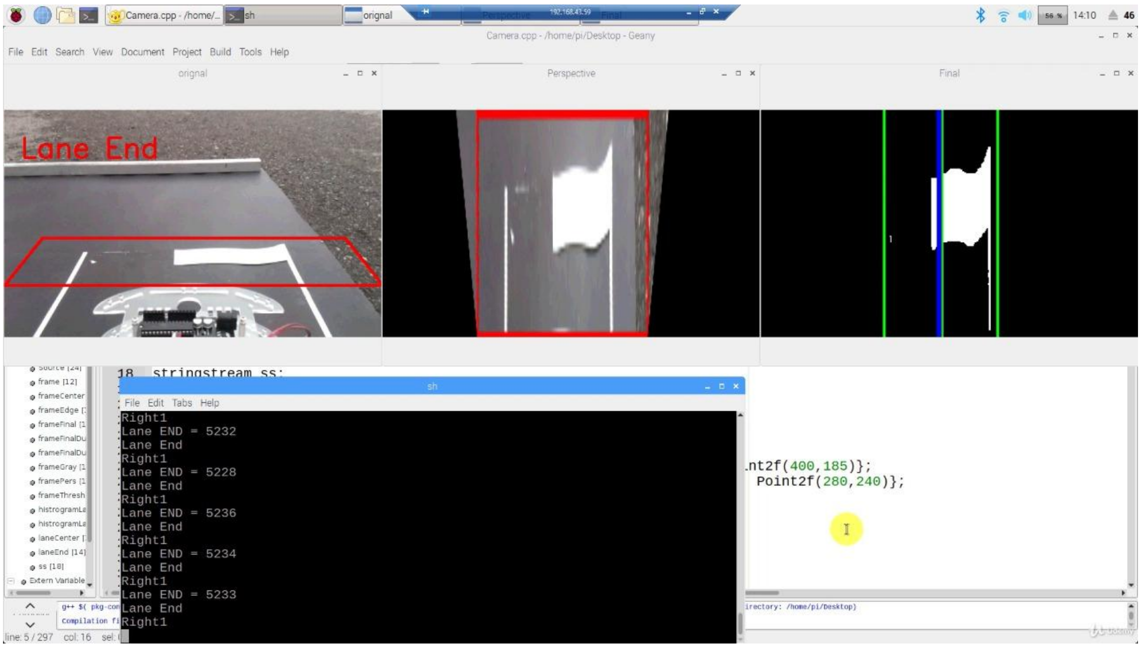
Fig 7: Arduino IDE [23]

OpenCV and Spyder environment :

OpenCV is an open-source computer vision library that is capable of handling images/videos from fairly basic tasks to utter complex tasks like facial recognition. It supports C++, C, Python, and Java programming languages and supports Windows, Linux, Mac OS, iOS, and Android. Written in optimized C or C++, the library can take the benefit of multi-core processing. Enabled with OpenCL, it can take advantage of the hardware acceleration of the underlying heterogeneous compute platform [18]. In this project it is serving as major support, it helps to crop out the section of the video from the Raspberry-Pi cam interface as shown above and converts it to the grayscale, resizes it, and then passes it to the Convolutional Neural Network. Spyder is a powerful interactive development environment for the Python language which has advanced editing, interactive testing, debugging and introspection features, and a numerical computing environment. It has a matplotlib as a plotting library which helps to plot 2D/3D graphs [19].

Raspberry Pi Cam Interface :

To remotely capture the live feed from the camera to the laptop we'd like to develop an interface that might serve this purpose. This is often where the software Rpi-cam interface comes into the image. It’s the program that helps you capture the live feed by just letting the IP address of the Raspberry pi. One can record and download video/image in various resolutions with several settings. The view of the software under action is shown below.



```
stringstream ss;
```

```
Right1
Lane END = 5232
Lane End
Right1
Lane END = 5228
Lane End
Right1
Lane END = 5236
Lane End
Right1
Lane END = 5234
Lane End
Right1
Lane END = 5233
Lane End
Right1
```

```
.nt2f(400,185)};
 Point2f(280,240)};
```

```
g++ $( pkg-con
Compilation fi
irectory: /home/pi/Desktop)
```

Fig 8: Raspberry-Pi camera interface

Data Flow Diagram :

