Autonomous Vehicle Initiative Project: Obstacle Detection System

Proposal Document

Frank Dow 101140402
Joshua Gatto 101150890
Gilles Myny 101145477
Kareem El Assad 101107739

Coordinator: Richard Dansereau

Table of Contents

Project Description

The main objective for this project is to create a vision system for a vehicle that can recognize multiple types of traffic cones, potholes, and barricades through our own vision system. This vision system will be made possible using a method of artificial intelligence called a neural network. We will be training our neural network to recognize our specified objects and perform various tests to ensure everything will work as planned. The main components of the vision system include a high frame rate camera, a test vehicle, and a laptop/processor to run our neural network for the system in real-time.

The end goal of this project is to somehow relay the information gained from our vision system to the vehicle’s control system, so the vehicle can react accordingly. At the moment, we are unsure of how this process will work, but as we get further along in development and by communicating with the group working on the control system, we may get a better understanding of this.

Background

In 2020 Statistics Canada released a list of the leading causes of mortality in Canada, where accidents were listed as number four [1]. We believe that the automation of vehicles could help reduce the number of accidents that happen. According to a study done in France, they have estimated they could halve both the amount of injury crashes and the number of fatal crashes by replacing all vehicles by autonomous vehicles [2]. The first step for us to solve this problem would be to create a vision system that can recognize obstacles on the road and keep our drivers out of harm's way.

Tesla is currently the leader in AI for autonomous vehicles. Tesla uses the method of neural networks for their autopilot system in their electronic vehicles. Tesla neural networks contain 48 networks that take thousands of GPU hours to train. A neural network is made up of several layers that each react differently based on the data received. When an image is processed by the first layer, an input of nodes is inputted. The layer will detect patterns in the image and output the data to the next layer. This process of layering continues until all layers have been processed and the system decides on what the object is [3]. It will not be feasible for our team to implement various layers as the process of making them is extremely difficult and time consuming. The achievable amount is around 2 or 3 layers. We are then able to train our neural network to recognize objects by training it with images of traffic cones, potholes, and barricades.

Objectives

Functional Requirements:

- Able to recognize multiple types of traffic cones, potholes, and barricades through the vision system.
- System should communicate visual events triggered by known objects to other systems on the autonomous vehicle within a safe and usable time frame.
- System should be able to label and mask objects in real time.

Non–Functional Requirements:

- Create and train a neural network on a Google Cloud Virtual Machine.
- System shall recognize objects in a real-time environment.
- System shall identify trained objects with an accuracy level of 90% or higher in object recognition .
- System should be compatible with the vehicle’s control system.
- The neural network should have multiple hidden layers to allow it to represent an arbitrary decision boundary to arbitrary accuracy.
- The neural network hidden layers shall have enough neurons to avoid underfitting and overfitting of signals in such a complicated data set.
- The neural network shall use a rectified linear firing function, as industry standard.
- The neural network shall use a stochastic gradient descent model for the optimizer, typically industry standard along with the “adam” model.
- The neural network shall use a binary categorical cross entropy loss model to calculate loss in neuron hidden layer decision making.
- The system’s development will be tracked in various ways such as the use of Notion - a project management tool - and weekly group meetings with the coordinator.

Timetable for Completion of Intermediate & Final Milestones

Proposal Timeline

Please find attached a live-link to the Proposal Timeline document. We have also attached an image under each section of the timeline but it is best viewed using the link below.
Link:
https://descriptive-binder-341.notion.site/Proposal-Timeline-8247c5d7da774430b12ea5df9a88bd0d

Milestones

Milestone 1: Determine Neural Network Type and Gathering Data for Testing	Multiple Convolutional Neural Networks (CNNs) will be experimented with during this phase. We intend to train and compare different types of neural networks with each other to find an optimal configuration that suits our needs. Gather adequately sized data to create trial training dataset for the following milestone. The photos gathered will be in ideal conditions. Data will be sourced from the internet, empirically, and by simulation. The majority of the empirical data will be obtained from the Carleton University Campus, particularly on Campus Avenue., Library Road., and University Drive.	December 9, 2022	Milestone	☐
Milestone 2: Train Neural Network	After the neural network is chosen, we will split it into two groups. One group will be responsible for cleaning and preparing the training dataset. The other group will be responsible for implementing the neural network and using the dataset.	January 14, 2023	Milestone	☐
Milestone 3: Implement on Prototype Hardware	After training the neural network, we intend to implement it on physical hardware using a Raspberry Pi and a Pi Camera. This will aid in verifying the effectiveness of the trained network in real-world situations.	February 1, 2023	Milestone	☐
Milestone 4: Real World Testing & Simulation	This portion will determine the accuracy of the object detection. Once the network has been trained to 70% accuracy, we intend to put it into a simulated environment to test its effectiveness and consistency.	March 1, 2023	Milestone	☐
Milestone 5: Implement on Final Hardware & Integration on Vehicle	Finally, the system will be mounted onto the vehicle's chassis. The microcontroller will then be connected to the central controlling system.	April 12, 2023	Milestone	☐

