



University of Wales Trinity Saint David
School of Computing

**An Investigation into Artificial Intelligence and Robotics and the
methods and technologies that are used in the production of the
applications**

By Kieran John Ayres

Supervisor : Dr. Gary Jones

Second Marker : Dr. Tim Bashford

Project submitted as part of the requirements for the award of BSc (Hons) Computing
(Software Engineering).

23rd April 2021

Abstract

This report has been produced in response to the project aim, *an investigation into Artificial Intelligence and Robotics and the methods and technologies that are used in the production of the applications*. Artificial Intelligence is one of the biggest trends, currently in the technology world. It is commonly put into practice along with another field, Robotics. The purpose of this, is to investigate it and to implement a solution for a miniature self-driving car. “The global autonomous car market is expected to grow from \$5.6 billion in 2018 to \$60 billion in 2030. The market definition includes component production and investments”. [1] This is a 100% increase so the demand for autonomous vehicles is growing increasingly

The objectives of the project are as follows:

1. An Investigation into Autonomous Robotics and what methods and technologies are used to produce them.
2. To design and develop an Automated Robotic System to simulate an autonomously driven vehicle.
3. To evaluate and synthesize the development project of an Automated Robotic System.

Table of Contents

Contents

Abstract.....	2
Table of Contents.....	3
Table of Figures	Error! Bookmark not defined.
Chapter 1 - Introduction	7
Chapter 2 – Research & Review of Literature	9
2.1 – An Introduction to the Field of Research	9
2.2 Artificial Intelligence – Brief History.....	12
2.3 - Robotics – Brief History	13
2.4 – Types of Machine Learning Algorithms	14
2.4.1 – What language to use to write Artificial Intelligence Algorithms?	15
2.4.2 – Machine Learning Algorithms Continued – <i>Reinforcement Learning</i>	16
2.5 - Computer Vision.....	17
2.6 – Image Processing.....	18
2.6.1 – Image Processing Techniques	18
2.7 – Autonomous Vehicles	20
2.7.1 – Miniature Scale Car Project Research – What techniques are used?	21
2.8 – Circuit Boards	22
2.8.1 – Arduino	22
2.8.2 – Raspberry Pi	22
2.8.3 - PIC.....	22
2.9. How does the Research effect the Project?	Error! Bookmark not defined.
Chapter 3 - Research and Development Methodology	23
3.1. Philosophy	23
3.2. Strategy	Error! Bookmark not defined.
3.3. Development Methodology	23
Chapter 4 – Design and Implementation	24

4.1 – System Specification	24
4.1.1. Hardware	24
4.1.2. Software.....	24
4.1.3. Design Decisions	25
4.2. 1 st Feature.....	25
4.2.1. Design	26
4.2.1.1. Algorithm Process	26
4.2. Implementation	27
4.2.3 – Testing.....	27
4.2.3.1. Unit Testing.....	27
4.2.3.2. System Testing	27
4.2.3.3. Usability Testing.....	28
4.2.3.4. Acceptance Testing.....	28
4.3. 2 nd Feature	29
4.3.1. Design	29
4.3.1.1. Robotic Process	29
4.3.2. Implementation	29
4.3.3. Testing.....	30
4.3.3.1. Unit Testing.....	30
4.3.3.2. System Testing	30
4.3.3.3. Usability Testing.....	Error! Bookmark not defined.
4.3.3.4. Acceptance Testing.....	Error! Bookmark not defined.
3 rd Feature	Error! Bookmark not defined.
Chapter 5 – Evaluation.....	30
5.1. Results and Discussion	Error! Bookmark not defined.
Chapter 6 – Conclusions.....	30
6.1. Recommendations for Further Work.....	31
Chapter 7 – Reflection	32
References and Bibliography	32

Appendices	36
------------------	----

Chapter 1 - Introduction

The focal point of this document is to produce a report to investigate, implement and conclude the project that is to be completed during the 2020/2021 academic year. This project was to design and implement a machine learning algorithm within a Robotic car to simulate an autonomous vehicle, this would be implemented with a focus on a field of artificial intelligence that will be researched in chapter 2. The Robotic vehicle could be assembled in a number of ways, such as using an Arduino or Raspberry Pi, these are extremely powerful programmable circuit board and computers. They can be utilised in various different ways, possible Arduino projects include; a Robotic arm, and an RC controller [2] an FM radio and a network monitoring station [3]. This would be either fitted to the car or would be built as the car where the algorithm would be implemented to control the car using the related IDE. There are a variety of ways in which the image processing could be performed, there are cameras on the market to do this specifically or another way is to have two circuit boards stacked together. Such as the Raspberry Pi attached to the Arduino as the Arduino isn't as capable as the Raspberry Pi in relation to running Artificial Intelligence Algorithms.

The aim of this project is to simulate an autonomous vehicle, to reach this overall aim there will be several objectives that will need to be achieved. The student had set themselves a project that would involve them learning about technologies that they had very minimal experience using these so the objectives would be completed through doing a series of tasks in order to achieve them. The objectives are:

1. An Investigation into Autonomous Robotics and what methods and technologies are used to produce them.
2. To design and develop an Automated Robotic System to simulate an autonomously driven vehicle.
3. To evaluate and synthesize the development project of an Automated Robotic System.

These objectives are in broad terms, the first objective, *an investigation into Autonomous Robotics and what methods and technologies are used to produce them*, this objective will be completed over several months as this will be the base of the project in which it can be expanded and further built upon to complete the remaining objectives, there are new technologies that need to be investigated and then learnt to start the project. The student will need to research to learn a further insight to which are the best technologies to be utilised in the project, this will be done by reading literature and practicing with the chosen technologies. As an example, when deciding which programming language to integrate, a substantial amount of research would need to be produced to choose which would be the best, as some programming languages may be designed to support machine learning algorithms better than what others are. They will then need to go further and research the field of Artificial Intelligence and then practice in writing machine learning algorithms. This will also be researched to choose the correct frameworks/software packages, such as Tensorflow or OpenCV. Once the following has been investigated, the student will produce the first part of the written report, detailing their research and decisions to progress onto objective two.

This will then give the student a foundation in which they can use to design and develop the software and the miniature car which will allow them to complete objective two, this objective is *To design and develop an Automated Robotic System to Simulate an Autonomously Driven Vehicle*. This objective will be completed through a series of tasks, this includes, using software to design the miniature Robot as this is helpful when they assemble the electronics later on in the project. This is so it designs a blueprint of what the hardware will look like when it is at the build stage. They will then build the car, they have opted to give the car a wooden base as this is easy to edit, and drill holes within the wood to hold any wires in place safely. They will then need to develop and implement their code onto the microchip to be downloaded on board the miniature scale car, this should allow the car to run as the relevant code will be running which will then instruct the motors to run, while performing the image processing where applicable. When choosing what software/hardware packages to use when designing the Robotic car, it is essential that all avenues are explored, such as what hardware simulation software, or what hardware or software to use for the development aspect of the project. This objective will end by including thorough testing of the hardware/software. With the test cases, written out and explained.

