AFFORDABLE AUTONOMOUS VEHICLES FOR DEPLOYMENT AFTER DISASTROUS EVENTS

by

SHANNON LAMEAGE ABOLMAALI

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON
May 2022

Acknowledgements

"The day I chose to take this journey, I knew there would be a lot of ups and downs, roads bumps and dips in the road. There was one thing that I knew for certain that I would be stronger and that patience is key."

First and foremost, I would like to give my sincere gratitude and appreciation to my advisor and mentor, Dr. Erick C. Jones, Sr. You always motivated me and made me enthusiastic about this research even the days that were the most difficult. Thank you for your mentorship and always guiding me through the right path. When I first started my research, I knew it would be challenging, but the late night, weekend phone calls and zoom meetings always gave me confidence and a feeling of relief. Thank you very much for believing in me.

To my committee members, Dr. Emma Yang, Dr. Shouyi Wang, and Dr. Edmund Prater, thank you for serving on my committee and providing me with the feedback to make this research all the more better. I am forever grateful for your guidance and advice.

To Dr. Jones Jr., I would like to express my gratitude for all your support and motivating me during this journey. Thank you very much!

To my parents, who have put up with me, and all the ups and downs that I faced during this process. Sometimes, I question myself on how they put up with me on some days as my emotions ran high. I am grateful for their unconditional love, especially when I would come home late from the lab as I was always taken care of. Thank you very much for all that you have

done for me all my life. I would not have been where I am today if it was not for you two. Love you very much Dad and Mom!

To my other family members both in the United States and out of country, I really appreciate all the support and encouragement. I am so grateful for all the video and phone calls considering the time difference.

To my friends, thank you so much for always being there for me and checking in on me to make sure I was doing alright. I will never forget the days that I did not feel like going out, but if it was not for you asking me to go out, I would have really gone crazy.

I would like to dedicate my dissertation to my professor and mentor, Dr. Erick C. Jones, Sr.

May 15, 2022

Abstract

AFFORDABLE AUTONOMOUS VEHICLES
FOR DEPLOYMENT AFTER
DISASTROUS EVENTS

Shannon Lameage Abolmaali, Ph.D.
The University of Texas at Arlington, 2022

Supervising Professor: Erick C. Jones, Sr.

In disastrous events such as hurricanes and tornadoes, it has been observed that people get stranded and helpless without a feasible way to escape during those emergency situations. This became very evident during hurricanes, such as Katrina and Ida affecting millions of people seeking immediate rescue efforts. With the use of artificial intelligence and machine learning, we envision an autonomous vehicle, AV, which is able to find the most optimal and safest way to help those who are stranded to get them to a safe location. Electric vehicles, EV, and Autonomous Vehicles, AV, is becoming the future; minimizing the carbon footprint, reducing accidents, and revolutionizing the car industry with new car production. The challenge for the next generation is the affordability of these new vehicles and their intelligence. Technology has been advancing making these vehicles autonomous from a level 0 (no automation) to level 5 (full driving automation), but cost is a significant factor making these vehicles very expensive.

This research seeks to leverage affordable sensors, artificial intelligence, and machine learning to develop a smart car kit that can be retrofitted to any type of vehicle to make it smart. The novelty of this research is that we are utilizing less expensive technologies, such as, cameras,

proximity sensors, and RFID technologies and also that this kit can be used to make older vehicle models smart. The future of work components of this research is the autonomous learning to minimize human risk in disaster recovery.

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Autonomous Vehicle

Electric Vehicle, EV, and Autonomous Vehicles, AV, is becoming the future of the automobile industry; minimizing the carbon footprint, reducing accidents, and revolutionizing the car industry with new car production. Researchers anticipate that the Level 4 autonomous vehicles, the minimum requirement in order for a vehicle to be autonomous is to be adopted ranging from 25% to 87% by 2045 (Bansal and Kockelman 2017). Other projections indicate that autonomous vehicles will account for 20-40% of total vehicle fleet by 2040s (Litman 2019). The main reason for slower rate of adoption for autonomous vehicles has been considered the relatively high cost (Litman 2020). Transportation has surpassed electricity generation as the top greenhouse gas emitter in the US Transportation which accounts for nearly 28.5% of US emissions. (Bloomberg New Energy Finance, 2018). Autonomous vehicles could reduce greenhouse gas emissions by driving more efficiently, avoiding traffic congestion, accelerating adoption of alternative fuel vehicles, and charging in alignment with renewable electricity generation.

The National Highway Traffic Administration defines five levels for vehicle autonomy, Level 0 (no automation) to Level 5 (full automation). According to the National Highway Traffic Administration, car accidents occur every minute, which is approximately 5 million accidents annually in the United States, making human error the main cause of the automobile accidents. With an increased number of autonomous vehicles on the road, human error can be mitigated. Also, with the use of AVs, injuries, damage cost associated with vehicle, and fuel consumption and congestion will decrease in large-scale traffic (Piao et al., 2016). AVs include several benefits, including safety, efficiency, and increased mobility. AVs have the ability t

o communicate with each other, optimizing routes persistently as well as helping improve the traffic congestion (Andersen et. al 2014). Autonomous vehicles are beneficial to help in dangerous and congested areas where there is a threat to people during disastrous situations. The use of unmanned vehicles also has become popular in other situations as well, such as traffic control and disaster management.

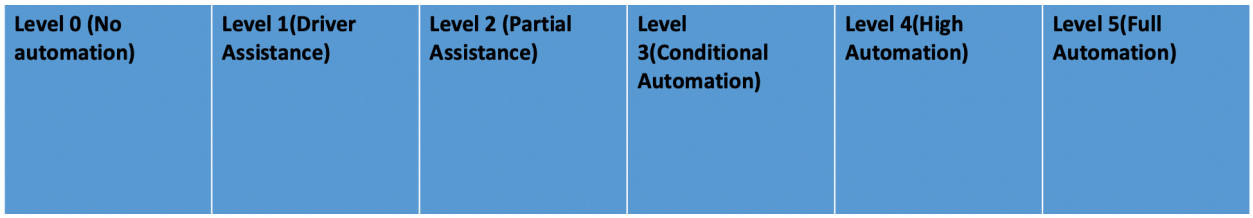


Figure 1-1: Level of Automation

Figure 1-1 outlines the levels of automation, Level 0 being no automation to Level 1 with driver assistance. From Level 1 to Level 2, we have partial assistance which gives the vehicle assistance functions such as: steering and acceleration. Next, we have Level 3: conditional automation in which the vehicle has the capabilities to monitor its surroundings in real-time, permitting the driver to take control only when alerted by the system. Level 4 and 5 are high automation and full automation; respectively. To reach these levels, there is still much advancement required.

Recently, in the United States, there have been several hurricanes which have hit and affected several areas. In disastrous events such as hurricanes and tornadoes, it has been observed that people get stranded and helpless without a feasible way to escape during those emergency situations. This became very evident during hurricanes, such as Katrina and Ida affecting millions of people seeking immediate rescue efforts. Also, since 2015, the southeastern United States has been affected with several hurricanes, such as Joaquin in 2015, and Irma in 2017; just to name a

few. Although technology has been evolving for the autonomous vehicles and is becoming the future, there is limited information about the use of these vehicles during disastrous events.

1.2 Problem Statement

With the use of artificial intelligence and machine learning, we envision an autonomous vehicle, AV, which is able to find the most optimal and safest way to help those who are stranded to get them to a safe location. The challenge for the next generation is the affordability of these new vehicles and their intelligence. Technology has been advancing making these vehicles autonomous but cost is a significant factor making these vehicles very expensive. Since cost is a major factor in this, it results in people being more reluctant in purchasing an autonomous vehicle. Newer technology usually brings a sense of being more reluctant to people who are resistant to change; however, even in such scenarios, when people are in emergency situations and need immediate help, an autonomous vehicle which will save the people stranded will be of significant help. A study by Wagner et al., 2018, states that human-machine interaction requires substantial time and effort to be able to gain full trust in autonomous vehicles. Our research seeks to leverage affordable sensors, artificial intelligence, and machine learning to develop a smart car kit that can be retrofitted to any type of vehicle to make it smart. The future of work components of this research is the autonomous learning to minimize human risk in disaster recovery.

1.3 Research Objectives

The question that is to be addressed in this study is if a low-cost AI-based sensor be retrofitted to make an affordable car self-driving cost effective. Our long-term research goal is to investigate AI driven low-cost sensors that can facilitate automated systems effectively as well as being affordable. The research objective is to provide an effective demonstration of a low-cost AI-based sensor that can enable an affordable car system to become smart and ultimately transition into a self-driving car, or autonomous vehicle.

1.4 Purpose of this Research

In today's society, autonomous vehicle technology has been advancing, and as a result, people are more inclined in purchasing newer technology; however, with a much more affordable option, they would be able to have that same technology with their current vehicle. The purpose of this research is to not only provide people with the opportunity to get acquainted with newer technology for their current vehicles, but also to feel more comfortable about this change, as that is where people resist the most.

1.5 Organization of this Dissertation

This dissertation follows a five-chapter format of introduction, background, methodology, results, and conclusion. Chapter 2 discusses the literature review of disaster relief and evacuation, autonomous vehicles in disastrous events, path planning for autonomous vehicles, challenges for autonomous vehicles, etc. Chapter 3 defines the methodology of this research, including the specific objectives, hypothesis, and research methodology. This chapter also includes test setup and details on the location where the tests were performed. Chapter 4 discusses the results of the experiments performed during this research. Interpretations and analysis of the results are also included in this chapter. Chapter 5 is the conclusion and discussion of the research with a summary of the overall study, findings and recommendations, and a contribution to the body of knowledge. It also includes the limitations and future steps in the research.

2 Background

2.1. Evacuation and Disaster Relief

During disastrous situations, the main goal is to save those who are stranded to a safe location prior to the disaster as well as get the necessities for post-disaster. With the increasing hurricanes and the frequency of these events, large-scale evacuations are also expected to increase in the future (Broccoli and Manabe, 1990). For large-scale evacuations, rescue efforts should be optimized in order to meet the needs of the individuals seeking immediate help. In previous cases of hurricane evacuations, state department officials and professionals have worked together to initiate new strategies, some of which includes improving supply through intelligent transportation systems, or ITS (Urbina and Wolshon, 2003). Some states have been experiencing more frequent cases of disasters, an urgency for strategic and more effective evacuation plans are required (Wong et al., 2018). According to Murray-Truite and Wolshon, 2013, transportation and emergency management agencies account for an average of 6-10% of its population to be classified as requiring assistance when needed to evacuate for hurricanes, in the United States (Murray-Truite and Wolshon, 2013). A study was done focusing on South Carolina, indicating that approximately 5% of the population will need evacuation assistance with category 5 storm (South Carolina Emergency Management Division, 2019).

2.2 Autonomous vehicles in disastrous events

Autonomous vehicles in disastrous and emergency situations will help people in various scenarios receive immediate rescue efforts. The knowledge and understanding about the most affected disaster-hit areas through images provided by the unmanned vehicle will help emergency evacuations by finding the safest and most optimal route to reach where is necessary. In order to find the most optimal route for the AVs, it is crucial to utilize the real-time images from 360-degree views to help assess the damage which has already been caused by the

hurricanes, tornadoes, etc. to be able to also assist the emergency response teams. (Ullah et al 2018).

In regards to autonomous vehicles in an emergency setting, limited research has been done. The studies which have been performed mainly focuses on the operation rather than the aspect and acceptance of society. According to Murray-Tuite et al., 2017, the use of AVs to aid those who had limited access disaster situations was introduced in 2017. In cases when evacuation is necessary, the autonomous vehicles demonstrate less delay period (Chang and Edara, 2018). A study by Yin et al. 2018 demonstrated optimization of routes during evacuations to reduce the traffic congestion. During post-disaster, these autonomous vehicles are required to provide assistance with the evaluating and assessing the damage, vehicle routing, delivery of medical supplies and other required supplies.

2.3 Path planning for autonomous vehicles

Path planning is utilized to avoid any obstacles along the way. The purpose of path planning is to determine the best route to be followed and to also be able to travel from point A to point B without risks of collision with objects along the way (Ma et al. 2018). The algorithms used for path planning in autonomous vehicles include the following: conventional, cell-based, model-based, and learning-based algorithms. The rapid exploring random trees are branch and bound algorithms which find a path from the source to the goal, attempting to connect the source to the goal by a suitable path. For this type of algorithm, the limitation is that it requires better approximations of the path.

As stated previously, path planning is used to find a route from initial point to the destination in order to find the unmanned vehicle. The path should be clear of any object destruction. Latest research has focused on AVs during disasters (Wu et al. 2018; Shiri et al. 2019). With autonomous

vehicles becoming popular, the actions required to take for disastrous situations is important for creating solutions to improve the path planning during these times (Qadir et al., 2021). In this research, we plan to leverage the simplicity of post-disaster roads, AI, and ML to solve simple path planning problems with low-cost sensors. Tables 2-1, 2-2, 2-3, represent the different types of algorithms that were taken into consideration during the preliminary stages of this research, along with the advantages and disadvantages of each type of algorithms. Evaluating these algorithms gave us the opportunity to be able to narrow down the "best fit" algorithm for conducting and to meet the goals of the research study.

Table 2-1: Conventional Algorithms

Path Planning Algorithms	Algorithm type	Method	Advantages	Disadvantages and Limitations	Reference
Rapidly exploring random tree(RRT)	Conventional	Develops a tree of state distribution representing feasible paths	--Probabilistically robust path planner; --Able to avoid uncertain dynamic obstacles	Need for better approximations of path	Gong et al., 2019 Yang et al., 2018 Zhang et al., 2020
Artificial Potential Field Algorithm	Conventional	Using distance unit to avoid collisions and reach the target and builds a gravitational field around the destination and repulsive field around the obstacles to look for safe path to the target	--For path planning: Looks out for obstacles around and for safe path to the destination. --Avoids collisions and reaches targeted location.	N/A	Aggarwal and Kumar, 2019 Shima et al., 2009
Voronoi	Conventional	Offers effective area coverage control in a multi-environment.	Solves the problem of area coverage control for multiple AV	N/A	Zhang et al., 2020 Aggarwal and Kumar, 2019

Table 2-2: Cell-based Algorithms

Path Planning Algorithms	Algorithm type	Method	Advantages	Disadvantages and Limitations	Reference
A-star	Cell-based	--Look for the shortest path between source and target to avoid a dead end.	Less costly path	The algorithm may not reach a global optimum solution	Gong et al., 2019 Yang et al., 2018 Zhang et al., 2020
D-star	Cell-based	The informed incremental search algorithm	--Analyzes and determines the path by measuring the environment. --Analyzes the obstacles present and creates a map of the environment	N/A	Aggarwal and Kumar, 2019 Shima et al., 2009
Dijkstra	Cell-based	--Determines the shortest path --Used for path optimization and obstacle avoidance	Successful collision avoidance and route optimization demonstrated	Takes more time as compared to the A-Star Method	Zhang et al., 2020 Aggarwal and Kumar, 2019 Ullah et al., 2020

Table 2-3: Model-based and Learning-based Algorithms

Path Planning Algorithms	Algorithm type	Method	Advantages	Disadvantages and Limitations	Reference
Mixed Integer Linear Programming	Model-based	Optimizes an objective function where at-least one of the variables must be an integer	Successful collision avoidance and route optimization.	Single AV in experiments.	Gong et al., 2019 Yang et al., 2018 Zhang et al., 2020
Neural network	Learning-based	Use of deep learning model to enable an AV to understand its external environment Determine the UAV state at the next node using a neural network	--Provides complete path from any arbitrary initial state to final state while avoiding obstacles --A small neural network with faster speed than traditional path planning methods	Path planning of single AV rather than multiple ones.	Ullah et al., 2020 Shima et al., 2009 Aggarwal and Kumar, 2019

Tables 2-1, 2-2, and 2-3, conventional, cell-based, model-based and learning algorithms; respectively, as shown above represent the different path planning algorithms which were studied for this research. Each algorithm type, method, advantages, and disadvantages have been outlined meticulously in the tables.

2.4 Obstacle Avoidance

As mentioned before, autonomous vehicles have become more and more popular during the past several years. Ultimately, autonomous vehicles will replace human drivers reducing human error and accidents on the road, as well as improving road efficiency and driving safety. With the combined advancing technologies of sensor, electronic control, and artificial intelligence, driverless vehicles are becoming more intelligent while making quick and efficient decisions. Obstacle avoidance is one of the main functions of decision-making, path planning, and path tracking. Studies on obstacle avoidance for autonomous vehicles are important for application purposes. Path planning has been used to avoid obstacles along with keeping the road safe. Methods and algorithm studies and research has been performed in regards to obstacle avoidance. Algorithms related to this research area includes, artificial potential field method, rapidly exploring random tree method (RRT), fuzzy logic, and neural networks. Path planning algorithms for autonomous vehicles should meet specific requirements, which include path safety, computation speed, and system robustness, simultaneously. A simple algorithm, the artificial potential field algorithm can satisfy the requirement of real time control.

The advantage of using this algorithm is the ability for path planning for live obstacle avoidance and due to this; the algorithm has been used in practical projects (Fu et al. 2015). Not only have studies been done to improve the performance of obstacle avoidance path planning algorithms, but also researchers have been working towards the risk change in the obstacle avoidance process (Llorca et al. 2011; Stephanie et al. 2014). According to the National Highway Traffic Safety Administration, about 30% of accidents occur during lane changes (National Highway Traffic Safety Administration 2019).

2.5 Object Detection

Object detection is a technique in computer vision which identifies and locates objects within an image and/or video. It is able to draw a boundary around each of the objects so that it locates where the objects are in a specified region. Object detection algorithm for autonomous vehicles should satisfy two conditions: high detection accuracy of road conditions and objects and a real-time detection speed to determine whether the detector is possible to be used in driving or not. Deep learning models and specifically Object detection algorithms consists of two categories, one-stage and two-stage algorithms. Examples of one-stage algorithms include SSD and YOLO, which perform regression and classification in one stage. These algorithms have a generally low accuracy, but high detection speed. On the contrary, the two-stage algorithms have a high accuracy but slow detection speed such as R-CNN and Fast R-CNN. The YOLO algorithm detects objects by splitting the image into several grid cells. By doing so, it detects multiple objects through a single stage and the speed is relatively faster. The downside of this type of algorithm is that it is difficult to detect small objects.

The majority of object detection algorithms are able to detect large objects, but smaller objects are more difficult and often neglected by the detector. In cases of autonomous driving, it is dangerous to miss pedestrians, traffic lights and signs.

2.6 Object Detection algorithms

Several object detection algorithms were studied throughout this research in order to choose the best algorithm which fits the scope of the research. The algorithms studied in detail for this research are as follows: Fast R-CNN, Faster R-CNN, Histogram of Oriented Gradients (HOG), Region-based Convolutional Neural Networks (R-CNN), Region-based Fully Convolutional Network(R-FCN), Single Shot Detector (SSD), Spatial Pyramid Pooling (SPP-net) and You Only

Look Once (YOLO). The accuracy and precision of each of these algorithms vary according to several studies which have been performed. For example, according to one study by Du et al. 2020, the mean average precision for the Faster R-CNN and R-FCN are 93.63% and 94.20%; respectively. It is important to note that the higher these percentages are, the better the performance is for object detection purposes.

The Region-based Fully Convolutional Network, or R-FCN, an algorithm, by Dai et al. 2016 is based on the structure of the Faster R-CNN, maximizing the parameters in order to increase speed. Similarly, the Single Shot Detector, or SSD, was proposed by Liu et al. 2015 performs both small-scale object detection and positioning in one shot.

For the purposes of this study, the You Only Look Once (YOLO) algorithm was chosen as it is used for object detection and specifically self-driving vehicles.

2.7 Challenges in autonomous vehicles path planning

Challenges arise in path planning for unmanned vehicles. It becomes extremely challenging when there are scenarios of multiple autonomous vehicles due to several constraints, high-dimensionality, and the timing in which the vehicle needs to have a response time; especially in times of disastrous situations, time is a big factor in order to be able to rescue those who are stranded. In order to reduce and/or avoid crashes, scientists have been working to develop resolutions for enhancing path planning algorithms. Research and studies have indicated that in 2D environments, the algorithms have grown exponentially; however, in 3D environments, the algorithms do not provide adequate navigation due to various factors such as, physical, geometric, and time-related factors (Yang et al. 2017; Yang et al. 2018). Several challenges come with path planning. One common challenge, evidenced by recent research studies is the optimization of the system achieved by path planning. Challenges were outlined in path planning for AV in regards to

the optimal path length, which must cover from the start to the destination (Aggarwal and Kumar, 2019). The main barriers in path planning include optimizing the system, determining the optimum path length, path completeness, optimality, efficiency, energy consumption, and achieving robustness. In this research project, we will be starting with using simpler situation, with not as many vehicles and then based on accuracy and precision in our study, we will add situations that are more complex.

2.8 Sensors in autonomous vehicles and its technology

Recently, technology in autonomous vehicles and the sensors used has grown exponentially. The main goal of an AV is to be aware of its surroundings and make decisions quickly and effectively, requiring a precise and accurate understanding of the location of the vehicle to then be transferred into an algorithm. Currently, two main types of sensors are used for an AV, an exteroceptive sensor which is used for observing the environment and providing the distance to objects in addition to proprioceptive sensors which measure values from within the system.

Ultrasonic, or proximity sensors are used to detect when a vehicle is close to a person or object. This type of sensor has the capability to identify how far the object is from the vehicle, providing a signal to the driver within the vicinity. Another type of sensor that has been used in the autonomous vehicle is the LiDAR, or Light Detection and Ranging. The system consists of a 64-line three-dimensional radar system that sends a detection signal to a target, and then compares the received signal reflected from the target with the transmitted signal (Han et al 2018). The system is installed at the top center of the vehicle and is able to detect environmental information through high-speed rotational scanning of 360 degrees. LiDAR is a type of sensor technology that measures distance with high precision and accuracy. This is beneficial for autonomous vehicles as it gives accurate mapping and localization with 360-degree views. Advantages of a LiDAR consists of the

following: detecting and tracking obstacle detection, measuring speed, navigating and positioning of the vehicle (Han et al 2018).

Research by Azim et al. 2012 determined a method to distinguish between various moving obstacles. LiDAR is complex and with traffic environments changing constantly, color cameras and LiDAR have been utilized to detect the target and/or object on the autonomous so that sufficient information is taken. According to a study Asvadi et al. 2017, a convolutional neural network method is used for the obstacle information based on three detectors designed by combining the dense depth map and dense reflection map output from the 3D LiDAR and the color images output from the camera. The image information of color camera will be affected by the ambient light, and LiDAR cannot give full play to its advantages in foggy and hot weather. Therefore, the performance and recognition accuracy of the single sensor is low in the complex urban traffic environment, which cannot meet the security needs of autonomous vehicles (Han et al 2018).

Most autonomous vehicles today use cameras as it is capable of detecting static and dynamic objects in their surroundings. Also, cameras are able to determine distances to a particular object, even though this requires complex processing algorithms (Xiaomong et al. 2010). Cameras are preferable compared to other sensors because of their capability to see colors and textures. The benefit in this is that it increases the perception system of the AV as it allows the vehicle to identify road signs, traffic lights, etc., as well as the affordability and availability.

These sensors and technologies discussed have been used for new vehicle models. Cost becomes a challenge with autonomous vehicles and its technology. The goal of this research is to use these new technologies with artificial intelligence and machine learning to implement and integrate into older vehicle models to make it smart.

2.9 Implementing RFID in vehicles

Accidents occur due to carelessness and improvised technology in certain vehicles. This can be improved by modern technologies such as RFID. This system helps in communication of one vehicle to another, thus reducing the impact of road accidents. The tags readily identifies and receives the data and shares the data among the vehicles on road. For instance, two cars on road, they reach a range where they are able to communicate through the RFID tags in the form of visual data. RFID is also able to indicate the driver through text messages about the fuel level, low tire pressure, battery life. RFID can be deployed in some parts of the road to help in reducing headlight glares, with the corresponding information the headlight of the approaching car can be controlled. This system alerts the driver about congested roads, accidents, traffic and chooses an alternative way.

The device tracks another car by the distance and the angle at which the signal is transmitted. The speed of the car is tracked by the voltage in the speedometer that is applied to an electromagnet to produce electromagnetic field. Installing this system in emergency vehicles like ambulance, police cars, fire engines will help to control the traffic. The traffic lights would work according to the approaching emergency vehicle so that it can pass through the traffic without any hindrance. We can also reduce the number of accidents due to rash driving by deploying some of these systems on the roads and whenever a car exceeds the speed limit by tracking the RFID tag in the car the

fine is automatically given to the driver. With every application, limitations follow as well. In this case, the device does not have the capability to read multiple tags at once in a small region. As a result, privacy of the user is affected since any device in that range can read the RFID tag without their knowledge leading to security and ethical problems.

2.10 Installation

An RFID reader fails to read an RFID tag even in static circumstances, due to collisions. In a mobile vehicular environment, latency becomes the key aspect because of the high speed of cars. In this paper, it has been discovered that the RFID study latency and as a consequence effectiveness of on-vehicles reader installations for a huge variety of speeds. First, we experimentally look at the impact of reader and tag relative positions on study mistakes and examine prices.

Microcontroller is used for data collection and processing. The RFID reader reads the information from the tag and transmits it to the microcontroller, which then the tag carries that information for the car to move left, right, speed up and speed down. All of this data is processed and stored in the microcontroller. The RFID system consists of RFID reader, reader antenna, microcontroller and tag. The application of RFID here is that the car follows the instructions from the tag through the reader to move around.

This system brings out the collision avoidance system that is effectively supported by the vehicle's RFID reader and tags. The tags are placed in a particular sequence so if the car moves out of the path an alarm message is sent accordingly to the emergency contact specified. There are two types of collision, one being the reader collision and the second one: tag collision.

A car is a fast moving so the reader and tag should also be as fast as it. The average read rate for this system is 38.89ms. The flaw here is that this read rate is not sufficient the reader misses some of the tags while travelling in high speed. Hence a more advanced system should be used for shorter random-access time.

RFID tag cluster models are explained in the paper too. There are 4 clusters models, which is based on the pitch angles varying from zero degrees to twenty degrees. When the angle is 60, 90, 120 there is no communication hence we use 0 to 20 pitch angles. A long pause time helps prevent RFID tags from collision. On the other hand, it could degrade RFID performance in time-sensitive vehicular applications that require very short read latency. For estimation of the RFID performance, the paper measured the RFID read latency and read rate by using a commercial RFID system. In particular, the road experiments showed the feasibility of applying the RFID system to vehicles. Two techniques, i.e. dual RFID reader antenna and RFID tag cluster, were tested to discuss factors affecting performance. They also suggest directions for improvement. The results show that the proper setting of certain critical parameters leads to success.

3. Methodology

3.1 Research Methodology

This research started with deciding the required hardware necessary for the tasks that we needed to implement. We decided to go ahead with the Arduino Mega microcontroller, then we started with the basic connections of the Arduino Mega with the RFID sensor and began exploring the different tasks that can be implemented using the RFID sensor. We started testing the RFID sensor by implementing some rudimentary codes that can help us understand the working of the RFID sensor. This was done by writing a code to show us some random data in the serial monitor when we scan our RFID tag to the RFID sensor which reads that tag. After that, we decided to integrate other distance measuring sensors with RFID. We tested two different sensors i.e. Proximity sensor and Ultrasonic sensor.

After testing was complete, we decided to use an ultrasonic sensor because of the accuracy that it provides and also the low power consumption in comparison to our proximity sensor. We integrated the RFID sensor and the ultrasonic sensor by controlling the working of the ultrasonic sensor with the RFID tags. Next, we implemented the ultrasonic sensor initiated only when we scan the right tag. If we try to activate the ultrasonic sensor with the help of the wrong tag then we do not get the output from the ultrasonic sensor. Our code is formatted in that we have specified that when we scan the RFID tag that has the correct UID then only the ultrasonic sensor is activated and we receive the output from it. We also have implemented this code in two different requirement levels. The first way is that we get the output from an ultrasonic sensor when we constantly scan the RFID tag in front of the RFID sensor. The second way of implementation is that we have introduced a counter in our Arduino code that allows the ultrasonic sensor to print the output values for a couple of seconds by scanning the RFID tag only once. Each of the implementations has its own advantage. In the initial implementation, the advantage is that

activating the ultrasonic sensor according to our need and it will give us an output only up to the time that we require. The advantage in the second implementation is that we will not need to keep scanning the RFID tag. It will give us an output for a couple of seconds regardless of the fact that we have only scanned the RFID tag once.

Next, we implemented the camera module with the help of Arduino Mega microcontroller. We began the experiment with the OV7670 camera module. This particular module is difficult to connect with the microcontroller, as it requires specific type of resistors for connection in order to be successful and provide accurate and precise results. There are multiple pins in this module where we need to provide the 3.3V supply voltage. This presents the problem that not every required pin gets the required supply voltage of 3.3V. This hinders the working of this particular camera module. The solution to this might be to provide a different power source for each of the pins that require the supply voltage of 3.3V. Due to this issue, we decided to move on to another camera module, the OV2640. This camera module allows us to capture images of different resolutions and on various zoom sizes. This camera module works on a 5V supply voltage and it requires eight pins including 5V Vcc and the ground terminal. The camera can be used in MCU, Raspberry Pi, ARM, DSP, and FPGA platforms. After the connections and installations of the appropriate required libraries, we uploaded the sample code that is provided in those libraries. This makes it possible to open the ArduCAM host application to capture any image. It is important to make changes in the library files so the correct camera module in selected before use. This helps us to avoid any SPI interface error.

For the RFID tag testing, three approaches were taken. The photographs of the RFID tags are shown in Figure 3-1 and 3-2. In Approach 1, several different sets of RFID tags based on thickness and distances was tested. Both linear and nonlinear regression analyses were conducted in the

following form: $Y=\beta_0+\beta_1X_1+\beta_2X_2$, where Y=read rate, x_1=distance, x_2=thickness in mils. In Approach 2, a different set of RFID tags based on distance and the range of RFID was tested for the ranges of 12, 13, 14, and 27 feet. For each range, three experiments were conducted for repeatability; thus, therefore, twelve tests were conducted. Both linear and nonlinear regression analyses were conducted in the following form: $Y=G_0+G_1X_1+G_2X_2$, where RSSI=read signal strength intensity, x_1=distance, x_2=RFID ranges specifications. In Approach 3, both linear and nonlinear regression analyses were conducted in the following form: $Y=G_0+G_1X_1+G_2X_2+G_3X_3$, where RSSI=read signal strength intensity, x_1=distance, x_2=RFID ranges specifications, x_3=thickness of tag.

Figure 3-1 Tags Tested for Approach 1

Figure 3-1 represents one of the tags which was tested for Approach 1.

Figure 3-2 Tags Tested for Approach 2 and 3

Figure 3-2 represents the tags tested for Approach 2 and 3 with RFID ranges specifications and thickness.

Figure 3-3 RFID Tag Experimental

Figure 3-3 represents the testing of RFID Tag during experiments.

3.2 Convolutional neural networks

First, convolutional neural network (CNN) is a type of artificial neural network utilized in image recognition and processing specifically designed to process pixel data. CNNs are powerful models in object and detection classification. Although CNNs are great in accuracy for object detection, their ability to analyze is known to be arduous. For example, it is always difficult to prove the correctness in detecting specific objects in CNN models. In deep learning models, convolutional neural networks combine artificial neural networks and convolutional algorithms to identify a broad spectrum of targets. According to Szegedy et al. 2013; Papernot et al. 2016, to address this issue, the use of optimization-based techniques, it is possible to find the factors that are causing the model to misinterpret or classify the image. Another method available to prove the correctness of neural networks is by using linear programming, according to Huang et al., 2016; Katz et al., 2017. A research study by Dreossi et al. 2017 created a framework to test CNNs by generating synthetic datasets with a specific focus on autonomous vehicles. The framework consisted of an image generator, a collection of sampling methods, and visualization tools. The purpose of the image generator was to extract images of road scenarios, images taken by arranging basic objects by background and cars; and also, by the tuning the image categorized into brightness, contrast and saturation. With this approach, larger amount of realistic images are obtained so that all configurations of the objects and images are defined in the case of road scenarios. On the other hand, the method of sampling purpose is to provide modification points to the image generator to produce images used to render information from the neural network. In the sampling methods, the focus is to utilize optimization methods to generate the images which are classified incorrectly, or

with slight variation by the CNN and lastly the visualization tools is then to display information taken previously.