Funding

Student Group Funding: Fall SGF Applications Open	Fall SGF Applications Open	September 19, 2022	Funding	☑
CUESEF: Proposal Submission Opens	Proposal Submission Opens	October 3, 2022	Funding	☑
Student Group Funding: Fall SGF Applications Close	Fall SGF Applications Close	October 9, 2022	Funding	☑
Student Group Funding: SGF Funding Decided	SGF Funding Decided	October 16, 2022	Funding	☑
Student Group Funding (SGF): SGF Funding Approved	SGF Funding Approved	October 19, 2022	Funding	☐
Student Group Funding (SGF): SGF Emails Sent & Reimbursement Period Starts	SGF Emails Sent & Reimbursement Period Starts	October 22, 2022	Funding	☐
CUESEF: Proposal Submission Closes	Proposal Submission Closes	October 28, 2022	Funding	☐
CUESEF: Funding Decisions	Funding Decisions	November 7, 2022	Funding	☐
Student Group Funding: SGF Interviews	SGF Interviews	November 10, 2022	Funding	☐
Student Group Funding (SGF): SGF Reimbursement Period Closes	SGF Reimbursement Period Closes	November 12, 2022	Funding	☐
CUESEF: Refund Deadline	Refund Deadline	December 9, 2022	Funding	☐

Project Submission

SYSC4907: Project Proposal Due	Project Proposal Due	October 21, 2022	Submission	☐
SYSC4907: Progress Report	Progress Report	December 9, 2022	Submission	☐
SYSC4907: Oral Presentation Form	Oral Presentation Form	December 9, 2022 12:00 PM	Submission	☐
SYSC4907: Oral Presentation Time Changes	Oral Presentation Time Changes	January 9, 2023	Submission	☐
SYSC4907: Poster Fair Demo Form	Poster Fair Demo Form	January 9, 2023 → March 1, 2023	Submission	☐
SYSC4907: Oral Presentations	Oral Presentations	January 23, 2023 → January 27, 2023	Submission	☐
SYSC4907: Final Report Draft	Final Report Draft	March 14, 2023	Submission	☐
SYSC4907: Poster Fair	Poster Fair	March 17, 2023 9:00 AM	Submission	☐
SYSC4907: Final Project Report Due	Final Project Report Due	April 12, 2023	Submission	☐

Distribution of Tasks per Group Member

Defining the distribution of tasks in this type of project can be difficult due to the various milestones within the project that involve collaborative and independent workflows at different stages of the project term. This section will define the distribution of tasks per group member while also highlighting specific areas that will naturally cause an overlap in workflow. These distributions are bound to change as the project evolves however the team is utilizing a Jira style project management sprint board to track what tasks each team member has worked on throughout the project term to better understand the level of involvement each group member had with various stages of the project.

For Milestone 1 the goal is to gather data for testing and determining the best neural network to proceed with for the project. Each person will research different variations of neural networks, and as a group we create a trial training dataset that will be used for each neural network trial and error case. After picking the possible neural networks to be used, there will be the splitting of neural network trial and error cases four ways, equally per person to tackle.