The third objective was *To evaluate and synthesize the development project of an Automated Robotic System*. The last objective is reliant on the previous two being completed efficiently. This would need a timescale of a minimum two months. This will be completed in the form of a written report and will show different things from the project. For example, this could include an analysis of the algorithm that they developed, showing it working on a designated test case. They could further analyse the efficiency of the algorithm. By allowing themselves two months, this then allows the student time to collect the relevant data so they can thoroughly analyse their project. To follow on, it will grant them enough time to explore what their results have taught them while evaluating them before closing with the conclusion and reflection of the project.

The last paragraph of the introduction will further outline the rest of the report, this will include, A section is written so there can be an extensive look at the research that the student has done in order to clearly construct the project. This will include a range of books and journals and research papers to support why they made the decisions for the project. For example, the student would have done the relevant research when picking the choice of language which is efficient for their use in the project. This is also a chance to research the field, as it is a broad field. This will help discover the other types of research that has been done, and how they may relate or differ from other sets of research. This will be presented in the format of an essay within the project report. Secondly, a breakdown of the design and implementation of the project, including the development methods that were used to design and write the Artificial Intelligence algorithm. There will also be a testing schedule, these will be broken down into four sets of testing. Which include unit testing, system testing, usability testing and acceptance testing. The testing stages will need to be thorough, with a number of tests completed to validate the software. The usability and acceptance testing will be completed along with selected people so it can be clarified as the miniature scale vehicle and code performing what they are supposed to. Then there will be an overall analysis of the work that has been done, followed by a full evaluation of these with an emphasis on the performance and completion of the project. The conclusions will then be drawn followed on by a reflection of the project.

Chapter 2 – Research & Review of Literature

Chapter 2 of the report was written and intended to be formatted as an essay. This gave freedom when researching the chosen field that they wanted to explore for the project, that being the fields of Artificial Intelligence and Robotics. These can be classed as individual fields', but they integrate with one another in developing real world applications. This allowed research in multiple areas, research areas investigated include: //include list here.

2.1 – An Introduction to the Field of Research

This chapter aims to give the reader a short introduction into the fields of research, the first being Artificial Intelligence and the second being Robotics. These are huge areas of research and development, with many outcomes rising from this.

To begin, there are many people who often wonder “What is Artificial Intelligence?”, this question strikes many people, outside the world of the professionals, it's almost an abstract and grey area. Truthfully, the exaggerated applications through the media make it more in demand and questionable amongst everyone. What it is, is a blend of science and engineering in which it is utilised to build programs that can mimic human level intelligence behaviour, these are implemented in different types of applications and in some cases, they are integrated within Robotic applications. This makes it a very interesting topic amongst professionals as it can be used to do things that normal applications maybe couldn't do. When designing these applications, they are designed and programmed with Computational Psychology in mind, Computational Psychology is the idea that computer algorithms can be designed with the intention that the application can “behave in the same way people do” [4]. These, however, are never for producing perfect results, Computational Psychology intends the programs to behave as a person would. For example, where mistakes are possible, they would be possible for the program too [4]. There are many sub-fields of Artificial Intelligence, the most popular is, Machine Learning which in itself creates more sub-branches, one of these is Computer Vision. The main idea behind Machine Learning is the idea that a learning algorithm can be implemented so a machine can be trained to understand something with extensive data inputted like a human would be able to [5]. For example, there's models which collect the data in order to tailor a customer's shop to their interests when shopping online. Computer Vision adopts the same method as Machine Learning, you would train an application to recognise something. What's interesting about Computer Vision is that it is commonly implemented within Robotic applications to allow the Robot to navigate.

To continue, Robotics is another field where, as a collective, the media also over-exaggerate the capability of the applications within Robotics. However, this isn't to say that there isn't a large amount of depth within this field. The field makes strides every year to improve with people doing new and wonderful things with this technology. Robotics, like AI, is another grey area where no one has a definitive definition. However, they can be defined as "A machine capable of carrying out a complex series of actions automatically, especially one programmable by a computer" [6]. Within the field of Robotics, there are some proposed laws associated with Robotics. A man named Isaac Asimov devised three laws which engineers should follow in the production of Robotics. The laws are as follows:

1. "A Robot may not harm a human being, or, through inaction, allow a human being to come to harm". [7]
2. "A Robot must obey the orders given to it by human beings, except where such orders would conflict with the First Law." [7]
3. "A Robot must protect its own existence, as long as such protection does not conflict with the First or Second Law" [7]

In this journal, the authors' interestingly point out that Asimov views the human race as superior to any engineered Robots. The journal also states that these rules are neglected amongst Robot developers. As these are not seen as realistic because the Robots that we see in the media are not a realistic representation of the real-world applications that are developed in today's society. This paper also summarises the difficulties behind the constraints of Robots, this is explored by showing the difficulty behind the capability of a Robot which has human like behaviour in terms of how they act. As computers would have a harder time understanding the concepts that humans wouldn't struggle to grasp. For example, the idea of following the law may not be the easiest idea to grasp for a machine [7]. There are two common types of Robotics. Mobile Robots which are capable of being autonomous and are utilised for both research and practical purposes [8] a practical example of this, is the Robotic cleaner. This application is part of a group of mobile Robotics that are developed to complete some of the tedious tasks that humans don't always enjoy. These Robots are in some compliance with the laws thought of by Asimov. It's essential that these can navigate themselves without being in awkward instances where it may cause danger. These Robots do not solely rely on hard coded human design as these can fail, to expand on this, if a sensor was to fail, and a Robot relied on it for object avoidance, such as an ultrasonic sensor, it would then not be able to do this anymore. In some cases, neural networks are implemented within Robots so they will associate their "sensory reception with motor response" [9] A paper by Ulrich Nehmzow explored this. This then would allow the

Robot to learn and adapt, which is an interesting idea to apply to the project. Nehmzow also discusses different aspects of Robotics in the paper, such as navigation fundamentals. Industrial are utilised commonly across different sectors in the world. These can be stand-alone machines which perform tasks more efficiently than what a human would, or they perform due to the danger that job could propose on a human. Tasks such as welding would be a safer utilisation of these Robotics. These do require safety measures in place to ensure human safety when working with the machinery. There are different kinds of stand-alone robots, some include, articulated, which are applied for welding, there is delta, which is applied when packing. When a book publisher is wanting to release a new book, these would be utilised to pack the boxes with great careful precision [10]. This is not applicable during the project as the application intended will be mobile.

These topics are both wide areas of research and professional implementations with new associating technologies being discovered and explored every year. Artificial Intelligence and Robotics and some of the research and implementations of it will be explored more in depth in the following chapters.

2.2 Artificial Intelligence – Brief History

For decades, computer scientists have implemented applications that mirror human like behaviour. These were designed from some researchers with the idea of giving the application rules of logic to respond to situations [11]. In 1950, Alan Turing viewed as the “Godfather” of computer science posed the question of “Can Machines Think?”. This was answered with a game in which was known as the imitation game but is now known as the Turing test, this was a game which would consist of three players, these can be defined as A, B and C. Where, A is the Man, B is the Woman and the C is the interrogator, however, the three players can’t see nor hear each other’s real voices, in the 1950s they would communicate via teletype which is the equivalent of a Microsoft Teams chatroom. The interrogator then would ask questions where B would need to prove her identity and A would attempt to fool the interrogator that they are B. However, during the game, A would be swapped for a machine and if player C is unable to tell A from C still, then the machine (A) is concluded as intelligent [12] [13]. This was a milestone in AI, although there had been research prior, this advanced the field forward to develop AI further. 6 years later, the logic theorist was developed, this wasn’t a fully fledged AI, but this was a program which was deemed, intelligent as it could provide proof of theorems in Mathematics, one of the first of its kind to mirror human ability in problem solving. [14]. Two years after this, the LISP programming language was developed which has been utilised to write AI programs on countless occasions since. [15].

