# RC-sized Autonomous vehicle with Prediction Model

Om Bheda
*Department of Information Technology*
*A.P. Shah Institute of Technology*
Thane, India
ombheda@apsit.edu.in

Fenil Bhimani
*Department of Information Technology*
*A.P. Shah Institute of Technology*
Thane, India
fenilbhimani@apsit.edu.in

Amey Mohite
*Department of Information Technology*
*A.P. Shah Institute of Technology*
Thane, India
ameymohite@apsit.edu.in

Prof. Apeksha Mohite
*Department of Information Technology*
*A.P. Shah Institute of Technology*
Thane, India
atmohite@apsit.edu.in

***Abstract*—Vision begins in eyes, but truly takes place in the brain. So, in today's world with the best high-definition cameras, high-speed computers, and artificial intelligence, computer vision was introduced. Computer vision is one of the latest advancements in technology that helps on giving the abilities of vision and understanding of the environment to computers so that they can extract high-level understanding from digital images and videos.**

**Autonomous vehicles are the application of computer vision with main motive of eliminating human errors and drive the vehicles with high precision and remove the efforts taken by a person to drive.**

**This paper provides a proof of concept for implementing an autonomous vehicle with a vehicle behaviour prediction model. It will be implemented over a small scale on a RC-sized car. The car would traverse autonomously in a demo map where we can simulate dynamic conditions and assess the model's performance accordingly.**

**This is the main motivation behind the project. To create a completely autonomous system capable of navigating around on its own. Detect, identify and follow traffic signs, signals and rules. Take appropriate actions for dynamic conditions and avoid any sort of loss. Provide user with an application interface to experience a safe journey to selected destination.**

**The proposed system will be using LiDAR for mapping the telemetry and detecting surroundings of the vehicle, a camera module to identify the objects detected, a development board with the deployed autonomous driving model, being the brains of all the operations, and actuators for moving the car around.**

**The autonomous driving model to be deployed, will be trained using YOLO for object detection and identification, along with a vehicle behaviour prediction subsystem assisting the dynamic decision-making subsystem to take some decisions based on the prediction to reduce severity of certain accidents or even avoid them, before they happen.**

***Index Terms*—Computer Vision, Artificial Intelligence, Autonomous Vehicle, Vehicle Behaviour Prediction model, RC-sized Car, LiDAR, Telemetry, YOLO, Decision-making subsystem.**

## I. Introduction

### A. Issues

Driving a motor-vehicle is one of the basic tasks performed by humans. Driving a car requires judgement, control over body, and constant observation of the things happening in the surrounding to traverse through the roads precisely. This task seems easy but a lot of human error takes place around us causing accidents, these accidents are caused by unawareness, drunk driving, incorrect judgements, etc.

Some part of the consumers consists of physically challenged people. They either require someone to help them move, from one place to another or they are physically able to drive a vehicle, but require heavy amount of modifications to be done to existing cars. Both of which costs a lot of human and monetary resources.

Computers are known for calculating and executing complex instructions repeatedly, faster and after appropriate training better than a human which can remove/minimise accidents and other problems with the help of Autonomous Vehicles.

### B. Proposed System

Our proposed system architecture will be a completely autonomous riding system scaled down to the size of an RC-car that will have its own –

- Environment sensing techniques for detection and identification of objects.
- Decision making system driven by traffic rules and selective moral/ethical values.
- Prediction model to predict certain movements in environment and prevent/minimise harm caused from severe accidents, by taking actions before they occur.
- Shortest path finding system to find the route with shortest distance to reach the destination

- Hardware system with actuators for smooth movement of car based on commands given by the decision-making system.
- Mobile Application that will have the options for selecting destination, starting and stopping, and video feedback from car's cameras. The rider in the car will also have override access to the cars control for emergency through the mobile application.

We will be using technologies like Embedded Systems for perceiving the environment and actuate the car based on these perceptions, Computer Vision for analysing these inputs and detecting identifying obstacles, road signs and symbols, and take dynamic actions for it [11] [13].

The main control of all these operations will be the Nvidia Jetson Nano development board, with powerful processing and tensor cores making it a perfect use for our application of the RC-sized autonomous vehicle.

LiDAR will be used for setting the telemetries of the surrounding world, and a camera module will be used to label these telemetries identified by the LiDAR into classes.