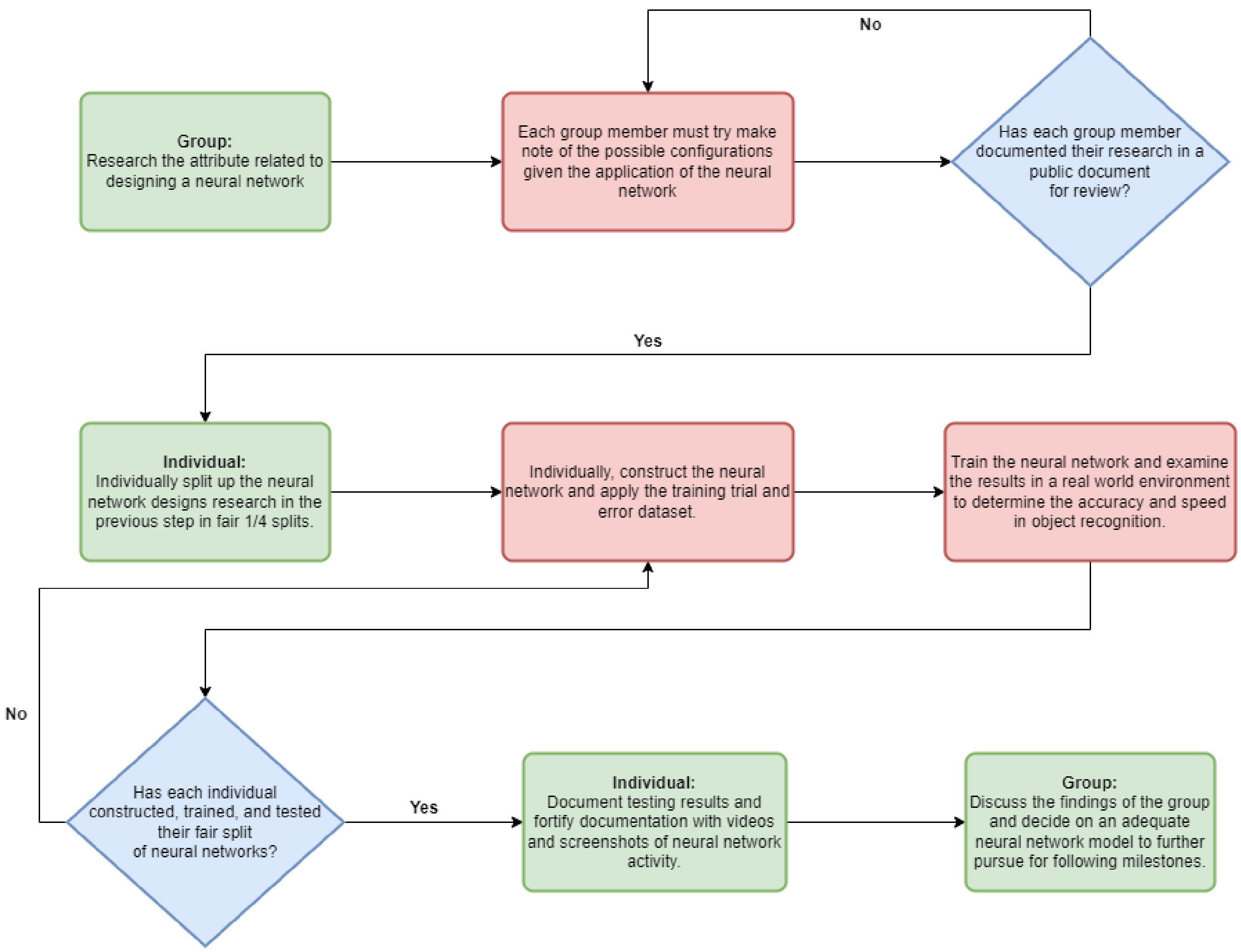

Figure 1: Milestone 1.1 Flowchart

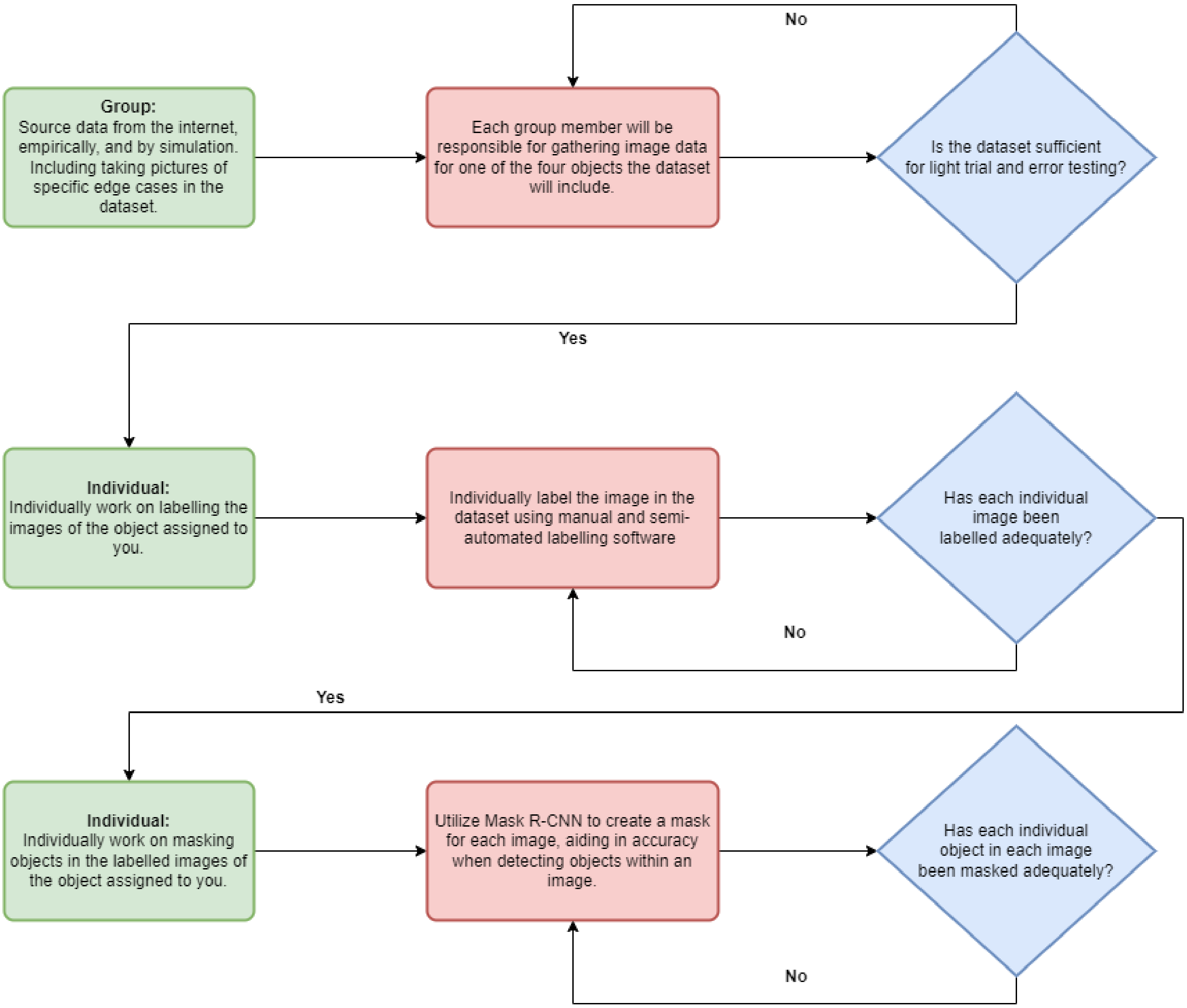

Figure 2: Milestone 1.2 Flowchart

Once a neural network is decided upon, Milestone 2's goal is training the final neural network will be split into two groups, the first group (Gilles, Josh) will create the large training dataset, and the second group (Kareem, Frank) will develop the neural network we've decided on and then run the training in the background.