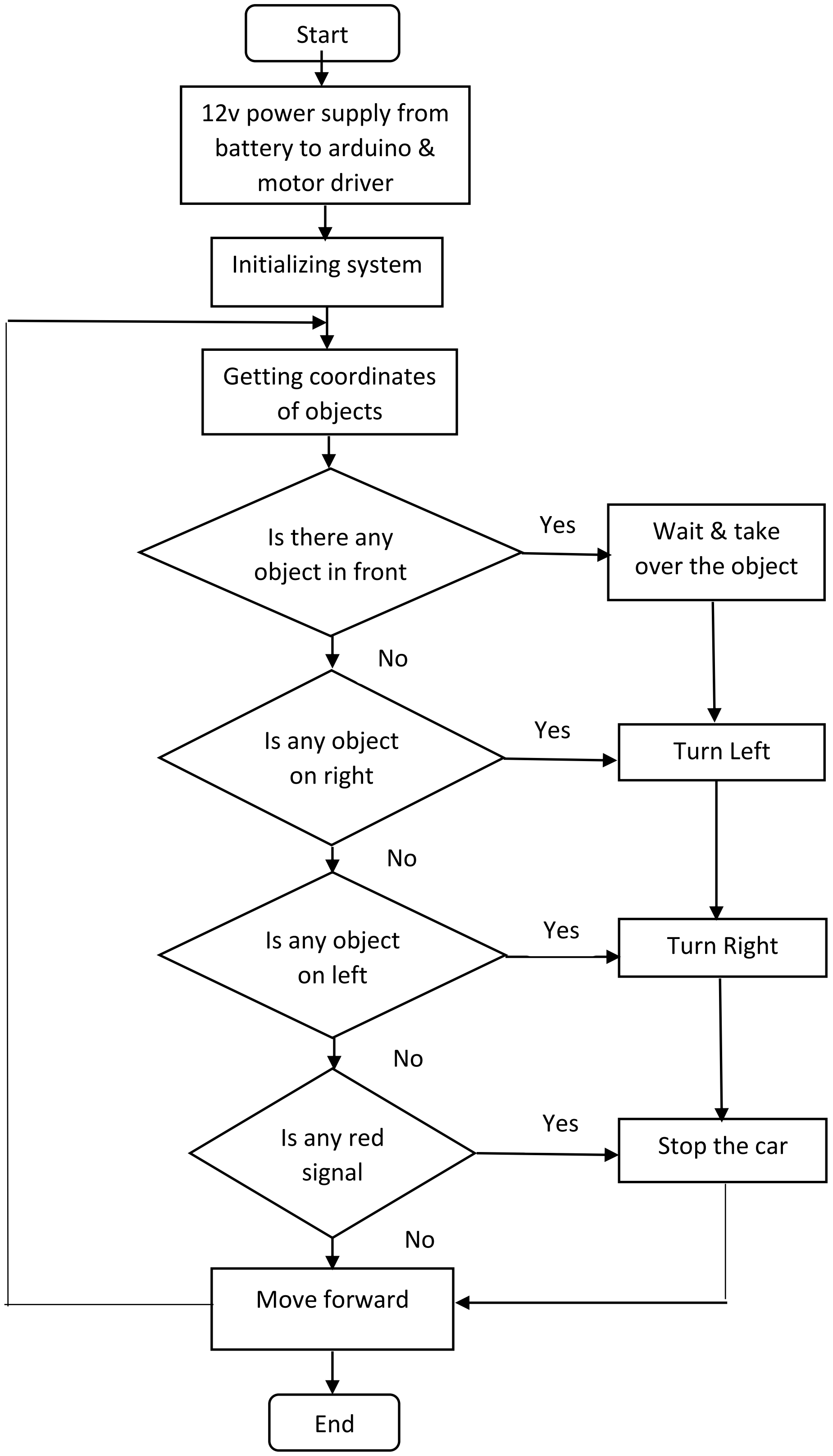


Fig 9: Data flow diagram

V. RESULT

The car was trained under different combinations of the track i.e. curved, straight, a combination of straight and curved, etc. A total of twenty-four videos were recorded out of which images were extracted. Ten thousand eight hundred and sixty-eight images were extracted and were categorically placed in different folders like right, left, straight, and stop. Below is the sample image of each of the scenarios in its gray-scaled version.

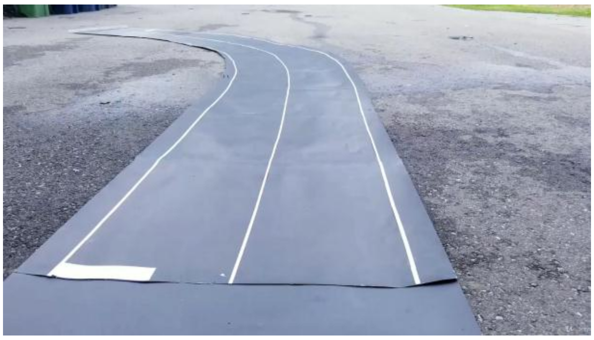


Fig 10: Sample image of the track

These images were resized to 320 x 240 and on which the network was trained. The Convolutional Neural Network had 128 input nodes, 2 hidden layers of 32 nodes each, and finally the output layer consisting of 4 nodes for each of the 4 outputs. To avoid overlearning the network dropout of 0.5 was considered. 'Relu' activation function was used between the input and hidden layers and the 'Softmax' activation function was used in the output layer. The Batch size was set to 10 and the number of epochs was set to 3 and it took 5-6 hours to train on the GPU mode. This was all about the network configuration that was used to train the model. Let's now see how well the car performed on each track.