This entire model will be scaled down to a pre-determined and controlled map with manual exceptions to simulate the working in real life and train the algorithm extensively. The main objective of this is to provide a proof of concept for autonomous driving system with a prediction model for future implementation on an actual sized car that would be ready for deployment in real world.

## II. Literature Review

1) **Title** – Building a Self-Driving RC Car Master's Thesis
**Publication** – Master's thesis to University of Zagreb, 2020
**Author Name** – Ivan Oršolic´
**Conclusion** – [12]In the thesis the author created a self-driving RC car using Embedded Systems, OpenCV, DonkeyCar, Keras methodologies. The paper showed detailed description regarding the hardware and the software used in the project. It included comparisons, setting up of the environment and connections for the components used. It introduced to the DonkeyCar library that is developed for creating self-driving cars similar to that of the scope of our project. The entire project was quite similar to the ideas that we have and implemented some of the objectives that we used. Overall, the thesis paper solved a few doubts and discrepancies regarding the hardware selection and gave a new light on DonkeyCar.

2) **Title** – A Comparative Study on Machine Learning Algorithms for the Control of a Wall Following Robot
**Publication** – IEEE International Conference on Robotics and Biomimetics (ROBIO) – 2019
Author Name – Issam Hammad, Kamal El-Sankary, and Jason Gu
Conclusion – [7]The methodologies used by in paper were Keras, Scikit-learn, Monte-Carlocross-validation, Machine learning algorithms – Decision tree, Gradient Boost Classifier, Sup-port Vector Machines, KNN, Gausian Naive Bayes. The understanding from the paper included that according to the popular No Free Lunch Theorem, there is no golden machine learning algorithm that can outperform all the other machine learning algorithms in solving all possible problems. Identifying correct algorithm based on application is important. The paper provided comparison between the algorithms with different data-sets and showed the results for the best performing algorithm for the given application i.e. wall following robot.

3) **Title** – Autonomous Vehicles and Embedded Artificial Intelligence: The Challenges of Framing Machine Driving Decisions
**Publication** – XIII Conference on Transport Engineering, CIT2018
**Author Name** – Martin Cunneen, Martin Mullins, and Finbarr Murphy
**Conclusion** – [5]The paper covered the moral decisioning problems faced by algorithms while driving. How not having an emotional quotient may affect the weighted inputs and their corresponding outputs. This paper focused more on ethical challenges as well as the social dilemma that can revolve around autonomous cars and how over-fitting of data while training can make the AI model biased to certain aspects and how this problem is solved by having appropriate diversified datasets and sufficient training of the model.

4) **Title** – Comparative Analysis of Machine Learning Algorithms on Different Datasets
**Publication** – International Conference on Innovations in Computing (ICIC), 2017
**Author Name** - Kapil Sethi, Ankit Gupta, Gaurav Gupta, Varun Jaiswal
**Conclusion** – [16]In the paper the authors used MATLAB to simulate the working of different algorithms for varying datasets. The sensitivity and accuracy of NN, SVM and KNN are determined from the simulations. The schema used by the authors for the experiment was identified and as per the examination, SVM classifier contrasted better than KNN and NN with accuracy of close to 99.38

5) **Title** – Waymo Driver
**Conclusion** – [19, 20, 21]Waymo is an independent autonomous driving technology company established under Alphabet. Waymo Driver is their autonomous driving technology designed with safety and a lot of experience. The Waymo Driver is the world's most experienced driver which has travelled millions of miles on public road and billions of miles in simulation since 2009. The Waymo Driver drives 2 different vehicles i.e.

Waymo One ride-hailing service for moving people and Waymo Via focuses on transporting commercial goods. The designed system first requires the map of a new territory with high granularity, this not only assists in the autonomous working of the driver but also removes the reliability on external GPS data. The Waymo Driver has detailed custom maps to determine its exact location in real-time. It also uses its own perception system to collect and understand its surrounding using high advanced sensors and technologies like Machine Learning. This real-time data and the previous driving experience that the system is trained on, is used to anticipate the behaviour of other traffic scenarios and possible path that other objects can take. It understands how a car moves different than a pedestrian. Using all of this information the best plan of action is decided for the vehicle.

