SW5

AALBORG UNIVERSITY
STUDENT REPORT

Embedded Systems

Aalborg University

Group SW508e19

Software 5: "Embedded systems"

Group SW508e19

19. December 2019

AALBORG UNIVERSITY
STUDENT REPORT

Semester:
5. semester

Project theme:
Embedded Systems

Project duration:
02/09/2019 - 19/12/2019

ECTS:
15

Supervisor:
Olivier Paul Pelgrin
olivier@cs.aau.dk

Authors:
Alexander Jankowski
ajanko17@student.aau.dk
Jeppe Skovby Bjørn
jbjarn17@student.aau.dk
Jonas Khooshabi Rodell
jrodel17@student.aau.dk
Malte Zoëga Andreasen
mzan16@student.aau.dk
Ronnie Svendsen
rsvend17@student.aau.dk
Tobias Kastbjerg Hauge Nielsen
tkhn17@student.aau.dk

Number of pages: 56

Abstract:
Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Preface

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Resume

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

List of Figures

List of Tables

Contents

Todo list

Part I

Introduction

1 Introduction

Autonomous vehicles or self-driving cars are one of the hot topics in today's car industry as well as in the general media [1]–[3]. The technology has improved over the last 10 years, however it has been an active point of interest for researchers and companies alike for a much longer time. Already in 1925 the first attempts at autonomous driving began, with a radio-controlled car with no one behind the steering wheel [4]. In the 1980s, successful attempts at making self-driving robots were made at the Carnegie-Mellon University, where the robots were able to avoid obstacles and follow roads [5]. In modern day, several industry giants have made their contribution to the field, among these are Google and Tesla. Google started working on a self-driving car project called Waymo in 2009, and, as of July 2019, it has driven autonomously for over 10 million miles in the real world and 10 billion miles in simulation [6], [7]. Tesla has also been pursuing the development of fully autonomous vehicles and Elon Musk, the CEO of Tesla, has proclaimed that they will be ready with a self-driving taxis by the end of 2020 [8].

Humans have always pursued technologies to make life more convenient for them, of which self-driving cars is one of them, however, there are several other benefits, besides conveniences. One of these is the potential avoidance of traffic accidents. Cars and taxies are the top contributors to fatalities in traffic accidents, up to 47% are caused by these in Europe [9]. The most common cause of traffic accidents is timing errors, where up to 51% of car accidents were caused by the failure of humans to respond in a timely manner in the traffic [10]. A closer look on Great Britain's cause of accidents, show that a very common issue leading to accidents is simply that the driver is not orienting properly when driving a car [11].

TODO: Add more later, when more of the report is written - revisit above text and potentially add more.

Problem Analysis 2

2.1 Problems in traffic

When driving cars in traffic, problems can occur. These problems include, but aren't limited to, traffic congestion and traffic accidents [9], [12]. In this section, the above problems will be introduced and described.

2.1.1 Traffic congestion

Traffic congestion usually happens when a road is in heavy use and is usually characterized by slower speeds and vehicles queueing [13]. This is, however, a very vague definition of traffic congestion, so a more practical definition of traffic congestion has been found. Traffic congestion is "*the condition in which the introduction of one vehicle into a traffic flow increases the delay for the remainder by more than x%*" [12]. In other words, traffic congestion happens when a new vehicle enters traffic, and the other vehicles are delayed by a certain amount.

Traffic congestion is largely caused by cars and private car users. Cars typically have fewer passengers than most forms of public transport, which means that they take up more space per passenger. While public transport also causes congestion this is typically less than cars, as long as there is not an excessive amount of vehicles in use. Additionally, the behavior of some drivers can also exacerbate the problem. When drivers show little to no respect for others, it can block traffic, an example of this could be a situation where two vehicles have to merge. If one vehicle has the right of way, and the traffic is heavy, the lack of respect could result in no vehicles, from the lane that has to give way, would be able to merge into the other lane. Poor road design and maintenance also have an effect on the degree of traffic congestion, as each new vehicle, that enters traffic, slows traffic even more. An example of poor road design could be placing a bus stop where the road narrows down, meaning other vehicles have to stop when the bus stops. [12]–[14]

Traffic congestions can have several different effects. These effects do not just impact the cars and drivers in traffic, but also have an impact on the local environment and the economy. Apart from the delays suffered by the vehicles in traffic, traffic congestions also cause noise and air pollution, which means all people in the area are affected. It also has economic effects, such as higher bus fares. [12], [13]

2.1.2 Traffic accidents

According to the "Annual Accident Report 2018" made for the European Union (EU), there were more than a million traffic accidents in the EU in 2016 [9]. Among these accidents, there were 25,000 fatalities, which makes it 51 fatalities per million inhabitants in the EU. Similar numbers were seen in USA with 37,000 motor vehicle related fatalities [15]. While fatalities in the EU have fallen by more than half since 2001, they still are not in line with the EU's target for 2020 [16].

Among the 37,000 fatalities in the USA, 10,000 were related to drunk driving, another 10,000 were caused by speeding drivers[15]. All these accidents are related to human errors. In a book from Danish Road Traffic Accident Investigation Board (AIB) "Why do road traffic accidents occur?", 291 traffic accidents were investigated, to determine the causes of said accidents [17]. It was found that in all but one accident, there was at least one factor related to the road user that caused the accident. In 160 accidents, it was found that factors relating to the road user was the only factors in causing the accident. The exact findings can be seen in Figure 2.1.

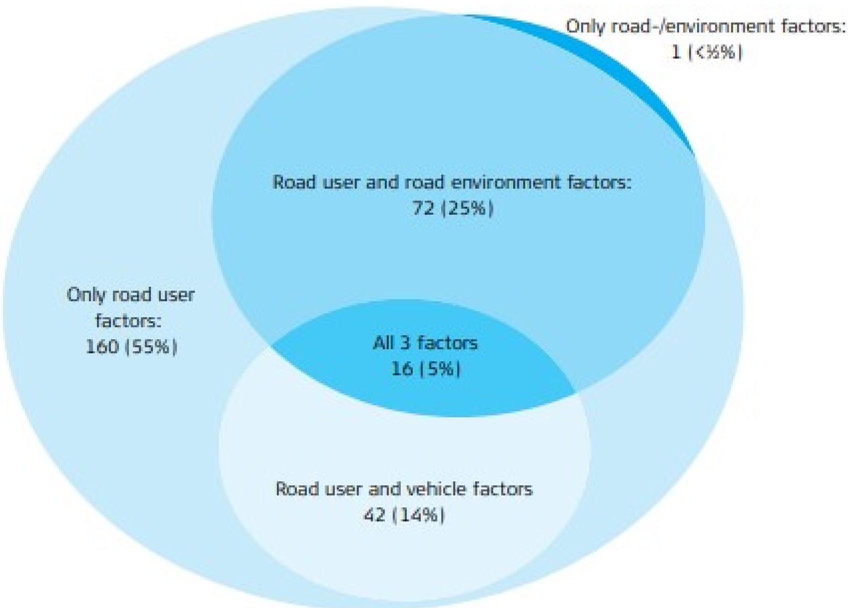


Figure 2.1: Accident factors for investigated accidents[17]

2.1.3 What can be done?

This section examines the findings of section 2.1, in an attempt to find a way of reducing the problems in traffic. This is done to narrow down the problem domain to a more manageable size.

In the previous section 2.1, two issues (congestion and accidents) in traffic were described and analyzed. It was found that these issues can lead to serious problems. These problems were anything from delays to loss of human life. While there were various factors involved in causing these problems, one factor that seemed to have an especially big impact was human error. It was seen that human errors were very frequent in traffic accidents, and while it did not have the same impact on traffic congestion, it still was a factor. One way to lower the number of human errors in traffic would be to reduce the influence of people in traffic. This can be done with self-driving cars or encouraging people to use public transport.

A self-driving car is a car that can control, or partially control itself, independent of human intervention. A more in-depth description of self-driving cars can be found in section 2.2. By implementing self-driving cars into traffic, it would be possible to lower the amount of human errors, or even entirely remove them, with a fully self-driving car. Additionally, using fully self-driving cars would enable the previous driver, now passenger, to use the time spent commuting on other activities. A fully self-driving car could potentially reduce the amount of hours spent in traffic by 250 million, in the 50 most congested cities[18].

If more people start using public transport, it would mean fewer private cars on the road, leading to less traffic. It would also mean that fewer people have a direct influence on traffic, as they are now passengers on a bus or train. There is, however, no guarantee that people are willing to use public transport over their own vehicles, as public transport tends to be less flexible, e.g. busses drive on fixed routes between bus stops. Even with less people in traffic, bus drivers are still capable of making mistakes, and any mistakes leading to accidents could endanger the bus passengers.

It is already possible to find a degree of self-driving technology in some cars today [19]. Despite the possibility of lowering the number of human errors, there is currently no full fledged self-driving car available commercially. As such the next section will further investigate what a self-driving car is.

2.2 Self-driving cars

There are different factors that can have an impact on the way traffic flows and in some cases whether people get hurt or not. One factor that is seen in all vehicles on the road today is the human. This factor cannot be precisely determined, because not all humans react with the same speed or in the same way. Other things, like alcohol or lack of sleep, can in most cases have a negative effect on this factor. [15]

The human factor can therefore make traffic and vehicles more or less unpredictable. If an intelligent system should control and maybe predict traffic flow, to avoid congestion and accidents, the vehicles on the road must be as close to completely predictable as possible. A possible solution to this, is to remove the human factor from the equation, by making the car drive itself.

2.2.1 The six levels of self-driving cars

There are 6 different levels of self-driving technology according to the US National Highway Traffic Safety Administration [20]. These different levels are as follows:

A **level 0** vehicle is not able to anything by itself. The driver is in charge of everything that is happening.

At **level 1**, the vehicle is able to assist the driver. An example of a feature that could be a part of a level 1 vehicle is adaptive cruise control.

When multiple automated features are combined it is a **level 2** vehicle. At this level the vehicle could, as an example, steer direction and control acceleration at the same time. The driver must be able to take control of the vehicle at any time, though.

At **level 3**, the vehicle is able to monitor the environment around the vehicle. If the vehicle get into a situation that might be confusing, the vehicle will return the steering to the driver.

Level 4 describes an almost self-driving car. It is able to read the environment and take action upon these inputs in most conditions. There will be situations where the driver needs to take control of the vehicle.

For a vehicle to be 100 percent self-driving it needs to be at **level 5**. At this level, the vehicle is able to be in control and handle every situation, no matter the conditions. The driver is still able to take control of the vehicle, but there will never be a situation where this is necessary, this means that the human factor can be eliminated.

Most production vehicles do not exceed level 2, with a few exception like the Audi R8 which is a level 3 vehicle. Even Tesla's Autopilot is categorized as a level 2. This does not mean that no high level Autonomous Vehicle (AV) exists, however. Companies like Google have developed level 4 vehicles and several concept cars are at level 5. The reasons for the absence of high level production AVs will be discussed in section 2.3. [21]

2.3 Implementation Issues With Self Driving Cars

This section will explore the reasons why self-driving cars are not yet implemented fully in commercially available cars. While Google has, in certain areas like California, employed a driverless taxi service with the Waymo project, which is around a level 4 or 5 on the autonomy scale. There are still several hurdles to overcome before this will be widespread, especially for cars that are available for private purchase [22]. The analysis in this section will explore these issues and focus on issues within the EU and US. In Asia, there are a lot of additional issues due to irregular traffic behavior by humans, and in some places, very poor infrastructure and environment for self-driving cars [23].