3.3 YOLOv4 algorithm

In this research, we will be using the YOLOv4 (You Only Look Once), a target real-time detection algorithm, a popular convolutional neural network is an algorithm used for object detection and obstacle avoidance. Several studies have represented improvements and advancements to the previous YOLO generation models. The previous versions of the model provided low detection accuracy. The model provides multiple object detection in image and video for traffic surveillance applications using customized datasets. YOLOv4 contains 80 built-in object classes that is able to detect. Accurate and precise algorithms for object detection allows computers to drive vehicles without sensors, enable assistive devices to convey real-time scene information to human users. The YOLOv4 model detect targets efficiently and accurately. One of the greatest advantages that the algorithm is its incredible speed with its ability of process 45 frames per second. The YOLOv4 algorithm also is capable of understanding object representation. Other advantages include that it is trained to perform classification and bounding box regression simultaneously. We have used the KITTI and BDD datasets for our research study. The KITTI and BDD are datasets used for autonomous driving and object detection.

In order to get the YOLOv4 algorithm to work is by dividing the image into N grids, each having an equal dimensional region of SxS. The main idea is “slicing”, or splitting an image into smaller images.

3.4 Structure of Model

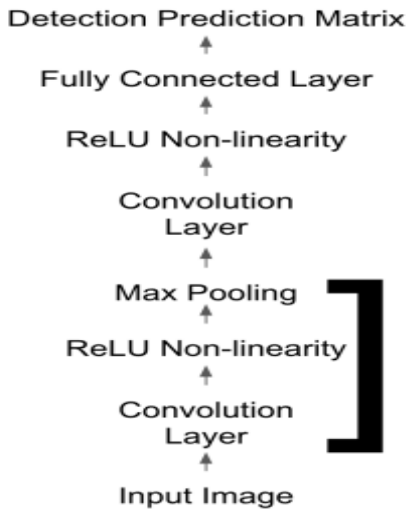
The structure of the proposed model is based on Convolutional Neural Networks (CNN). In a CNN model the input image is passed through layers and each of them performs of multiple operations

on the image. The output generated using these layers is a matrix of bounding box predictions. For a CNN model the operations generally follow a pattern. First, a convolutional operation which is followed by a non-linear activation function and a maximum pooling operation is at the end. This is a general pattern found in every single layer, but the functions can itself vary according to the task at hand. In this architecture, the prediction is a result of n such layers and a layer for applying convolution/non-linear for the last time followed by a dense fully connected layer. To check how the predictions are affected by the number of layers, a 3 layer and 5-layer model are considered.

It is critical for the model to detect small details like traffic signals and distant objects from the input. Thus, the filter of 3x3 pixels is used for training. This is helpful in achieving a greater number of activations. To match the size of volume to the volume of the input, the convolution layers have three neurons as well. Less number of neurons in each layer means the number of parameters of the model also decreases. This means that the model will have less training time and the required computation power also decreases.

Non-linear Activation Function:

Rectified Linear Unit (ReLU) activations is used as the non-linear activation function, which take the form: $ReLU(x) = \max\{0, x\}$



SimpleNet Architecture. Dark bracket indicates portion with repeated layers

Rectified Linear activation function almost acts like a linear function, however, a nonlinear function which helps to learn complex correspondences in the data. This function also avoids saturation. It allows for large gradients and suppresses negative activations.

Detection Prediction:

The x and y coordinates of top-left corner are the first two elements in the vector b ∈ R^4 which represents the detection box. Similarly, the x and y coordinates of the bottom-right corner are the last two elements in this vector. The number of bounding boxes in the output is limited as the detection matrix has fixed number of rows.

Horizontal Suppression:

There are some objects in the input which are too far or in the sky and thus the decisions of autonomous vehicles should not depend on such objects. If a model detects such objects, then it is defined as a false positive. A model's performance also depends on avoiding these false positives. A heuristic is used for avoiding the false positives. The angle of road surfaces and the cameras is the same and thus they both have same horizontal line. Therefore, if the detected object is above the horizon line, then it is a false positive. Hence, we suppress any detection box whose bottom-right corner appears above the estimated horizon line y = 200px according to the heuristic. By doing this we avoid detecting unnecessary object and detect the objects which are more important and closer. This process is called as Horizontal Suppression.

Loss Function:

Element-wise difference between the predicted values and the actual values is examined using two different loss function viz. L1 loss and L2 loss. Let vec (X) denote the operation reshaping the matrix X ∈ R n×m into a vector x ∈ R nm, let matrix A hold the prediction, and matrix B point to the actual values our two losses are then:

$$L1(A-B) = \sum_i |vec(A-B)_i|$$

$$L2(A-B) = \sum_i vec(A-B)_i^2$$

Note that the L2 loss penalizes large differences between the predicted and actual matrices more harshly than the L1 loss. L2 regularization is incorporated on the weight matrices to decrease overfitting; over 10 trial runs, The regularization value of 1e − 5 works best.

Training Algorithm:

Adam optimizer is used to accelerate training and convergence. Each parameter of the network updated in a kind of average of previous gradient updates. For our learning rate, we used 1e − 3. To make full use of each training example, one image is trained at a time. It takes 30 epochs for training the model.

3.5 Hypothesis Testing

Hypothesis 1:

H_0: There is an unacceptable lag time in the readings between the proximity sensors, RFID tag, and camera.

H_1: There is an acceptable lag time (can integrate into vehicle system) between the proximity sensor, RFID tag, and camera.

Reject null hypothesis when p value <0.05

Reject H_0: Acceptable lag time above 85% of the time.

Hypothesis 2:

H_0: The AI model will not give an accurate and precise direction on the path to take.

H_1: The AI model will give an accurate and precise direction on the path to take.

Reject H_0 : The AI model gave an accurate and precise direction on the path to take at a level above 90%

Hypothesis 3:

H_0: The AI model will not accurately identify and detect objects coming towards the vehicle.

H_1: The AI model will accurately identify and detect objects coming towards the vehicle.

Reject H_0: The AI model accurately identify and detect objects coming towards the vehicle at a level above 80%

Hypothesis 4:

H_0: The AI model will not minimize the objects/destructions in the vicinity of the vehicle.

H_1: The AI model will minimize the objects/destructions in the vicinity of the vehicle.

Reject H_0: The AI model minimized the objects/destructions in the vicinity of the vehicle at a level above 90%

3.6 Research Goal and Objectives

The question that is to be addressed in this study is if a low-cost AI-based sensor be retrofitted to make an affordable car self-driving cost effective. Our long-term research goal is to investigate AI driven low-cost sensors that can facilitate automated systems effectively as well as being affordable. The research objective is to provide an effective demonstration of a low-cost AI-based sensor that can enable an affordable car system to become smart and ultimately transition into a self-driving car, or autonomous vehicle.

The specific objectives of this research are:

Specific Objective 1: Compare the performance of the RFID tag, proximity sensor, and camera readings that can simulate a LiDAR type sensor used for affordable vehicles.

Task 1: Providing an effective demonstration of a low-cost AI-based sensor that can enable a used vehicle system to become smart and transition to self-driving.

Task 2: Understand and identify the factors that impact a used car in making it become a smart self-driving vehicle.

Specific Objective 2: Develop and compare algorithms that use the dataset and integrate into older car data systems.

Task 1: Learn---Identify what is coming towards the vehicle which is potentially hazardous/dangerous. In this objective, we want to minimize the objects/ destructions coming at the vehicle.

Task 2: Optimize---Based on the available paths, the AI model will allow and give direction as to the path to take.

Task 3: Execute---Override and give information to the vehicle to drive to location.

Specific Objective 3: Perform an economic and disaster analysis.

Task 1: Evaluate and understand what decisions will be made by industry, consumers and advocates to use these technologies and at what level of safety.

Task 2: Engineering economics calculations and analysis will be performed using software to validate.

Task 3: Analysis and calculations will be performed and software will be used to validate results

Specific Objective 1: Compare the performance of the RFID tag, proximity sensor, and camera readings that can simulate a LiDAR type sensor used for affordable vehicles.

We will perform a variance analysis for all three sensors by using regression analysis and design of experiments. The intent is to maximize the distance before the RFID tag, proximity sensor

and/or camera gives a signal that is going off. Variance analysis will be used for all three sensors. Mean squared error formula as shown below.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

MSE = mean squared error
n = number of data points
Y_i = observed values
$\hat{Y}_i$ = predicted values

Specific Objective 2: Develop and compare algorithms that use the dataset and integrate into older car data systems.

There are three AI components to the research: learn, optimize, and execute. For the AI component of learn, the machine learning algorithm will seek optimization. The model will be able to identify what is coming towards the vehicle that is potentially hazardous/dangerous as well as minimize the objects/destructions in the vicinity of the vehicle. The optimize component will use the data to evaluate the best available path to allow and give direction as to the path to be taken. Lastly, for the execution, we will integrate the algorithms for the vehicle to be able to drive.

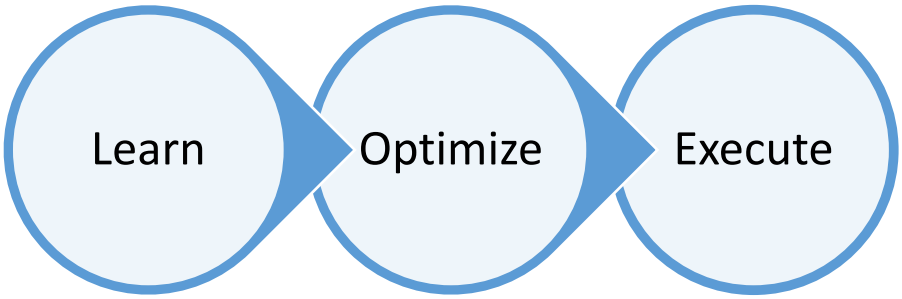


Figure 3-4 Three-step process for algorithm

Figure 3-4 shows the three AI components to the research.

Specific Objective 3: Perform an economic and disaster analysis.

We will perform calculations and analysis will be done; software will be used to validate. For the economic analysis, the future value formula as shown below will be used.

$$FV = PV(1 + r)^n$$

FV = future value
PV = present value
r = annual interest rate
n = number of periods interest held

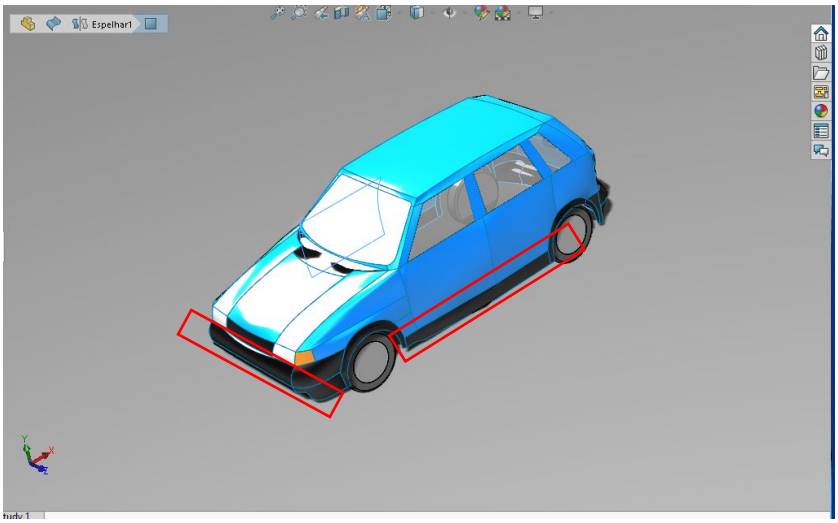


Figure 3-5: 3D Model of Vehicle.

Figure 3-6: Pinout Diagram for Ultrasonic Sensor.

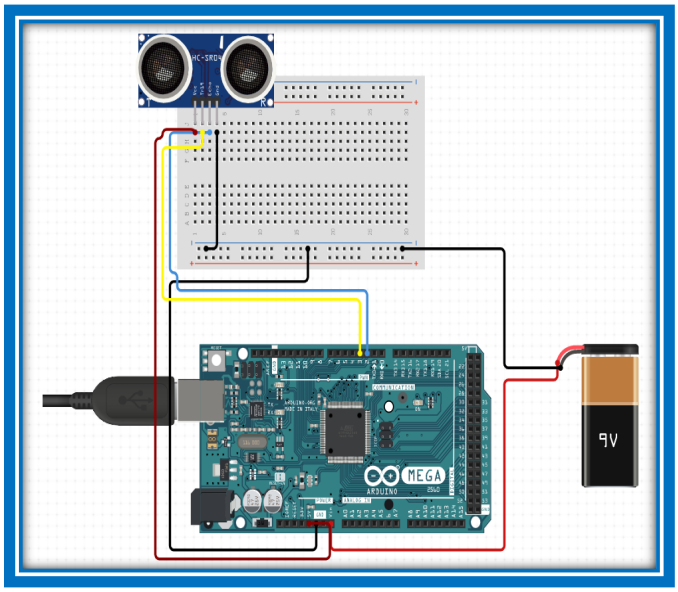


Figure 3-7: Connecting Ultrasonic Sensor with Arduino Microcontroller

Figure 3-7 represents the pinout diagram of the connection of the ultrasonic sensor using the Arduino Microcontroller.

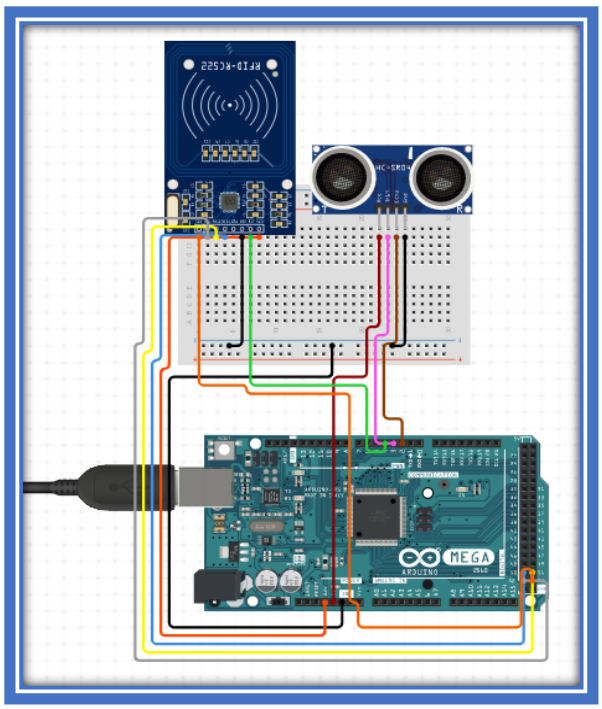


Figure 3-8: Connecting Ultrasonic Sensor with Arduino Microcontroller with RFID Tag

Figure 3-8 represents the pinout diagram of connecting the ultrasonic sensor with the Arduino Microcontroller with the RFID Tag.

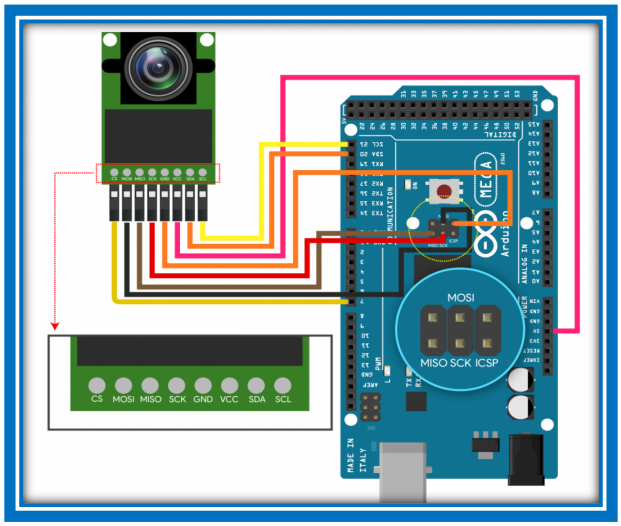


Figure 3-9: Connecting Ultrasonic Sensor with Arduino Microcontroller with RFID Tag and Camera Lens

Figure 3-9 represents the connection of the ultrasonic sensor with Arduino microcontroller with RFID Tag and Camera Lens.

Figures 3-7, 3-8, and 3-9, represent the pinout diagram(s) of the connection of the ultrasonic sensor with the Arduino microcontroller with each of the elements: the RFID tag and the camera.

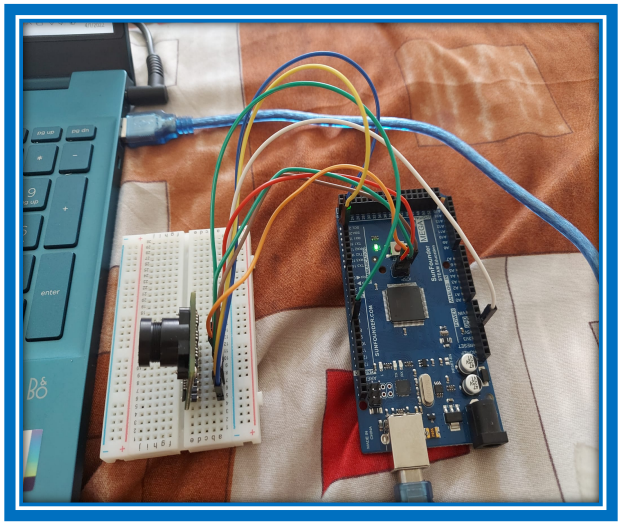


Figure 3-10: Hardware connection of camera module

Figure 3-10 represents the actual hardware connection of the camera module, Mini Module Camera Shield with OV2640. The challenge here was the setup which was tedious.

3.7 AI Model Methodology

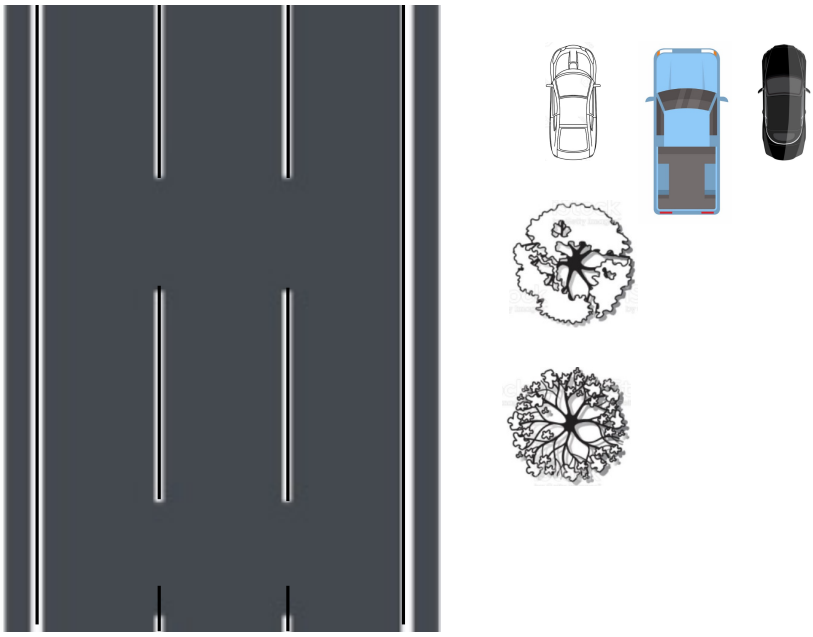


Figure 3-11 Overview of Schematic

Figure 3-11 represents the overview of the schematic for the vehicle. The road is assumed to be a two-lane road. The obstacle/object may be anything from another vehicle, tree, stone, and/or pedestrians.

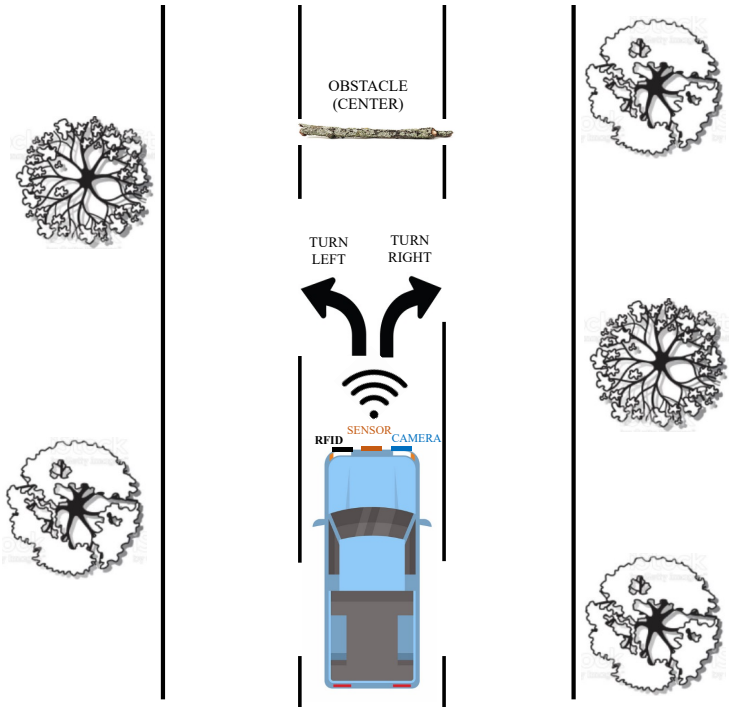


Figure 3-12 Overview of Vehicle Scenario

Figure 3-12 shows the overview of the vehicle scenario in a potential natural disaster situation. The vehicle will be equipped with a RFID tag, proximity sensor, and camera, so that the camera and proximity sensor will be synced to communicate the location of the vehicle.

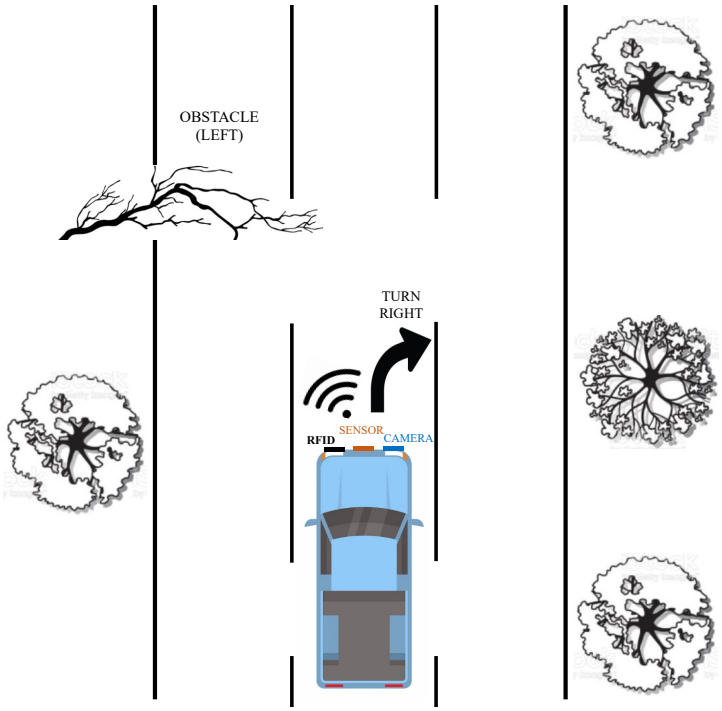


Figure 3-13 Scenario 1 for Vehicle in Danger

Figure 3-13 exhibits scenario 1 for vehicle in danger. The schematic shows the scenario that there is an obstacle in the vicinity to the left of the vehicle, leading the vehicle to make an efficient and safe right turn.

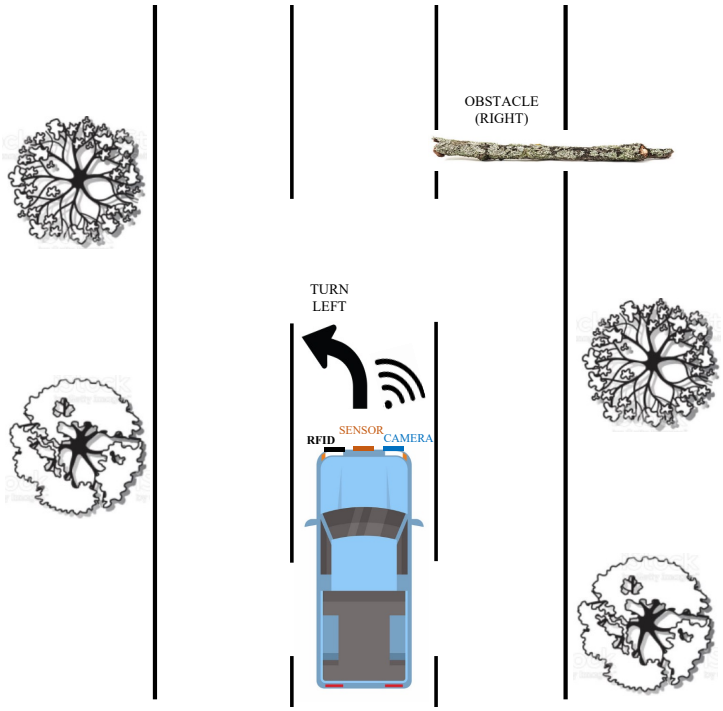


Figure 3-14 Scenario 2 for Vehicle in Danger

Figure 3-14 illustrates scenario 2 for the vehicle in potential danger. In this case, the vehicle has detected an obstacle to its right, requiring the vehicle to make an efficient and quick decision to turn to the left.

3.8 Equipment List

Table 3-1: List of equipment

Name of Equipment	**Cost**
Arduino Mega Microcontroller	$40.30
DEYUE 3 Set of Solderless Prototype Breadboard 830 tie Points Breadboard	$8.99
MFRC522 RFID sensor and tags	$10.00
HC-SR04 ultrasonic distance sensor	$8.29
Jumper wires (Male to Male)	$12.99
Jumper wires (Male to Female)	$13.59
Comimark 2Pcs VGA OV7670 CMOS Camera Module Lens CMOS 640X480 SCCB I2C Interface for Arduino	$8.99
Arducam Mini Module Camera Shield with OV2640 2 Megapixels Lens Compatible with Arduino UNO Mega2560 Board and Raspberry Pi Pico	$25.99
4.7K ohm Resistor	$5.99
10K ohm Resistor	$5.99
Total	$141.38

Table 3-1 shown above is the list of equipment, which utilized for this research project. Before the equipment was finalized for the project, several factors were taken into consideration. Some of the factors taken into consideration included cost, feasibility, and the amount of time required for the

equipment to be implemented. In some cases, equipment was purchased, only to find out that it was not compatible with other equipment.

3.9 Location of Experiments

All testing and experiments were performed at the RFID, Artificial Intelligence, and Data Science Laboratories (RAID LABS) at the University of Texas at Arlington. Setup for testing was prepared one to two days prior to actual experiments. Each of the experiment setups took about an hour to set up. Actual experiments took about two hours, considering there were no issues during the testing process.

3.10 Data Collection Phase

The data collection for this research will be divided into three phases. The phases are the following:

Phase one - Collect data from RFID, camera, and proximity sensor into a database and design an algorithm that takes that data and can optimize something that can be sent to a vehicle like a computer.

Phase two - Optimizing the algorithm to make sure that it supports smart vehicle functions and objectives.

Phase three - Testing the ability for the vehicle to be autonomous.

3.11 Software Used

The software used in this research study consists of the following: Python, MATLAB, and Arduino software. Python and MATLAB have been used for the coding of the artificial intelligence (AI) model, and the Arduino software has been used for the coding of the RFID tag, camera, and proximity sensor. Throughout this research study, the software caused roadblocks along the way. In such scenarios, it was required for the code to be debugged and/or altered so that the software

provided a correct and useful output. This was the case, especially with the accuracy of the detection for AI model and the Arduino software for the camera sensor module.

4. Results

4.1 Preliminary Results for RFID Tag

We have the output data for the ultrasonic sensor by itself with the timestamp of when the output was received.

The output data from the ultrasonic sensor that is being stored in an excel database along with the timestamp of when that particular output was received from the ultrasonic sensor has been included. Table 4-1 represents the output from the code where we have implemented the counter to allow the ultrasonic sensor to run on its own for a couple of seconds. Table 4-2, on the other hand, shows us the output from the code where we receive the output until the time we keep scanning the RFID tag.

Table 4-1: Sample Output Data for Ultrasonic Sensor with continuous scan of RFID tag

TIME	CH1
12:41:28.39	UID tag: 3A EE 2C 3C
12:41:28.44	Message: Ultrasonic sensor initiated
12:41:28.45	Distance: 7
12:41:28.47	UID tag : 3A EE 2C 3C
12:41:28.52	Message : Ultrasonic sensor initiated
12:41:28.54	Distance: 8
12:41:28.56	UID tag : 3A EE 2C 3C
12:41:28.59	Ultrasonic sensor initiated
12:41:28.61	
12:41:28.61	Distance: 8
12:41:28.65	UID tag : 3A EE 2C 3C
12:41:28.68	Message : Ultrasonic sensor initiated
12:41:28.69	Distance: 8
12:41:28.72	UID tag : 3A EE 2C 3C

Table 4-2: Sample Output data with scan of RFID once and sensor readings

12:38:22.68	Distance: 6
12:38:22.71	Ultrasonic sensor initiated
12:38:22.72	Distance: 6
12:38:22.75	Ultrasonic sensor initiated
12:38:22.77	Distance: 6
12:38:22.80	Ultrasonic sensor initiated
12:38:22.82	Distance: 6
12:38:22.83	Ultrasonic sensor initiated
12:38:22.85	Distance: 7
12:38:22.89	Ultrasonic sensor initiated
12:38:22.90	Distance: 7
12:38:22.93	Ultrasonic sensor initiated
12:38:22.94	Distance: 6
12:38:22.98	Ultrasonic sensor initiated
12:38:23.00	Distance: 7
12:38:23.03	Ultrasonic sensor initiated
12:38:23.04	Distance: 8
12:38:23.08	Ultrasonic sensor initiated
12:38:23.09	Distance: 9
12:38:24.00	Distance: 7

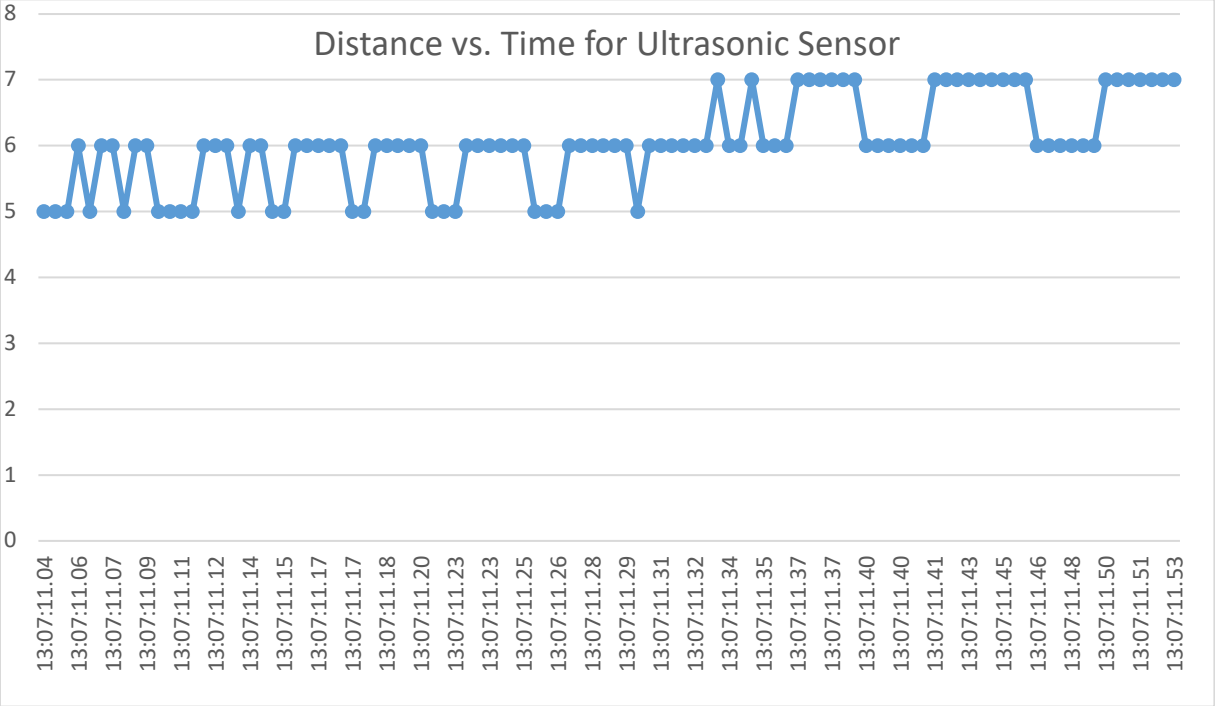


Figure 4-1 RFID tag scan once with continuous Ultrasonic readings

Figure 4-1 represents the distance vs. time of the RFID tag scanned once while the ultrasonic sensor provides continuous readings for the timeframe.