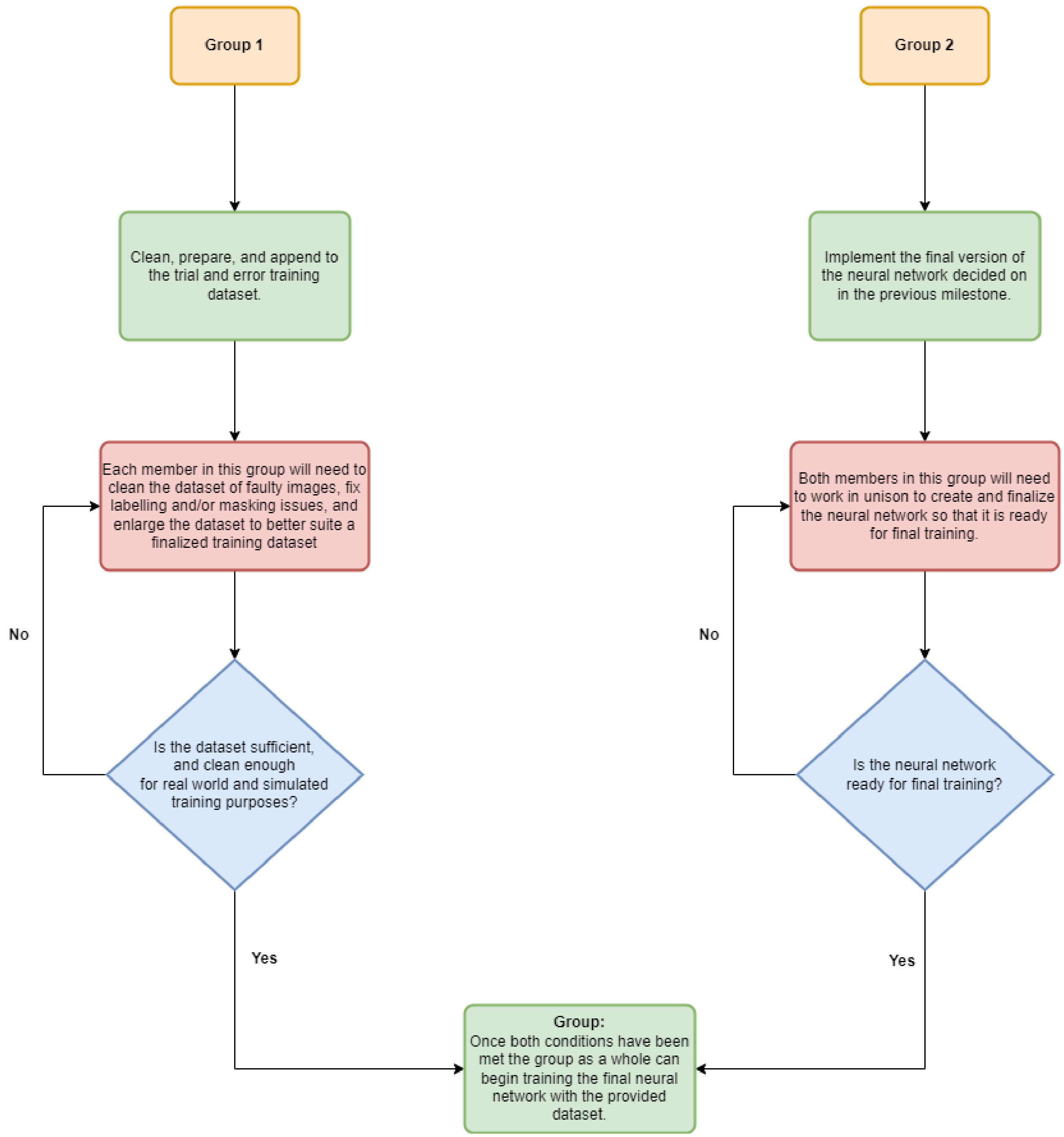

Figure 3: Milestone 2 Flowchart

Milestone 3 will be the implementation phase. As a group we implement the trained neural network on the prototype hardware. This will be done using the specified prototype hardware as described in the Design Decisions and Costs section.