2.3.1 Infrastructure issues

Road infrastructure and maintenance are important elements of a successful widespread introduction of self-driving cars. They require that the different types of road markings, like lanes, edges, and speed limit signs, are highly visible and well maintained so that they are not faded and appear clear and visible for the self driving cars' sensors to detect and interpret. Furthermore, global standardization of road markings would also help the self-driving cars. Other issues pertain to temporary road markings related to construction work and accidents, that are often poorly marked. Furthermore, the precise and real-time communication of traffic incidents to the self-driving car is also helpful. These are some of the issues that make it harder for self-driving cars to make decisions and navigate correctly in a highly changing and volatile environment, there is a plateau of other interesting infrastructural issues which are further elaborated on in [24]. [24]

2.3.2 Ethical and safety issues

Ethical issues are also an important element regarding self-driving cars. One of the issues regarding this is the moral dilemmas when facing an inevitable car crash. This does not have as big an influence when it comes to car crashes involving a human driver, since the person driving the car, mostly react per reflexes and does not have time to thoroughly analyze the situation. With self-driving cars questions such as "how to avoid a crash" comes more into play. How should the car react if saving the driver means driving into other people and potentially killing them? This obviously poses a big moral dilemma for designers of self-driving cars on how the self-driving car should react when facing a car crash. This is regarding, whether an AV should use a self-preserving or self-sacrificing decision of the car crash. Or maybe even use a utilitarian way, where the car seeks to make decisions based on maximizing utility. A problem with this could be the introduction of self-driving cars, with different ethical preferences. An example of this is described in [25], where a self driving car has strong self-preservation ethical settings. This car would choose its passengers possibly over a school bus full of children. [25]

Along with ethical issues, there are also safety issues. One of these safety issues is the interlacement of self-driving cars and conventional cars and that the predictability of movement is different for people contra machines. The reason for this is that people tend to better anticipate other peoples behavior than machines behavior even though self-driving cars should be more predictable than people, since they would always follow the rules, whereas people might not follow the rules, but more so their common sense. This could lead to misunderstandings in traffic which in turn could lead to traffic accidents. Another example of the difficulties for machines to predict human behavior in the traffic is that there can be unwritten road rules that involve tacit knowledge which can be hard for computers to identify or anticipate. [26]

Another safety issue regarding self-driving cars is the possibility of hacking the self-driving car's computer. By doing this, the potential hacker could get access to the car's sensors, cameras, etc., and disrupt its view of the world, which could lead to crashes. This could especially be damaging if they hacked a network of self-driving cars, as the Connected Autonomous Vehicles (CAVs) mentioned in subsection 2.3.4. [27]

Both ethical and safety issues are in large regulated and therefore up to policymakers, and regulation has been one of the big hurdles for self-driving cars and continues to be. In subsection 2.3.3, the regulatory landscape will briefly be elaborated on.

2.3.3 Policy Issues

As a consequence of the ethical and safety issues, it is also paramount to the widespread introduction of self-driving cars, that proper legislation is provided. This remains one of the large roadblocks for further expansion of self-driving cars. This is in large due to lobbying against self-driving cars, as well as ethical and safety concerns. In the USA, 20 states have enacted their own legislation related to self-driving cars and in EU several countries, e.g. UK, Germany, Spain, France, and Italy, have introduced related laws. Except for Spain, these, however, are not aimed at level 5 autonomy with no driver behind the seat. The many different regulations from different countries, or states in the US, also provide issues, because there is no single standard requirement to satisfy for the manufacturers. [28], [29]

2.3.4 Technological issues

One of the main technological issues for self-driving cars is to make them safe. The car needs to be better than humans to perceive its environment and better and faster at decision making in the traffic. The decision making is already faster than human reaction time and how it decides is mostly an issue of politics. Amnon Shashua, CEO of the Intel company called Mobileye, which delivers technology solutions for self-driving cars, mentions that there are three orders of magnitude gap between when self-driving cars incorrectly perceive something in the environment and the fatality rate of human drivers. He suggests solving this with redundancies in the cars' perception system and provide maps of the environment that are highly detailed. [30]

One example of situations, where the self-driving cars' perception system struggles is in harsh weather conditions like heavy rain and snow, where the cars Light Detection and Ranging (LiDAR) system, as well as the cameras, get inputs that are heavily distorted, making it harder for the car to perceive its environment. LiDAR can also be challenged when there is a lot of radio interference in the surrounding area, e.g. with more self-driving cars nearby. [31]

As mentioned in subsection 2.3.1, communication is important as a technological aid for self-driving cars. An example of this can be autonomous vehicles that communicate with each other in a communication network, these are often referred to as CAVs. This allows for external computation, as well as safer and more efficient solutions, where information can be shared externally through the internet as well as between cars. This is still an area that is not fully developed. [32]

2.4 Tasks for AVs

This section will elaborate on which problems or tasks an AV system needs to address in order to be operational. The purpose of this is to make an overview of specific technical problems pertaining to the development of a functional AV system.

If the purpose of an AV is to replace a human driver, it should as a minimum be able to do the same job as effectively and safely as the average human driver. There are several tasks involving this overall objective. The AV needs to be able to sense its environment and interpret this, as well as make the necessary planning and decisions to reach its goal destination safely. An overview of a general AV system can be seen in Figure 2.2. It consists of hardware and software, which are used to perceive and make decisions in its environment.

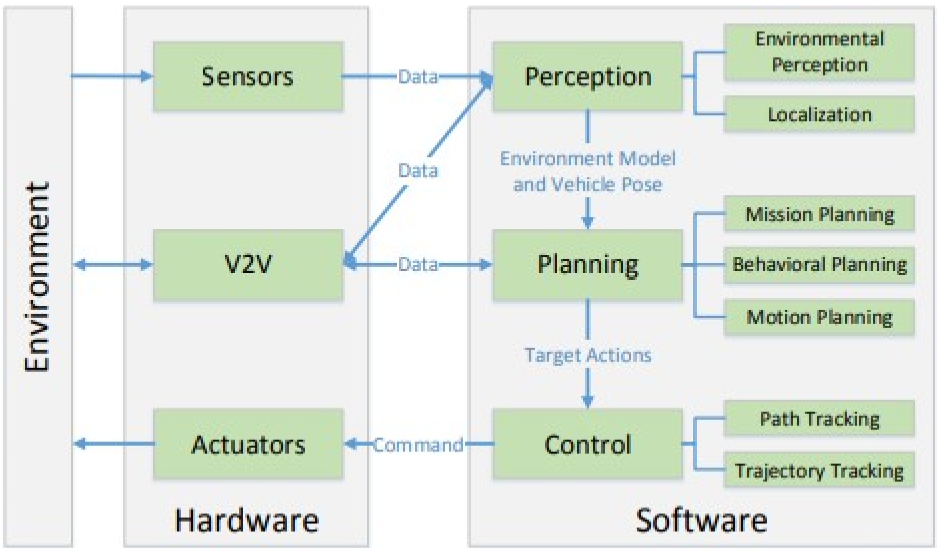


Figure 2.2: Overview of a general AV system [33]

The hardware is the components used to gather data about the environment, such as cameras, Light Detection and Ranging (LiDAR)[1], GPS, infrared sensors, etc.

[1]LiDAR is used to create a three-dimensional map of the environment using the reflections of light pulses being sent out [33].

Other hardware components are Vehicle-to-Vehicle (V2V) communication and the actuators, such as vents on the car, the motor, etc.

The software components, as seen in Figure 2.2, interact with the hardware in order for it to *perceive* its environment, *plan* its actions and *control* the actuators according to the plans. These three software components are elaborated on further in the following subsections.

2.4.1 Perception

During the perception task, the AV system starts the process of understanding the data, and changing it into something it can use to make decisions. The objective is to make a model of the environment. This is where the AV system identifies objects, such as pedestrians and other cars. In addition to identifying objects, the AV system must also understand the space it moves within, e.g. by keeping track of the car's position, both in its current lane, on a global scale, as well as keeping track on how close different objects are to the vehicle. In a simple setting, the perception task could be just to model the car's position and orientation in relation to the road lanes. The result of the perception component should be a sufficient model of the environment (including the vehicle's position), which will then be used by the planning component in order to plan its actions. [33]

2.4.2 Planning

Once the AV system has processed the raw data into something it can understand, it needs to make decisions based on this information. This includes immediate decisions, such as stopping for red lights or making lane changes, or more long term decisions, such as planning the best route to drive. [33]

2.4.3 Control

This is where the plans are actualized by converting them into actual commands for the car, e.g. what angle do the wheels need to be at, or how fast should the vehicle drive. [33]

2.5 State of the Art

This section covers state of the art information on the implementation, hardware, and methodology of AVs, in an attempt to gain a better understanding of where AV vehicles are at today and what the pros and cons are.

2.5.1 Hardware

AVs have a lot of functionalities that require input in some form to function properly. These input can be described as the vehicle's eyes, and if they were to malfunction, the vehicle would be blinded and unable to respond to the environment. The inputs are obtained through different sensors such as *cameras* and *LiDARs*.

The camera is a sensor that collects light. This sensor is very popular and is therefore seen mounted on a lot of the AVs that are being used today. It gives a rich perspective of the environment, and the amount of data that the sensor captures is determined by the specifications of the camera. Two examples of specifications could be *resolution* and *field of view* (FOV). The resolution of the camera determines how much information can be collected within a specific area of a light sensor, and the FOV determines how much of the environment can be collected in one image. Based on these specifications, a camera suited for the AV can be selected. But not every AV needs the same camera. If there is a lack of other sensors on the AV, a camera with a high resolution and FOV is needed to capture as much of the environment as possible. If the camera is not able to capture the environment well enough and there is a lack of other sensors, there is a risk that the AV will overlook vital information. But if there are other sensors present, it might not be necessary with the best of quality pictures and a huge FOV. There are some drawbacks to the camera. It needs light to be able to get a good picture the AV can use as a valid input. Also, if the weather is foggy, it can interfere with the picture and again make it almost impossible for the AV to see anything. [34]

What if the visibility is very low or almost non existent? The AV will be blind if it is too foggy or there is no light to generate a picture for the camera. To solve this problem AVs could use LiDAR. The LiDAR functions a bit like RADAR, but, instead of using radio waves, it emits light. This makes the LiDAR unaffected by the environmental factors affecting the natural light, and can therefore be used during the night and in foggy weather. The LiDAR input is used to create a 3D scene of the environment around the AV. The quality of this scene depends on the number of beams and rotation rate of the LiDAR module. [34]

For an AV to be able to navigate, it needs some way of knowing where it is and where its destination is located. To get this information, a *Global Navigation Satellite System* (GNSS) can be used. A GNSS is a combined system consisting of satellite systems from all over the world, such as the USA's Global Positioning System (GPS). A receiver is then mounted on the AV. This receiver will then collect data on position and time from the satellites. [35]

To assist the GNSS an *Inertial Measurement Unit* (IMU) is installed. This unit supplies the AV with information on angular rotation rate and acceleration. This, combined with the GNSS, should allow the AV to estimate the heading of the vehicle. [34], [36] To collect and process all the data from various sensors, a computer, able to handle the presented amount of information, is needed. It is also the computer's responsibility to synchronize the modules used by the AV. If the data provided by the sensors is not timestamped, or synchronized with the rest of the system, it could provide the AV with misleading information. An example of where the system could provide misleading information is when two different sensors detects the same car but in different positions. If there is no way of telling which information is newer or older, the system will not be able to predict the heading of that vehicle. [34], [36]

2.5.2 Methodology

In this subsection, the methods for localization, mapping, perception, and assessment will be described in regards to their implementation methods.

Localization and mapping

When creating a fully autonomous vehicle it is important to always know where the car is. This can be done with using methods such as *Global Positioning System and Inertial Measurement Unit* (GPS-IMU fusion), *Simultaneous Localization and Mapping* (SLAM), or an *a priori map-based localization*.

The GPS-IMU fusion is a method that uses a sensor-fusion approach which integrates gyroscope, velocity sensors, and GPS, where the goal is to calculate more precisely the position of the AV in urban areas. This approach uses GPS and an odometry system to calculate the AV's position [37]. This system has its shortcomings, such as with the odometry system, which can, over time, drift away from the AV's position due to calculation errors, and the GPS can sometimes suffer from lower accuracy, or even stop functioning all together, in dense urban areas, when entering a tunnel, or other similar circumstances where the signal is lost or there is interference. [38]

Another localization and mapping technique is SLAM. This method will localize the AV and simultaneously create an online map. The method does not require a priori information, and is common practice in indoor environments. A con for this method is that it requires a large amount of computations and has environmental challenges. This means that the algorithm is not as effective as using a pre-built map localization method. [38]

Lastly is the a priori map-based localization method which uses online readings such as landmark searching; i.e. taking landmarks and comparing them with the a priori map to localize the AV. A flaw with the landmark search is that it is prone to error when not enough landmarks are available at the location. Another way, besides landmark searching, is the point cloud matching. This method uses a multi-modal point-cloud based approach. This means that it creates a point-cloud-map using sensors, which it then compares to the a priori cloud-map, and then estimates where the position of the AV is. The a priori map-based localization method is time consuming and takes a lot of resources, since the maps need to be fully updated all the time. This method is computationally more expensive than landmark search. The last sub-method for priori map-based localization is the 2D to 3D matching, which is a fairly new method. This method uses a camera mounted to the Automatic Driving System (ADS), which will take 2D pictures and compare with synthetic 2D images from a 3D LiDAR map. This approach, however, increases the computational load of the localization. [38]

Perception

As described in subsection 2.4.1, one of the most critical parts of an AV is the perception area. When an AV has implemented a good perception, it should be capable of seeing objects and travel on the road safely. All of the following information has been acquired from [38].