To execute the mentioned tasks the Waymo vehicle requires LiDAR sensors for giving a 3D picture of the vehicle's surroundings, RADAR sensors for providing details like an object's distance and speed, Cameras providing 360° view around the vehicle, and onboard computer with latest server grade CPUs and GPUs. It takes information provided by n-number of sensors on the car, identifies the different objects and plans a safe route towards your destination in real time. The carefully designed system also comes with back-up systems namely the secondary compute, Backup collision detection and avoidance system, Redundant steering, Cybersecurity, Backup power systems, and Redundant braking.

6) **Title** – Tesla Autopilot
**Conclusion** – [17, 18]Tesla cars come advanced hardware capable of providing Autopilot features, and full self-driving capabilities. Advanced safety and convenience features are designed to assist the drivers with the most burdensome parts of driving, it enables the car to steer, accelerate and brake automatically. Due to legal constraints the current autopilot features require active driver supervision and hence, the cars are not completely autonomous. The tesla autopilot has many different features namely –

- **Navigate on Autopilot** – Navigate on Autopilot suggests lane changes to optimize your route, adjusts so the vehicle does not get stuck behind slower traffic. It will also automatically steer your vehicle toward highway interchanges and exits based on your destination.
- **Autosteer+** – Using advanced sensors and computing power, the Tesla will navigate through tighter and complex roads effortlessly.
- **Smart Summon** – The car will navigate more complex environments and parking spaces, maneuvering around objects to come to the driver in a parking lot/garage.
- **Full Self-Driving Capability** – Tesla cars have the hardware needed in the future for full self-driving in almost all circumstances. From home to your destination the entire commute would be possible autonomously. The person travelling would just require to put a destination, the car would come to the pick-up and travel to destination choosing the optimal route, traffic and other parameters. On reaching destination the car will also automatically identify and park the car in the accurate spot.

Tesla autopilot is equipped with safety features that include Automatic Emergency Braking for detected objects that may collide with the car and automatically apply brakes accordingly, Auto High Beams automatically adjusts high/low beams as required, and Front Side Collision Warning that helps warn the driver for collisions that might occur with stationary objects/obstacles or slower moving traffic around the vehicle.

The cars are covered advanced sensors and Tesla Vision. The car has 8 surround cameras provide 360 degrees of visibility around the car at up to 250 meters of range. They include –

- 3 narrow forward cameras
  - A Wide 120-degree fisheye lens - captures traffic lights, obstacles cutting into the path of travel and the objects that are at close range.
  - A Main camera covering a broad spectrum of different use cases.
  - A narrow camera providing a focused, long-range view of the objects present at a distant.
- Forward looking side cameras, these are 90-degree side cameras for alerts against real-time exceptions that might occur.
- Rearward Looking Side Cameras monitoring the rear blind spots on both sides of the car,
- for safely changing lanes and merging into traffic. Rear View Camera is useful when performing complex parking maneuvers and also assists the autopilot for enhancement of optics.

Along with the camera modules the car is covered with 12 ultrasonic sensors that allow detecting objects that are both hard and soft to be able detect nearby cars and provide guidance when parking. The 12 sensors are strategically placed to cover all possible areas.

To understand and take actions for the data collected from all these different sensors and camera modules 3 onboard computers are present. These computer runs the Tesla-developed neural net which used to train and develop Autopilot. The tesla developed FSD and Dojo chips are the AI inference and AI training chips powering Dojo systems. The FSD Chip is designed to run the full Self-Driving software. These FSD chips by tesla are implementing small architectural improvements to maximize Performance-to-Power ratio, Robust

testing of the trained model along with evaluation for the functionality and performance. It has drivers and programs that focuses on performance optimization and redundancy by communicating with the chip.

## III. Design & Methodology

With the help of understandings from the literature review, the design and methodology to be followed for implementation, is discussed in this section. The implementation methodology can be divided into 3 different parts of development –

1) Mobile Application – To provide connectivity between the rider and the car, with the features of setting up source and destination for the journey, controlling the car movements in case of emergency and provide the required security from external threats particularly the cyberspace.
2) Car – The car would consist of the complete hardware system. It will include the sensors and actuators required to perceive the data and take actions for the same, and the development board for making sense out of the information collected by sensors and moving the car based on the understanding through actuators.
3) Autonomous Riding System – This system will include 3 sub-systems viz. Object Detection and Classification system, Decision Making system, and Vehicle Behaviour Prediction System. These 3 subsystems work together to form the autonomous riding system which will command the Car's hardware system to traverse through the maps.