2.3 - Robotics – Brief History

Robotics has been a massive computer science discipline with predictions that it will be the driving force behind the fourth industrial revolution. It has made advancements over the last few decades, propelling it to what it is now. The term robot, dates back to 1921, where it was mentioned in a play by Karel Čapek. Then Isaac Asimov spoke of them in a book where he spoke of the laws briefly mentioned in the introduction [16].

There were real strides in later decades. It was the early 1950s when the first generation of robotics were introduced, this generation were the first manipulators, they didn't have much date of their surroundings and they were controlled point to point, simple in comparison to what the world sees in today's society. This saw prototypes and tools developed up until 1961 where UNIMATE was developed. This is viewed as one of the first programmable industrial robots, this allowed the developers to start the first ever robot company, UNIMATION. This boosted the industry to offer more resources in the development of robotic systems [16].

To continue, the second generation of robotics were introduced in 1968, these were robots that had sensors. These were better as they could react better to their environment, for example by using an ultrasonic sensor they could stop themselves from crashing into things, these are utilised especially in the automotive industry. It was during the second generation where programmable logic controllers became more commonly used in the automation industry. It was also this period, in 1973, which saw the first industrial robot [16].

Subsequently, in 1978, the third generation of robotics started, this expanded for 21 years, these robots were built with new dedicated programming languages, this was partly because competition was so intense that companies wanted the robots to be hard to duplicate. These robots were reprogrammable and there was a partial integration of artificial vision, which are both technologies that are used in current applications. The third generation also saw the applications being developed more commonly outside of industrial robotics, with companies building robotic kits for consumers to purchase. [16]

The fourth was in the year 2000, which saw the implementation of intelligent robotics, these are what are focused on currently as they have strong computing capability and can learn due to AI technologies which allows them to act on more reliable input data. The one negative to these developments is that it has slowed down the production on industrial scale robotics. This generation has saw the introduction of the implementation of neural networks within robotics along with fields such as computer vision being vastly improved and brought closer to being truly artificial vision. [16]

2.4 – Types of Machine Learning Algorithms

An algorithm can be defined as a set of instructions for any given task [17] and not just in computer science terms, an algorithm can be making a cup of tea, where the algorithm begins when the kettle is boiling and finished when the tea is ready. There are many variations of AI algorithms. Which makes it impossible to explain them all, instead this will briefly discuss some algorithms for machine learning. Machine Learning has subsets, these are deep learning, computer vision and many more, these have algorithm types. Some of these are supervised learning, unsupervised learning, and reinforcement learning algorithms. [18].

Supervised learning is an algorithm where the machine is provided with a set of data, these will be labelled with expected output when given an input. This is trained until the machine can produce the accuracy that the developer would want. These algorithms would be trained to recognise patterns in data sets. Neural Networks are a form of supervised learning and one of the common applications of supervised learning is email-classification, used by companies to determine spam from reliable emails. Unsupervised learning is less predictable as the algorithm is supplied with input variables but not any expected outcome and then the goal for it is to recognise patterns and structures in the data. This isn't formerly trained whereas in supervised learning the algorithm is trained. This isn't as commonly assessed against the accuracy as it isn't formerly trained. Reinforcement learning is a little different as it has a trial and error process where it learns from its own actions. Each time it succeeds in what it is intended to do, it is given a reinforcement signal and that information of the accurate action is stored which it will then use when it is in a similar situation. This could be applicable to autonomous robotics, for example when training a car to drive itself along a road. [18] [19].

In terms of the project, a way on which these could be used is by implementing the reinforcement learning algorithm to train the miniature scale robot to drive along a track. This would be useful as each time it reaches a point where it should stop driving it would continue to make this error until it can drive past that section confidently.

2.4.1 – What language to use to write Artificial Intelligence Algorithms?

It's essential to research before development, this allows the student to choose the most suitable programming language for the parts which involve Artificial Intelligence. They could choose a language for a number of certain factors but could purely because it's what they feel most comfortable using. If this was the case, then C/C++ would be chosen as they have the most experience writing in those languages.

The first process to this is to compile some of the most popular languages used in these technologies. Some of the languages include C++, Python, and Lisp. However, it's good to note these are only a few languages that are utilised in the use of Artificial Intelligence programming. [20].

To begin. At the heart of it, you can't truly name a best language for AI, Python is widely regarded as one of the best, it vastly grows each year. Python is a high-level language, the language itself is regarded so highly for the readability, which in turn makes it easier to learn as people refer to it as the language that is the closest to plain English when written, where as long as you understand simple syntax and processes of programming it shouldn't be too difficult to learn. The code is also written in far less lines of code. Python can handle all types of coding requirement, whether it's coding small projects or huge business scale projects so it's suitable for every type of developer's needs. One of the main advantages to Python is that it has a huge variety of libraries, this is credited in a paper by K.R. Srinath as the reason behind its growth, because it is attractive to those interested in Data Science. [21]. Another advantage the paper mentions, is that as it is high-level, it completed difficult tasks for the developer, like managing the memory of the code. Which in itself would be a compelling reason in choosing it over other languages. As C++ requires this. Other advantages include that, it is open source, it's portable, for example, it can be used in Linux and windows, useful if the project was to use a raspberry pi and an Arduino. The dominant advantage is the libraries available to Python, for example, Tensorflow can be used to develop neural networks. However, there are some disadvantages, it is slow, as it is flexible with how it is written, when it runs, the machine requires some double checking, for the want of a better phrase, to understand the linking in the program, this slows the overall performance but, this isn't overly important because the project doesn't require rapid code build times. A drawback that could affect the project, is the maintainability of Python, as the code becomes a larger script, overtime, this will become harder to find and fix any errors. This can be avoided by keeping the code well documented as it is written. [21]

There are other languages that are used in AI programming, LISP won't be researched for this as it is seen as more of a study in programming history as this language was even before AI was firstly presented but it is known that a lot of programming languages build on its core features, but LISP doesn't have an updated library and is known to be difficult to understand. [20]

C++ is the language that the student is most familiar with, thus, making this seem like the easiest first choice. This isn't entirely true, academic research on it as a programming language has been challenging to find, but in general online reading, it is viewed as one of the fastest for execution time which makes it good for those code builds that need to be efficient, not necessarily for this project though. It is also widely used in neural networks and machine learning, like Python, it has a wide variation of libraries to use, so for a developer, this is helpful as they have more tools to use when making their AI program. It can also use the OpenCV library, as can other languages. A drawback to this though is that C++ is better when using it to build particular algorithms. As a whole though, it is seen by the community as a way to build a program which is efficient and fast to run. [20] [22].