Detection of objects is key for perception. For detecting objects, there are several possibilities for doing so. *Image-based object-detection* is to identify the location and size of objects. In this method, the detection of traffic lights, signs to road crossings, moving vehicles, etc., will be detected and analyzed. All state of the art practices for image object detection methods all rely on a deep convolutional neural network (DCNN). An example of this is *You Only Look Once*, which is a real time object detection system [39]. There is a distinction between the two methods in image-based object detection, as described in [38]. These two methods are:

- *Single stage detection frameworks*, which uses a single network to detect objects and make class predictions simultaneously.
- *Region proposal detection frameworks*, which uses two distinct steps, where general regions of interest are first identified, then categorized by separate classifier networks.

The difference between these two distinct methods is that the region proposal detection frameworks are putting out the better benchmark however at a higher computational cost. This method, however, is harder to implement, train, and fine-tune. The single stage detection frameworks use low memory and have a fast inference time.

Another method in perception is *semantic segmentation*. Semantic segmentation is an algorithm, which classify each pixel in a picture with a label. The reason for doing this is because of some objects, are hard to properly define using bounding boxes [2]. Examples of these objects are roads and buildings.

Instance segmentation is the classification of each pixel in a picture, but will also differentiate between objects based on their trajectories and behaviors. When using

[2] A bounding box is a rectangular box, which is used as a description to where an object is positioned inside a frame with x and y coordinates. [40]

Instance segmentation the classification would be able to see overlapping objects and bound box them [41]. This means that the bound boxes would overlap each other, depending on the variation of overlapping from the object.

When creating an AV it is important that the vehicle has depth perception, in order for it to see objects further ahead and avoid a possible collision. Depth perception can be done with a single mounted front camera, but these algorithms require a big amount of processing power. Therefore, in most situations, it is easier and better to use stereo- or multi-view systems. An example of this could be the 3D LiDAR, which is not as dependable on lighting conditions as cameras.

Another thing which is important for AVs is the tracking of dynamic objects, such as pedestrians and cyclists. It is important to know their trajectory and the speed they are going. The reason for this tracking-need is to estimate their future trajectory and speed, to avoid possible crashes, or maybe to overtake a car on the highway. To estimate this, a sensor fusion is often used for tracking. The most common object trackers use a simple data set association technique, which then uses traditional filtering methods. When using image-based methods for object tracking, there needs to be an appearance model which could use color histograms, gradients, and other features.

One crucial point that AVs have to take into consideration is road and lane detection. This is usually divided into sub tasks, which each heightens the level of automation in vehicles. For a vehicle to be fully autonomous, it needs a complete understanding of the road structure, and of the detection of lanes at a long distance. The currently used methods for road- and lane detection rely on external data preprocessing. An important step in the preprocessing is distinguishing between static road scenery and dynamic objects, which allows for extraction of road- and lane detection to be done by the correct data. Some of the methods use a camera for the detection. When using a camera, it is necessary to normalize the colors with the help of color correction. Other methods, such as LiDAR, can use several different filters to reduce cluttering in data, such as map-based filtering. Some systems have already been created to assist with lane keeping and are well integrated. Most systems, however, still use assumptions and have limitations. Systems which can handle complex road systems, are not yet developed. But systems are slowly being made using standardized road maps, and machine learning-based road and lane classification methods.

Assessment

Assessment is a crucial part of AVs. The reason for this is that a good assessment of possible dangerous scenarios can increase the overall safety of AVs. A method for risk- and uncertainty assessment is to use Bayesian methods, which will quantify, and measure, uncertainties of deep neural networks. Another method for risk assessment is the use of sensory input, in a risk framework, which uses hidden Markov models to detect unsafe lane change events. Lastly, there is a method that accumulates the overall risk of a driving scene using a monocular camera, which can be implemented through an open source from [42]. [38]

Another assessment, which the ADS should consider, is the behavior of drivers. This assessment does not yet have common practice, but different methods using a hidden Markov model have been used with some success. An example using the mentioned method is the detection of dangerous cuts. The main challenge in this area is the short observation windows, when analyzing human behavior, since most of the time the AV's system only have seconds to observe the vehicle, and therefore complex behavior models cannot be implemented as it requires longer observations. [38]

AVs need to be able to predict the driving style of the other drivers around it. This is a necessity because not all drivers behave in the same way. To do this, there are multiple categories an AV can put each driver into. This could, as an example, be based on how aggressive they behave or based on their skill level. If a driver is categorized as a skilled and aware driver, the chances of him yielding in a lane is relatively high. [38]

Rules are needed to determine a drivers behavior. This could be based on how much the car moves around in its lane, or how much it jerks when performing a maneuver. [38]

To this day, a successfully implementation of driving style recognition has not been reported yet. But studies has shown that such a system is possible. [38]

2.6 Problem definition

In section 2.1 it's found that there were problems in traffic. These problems were, to some degree, caused by humans and their behavior in traffic. Other problems were related to the infrastructure of the roads. A level 5 AV could remove the need for humans in traffic, meaning the human influence in traffic could be eliminated. This could significantly impact the traffic problems related to humans. Additionally, it could also influence other problems related to traffic congestion. For an AV system to be successful, several problems need to be solved. The general problems an AV system faces are described in section 2.4, while more specific problems are described in section 2.3 and section 2.5. Due to the size of the subject, it has been decided to focus on a smaller subset of the problems faced by AVs.

Therefore the following problem statement has been developed.

Problem Statement:
How can we design and implement a prototype of an AV, that can identify and drive within its lane?

2.6.1 Problem requirements

The intent of this subsection is to form requirements for the project which, when fulfilled, will satisfy the purpose of the project and the problem statement.

The purpose of this project is to design and implement an embedded system. This is reflected in this project by creating an autonomous driving system which will be embedded in a lego vehicle.

In order to determine whether the purpose and problem statement of the project has been reached, these requirements have been found:

- ☐ A working lego vehicle similar to a car
- ☐ An autonomous driving system which is capable of driving and following its lane

As the vehicle is a model of an AV, it is important that the it operates in a similar manner to a car. For the purposes of the vehicle, this includes having four wheels, steering, and control of speed/acceleration.

The goal of the project is to create a working prototype of an AV system that can drive on its own, which implies a necessity for the vehicle to be capable of following a road. The road can either be straight or turn to the left or right, but does not include more advanced things such as intersections or roundabouts. The lane will be represented by two same colored lines with the color being in contrast to the ground. The colors do not have to be white and black/gray, as they typically are on regular paved roads.

When these requirements have been achieved, the problem statement is considered resolved.

Part II

Technology analysis

Optimal Solution 3

This chapter is meant for framing our take on the best possible hardware and software for us to work (and experiment) with. The word *optimal* should therefore not be seen as a carefully measured selection of resources, but rather as a greedy pick, had money and time been no problem. By that logic, anyone referencing this report (perhaps while doing a similar, but more ambitious, project) might find useful suggestions for hardware and software in this chapter.

The chapter ends with a summary of the most essential features of the greedily picked materials, which will help when evaluating the actual hardware and software we will have access to.

3.1 Hardware

As part of the requirements stated in subsection 2.6.1, our vehicle should be car-like and host an on-board AV system. When aiming for a car-like vehicle, some sort of *propulsion* (and steering) is implied, and, to fulfill our requirements for the AV system, some sort of *computing platform* is needed, as well as a *world input.* All three headlines are further described in this section.

3.1.1 Propulsion Platform

A candidate for a combination of chassis, wheels, propulsion, steering, and room for additional hardware, is the *Traxxas TRX-4 Chassis Kit* [43].

Figure 3.1: Traxxas TRX-4 Chassis Kit

As can be seen on Figure 3.1, this kit provides a minimalistic starting point, allowing as much room as possible for our own additions and adjustments. All support for wheels, propulsion, and steering would be taken care of with this hardware, as well as any routing of power to, and control of, all motors. The biggest challenge would be to fit and power our own additional hardware, but, given the minimalism of the chassis, is a problem solvable by 3D-printing the needed brackets/fittings.

3.1.2 AV System Platform

The term "AV System Platform" is, in this section, understood as a link between the engine of the car and the world the car is driving in. This link is expected to be able to perceive (directly or indirectly) both the world around the car as well as the state of the car itself, make a decision based on all gathered information, and finally have the car perform the requested action. To accommodate these requirements, the *NVIDIA Jetson Nano Developer Kit* (depicted on Figure 3.2) would appear to be a capable platform [44].

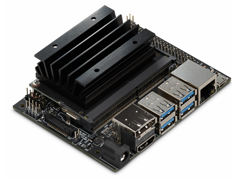


Figure 3.2: NVIDIA Jetson Nano Developer Kit

As per NVIDIA's own words, the Jetson Nano is "*[...] a small, powerful computer that lets you run multiple neural networks in parallel for applications like image classification, object detection, segmentation, and speech processing. All in an easy-to-use platform that runs in as little as 5 watts.*" The Jetson Nano contains a dedicated 128-core Graphical Processing Unit (GPU) based on the Maxwell architecture, as well as other advantages, making it a powerful starting point. [44]

3.1.3 World Input

As for world input to the AV system, not much is required other than ease of use and compatibility. Since the requirements specifies the need to follow a road represented by colored lines, some sort of camera appears as an obvious candidate. Any webcam with a USB interface should suffice, and a frequent entry when searching for the "best USB webcam" is the *Logitech C922 Pro Stream* [45]–[47] depicted on Figure 3.3.

Figure 3.3: Logitech C922 Pro Stream

This webcam is capable of capturing 60 frames per second in a resolution of 1280 times 720 pixels, which should give a satisfying amount of input data for the AV system platform to work with. As the viewing angle for this camera is 78 degrees, more than one camera could be installed to cover a larger area.

3.2 Software

As with the hardware, the choice of software is going to have a significant impact on the final product. Unlike with the hardware, the software is where we get to do the bulk of the work for this project, and identifying the most optimal choice is therefore not as straight forward. The following section is one way of answering the question:

The most optimal greedy-choice of software, in terms of delivering something that fulfills our requirements, would be to reuse the software found in existing autonomous vehicles, such as the ones produced by either Google or Tesla mentioned in chapter 1. These solutions have already been tried and tested, and should be fairly deploy-able, given the necessary hardware.

The next-most optimal choice of software is going to depend heavily on what hardware is chosen, as not all hardware supports the same software solutions. If the Jetson Nano from subsection 3.1.2 was chosen, the obvious choice would be the accompanying *NVIDIA JetPack SDK*, which gives officially supported access to several AI-oriented frameworks, such as *TensorRT* (neural networks) and *OpenCV* (image processing) [48].

3.3 Essential Features

Having looked at the possibilities for choice of hardware and software, the following features from each section will be held in mind when choosing from the provided selection we have access to.

3.3.1 Hardware Features

A common theme from the hardware described is that they are all designed and pre-built to a point where our job is to do final assembly, and then dive straight into the software design. This can be seen on the chassis from Traxxas containing all needed components for creating a rough idea of a car, it can be seen on the Jetson Nano being a complete system in itself (with official accompanying operating system and frameworks), and it can be seen on the webcam from Logitech ready to installation via a common USB type A plug.

3.3.2 Software Features

The software had two common themes worth mentioning: reuse of existing solutions, and easily accessible tools for building a new solution.

Given the needed hardware, we acknowledge the existence of solutions made with both scientific and commercial goals, and deploying those would indeed get us to the finish line for this project.

By taking the suggested Jetson Nano into account, and thereby making the hardware the limiting factor (instead of the software), the choice points more in the direction of pre-built frameworks and tools for us to experiment and work with.

Actual solution 4

In chapter 3, an analysis of the optimal setup for a solution was conducted without any restrictions to what was possible in terms of limitations for this project. This chapter will instead approach a more realistic solution compared to what is actually possible to do. Firstly, an analysis (and choosing of) the hardware will be conducted, after which the possible operating system for the chosen platform will be examined.

4.1 Hardware

This section will explore the different options for platforms, motors, and cameras in terms of what is available from either Aalborg University or what we own ourselves. This analysis of the different parts will create the basis for a selection of the components which will be described at the end of the chapter.

This chapter contains specifications of LEGO parts, in order to identify the exact LEGO part referred to, the title with the LEGO part name will also include the design ID of the specific LEGO part being described. The design ID is the LEGO part's unique identifier without referring to a specific color [49].

4.1.1 Platforms

The hardware platform would be the main component in the system, as it would run the application, and control all other components, in terms of reading their output and giving them input.

The criteria on which the platform is chosen, which will be further described in subsection 4.1.4, are as follows:

- By default the platform should be able to, without any unauthorized changes to the motor or the platform [1], control motors.
- The platform should be able to read input from a camera.
- The platform need computing power, because one of the tasks of the software being developed is to process images. This processing must be done relatively fast, since the car cannot stop and wait for the processing to finish.
- The platform needs to be accessible wirelessly.

[1]Unauthorized changes means no soldering, cutting wires, open up a part that is not meant to or any other such altering.