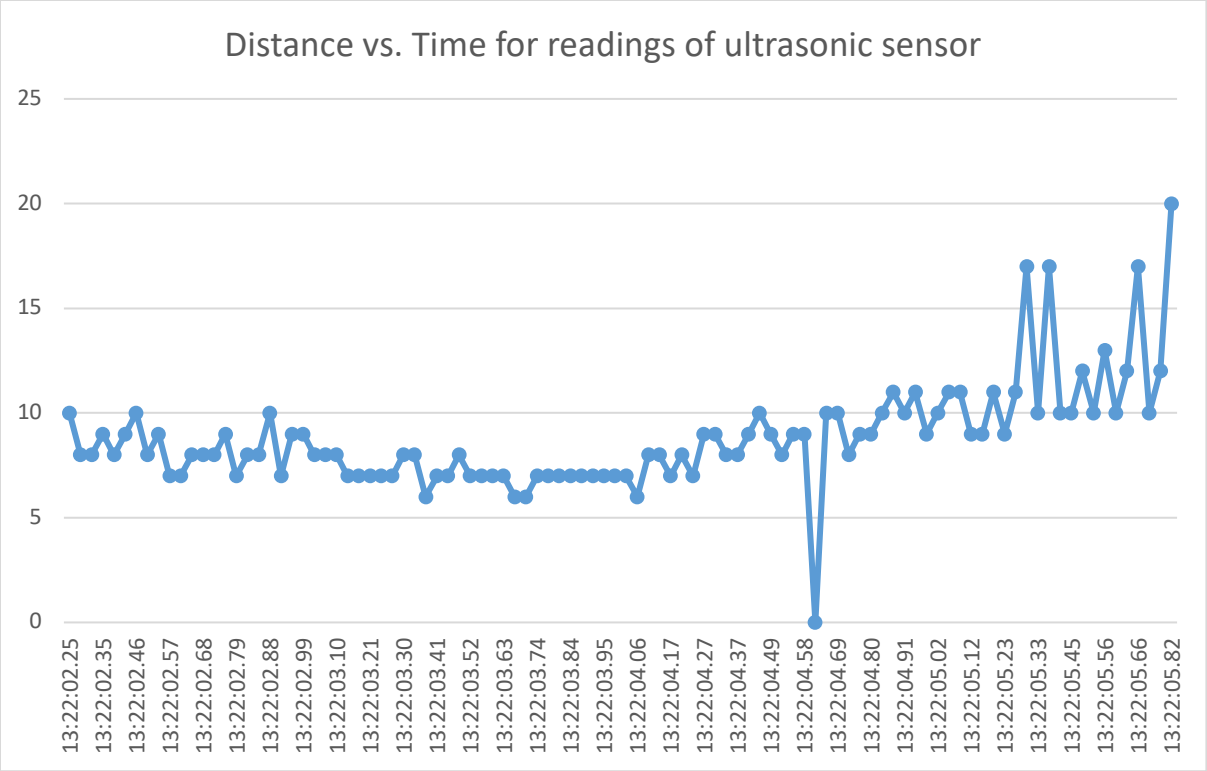


Figure 4-2 RFID tag scanned continuous to give sensor readings.

Figure 4-1 represents the distance vs. time of the RFID tag scanned once while the ultrasonic sensor provides continuous readings for the timeframe.

Figures 4-1 and 4-2 shows the graphs that involve scanning the tag and ultrasonic sensor. In these graphs, we see that distance vs. time has performed differently with RFID with counter and the RFID without the counter. Figure 4-1 gives us the ultrasonic sensor readings when the RFID tag has been scanned once. On the contrary, Figure 4-2 gives us the ultrasonic readings when RFID tag is required to be scanned continuously. During testing and trial run, there are outliers present in the graphs which can be taken as error. Figure 4-1 represents the distance vs. time of the RFID tag scanned once while the ultrasonic sensor provides continuous readings for the timeframe.

Table 4-3 Testing of RFID tag

Feet	31 Mils	41 Mils	62 Mils	63 Mils
3	100	108	109	110
6	84	95	104	110
9	77	90	109	110
12	59	81	94	95
15	49	36	85	103
18	0	22	104	97
21	0	0	109	110
24	0	0	109	110
27	0	0	88	97
30	0	0	43	86
33	0	0	0	110

Figure 4-3 Read Range vs. Distance

Table 4-3 represents the output for the testing of the various RFID tag at different distances and locations. Figure 4-3 represents the read range vs. distance of the RFID tag tested at various distances. As shown in the graph, we can see that the 63 mils, or green line has performed the most optimal. During testing, the tags were tested at 3, 6, 9, 12, 15, 18 ft., etc., to measure the most optimal.

As shown in Figure 4-3, the optimal tag was found to be the 63 mils tag. When testing to find the most optimal tag, the tag(s) were tested at 3, 6, … 33, with increments of 3 feet. Once the optimal tag was determined, the other tags were also tested starting from 1, 2, … 33, with increments of one foot to note and compare the performance and read rate of the tag and the effects, so that a regression could be performed on the data. The testing for the optimal tag, 63 mils has been shown in Table 4-5 Optimal Tag Testing, as an example.

Figure 4-4 Image captured from camera module

Figure 4-4 represents the real-time image captured from the camera module. This image, along with several other images were used to test the artificial intelligence model, the YOLOv4 algorithm for accuracy and precision. The results will later be discussed.

Table 4-4 shows the optimal tag testing (63mils) with the output results.

Table 4-4 Optimal Tag Testing (63 mils)

Distance	Read Rate
1	110
2	112
3	110
4	116
5	110
6	110
7	108
8	110
9	110
10	100
11	109
12	95
13	100
14	103
15	103
16	110
17	97
18	97
19	110
20	110
21	110
22	104
23	110
24	110
25	97
26	107
27	97
28	90
29	95
30	86
31	110
32	110
33	110

4.2 Regression Analysis

A regression analysis is performed. The purpose of the regression analysis is to maximize distance before the RFID tag, proximity sensor and the camera goes off. The simulation is providing a visualization of the optimized regression model. The p-value will be used to determine the optimal distance. The RFID tag, proximity sensor, and camera were scanned into a database to see what timestamp will be given. As mentioned earlier, there were three approaches analyzed for this research. For approach 1, the general formula for regression is stated below.

$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots \beta_i X_i$ [General Form]

For this regression analysis, two independent variables were used which include:

X_1 = distance

X_2= thickness

Regress $\hat{y}$ on X_1, X_2, to give the p-values and R-squared for independent variables.

The result based on regression analysis of the RFID tags is as follows:

$$Y = 6.402 + (-2.528)X_1 + 2.134X_2 \text{ [Specific to this regression]}$$

Where:

Y = Read Rate

X_1 = Distance

X_2 = Thickness of tag[mils]

β_0 = Intercept

β_1= Regression value for Independent Variable 1[distance]

β_2= Regression value for Independent Variable 2 [thickness]

Similarly, for Approach 2, the general formula for regression has been stated below.

$Y = \beta_0+\beta_1X_1+\beta_2X_2+\ldots. \beta_iX_i$ [General Form]

The result based on regression analysis of the RFID tags is as follows:

$$Y = 35.695+(-1.155)X_1+0.498X_2$$

Where:

Y = Read Signal Strength (RSSI)

X_1 = Distance

X_2 = Range of tag

β_0 = Intercept

β_1= Regression value for Independent Variable 1[distance]

β_2= Regression value for Independent Variable 2 [actual range]

Since R-squared for Approach 2 of linear regression analysis was 0.472 or 47.2%, a nonlinear regression analysis was performed. A nonlinear regression analysis was conducted to predict the values of RSSI based on the experimental observations of the four RFID tags experimentally, three times each tested in the following form:

$$RSSI = {X_1}^{G_1}.{X_2}^{G_2}$$

Where:

RSSI = Read Signal Strength Intensity

X_1 = Distance

X_2 = RFID Range

G_1 = Regression values for Independent Variable 1 [distance]

G_2 = Regression constant for Independent Variable 2 [range]

In order to perform the regression analysis, the nonlinear form is first linearized by taking the natural logarithm of both sides of the equation:

$$LN(RSSI) = LN\ ({X_1}^{G_1}.{X_2}^{G_2})$$

$$LN\ (RSSI) = G_1\ LN\ (X_1) + \ G_2\ LN\ (X_2)$$

Thus, the regression analysis is now performed on the linearized form and the results is converted back to nonlinear form by taking the mathematical exponent of both sides of the equation:

$$EXP\ [LN\ (RSSI)] = EXP[G_1 LN(X_1) + G_2 LN\ (X_2)]$$

$$RSSI = G_0\ \left({X_1}^{G_1}\right)\ ({X_2}^{G_2}) \text{ - General Form}$$

$$RSSI = 3.288\ \left({X_1}^{-0.173}\right)\ ({X_2}^{0.208}) \text{ - Specific to this regression}$$

Where:

G_0=3.288 Regression constant obtained after regression analysis for non-zero intercept

G_1= -0.173

G_2 =0.208

After the nonlinear regression analysis was performed for Approach 2, the R-squared gave a result of 0.585 or 58.5%.

A third approach was taken for this research and the general formula for regression is stated below.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3\ \ldots.\ \beta_i X_i \text{ [General Form]}$$

For this regression analysis, three independent variables were used which include:

X_1 = distance

X_2 = range

X_3 = thickness of tag

The result based on regression analysis for Approach 3 is as follows:

$$Y = 27.257+(-1.206)X_1+0.235X_2+0.443X_3$$

Where:

Y= Read Signal Strength (RSSI)

X_1= distance

X_2= range of tag

X_3=thickness

β_0= Intercept

β_1= Regression value for Independent Variable 1[distance]

β_2= Regression value for Independent Variable 2[actual range]

β_3= Regression value for Independent Variable 3 [thickness]

With Approach 3 and a linear regression analysis, the R-squared gave a result of 0.582 or 58.2%.

Next, a nonlinear regression approach was taken.

$$RSSI = {X_1}^{G_1}.{X_2}^{G_2}.{X_3}^{G_3}$$

Where:

RSSI = Read Signal Strength Intensity

X_1 = Distance

X_2 = RFID Range

X_3= thickness

G_1 = Regression value for Independent Variable 1 [distance]

G_2 = Regression value for Independent Variable 2 [range]

G_3=Regression value for Independent Variable 3 [thickness]

To perform the regression analysis, the nonlinear form is first linearized by taking the natural logarithm of both sides of the equation:

$$LN(RSSI) = LN\ ({X_1}^{G_1}.{X_2}^{G_2}.{X_3}^{G_3})$$

$$LN\ (RSSI) = G_1\ LN\ (X_1) +\ G_2\ LN\ (X_2) + G_3 LN\ (X_3)$$

Next, the regression analysis is now performed on the linearized form and the results is converted back to nonlinear form by taking the mathematical exponent of both sides of the equation:

$$EXP\ [LN\ (RSSI)] = EXP[G_1 LN(X_1) + G_2 LN\ (X_2) + G_3 LN\ (X_3)]$$

After performing the regression analysis for the case under consideration the following equation is obtained:

$RSSI = G_0\ \left({X_1}^{G_1}\right)\ ({X_2}^{G_2})\ ({X_3}^{G_3})$ - General Form

$RSSI = 2.569\ \left({X_1}^{-0.178}\right)\ ({X_2}^{0.073})\ ({X_3}^{0.328})$ - Specific to this regression

Where:

G_0=2.567 Regression constant obtained after regression analysis for non-zero intercept

G_1=-0.178

G_2 =0.073

G_3 =0.328

With the nonlinear regression analysis for Approach 3, the R-squared improved giving a result of 0.706 or 70.6%.

These results confirm that as each approach was altered, there was improvement to the results of the research study.

4.3 Read Rate vs. Distance for Tags

4.3.1 Approach 1 Read Rate vs. Distance Plots

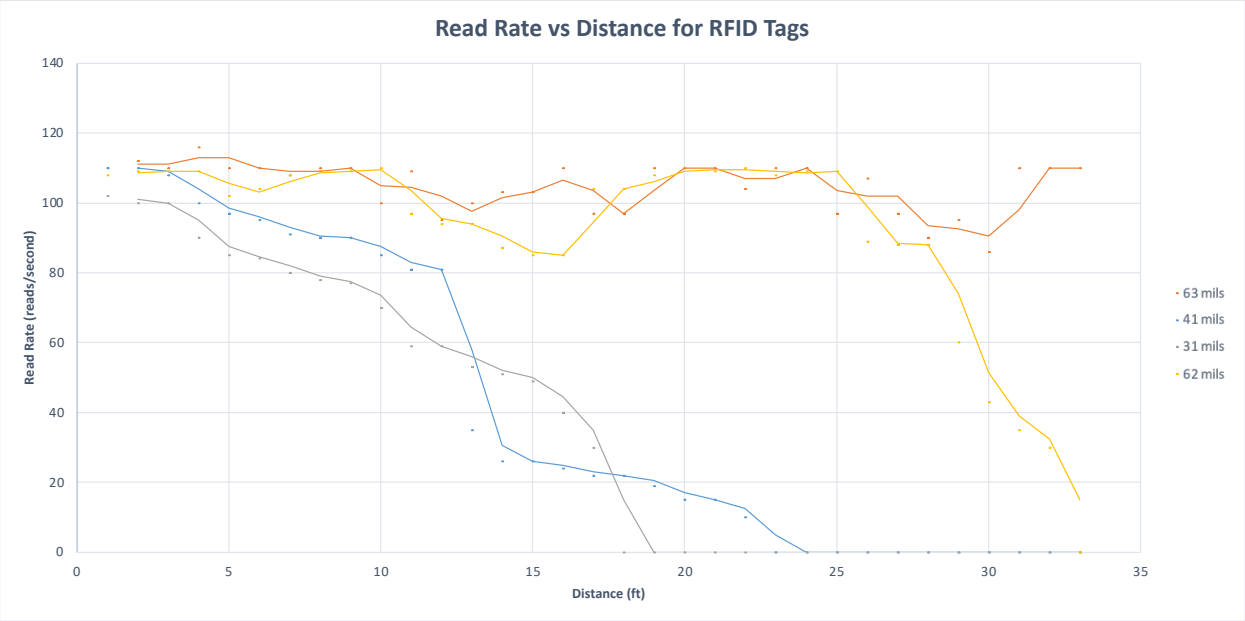


Figure 4-5 Read Rate vs Distance for RFID Tags

Figure 4-5 represents the read rate vs distance for RFID tags with 31, 41, 62, and 63 mils, tested at 1 ft to 33 ft with increments of 1ft. As demonstrated in the graph, the 63 mils tag performed the most optimal. This confirms the previous test which was experimented by testing the tags at 3ft increments, up to 33 ft.

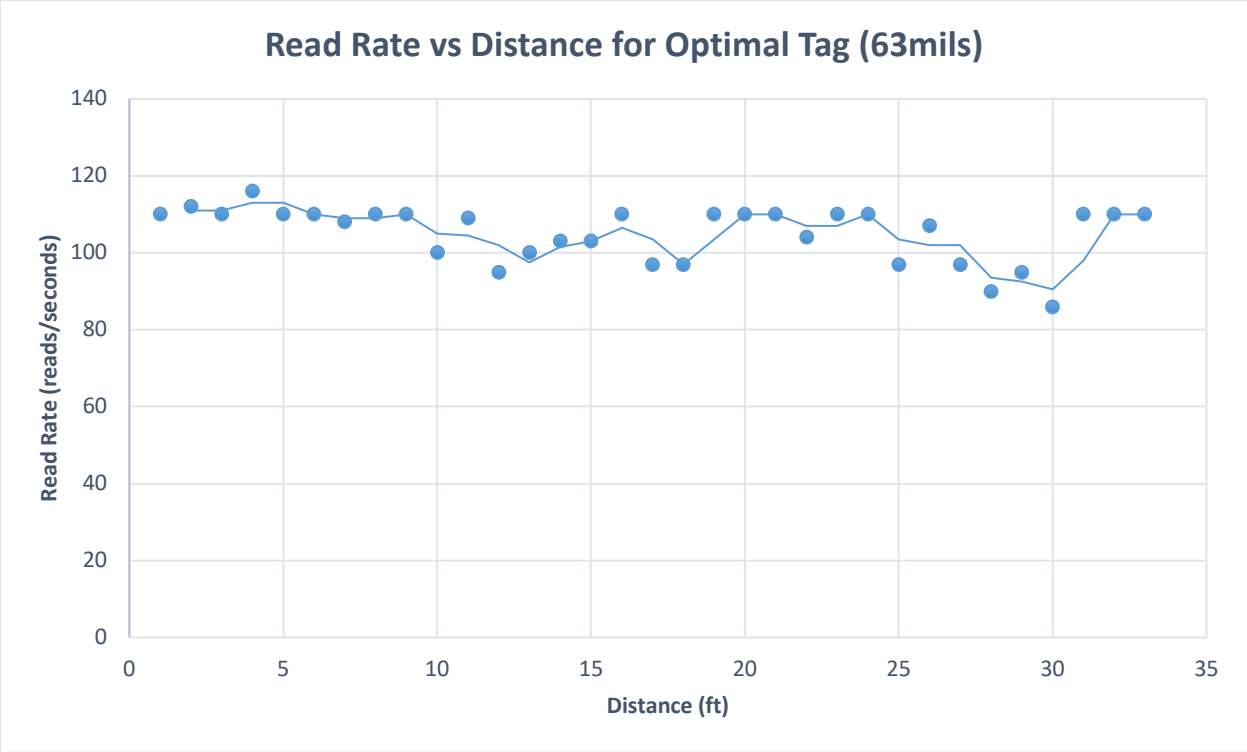


Figure 4-6 Read Rate vs. Distance 63mils

Figure 4-6 represents the read rate vs. distance for the 63 mils tag when it was tested at 1, 2, …33 by increments of one foot up to 33 feet.

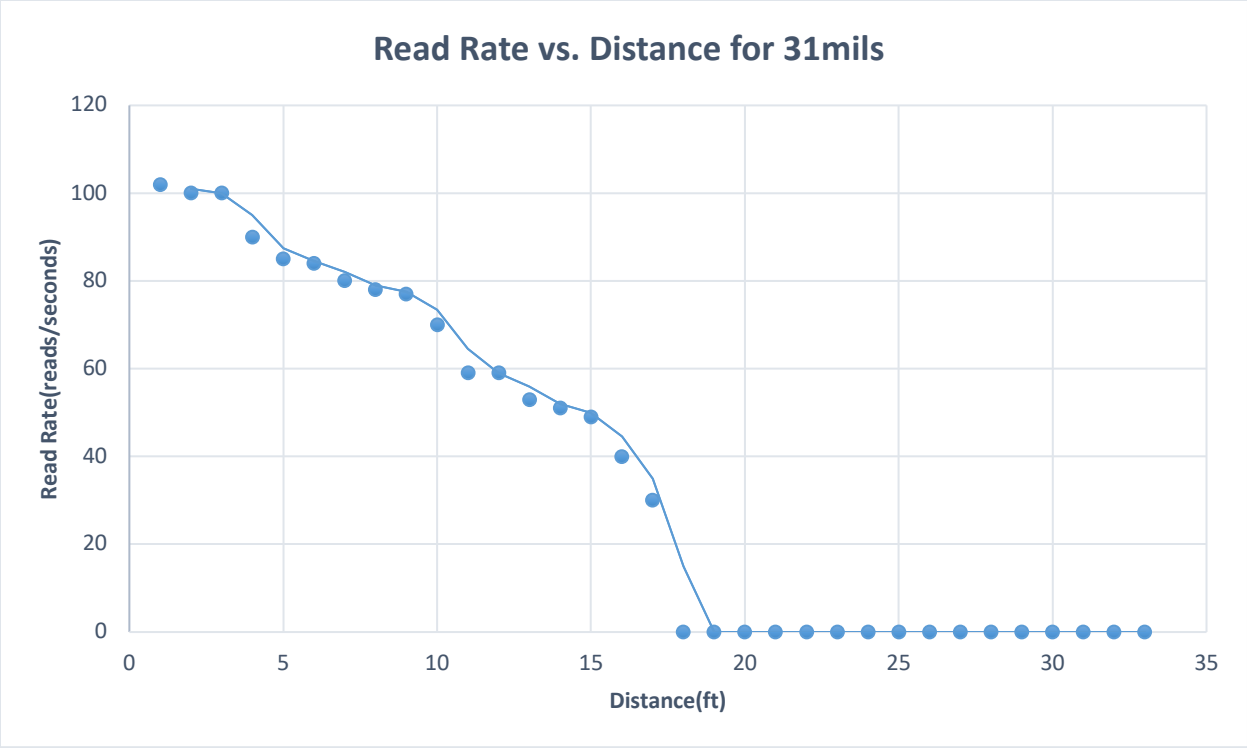


Figure 4-7 Read Rate vs. Distance for 31 mils Tag

Figure 4-7 represents the read rate vs. distance for the 31mils tag when it was tested at 1, 2, … 33 by increments of one foot up to 33 feet. The graph represents a downward trend as the distance is increased.

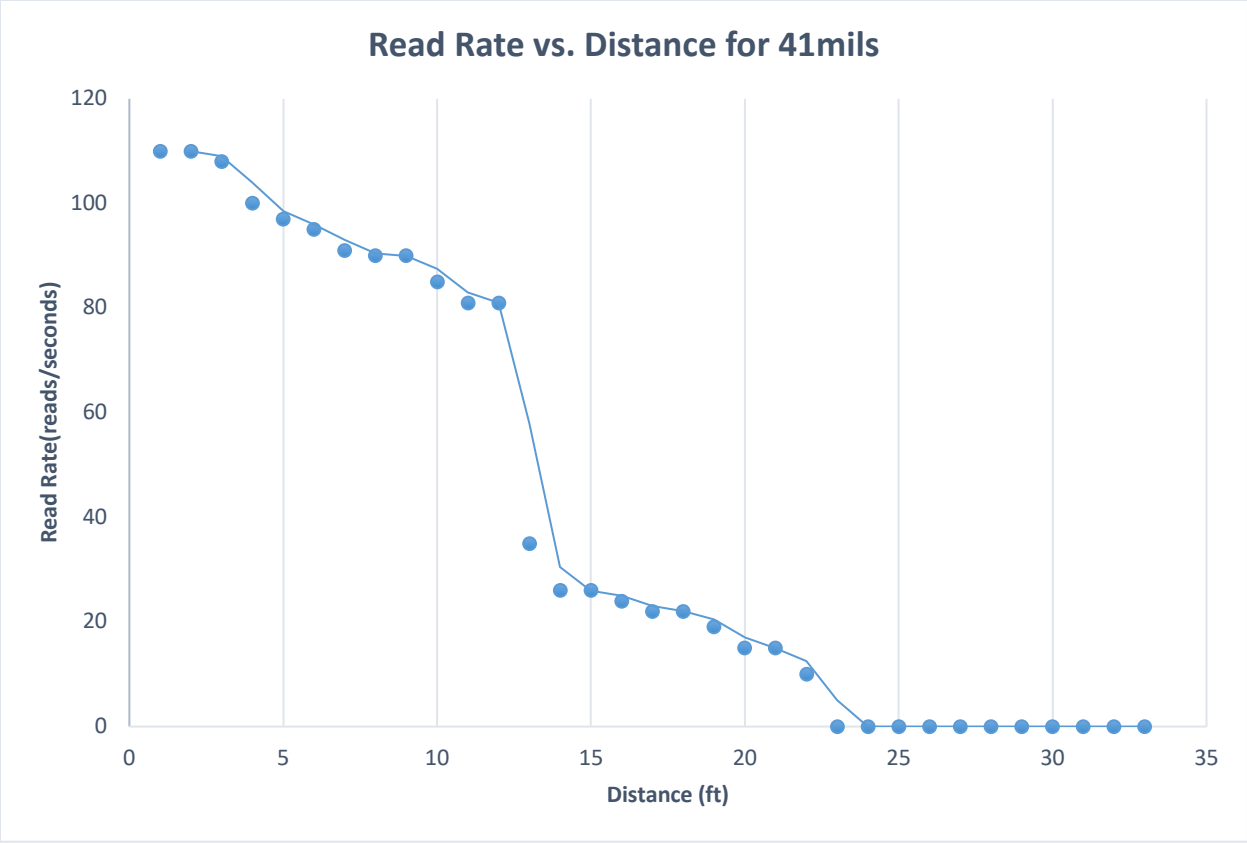


Figure 4-8 Read Rate vs. Distance for 41 mils Tag

Figure 4-8 represents the read rate vs. distance for the 41mils tag when it was tested at 1, 2, … 33 by increments of one foot up to 33 feet.

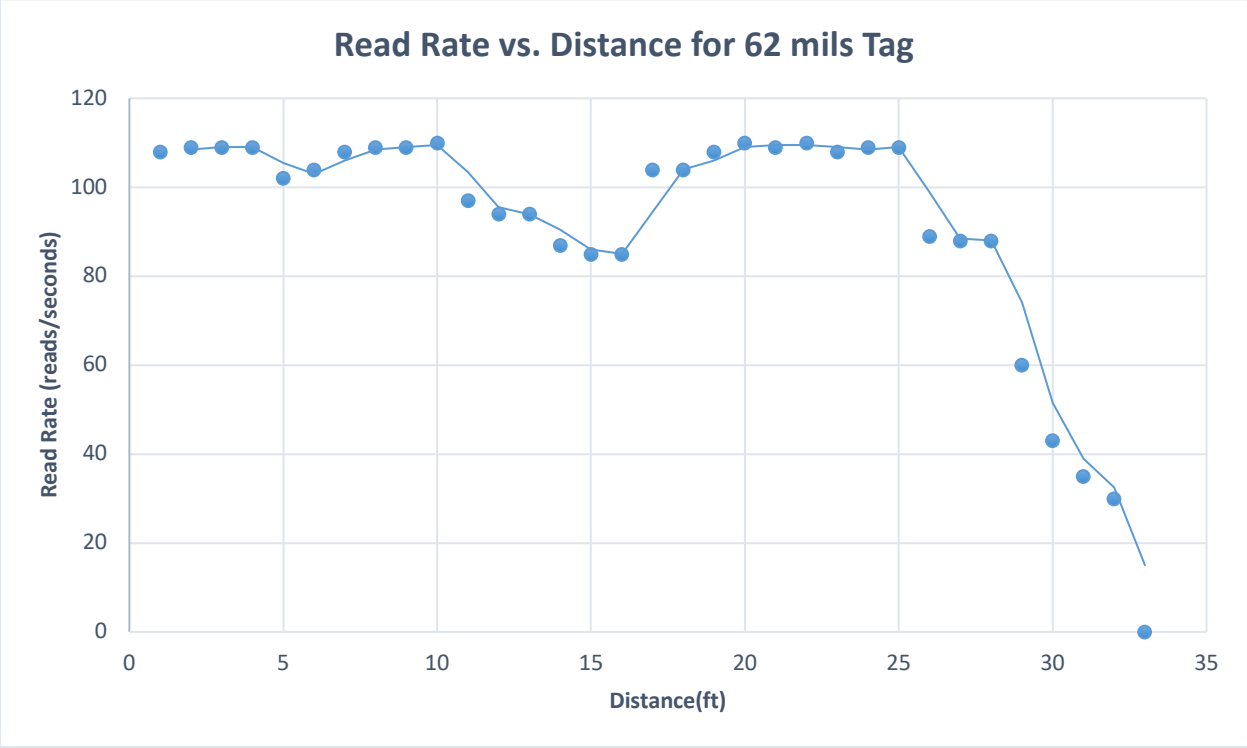


Figure 4-9 Read Rate vs. Distance for 62 mils Tag

Figure 4-9 represents the read rate vs. distance for the 62 mils tag when it was tested at 1, 2, … 33 by increments of one foot up to 33 feet.

The tags tested at each foot interval was performed to compare the trends of the data to the data of the tags which were initially tested at three feet intervals up to 33 feet. It has been determined that overall, the trends followed the same pattern between both datasets and output. This solidifies the initial findings that the optimal tag found was indeed the 63 mils RFID tag. A regression analysis was not only performed for the optimal tag, but also for the additional tags as well, of 31, 41, 62 mils.

4.4 Normal Probability Plots for RFID Tags

4.4.1 Regression Results (NPP Plots)

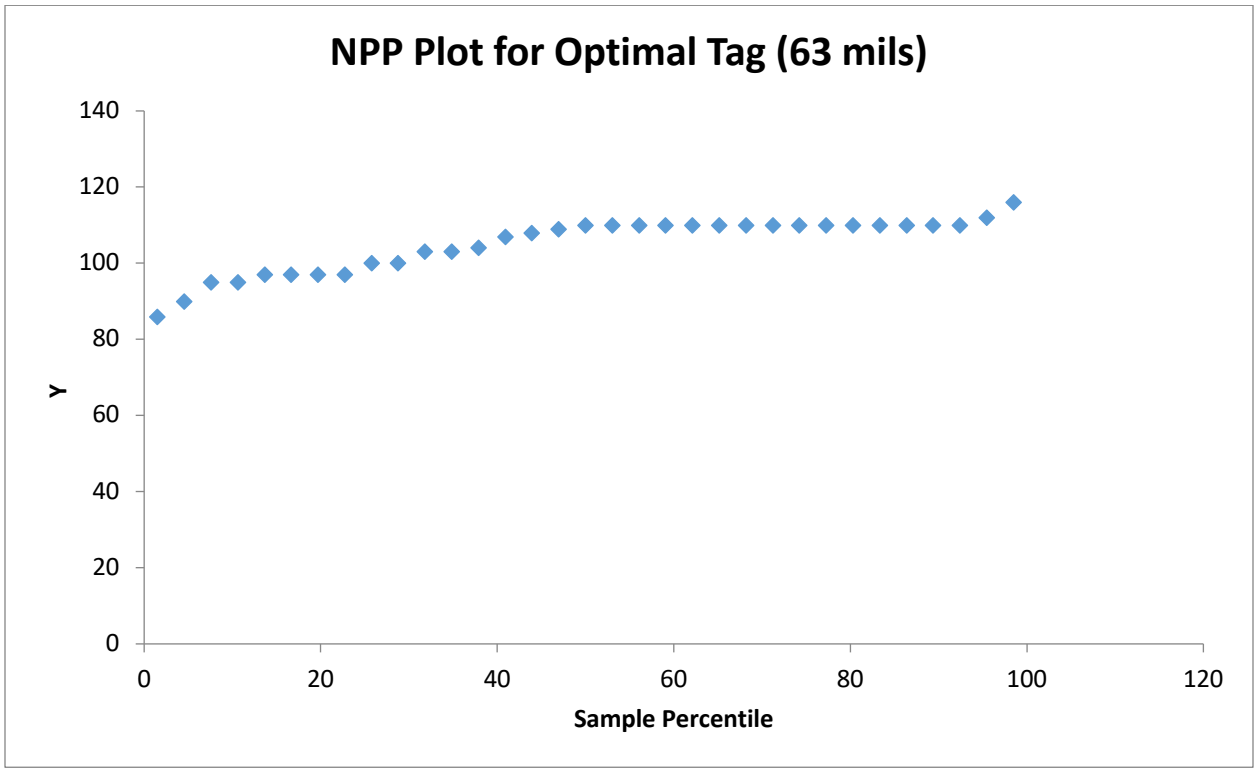


Figure 4-10 NPP for Optimal RFID Tag (63mils)

Figure 4-10 represents the NPP plot for the optimal RFID tag (63 mils) which was determined by the testing performed. From the figure, it is shown that that normality is satisfied as confirming that the 63 mils tag has performed optimally.

4.5 Residual Plots

4.5.1 Residual Plots Results

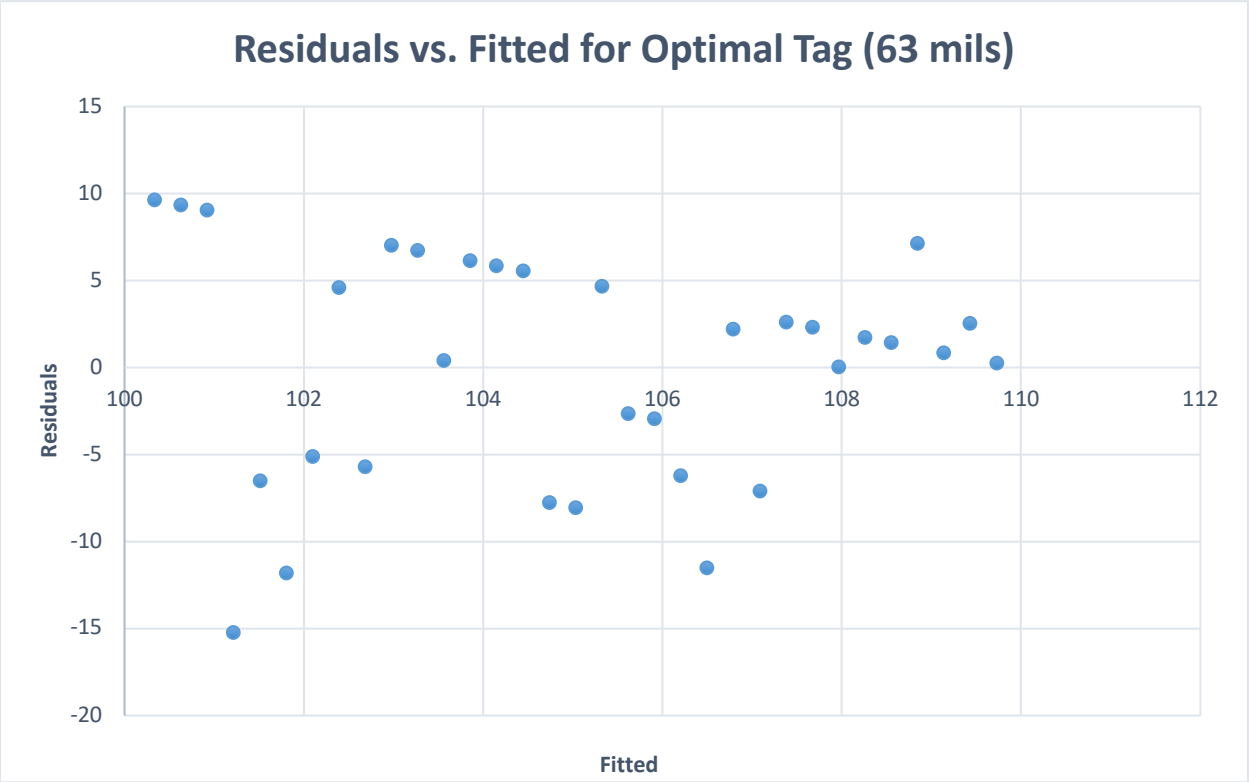


Figure 4-11 Residuals vs. Fitted for Optimal Tag(63mils)

Figure 4-11 shows the residuals vs. fitted for the optimal tag (63 mils). The graph illustrates that there is curvature present, which means that the constant variance has been violated and a transformation for the data is required.