//C++ is the gap in the research to exploit

2.4.2 – Machine Learning Algorithms Continued – *Reinforcement Learning*

Reinforcement learning was the technique that was of the most interest and seems as if it would be the most applicable to a car. An example of pseudocode to show the possible process is here:

These algorithms are more likely to be trained in a simulation rather than a real environment as that poses more risk.

```
Begin
Drive
Car would crash
Store the variable recording the crash
Test again

/it would then be trained continuously until the crashing doesn't occur
```

2.5 - Computer Vision

Computer Vision is another sub-branch of Machine Learning. Computer vision is developed for a computer to look at an image and recognise something, one of these things that it does first, is image processing. This, in a way prepares the image before it is extracted and understood fully by the machine. Most importantly to note, it is so a computer can do what a human eye can do, recognise something naturally and extract information from it. In the case of a human, they would be able to look at a pet and be able to understand instantly what it is, computer vision aims to do this as well. [23]

There are many applications of computer vision. One is for interaction from human to computer, this is designed with the computer mimicking human level intelligence, in a way where the computer can, for example, accept audio inputs and then respond how a human would. It's also a field where researchers want machines to understand human behaviours as well as speech. Another application is a environment which is intelligent, so for example, a smart house, where once a person has entered the room, the system would identify the person and then act accordingly, if it was someone who lived at the house it would respond by switching on the heating. However, if it wasn't, it would react by holding footage to record the break in. Another interesting area is in intelligence embedded robotics, this is the development of robotics which will be able to move, see and act in a way similar to how a human would behave. [24]

The ways in which computer vision could be related to the project is using a camera on the autonomous vehicle to identify smaller scale road signs and respond by following the rules of the road sign, for example, if there was a sign instructing the vehicle to stop it would then stop and would be stationary until the sign became a sign signalling to move.

2.6 – Image Processing

Image processing is a vast part of both AI and Robotics. This is used in cases where sensors can't be utilised to perform safely. By using the technique of image processing, a Robot can capture much more detailed data based on the surroundings. This is useful when implementing an autonomous robot. The issue with this is that there is a lot involved when image processing, humans perform this so naturally, when in reality the brain uses 30% for vision, which is nearly a third. Although this field has advanced massively, it still isn't quite capable of what a human is.

In terms of the project, the idea behind the image processing originally was it to be configured to recognise a road sign, to which it would change the car's status depending on what the road sign is [25]. They would do this by enhancing an image's colour in certain areas to make it recognised for a trained algorithm. For example, figure 1 shows a possible simple process.

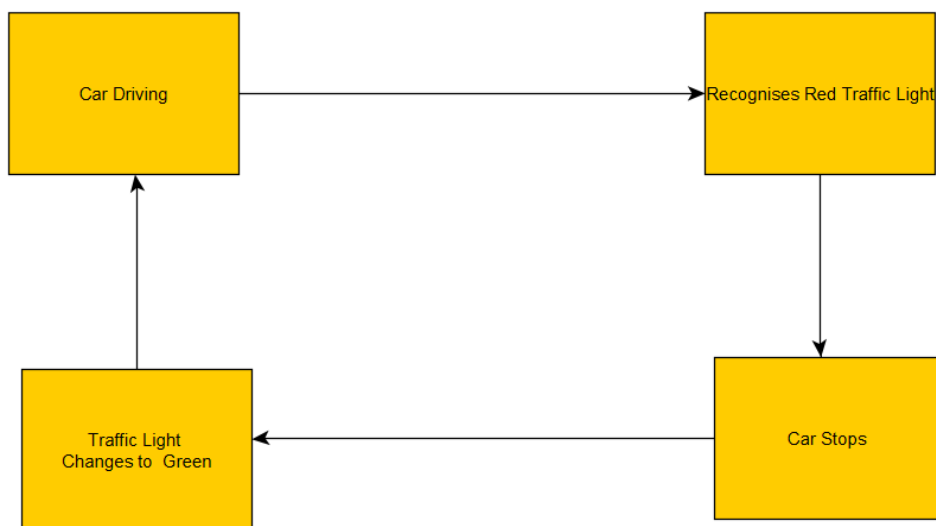


Figure 1 - Traffic Light Detection Process

2.6.1 – Image Processing Techniques

There are loads of different techniques used when image processing, these include techniques such as anisotropic diffusion, which enables the developer to reduce the image noise without having to delete parts of the image, then there is linear filtering, this is when time varying input signals are processed to produce output signals, then there are also self-organising maps, which is related to classifying images into different classes. There are also techniques such as principal components analysis which can be used when feature extracting. A developer could also perform image restoration, this refers to the estimation of a clean original image out of the corrupt image taken to get back the information lost.

2.7 – Autonomous Vehicles

This project is focused on the autonomy of robotics, with emphasis on a miniature scale car. It felt that it was right to research this area to understand how an autonomous vehicle would work. A better understanding was built by looking at both major production autonomous vehicles and research papers that were based around miniature scale cars. The term autonomous car refers to a car which can drive itself without any assistance. They can also be driven through the use of “sensing it’s environment” [26]. Autonomous cars have been developed for a long time but have only been approved for public road testing in the last 7-8 years. The Society of Automotive Engineers (SAE) developed a classification standard for these vehicles. They include six levels. These are a guide and the first three levels dictate that the driver is in charge of monitoring the environment around the car. The levels are:

- No Automation – Level 0. These are mostly driven today, these are controlled and monitored fully by the driver [26].
- Driver Assistance – Level 1. These vehicles can help with things such as braking but the drivers must always be ready to assume control of the car [26].
- Partial Assistance – Level 2. Cars have control over Steering, Throttle and Braking. Immediately returns control to the driver should it detect something it can’t understand [26].
- Conditional Assistance – Level 3. The car can monitor the environment. Use the Steering, Throttle and Braking, in less complicated environments such as Motorway but will prompt the driver to resume control when needed [26].
- High Automation – Level 4. Mirrors everything that of Level 3 but for a wider variety of environments [26].
- Full Automation – Level 5. These can travel from destination X to Y and make its own decisions [26].

Autonomous cars can be implemented to work in different ways. For example, they can use a LIDAR sensor which can calculate a distance surrounding the car. This will help produce an artificial scale map of the surroundings to which the car will then use corresponding sensors to detect their road position and distance from objects. The AI within the car will control this while consulting google maps to be alert for speed changes, speed cameras etc. These cars could be driven by humans in theory but as they are level 5, they would be tested to work with little flaw to dwell on [26]. [16]

2.7.1 – Miniature Scale Car Project Research – What techniques are used?

This chapter is based on research projects that make mini robotic cars, it will briefly mention what's been used in their projects and reference those papers as evidence. It is divided into three sections, these are:

1. AI techniques
2. Hardware
3. Software

The miniature scale versions are much less complex in terms of technology, development, production, and cost efficiency. There are different techniques in ways the autonomy is achieved. Two of the most common methods are through neural networks and using reinforcement learning [27] [28]. Reinforcement Learning is applied in these projects, this is safer in a smaller scale car as it is based on the car trailing and erroring its decisions, meaning should such a car be tested on a public road, it would get better by making error making it unreliable.