LEGO MINDSTORMS EV3 (ID: 95646)

Figure 4.1: LEGO MINDSTORMS EV3 brick [50]

The LEGO MINDSTORMS is about creating and programming robots to do different tasks. The key component in this is the EV3 brick, which is a programmable computer, that can be connected to sensors and motors, in order to accomplish its tasks. It's driven by a 300MHz 32bit ARM9 processor, 64MB of RAM and 16MB FLASH memory, which can be extended with the built-in SDHC-card reader. The brick has four ports to read input from devices such as touch-sensor, gyro sensor or any of the others compatible sensors within the LEGO universe. The brick also has four output ports for motors. The cables for all the input and output ports are RJ12, which are the cables used for the different sensors and motors within the MINDSTORMS brand of LEGO, and there exist converter cables to other LEGO motors, such as the LEGO Technic motors [51]. The connection to the EV3 can be done by either Bluetooth, USB-port, or WIFI, depending on what the user considers as the most optimal. The brick also allows for controlling through the physical buttons on the brick itself, where the 178x128 pixel built-in display provides information. The Ev3 brick also allow sound output through its built-in speakers. Furthermore, it can be powered by a power supply, or by battery; either six AA batteries or the LEGO MINDSTORMS EV3 rechargeable DC Battery which is a 2050mAh lithium-ion battery [52]. [53] [54]

LEGO MINDSTORMS NXT 2.0 (ID: 53788)

Figure 4.2: LEGO MINDSTORMS NXT 2.0 brick [55]

Where the EV3 brick is the key component in the LEGO MINDSTORMS universe, the predecessor to the EV3 brick is the NXT 2.0 brick. Compared to the EV3, the NXT 2.0 is more limited in regards to its hardware specifications, as it only has a 48MHz 32-bit ARM7 processor with 64KB of RAM and 256KB flash memory, which can not be extended with a SDHC card. The brick has four sensor ports and three motor ports, which is one motor port less than the EV3. These are attached to the brick though RJ12 cables, like the EV3 [51]. The NXT 2.0 can communicate through USB cable or Bluetooth, but not WIFI, allows for controlling through its physically buttons on the brick, and information can be seen on the 100x64 pixel built-in display. Like the EV3 the NXT 2.0 brick also have speakers, but the sound quality is not as good. Like the EV3, the NXT 2.0 can be powered by power supply or by batteries which can either be six AA batteries or a rechargeable battery pack. [56], [57]

Raspberry Pi

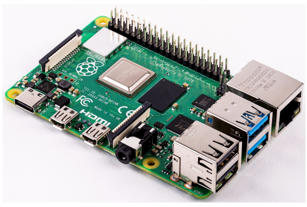


Figure 4.3: Raspberry Pi 4B [58]

Raspberry Pi is a line of small computers which are powerful compared too their price [59] [60]. The first generation of the Pi called model A has a 32-bit processor running at 700MHz, 256MB RAM only a single USB port and no ethernet, WIFI or Bluetooth [59] [61] [62]. Since the first model, multiple generations have been released with improvements in the hardware specifications. The newest model is the fourth generation model B, which can be seen in Figure 4.3. This model has a 64bit ARMv8-A processor running at 1,5GHz and has either 1GB, 2GB or 4GB of RAM. It also features multiple USB ports, an ethernet port, WIFI, and Bluetooth. The predecessor to the 4B used the same GPU, but along with other upgrades, the GPU in the 4B is also updated to a more powerful one [63]. All other versions than the 4B have HDMI connection for video and sound, where the new version of the Raspberry Pi has a newer HDMI standard and two micro-HDMI ports, instead of one normal sized HDMI port as the rest. The first generations had an RCA composite jack, but this has been replaced by a 3.5mm jack in the subsequent versions. Beside this, all versions have general-purpose input/output pins, DSI port for raw displays, CSI-2 camera port, storage through SD cards in the old version, and through SDHC cards in the newer versions. The newer the version are, the more power it consumes, where the old 1B+ under max stress consumes 1.75 watt (0.35amp at 5v), the newer model 4B consumes 6.25 watt (1.25amp at 5v). [64]

The available Raspberry Pi models for the project is the B+, 2B, 3B, 3B+ and 4B with 4gb RAM.

Arduino

Arduino is a company producing electronic boards for the purpose of reading some input (e.g. from a connected sensor), and then turn this into some output (e.g. a motor or an LED). The decision depends on the program the user creates and implements on the Arduino board [65]. The board exists in several configurations depending on the needs, of which only the Arduino Uno and the Arduino Mega are available for us to use in this project [66].

Figure 4.4: Arduino Uno Rev3 [67]

The Arduino Uno, seen in Figure 4.4, is an entry level board in the Arduino universe, it features a clock speed of 16Mhz, 2KB RAM and 32KB flash memory. The communication with sensors, motors and other components is done through the boards digital input/output pins, which this board has 14 of, or through the 6 analog pins. The board can be connected to a computer through its USB port. The board is powered through an adapter at recommended 7v to 12v, depending on which and how many components that are connected, or can be powered through the USB port. [68]

Figure 4.5: Arduino Mega 2560 Rev3 [69]

The Arduino Mega, which can be seen in Figure 4.5, when compared to the Arduino Uno, is a more powerful board with more options. Like the Arduino Uno, the Arduino Mega is powered at 16MHz clock speed, but it features 8KB RAM and 256KB flash memory. The board provides more options with it's 54 digital I/O pins, and 16 analog pins. And, like the Arduino Uno, the board can be connected to a computer through its USB. Like the Arduino Uno the Arduino Mega can be powered by either a adapter, or by the USB port. [70]

4.1.2 Motors

In order for the autonomous car to physically move itself, it would need some motors. As described in the car will need a motor to drive it forward and backwards, as well as a motor, in order to steer the car in a direction.

Autoref to requirements section, stating that the

In addition to the motors, which will be elaborated herein, other motors are also available. These motors include different DC looking motors, which are powered and controlled through two power wires. However, since no model name or other information could be found for these motors, it's not possible to find the specifications for these. Therefore, these motors will not be described further.

The motors will be selected, in subsection 4.1.4, based upon the following criteria:

- By default the motors should be able to, without any unauthorized changes, to be connected to and controlled by the chosen platform, on which no unauthorized changes may be made.
- Precision is prioritized, when choosing the motor that drives the car, as well as the motor that steers it, in order to control the car accurately.
- In the selection of the motor, to drive the autonomous car forward and backwards, a sufficiently powerful motor is prioritized, so that the weight of the car is not limited by the motor power.
- For the choice of the steering motor, speed is somewhat prioritized, in order to change direction quickly.

Large EV3 Servomotor (ID: 95658)

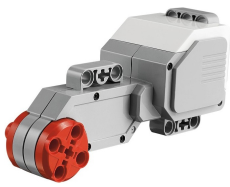


Figure 4.6: LEGO MINDSTORMS EV3 large servomotor [71]

The EV3 large servomotor, which can be seen in Figure 4.6, is a servomotor within the LEGO MINDSTORM universe. It is the most powerful lego motor available which, through its built-in rotation sensor, allows for precise control down to an accuracy of one single degree. The servomotor spins at a maximum of 160-170 rotations per minute (rpm), and is powered and connected through its RJ12 port. The servomotor works as default on both the LEGO MINDSTORMS EV3 brick, as well as on the LEGO MINDSTORM NXT 2.0 brick [72]. [53]

Medium EV3 Servomotor (ID: 99455)

Figure 4.7: LEGO MINDSTORMS EV3 medium servomotor [73]

In LEGO MINDSTORM, another type of servomotor, other than the large one, exists, namely the medium servomotor, which can be seen in Figure 4.7. Like the large edition, this servomotor features a built-in rotation sensor, making it possible to control it within a single degree of accuracy. The servomotor is also like the large servomotor powered through an RJ12 port, which it also uses to connect to the EV3. The medium servomotor is with its maximum of 240-250rpm faster than the large servomotor, but is not as strong. Like the large edition, the medium servomotor is compatible with both the LEGO MINDSTORMS EV3 brick and the LEGO MINDSTORMS NXT 2.0 brick [72]. [53]

NXT Servomotor (ID: 53787)

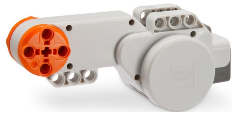


Figure 4.8: LEGO MINDSTORMS NXT 2.0 servomotor [74]

As previously described, the NXT 2.0 was the predecessor to the EV3 within the LEGO MINDSTORMS universe, this also applies to the NXT servomotor, which was the motor for the NXT 2.0. This servomotor, shown in Figure 4.8, like the EV3 large servomotor, allows for precise control to a single degree of accuracy, with its built-in rotation sensor. It works on both the LEGO MINDSTORMS EV3 brick and the LEGO MINDSTORM NXT 2.0 brick by default, and is powered and connected through its RJ12 port, like the EV3 large servomotor [72]. It is tested by Philippe Hurbain, an engineer who was one of the authors of the book "Extreme NXT", to run at a maximum of just above 160rpm [75] [76]. [57]

Electric Technic Mini-Motor 9v (ID: 71427)

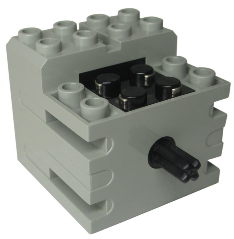


Figure 4.9: LEGO Technic Mini-Motor 9v [77]

This motor, shown in Figure 4.9, is an older motor for the LEGO Technic universe. According to Philippe Hurbain, it is not as powerful as EV3 or NXT 2.0 motors, but with its 360rpm, it is faster than those motors. And unlike the previously described servomotors, this motor does not feature a built-in rotation sensor and therefore does not allow for precise control in terms of degrees. The motor is powered by an electric wire plate, attached to the top of the motor on the black square. There exists an adapter cable which allows for the motor to plug-in to an RJ12 port, allowing the motor to be connected to both the EV3 and NXT 2.0 brick by default, which was tested during this analysis. [78]

Electric Technic Motor 9v (ID: 2838)

Figure 4.10: LEGO Technic Motor 9v [79]

According to Philippe Hurbain, this motor, which can be seen in Figure 4.10, is with its 4100rpm, a faster edition in the LEGO Technic universe, than the previously described Mini-motor. However it is trading its speed with power, and is the weakest LEGO motor available. Like the Mini-motor, this one does not have a built-in sensor, meaning that it does not allow for precise control in terms of degrees. This motor also works by default on the EV3 and NXT 2.0 brick, through the RJ12 adapter cable, which is attached to the bottom middle of the motor. [78]

4.1.3 Cameras

For the system to be able to get a visual input about its environment, a camera is needed. There is only one camera available which will be described in this subsection.

Even though only one camera is available, it must still comply with the following criteria:

- The camera must be able to connect through USB.
- Must have a decent resolution, however, it does not have to be high.
- The camera must have a wide enough field of view, both horizontally and vertically, such that the it can provide sufficient information about the autonomous car's environment.

Logitech C910

Figure 4.11: Logitech C910 camera [80]

The Logitech C910, shown on Figure 4.11, is a 5MP camera with autofocus, with an 83 degree diagonal field of view. It captures video at either 4:3 format up to 1600x1200 pixels, or 16:9 format at up to 1920x1080 pixels. It connects to a computer through its USB cable and can be tilted 180 degrees vertically. The camera also features a built-in stereo microphone which allows it to capture audio. [81]

4.1.4 Choice of hardware

Based on the requirements and analysis in the preceding sections, the hardware components are selected.

Platform

Based on the examination of the available platforms, it can be concluded that both the LEGO MINDSTORMS EV3 and the LEGO MINDSTORMS NXT 2.0 could meet the requirement of controlling motors by default, of which multiple options for motors was available. The EV3 could do what the NXT 2.0 could do, and had more features as well as being more powerful, and is therefore selected as the platform. However, even though the EV3 with its 300MHz has more computing power than the NXT 2.0, it is not considered undoubtedly powerful enough and therefore the powerful Raspberry Pi 4B with 4gb was selected too. The EV3 may turn out to be powerful enough, but at this point is not possible to guarantee whether or not it is, therefore it is chosen to go with the Raspberry Pi 4B, which we acknowledge might be an overkill. If its later determined that the EV3 has enough processing power, the Raspberry Pi 4B can be deselected. With this setup, all the requirements listed in subsection 4.1.1 are achieved.

Motors

Since one of the chosen platforms is the LEGO MINDSTORMS EV3, all the LEGO motors would work with this.

Firstly, for the motor to drive the autonomous car, one of the requirements was that it needed to be powerful. As it could be concluded in subsection 4.1.2, the most powerful motors was the EV3 large servomotor and the NXT servomotor. Both servomotors can also be precisely controlled, which was another priority. Since both the of these motors are almost identical in terms of specifications, either one of those would work, therefore the EV3 large servomotor is selected as the drive motor, one for each of the back wheels.