4.6 Predicted Y vs. Actual Y Regression Approach 1

4.6.1 Approach 1 Results

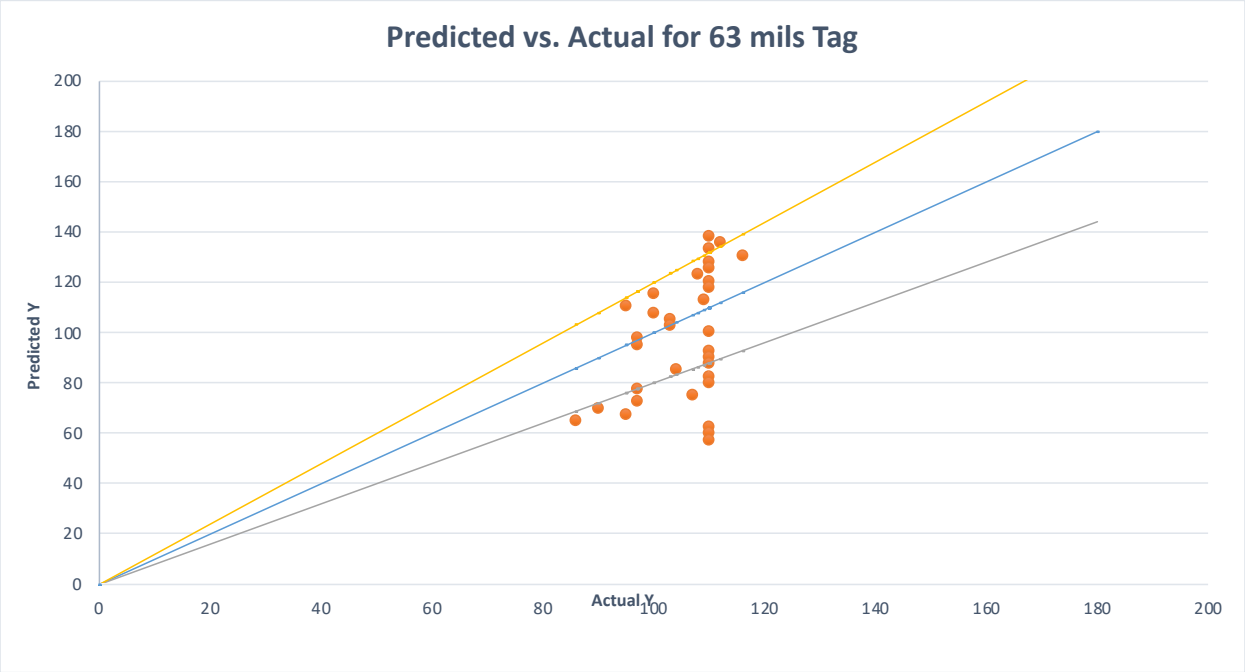


Figure 4-12 Predicted Y (Read Rate) vs. Actual Y (Read Rate) for 63 mils Tag

Figure 4-12 shows the Predicted Y vs. Actual Y for the 63 mils tag, or optimal tag. From the representation, we can see that the data points are between +/- 20% error. The yellow and gray lines in the graph represent the error bands, while the blue demonstrates the 45-degrees line which passes through the data points, representing a one-to-one relationship between the actual and predicted values. It can be noticed that in this dataset, there are outliers, which can be due to human error during experimentation. Also, since this tag was found to be the optimal tag, it has the higher accuracy.

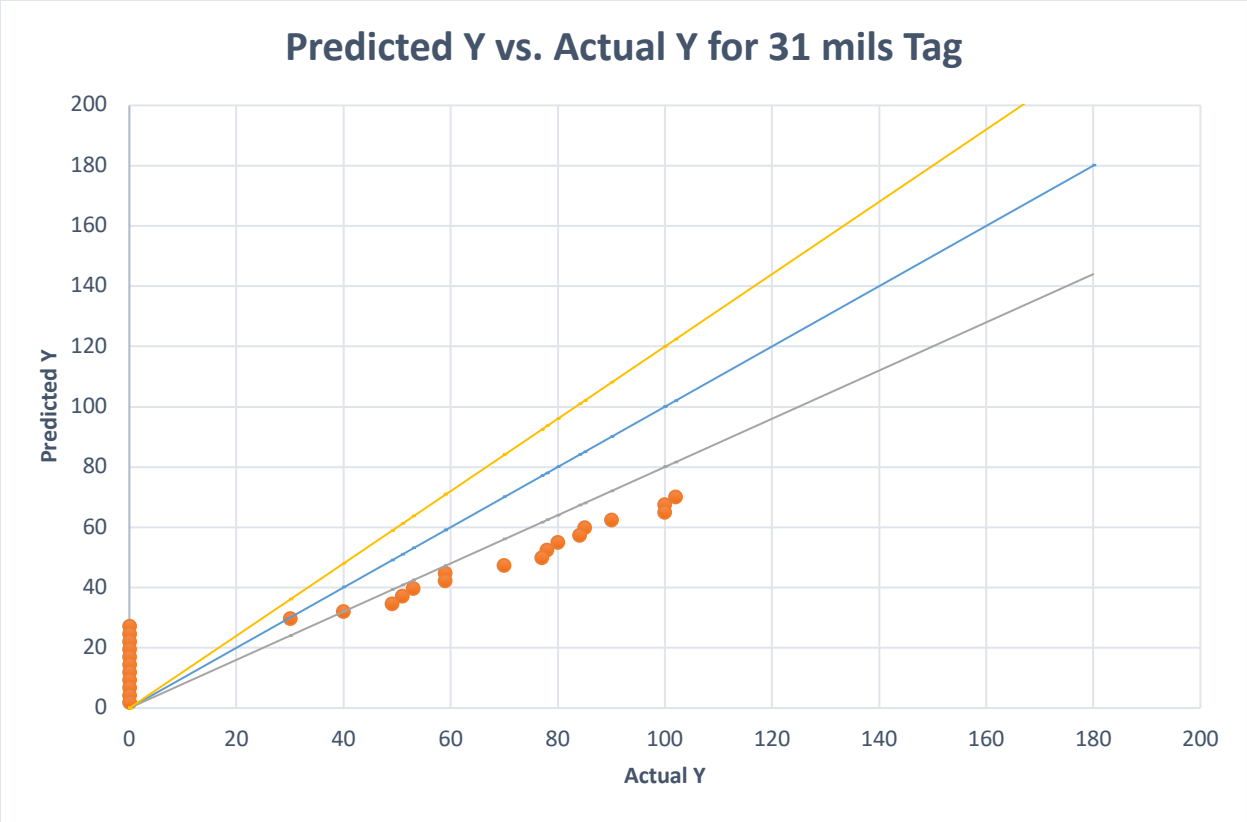


Figure 4-13 Predicted Y (Read Rate) vs. Actual Y (Read Rate) for 31 mils Tag

Figure 4-13 illustrates the Predicted Y vs. Actual Y for the 31 mils Tag. In this figure, we can see several outliers which is present in the data. The majority of the data lies outside the error bands, although there are data points which are on the error band, as well. These outliers could be due to a defective RFID tag and/or human error. This error band shown is +/- 20%. It can be said that the readings of this tag had several errors which led to this result.

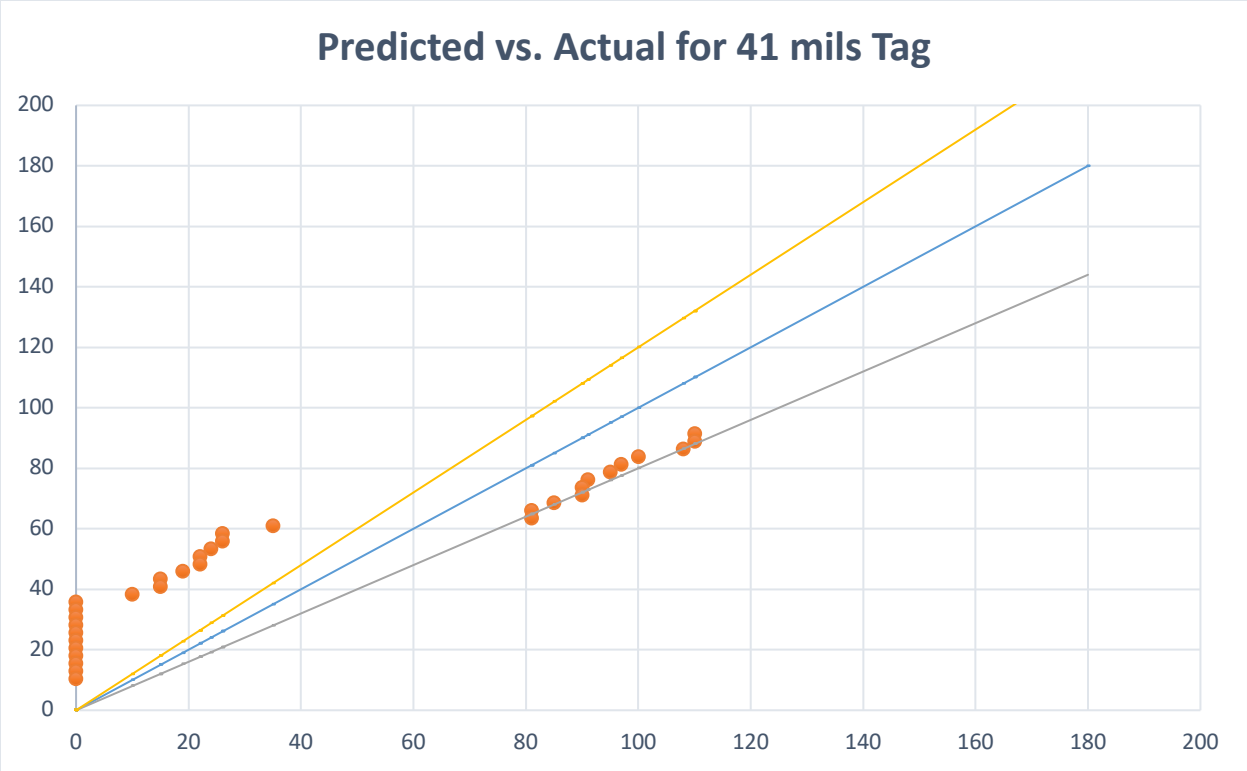


Figure 4-14 Predicted Y (Read Rate) vs. Actual Y (Read Rate) for 41 mils Tag

In Figure 4-14, the Predicted Y (Read Rate) vs. Actual Y (Read Rate) for the 41 mils Tag is represented. The points lie within the error band, however, there are outliers due to human error.

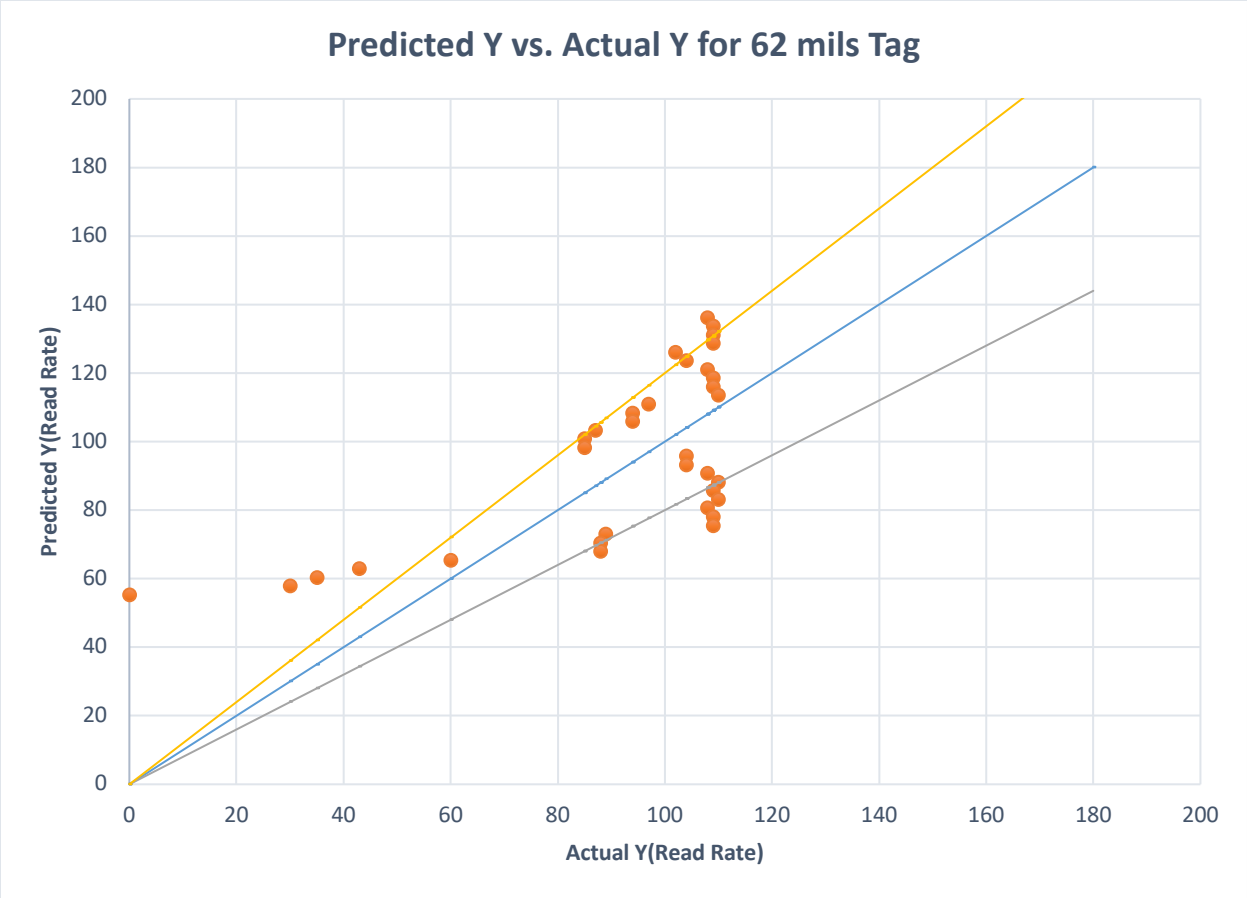


Figure 4-15 Predicted Y (Read Rate) vs. Actual Y (Read Rate) for 62 mils Tag

Figure 4-15 represents the Predicted Y (Read Rate) vs. Actual Y (Read Rate) for the 62 mils Tag. The majority of the points are within the bounds of the error band.

Comparing Figures 4-12-4-15 it can be seen that the best representation of the tags for the predicted vs actual is the 63 mils tag, which has been found to be the optimal tag.

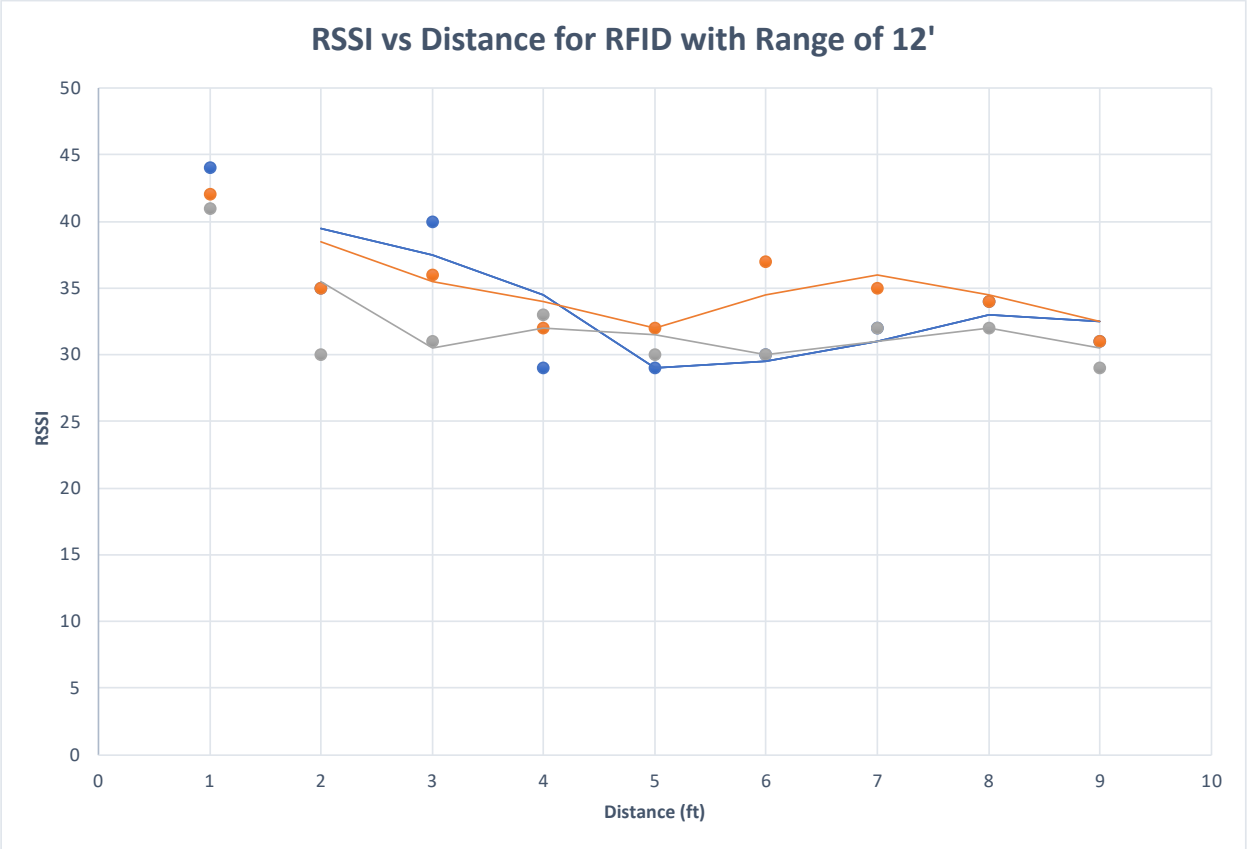


Figure 4-16 Repeatability Test for RSSI vs Distance for Range 12'

Figure 4-16 represents the repeatability test for RSSI vs Distance for RFID Tag with range of 12 feet, demonstrating the three trial runs that were conducted during the experiment. As seen in the graph, the results, or data points of the three tests are in proximity of one another. Based on the given RFID tag read range of 12 feet, during the experiment, the tag was only able to detect 9 feet.

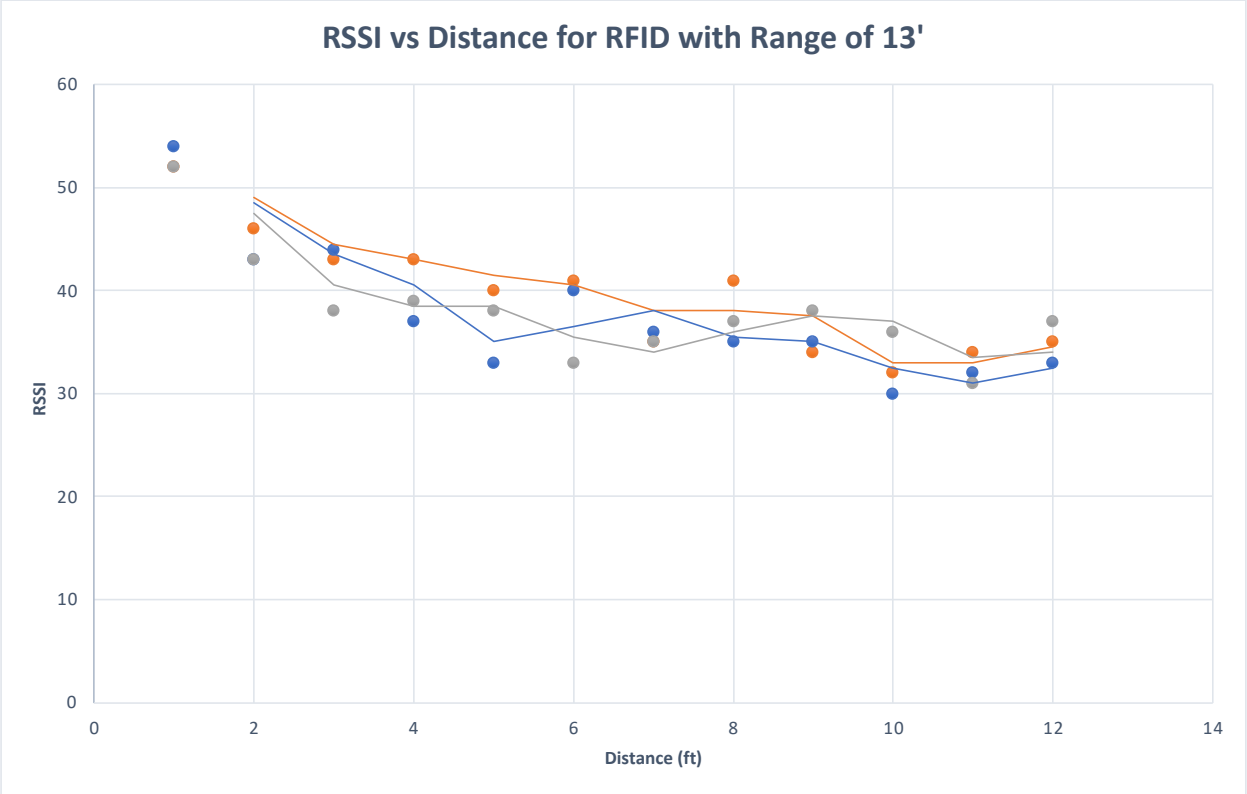


Figure 4-17 Repeatability Test for RSSI vs Distance for Range 13'

Figure 4-17 represents the repeatability test for RSSI vs Distance for RFID Tag with range of 13 feet. For this experiment, the range which the tag was able to detect went up to 12 feet. The actual read range of this tag was closer to the given read range, provided by specifications. As illustrated in the graph, the points are in proximity of each other, which results in a higher accuracy and precision

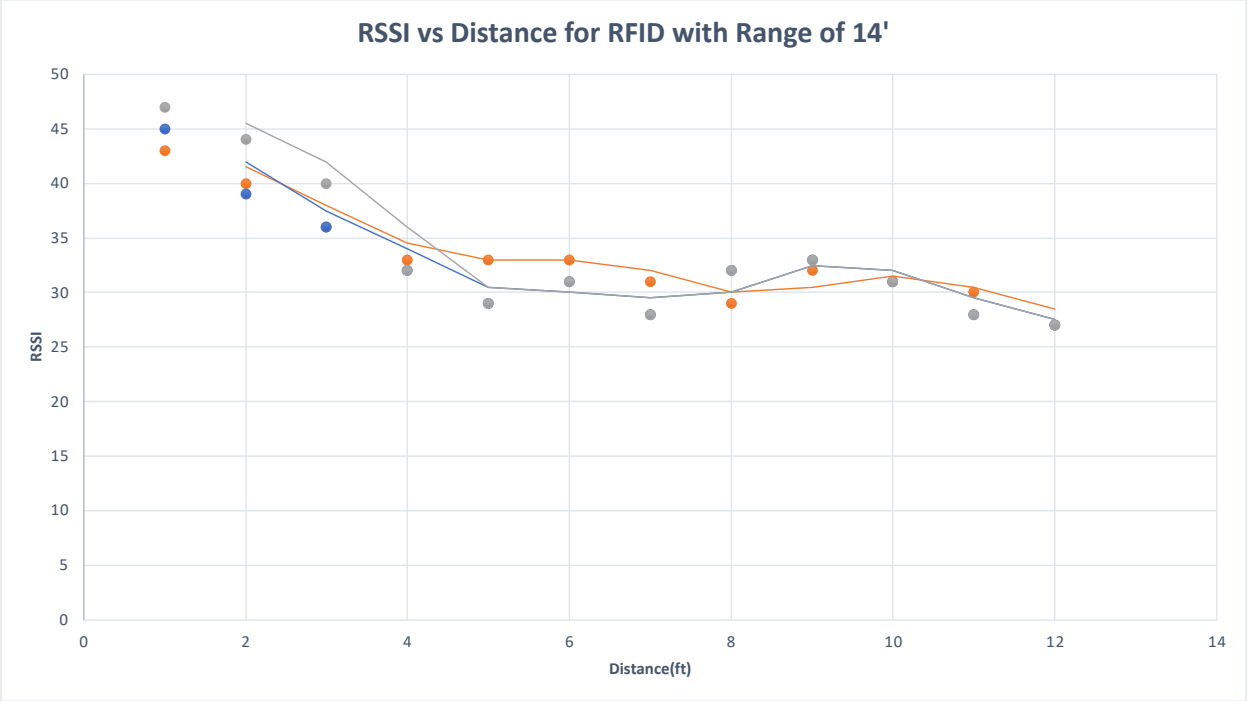


Figure 4-18 Repeatability Test for RSSI vs Distance for Range 14'

Figure 4-18 represents the repeatability test for RSSI vs Distance for RFID tag for range of 14 feet. With this specific RFID tag, the read range provided was up to 14 feet, whereas the actual read range, based on our experiment was no more than 12 feet. For this experiment, we see that there is variation between the three experiment runs.

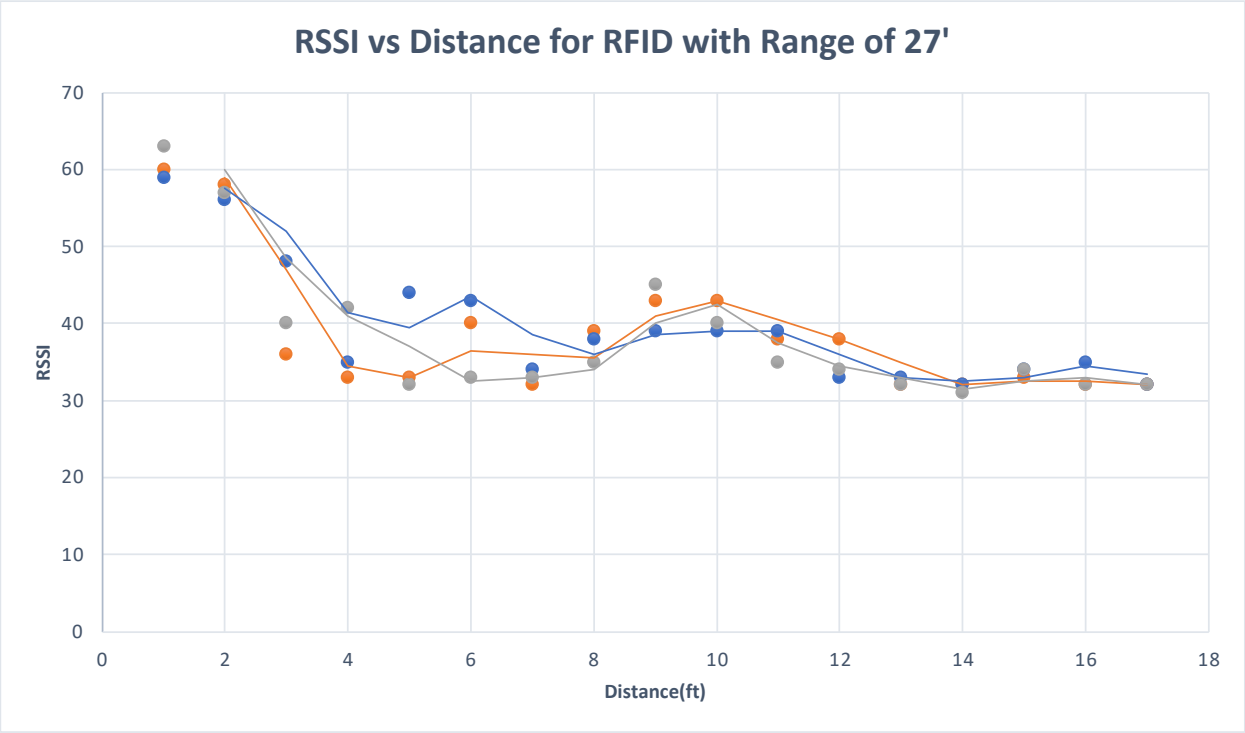


Figure 4-19 Repeatability Test RSSI vs Distance for Range 27'

Figure 4-19 represents the repeatability test for RSSI vs Distance for RFID tag for range of 27 feet. In regards to the provided read range and capability of detection, it was given up to 27 feet, however, during experimental process, the maximum that the tag was able to detect up to 17 feet.

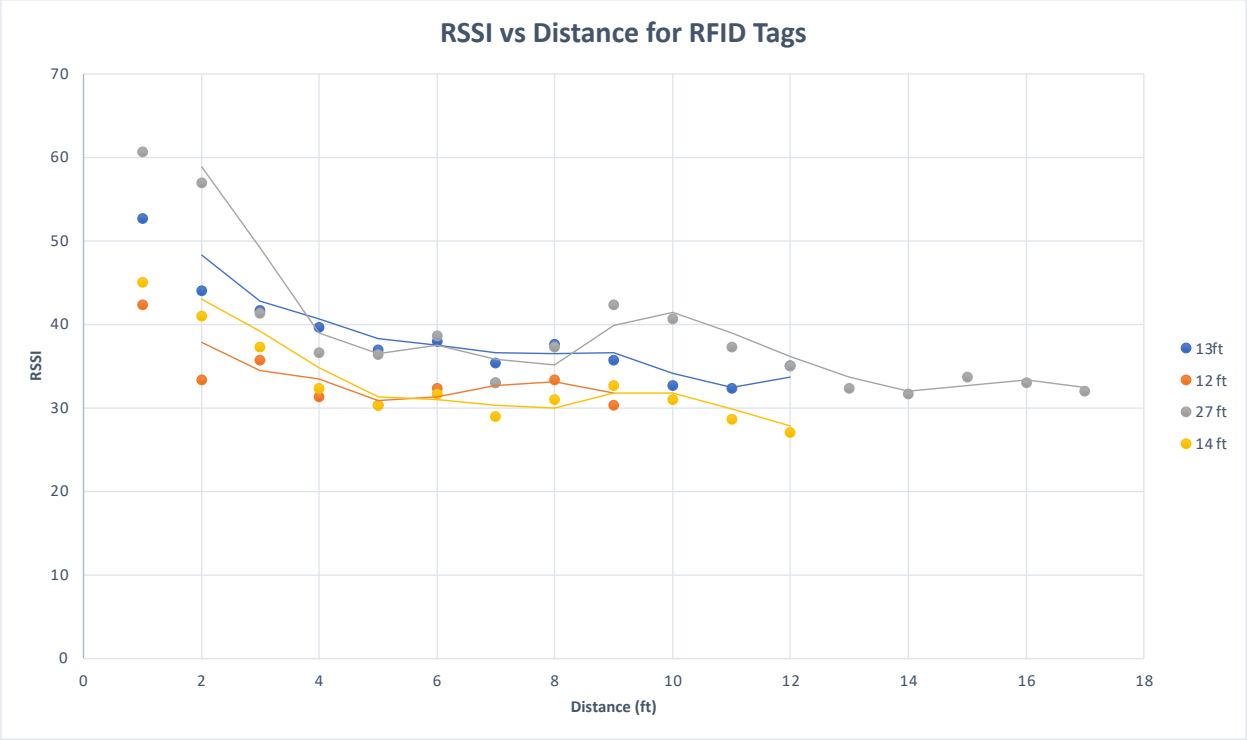


Figure 4-20 RSSI vs Distance for RFID Tags

Figure 4-20 represents the RSSI vs Distance for the RFID tags which were tested for the averages of the RFID tags. In this graph, the averages of each RFID tag specification for the ranges of 12, 13, 14 and 27 feet were used to show the comparative results. The average of the three experiment trial runs for each tag was taken, then plotted against distance to show how close the readings were between the various RFID tags. As seen and demonstrated in the graph, the points are close in proximity, exhibiting that even though, there are some slight variation and error, it is not too significant. The results above that the RFID tag with the range of 27 ft has the capability of reading longest distance of 17 ft. In addition, this tag has the capability of reading the maximum read signal at 1ft. which is equal to 61 and 2 ft which is equal to 58. It should be noticed that the experimental tests follow the expectations of different RFID tag specification ranges. For example, at 2 feet, the RSSI decreases for tag specifications of 27ft, 13ft, 12ft and 14ft range, with the exception of the

variation between the 13 feet and 14 feet which is due to closeness in their specifications. However, in general, as the tag range increases, the RSSI value increases. Thus, it is recommended that for these series of RFID tag type and specifications, the 27 feet be employed which is capable of reading up to 17 feet.