Developing the parts and deploying it one after the other before the complete integration of the system would assist in problem-solving and better troubleshooting that might take place during the deployment.

### A. Mobile Application

The Mobile application will be developed using flutter and firebase. The flutter app will provide the required GUI for controls, camera feedback, map layout, and flutter will help implement the cross-platform app compatibility. Firebase will be used to authenticate rider credentials, before starting the journey.

After the rider is successfully authenticated the user will then get an option to select the sources and destinations of their journey. After selecting the journey, the car waits for the start command from the app, until and unless the start journey button is not pressed in the app the car will not start moving. When the user chooses a source and destination from the app, the information is sent to the car's on-board controller. This information is used to calculate the shortest path from the source to the destination w.r.t distance using the Dijkstra's shortest path algorithm.The model would be first trained to calculate distances from each location on the map to another location of our customized map and create a 2D graph of the map. Thus, during the training a graph along with distances as the weights are calculated and then the algorithm identifies the shortest path based on these weights. This will happen after the source and destination information is received by the car. For the scope of this project the path finding will be static, but it can be replaced with google maps API when required.

During the journey user can stop the car or can overwrite the system commands; this feature is given so that if any implausible condition occurs and if the machine gets confused on which thing to act-on then the user can take the decision. On starting the journey, user can get a complete view of car's location and camera POV on the app screen.

The Video loopback from the car's camera will be achieved using the WebRTC API by Google [4]. There are also other APIs like RemoteMe which provide browser support with JavaScript and CSS however, WebRTC was found to be more reliable and appropriate for the specified use-case. WebRTC deals with connecting two applications/devices using a peer-to-peer protocol. For communication between the peers ICE (Internet Connectivity Establishment) server configuration will be required. Using HTTP-based web API such as REST services the signalling procedure between the peers is achieved and the peers decided how and what data is to be shared among them [6].

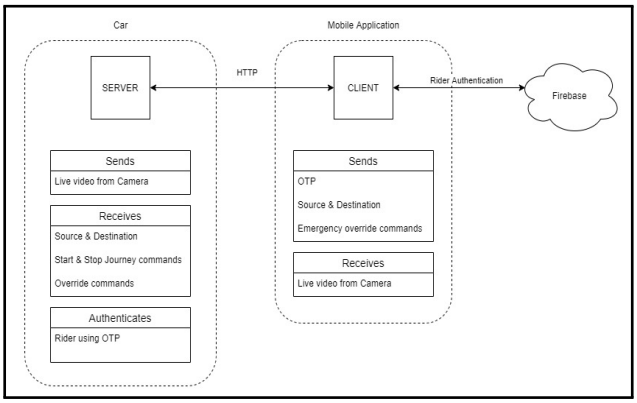


Fig. 1. Mobile Application Communication Block Diagram

### B. Car

The hardware platform for completely autonomous RC-sized car consists of the following components –

- Jetson Nano 2GB Developer Kit
- Battery Operated (BO) DC motors
- Servo motors
- PCA9685 – motor driver
- Electronic Speed Controller (ESC)
- LiDAR sensor
- Ultra-sonic sensors
- Camera Module
- Wi-Fi Module

These components are interfaced with the Jetson Nano developer kit using the GPIO. It is the brains of the entire car and will have all the control commands and autonomous systems deployed on it. Jetson Nano is a very advanced

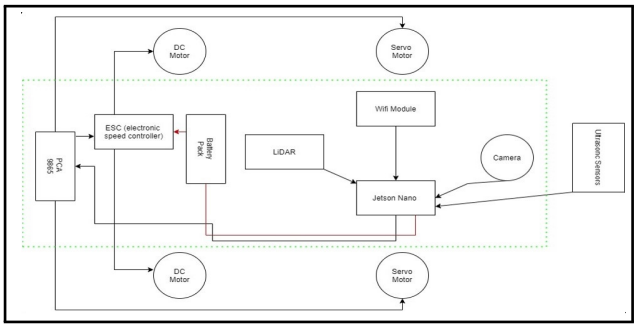


Fig. 2. Car Block Diagram

high-performance controller. It has dedicated Maxwell GPU, Quad-core ARM CPU, high speed RAM, and many other latest features in an affordable price range, providing better cost-to-performance ratio. The board is designed with the goal to deliver incredible AI performance at low price ranges and make the fields of AI and robotics more accessible. The board uses python for communication with the GPIO and programming use-cases. [12]Hence, making it the perfect choice for the application and scope of our project.