Secondly, for the motor that steers the autonomous car a combination of accuracy and speed was prioritized. Both the EV3 large servomotor, EV3 medium servomotor and the NXT servomotor allowed for precise control, however where the EV3 large servomotor and NXT servomotor are powerful, they are not as fast as the EV3 medium motor. The EV3 medium motor is therefore selected as the steering motor.

Camera

There was only one immediate choice of camera, however it still had to fulfill the needs listed in subsection 4.1.3. Based on the examination, it can be concluded that the camera meets the requirements, and it is therefore chosen.

4.1.5 Hardware test

Now that the hardware has been chosen, it is necessary to test whether or not it lives up to its specifications. The EV3 motors may have been used by many people before and could potentially have been damaged. As such, it is important to test whether or not these motors will perform as advertised.

For each motor, both a precision test and a speed test have been performed. The goal of the precision test is to determine how accurate the built-in rotation sensor is, while the speed test ascertains whether the rpm is accurate. All motors were tested; the medium motor and both large motors. The results will be presented starting with the medium servomotor, then the large left servomotor, and lastly the large right servomotor.

Precision Test

In order to test a motor's precision, the motor was instructed to turn a total of 360 degrees, after which the angle of the motor was noted with a dot on a measuring paper. The setup can be seen in Figure 4.12 This was done a total of 10 times for each motor, after which the spread of the angles were evaluated, to determine how much the angles had deviated.

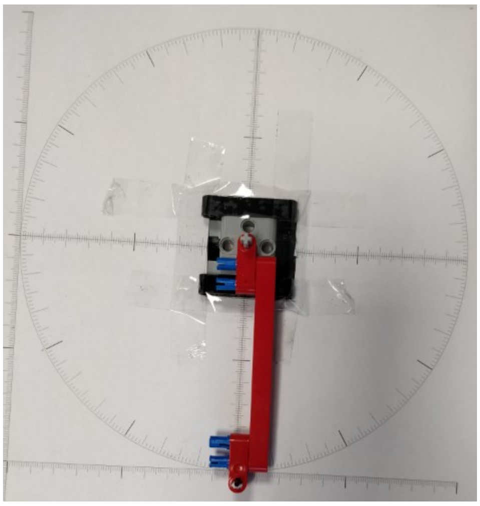


Figure 4.12: The setup for testing the medium motor

The results of the tests showed that all motors stayed within its single degree accuracy, with the large left motor having the least amount of variance. The results for the medium motor can be seen in Figure 4.13

Figure 4.13: Results from the precision test for the medium motor

Speed Test

The speed test was conducted by instructing the motor to run at max speed for a total of six seconds. This was recorded in slow-motion, and the number of rotations during this slow-motion video was manually counted. The number of rotations counted during the six seconds can then be scaled up to one minute. A small lego brick was used to indicate the motor's starting position. The physical setup for this test was the same as for the precision test. This test was done five times for each motor. The number of rotations was rounded to the nearest quarter-turn.

The results for the medium motor after running the speed test five times were: 24.75, 24.5, 26.25, 26.25, 26.0, this gives an average of 25.55 or 255 rpm.

The results for the large left motor after running the speed test five times were: 15.0, 15.75, 15.5, 15.0, 14.75, this gives an average of 15.2 or 152 rpm.

The results for the large right motor after running the speed test five times were: 15.5, 15.5, 14.5, 15.25, 16.0, this gives an average of 15.35 or 153 rpm.

reformat the visual presentation of the test-results

Considering slight inaccuracies when scaling the results, the motors performed well within the specifications found in subsection 4.1.2. However it must be noted that during the test extra load was added, which could have had a impact on the result.

Part III

Design

Change part/chapter/-section structure from hereon out.

Design 5

5.1 Autonomus Driving System Architecture

This section will describe the proposed overall architecture for the autonomous driving system developed in this project. An illustration of this can be seen in Figure 5.1. The system is divided into four phases, namely Data Collection, Training, Inference and Steering, where the former two takes place offline and the ladder two takes place during the live run of the system, where the vehicle is driving by itself.

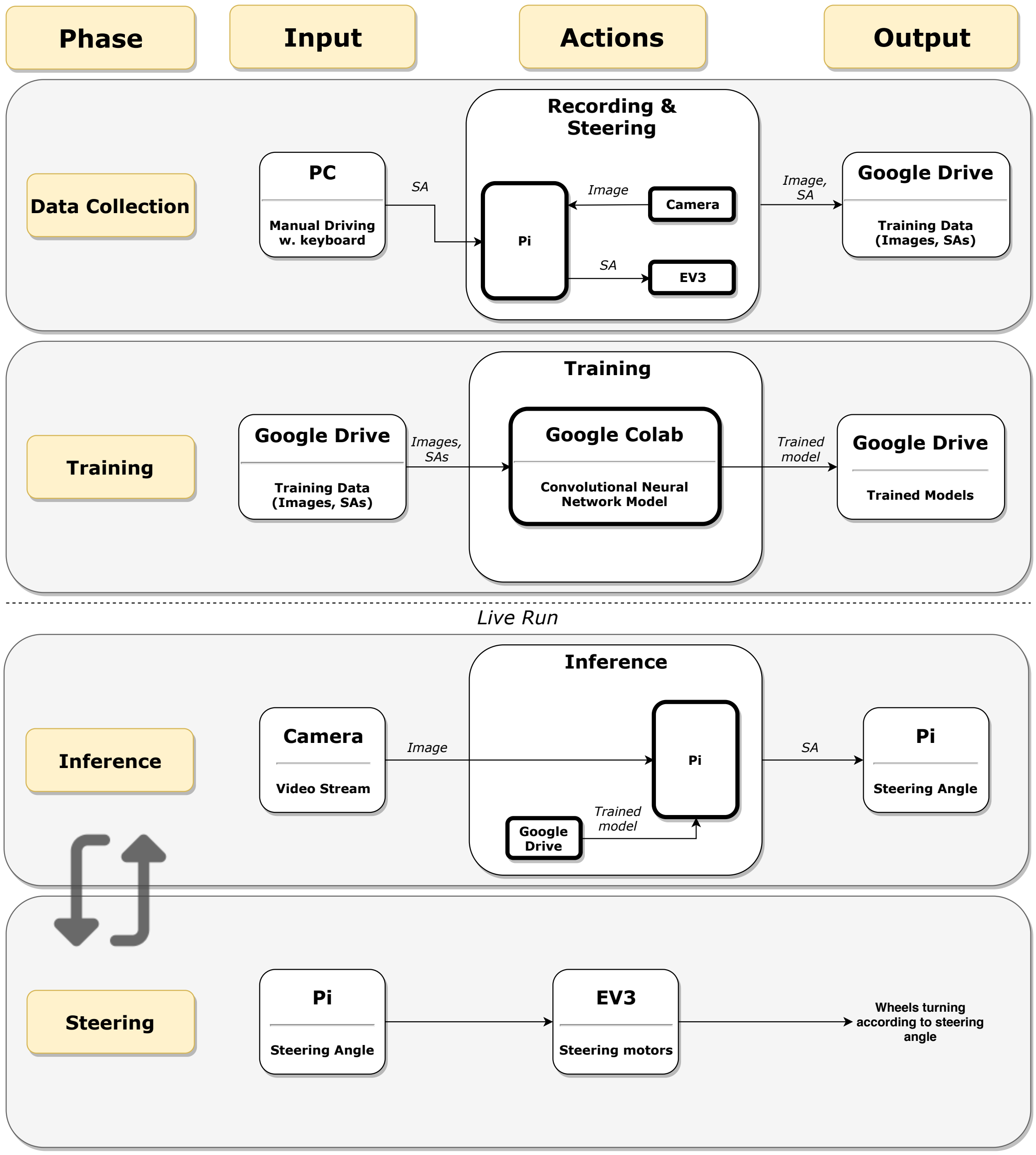


Figure 5.1: Overall architecture of the autonomous driving system. SA is short for steering angle.

The required data to train a convolutional neural network,will be gathered by driving the vehicle manually with a pc connected to the raspberry pi, which in turn is connected to the EV3 and a camera. This allows for passing steering angles to the ev3 through the keyboard on the pc, while obtaining images from the videostream on a camera. These images can then in turn be labeled with the steering angle and stored in Google Drive and constitutes the training data for that particular run.

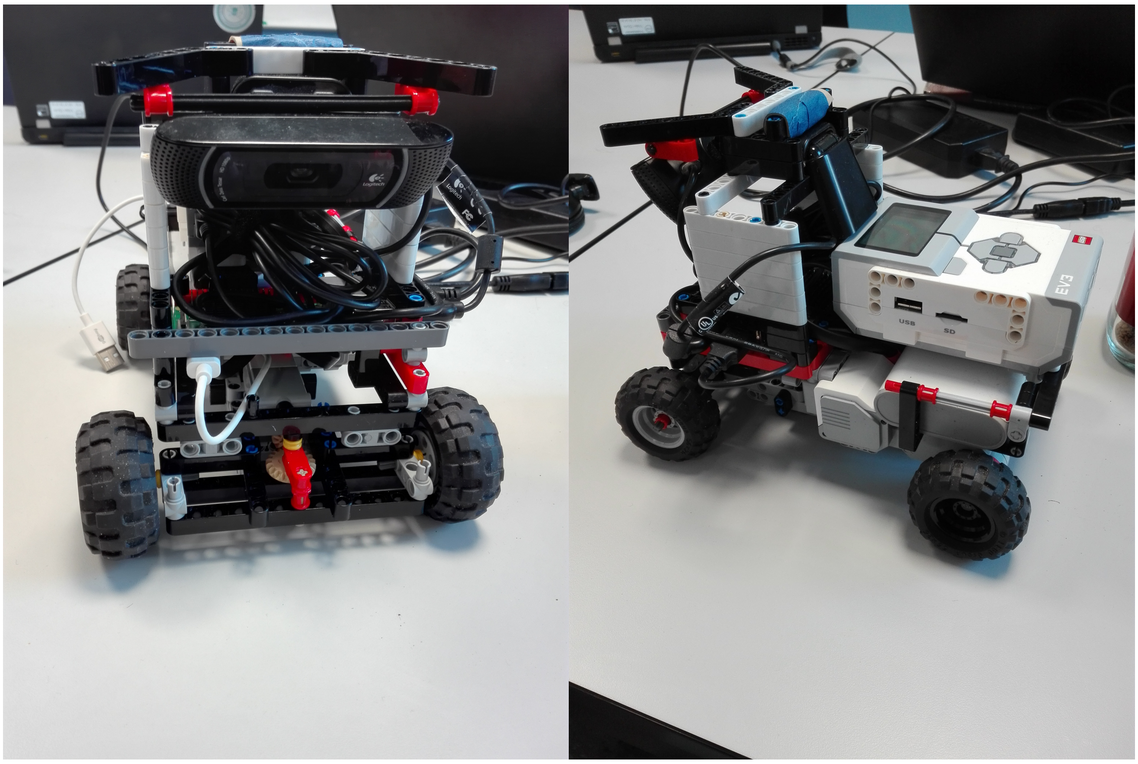
In order to train the model, it was decided to use Google Colab, which is a common cloud computing service for training artificial neural network models. The idea is to setup Google Colab to automatically gather the training data from Google Drive, in order to train the convolutional neural network model that will be implemented and in turn store that trained model back in Google Drive.

For the live run, the Inference and Steering phases, should run iteratively, where the Camera's video stream gives an image to the Pi as in the manual run. Now however,the steering angle should be obtained through inference on the trained model on the pi with that image, instead of manually from the PC. This steering angle should then be used in the Steering phase by the Pi to instruct the EV3 to steer its motors accordingly, resulting in the wheels being turned by that angle.

5.2 Vehicle Design

As stated in section section 2.6, one of the requirements for the vehicle is to be similar to an actual car. The vehicle has been built with four wheels, front wheel steering and is rear wheel driven. Typically cars have front wheel steering, though more cars have started using four wheel steering. Rear-wheel drive (RWD) was chosen because it would make the process of building the vehicle smoother, as it would seperate the responsibilities between front and rear wheels. While there exist all-wheel drive (AWD), front-wheel drive (FWD) and RWD cars, each approach having their own strenghts and weaknesses. These strengths and weaknesses aren't of vital importance for the project, as the focus is on AVs and automatic driving. As such, making the process of building the vehicle smoother was prioritized. [82]–[85]

To match our choice of hardware, the vehicle will be built using lego. As we use multiple pieces of hardware from lego, building the vehicle using lego would ensure compatability between the motors, EV3, and the vehicle. In addition to the lego hardware, we use two other pieces of hardware, namely the Raspberry Pi and the Logitech camera. In order to ensure both are properly secured to the vehicle, each piece of hardware needs it's own mount. Additionally, the camera needs to be secured in a way that prevents it from shaking or moving in its mount. Finally, in order to power the Pi while driving, we need a portable power source. This portable power source comes in the form of a power bank.