4.7 Predicted Y vs Actual Y Linear Regression Approach 2

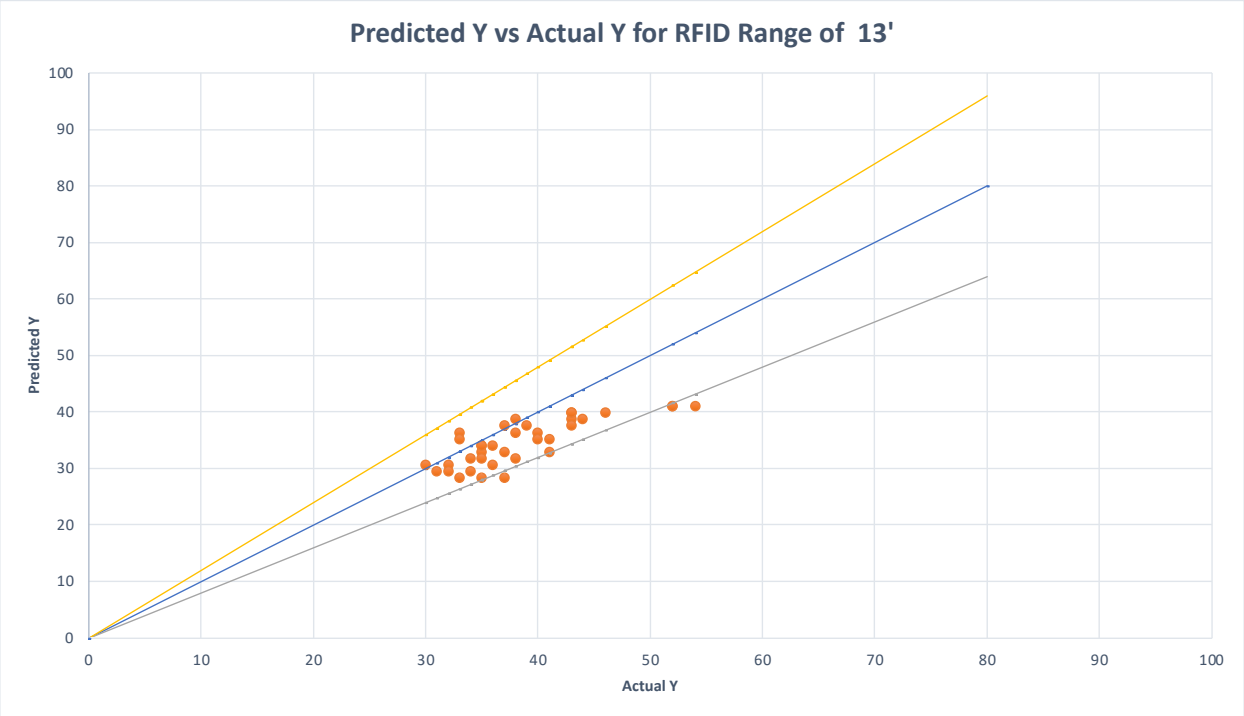


Figure 4-21 Predicted Y vs Actual Y for Range of 13 ft.

Figure 4-21 shows the predicted values of RSSI vs the experimentally observed values(actual) of RSSI for RFID with Range of 13 ft. A 45-degree line is drawn which shows a one-to-one relationship between the predicted value vs the actual value of RSSI for RFID with a range of 13 ft. In addition, a +/- 20% line is drawn to identify the error bands for the regression analysis for the predicted value of RSSI. As shown in the figure, the majority of the points are within +/-20% with several predicted values on the 45-degree line which means the regression analysis is capable of predicting the RSSI values accurately while being in the error band. It should be noted that only two of the predicted values are outside the -20% band, but very close to the band.

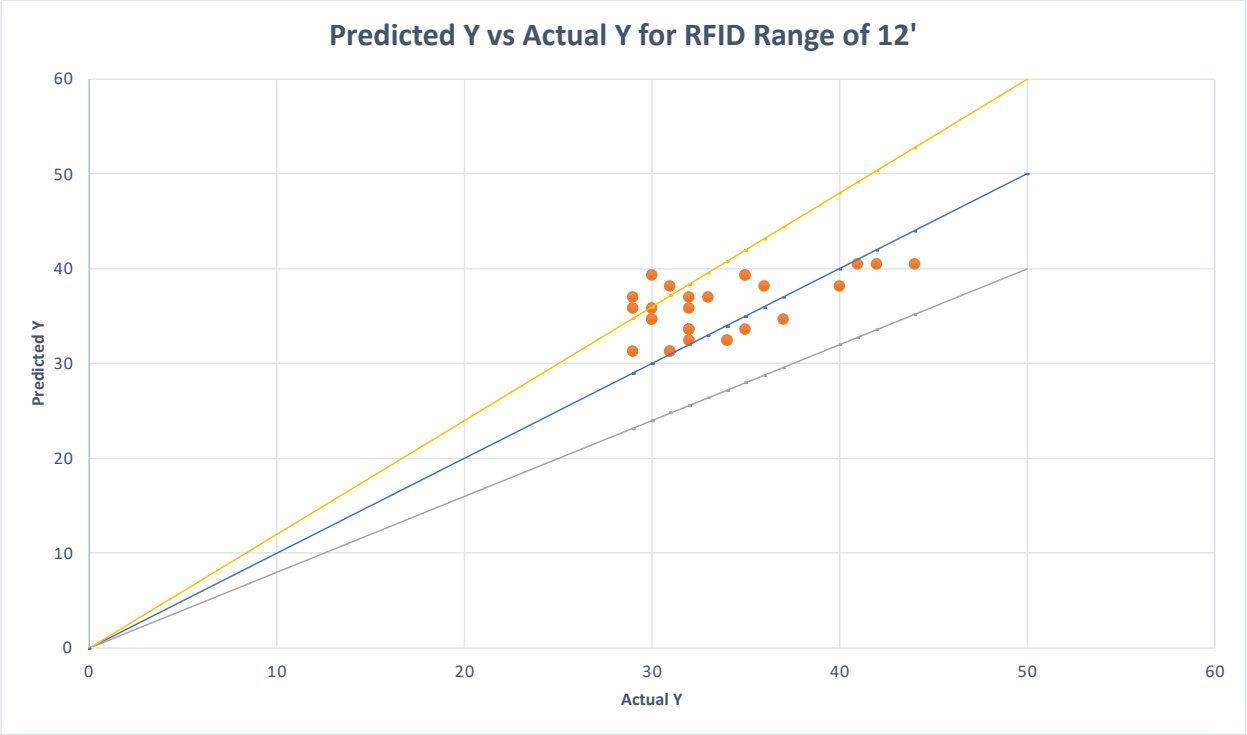


Figure 4-22 Predicted Y vs Actual Y for Range with 12'

Figure 4-22 shows the predicted values of RSSI vs the experimentally observed values(actual) of RSSI for RFID with Range of 12 ft. A 45-degree line is drawn which shows a one-to-one relationship between the predicted value vs the actual value of RSSI for RFID with a range of 12 ft. In addition, a +/- 20% line is drawn to identify the error bands for the regression analysis for the predicted value of RSSI. As shown in the figure, the majority of the points are within +/-20% with several predicted values on the 45-degree line which means the regression analysis is capable of predicting the RSSI values accurately while being in the error band. It should be noted that only four of the predicted values are outside the +20% band, but very close to the band.

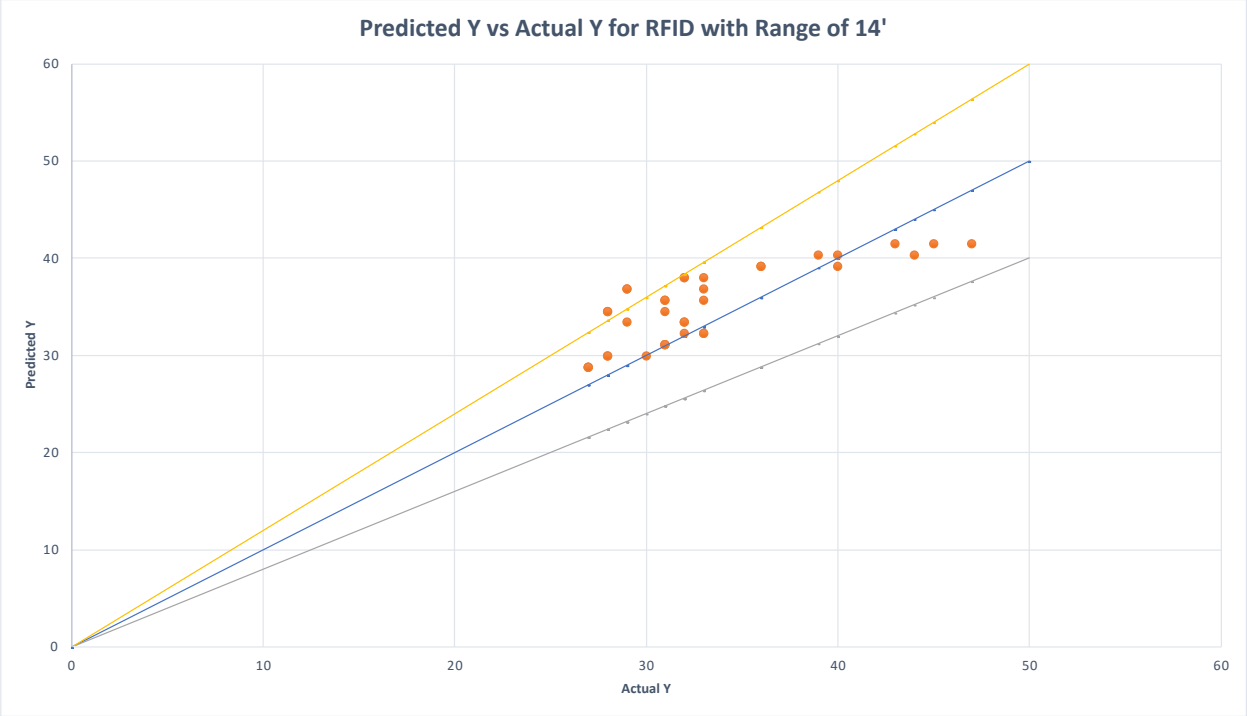


Figure 4-23 Predicted Y vs Actual Y for RFID with Range of 14 ft.

Figure 4-23 shows the predicted values of RSSI vs the experimentally observed values(actual) of RSSI for RFID with Range of 14 ft. A 45-degree line is drawn which shows a one-to-one relationship between the predicted value vs the actual value of RSSI for RFID with a range of 14 ft. In addition, a +/- 20% line is drawn to identify the error bands for the regression analysis for the predicted value of RSSI. As shown in the figure, the majority of the points are within +/-20% with several predicted values on the 45-degree line which means the regression analysis is capable of predicting the RSSI values accurately while being in the error band. It should be noted that only two of the predicted values are outside the +20% band, but very close to the band.

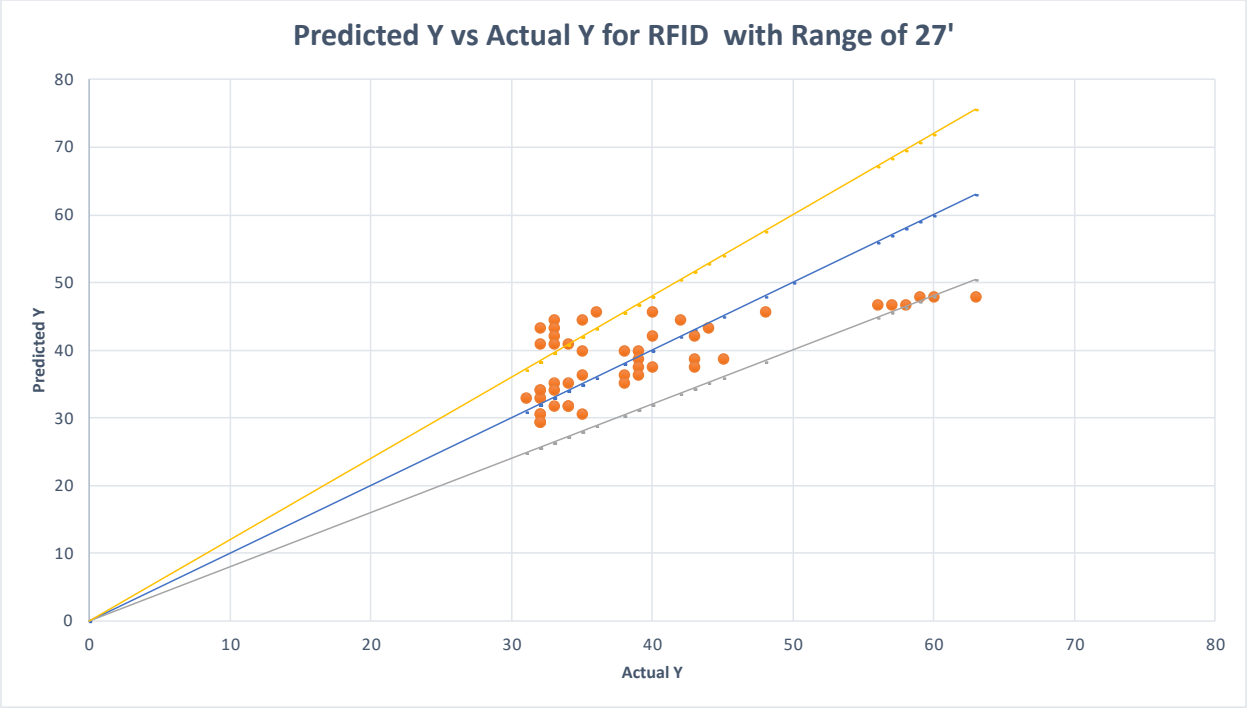


Figure 4-24 Predicted Y vs Actual Y for RFID with Range of 27'

Figure 4-24 shows the predicted values of RSSI vs the experimentally observed values(actual) of RSSI for RFID with Range of 27 ft. A 45-degree line is drawn which shows a one-to-one relationship between the predicted value vs the actual value of RSSI for RFID with a range of 27 ft. In addition, a +/- 20% line is drawn to identify the error bands for the regression analysis for the predicted value of RSSI. As shown in the figure, the majority of the points are within +/-20% with several predicted values on the 45-degree line which means the regression analysis is capable of predicting the RSSI values accurately while being in the error band. It should be noted that only nine of the predicted values are outside the +20% band and one point below the -20% error band, but very close to the line. It should be noted that for the RFID tag with the range of 27 ft, more data points were obtained as compared with those for ranges of 12, 13, and 14 ft. Therefore, the number of data points outside of 20% is more than that for tags with other ranges. However, interestingly, the majority of the points are on or very close to the 45-degree line representing

minimum error between the predicted and actual values of RSSI for RFID tag tested with the range of 27 ft.

4.8 Predicted Y vs Actual Y for Nonlinear Regression Approach 2

4.8.1 Plots

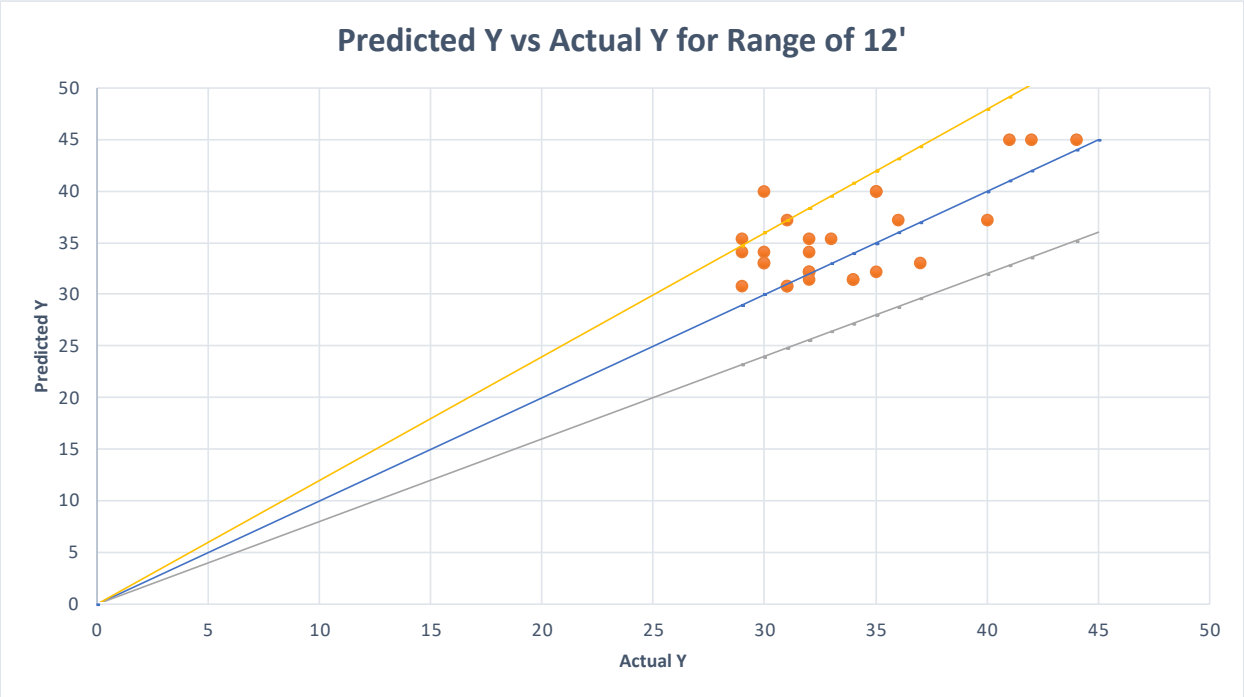


Figure 4-25 Predicted Y vs Actual Y for Range of 12 ft.

Figure 4-25 represents the predicted Y vs Actual Y for Range of 12 ft after the nonlinear regression analysis was performed. A 45-degree line is drawn which shows a one-to-one relationship between the predicted value vs the actual value of RSSI for RFID with a range of 12 ft. In addition, +/- 20% line is drawn to identify the error bands for the regression analysis for the predicted value of RSSI. As shown in the figure, all of the predicted values fall within the +/-20%, except one point outside the band. The values which lie on the 45-degree line means the regression analysis is capable of predicting the RSSI values accurately while being in the error band, except one point being outside

the error band. This is an improvement when using nonlinear regression as compared to linear regression.

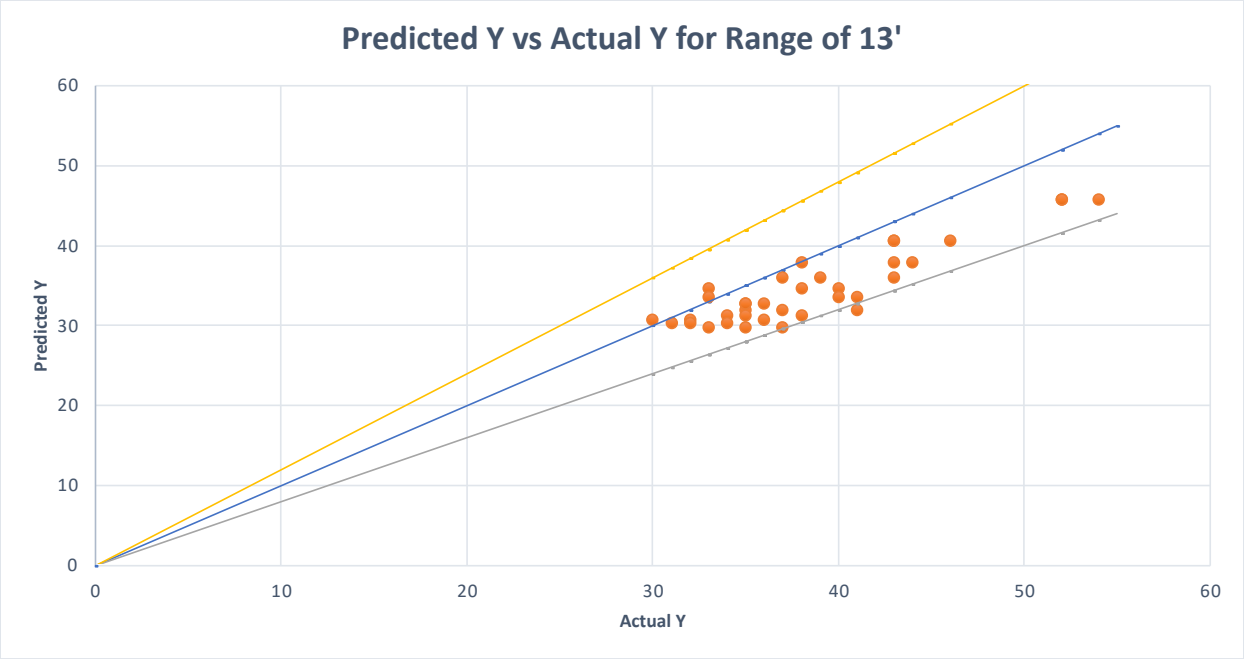


Figure 4-26 Predicted Y vs Actual Y for Range of 13 ft

Figure 4-26 represents the predicted Y vs Actual Y for Range of 13 ft after the nonlinear regression analysis was performed. A 45-degree line is drawn which shows a one-to-one relationship between the predicted value vs the actual value of RSSI for RFID with a range of 13 ft. In addition, a +/-20% line is drawn to identify the error bands for the regression analysis for the predicted value of RSSI. As shown in the figure, all of the predicted values fall within the +/-20% error band which is a significant improvement after performing a nonlinear regression analysis. Indeed, most predicted values are close to the 45-degree line which confirms a one-to-one relationship between the values.

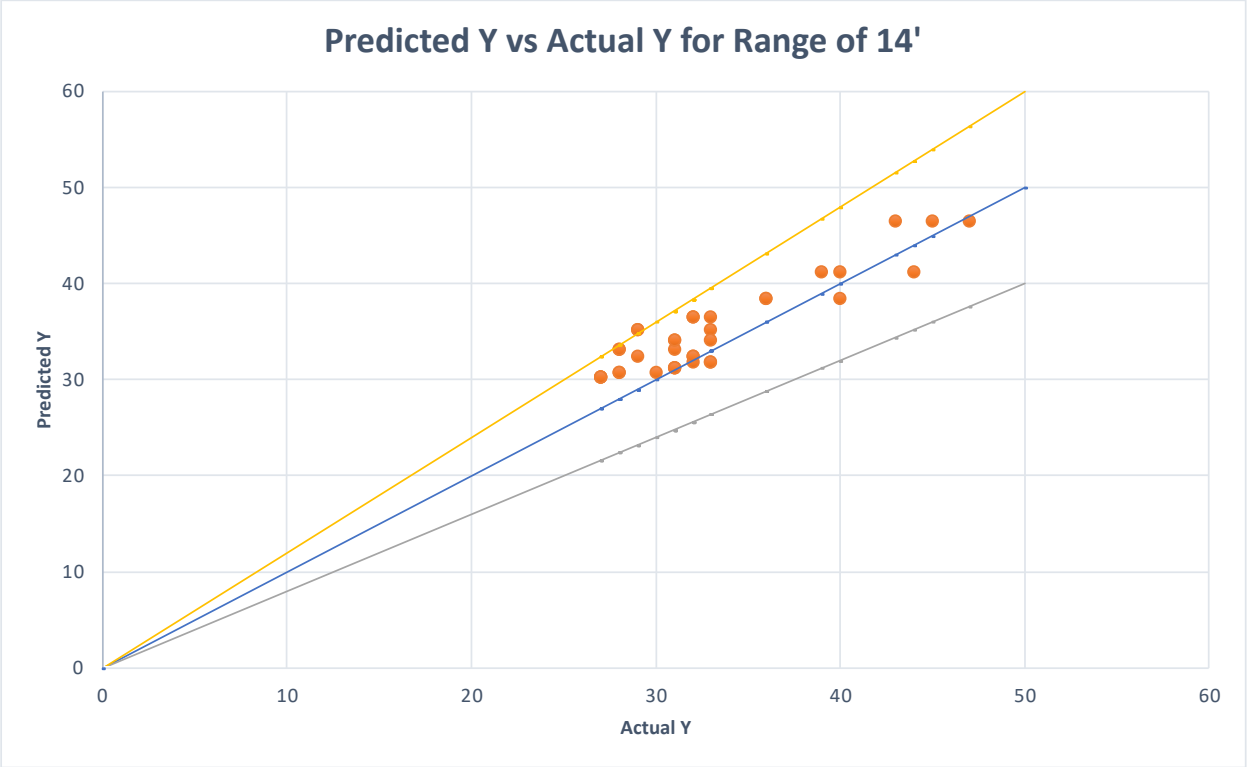


Figure 4-27 Predicted Y vs Actual Y for Range of 14 ft.

Figure 4-27 represents the predicted Y vs actual Y for range of 14 ft. All of the predicted points are within the error band of +/-20%, with multiple points on the 45-degree line. This again shows a significant improvement by using a nonlinear regression approach as compared with linear regression.

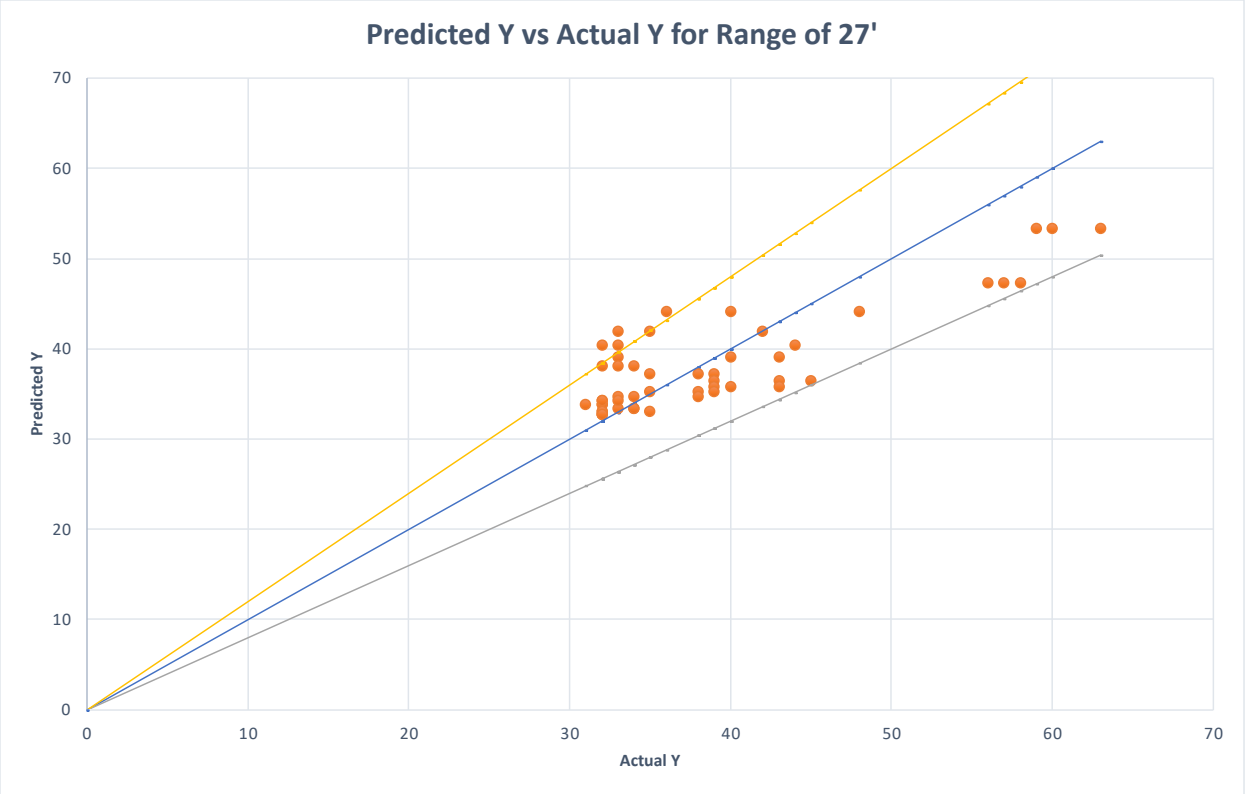


Figure 4-28 Predicted Y vs Actual Y for Range of 27 ft.

Figure 4-28 represents the predicted Y vs. actual Y for range of 27 ft. When using nonlinear regression approach, the results for RFID tag with a range of 27 feet has significantly improved by having all the predicted values of RSSI to be within the bands except three points. The same results for linear regression showed nine points being outside the error band. Indeed, there are multiple predicted values which are on the 45-degree line; this was not the case when linear regression was performed.

4.9 Predicted Y vs Actual Y Linear Regression Approach 3

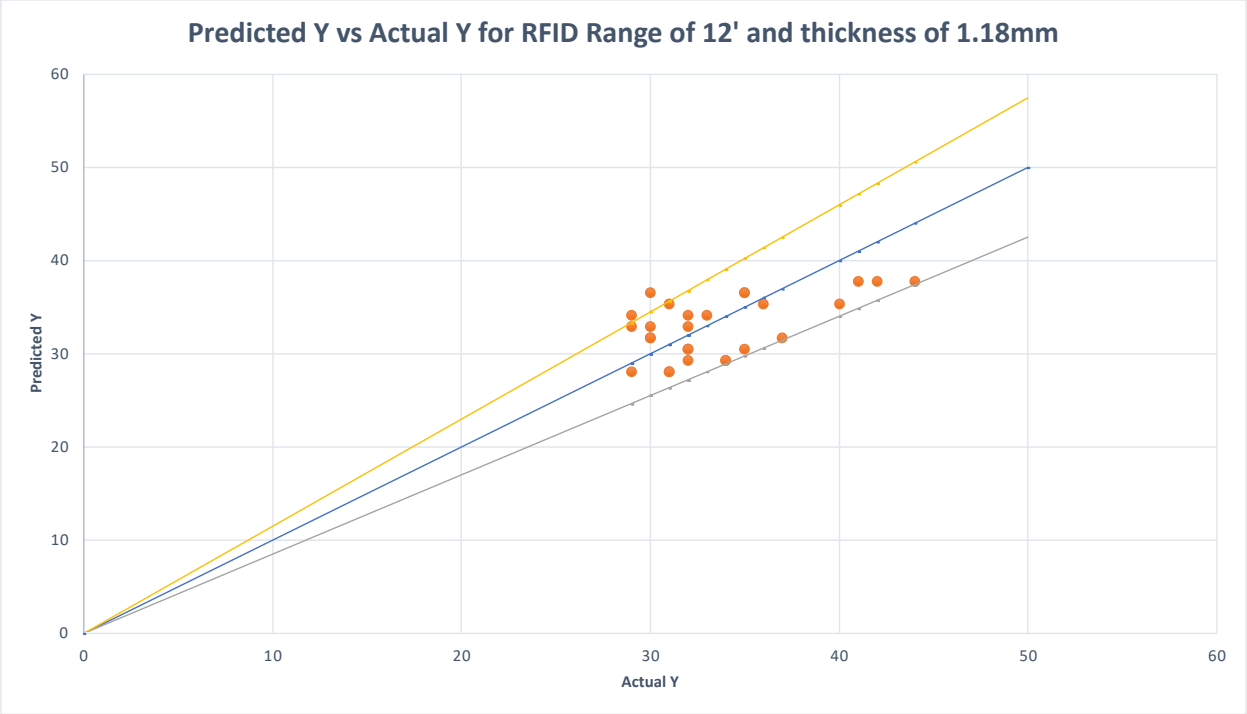


Figure 4-29 Predicted Y vs Actual Y for RFID Tag with Range of 12ft and thickness of 1.18mm

Figure 4-29 represents the predicted Y vs Actual Y for RFID Tag with range of 12ft and a thickness of 1.18mm. The blue line represents a 45-degree which is a one-to-one relationship between the predicted values and the actual values of RSSI for RFID tag with a range of 12 ft and a thickness of 1.18mm. Additionally, a +/-15% line is drawn to identify the error bands for the regression analysis for the predicted value of RSSI. As shown in the graph, all of the predicted points, except two points are within the +/-15% error band, resulting in an improvement, as compared to the previous approaches. It should be noted that the two predicted values which are outside the +15% band are very close to the band.