There are common hardware choices in these implementations. From research it is clear to see that the ultrasonic sensor can be used for a multitude of things, such as parking the car and obstacle detection [29]. The Raspberry Pi is the popular choice for the AI programming [25] [27] [29]. There are other options for real-time camera feedback, such as the ESP-32 CAM [30]. This can offer a video link streamed to the machine from the camera. There is also the Pi Camera which can perform similarly to that of the ESP-32 CAM. They could also use a low-cost web camera. Some of the cars in the research use a L298 H-Bridge Motor with a PWM system to control the “speed and direction” [28]. This motor allows you to control “two DC motors independently in either direction” [25]. A similar hardware choice is the L298N, this allows control of four motors with ease [31].

These are compatible with both Arduino and Raspberry Pi too [25]. Arduino is another popular choice,

Python has been spoken about previously, mentioned for its usability and access to a vast selection of libraries, with huge compatibility for AI programming, this language has been used in various projects, with almost all projects using it as its programming language for AI [25] [29]. A project completed by Tiwari and Singh used a car which was predominantly controlled using Raspberry Pi and Python [29]. Another library which is used, is OpenCV, OpenCV is a library, written in C++ and is utilised for “real time computer visions” [25] [29]. This is open source so it can be used in developing applications to image process. There is also similar open-source software, called Tesseract OCR for image processing which can be used [25].

2.8 – Circuit Boards

This section of the research was to help with what circuit board type hardware to choose to build the project. The following chapters include brief information on three of these; Arduino, Raspberry Pi and Pic

2.8.1 – Arduino

Arduino is a form of microcontroller, it has lots of hardware expansions, similar to Raspberry Pi, with the two being able to connect together for projects. It has its own dedicated IDE [30]. Which is lightweight and reasonable easy to understand, the programs are compiled in the IDE before being moved onto the Arduino where they run. There is a range of projects, such as obstacle avoiding cars [30].

2.8.2 – Raspberry Pi

The Raspberry Pi was first introduced in 2012 [16]. It's a circuit board but with more power than what Arduino or the PIC has. They are mini single board computers with high computing power, they have been used for many different kinds of projects by computer science enthusiasts and not one field, many different areas, such as to be used as a server or for drones [32]. They can do this while using the various GPIO (General Purpose Input Outputs) available to them. The board is made up of several different components, such as USB ports and a camera connector, the developers of Raspberry Pi also develop hardware to increase the variety of projects that can be built with it [32], such as a miniature car [25]. There are advantages to Raspberry Pi, like it has loads of hardware add-ons to explore, it's energy efficient and is based on a Linux OS with a lot of different applications to it [32]. There are also disadvantages, it isn't compatible with windows, so for those who aren't familiar with Linux, it makes it a daunting task to learn [32].

2.8.3 - PIC

PIC is a micro-controller. The first one was built in 1971, by Gary Boone and Michael Cochran. They are built for specific tasks. They can be utilised in things such as a TV-Remote. The circuits for PIC are made via software called Proteus, this is a simulation tool which can run the code for the circuit in a virtual environment. This is an IDE with various components at the disposal of the developer, it also has other advantages such as bracket highlighting when developing code. It is known as a micro-controller as it is small, and it can control objects or processes [33]. Microcontrollers are widely used in cars. Some have one to control the engine, or a number of them to control different parts of the car. They are used in smaller systems due to smaller storage space. They are programmed in Micro C and the code compiles as a hex file before being ran on the chip [33].

Chapter 3 - Research and Development Methodology

3.1. Philosophy

The philosophy chosen for the project is positivistic, this is utilised in this project as it relies on data collection and observation. This applies as during the development of this project, during the testing of the car, it will be observed during testing the features, it will be observed to see if it is functioning in the way that was predicted, in the design the car would be expected to obstacle avoid which is what would be observed, with the data being collected from acceptance testers, this would be used to gain data to decide whether the software was doing what it had been designed to do. Data will also be collected through the sensor input from the robotic circuit, this will be used for analysis to then refine the code for it to be developed further.

3.2. Development Methodology

The software development methodology that has been chosen for this project is Feature Driven Development, it is an agile methodology. This is being utilised as this is a large project so it would be better to be able to implement features, and if all features aren't achievable, the project will still have functionality and even in its planned first stage, it still would have good functionality, it's advantageous to the project as it breaks up the functionality into features so the overall achievable target of an obstacle avoiding robot can be divided into achievable features. Feature Driven Development, or FDD allows the development to be achieved based on added functionality through stages. The one disadvantage to FDD is that if things are not planned carefully, then this could lead to trouble in the implementation of the feature. This will be avoided by carefully planning out the implementation through a blend of research, pseudocode/algorithm planning and flow diagrams to show how that feature would work. There are three stages to this project. Each feature would be planned, implemented, and tested, each feature will be tested thoroughly with a mix of system, acceptance and unit testing being conducted upon the completion of each feature. The first feature will be to have a robot which can detect an obstacle in front of it and then react accordingly by not colliding. The second feature of the project will be to have an object avoiding robot, so it can detect and navigate around any potential obstacles. At this stage, the project could be perceived as an autonomous vehicle. The 3rd feature would then be to use a Raspberry Pi to implement an image processing algorithm which would allow the car to stop completely if it comes to a miniature-scale Stop sign and then start again based on the Go sign. This is an ambitious feature but will add an extra dimension of autonomy and a good example of how Artificial Intelligence is used within vehicles.

For the development, there's specific hardware needed, the first is a Robo-Vacuum, this is being utilised as it has all the necessary sensors/bumpers and motors to drive the car without needing it to be built from scratch. Raspberry Pi is also a piece of hardware that will be used. The software being used will be Proteus as this is great for integrated development within the microcontrollers. YED Graph Editor is also being used as a tool to show flow diagrams of any designed algorithms.

Chapter 4 – Design and Implementation

This section is to explain the decisions made for the design of the project before showing the implementation stage of it.

4.1 – System Specification

As part of the design and development of the system. A specification is required to outline what will be needed to complete the design and implementation. The following chapters list the hardware and software needed.

4.1.1. Hardware

- Arduino-Uno
- Raspberry PI
- Ultrasonic Sensor X3
- L298N Driver
- Motors
- Expansion Board
- Wires
- Wheels
- Breadboard
- Plastic Chassis

4.1.2. Software

- Arduino IDE
- OpenCV
- Visual Studio
- YED Graph Editor

4.1.3. Design Decisions

The components chosen for the study were important in that they met a specification to be able to meet the requirements of the overall aimed result of the project. In this case, an obstacle avoiding robot.

The choice of hardware was the Elegoo Uno, this is an Arduino but more cost effective but has the same capabilities.