(a) Picture from the front of the vehicle (b) Picture from the side of the vehicle

Figure 5.2: Pictures of the finished vehicle taken from different angles

In Figure 5.2 are two pictures of the completed vehicle, from the front and side respectivly. The vehicle has space for all of the components previously described, as well as a functioning mechanism for turning the front wheels. To allow for the camera to better observe the road, its mount have been built at a higher elevation than the rest of the vehicle. The mount also uses a rubber band to help secure the camera and prevent it from moving in its mount. The horizontal "horns" on the mount is to help orient the camera in a consistent manner. Underneath the camera is a small "case" to secure the raspberry Pi. This case also allows for the required cables to connect to the Pi. The EV3 and the powerbank is placed in the of the vehicle. The EV3 can be connected directly to the vehicle and as such does not need a "case" or mount to secure it to the vehicle. The powerbank, however, is placed in a small "case" underneath the EV3 where it is secured. Unfortunately the steering mechanism has a small amount of backlash, which means it is possible to turn the wheels slightly, without affecting the angle of the motor. This doesn't have an effect on how the software controls the angle of the motor, but it does mean there is an additional inaccuracy when turning the front wheels.

5.3 Network Architecture

In section 5.1 an overall system architecture was described, this section will describe in more detail the architecture of the network that allows for the communication between the PC, Raspberry Pi and the EV3. As the vehicle will use more than one device when running, it will need a way to connect its devices. Each device has its own individual responsibilities, the EV3 controls the motors on the vehicle and the Pi is responsible for calculating the steering angle based on the input from the camera. Additionally, a PC will be used for communicating with the vehicle, in order to manually steer the vehicle during the Data Collection phase, as mentioned in section 5.1.

Because they each have individual responsibilities they need a way to communicate. In order for the EV3 to control the steering motor of the vehicle correctly, it requires the angle from the Pi. Without it, the vehicle wouldn't be capable of driving along the road.

There are two ways of establishing a line of communication between the two devices: Wired or Wireless

Connecting the devices through a wired connection, would involve a physical cable running between the devices. There exists a number of different ways to do this, like Ethernet, USB, I2C and more. Wired connections tend to have faster and more stable connection than it's wireless counterparts.

Connecting the devices wirelessly, would involve connecting them through another network like WiFi or directly through Bluetooth. This would avoid a cable between the devices and would mean they weren't physically connected. This, in turn, would put less limitations on the distance between the devices and make them more portable.

Due to the proximity of the EV3 and Pi, as well as the fact that both devices will be mounted directly on the vehicle, a wired connection will be used for connecting the devices. Since performance is prefered over portability, it has been decided to use a wired connection.

While a wireless connection wasn't useful when connecting the EV3 and Pi, it is when connecting the PC to the Pi. As the PC won't be mounted on the vehicle, having cables between the vehicle and the PC could potentially cause problems while driving. Because of this, the PC will be connected wirelessly to the Pi.

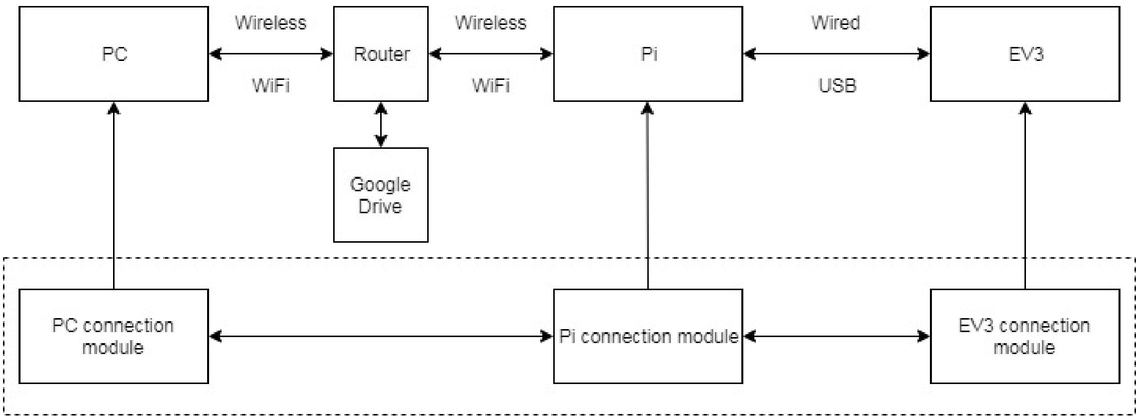


Figure 5.3: Diagram of the network architecture

As seen in Figure 5.3, the PC will connect wirelessly to the Pi through a router and the Pi and EV3 are connected through USB. The purpose of connecting the Pi to the router is to make it more convenient to upload data to Google Drive, download new trained models from Google Drive, as well as updating code on the Pi from the Github repository. Any communication between the PC and the EV3 will happen through the Pi. So when the PC sends a command that has to go to the EV3, the Pi is responsible for forwarding the command. Each device will have a connection module, as seen in the dotted box. These modules are responsible for all the communication happening between the devices.

5.4 Proposed Artificial Neural Network Archicteture

This section proposes an artificial neural network architecture (ANN), to be used in this project. Modern approaches to autonomous driving often involves a type of ANN called deep neural networks (DNN), specifically convolutional neural networks (CNN), for tasks relating to computer vision, such as lane following and other decision making based off of image data. [86]–[91]

Designing an artificial neural network from scratch is a time consuming endeavour in itself, involving many subsequent evaluations of the models, using training data to arrive at a satisfactory model. For the purpose of this project, this would be out of scope, but fortunately several models used for autonomous driving have already been designed [88], [89], [91]. One of the popular ones, are the Nvidea model, also called PilotNet, which has been used for real life applications as well as for projects in artificial settings such as this. For this project PilotNet is proposed as the vantage point for the convolutional neural network architecture.

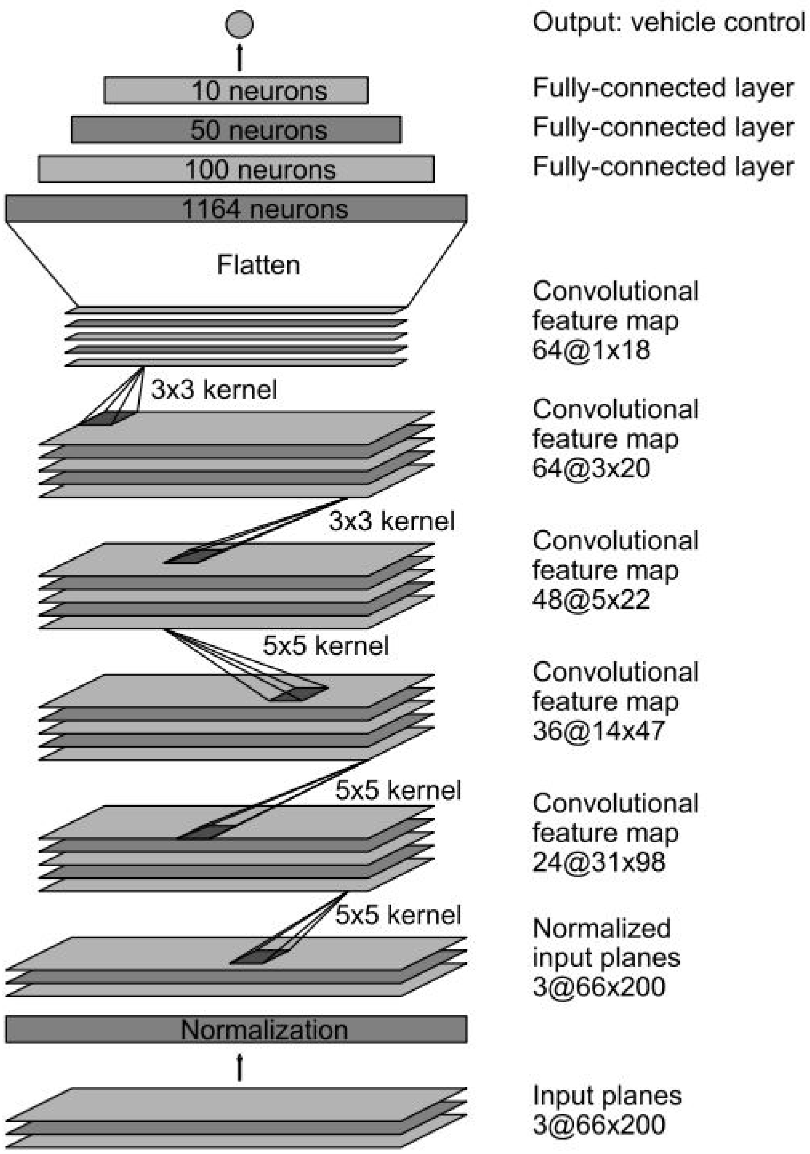


Figure 5.4: Diagram of the neural network architecture of PilotNet [89]

PilotNet utilizes a CNN with a normalization layer, 5 convolutional layers and 3 fully connected layers, which can be seen in Figure 5.4. The inputs are images, 3 channels deep, with a width of 200 and height of 66, which can be adjusted depending on the needs for the image size. The 3 channels are YUV, as opposed to RGB [1]. These images are normalized before feeding them into the network. The filters for the convolutional layers are 5x5 kernels for the first three convolutions with a stride of 2x2 and 3x3 for the last two convolutions, with no stride. Finally the network is flattened, to acommodate dimensionality of the three fully connected layers in the end, resulting in a singular output, that is the steering angle for the car. [89]

maybe better source than wiki

[1] YUV is a different color space encoding than RGB that divides the image into a brightness component (Y) and a blue projection (U) and red projection (V), it is more efficient than RGB and comparably reduces the bandwidth, which is why most videocards render images using YUV directly [92], [93].

For purposes of saving time, this architecture is the point of departure for this project, however it will be subject to a few changes, which will be described in chapter 6.

Ensure this fits with the changes made if any

Part IV

Implementation

Implementation 6

Intro text to Implementation.

6.1 Environment

This section will briefly describe the environment the vehicle was driving in. As mentioned in chapter 2, there are several problems in autonomous driving besides following a lane. Since this project explores the issue of intelligent lane following, it was decided to isolate this as much as possible. The environment for the autonomous vehicle has therefore been made out of a single cardboard material of the same brownish sand color, while for the lanes, blue masking tape has been used for contrast. Examples of this can be seen in Figure 6.1.

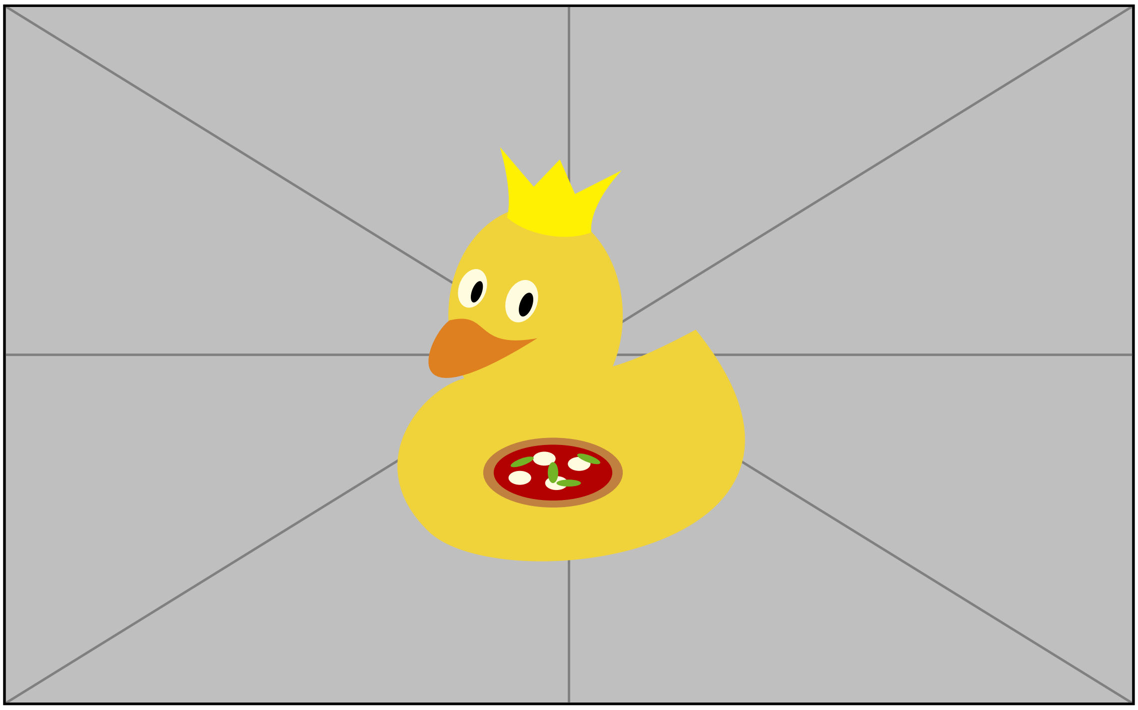


Figure 6.1: One figure or several subfigures of the different lanes

Glemte, hvad jeg mere ville skrive, men der skal i hvert

6.2 Data

This section describes the data used in the implementation of this project.