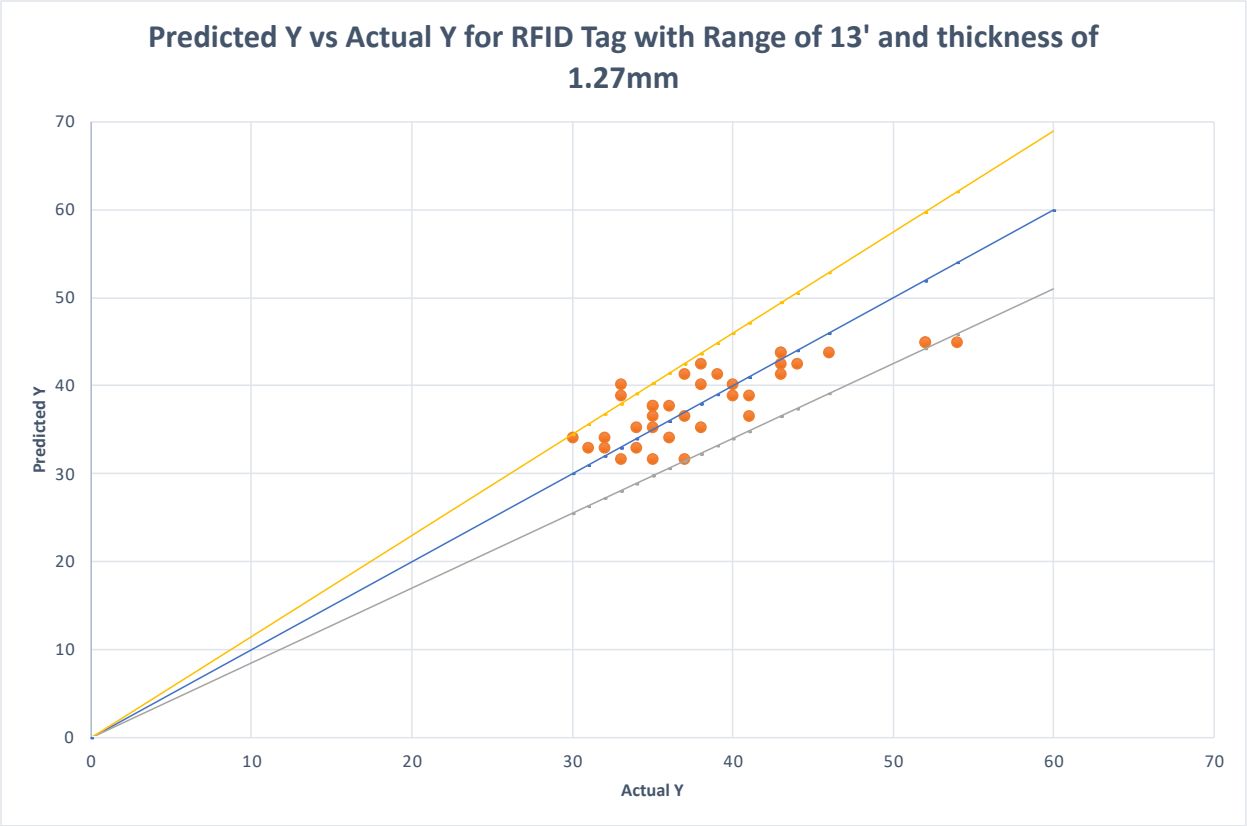


Figure 4-30 Predicted Y vs Actual Y for RFID Tag with Range of 13 ft and thickness of 1.27mm

Figure 4-30 represents the predicted Y vs Actual Y for RFID Tag with range of 13ft and a thickness of 1.27mm. The blue line represents a 45-degree which is a one-to-one relationship between the predicted values and the actual values of RSSI for RFID tag with a range of 13 ft and a thickness of 1.27 mm. In addition, a +/-15% line is shown to identify the error bands for the analysis for the predicted value of RSSI. As shown in the graph, all of the predicted points, except three points are within the +/-15% error band. It should be noted that the three data points which are outside the +/-15% band are very close to the band.

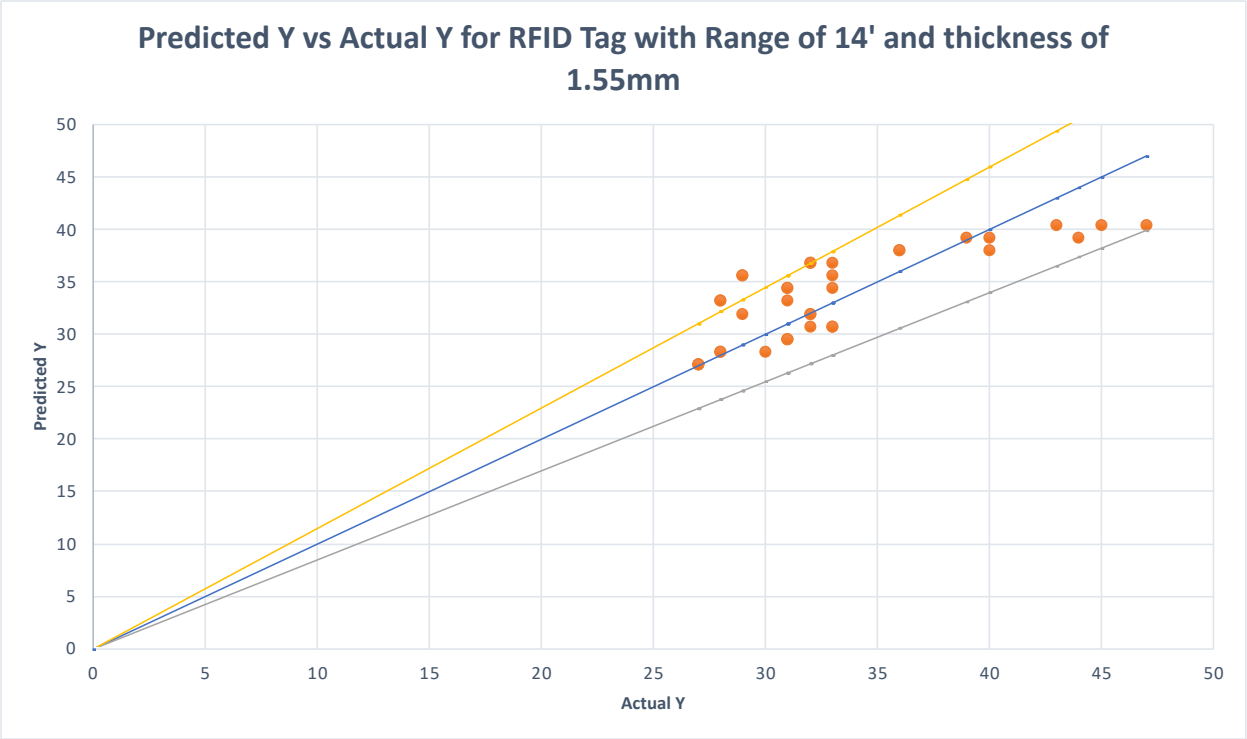


Figure 4-31 Predicted Y vs Actual Y for RFID Tag with Range of 14' and thickness of 1.55mm

Figure 4-31 represents the predicted Y vs Actual Y for RFID Tag with range of 14ft and a thickness of 1.55mm. A blue is drawn to represent the 45-degree, and a one-to-one relationship between the predicted values and the actual values of RSSI for RFID tag with a range of 14 ft and a thickness of 1.55 mm. In addition, the yellow and gray lines are drawn to show +/-15% error bands for the analysis for the predicted value of RSSI. It should be noted that all the predicted values except two points lie within the +15% band. This has shown an improvement to the results, as compared to the previous approaches.

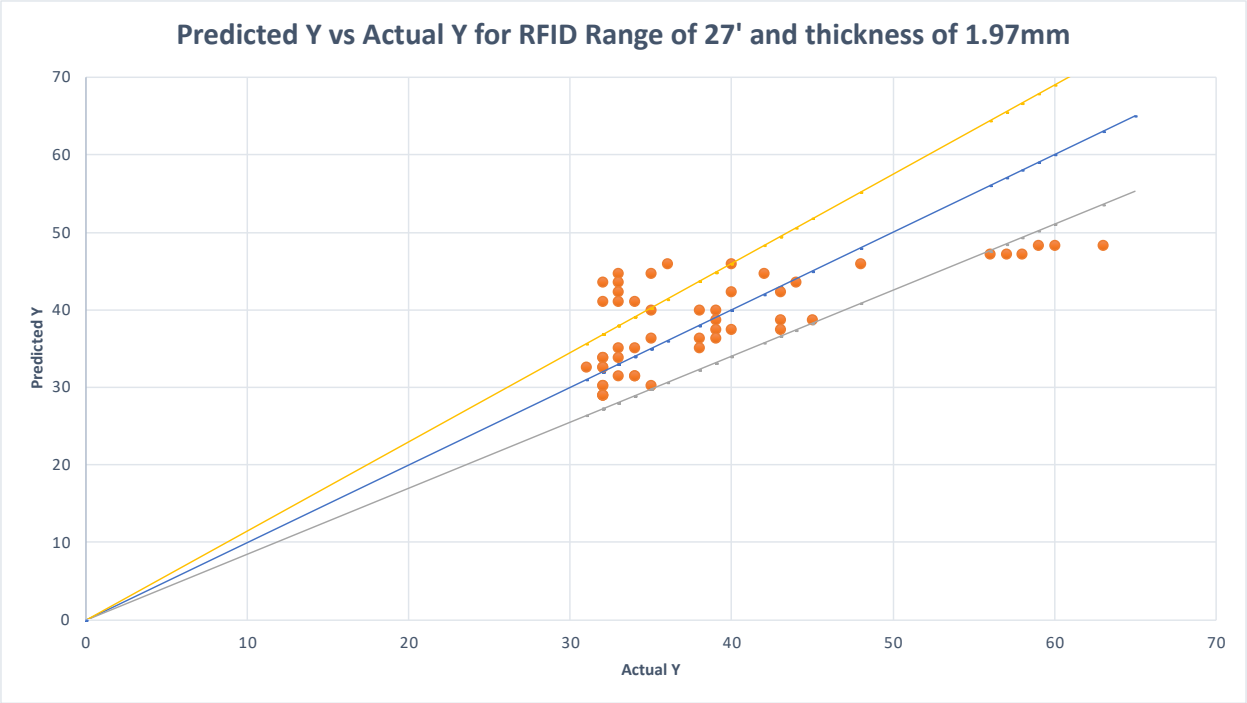


Figure 4-32 Predicted Y vs Actual Y for RFID Range of 27' and thickness of 1.97mm

Figure 4-32 represents the predicted Y vs Actual Y for RFID Tag with range of 27 ft and a thickness of 1.97mm. A one-to-one relationship is demonstrated by the blue line, between the predicted values and the actual values of RSSI for RFID tag with a range of 27 ft and a thickness of 1.97 mm. The +/-15% error bands for the analysis of the predicted value of RSSI are also represented in the figure. It can be that the majority of the predicted values fall within the error bands.

4.10 Predicted Y vs Actual Y Nonlinear Regression Approach 3

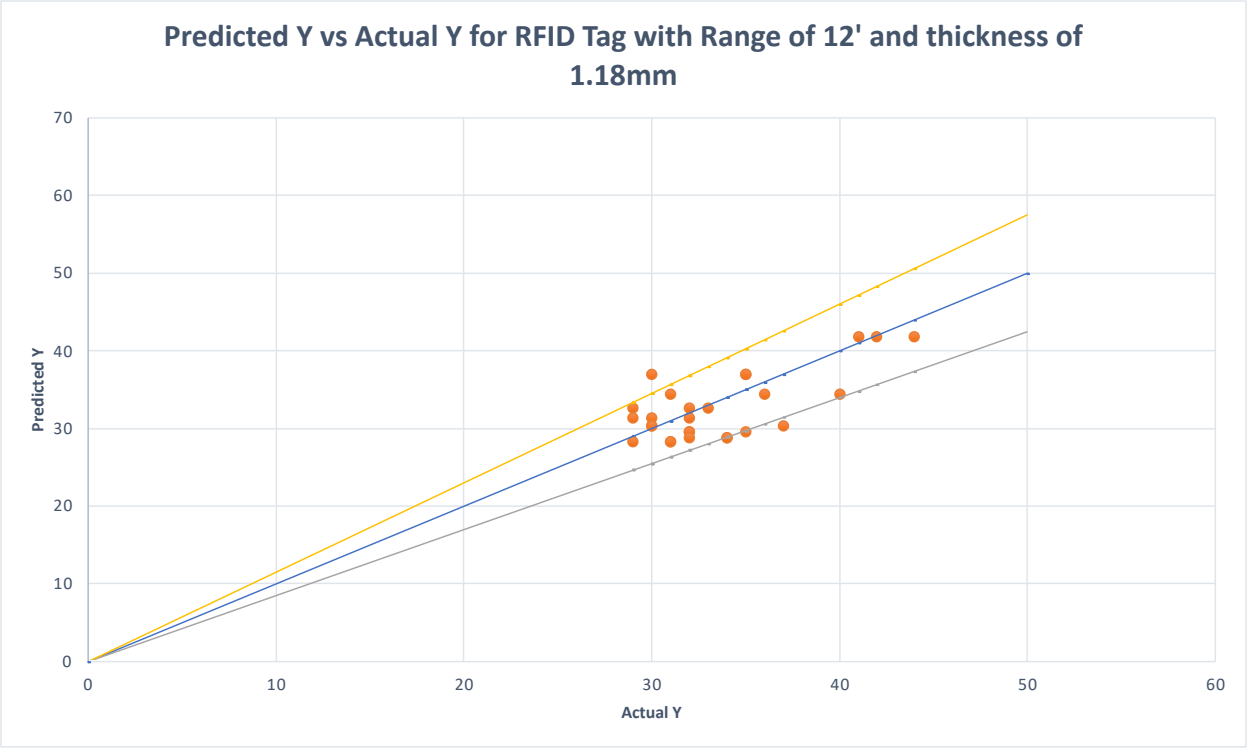


Figure 4-33 Predicted Y vs Actual Y for RFID Range of 12' and thickness of 1.18mm

Figure 4-33 represents the predicted Y vs Actual Y for RFID Tag Range of 12ft and a thickness of 1.18mm after performing the nonlinear regression analysis. A 45-degree line is drawn which shows a one-to-one relationship between the predicted value vs the actual value of RSSI for RFID with a range of 12 ft. In addition, a +/-15% line is drawn to identify the error bands for the regression analysis for the predicted value of RSSI. As demonstrated in the figure, all of the predicted values, except one fall within the +/-15% error band which is a significant improvement after performing a nonlinear regression analysis. Indeed, most predicted values are close to the 45-degree line which confirms a one-to-one relationship between the values.

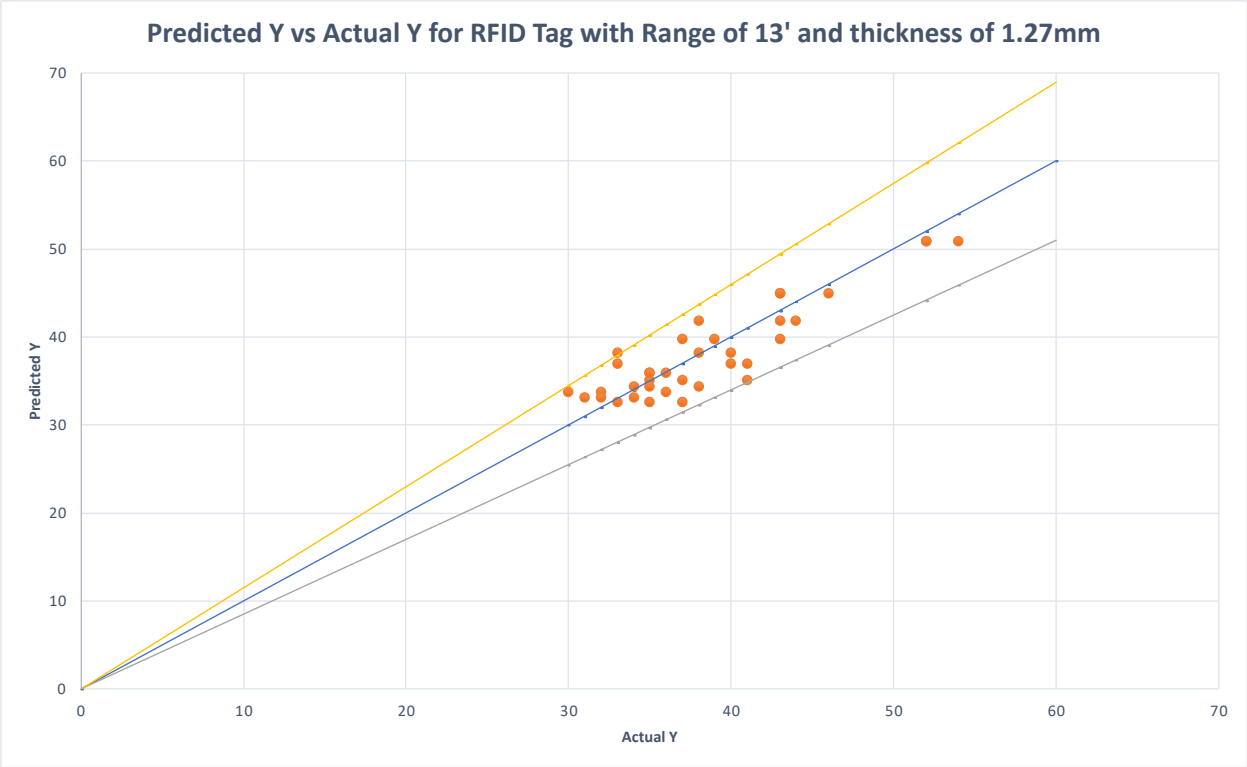


Figure 4-34 Predicted Y vs Actual Y for RFID Tag with Range of 13 ft and thickness of 1.27mm

Figure 4-34 represents the predicted Y vs actual Y for RFID Tag with Range of 13 ft and a thickness of 1.27mm after performing a nonlinear regression analysis. The blue line drawn is illustrated to show the 45-degree, showing a one-to-one relationship between the predicted and actual values of the RSSI for the tag with a range of 13 ft and thickness of 1.27mm. Furthermore, error bands of +/-15% is drawn for the analysis for the predicted value of RSSI. As demonstrated in the figure, all of the predicted values lie within the +/-15% error band which is a substantial improvement after performing a nonlinear regression analysis. Indeed, all of the predicted values are close to the 45-degree line which confirms a one-to-one relationship between the values.

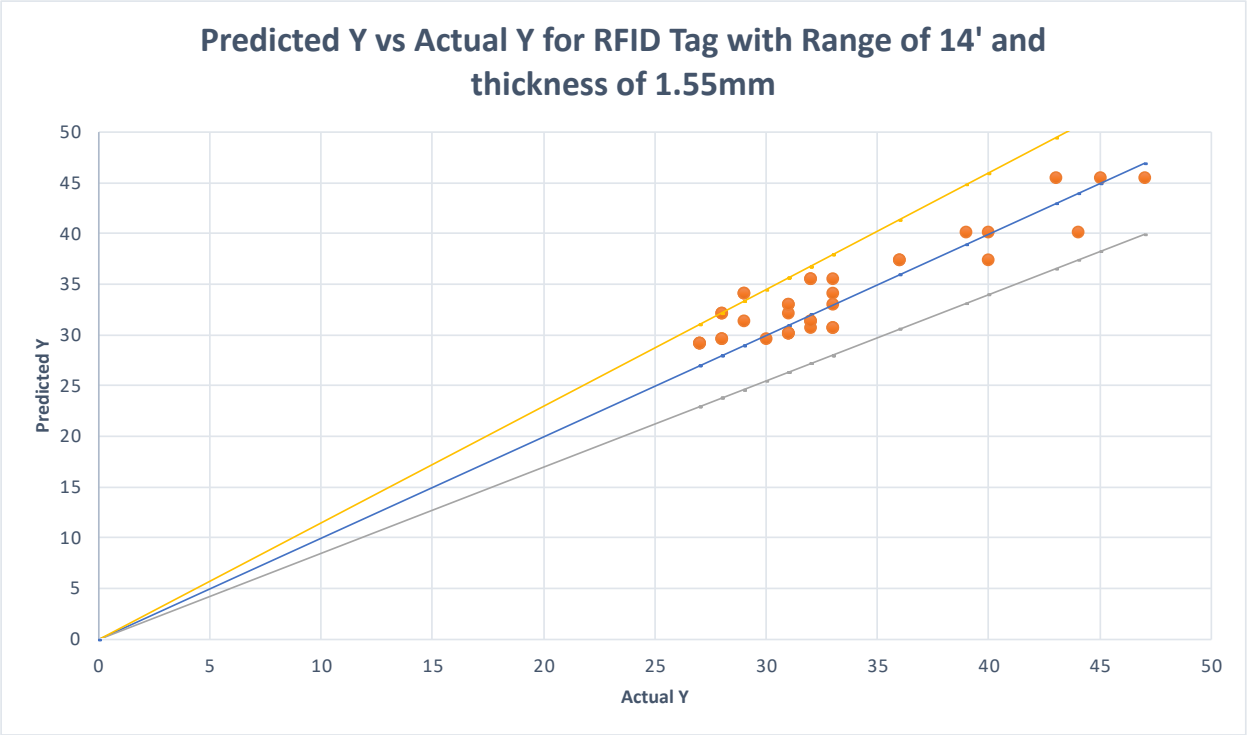


Figure 4-35 Predicted Y vs Actual Y for RFID Tag with Range of 14 ft and thickness of 1.55mm

Figure 4-35 represents the predicted Y vs actual Y for RFID Tag with Range of 14 ft and a thickness of 1.55mm after nonlinear regression analysis. The 45-degree line shows a one-to-one relationship between the predicted vs actual values of the RSSI for the RFID with a range of 14 ft and a thickness of 1.55mm. The error bands of +/-15% are drawn, represented by the yellow and gray lines for the predicted value of RSSI. It should be noted from the figure that all of the values are within the bands which shows a significant improvement, as compared Figure 4-37, showing two points being outside the error band. The predicted values are close to the 45-degree line, confirming a one-to-one relationship between the values.

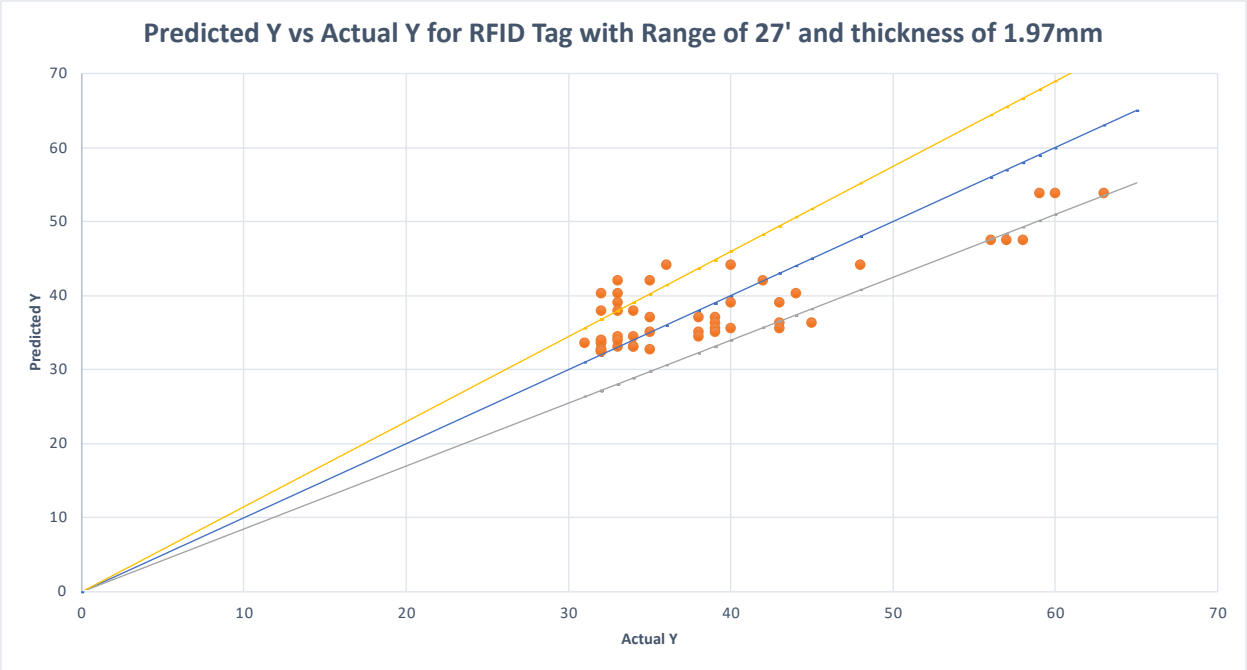


Figure 4-36 Predicted Y vs Actual Y for RFID Tag with Range of 27 ft and thickness of 1.97mm

Figure 4-36 represents the predicted Y vs actual Y for RFID Tag with Range of 27 ft and thickness of 1.97mm after nonlinear regression analysis. As represented in the figure, the number of predicted points outside of the +/-15% error bands have decreased after performing a nonlinear regression analysis approach. The majority of the values are very close to the 45-degree line which shows a one-to-one relationship. This is an improvement from the results provided in Figure 4-38.

4.11 Accuracy and results of AI model

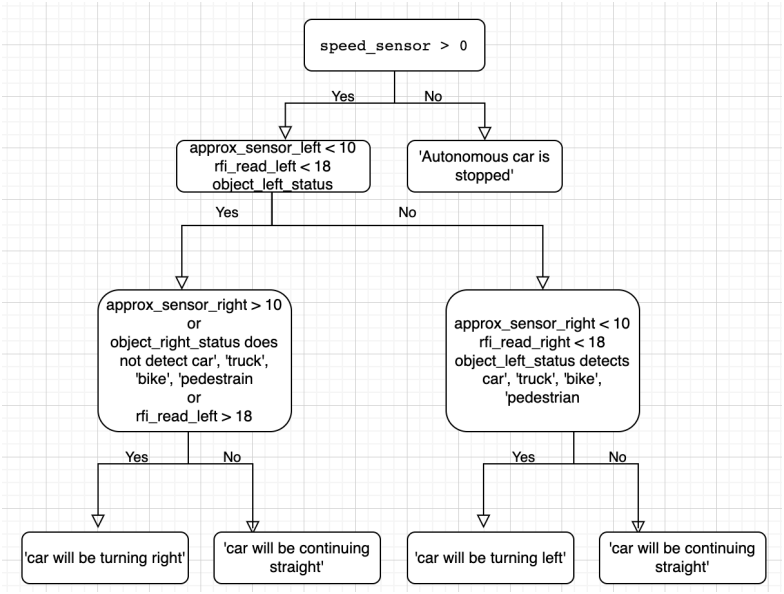


Figure 4-37 Flowchart of AI Model for Optimized Route

Figure 4-37 represents the flowchart of the AI Model utilized for the vehicle when in a disastrous situation, the vehicle has two situations, requiring a decision to be made to maneuver the current scenario. For example, if there is an obstacle or object to the left of vehicle, the vehicle will turn right, and vice versa.

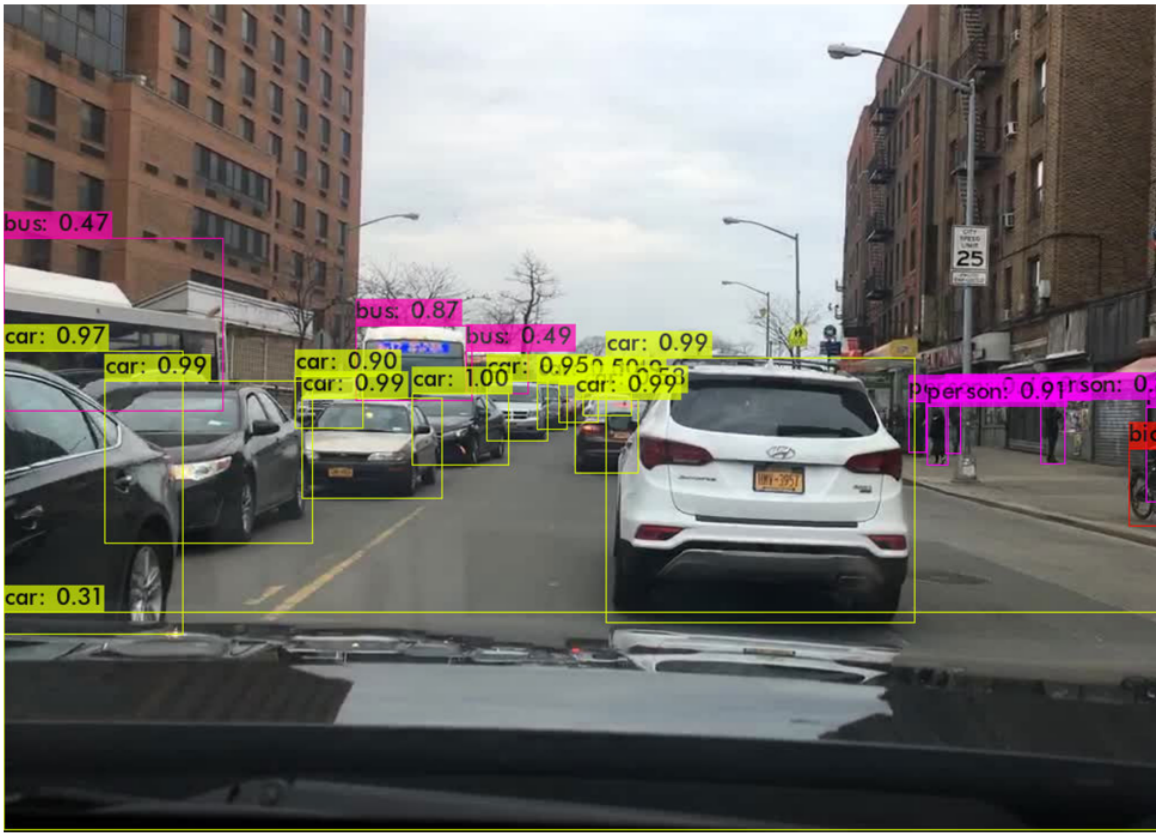


Figure 4-38 Object Detection of Yolov4 Model Run 1

Figure 4-38 represents the object detection of Yolov4 Model Run 1. In the figure, it is shown that the accuracy of the model is above 70%. For the detection with accuracy of 47%, one of the possibilities for the lower accuracy and precision rate is the vision range of the camera.

Figure 4-39 Object Detection of Yolov4 Model Run 2

Figure 4-39 represents the object detection of Yolov4 Model Run 2. The object detection accuracy rate is at a much higher percentage, as shown in the figure. In the figure, it is shown that the model has detected car and trucks at a level of above 70% with most of the detection above 80%.

Figure 4-40 Object Detection of Yolov4 Model Run 3

Figure 4-40 represents the object detection of Yolov4 Model Run 3. The object detection accuracy are at the following levels: 0.93, 0.94, 0.75, and 0.42; as shown in the figure. The car which is detected at 0.47, or 47% is lower as compared to the others. One possibility is the vision range of the camera.