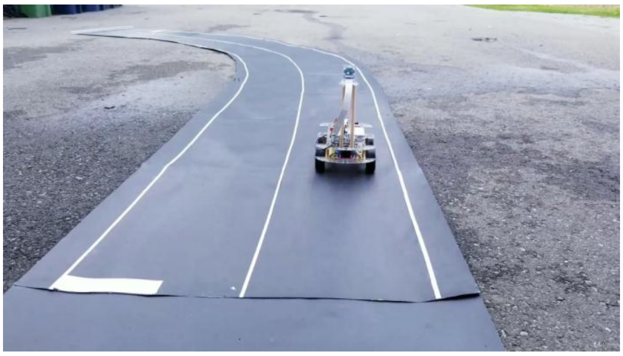


Fig 11: Car moving on track

VI. CONCLUSION

By using the YOLO algorithm, vehicles are often set to automatically navigate to the destination location by continuously identifying objects moving ahead to the same destination. The goal of the navigation process for a robotic vehicle is to move the robot to a known destination in an unknown environment. Navigation planning is a vital aspect of autonomous systems. When the autonomous vehicle starts to move towards the planned route it should find unknown obstacles from the present location to the destined location, hence the autonomous vehicle must avoid the obstacles and follow an optimal route to reach the destination. The potential applications of this robotic vehicle are to use these types of autonomous vehicles on heavy traffic roads or highways. These types of autonomous vehicles can be used when a driver travels to new areas. It's an improved navigation system for autonomous vehicles.

VII. REFERENCES

[1] Ravindra Jogekar and Nandita Tiwari, "unconventional technique for improving farmer yields by exposing and mitigating foliage diseases in an extensively adaptable deep learning and computational model through microbiological vegetation assessment" International Journal of Future Generation and Communication Networks. 2020;13(3): 3516-3526. ISSN: 2233-7857.

[2] Ravindra Jogekar and Dr. Nandita Tiwari, “Summary of Leaf-based plant disease detection systems: A compilation of systematic study findings to classify the leaf disease classification schemes” 2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4).

[3] Ravindra Jogekar and Nandita Tiwari, “A Review of Deep Learning Techniques for Identification and Diagnosis of Plant Leaf Disease” Smart Trends in Computing and Communications: Proceedings of SmartCom 2020, Smart Innovation, Systems and Technologies 182.

[4] Ravindra Jogekar and Dr. Nandita Tiwari, “Leaf-based plant disease detection systems depiction: An overview of outcomes of statistical studies for the identification of classification schemes for leaf diseases”

[5] Ravindra Jogekar and Nandita Tiwari, “Enhanced adaptive creation of visualisation network by detection of leaves” Materials Today: Proceedings.

[6] http://asirt.org/initiatives/informing-road-users/road-safety-facts/roadcrash-statistics

[7] https://sites.ndtv.com/roadsafety/important-feature-to-you-in-your-car-5/

[8] SadanandHowal, Aishwarya Jadhav, ChandrakirtiArthshi, SapanaNalavade, Sonam Shinde, “Object Detection for Autonomous Vehicle Using TensorFlow”, 2019.

[9] NVIDIA Autonomous Machine (Jetson Nano Development Kits). Retrieved from 2020, 17th International Conference on Electrical Engineering, Computer, Telecommunications and Information Technology (ECTI-CON)

[10] Pannu, Gurjashan Singh, Mohammad Dawud Ansari, and Pritha Gupta. "Design and implementation of the autonomous car using Raspberry Pi." International Journal of Computer Applications 113.9 (2015)

[11] Dhanasingaraja, R., S. Kalaimagal, and G. Muralidharan. "Autonomous Vehicle Navigation and Mapping System." Division of Mechatronics, 2014 International Conference on Innovations in Engineering and Technology. Vol. 3. No. 3. 2014.

[12] https://www.geeksforgeeks.org/opencv-overview/

[13] https://en.wikipedia.org/wiki/Raspberry_Pi_OS

[14] https://en.wikipedia.org/wiki/Raspberry_Pi

[15] https://www.hongkiat.com/blog/pi-operating-systems/

[16] https://www.raspberrypi.org/documentation/hardware/camera/

[17] https://www.arduino.cc/en/Reference/Board?from=Guide.Board

[18] https://opencv.org/

[19] https://pythonhosted.org/spyder/

[20] https://opensource.com/sites/default/files/uploads/raspberry-pi-3bplus-1.jpg

[21] https://www.amazon.in/Raspberry-5MP-Camera-Board-Module/dp/B00E1GGE40

[22] https://www.omsremodels.com/wp-content/uploads/2019/03/Arduino-UNO-R3-CH340G-02.jpg

[23] https://support.content.office.net/en-us/media/e8c360e1-2b32-45db-b9d7-d43abc86af2f.png