The chassis of the car can be either 3D-printed or be chosen from the readily available stencils in the market. With the facilities of 3D-printers available to us at institute we will opt for 3D-printing the base plates and the mounts. The wheels mounted to the motors will be bought from the market with standard designs.
The LiDAR and camera modules are the sources for all the observations being made in the surroundings of the car. The I/O ports and MIPI-CSI slot will be used to interface these sensors respectively. Python being open-source has easy to use and configurable libraries for interfacing these sensors to the board and start the perception of data instantaneously. The ultra-sonic sensors will be used to monitor objects in near range and also assists during parking the car. The sensors are digital sensors mounted on the GPIO pins of the Jetson Nano, and alert the decision-making system when they are in a particular range, for detecting and avoiding collisions.
After providing the eyes for our computer vision system, we require to provide actuators to move the car around. This is taken care by 2, BO DC motors with 200 rpm and 1kg/cm of torque to carry the weight of our entire car around, and the servo motors in the front to steer and manoeuvre the car. Thus, making the car a rear-wheel drive system. To control these parts, we will be using PCA9685 – motor driver and an electronic speed controller (ESC).
PCA9685 is a 16-channel PWM motor driver with 12-bit resolution. This driver communicates with the Jetson Nano over I2C protocol using 2 pins, then the on-board PWM controller drives the motors using PWM over 16 channels, making it possible to control up to 16 motors using just 2 pins. Based on the precepted information the commands to move as per the respective situation will be given by Jetson Nano through I2C, this will then be converted into PWM and be given to the ESC for controlling longitudinal movements, the ESC will then supply power to the motor accordingly. The motor driver also takes care of the servos directly to control the cars steering.
To connect the car with the rider's app we will be using the Wi-Fi module for providing peer-to-peer connectivity to the Jetson Nano. The car would take commands and act upon them and it will continuously feedback live data from the camera module to the mobile app. The Wi-Fi module connects to the M.2 slot on the developer board.

### C. Autonomous Riding System

The autonomous riding system gives the car the capability of driving autonomously. It takes inputs from sensors, detects and identifies objects, obstacles, road signs, traffic rules, and based on these received inputs it predicts the behaviour of the traffic around the car and chooses the most efficient and safe path to travel to the destination. After choosing the path it commands the connected actuators that are the motors, to move according to the selected travel path. To perform these tasks efficiently there are 3 subsystems within the Autonomous Riding System –

#### 1) Object Detection and Classification:

The object detection and classification will be done using YOLO [15] [14]. YOLO stands for You Only Look Once. It is a state-of-the-art, real-time object detection system. YOLO algorithm employs convolutional neural networks (CNN) to detect objects in real-time [11]. The CNN is used to predict various class probabilities and bounding boxes simultaneously. These bounding boxes are weighted by the predicted probabilities.
Using the 3D-mapped data of the surrounding by the LiDAR and the cameras along with YOLO to classify the surrounding objects, we can keep a track of the traffic around the car [10] [2] [3]. This subsystem helps us with the very first part of any autonomous vehicle i.e. making the car's on-board processing device have the ability of vision.
[9]In this system we will classify all detected objects into the following classes and sub-classes inside them –

- Traffic Users – this class comprises of all the road users along with the autonomous car that will occupy the road area.
  - Vehicles – All 2, 3, and 4-wheel vehicles will be identified as vehicles.
  - Pedestrians – People and animals on the road travelling on foot will be classified as Pedestrians.
- Traffic Rules
  - Mandatory Signs – These include signs that have to compulsorily be followed by the car. E.g. – stop, speed limits, horn prohibited, no entry, etc.
  - Cautionary Signs – These include signs to inform in advance the potential hazards that might me on the road ahead. E.g. – narrow road ahead, pedestrian crossing, school ahead, right hand curve, etc.