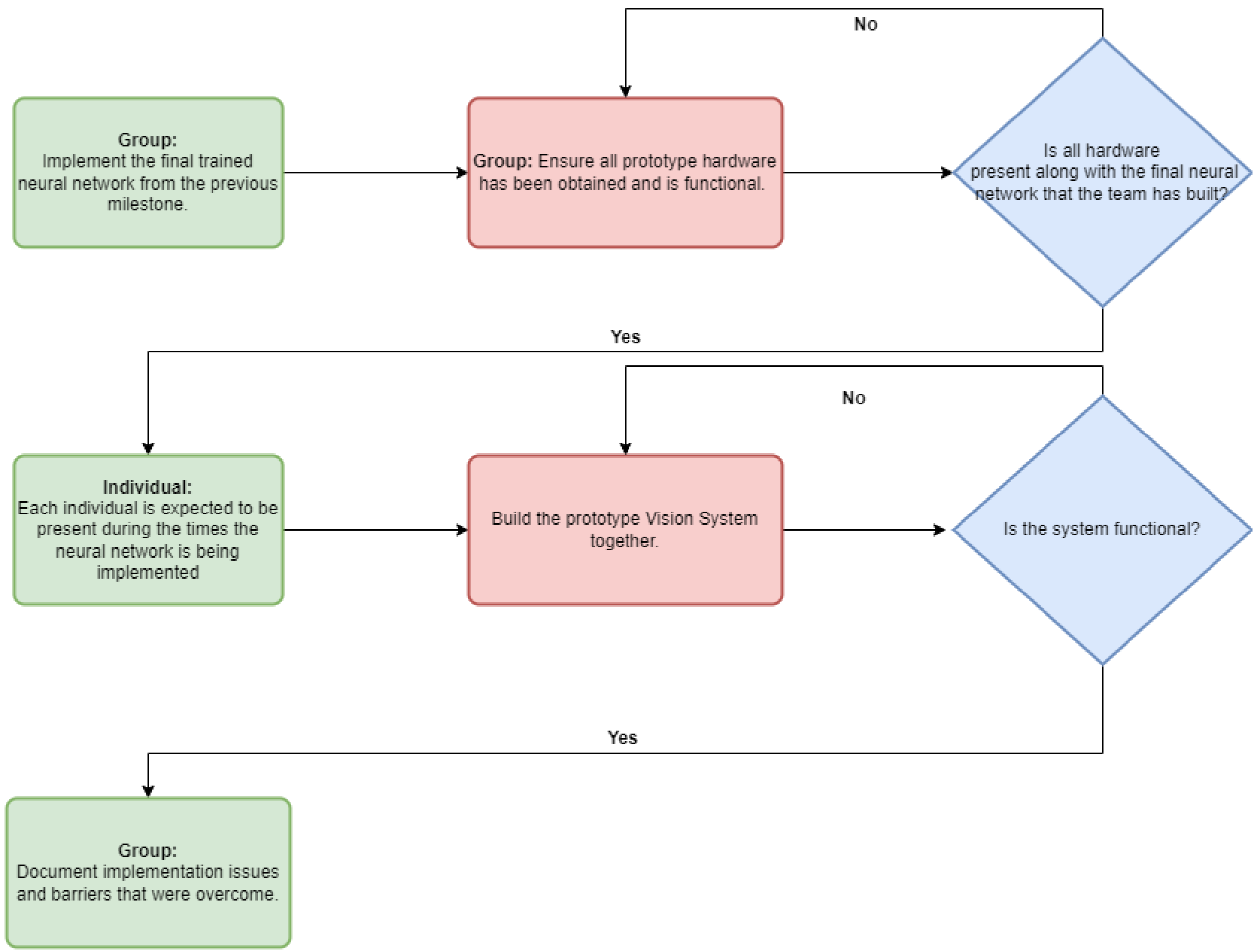

Figure 4: Milestone 3 Flowchart

Milestone 4 will be the testing phase for real world and simulation testing. The group will be split into two groups, the first 2 (Gilles, Josh) will perform real world testing, and the other 2 (Kareem, Frank) will perform simulation of the trained neural network. This testing will be done using specific methods as outlined in the Methods Used to Solve Problem section.