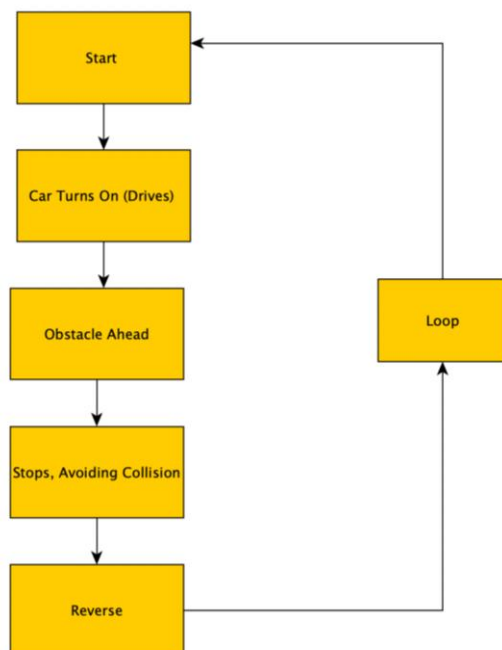
The first program that was chosen was the Arduino IDE, it is a lightweight IDE, with a simple interface which allows the code to be uploaded to the Arduino with quick success. It also has a simplistic debugger where if there is a problem in your code, it firstly highlights it and then provides a clear, yet concise sentence on what the problem. Yed Graph Editor was chosen as it is useful for designing flow diagrams when designing features in the project. OpenCV was the framework chosen as it is used in developing image processing applications and it is written in C++ which means that it explores a gap in the research, this was that there wasn't a vast amount of research which used C++ for producing AI applications.

4.2. 1st Feature

The 1st feature in the system is to be designed so the robotic system can recognise an object, thus, a collision would not occur. This is the first stage of developing the system to avoid obstacles. This will be expanded further in the following chapters.

4.2.1. Design

4.2.1.1. Algorithm Process



The algorithm was designed firstly in YED Graph Editor to show the simplistic process in which the algorithm was intended to function. This was developed further in the form of pseudocode.

The algorithm is written to make a decision in stopping the car before the object by using the ultrasonic sensor to detect objects in front of it, this simulates the autonomous vehicle as they also use sensors to avoid collision with either other cars, walls or other objects. A pseudocode-based algorithm was designed for the robot so it would be able to detect an object in front of it, and in doing so would then reverse backwards. This creates scope for testing as different surfaces could be tested against the sensor, such as a solid object in comparison to something lighter, such as paper. This is the first stage in the car being able to react to its surroundings.

The first component of the initial algorithm that was decided was driving the wheels, a L298N motor driver was selected as the motor driver for the motors on the build of the car, this allowed control of all the wheels.

```
#define motorFL 7  
#define motorFR 8  
#define motorRL 9  
#define motorRR 11
```

The figure above shows the defining of these motors, this makes it easier to utilise the motors in the algorithm, For example, the figure below shows how they'd be used.

```
...  
Void driveForward()  
{  
    pinMode(motorFrontLeft, HIGH);  
    pinMode(motorFrontRight, HIGH);  
    pinMode(motorRearLeft, LOW);  
    pinMode(motorRearRight, LOW);  
}
```

It was also decided that the algorithm would be split into functions to call during the code, this is a way to make the code readable and overall cleaner, this helps debug the code.

4.3. Implementation

The algorithm was when built, it was split into other functions, the functions were as follows:

1. driveForward(); - A function to drive the car forward
2. reverseCar(); - A function to reverse the car
3. check_Distance(); Using the ultrasonic sensor situated at the front of the car

//include photos of each function

Firstly, the algorithm specified a select selection of variables that were defined so they would be constants, these were to set the pins being utilised on the device. These set out the pins being used for the ultrasonic sensor, the motors on the L298N and also defining ENA and ENB, for controlling speed.

4.4. – Testing

4.4.1. Unit Testing

A figure is shown below showing the function which allowed the car to drive forward.

```
//function to drive forward  
void driveForward()  
{  
    analogWrite(ENA, SpeedControlSet);  
    analogWrite(ENB, SpeedControlSet);  
    digitalWrite(Motor1, HIGH); //left  
    digitalWrite(Motor2, LOW); //right  
    digitalWrite(Motor3, LOW); //left  
    digitalWrite(Motor4, HIGH); //right  
}
```

Figure 2 - Test

This function was a success.

The block of code which was tested was the function, this went through several iterations of testing as it wasn't understood as to why it wasn't working. But it was discovered that the equation that was being inputted was wrong. This was then moved into the main loop rather than in the form of a function. Which was then a success, this calculation to work out distance was formed using the datasheet of the ultrasonic [33] . This can be seen below the serial monitor to view the distance readings.

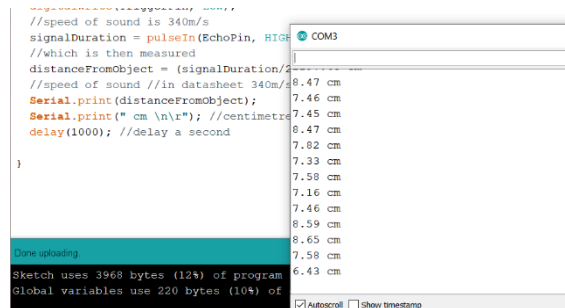


Figure 3 - Test

The last function was the function to drive the car backwards, the code is below and was also a success.

```

analogWrite(ENA, SpeedControlSet); //200 as in the #define
analogWrite(ENB, SpeedControlSet); //200 as in the #define
digitalWrite(Motor1, LOW);
digitalWrite(Motor2, HIGH);
digitalWrite(Motor3, HIGH);
digitalWrite(Motor4, LOW);
//testing
Serial.print("Reversing\n\r"); //print to check

```

Figure 4 - Test

4.4.2. System Testing

The system testing wasn't as successful, the car wouldn't react in the way that was planned with little understanding as to why this is.

4.4.3. Usability Testing

4.4.4. Acceptance Testing

4.5. 2nd Feature

4.5.1. Design

4.5.1.1. Final Robotic Process/Algorithm Process



In the current state of the robot, it operates on one Ultrasonic Sensor, this isn't at the sophistication level that was wanted as the original intended design was to have three ultrasonic sensors, so that when the car turned, whichever side it turned to, that sensor could then take readings for a range until it was read as clear to move.

Pseudocode for the Original Idea can be seen below. This is what the algorithm would look like in its earliest form.

```

Begin
Car Drives
Obstacle Detected
Turn Left
Right Ultrasonic Obtains Readings
If(clear)
{go} else {keep driving forward until
clear}

Once clear, Turn right //go around the
obstacle
Right Ultrasonic Obtains Readings
If(clear)
{go} else {keep driving forward until
clear}
  
```

4.5.2. Implementation

Implementation was not reached due to time constraints.

4.5.3. Testing

4.5.3.1. Unit Testing

Wanted to test the Ultrasonics' in this algorithm. Wanted to test the efficiency of them against different surfaces. Wanted to find the perfect stopping point through this.

4.5.3.2. System Testing

Wanted to test it as a whole system, for example, how well it would have handled the obstacles. Would have also done Acceptance testing.