- Informatory Signs – These include signs to inform the upcoming halts, places for rest, communication and public convenience, etc.
- Traffic Lights – These are the lights at intersection that command to stop, move, turn and go ahead slowly. These are important for managing traffic at complex places.
- Road Lines – Road Lines are the lane divisions, zebra crossing, curbs, stop lines, and other markings on road that help in understanding of the path to be taken.

Based on these objects and their classes detected flags will be set, these flags would be used by the decision-making system to decide the course of action and/or the change in the course of action upon detection of the same. The bounded objects on the detected objects from the output of this subsystem is then passed on to the vehicle behaviour prediction subsystem.

*2) Vehicle Behaviour Prediction:*

The detected traffic users are taken as an input by the vehicle behaviour prediction model. Vehicle behaviour is one of the most unstable parameters to compute in an autonomous vehicle. Predicting the movements of other traffic users may it be cars or pedestrians has been a crucial challenge faced to achieve fully autonomous vehicles.

[8]We will be using the prediction dataset provided by Lyft's Level 5 Open Datasets. These datasets consist of the data for 1185 hours recording of traffic agent movement. These datasets are ready to use and very easily configure. The training model suggested for creating vehicle behaviour prediction uses Bird's-Eye View (BEV) rasterization for system's input, with top view of the road [1]. A base model with standard CNN architecture can be easily used for this purpose [13]. The only requirement is to match the input and output layer to match our real-time settings.

The model is trained by assigning unique IDs to the detected objects and tracks them through multiple captured frames to record the movements and what action is taken by these traffic agents in different scenarios. This subsystem now takes inputs from the surroundings, converts them into a BEV input by taking into consideration for the telemetry of our car. It then predicts and plots the trajectories of these different classes of objects around the car.

*3) Decision Making:*

Decision making subsystem is the final step and the part that combines all the data from other subsystems and sensors, based on which it commands the actions that needs to be taken. This system has defined set of rules that need to be followed to keep the rider and the neighbouring traffic agents safe.

Consider a particular scene on road. For the decision-making system to decide a certain set of actions, the subsystem considers the present position of the car, the scenario around its surroundings, the behaviour of other vehicles, traffic signs and boards in the scene, [5]and the set of moral/ethical rules defined. After assessing all of these considerations the subsystem decides the path that it will be following along with other actions like, the speed, steering angle, etc.

For calculating the values for the car's movements along the longitudinal and lateral axis, a simple physics model will be used to calculate these values. These parameters that need to be set to move the car along is decided path is converted into a format that can be understood by the actuators and sent to them. The speed of the car is set by changing the duty cycle of the DC motors using PWM signals. The servo motors also take PWM signals to rotate at a particular degree and in which direction. The Jetson Nano sends these calculated values to the PCA9685 – motor driver via I2C protocol and the motor driver then distributes the PWM signals to the respective motors through its PWM channels.

Thus, all these subsystems work together to handle all the real-time data to plan and perform the plan of action for normal journey and the dynamic exceptions that might come along the way within fraction of seconds helping the car achieve its goal to be completely autonomous with a vehicle behaviour prediction model.

## IV. Conclusion

By this we understand the design and methods of implementing a fully autonomous RC-sized car. The technologies that will be required in achieving the objectives set are made clear. This project seems to be a great way in which one can learn about technologies like Embedded Systems, Machine learning, and Autonomous Vehicles as a whole. The project might seem difficult and costly to implement and orchestrate however, taking steps one at a time and careful implementation and integration of all the systems will make the things easier. The use of open-source techniques for the implementation of the subsystems is made clear. The literature review from different published papers gave a clear idea on the different perspectives of implementing autonomous systems. The data sets of traffic agents and different traffic scenarios form these implementations and their experiences is used in training the model and creating rules for the vehicles, making our model be prepared for many scenarios in advance and avoid under-fitting of data to these trained models. Prediction models have become a necessity, a simple autonomous riding system along with faster processing is not enough to fulfil the rising demands of today's traffic scenarios. Predicting future movements increases efficiency and reliability on the system greatly.