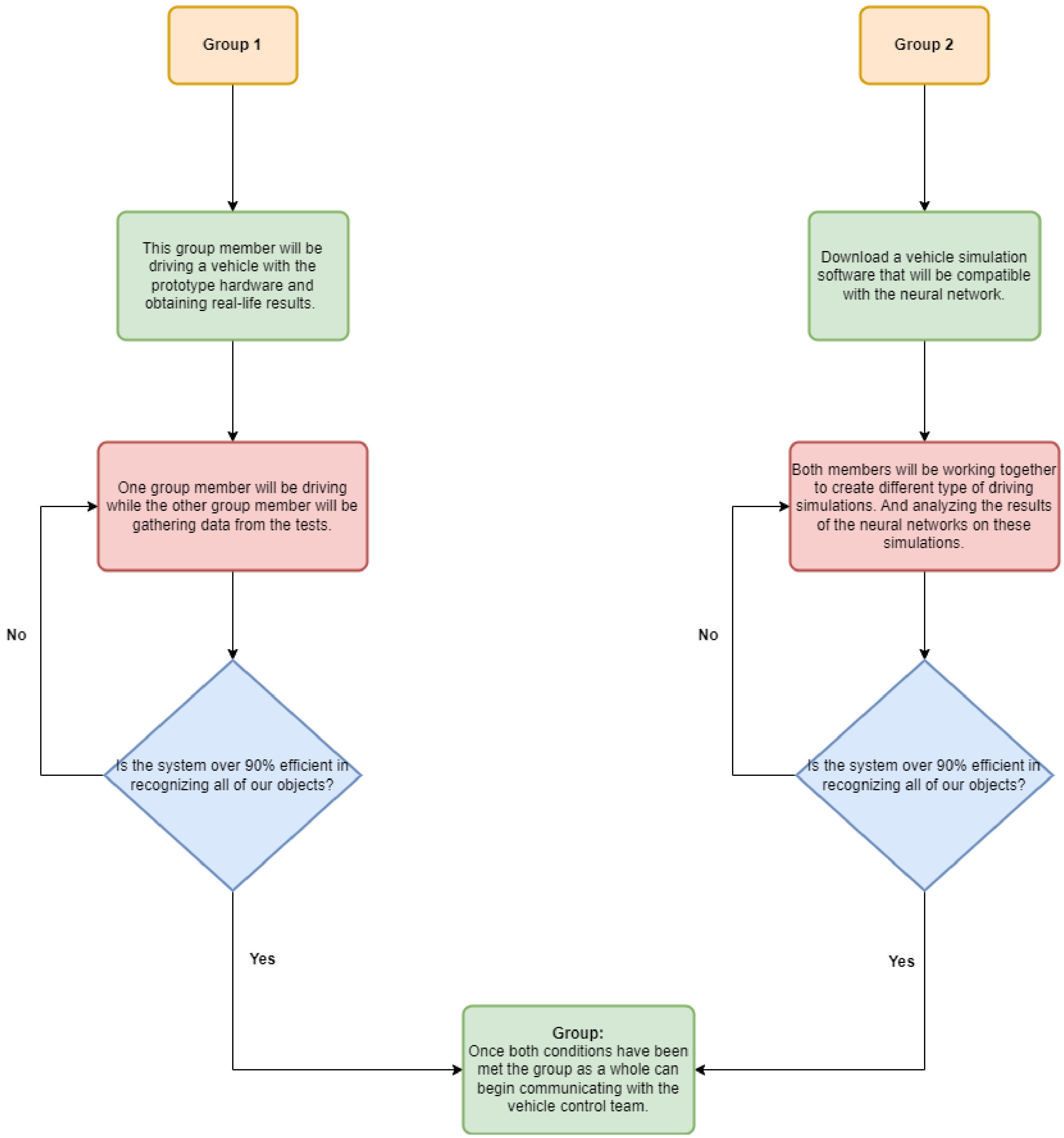

Figure 5: Milestone 4 Flowchart

The final milestone, Milestone 5, will be the implementation of the project on the final hardware and the integration of the system on the vehicle. This will be done as a group and by communicating with all other groups in the autonomous vehicle initiative project, specifically the vehicle control group. Discord will be our means of primary communication but there may also be some in-person meetings. Once everything is working together on the final hardware, the project goals have been completed. If this happens to be sooner than expected there are additional goals outlined in the Additional Optional Objectives for the Project section.