Chapter 5 – Evaluation

Firstly, the literature review did prove to have aspects which proved right, such as the HC-SR04 is a good, cost effective ultrasonic sensor. It also aligned with the research that the L298N is a reliable easy to use motor controller. The objectives weren't met during the project, the first was achieved once the literature review was complete, but the second wasn't completed and the third is but isn't completed to the standard that was aimed for originally, as a full planned implementation wasn't carried out, which resulted in no testing thus, not having results of their implementation. Which resulted in them not being evaluated that to a good standard. Not meeting the objectives was due to a variety of things, such as sometimes poor management, sometimes due to an overload of personal commitments. Also due to not enough experience and familiarity, a lot was learnt with the microcontroller programming but not enough to produce the sophistication that was originally intended. The aim wasn't met, in the context of not having the objectives completed but was complete in the context of investigating the topic thoroughly. If it could be done again, the implementation would be planned in advance so there was a greater deal of time to implement, test, and evaluate thoroughly.

Chapter 6 – Conclusions

In conclusion, the project has been challenging, in the beginning, there were many things to learn but it's clear to conclude that there were vast amounts of depth in this topic. The research that was done in the literature review was interesting as the amount of different sub-fields that were able to be read about. From this, the conclusions that were drawn were that; Asimov's three laws of robotics aren't realistic as by following one law, you can sometimes disobey another. The other conclusion is that Python is at the forefront of AI, in particular, when writing AI applications, other languages are still used widely but Python

seems the one that is utilised the most in these applications. It was also discovered that there are different types of autonomy and the world isn't at the full autonomous level. It was also concluded that there are many microcontrollers and with the right set of knowledge, any can be used to build something that is sophisticated and brilliant. A gap in the literature review was discovered while researching, it was concluded that there wasn't a lot of research which explored writing applications using C++. This is something that could be explored further. The first objective was met during the research, an investigation spanning a few months was performed so the topic could be researched thoroughly. The second objective wasn't met, this was disappointing as the topic was interesting and engaging but the student couldn't completely build and implement an autonomous vehicle, they tried but a number of factors limited them. They designed them to the best of his ability but could not implement. This leaves room for further work to be done. The third objective was met to a certain extent, it was evaluated but there wasn't a rich depth of testing performed due to the lack of feature implementations, like they planned so this left the objective not fully complete.

6.1. Recommendations for Further Work

There are many avenues that could still be explored in this field as it has such a wide area for different applications and technologies to be researched and applied. The first that would want to be further worked upon, is that the student would like to have a more sophisticated system, whether this would be with an attempted reinforcement learning algorithm, this could be done in the form of a robot, through simulation or a real time environment. This would then be trained to be autonomous to be able to react to their surroundings. This would allow a higher level of autonomy. Another recommendation for further work is using a neural network, this is a supervised learning algorithm where the NN would be provided with a dataset and it would then learn to adapt to the surroundings, these can be written and applied using the Raspberry Pi. The other recommended piece of further work is using the Raspberry Pi to apply computer vision so it could be trained to recognise things such as traffic signs, road lanes, parking bay lanes to simulate an autonomous vehicle.

Chapter 7 – Reflection

This reflection is being produced in the expectation of allowing the student to improve their overall performance in future projects, whether academic or personal. This project has been a challenging task for the student. It was daunting to begin with as it was a topic in which they had no familiarity with so has been a huge learning curve. The first area of improvement for the project would be preparation. Originally, the student thought they were prepared but they truthfully underestimated how long or the depth the research would go into. Which lead to valuable time used to research and write up the first two chapters of the report. The second thing that has made it more difficult is the unfamiliarity that the student had with electronics (components, code, simulations, etc.) This also lead, to delays due to struggling to grasp certain aspects of this, one of the main aspects was the number of bytes that was available. The student was familiar with using gigabytes of storage and ram but within the microcontrollers there were a very small level of concise bytes that if used carefully, can have immense power to do a wide variety of things, even with such limited power. The unfamiliarity also meant that the debugging was difficult to complete due to the new complexity. The student also, at times, resisted to take advice from their supervisor, an example of this was when advised to purchase a Robo-Vacuum, they felt they would have done wrong, so it made them resistant to do it. There are many things that happened that they felt they could have been improved, they hoped their objectives would have been completed, they needed to manage their time by following the Gantt chart closer, by setting those deadlines for themselves and sticking to them they wouldn't have gotten themselves into the situation that they did. They also could have split the objectives up into smaller sections so they could have achieved them with more ease. The other thing they could have done better is practice thoroughly with programming microcontrollers so it would have been easier to understand when they went to use them. The last thing they would improve on is listening to advice more efficiently as they were often forgetful of useful advice, such as referring to datasheets when using electronic components.

References and Bibliography

- [1] N/A, "35 Statistics about Self-Driving Vehicles Market in 2021," N/A, 2nd February 2021. [Online]. Available: <https://research.aimultiple.com/self-driving-cars-stats/>. [Accessed April 2021].

- [2] N/A, "Arduino Projects With DIY Instructions, Codes, Diagrams, 3D Models," How To Mechatronics, N/A N/A N/A. [Online]. Available: <https://howtomechatronics.com/arduino-projects/>. [Accessed 13 October 2020].
- [3] M. Hasan, "Top 20 Best Raspberry Pi Projects That You Can Start Right Now," Ubuntupit, 21 July 2019. [Online]. Available: <https://www.ubuntupit.com/20-best-raspberry-pi-projects-that-you-can-start-right-now/>. [Accessed 11 October 2020].
- [4] S. C. Shapiro, "Artificial Intelligence," *Artificial Intelligence*, vol. N/A, no. N/A, p. 15, N/A.
- [5] M. Kubat, "Introduction," in *An Introduction to Machine Learning*, Miami, Springer, 2017, pp. 10-11.
- [6] F. M. Mordechai Ben-Ari, "Chapter 1," in *Elements of Robotics*, N/A, SpringerOpen, 2020, p. 311.
- [7] F. L. Ronald Leenes, "Laws on Robots, Laws by Robots, Laws in Robots: Regulating Robot Behaviour by Design," *Law, Innovation Technology*, vol. N/A, no. N/A, p. 33, 2014.
- [8] G. Mester, "Applications of Mobile Robots," in *ResearchGate*, Budapest, 2006.
- [9] U. Nehmzow, "Mobile Robotics: Research, Applications and Challenges," Department of Computer Science, Manchester, N/A.
- [10] McKinsey, "Industrial Robotics: Insights into the sector's future growth dynamics," McKinsey & Company, N/A, 2019.
- [11] B. G. Buchanan, "A (Very) Brief History of Artificial Intelligence," *AI Magazine*, vol. 26, no. 4, p. 8, 2006.
- [12] A. Turing, "COMPUTING MACHINERY AND INTELLIGENCE," N/A, vol. N/A, no. N/A, p. 22, 1950.
- [13] R. French, "The Turing Test: The first 50 years," in *Trends in Cognitive Sciences*, Belgium, 2000.
- [14] L. Gugerty, "Newell and Simon's Logic Theorist : Historical Background and Impact on Cognitive Modelling," in *Human Factors and Ergonomics Society Annual Meeting Proceedings*, Clemson, 2006.