## V. References


[1] Woven Planet Level 5. *How to Build a Motion Prediction Model for Autonomous Vehicles*. 2021. URL: https://medium.com/wovenplanetlevel5/how-to-build-a-motion-prediction-model-for-autonomous-vehicles-29f7f81f1580.

[2] Jason Brownlee. *How to Perform Object Detection With YOLOv3 in Keras*. 2021. URL: https://machinelearningmastery.com/how-to-perform-object-detection-with-yolov3-in-keras/.

[3] Manish Chablani. *YOLO — You only look once, real time object detection explained*. 2021. URL: https://towardsdatascience.com/yolo-you-only-look-once-real-time-object-detection-explained-492dc9230006.

[4] Google Developers. *Real-time communication for the web*. 2021. URL: https://webrtc.org.

[5] Martin Cunneen Finbarr Murphy Martin Mullins. "Autonomous Vehicles and Embedded Artificial Intelligence: The Challenges of Framing Machine Driving Decisions". In: *Applied Artificial Intelligence* 33.8 (2019), pp. 706–731. DOI: https://doi.org/10.1080/08839514.2019.1600301.

[6] Flutter. *Work with WebSockets*. 2021. URL: https://flutter.dev/docs/cookbook/networking/web-sockets.

[7] Issam Hammad, Kamal El-Sankary, and Jason Gu. "A Comparative Study on Machine Learning Algorithms for the Control of a Wall Following Robot". In: *IEEE International Conference on Robotics and Biomimetics (ROBIO)* (2019), pp. 2995–3000. DOI: https://doi.org/10.1109/ROBIO49542.2019.8961836.

[8] John Houston et al. "One Thousand and One Hours: Self-driving Motion Prediction Dataset". In: *CoRR* abs/2006.14480v2 (2020). arXiv: 2006.14480v2. URL: https://arxiv.org/abs/2006.14480v2.

[9] Andreas Peeter Lätt. *2D Detection on Waymo Open Dataset*. 2021. URL: https://medium.com/@lattandreas/2d-detection-on-waymo-open-dataset-f111e760d15b.

[10] Great Learning. *YOLO object detection using OpenCV*. 2021. URL: https://www.mygreatlearning.com/blog/yolo-object-detection-using-opencv.

[11] Analytics India Magazine Pvt Ltd. *Getting Started With Computer Vision Using TensorFlow Keras*. 2021. URL: https://analyticsindiamag.com/computer-vision-using-tensorflow-keras/.

[12] Ivan Oršolic´. *Building a Self-Driving RC Car Master's Thesis*. 2020. URL: https://bib.irb.hr/datoteka/1071914.5e4dc43800410_Building_a_Self-Driving_RC_Car_-_Ivan_Orsolic_0016114170_Masters_thesis.pdf.

[13] Savaram Ravindra. *The Machine Learning Algorithms Used in Self-Driving Cars*. 2021. URL: https://www.kdnuggets.com/2017/06/machine-learning-algorithms-used-self-driving-cars.html.

[14] Joseph Redmon and Ali Farhadi. "YOLOv3: An Incremental Improvement". In: *CoRR* abs/1804.02767v1 (2018). arXiv: 1804.02767v1. URL: https://arxiv.org/abs/1804.02767v1.

[15] Joseph Redmon et al. "You Only Look Once: Unified, Real-Time Object Detection". In: *CoRR* abs/1506.02640v5 (2016). arXiv: 1506.02640v5. URL: https://arxiv.org/abs/1506.02640v5.

[16] Kapil Sethi et al. "Comparative Analysis of Machine Learning Algorithms on Different Datasets". In: *Circulation in Computer Science International Conference on Innovations in Computing (ICIC)* (2017), pp. 87–91.

[17] Tesla. *AI Technologies*. 2021. URL: https://www.tesla.com/AI.

[18] Tesla. *Autopilot*. 2021. URL: https://www.tesla.com/autopilot.

[19] Waymo. *Waymo Driver*. 2021. URL: https://waymo.com/waymo-driver/.

[20] Waymo. *Waymo Driver Technology*. 2021. URL: https://waymo.com/tech/.

[21] Waymo. *Waypoint - The Official Waymo Blog: Designing the 5th-Generation Waymo Driver*. 2021. URL: https://blog.waymo.com/2020/03/designing-5th-generation-waymo-driver.html.