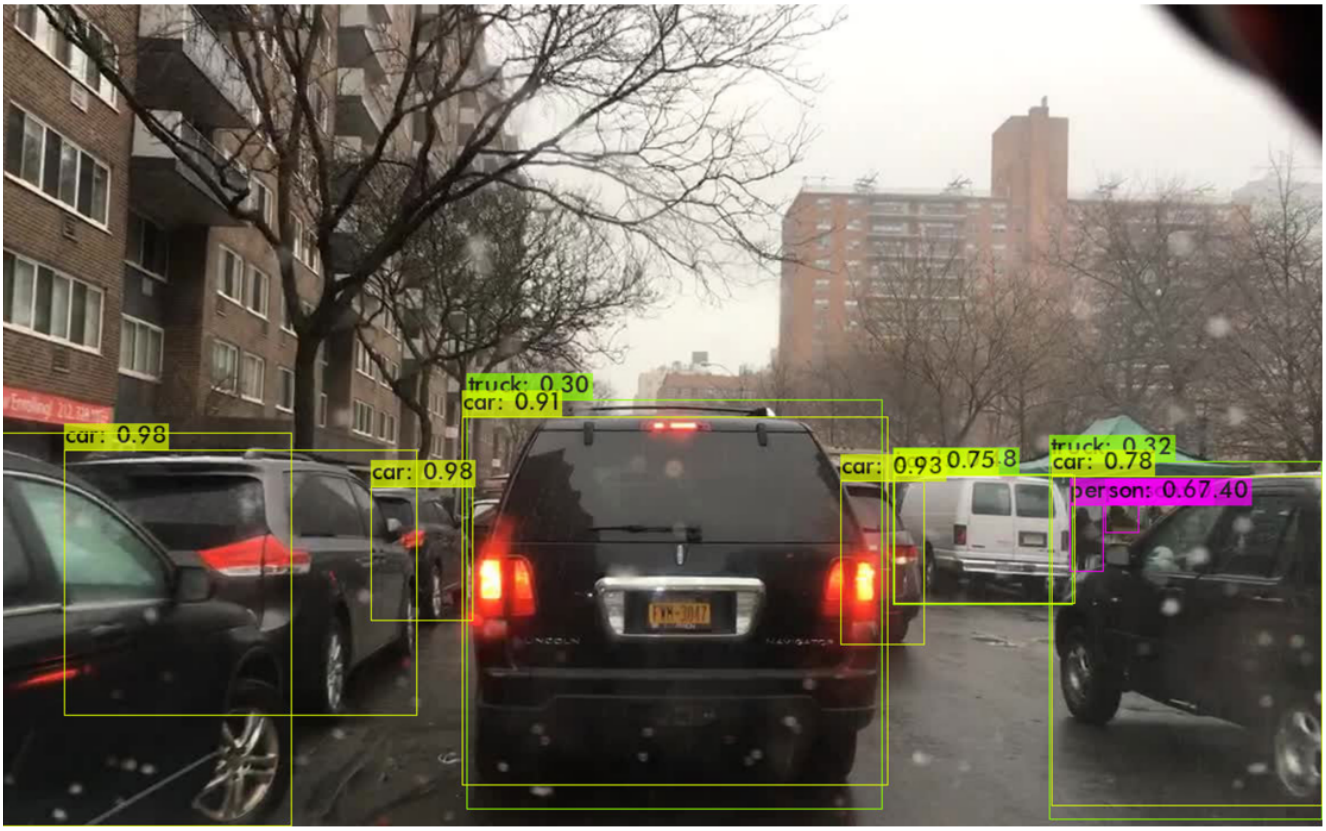


Figure 4-41 Object Detection of Yolov4 Model Run 4

Figure 4-41 represents the object detection of Yolov4 Model Run 4. In the figure, it is shown that the detection is mostly at levels above 0.75, or 75%. It should be noted there is fog and light rain, but the detection of the objects is accurately detecting with high precision. It is seen from the figure that the detection of some objects are at levels of 0.30, 0.32, and 0.67. One possibility of this is the vision range of the camera as well as the inclement weather conditions.

4.12 Economic Analysis

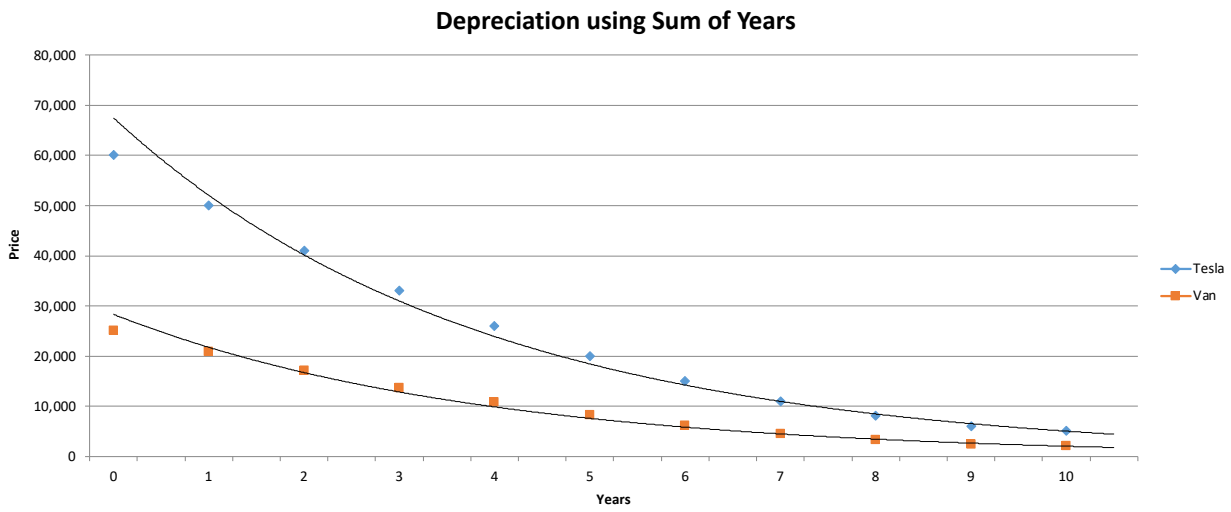


Figure 4-42 Depreciation using Sum of Years

Figure 4-42 represents the depreciation over time using the sum of years method for the tesla and the van. From the figure, the depreciation rate of the tesla is significantly higher than the depreciation rate of the van and this is because of the higher purchase value of the tesla as compared to the van.

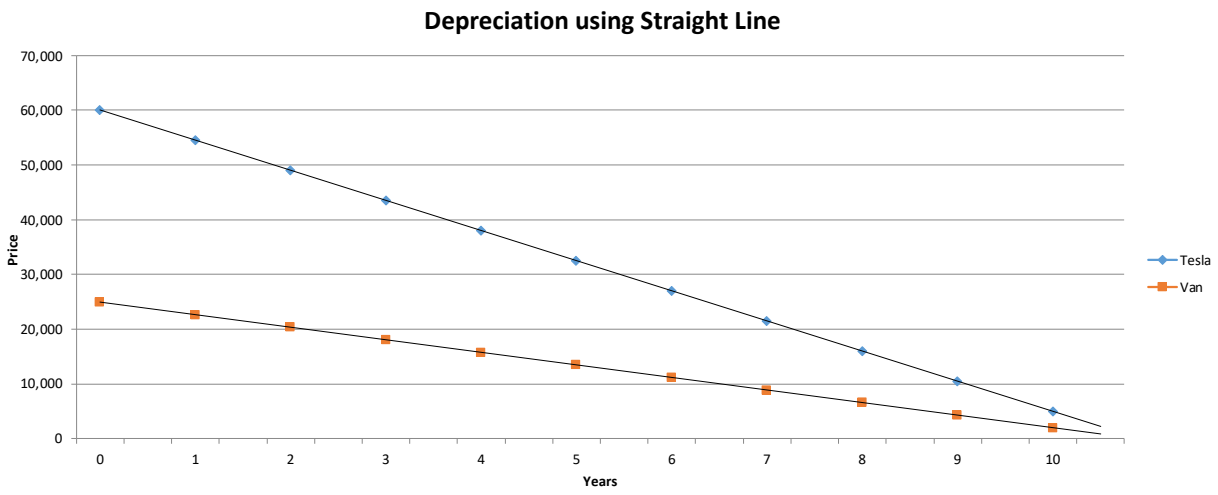


Figure 4-43 Depreciation using Straight Line

Figure 4-43 represents the depreciation over time using the straight-line method for both the tesla and the van. In the figure, it is represented that the tesla depreciates at a higher rate as compared to the van because of the initial value of both vehicles.

The depreciation has been considered using two different methods: the sum of years and straight-line method. For the straight-line method, it considers a constant depreciation rate over the service life of the asset. On the other hand, the sum of years method is used to accelerate the depreciation rate for the first couple of years which is more reliable and accurate. It should be that the sum of years method does not depreciate at a constant rate; therefore, it is significantly more accurate.

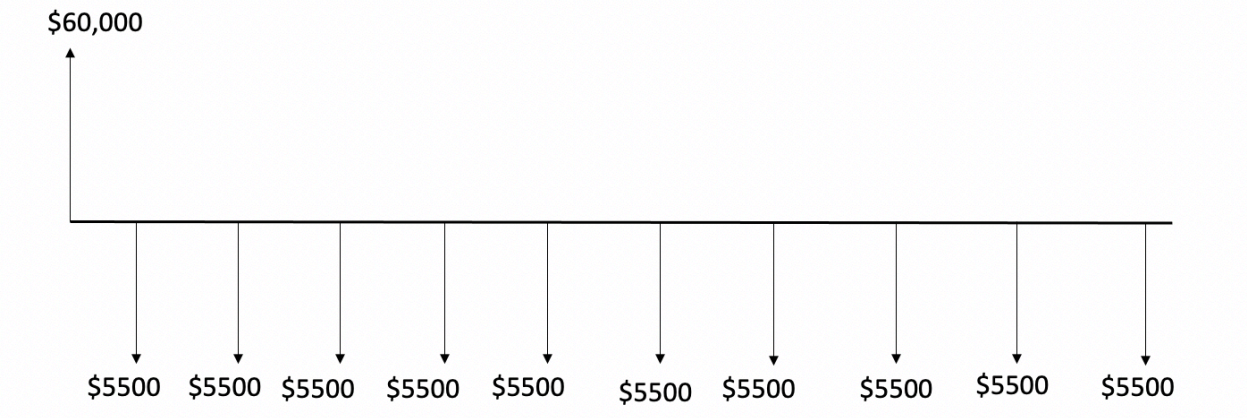


Figure 4-44 Cash Flow Diagram for Tesla

Figure 4-44 represents the cash flow diagram for the Tesla showing the depreciation over a 10-year period using the straight-line method.

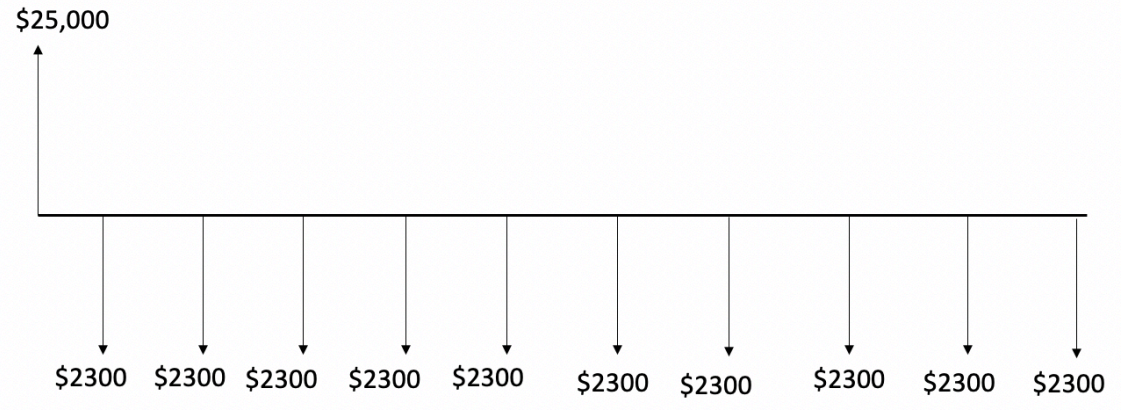


Figure 4-45 Cash Flow Diagram for Van

Figure 4-45 represents the cash flow diagram for the van demonstrating the depreciation over a 10-year period using the straight-line method.

An economic analysis was performed using future value of a Tesla and a van over a 10-year period. Table 4-5 represents the economic analysis comparison of the future value of Tesla and the van over a 10-year period.

Table 4-5 Economic Analysis for Future Value

	Tesla	**Van**
Future Value	Future Value after 10 years: $88,814.65 Interest rate 4% P=$60,000	Future Value after 10 years: $29,246.46 Interest rate 4% P=$25,000

5. Conclusions and Discussion

5.1 Conclusion

This research study titled, “Affordable Autonomous Vehicles for Deployment after Disastrous Events” is to use artificial intelligence and machine learning so that autonomous vehicles find a safe and efficient way to help those who are stranded during disastrous events and emergency situations. With self-driving vehicles becoming more and more common every day as the technology has been advancing, people are still reluctant to adapt to this new technology, for some it may be cost while for others it may be that they are not quite ready to trust these new technologies as it is still in a relatively early phase. The long-term goal of this research to be addressed in this study is if a low-cost AI-based sensor be retrofitted to make an affordable car self-driving cost effective. Our long-term research goal is to investigate AI driven low-cost sensors that can facilitate automated systems effectively as well as being affordable. The research objective is to provide an effective demonstration of a low-cost AI-based sensor that can enable an affordable car system to become smart and ultimately transition into a self-driving car, or autonomous vehicle. This research seeks to leverage affordable sensors, artificial intelligence, and machine learning to develop a smart car kit that can be retrofitted to any type of vehicle to make it smart. Approach 3 (Nonlinear) with range of 27 ft and a thickness of 1.97mm performed the best. Nonlinear Regression showed improved results over linear regression for all cases. The nonlinear regression with three independent variables showed the best correlated between predicted and experimental results for RFID tags. In addition, the nonlinear Regression with 3 independent variables yielded to the maximum error band of +/-15%. The AI model detects the object and gives the direction for the vehicle to turn left, right, or stop, according to the situation. Lastly, the current AI model is in detection of above 70%.

5.2 Findings and Recommendations

The result of this study is promising. It is recommended that major car manufacturers testing and employing the concept developed in this research. Second, it is also recommended that additional sensor technology, such as LiDAR and laser are also used for multiple verification of the result of the proximity sensor. The current AI model accuracy is in excess of 70%; therefore, it is recommended that the AI model be further refined to extend the level of accuracy.

5.3 Contributions to body of knowledge and Gantt chart

Table 5-1: Contributions to body of knowledge

Course Name	Application	Professor
IE 5301: Operations Research/Manufacturing	Using Operations Research techniques and their application to decision problems is essential for this project.	Dr. Edmund Prater and Dr. Emma Yang
IE 5318: Advanced Regression Analysis	A regression analysis will be performed using statistical concepts and analysis.	Dr. Emma Yang
IE 6308: Design of Experiments	Understanding the topics in design of experiments is vital for this project. Analysis of variance and designs will be used throughout the duration of this project.	Dr. Erick C. Jones, Sr.
IE 6318: Data Mining and Analytics	Understanding the algorithms discussed in Data mining is important as used for the optimization of the path planning.	Dr. Shouyi Wang
IE 5304: Advanced Engineering Economy	Understanding the topics in engineering economy is essential for this research. Benefit-cost analysis and improving profit and revenue will be vital for this project.	Dr. Erick C. Jones, Sr.

Table 5-2: Gantt chart

	Task	2021		2022		2023
1	*Literature Review and Preliminary Research*					
2	*Compare the performance of the RFID tag, proximity sensor, and camera readings that can simulate a LiDAR type sensor for affordable vehicles.*					
3	*Develop and compare algorithms that use the dataset and integrate into older vehicle data systems.*					
4	*Perform an economic and disaster relief analysis*					
5	*Dissertation Writing*					

5.4 Limitations

The scope of this research is at the following forefront: the multi-sensor device use are limited to RFID tag, camera, and proximity sensors. The measurement limitation for RFID tag was at every three feet increments. The measurement for the ultrasonic sensor was at 10 centimeters. A car bumper was used to accurately represent the actual vehicle moving towards an adjacent object and/or vehicle. The measurement for the RFID tags was at the following ranges: 12, 13, 14, and 27 feet. The artificial intelligence is developed for predictions based on the aforementioned limitations.

5.5 Future Work

This research study consisted of three objectives. In the future scope of this research, the artificial intelligence model will be improved so that it will have the capability to recognize other smaller objects with a higher precision and accuracy. Also, the vehicle will be at a level of autonomous so that during disastrous events, the vehicle will be able to rescue those who are stranded in emergency situations. The self-driving vehicle AI model will communicate with GPS for improved accuracy. A GPS system and self-driving vehicle algorithm will be synced and communicate so

that the vehicle will have a direction as to which path it needs to take, provided more obstacles in its way. This concept can not only be used for commercial vehicles, but also for emergency vehicles such as: police, ambulances, and firetrucks. It is recommended that the result of this study be used on an actual vehicle in motion on larger research funded by major car manufacturers and/or National Science Foundation.

Appendix A(Tables)

A.1 Output Data Tables for RFID Tags Approach 1

Output Data for 31mils

Distance	Read Rate
1	102
2	100
3	100
4	90
5	85
6	84
7	80
8	78
9	77
10	70
11	59
12	59
13	53
14	51
15	49
16	40
17	30
18	0
19	0

20	0
21	0
22	0
23	0
24	0
25	0
26	0
27	0
28	0
29	0
30	0
31	0
32	0
33	0

Output Data for 41mils

Distance	Read Rate
1	110
2	110
3	108
4	100
5	97
6	95
7	91
8	90
9	90
10	85
11	81
12	81
13	35
14	26
15	26
16	24
17	22
18	22
19	19
20	15
21	0
22	0
23	0
24	0

25	0
26	0
27	0
28	0
29	0
30	0
31	0
32	0
33	0

Output Data for 62 mils

Distance	Read Rate
1	108
2	109
3	109
4	109
5	102
6	104
7	108
8	109
9	109
10	110
11	97
12	94
13	94
14	87
15	85
16	85
17	104
18	104
19	108
20	110
21	109
22	110
23	108

24	109
25	109
26	89
27	88
28	88
29	60
30	43
31	35
32	30
33	0

Output Data for 72 mils

Distance	Read Rate
1	108
2	108
3	108
4	100
5	99
6	99
7	107
8	110
9	108
10	108
11	75
12	62
13	75
14	78
15	78
16	72
17	78
18	70
19	96
20	95
21	96
22	70
23	78

24	68
25	74
26	60
27	62
28	30
29	0
30	0
31	0
32	0
33	0

A.2 Output Data Tables for RFID Tags Approach 2

Appendix B (AI Codes)

B.1 Object Detection Code

B.2 Route Code

```
class AutonomousCar:
    def __init__(self,
                 rfi_read_left,
                 object_detect_left,
                 approx_sensor_left,
                 rfi_read_right,
                 approx_sensor_right,
                 speed_sensor):

    self.rfi_read_left = rfi_read_left
    self.object_detect_left = object_detect_left
    self.approx_sensor_left = approx_sensor_left
    self.rfi_read_right = rfi_read_right
    self.approx_sensor_right = approx_sensor_right
    self.speed_sensor = speed_sensor
    self.object_left_status = False
    self.object_right_status = False
    self.car_status = None


    def get_rfi_read_left(self):
        return self.rfi_read_left
```

```
    def get_object_detect_left(self):
        return self.object_detect_left

    def get_approx_sensor_left(self):
        return self.approx_sensor_left

    def get_rfi_read_right(self):
        return self.rfi_read_right

    def get_object_detect_right(self):
        return self.object_detect_right

    def get_approx_sensor_right(self):
        return self.approx_sensor_right

    def get_speed_sensor(self):
        return self.__speed_sensor

    def action(self):
        if self.speed_sensor > 0:
          if self.object_detect_left in ['car', 'truck', 'bike',
'pedestrain']:
            self.object_left_status = True
          if self.object_detect_right in ['car', 'truck', 'bike',
'pedestrain']:
            self.object_right_status = True

          if self.approx_sensor_left < 10 and self.rfi_read_left < 18 and
self.object_left_status:
            if self.approx_sensor_right > 10 or not
self.object_right_status or self.rfi_read_left > 18:
              car_status = 'autonomous car will turn right'
              return self.car_status
            else:
              self.car_status = 'autonomous car will continue straight'
              return self.car_status

          elif:
            if self.approx_sensor_right < 10 and self.rfi_read_right < 18
and self.object_right_status:
              self.car_status = 'autonomous car will turn left'
              return self.car_status
            else:
              self.car_status = 'autonomous car will continue straight'
              return self.car_status
        else:
          self.car_status = 'autonomous car is stopped'

        return self.car_status

def main():
    #API rfi sensor to get rfi data from left side of the car
    rfi_read_left = rfi_read_left()
```

```
    #Object detection using Yolov5 frome left camera
    object_detect_left = object_detect_left()
    #API aproximately left sensor
    approx_sensor_left = approx_sensor_left()
    #API rfi sensor to get rfi data from right side of the car
    rfi_read_right = rfi_read_right()
    #Object detection using Yolov5 frome right camera
    object_detect_right = object_detect_right()
    #API aproximately right sensor
    approx_sensor_right = approx_sensor_right()
    #API autonomous car speed
    speed_sensor = speed_sensor()

    MyAuto = AutonomousCar(rfi_read_left,
                           object_detect_left,
                           approx_sensor_left,
                           rfi_read_right,
                           approx_sensor_right,
                           speed_sensor)

    #show status/action of autonomous car
    MyAuto.action()

#main function
main()
```

Appendix C (RFID tag, Camera, and Sensor Codes)

C.1 RFID tag code:

```
#include <SPI.h>

#include <MFRC522.h>



#define SS_PIN 53

#define RST_PIN 5

#define echoPin 10

#define trigPin 9
```

```
long duration; // variable for the duration of sound wave travel
long distance; // variable for the distance measurement

MFRC522 mfrc522(SS_PIN, RST_PIN);   // Create MFRC522 instance.
// White tag 3A EE 2C 3C
//Blue tag  47 0F 5E 2D
void setup()
{

  pinMode(trigPin, OUTPUT); // Sets the trigPin as an OUTPUT
  pinMode(echoPin, INPUT); // Sets the echoPin as an INPUT



  Serial.begin(9600);   // Initiate a serial communication
  SPI.begin();      // Initiate  SPI bus
  mfrc522.PCD_Init();   // Initiate MFRC522
  Serial.println("Approximate your card to the reader...");
  Serial.println();
  Serial.println("Ultrasonic Sensor HC-SR04 waiting for RFID verification.");

}
void loop()
{
```

```
// Look for new cards
if ( ! mfrc522.PICC_IsNewCardPresent())
{
  return;
}
// Select one of the cards
if ( ! mfrc522.PICC_ReadCardSerial())
{
  return;
}
//Show UID on serial monitor
Serial.print("UID tag :");
String content= "";
byte letter;
for (byte i = 0; i < mfrc522.uid.size; i++)
{
   Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
   Serial.print(mfrc522.uid.uidByte[i], HEX);
   content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
   content.concat(String(mfrc522.uid.uidByte[i], HEX));
}
```

```
Serial.println();
Serial.print("Message : ");
content.toUpperCase();
delayMicroseconds(1000);

  if (content.substring(1) == "3A EE 2C 3C"){



      digitalWrite(trigPin, LOW);
      delayMicroseconds(2);
      digitalWrite(trigPin, HIGH);
      delayMicroseconds(10);


      digitalWrite(trigPin, LOW);
      // Reads the echoPin, returns the sound wave travel time in microseconds
      duration = pulseIn(echoPin, HIGH);
      // Calculating the distance
      distance = duration * 0.034 / 2;
      // Prints the distance on the Serial Monitor
      Serial.println("Ultrasonic sensor initiated");
      Serial.println();
      delayMicroseconds(2000);
```

```
        Serial.print("Distance: ");
        Serial.println(distance);
        delayMicroseconds(2000);




  }

  else
  {
    Serial.println("RFID authentication failed.....");
  }

}

#include <SPI.h>
#include <MFRC522.h>

#define SS_PIN 53
#define RST_PIN 5
#define echoPin 10
#define trigPin 9
```

```
long duration; // variable for the duration of sound wave travel
long distance; // variable for the distance measurement
int i=0;
MFRC522 mfrc522(SS_PIN, RST_PIN);   // Create MFRC522 instance.
// White tag 3A EE 2C 3C
//Blue tag  47 0F 5E 2D
void setup()
{

  pinMode(trigPin, OUTPUT); // Sets the trigPin as an OUTPUT
  pinMode(echoPin, INPUT); // Sets the echoPin as an INPUT



  Serial.begin(9600);   // Initiate a serial communication
  SPI.begin();      // Initiate  SPI bus
  mfrc522.PCD_Init();   // Initiate MFRC522
  Serial.println("Approximate your card to the reader...");
  Serial.println();
  Serial.println("Ultrasonic Sensor HC-SR04 waiting for RFID verification.");


}
void loop()
```

```
{



 // Look for new cards
 if ( ! mfrc522.PICC_IsNewCardPresent())
 {
  return;
 }
 // Select one of the cards
 if ( ! mfrc522.PICC_ReadCardSerial())
 {
  return;
 }
 //Show UID on serial monitor
 Serial.print("UID tag :");
 String content= "";
 byte letter;
 for (byte i = 0; i < mfrc522.uid.size; i++)
 {
    Serial.print(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " ");
    Serial.print(mfrc522.uid.uidByte[i], HEX);
    content.concat(String(mfrc522.uid.uidByte[i] < 0x10 ? " 0" : " "));
    content.concat(String(mfrc522.uid.uidByte[i], HEX));
```

```
}
Serial.println();
Serial.print("Message : ");
content.toUpperCase();
delayMicroseconds(1000);

  if (content.substring(1) == "3A EE 2C 3C"){

    for (i = 0; i<=1000; i++){
        digitalWrite(trigPin, LOW);
        delayMicroseconds(2);
        digitalWrite(trigPin, HIGH);
        delayMicroseconds(10);

        digitalWrite(trigPin, LOW);
        // Reads the echoPin, returns the sound wave travel time in microseconds
        duration = pulseIn(echoPin, HIGH);
        // Calculating the distance
        distance = duration * 0.034 / 2;
        // Prints the distance on the Serial Monitor
        Serial.println("Ultrasonic sensor initiated");
        Serial.println();
        delayMicroseconds(2000);
```

```
        Serial.print("Distance: ");
        Serial.println(distance);
        delayMicroseconds(2000);

        i = i + 1;
      }

  }

  else
  {
    Serial.println("RFID authentication failed.....");
  }

}
```