6.2.1 Dataset

As mentioned in chapter 5, the data should be obtained by manually driving the ev3 and gathering images of the lane in front of the ev3 as well as the steering angle at that point in time. In order to increase variance in the data, it was decided to gather data from three (may change) different lane designs. This introduces several different curvatures, which is useful to train the model to handle a larger variety of situations. The process was time consuming and therefore the data collection was divided into steps, one for each lane design.

The images have dimensions 320x240 with three channels (RGB) and are .png files, sampled with a framerate of 3 fps. In order to decide on a framerate, the speed was taken into account as well as not getting too similar images. A higher vehicle speed, which was held constant, would require the sampled framerate to be higher and vice versa. Since the data was gathered with a relatively low constant speed of the vehicle, 3 fps seemed an appropriate sample rate.

The steering angles are integer values between -60 and 60. This range is due to the limitation of the rotation angles for the wheels in the vehicle design. A steering angle of 0 corresponds to driving straight, while negative and possitive values corresponds to steering left and right, respectively. When driving manually, to obtain the training data, keystrokes from the PC will change the steering angle by a fixed amount of +/- 3.

Check om denne sætning er korrekt

The images are labeled with the corresponding steering angle and by a combination of the training start time as well as the image number, in order to uniquely identify the images and relate them to a specific training session.

Tilføj mere om dataset, evt. se An End-to-End Deep Neural Network for Autonomous Driving Designed for Embedded Automotive Platforms article

6.2.2 Augmentation

6.2.3 Image Preprocessing

Bibliography

[1] *Autonomous Cars*, en. [Online]. Available: https://www.theverge.com/autonomous-cars (visited on 09/26/2019).

[2] *Driverless cars: Explained*, en-ZA. [Online]. Available: http://a.msn.com/08/en-za/BBTqQ8j (visited on 09/26/2019).

[3] *The Future of Autonomous Vehicles*, en. [Online]. Available: https://www.futuresplatform.com/blog/future-autonomous-vehicles (visited on 09/26/2019).

[4] *10 Major Milestones in the History of Self-Driving Cars*, en, Feb. 2019. [Online]. Available: https://www.digitaltrends.com/cars/history-of-self-driving-cars-milestones/ (visited on 09/19/2019).

[5] R. Wallace, K. Matsuzaki, Y. Goto, J. Crisman, J. Webb, and T. Kanade, "Progress in robot road-following", in *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, vol. 3, San Francisco, CA, USA: Institute of Electrical and Electronics Engineers, 1986, pp. 1615–1621. DOI: 10.1109/ROBOT.1986.1087503. [Online]. Available: http://ieeexplore.ieee.org/document/1087503/ (visited on 09/19/2019).

[6] *Waymo*, en. [Online]. Available: https://waymo.com/ (visited on 09/19/2019).

[7] *Waymo has now driven 10 billion autonomous miles in simulation*, en-US. [Online]. Available: http://social.techcrunch.com/2019/07/10/waymo-has-now-driven-10-billion-autonomous-miles-in-simulation/ (visited on 09/19/2019).

[8] N. E. Boudette, "Elon Musk Predicts Tesla Driverless Taxi Fleet Next Year", en-US, *The New York Times*, Apr. 2019, ISSN: 0362-4331. [Online]. Available: https://www.nytimes.com/2019/04/22/business/elon-musk-tesla-autopilot.html (visited on 09/19/2019).

[9] National Technical University of Athens, Austrian Road Safety Board, and European Union Road Federation, "Annual Accident Report 2018", English, Tech. Rep., May 2018. [Online]. Available: https://ec.europa.eu/transport/road_safety/sites/roadsafety/files/pdf/statistics/dacota/asr2018.pdf (visited on 09/19/2019).

[10] P. Thomas, A. Morris, R. Talbot, and H. Fagerlind, “Identifying the causes of road crashes in Europe”, eng, *Annals of Advances in Automotive Medicine. Association for the Advancement of Automotive Medicine. Annual Scientific Conference*, vol. 57, pp. 13–22, 2013, ISSN: 1943-2461.

[11] *Contributing factors leading to road accidents in Great Britain 2017*, en. [Online]. Available: https://www.statista.com/statistics/323079/contributing-factors-leading-to-road-accidents-in-great-britain-uk/ (visited on 09/19/2019).

[12] A. Bull and I. Thomson, “Urban traffic congestion: Its economic and social causes and consequences”, en, Apr. 2002. [Online]. Available: https://repositorio.cepal.org//handle/11362/10867 (visited on 09/17/2019).

[13] A. Bull, United Nations, and Deutsche Gesellschaft für Technische Zusammenarbeit, Eds., *Traffic congestion: the problem and how to deal with it*, ser. Cuadernos de la CEPAL 87. Santiago, Chile: United Nations, Economic Commission for Latin America and the Caribbean, 2003, OCLC: ocm55119072, ISBN: 9789211214321.

[14] J. O. Ukpata and A. A. Etika, “Traffic Congestion in Major Cities of Nigeria”, English, *International Journal of Engineering and Technology*, vol. Volume 2, no. No. 8, pp. 1433–1438, Aug. 2012, ISSN: 2049-3444.

[15] vi.chilukuri.ctr@dot.gov, *USDOT Releases 2016 Fatal Traffic Crash Data*, en, Text, Oct. 2017. [Online]. Available: https://www.nhtsa.gov/press-releases/usdot-releases-2016-fatal-traffic-crash-data (visited on 09/19/2019).

[16] I. KHAN, *Road Safety Facts & Figures*, en, Text, May 2018. [Online]. Available: https://ec.europa.eu/transport/road_safety/road-safety-facts-figures-0_en (visited on 09/19/2019).

[17] Danmark and Havarikommissionen for Vejtrafikulykker, *Why do road traffic accidents happen?*, English. Danish Road Traffic Accident Investigation Board, 2014, OCLC: 904518592, ISBN: 978-87-91458-36-1.

[18] R. Lanctot, *Accelerating the Future: The Economic Impact of the Emerging Passenger Economy*, en, Jun. 2017. [Online]. Available: https://www.strategyanalytics.com/access-services/automotive/autonomous-vehicles/reports/report-detail/accelerating-the-future-the-economic-impact-of-the-emerging-passenger-economy?slid=923371&spg=1 (visited on 09/24/2019).

[19] *3 eller 30 år: Hvornår kommer selvkørende biler på vejene?*, da-DK. [Online]. Available: https://www.dr.dk/nyheder/viden/tech/3-eller-30-aar-hvornaar-kommer-selvkoerende-biler-paa-vejene (visited on 09/24/2019).

[20] matthew.lynberg.ctr@dot.gov, *Automated Vehicles for Safety*, en, Text, Sep. 2017. [Online]. Available: https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety (visited on 09/19/2019).

[21] M. May, *The 6 levels of self-driving car - and what they mean for motorists*, en. [Online]. Available: https://www.thejournal.ie/self-driving-cars-autonomy-levels-3603253-Sep2017/ (visited on 10/07/2019).

[22] *Waymo*, en. [Online]. Available: https://waymo.com/ (visited on 09/19/2019).

[23] M. Daily, S. Medasani, R. Behringer, and M. Trivedi, "Self-Driving Cars", *Computer*, vol. 50, no. 12, pp. 18–23, Dec. 2017, ISSN: 0018-9162. DOI: 10.1109/MC.2017.4451204. [Online]. Available: http://ieeexplore.ieee.org/document/8220479/ (visited on 09/19/2019).

[24] Y. Liu, M. Tight, Q. Sun, and R. Kang, "A systematic review: Road infrastructure requirement for Connected and Autonomous Vehicles (CAVs)", *Journal of Physics: Conference Series*, vol. 1187, no. 4, p. 042 073, Apr. 2019, ISSN: 1742-6588, 1742-6596. DOI: 10.1088/1742-6596/1187/4/042073. [Online]. Available: https://iopscience.iop.org/article/10.1088/1742-6596/1187/4/042073 (visited on 09/24/2019).

[25] A. K. Faulhaber, A. Dittmer, F. Blind, M. A. Wächter, S. Timm, L. R. Sütfeld, A. Stephan, G. Pipa, and P. König, "Human Decisions in Moral Dilemmas are Largely Described by Utilitarianism: Virtual Car Driving Study Provides Guidelines for Autonomous Driving Vehicles", en, *Science and Engineering Ethics*, vol. 25, no. 2, pp. 399–418, Apr. 2019, ISSN: 1471-5546. DOI: 10.1007/s11948-018-0020-x. [Online]. Available: https://doi.org/10.1007/s11948-018-0020-x (visited on 09/24/2019).

[26] H. Surden and M.-A. Williams, "Technological Opacity, Predictability, and Self-Driving Cars", en, *SSRN Electronic Journal*, 2016, ISSN: 1556-5068. DOI: 10.2139/ssrn.2747491. [Online]. Available: https://www.ssrn.com/abstract=2747491 (visited on 09/24/2019).

[27] D. J. Fagnant and K. Kockelman, "Preparing a nation for autonomous vehicles: Opportunities, barriers and policy recommendations", *Transportation Research Part A: Policy and Practice*, vol. 77, pp. 167–181, Jul. 2015, ISSN: 0965-8564. DOI: 10.1016/j.tra.2015.04.003. [Online]. Available: http:

//www.sciencedirect.com/science/article/pii/S0965856415000804 (visited on 09/24/2019).

[28] J. Walker, *Autonomous Vehicle Regulations – Near-Term Challenges and Consequences*, en-US. [Online]. Available: https://emerj.com/ai-sector-overviews/autonomous-vehicle-regulations-near-term/ (visited on 09/24/2019).

[29] A. Group, *The state of autonomous legislation in Europe | Autovista Group*, en-US. [Online]. Available: https://autovistagroup.com/news-and-insights/state-autonomous-legislation-europe (visited on 09/24/2019).

[30] K. Hao, *The three challenges keeping cars from being fully autonomous*, en-US. [Online]. Available: https://www.technologyreview.com/s/613399/the-three-challenges-keeping-cars-from-being-fully-autonomous/ (visited on 09/24/2019).

[31] G. Contributor, *Tech and Legal Challenges the Autonomous Car Industry is Facing*, en-US, May 2018. [Online]. Available: https://www.aberdeen.com/techpro-essentials/tech-legal-challenges-autonomous-car-industry-facing/ (visited on 09/24/2019).

[32] M. Martínez-Díaz and F. Soriguera, "Autonomous vehicles: Theoretical and practical challenges", en, *Transportation Research Procedia*, vol. 33, pp. 275–282, 2018, ISSN: 23521465. DOI: 10.1016/j.trpro.2018.10.103. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2352146518302606 (visited on 09/24/2019).

[33] S. Pendleton, H. Andersen, X. Du, X. Shen, M. Meghjani, Y. Eng, D. Rus, and M. Ang, "Perception, Planning, Control, and Coordination for Autonomous Vehicles", en, *Machines*, vol. 5, no. 1, p. 6, Feb. 2017, ISSN: 2075-1702. DOI: 10.3390/machines5010006. [Online]. Available: http://www.mdpi.com/2075-1702/5/1/6 (visited on 09/30/2019).

[34] *Lesson 1: Sensors and Computing Hardware - Module 2: Self-Driving Hardware and Software Architectures*, en. [Online]. Available: https://www.coursera.org/lecture/intro-self-driving-cars/lesson-1-sensors-and-computing-hardware-LrLty (visited on 09/27/2019).

[35] *What is GNSS?*, en, Mar. 2016. [Online]. Available: https://www.gsa.europa.eu/european-gnss/what-gnss (visited on 09/30/2019).

[36] *The importance of IMU Motion Sensors*, en-US, Nov. 2018. [Online]. Available: https://www.ceva-dsp.com/ourblog/what-is-an-imu-sensor/ (visited on 09/30/2019).

[37] F. Zhang, H. Stähle, G. Chen, C. C. C. Simon, C. Buckl, and A. Knoll, "A sensor fusion approach for localization with cumulative error elimination", in *2012 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, Sep. 2012, pp. 1–6. DOI: 10.1109/MFI.2012.6343009.