- [15] M. M. Mijwel, "History of Artificial Intelligence," in *Baghdad College of Economics Sciences University*, Baghdad, 2015.
- [16] I. Z. e. al, "Dissecting Robotics - Historical Overview and Future Perspectives," *Acutronic Robotics*, vol. N/A, no. N/A, p. N/A, 2017.
- [17] F. Dedov, "What are Algorithms?," in *The Bible of Algorithms and Data Structures : A Complex Subject Simply Explained*, USA, Amazon Digital Services LLC, 2020, p. 12.
- [18] P. S. e. al, "Introduction to Machine Learning and Its Basic Application in Python," *SSRN Electronic Journal*, vol. N.A, no. N/A, p. 22, 2019.
- [19] A. Dey, "Machine Learning Algorithms : A Review," *International Journal of Computer Science and Information Technologies* , vol. 7, no. 3, pp. 1174 - 1179, 2016.
- [20] D. Zarya, "The Best Programming Language for AI: Your Ultimate Guide," Friendly Technologies Zfort Group, 12 June 2020. [Online]. Available: <https://www.zfort.com/blog/best-programming-language-for-ai>. [Accessed 30 December 2020].
- [21] K. R. Srinath, "Python - The Fastest Growing Programming Languages," *International Research Journal of Engineering and Technology*, vol. 4, no. 12, p. 4, 2017.
- [22] "AI Programming : 5 Most Popular AI Programming Languages," Existek, 7 February 2018. [Online]. Available: <https://existek.com/blog/ai-programming-and-ai-programming-languages/>. [Accessed 2 January 2021].
- [23] D. Vernon, "An Introduction to Computer Vision," in *Machine Vision*, Hertfordshire, Prentice Hall International (UK) Ltd, 1991, pp. 9-10.
- [24] Y. Wu, *An Introduction to Computer Vision*, Evanston: Northwestern University, N/A, pp. 3-4.
- [25] V. P. Kharkar, "A Road Sign Detection and Recognition Robot using Raspberry-Pi," *International Research Journal of Engineering and Technology*, vol. 5, no. 9, p. 5, 2018.
- [26] J. O. e. al, "How do Autonomous Cars Work?," *Transportation Research Procedia*, vol. 44, no. N/A, p. 7, 2020.

- [27] e. a. Qi Zhang, "Self-driving scale car trained by Deep reinforcement Learning," Qi Zhang, N.A, 2019.
- [28] B. e. al, "Self Driving Car using Raspberry pi," *International Journal of Advanced Science and Technology*, vol. 29, no. 5, p. 8, 2020.
- [29] D. K. S. Rohit Tiwari, "Vehicle Control Using Raspberrypi and Image Processing," *Innovative Systems Design and Engineering*, vol. 8, no. 2, p. 5, 2017.
- [30] N. B. M. G. P Divya, "Arduino Based Obstacle Detecting System," in *International Conference of Advance Research and Innovation*, Vijayawada, 2020.
- [31] H. Technology, "L298N Motor Driver," Handson Technology, N/A, N/A.
- [32] V. P. Amand Nayyar, "Raspberry Pi- A Small Powerful, Cost Effective and Efficient Form Factor Computer: A Review," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 5, no. 12, p. 19, 2015.
- [33] A. H. e. al, "Programming a Microcontroller," *International Journal of Computer Applications*, vol. 155, no. 1, p. 6, 2016.
- [34] P. N. Stuart Russel, *Artificial Intelligence : A Modern Approach*, New Jersey: Pearson, 1995.
- [35] M. D. B. e. al, "Arduino Based Speed Control of DC Motor by," *International Journal of Innovative Research in Science,,* vol. 6, no. 5, p. 4, 2017.
- [36] H. R. Hatem, "Control of Pulse Width Modulation ON Direction and Speed of DC Motor using Arduino," *Journal of Engineering and Applied Science*, vol. 5, no. N/A, p. 6, 2019.
- [37] J. Dudovskiy, "Positivism Research Methodology," *Business Research Methodology*, N/A N/A N/A. [Online]. Available: <https://research-methodology.net/research-philosophy/positivism/>. [Accessed 4 April 2021].
- [38] Microchip, "PIC18F45K50 DataSheet," Microchip, N/A, N/A.
- [39] e. a. Zahid Nawaz, "Simplified FDD Process Model," *Modern Education and Computer Science Press*, vol. N/A, no. N/A, p. 8, 2017.

Appendices

Code Listing for Feature 1

//by defining we set these values

#define EchoPin A4

#define TriggerPin A5

#define Motor1 7

#define Motor2 8

#define Motor3 9

#define Motor4 11

#define ENA 6 //one side of L298N

#define ENB 5 //other side of L298N

#define SpeedControlSet 150

//for ultrasonic

//floating points for accuracy

float distanceFromObject;

float signalDuration;

//function to drive forward

void driveForward()

{

 analogWrite(ENA, SpeedControlSet); //enable the motors to rotate at a set

 analogWrite(ENB, SpeedControlSet);

 digitalWrite(Motor1, HIGH);

 digitalWrite(Motor2, LOW);

```
digitalWrite(Motor3, LOW);  
digitalWrite(Motor4, HIGH);  
}  
  
//function to go backwards  
void reverseCar()  
{  
    analogWrite(ENA, SpeedControlSet); //200 as in the #define  
    analogWrite(ENB, SpeedControlSet); //200 as in the #define  
    digitalWrite(Motor1, LOW);  
    digitalWrite(Motor2, HIGH);  
    digitalWrite(Motor3, HIGH);  
    digitalWrite(Motor4, LOW);  
    //testing  
    Serial.print("Reversing\n\r"); //print to check  
}  
  
void stopCar() //uses the ENA pins from the L298N to control the speed and to stop it  
//by setting output to low  
{  
    digitalWrite(Motor1, LOW);  
    digitalWrite(Motor2, LOW);  
    digitalWrite(Motor3, LOW);  
    digitalWrite(Motor4, LOW);  
    delay(500); //delays are in milliseconds  
    //testing purposes  
    Serial.print("Car is Stopping...\n\r");  
}
```

```
void setup() {  
  
  Serial.begin(9600);  
  
  pinMode(EchoPin, INPUT); // reciever, echos the pulse, bounce back through here  
  
  pinMode(TriggerPin, OUTPUT); //transmits the pulse at 40khz, 8 pulses  
  
  //set the motors to power on  
  
  pinMode(ENA, OUTPUT);  
  
  pinMode(ENB, OUTPUT);  
  
  pinMode(Motor1, OUTPUT);  
  
  pinMode(Motor2, OUTPUT);  
  
  pinMode(Motor3, OUTPUT);  
  
  pinMode(Motor4, OUTPUT);  
  
  //stop(); sort out a stop function  
  
  Serial.println("Motors on!\n\r");  
  
}
```

```
void loop()  
  
{  
  
  //driveForward  
  
  digitalWrite(TriggerPin, LOW); //start the pulse  
  
  delayMicroseconds(2); //delay as microseconds  
  
  digitalWrite(TriggerPin, HIGH); //send pulse  
  
  delayMicroseconds(10);  
  
  digitalWrite(TriggerPin, LOW);  
  
  signalDuration = pulseIn(EchoPin, HIGH); //measures pulse duration
```

```
//use speed of sound, ultrasonic datasheet

distanceFromObject = (signalDuration/2)*0.034;

//get results

Serial.print(distanceFromObject);

//if(distanceFromObject < 15)

//{

//stopcar()

//}

//else

//{

//reverseCar();

//}

}

reverseCar();

}
```