C.2 Camera codes:

```
//
// Source code for application to transmit image from ov7670 to PC via USB
// By Siarhei Charkes in 2015
// http://privateblog.info
```

```
//


#include <stdint.h>
#include <avr/io.h>
#include <util/twi.h>
#include <util/delay.h>
#include <avr/pgmspace.h>

#define F_CPU 16000000UL
#define vga   0
#define qvga  1
#define qqvga   2
#define yuv422  0
#define rgb565  1
#define bayerRGB  2
#define camAddr_WR  0x42
#define camAddr_RD  0x43

/* Registers */
#define REG_GAIN    0x00  /* Gain lower 8 bits (rest in vref) */
#define REG_BLUE    0x01  /* blue gain */
#define REG_RED       0x02  /* red gain */
#define REG_VREF    0x03  /* Pieces of GAIN, VSTART, VSTOP */
```

```
#define REG_COM1    0x04  /* Control 1 */
#define COM1_CCIR656  0x40    /* CCIR656 enable */

#define REG_BAVE    0x05  /* U/B Average level */
#define REG_GbAVE   0x06  /* Y/Gb Average level */
#define REG_AECHH   0x07  /* AEC MS 5 bits */
#define REG_RAVE    0x08  /* V/R Average level */
#define REG_COM2    0x09  /* Control 2 */
#define COM2_SSLEEP         0x10  /* Soft sleep mode */
#define REG_PID           0x0a  /* Product ID MSB */
#define REG_VER           0x0b  /* Product ID LSB */
#define REG_COM3    0x0c  /* Control 3 */
#define COM3_SWAP         0x40  /* Byte swap */
#define COM3_SCALEEN          0x08  /* Enable scaling */
#define COM3_DCWEN          0x04  /* Enable downsamp/crop/window */
#define REG_COM4    0x0d  /* Control 4 */
#define REG_COM5    0x0e  /* All "reserved" */
#define REG_COM6    0x0f  /* Control 6 */
#define REG_AECH    0x10  /* More bits of AEC value */
#define REG_CLKRC   0x11  /* Clocl control */
#define CLK_EXT           0x40  /* Use external clock directly */
#define CLK_SCALE   0x3f  /* Mask for internal clock scale */
#define REG_COM7    0x12  /* Control 7 */ //REG mean address.
```

```
#define COM7_RESET         0x80  /* Register reset */
#define COM7_FMT_MASK      0x38
#define COM7_FMT_VGA       0x00
#define COM7_FMT_CIF       0x20  /* CIF format */
#define COM7_FMT_QVGA      0x10  /* QVGA format */
#define COM7_FMT_QCIF      0x08  /* QCIF format */
#define COM7_RGB           0x04  /* bits 0 and 2 - RGB format */
#define COM7_YUV           0x00  /* YUV */
#define COM7_BAYER         0x01  /* Bayer format */
#define COM7_PBAYER        0x05  /* "Processed bayer" */
#define REG_COM8    0x13  /* Control 8 */
#define COM8_FASTAEC       0x80  /* Enable fast AGC/AEC */
#define COM8_AECSTEP       0x40  /* Unlimited AEC step size */
#define COM8_BFILT    0x20  /* Band filter enable */
#define COM8_AGC    0x04  /* Auto gain enable */
#define COM8_AWB    0x02  /* White balance enable */
#define COM8_AEC    0x01  /* Auto exposure enable */
#define REG_COM9    0x14  /* Control 9- gain ceiling */
#define REG_COM10   0x15  /* Control 10 */
#define COM10_HSYNC        0x40  /* HSYNC instead of HREF */
#define COM10_PCLK_HB      0x20  /* Suppress PCLK on horiz blank */
#define COM10_HREF_REV     0x08  /* Reverse HREF */
#define COM10_VS_LEAD      0x04  /* VSYNC on clock leading edge */
```

```
#define COM10_VS_NEG        0x02  /* VSYNC negative */
#define COM10_HS_NEG        0x01  /* HSYNC negative */
#define REG_HSTART    0x17  /* Horiz start high bits */
#define REG_HSTOP   0x18  /* Horiz stop high bits */
#define REG_VSTART    0x19  /* Vert start high bits */
#define REG_VSTOP   0x1a  /* Vert stop high bits */
#define REG_PSHFT   0x1b  /* Pixel delay after HREF */
#define REG_MIDH    0x1c  /* Manuf. ID high */
#define REG_MIDL    0x1d  /* Manuf. ID low */
#define REG_MVFP    0x1e  /* Mirror / vflip */
#define MVFP_MIRROR       0x20  /* Mirror image */
#define MVFP_FLIP   0x10  /* Vertical flip */

#define REG_AEW         0x24  /* AGC upper limit */
#define REG_AEB         0x25    /* AGC lower limit */
#define REG_VPT         0x26  /* AGC/AEC fast mode op region */
#define REG_HSYST   0x30  /* HSYNC rising edge delay */
#define REG_HSYEN   0x31  /* HSYNC falling edge delay */
#define REG_HREF    0x32  /* HREF pieces */
#define REG_TSLB    0x3a  /* lots of stuff */
#define TSLB_YLAST    0x04  /* UYVY or VYUY - see com13 */
#define REG_COM11   0x3b  /* Control 11 */
#define COM11_NIGHT       0x80  /* NIght mode enable */
```

```
#define COM11_NMFR          0x60  /* Two bit NM frame rate */
#define COM11_HZAUTO        0x10  /* Auto detect 50/60 Hz */
#define COM11_50HZ          0x08  /* Manual 50Hz select */
#define COM11_EXP   0x02
#define REG_COM12   0x3c  /* Control 12 */
#define COM12_HREF          0x80  /* HREF always */
#define REG_COM13   0x3d  /* Control 13 */
#define COM13_GAMMA         0x80  /* Gamma enable */
#define COM13_UVSAT         0x40  /* UV saturation auto adjustment */
#define COM13_UVSWAP        0x01  /* V before U - w/TSLB */
#define REG_COM14   0x3e  /* Control 14 */
#define COM14_DCWEN         0x10  /* DCW/PCLK-scale enable */
#define REG_EDGE    0x3f  /* Edge enhancement factor */
#define REG_COM15   0x40  /* Control 15 */
#define COM15_R10F0         0x00  /* Data range 10 to F0 */
#define COM15_R01FE         0x80  /*      01 to FE */
#define COM15_R00FF         0xc0  /*      00 to FF */
#define COM15_RGB565        0x10  /* RGB565 output */
#define COM15_RGB555        0x30  /* RGB555 output */
#define REG_COM16   0x41  /* Control 16 */
#define COM16_AWBGAIN       0x08  /* AWB gain enable */
#define REG_COM17   0x42  /* Control 17 */
#define COM17_AECWIN        0xc0  /* AEC window - must match COM4 */
```

```
#define COM17_CBAR          0x08  /* DSP Color bar */

/*

* This matrix defines how the colors are generated, must be

* tweaked to adjust hue and saturation.

*

* Order: v-red, v-green, v-blue, u-red, u-green, u-blue

* They are nine-bit signed quantities, with the sign bit

* stored in0x58.Sign for v-red is bit 0, and up from there.

*/

#define REG_CMATRIX_BASE  0x4f

#define CMATRIX_LEN           6

#define REG_CMATRIX_SIGN  0x58

#define REG_BRIGHT    0x55  /* Brightness */

#define REG_CONTRAS         0x56  /* Contrast control */

#define REG_GFIX    0x69  /* Fix gain control */

#define REG_REG76   0x76  /* OV's name */

#define R76_BLKPCOR         0x80  /* Black pixel correction enable */

#define R76_WHTPCOR         0x40  /* White pixel correction enable */

#define REG_RGB444          0x8c  /* RGB 444 control */

#define R444_ENABLE         0x02  /* Turn on RGB444, overrides 5x5 */

#define R444_RGBX   0x01  /* Empty nibble at end */

#define REG_HAECC1    0x9f  /* Hist AEC/AGC control 1 */

#define REG_HAECC2    0xa0  /* Hist AEC/AGC control 2 */
```

```
#define REG_BD50MAX        0xa5  /* 50hz banding step limit */
#define REG_HAECC3    0xa6  /* Hist AEC/AGC control 3 */
#define REG_HAECC4    0xa7  /* Hist AEC/AGC control 4 */
#define REG_HAECC5    0xa8  /* Hist AEC/AGC control 5 */
#define REG_HAECC6    0xa9  /* Hist AEC/AGC control 6 */
#define REG_HAECC7    0xaa  /* Hist AEC/AGC control 7 */
#define REG_BD60MAX        0xab  /* 60hz banding step limit */
#define REG_GAIN    0x00  /* Gain lower 8 bits (rest in vref) */
#define REG_BLUE    0x01  /* blue gain */
#define REG_RED          0x02  /* red gain */
#define REG_VREF    0x03  /* Pieces of GAIN, VSTART, VSTOP */
#define REG_COM1    0x04  /* Control 1 */
#define COM1_CCIR656         0x40  /* CCIR656 enable */
#define REG_BAVE    0x05  /* U/B Average level */
#define REG_GbAVE   0x06  /* Y/Gb Average level */
#define REG_AECHH   0x07  /* AEC MS 5 bits */
#define REG_RAVE    0x08  /* V/R Average level */
#define REG_COM2    0x09  /* Control 2 */
#define COM2_SSLEEP        0x10  /* Soft sleep mode */
#define REG_PID          0x0a  /* Product ID MSB */
#define REG_VER          0x0b  /* Product ID LSB */
#define REG_COM3    0x0c  /* Control 3 */
#define COM3_SWAP        0x40  /* Byte swap */
```

```
#define COM3_SCALEEN          0x08  /* Enable scaling */
#define COM3_DCWEN            0x04  /* Enable downsamp/crop/window */
#define REG_COM4    0x0d  /* Control 4 */
#define REG_COM5    0x0e  /* All "reserved" */
#define REG_COM6    0x0f  /* Control 6 */
#define REG_AECH    0x10  /* More bits of AEC value */
#define REG_CLKRC   0x11  /* Clocl control */
#define CLK_EXT           0x40  /* Use external clock directly */
#define CLK_SCALE   0x3f  /* Mask for internal clock scale */
#define REG_COM7    0x12  /* Control 7 */
#define COM7_RESET          0x80  /* Register reset */
#define COM7_FMT_MASK         0x38
#define COM7_FMT_VGA          0x00
#define COM7_FMT_CIF          0x20  /* CIF format */
#define COM7_FMT_QVGA         0x10  /* QVGA format */
#define COM7_FMT_QCIF         0x08  /* QCIF format */
#define COM7_RGB    0x04  /* bits 0 and 2 - RGB format */
#define COM7_YUV    0x00  /* YUV */
#define COM7_BAYER          0x01  /* Bayer format */
#define COM7_PBAYER         0x05  /* "Processed bayer" */
#define REG_COM8    0x13  /* Control 8 */
#define COM8_FASTAEC          0x80  /* Enable fast AGC/AEC */
#define COM8_AECSTEP          0x40  /* Unlimited AEC step size */
```

```
#define COM8_BFILT    0x20  /* Band filter enable */
#define COM8_AGC    0x04  /* Auto gain enable */
#define COM8_AWB    0x02  /* White balance enable */
#define COM8_AEC    0x01  /* Auto exposure enable */
#define REG_COM9    0x14  /* Control 9- gain ceiling */
#define REG_COM10   0x15  /* Control 10 */
#define COM10_HSYNC         0x40  /* HSYNC instead of HREF */
#define COM10_PCLK_HB         0x20  /* Suppress PCLK on horiz blank */
#define COM10_HREF_REV          0x08  /* Reverse HREF */
#define COM10_VS_LEAD          0x04  /* VSYNC on clock leading edge */
#define COM10_VS_NEG          0x02  /* VSYNC negative */
#define COM10_HS_NEG          0x01  /* HSYNC negative */
#define REG_HSTART    0x17  /* Horiz start high bits */
#define REG_HSTOP   0x18  /* Horiz stop high bits */
#define REG_VSTART    0x19  /* Vert start high bits */
#define REG_VSTOP   0x1a  /* Vert stop high bits */
#define REG_PSHFT   0x1b  /* Pixel delay after HREF */
#define REG_MIDH    0x1c  /* Manuf. ID high */
#define REG_MIDL    0x1d  /* Manuf. ID low */
#define REG_MVFP    0x1e  /* Mirror / vflip */
#define MVFP_MIRROR         0x20  /* Mirror image */
#define MVFP_FLIP   0x10  /* Vertical flip */
#define REG_AEW           0x24  /* AGC upper limit */
```

```
#define REG_AEB          0x25  /* AGC lower limit */
#define REG_VPT          0x26  /* AGC/AEC fast mode op region */
#define REG_HSYST   0x30  /* HSYNC rising edge delay */
#define REG_HSYEN   0x31  /* HSYNC falling edge delay */
#define REG_HREF    0x32  /* HREF pieces */
#define REG_TSLB    0x3a  /* lots of stuff */
#define TSLB_YLAST    0x04  /* UYVY or VYUY - see com13 */
#define REG_COM11   0x3b  /* Control 11 */
#define COM11_NIGHT         0x80  /* NIght mode enable */
#define COM11_NMFR          0x60  /* Two bit NM frame rate */
#define COM11_HZAUTO          0x10  /* Auto detect 50/60 Hz */
#define COM11_50HZ          0x08  /* Manual 50Hz select */
#define COM11_EXP   0x02
#define REG_COM12   0x3c  /* Control 12 */
#define COM12_HREF          0x80  /* HREF always */
#define REG_COM13   0x3d  /* Control 13 */
#define COM13_GAMMA         0x80  /* Gamma enable */
#define COM13_UVSAT         0x40  /* UV saturation auto adjustment */
#define COM13_UVSWAP          0x01  /* V before U - w/TSLB */
#define REG_COM14   0x3e  /* Control 14 */
#define COM14_DCWEN         0x10  /* DCW/PCLK-scale enable */
#define REG_EDGE    0x3f  /* Edge enhancement factor */
#define REG_COM15   0x40  /* Control 15 */
```

```
#define COM15_R10F0        0x00  /* Data range 10 to F0 */
#define COM15_R01FE        0x80  /*      01 to FE */
#define COM15_R00FF        0xc0  /*      00 to FF */
#define COM15_RGB565         0x10  /* RGB565 output */
#define COM15_RGB555         0x30  /* RGB555 output */
#define REG_COM16   0x41  /* Control 16 */
#define COM16_AWBGAIN        0x08  /* AWB gain enable */
#define REG_COM17   0x42  /* Control 17 */
#define COM17_AECWIN         0xc0  /* AEC window - must match COM4 */
#define COM17_CBAR         0x08  /* DSP Color bar */

#define CMATRIX_LEN            6
#define REG_BRIGHT    0x55  /* Brightness */
#define REG_REG76   0x76  /* OV's name */
#define R76_BLKPCOR        0x80  /* Black pixel correction enable */
#define R76_WHTPCOR        0x40  /* White pixel correction enable */
#define REG_RGB444         0x8c  /* RGB 444 control */
#define R444_ENABLE        0x02  /* Turn on RGB444, overrides 5x5 */
#define R444_RGBX   0x01  /* Empty nibble at end */
#define REG_HAECC1    0x9f  /* Hist AEC/AGC control 1 */
#define REG_HAECC2    0xa0  /* Hist AEC/AGC control 2 */
#define REG_BD50MAX        0xa5  /* 50hz banding step limit */
#define REG_HAECC3    0xa6  /* Hist AEC/AGC control 3 */
```

```
#define REG_HAECC4    0xa7  /* Hist AEC/AGC control 4 */
#define REG_HAECC5    0xa8  /* Hist AEC/AGC control 5 */
#define REG_HAECC6    0xa9  /* Hist AEC/AGC control 6 */
#define REG_HAECC7    0xaa  /* Hist AEC/AGC control 7 */
#define REG_BD60MAX        0xab  /* 60hz banding step limit */
#define MTX1           0x4f  /* Matrix Coefficient 1 */
#define MTX2           0x50  /* Matrix Coefficient 2 */
#define MTX3           0x51  /* Matrix Coefficient 3 */
#define MTX4           0x52  /* Matrix Coefficient 4 */
#define MTX5           0x53  /* Matrix Coefficient 5 */
#define MTX6           0x54  /* Matrix Coefficient 6 */
#define REG_CONTRAS        0x56  /* Contrast control */
#define MTXS           0x58  /* Matrix Coefficient Sign */
#define AWBC7          0x59  /* AWB Control 7 */
#define AWBC8          0x5a  /* AWB Control 8 */
#define AWBC9          0x5b  /* AWB Control 9 */
#define AWBC10           0x5c  /* AWB Control 10 */
#define AWBC11           0x5d  /* AWB Control 11 */
#define AWBC12           0x5e  /* AWB Control 12 */
#define REG_GFI          0x69  /* Fix gain control */
#define GGAIN          0x6a  /* G Channel AWB Gain */
#define DBLV           0x6b
#define AWBCTR3          0x6c  /* AWB Control 3 */
```

```
#define AWBCTR2         0x6d  /* AWB Control 2 */
#define AWBCTR1         0x6e  /* AWB Control 1 */
#define AWBCTR0         0x6f  /* AWB Control 0 */

struct regval_list{
  uint8_t reg_num;
  uint16_t value;
};

const struct regval_list qvga_ov7670[] PROGMEM = {
  { REG_COM14, 0x19 },
  { 0x72, 0x11 },
  { 0x73, 0xf1 },

  { REG_HSTART, 0x16 },
  { REG_HSTOP, 0x04 },
  { REG_HREF, 0xa4 },
  { REG_VSTART, 0x02 },
  { REG_VSTOP, 0x7a },
  { REG_VREF, 0x0a },

/*  { REG_HSTART, 0x16 },
```

```
  { REG_HSTOP, 0x04 },
  { REG_HREF, 0x24 },
  { REG_VSTART, 0x02 },
  { REG_VSTOP, 0x7a },
  { REG_VREF, 0x0a },*/
  { 0xff, 0xff }, /* END MARKER */
};

const struct regval_list yuv422_ov7670[] PROGMEM = {
  { REG_COM7, 0x0 },  /* Selects YUV mode */
  { REG_RGB444, 0 },  /* No RGB444 please */
  { REG_COM1, 0 },
  { REG_COM15, COM15_R00FF },
  { REG_COM9, 0x6A }, /* 128x gain ceiling; 0x8 is reserved bit */
  { 0x4f, 0x80 },   /* "matrix coefficient 1" */
  { 0x50, 0x80 },   /* "matrix coefficient 2" */
  { 0x51, 0 },    /* vb */
  { 0x52, 0x22 },   /* "matrix coefficient 4" */
  { 0x53, 0x5e },   /* "matrix coefficient 5" */
  { 0x54, 0x80 },   /* "matrix coefficient 6" */
  { REG_COM13, COM13_UVSAT },
  { 0xff, 0xff },   /* END MARKER */
};
```

```
const struct regval_list ov7670_default_regs[] PROGMEM = {//from the linux driver
  { REG_COM7, COM7_RESET },
  { REG_TSLB, 0x04 }, /* OV */
  { REG_COM7, 0 },  /* VGA */
  /*
  * Set the hardware window.  These values from OV don't entirely
  * make sense - hstop is less than hstart.  But they work...
  */
  { REG_HSTART, 0x13 }, { REG_HSTOP, 0x01 },
  { REG_HREF, 0xb6 }, { REG_VSTART, 0x02 },
  { REG_VSTOP, 0x7a }, { REG_VREF, 0x0a },

  { REG_COM3, 0 }, { REG_COM14, 0 },
  /* Mystery scaling numbers */
  { 0x70, 0x3a }, { 0x71, 0x35 },
  { 0x72, 0x11 }, { 0x73, 0xf0 },
  { 0xa2,/* 0x02 changed to 1*/1 }, { REG_COM10, 0x0 },
  /* Gamma curve values */
  { 0x7a, 0x20 }, { 0x7b, 0x10 },
  { 0x7c, 0x1e }, { 0x7d, 0x35 },
  { 0x7e, 0x5a }, { 0x7f, 0x69 },
  { 0x80, 0x76 }, { 0x81, 0x80 },
```

```
{ 0x82, 0x88 }, { 0x83, 0x8f },
{ 0x84, 0x96 }, { 0x85, 0xa3 },
{ 0x86, 0xaf }, { 0x87, 0xc4 },
{ 0x88, 0xd7 }, { 0x89, 0xe8 },
/* AGC and AEC parameters.  Note we start by disabling those features,
then turn them only after tweaking the values. */
{ REG_COM8, COM8_FASTAEC | COM8_AECSTEP },
{ REG_GAIN, 0 }, { REG_AECH, 0 },
{ REG_COM4, 0x40 }, /* magic reserved bit */
{ REG_COM9, 0x18 }, /* 4x gain + magic rsvd bit */
{ REG_BD50MAX, 0x05 }, { REG_BD60MAX, 0x07 },
{ REG_AEW, 0x95 }, { REG_AEB, 0x33 },
{ REG_VPT, 0xe3 }, { REG_HAECC1, 0x78 },
{ REG_HAECC2, 0x68 }, { 0xa1, 0x03 }, /* magic */
{ REG_HAECC3, 0xd8 }, { REG_HAECC4, 0xd8 },
{ REG_HAECC5, 0xf0 }, { REG_HAECC6, 0x90 },
{ REG_HAECC7, 0x94 },
{ REG_COM8, COM8_FASTAEC | COM8_AECSTEP | COM8_AGC | COM8_AEC },
{ 0x30, 0 }, { 0x31, 0 },//disable some delays
/* Almost all of these are magic "reserved" values.  */
{ REG_COM5, 0x61 }, { REG_COM6, 0x4b },
{ 0x16, 0x02 }, { REG_MVFP, 0x07 },
{ 0x21, 0x02 }, { 0x22, 0x91 },
```

```
{ 0x29, 0x07 }, { 0x33, 0x0b },

{ 0x35, 0x0b }, { 0x37, 0x1d },

{ 0x38, 0x71 }, { 0x39, 0x2a },

{ REG_COM12, 0x78 }, { 0x4d, 0x40 },

{ 0x4e, 0x20 }, { REG_GFIX, 0 },

/*{0x6b, 0x4a},*/{ 0x74, 0x10 },

{ 0x8d, 0x4f }, { 0x8e, 0 },

{ 0x8f, 0 }, { 0x90, 0 },

{ 0x91, 0 }, { 0x96, 0 },

{ 0x9a, 0 }, { 0xb0, 0x84 },

{ 0xb1, 0x0c }, { 0xb2, 0x0e },

{ 0xb3, 0x82 }, { 0xb8, 0x0a },


/* More reserved magic, some of which tweaks white balance */

{ 0x43, 0x0a }, { 0x44, 0xf0 },

{ 0x45, 0x34 }, { 0x46, 0x58 },

{ 0x47, 0x28 }, { 0x48, 0x3a },

{ 0x59, 0x88 }, { 0x5a, 0x88 },

{ 0x5b, 0x44 }, { 0x5c, 0x67 },

{ 0x5d, 0x49 }, { 0x5e, 0x0e },

{ 0x6c, 0x0a }, { 0x6d, 0x55 },

{ 0x6e, 0x11 }, { 0x6f, 0x9e }, /* it was 0x9F "9e for advance AWB" */

{ 0x6a, 0x40 }, { REG_BLUE, 0x40 },
```

```
  { REG_RED, 0x60 },
  { REG_COM8, COM8_FASTAEC | COM8_AECSTEP | COM8_AGC | COM8_AEC |
COM8_AWB },

  /* Matrix coefficients */
  { 0x4f, 0x80 }, { 0x50, 0x80 },
  { 0x51, 0 },    { 0x52, 0x22 },
  { 0x53, 0x5e }, { 0x54, 0x80 },
  { 0x58, 0x9e },

  { REG_COM16, COM16_AWBGAIN }, { REG_EDGE, 0 },
  { 0x75, 0x05 }, { REG_REG76, 0xe1 },
  { 0x4c, 0 },    { 0x77, 0x01 },
  { REG_COM13, /*0xc3*/0x48 }, { 0x4b, 0x09 },
  { 0xc9, 0x60 },   /*{REG_COM16, 0x38},*/
  { 0x56, 0x40 },

  { 0x34, 0x11 }, { REG_COM11, COM11_EXP | COM11_HZAUTO },
  { 0xa4, 0x82/*Was 0x88*/ }, { 0x96, 0 },
  { 0x97, 0x30 }, { 0x98, 0x20 },
  { 0x99, 0x30 }, { 0x9a, 0x84 },
  { 0x9b, 0x29 }, { 0x9c, 0x03 },
  { 0x9d, 0x4c }, { 0x9e, 0x3f },
```

```
  { 0x78, 0x04 },

  /* Extra-weird stuff.  Some sort of multiplexor register */
  { 0x79, 0x01 }, { 0xc8, 0xf0 },
  { 0x79, 0x0f }, { 0xc8, 0x00 },
  { 0x79, 0x10 }, { 0xc8, 0x7e },
  { 0x79, 0x0a }, { 0xc8, 0x80 },
  { 0x79, 0x0b }, { 0xc8, 0x01 },
  { 0x79, 0x0c }, { 0xc8, 0x0f },
  { 0x79, 0x0d }, { 0xc8, 0x20 },
  { 0x79, 0x09 }, { 0xc8, 0x80 },
  { 0x79, 0x02 }, { 0xc8, 0xc0 },
  { 0x79, 0x03 }, { 0xc8, 0x40 },
  { 0x79, 0x05 }, { 0xc8, 0x30 },
  { 0x79, 0x26 },
  { 0xff, 0xff }, /* END MARKER */
};



void error_led(void){
  DDRB |= 32;//make sure led is output
  while (1){//wait for reset
    PORTB ^= 32;// toggle led
```

```
    _delay_ms(100);
  }
}

void twiStart(void){
  TWCR = _BV(TWINT) | _BV(TWSTA) | _BV(TWEN);//send start
  while (!(TWCR & (1 << TWINT)));//wait for start to be transmitted
  if ((TWSR & 0xF8) != TW_START)
    error_led();
}

void twiWriteByte(uint8_t DATA, uint8_t type){
  TWDR = DATA;
  TWCR = _BV(TWINT) | _BV(TWEN);
  while (!(TWCR & (1 << TWINT))) {}
  if ((TWSR & 0xF8) != type)
    error_led();
}

void twiAddr(uint8_t addr, uint8_t typeTWI){
  TWDR = addr;//send address
  TWCR = _BV(TWINT) | _BV(TWEN);    /* clear interrupt to start transmission */
  while ((TWCR & _BV(TWINT)) == 0); /* wait for transmission */
```

```
  if ((TWSR & 0xF8) != typeTWI)
    error_led();
}

void wrReg(uint8_t reg, uint8_t dat){
  //send start condition
  twiStart();
  twiAddr(camAddr_WR, TW_MT_SLA_ACK);
  twiWriteByte(reg, TW_MT_DATA_ACK);
  twiWriteByte(dat, TW_MT_DATA_ACK);
  TWCR = (1 << TWINT) | (1 << TWEN) | (1 << TWSTO);//send stop
  _delay_ms(1);
}

static uint8_t twiRd(uint8_t nack){
  if (nack){
    TWCR = _BV(TWINT) | _BV(TWEN);
    while ((TWCR & _BV(TWINT)) == 0); /* wait for transmission */
    if ((TWSR & 0xF8) != TW_MR_DATA_NACK)
      error_led();
    return TWDR;
  }
  else{
```

```
    TWCR = _BV(TWINT) | _BV(TWEN) | _BV(TWEA);
    while ((TWCR & _BV(TWINT)) == 0); /* wait for transmission */
    if ((TWSR & 0xF8) != TW_MR_DATA_ACK)
      error_led();
    return TWDR;
  }
}

uint8_t rdReg(uint8_t reg){
  uint8_t dat;
  twiStart();
  twiAddr(camAddr_WR, TW_MT_SLA_ACK);
  twiWriteByte(reg, TW_MT_DATA_ACK);
  TWCR = (1 << TWINT) | (1 << TWEN) | (1 << TWSTO);//send stop
  _delay_ms(1);
  twiStart();
  twiAddr(camAddr_RD, TW_MR_SLA_ACK);
  dat = twiRd(1);
  TWCR = (1 << TWINT) | (1 << TWEN) | (1 << TWSTO);//send stop
  _delay_ms(1);
  return dat;
}
```

```
void wrSensorRegs8_8(const struct regval_list reglist[]){
  uint8_t reg_addr, reg_val;
  const struct regval_list *next = reglist;
  while ((reg_addr != 0xff) | (reg_val != 0xff)){
    reg_addr = pgm_read_byte(&next->reg_num);
    reg_val = pgm_read_byte(&next->value);
    wrReg(reg_addr, reg_val);
    next++;
  }
}

void setColor(void){
  wrSensorRegs8_8(yuv422_ov7670);
}

void setRes(void){
  wrReg(REG_COM3, 4); // REG_COM3 enable scaling
  wrSensorRegs8_8(qvga_ov7670);
}

void camInit(void){
  wrReg(0x12, 0x80);
  _delay_ms(100);
```

```
  wrSensorRegs8_8(ov7670_default_regs);
  wrReg(REG_COM10, 32);//PCLK does not toggle on HBLANK.
}

void arduinoUnoInut(void) {
  cli();//disable interrupts

    /* Setup the 8mhz PWM clock
  * This will be on pin 11*/
  DDRB |= (1 << 3);//pin 11
  ASSR &= ~(_BV(EXCLK) | _BV(AS2));
  TCCR2A = (1 << COM2A0) | (1 << WGM21) | (1 << WGM20);
  TCCR2B = (1 << WGM22) | (1 << CS20);
  OCR2A = 0;//(F_CPU)/(2*(X+1))
  DDRC &= ~15;//low d0-d3 camera
  DDRD &= ~252;//d7-d4 and interrupt pins
  _delay_ms(3000);

    //set up twi for 100khz
  TWSR &= ~3;//disable prescaler for TWI
  TWBR = 72;//set to 100khz

    //enable serial
```

```
  UBRR0H = 0;
  UBRR0L = 1;//0 = 2M baud rate. 1 = 1M baud. 3 = 0.5M. 7 = 250k 207 is 9600 baud rate.
  UCSR0A |= 2;//double speed aysnc
  UCSR0B = (1 << RXEN0) | (1 << TXEN0);//Enable receiver and transmitter
  UCSR0C = 6;//async 1 stop bit 8bit char no parity bits
}



void StringPgm(const char * str){
  do{
     while (!(UCSR0A & (1 << UDRE0)));//wait for byte to transmit
     UDR0 = pgm_read_byte_near(str);
     while (!(UCSR0A & (1 << UDRE0)));//wait for byte to transmit
  } while (pgm_read_byte_near(++str));
}

static void captureImg(uint16_t wg, uint16_t hg){
  uint16_t y, x;

  StringPgm(PSTR("*RDY*"));

  while (!(PIND & 8));//wait for high
  while ((PIND & 8));//wait for low
```

```
   y = hg;
  while (y--){
       x = wg;
     //while (!(PIND & 256));//wait for high
    while (x--){
     while ((PIND & 4));//wait for low
          UDR0 = (PINC & 15) | (PIND & 240);
        while (!(UCSR0A & (1 << UDRE0)));//wait for byte to transmit
     while (!(PIND & 4));//wait for high
     while ((PIND & 4));//wait for low
     while (!(PIND & 4));//wait for high
    }
   //  while ((PIND & 256));//wait for low
  }
   _delay_ms(100);
}

void setup(){
  arduinoUnoInut();
  camInit();
  setRes();
  setColor();
```

```
  wrReg(0x11, 10); //Earlier it had the value: wrReg(0x11, 12); New version works better for me
:) !!!!
}


void loop(){
  captureImg(320, 240);
}
```

C.3 Ultrasonic Sensor Code

```
#define echoPin 2
#define trigPin 3
#define echoPin1 4
#define trigPin1 5
// defines variables
long duration; // variable for the duration of sound wave travel
long distance; // variable for the distance measurement
long duration1;
long distance1;

void setup() {
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an OUTPUT
  pinMode(echoPin, INPUT); // Sets the echoPin as an INPUT
```

```
  pinMode(trigPin1, OUTPUT); // Sets the trigPin as an OUTPUT
  pinMode(echoPin1, INPUT); // Sets the echoPin as an INPUT

  Serial.begin(9600); // // Serial Communication is starting with 9600 of baudrate speed
  Serial.println("Ultrasonic Sensor HC-SR04 Test"); // print some text in Serial Monitor
  Serial.println("with Arduino Mega ");
}
void loop() {
  // Clears the trigPin condition
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration * 0.034 / 2; // Speed of sound wave divided by 2 (go and back)
  Serial.print("Distance from 1st sensor: ");
  Serial.print(distance);
  Serial.println(" cm");
  delay(250);

  digitalWrite(trigPin1, LOW);
  delayMicroseconds(2);
```

```
  digitalWrite(trigPin1, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin1, LOW);
  duration1 = pulseIn(echoPin1, HIGH);
  distance1 = duration1 * 0.034 / 2; // Speed of sound wave divided by 2 (go and back)
  Serial.print("Distance from 2nd sensor: ");
  Serial.print(distance1);
  Serial.println(" cm");
  delay(250);

  if(distance < 10){
    Serial.println("Distance from sensor 1 too low");
  }
   delay(100);

   if(distance1 < 10){
    Serial.println("Distance from sensor 2 too low");
   }
   delay(100);
}
```

References

Bansal, P. & Kockelman, K. 2017, "Forecasting Americans' long-term adoption of connected and autonomous vehicle technologies", Transportation Research Part A: Policy and Practice, vol. 95, pp. 49-63.

T. Litman, Autonomous Vehicle Implementation Predictions, Victoria Transport Policy Institute, Victoria, Canada, 2017.

T. Litman, "Autonomous vehicle implementation predictions: implications for transport planning," 2020, https://www.vtpi. org/avip.pdf.

J. Piao, M. McDonald, N. Hounsell, M. Graindorge, T. Graindorge, and N. Malhene, "Public views towards implementation of automated vehicles in urban areas," Transportation research procedia, vol. 14, pp. 2168–2177, 2016.

L. Xiaoming, Qin Tian, Chen Wanchun, and Y. Xingliang, "Real-time distance measurement using a modified camera", in 2010 IEEE Sensors Applications Symposium (SAS), 2010, pp. 54-58.

Q. Wu, Y. Zeng, R. Zhang, Joint trajectory and communication design for multi-UAV enabled wireless networks, IEEE Trans. Wireless Commun. 17 (2018) 2109–2121

H. Shiri, J. Park, M. Bennis, Massive autonomous UAV path planning: A neural network based mean-field game theoretic approach, in: 2019 IEEE Global Communications Conference (GLOBECOM), IEEE, 2019, pp. 1–6.

Z. Ma, C. Wang, Y. Niu, X. Wang, L. Shen, A saliency-based reinforcement learning approach for a UAV to avoid flying obstacles, Robot. Auton. Syst. 100 (2018) 108–118.

A. R. Wagner, J. Borenstein, and A. Howard, "Overtrust in the robotic age," Communications of the ACM, vol. 61, no. 9, pp. 22–24, 2018

Q. Yang, S.-J. Yoo, Optimal UAV path planning: Sensing data acquisition over IoT sensor networks using multi-objective bio-inspired algorithms, IEEE Access 6 (2018) 13671–13684.

F. Yang, X. Ji, C. Yang, J. Li, B. Li, Cooperative search of UAV swarm based on improved ant colony algorithm in uncertain environment, in: 2017 IEEE International Conference on Unmanned Systems (ICUS), Ieee, 2017, pp. 231–236.

Qadir, Z., Ullah, F., Munawar, H. S., & Al-Turjman, F. (2021). Addressing disasters in smart cities through uavs path planning and 5G communications: A systematic review. *Computer Communications*, *168*, 114–135. https://doi.org/10.1016/j.comcom.2021.01.003

H.S. Munawar, A. Hammad, F. Ullah, T.H. Ali, After the flood: A novel application of image processing and machine learning for post-flood disaster management, in: Proceedings

of the 2nd International Conference on Sustainable Development in Civil Engineering (ICSDC 2019), Jamshoro Pakistan, 2019, pp. 5-7.

H.S. Munawar, S. Qayyum, F. Ullah, S.M.E. Sepasgozar, Big data and its applications in smart real estate and the disaster management life cycle: A systematic analysis, Big Data Cogn. Comput. 4 (2020) 4.

F. Ullah, S.M.E. Sepasgozar, C. Wang, A systematic review of smart real estate technology: Drivers of, and barriers to, the use of digital disruptive technologies and online platforms, Sustainability 10 (2018) 3142.

T. Litman, “Autonomous vehicle implementation predictions: implications for transport planning,” 2020, https://www.vtpi. org/avip.pdf.

A. R. Wagner, J. Borenstein, and A. Howard, “Overtrust in the robotic age,” Communications of the ACM, vol. 61, no. 9, pp. 22–24, 2018

Broccoli, A.J., Manabe, S. 1990, "Can existing climate models be used to study anthropogenic changes in tropical cyclone climate?", Geophysical Research Letters, no. 17(11), pp. 1917- 1920.

Murray-Tuite, P. & Wolshon, B. 2013, “Evacuation transportation modeling: An overview of research, development, and practice”, Transportation Research Part C: Emerging Technologies, vol. 27, pp. 25-45

Murray-Tuite, P., Wolshon, B. & Matherly, D. 2017, "Evacuation and Emergency Transportation Techniques and Strategies for Systems Resilience", TR News.

Piao, J., McDonald, M., Hounsell, N., Graindorge, M., Graindorge, T. & Malhene, N. 2016, “Public Views towards Implementation of Automated Vehicles in Urban Areas”, Transportation Research Procedia, vol. 14, pp. 2168-2177

Dreossi, Tommaso, Donze, Alexandre, and Seshia, San- ´ jit A. Compositional falsification of cyber-physical systems with machine learning components. In NASA Formal Methods Symposium, pp. 357–372. Springer, 2017.

S. Aggarwal, N. Kumar, Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges, Comput. Commun. 149 (2020) 270–299.

M. Kothari, I. Postlethwaite, A probabilistically robust path planning algorithm for UAVs using rapidly-exploring random trees, J. Intell. Robot. Syst. 71 (2013) 231–253.

T. Shima, S. Rasmussen, UAV Cooperative Decision and Control: Challenges and Practical Approaches, SIAM, 2009.

X. Gong, S. Chang, Y. Jiang, Z. Wang, Autogan: Neural architecture search for generative adversarial networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 3224-3234.

Llorca, D.F.; Milanes, V.; Alonso, I.P.; Gavilan, M.; Daza, I.G.; Perez, J.; Sotelo, M. Autonomous Pedestrian Collision Avoidance Using a Fuzzy Steering Controller. IEEE Trans. Intell. Transp. Syst. 2011, 12, 390–401. [CrossRef] Energies 2019, 12, 2342 14 of 14

Stéphanie, L.; Vasquez, D.; Laugier, C. A survey on motion prediction and risk assessment for intelligent vehicles. Robomech J. 2014, 1, 1.

Fu, X.; Jiang, Y.; Huang, D.; Huang, K.; Wang, J.; Lu, G. A novel real-time trajectory planning algorithm for intelligent vehicles. Control Decis. 2015, 30, 1751–1758.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y. and Berg, A.C., 2016,October. Ssd: Single shot multibox detector. In European conference on computer vision (pp. 21-37). Springer, Cham.

Dai, J., Li, Y., He, K. and Sun, J., 2016. R-fcn:Object detection via region-based fully convolutional networks. In Advances in neural information processing systems (pp. 379-387).

Ren, S., He, K., Girshick, R. and Sun, J., 2015. Faster RCNN: Towards real-time object detection with region proposal networks. In Advances

Han, J., Liao, Y., Zhang, J., Wang, S., & Li, S. (2018). Target fusion detection of lidar and camera based on the improved Yolo algorithm. *Mathematics*, *6*(10), 213. https://doi.org/10.3390/math6100213

Biographical Information

Shannon Abolmaali was born in Norman, Oklahoma. At the age of 7, she moved to Arlington, Texas. She attended elementary, middle school and high school in Arlington, Texas. After graduating high school, she attended The University of Texas at Arlington and received her Bachelor of Science degree in Civil Engineering, Master of Engineering in Structural Engineering, and PhD in Industrial Engineering with graduation dates of 2016, 2017, and 2022; respectively. After receiving her Masters degree, she began working as a entry-level bridge engineer, at the age of 23, at the Texas Department of Transportation in Dallas, Texas. After working for a year and a half, she decided to go back to school and pursue her PhD in Industrial Engineering in 2019. Her research is utilizing RFID technologies, artificial intelligence, and machine learning to make older vehicles smart.