[38] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A Survey of Autonomous Driving: Common Practices and Emerging Technologies", *arXiv:1906.05113 [cs, eess]*, Jun. 2019, arXiv: 1906.05113. [Online]. Available: http://arxiv.org/abs/1906.05113 (visited on 09/27/2019).

[39] *YOLO: Real-Time Object Detection*. [Online]. Available: https://pjreddie.com/darknet/yolo/ (visited on 10/02/2019).

[40] *12.3. Object Detection and Bounding Boxes — Dive into Deep Learning 0.7 documentation*. [Online]. Available: http://d2l.ai/chapter_computer-vision/bounding-box.html (visited on 10/10/2019).

[41] A. Mittal, *Instance segmentation using Mask r-CNN*, en, Jun. 2019. [Online]. Available: https://towardsdatascience.com/instance-segmentation-using-mask-r-cnn-7f77bdd46abd (visited on 10/10/2019).

[42] E. Yurtsever, *Driving risk assessment with deep learning using a monocular camera. Related paper: Https://arxiv.org/abs/1906.02859 : Ekim-Yurtsever/DeepTL-Lane-Change-Classification*, original-date: 2018-10-19T02:53:10Z, Oct. 2019. [Online]. Available: https://github.com/Ekim-Yurtsever/DeepTL-Lane-Change-Classification (visited on 10/02/2019).

[43] *TRX-4® Unassembled Kit: 4wd Chassis with TQi Traxxas Link™ Enabled 2.4ghz Radio System*, Feb. 2018. [Online]. Available: https://traxxas.com/products/models/electric/trx-4-kit (visited on 10/30/2019).

[44] *NVIDIA Jetson Nano Developer Kit*, Mar. 2019. [Online]. Available: https://developer.nvidia.com/embedded/jetson-nano-developer-kit (visited on 10/30/2019).

[45] *Best webcam 2019*, Oct. 2019. [Online]. Available: https://www.techradar.com/news/computing-components/peripherals/what-webcam-5-reviewed-and-rated-1027972 (visited on 10/30/2019).

[46] *Best webcams 2019*, Jun. 2019. [Online]. Available: https://www.expertreviews.co.uk/accessories/1406114/best-webcams (visited on 10/30/2019).

[47] *Best webcams for 2019*, Oct. 2019. [Online]. Available: https://www.pcgamer.com/best-webcams/ (visited on 11/07/2019).

[48] *JetPack | NVIDIA Developer*, Aug. 2019. [Online]. Available: https://developer.nvidia.com/embedded/jetpack (visited on 10/30/2019).

[49] LEGO, *Identifying LEGO® set and part numbers.* [Online]. Available: https://www.lego.com/en-us/service/help/bricks-building/replacement-parts/identifying-lego-set-and-part-numbers-408100000009668 (visited on 10/02/2019).

[50] Brick Owl. [Online]. Available: https://img.brickowl.com/files/image_cache/larger/lego-ev3-intelligent-brick-set-45500-15-1.jpg (visited on 09/30/2019).

[51] *EV3 Cable Pack 45514 | MINDSTORMS® | Buy online at the Official LEGO® Shop DK*, en. [Online]. Available: https://www.lego.com/en-dk/product/ev3-cable-pack-45514 (visited on 09/24/2019).

[52] *LEGO® MINDSTORMS® EV3 31313 | MINDSTORMS® | Buy online at the Official LEGO® Shop DK*, en. [Online]. Available: https://www.lego.com/en-dk/product/lego-mindstorms-ev3-31313 (visited on 09/24/2019).

[53] LEGO, *MINDSTORMS EV3 User Guide*, en, 2013. [Online]. Available: https://www.lego.com/cdn/cs/set/assets/bltbef4d6ce0f40363c/LMSUser_Guide_LEGO_MINDSTORMS_EV3_11_Tablet_ENUS.pdf (visited on 09/24/2019).

[54] *LEGO® MINDSTORMS® EV3 31313 | MINDSTORMS® | Buy online at the Official LEGO® Shop DK*, en. [Online]. Available: https://www.lego.com/en-dk/product/lego-mindstorms-ev3-31313 (visited on 09/24/2019).

[55] Brick Owl. [Online]. Available: https://img.brickowl.com/files/image_cache/larger/lego-nxt-intelligent-brick-set-9841-15.jpg (visited on 09/30/2019).

[56] N. Montés, N. Rosillo, M. Covadonga Mora, and L. Hilario, "Real-Time Matlab-Simulink-Lego EV3 Framework for Teaching Robotics Subjects", in *Robotics in education*, New York, NY: Springer Berlin Heidelberg, 2019, pp. 230–240, ISBN: 9783319970844.

[57] LEGO, *MINDSTORMS NXT 2.0 User Guide*, 2009. [Online]. Available: https://www.lego.com/biassets/bi/4589647.pdf (visited on 09/24/2019).

[58] Raspberry Pi Foundation. [Online]. Available: https://www.raspberrypi.org/app/uploads/2019/06/HERO-ALT.jpg (visited on 09/24/2019).

[59] ——, *Raspberry Pi Foundation - About Us*, en-GB. [Online]. Available: https://www.raspberrypi.org/about/ (visited on 09/24/2019).

[60] ——, *What is a Raspberry Pi?*, en-GB. [Online]. Available: https://www.raspberrypi.org/help/what-is-a-raspberry-pi/ (visited on 09/24/2019).

[61] ——, *BCM2835 - Raspberry Pi Documentation*. [Online]. Available: https://www.raspberrypi.org/documentation/hardware/raspberrypi/bcm2835/README.md (visited on 09/24/2019).

[62] Arm Limited, *ARM11 processors*. [Online]. Available: http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.set.arm11/index.html&_ga=2.155964850.889934770.1569486481-1895601341.1569328253 (visited on 09/24/2019).

[63] Raspberry Pi Foundation, "Raspberry Pi 4 specs and benchmarks", *The MagPi Magazine*, Jun. 2019. [Online]. Available: https://www.raspberrypi.org/magpi/raspberry-pi-4-specs-benchmarks/ (visited on 09/24/2019).

[64] ——, *FAQs - Raspberry Pi Documentation*. [Online]. Available: https://www.raspberrypi.org/documentation/faqs/ (visited on 09/24/2019).

[65] Arduino, *Arduino - Introduction*. [Online]. Available: https://www.arduino.cc/en/Guide/Introduction (visited on 09/30/2019).

[66] ——, *Arduino - Products*. [Online]. Available: https://www.arduino.cc/en/Main/Products (visited on 09/30/2019).

[67] ——, [Online]. Available: https://store-cdn.arduino.cc/uni/catalog/product/cache/1/image/520x330/604a3538c15e081937dbfbd20aa60aad/a/0/a000066_featured_1_.jpg (visited on 09/30/2019).

[68] ——, *Arduino Uno Rev3*. [Online]. Available: https://store.arduino.cc/usa/arduino-uno-rev3 (visited on 09/30/2019).

[69] ——, [Online]. Available: https://store-cdn.arduino.cc/uni/catalog/product/cache/1/image/520x330/604a3538c15e081937dbfbd20aa60aad/a/0/a000067_featured_1_.jpg (visited on 09/30/2019).

[70] ——, *Arduino Mega 2560 Rev3*. [Online]. Available: https://store.arduino.cc/usa/mega-2560-r3 (visited on 09/30/2019).

[71] Brick Owl. [Online]. Available: https://img.brickowl.com/files/image_cache/larger/lego-ev3-large-servo-motor-set-45502-4.jpg (visited on 09/30/2019).

[72] LEGO, *LEGO® MINDSTORMS® EV3 and NXT compatibility.* [Online]. Available: https://www.lego.com/en-gb/service/help/products/themes-sets/mindstorms/lego-mindstorms-ev3-and-nxt-compatibility-408100000007885 (visited on 09/30/2019).

[73] Brick Owl. [Online]. Available: https://img.brickowl.com/files/image_cache/larger/lego-ev3-medium-servo-motor-set-45503-15.jpg (visited on 09/30/2019).

[74] ——, [Online]. Available: https://img.brickowl.com/files/image_cache/larger/lego-interactive-servo-motor-set-9842-15.jpg (visited on 09/30/2019).

[75] M. Gasperi and P. " Hurbain, *Extreme NXT: Extending the LEGO MINDSTORMS NXT to the Next Level*, en. Berkeley, CA: Apress, 2009, ISBN: 9781430224532 9781430224549. DOI: 10.1007/978-1-4302-2454-9. [Online]. Available: http://link.springer.com/10.1007/978-1-4302-2454-9 (visited on 10/02/2019).

[76] P. Hurbain, *NXT® motor internals.* [Online]. Available: http://www.philohome.com/nxtmotor/nxtmotor.htm (visited on 10/02/2019).

[77] Brick Owl. [Online]. Available: https://img.brickowl.com/files/image_cache/larger/lego-small-technic-motor-42-grams-28.jpg (visited on 09/30/2019).

[78] P. Hurbain, *LEGO 9v Technic Motors compared characteristics.* [Online]. Available: http://www.philohome.com/motors/motorcomp.htm (visited on 10/02/2019).

[79] Brick Owl. [Online]. Available: https://img.brickowl.com/files/image_cache/larger/lego-large-technic-motor-9v-2838-28.jpg (visited on 09/30/2019).

[80] Logitech. [Online]. Available: https://secure.logitech.com/assets/31671/c910-front-view.jpg (visited on 10/02/2019).

[81] ——, *Logitech HD Pro Webcam c910 Technical Specifications*, en-IN. [Online]. Available: http://support.logi.com/hc/en-in/articles/360023306114-Logitech-HD-Pro-Webcam-C910-Technical-Specifications (visited on 10/15/2019).

[82] MOOG, *How a car steering system works: Easy guide | MOOG*, en. [Online]. Available: https://www.moogparts.eu/blog/how-a-steering-system-works.html (visited on 10/30/2019).

[83] D. Collins, *How Car Steering Systems Work*, en, Jan. 2018. [Online]. Available: https://www.carbibles.com/car-steering-systems/ (visited on 10/30/2019).

[84] P. Gareffa, *All About Front-, Rear-, Four- and All-Wheel Drive*, en-us. [Online]. Available: https://www.edmunds.com/car-technology/what-wheel-drive.html (visited on 10/30/2019).

[85] R. Glon, *FWD vs. RWD vs. AWD: How the wheels that turn change the way you drive*, en, Sep. 2019. [Online]. Available: https://www.digitaltrends.com/cars/fwd-vs-awd-vs-rwd/ (visited on 10/30/2019).

[86] Z. Yang, Y. Zhang, J. Yu, J. Cai, and J. Luo, "End-to-end Multi-Modal Multi-Task Vehicle Control for Self-Driving Cars with Visual Perceptions", in *2018 24th International Conference on Pattern Recognition (ICPR)*, ISSN: 1051-4651, Aug. 2018, pp. 2289–2294. DOI: 10.1109/ICPR.2018.8546189.

[87] A. Patil, *ConvNets & Driverless Cars*, en, Mar. 2019. [Online]. Available: https://medium.com/datadriveninvestor/convnets-driverless-cars-997976c5ee4c (visited on 11/18/2019).

[88] J. Kocić, N. Jovičić, and V. Drndarević, "An End-to-End Deep Neural Network for Autonomous Driving Designed for Embedded Automotive Platforms", *Sensors (Basel, Switzerland)*, vol. 19, no. 9, May 2019, ISSN: 1424-8220. DOI: 10.3390/s19092064. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6539483/ (visited on 09/30/2019).

[89] M. Bojarski, D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to end learning for self-driving cars", Apr. 2016.

[90] A. Nain, *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*, en, Jun. 2019. [Online]. Available: https://medium.com/@nainaakash012/efficientnet-rethinking-model-scaling-for-convolutional-neural-networks-92941c5bfb95 (visited on 11/18/2019).

[91] X. Zhang, W. Yang, X. Tang, and J. Liu, "A Fast Learning Method for Accurate and Robust Lane Detection Using Two-Stage Feature Extraction with YOLO v3", *Sensors (Basel, Switzerland)*, vol. 18, no. 12, Dec. 2018, ISSN: 1424-8220. DOI: 10.3390/s18124308. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6308794/ (visited on 11/18/2019).

[92] *YUV*, en, Page Version ID: 921160678, Oct. 2019. [Online]. Available: https://en.wikipedia.org/w/index.php?title=YUV&oldid=921160678 (visited on 11/18/2019).

[93] *Color Spaces in Frame Grabbers: RGB vs. YUV*. [Online]. Available: http://www.sensoray.com/support/appnotes/frame_grabber_capture_modes.htm (visited on 11/18/2019).

Part V

- Appendix -