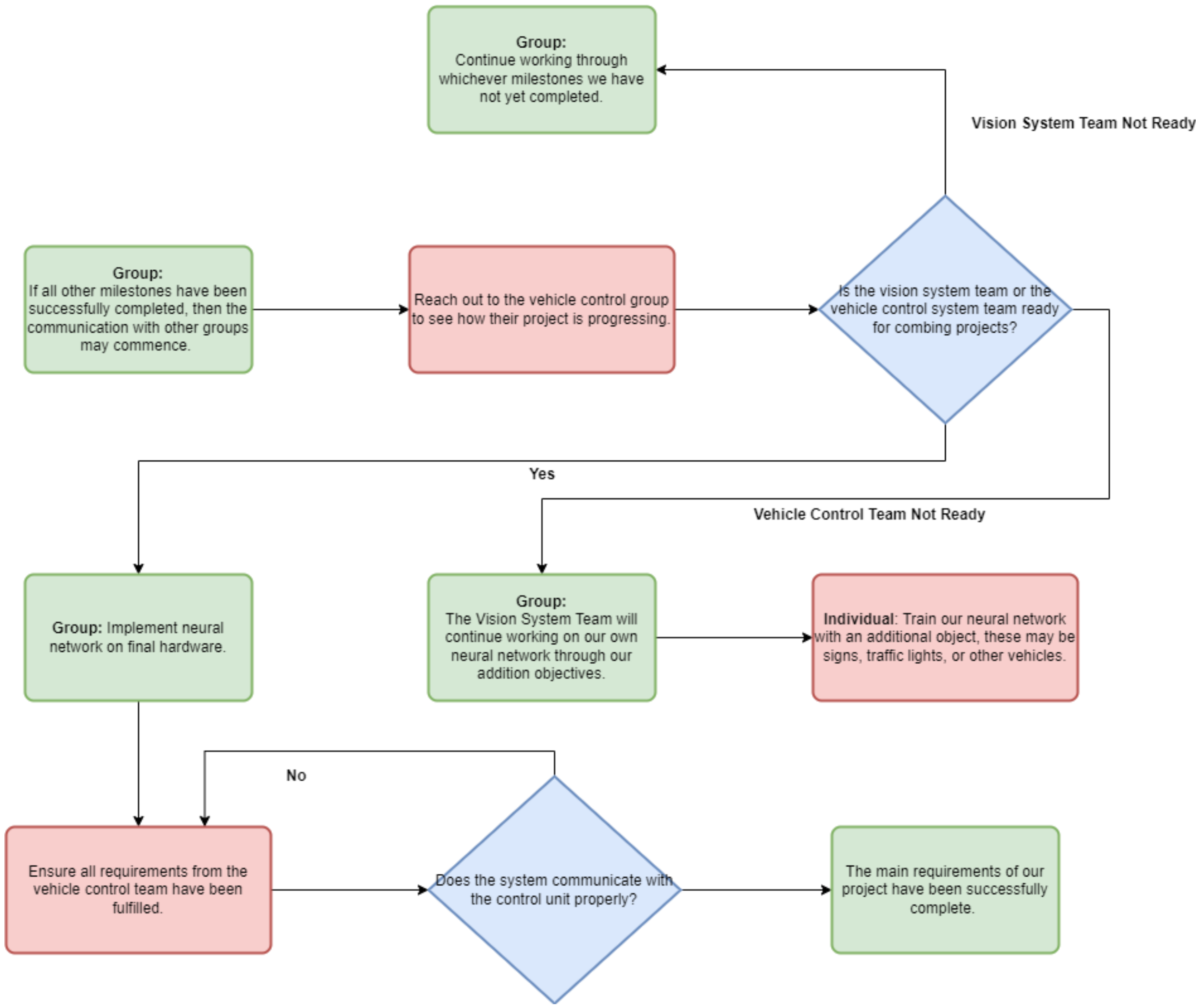

Figure 6: Milestone 5 Flowchart

How Project Relates to Software Engineering Degree

The project will require knowledge previously learned in the software engineering degree. These skills are be categorized by common programming practices, testing and Our team shares a strong background in creating robust and modular systems, seen throughout our degree. Secondly, our team also has a strong understanding of testing requirements and surrounding safety requirements from course work. Lastly, our team has an understanding of linear algebra and thereby the fundamental building blocks of neural networks.

Methods Used to Solve Problem

To design an object detection software, the first task is to develop a dataset to teach the specified object set. The input to the object detection system will be high speed video that will be analyzed frame by frame. For the purposes of training, the algorithm will be fed a series of still images featuring one or more subjects from the specified object set. The subjects will be captured in mostly ideal conditions. That is, the photos will be of subjects in daylight with clear weather conditions. The training data will be of comparable resolution to the final hardware capabilities to be consistent in the granularity of the data being processed. Data will be sourced from the internet, empirically and by simulation. Imagefilter and post processing used in different hardware is not expected to affect the outcome of training as post processing does not change the amount of information present in the image. The empirical data will be captured on Campus Ave., Library Rd. and University Dr. (Carleton University Campus) due to the large density of subjects. Data will also be produced using CARLA simulation software, allowing the environment to be highly controlled for specific situations.

The next objective is to determine which type of neural network (NN) will be most efficient at learning the given dataset. Through experimentation on an Ubuntu based Google Cloud server, several different convolutional neural network (CNN) layer combinations will be trained and compared [7]. The convolutional layer will have a depth of three (3) so that each RGB value may be analyzed independently. The padding around the image, as well as the kernel size will be determined through experimentation. The pooling layer is expected to perform better with max pooling as opposed to average pooling; however, both will be tested to confirm the results for the application of object recognition in an autonomous vehicle environment. One stage and two stage fully connected layers will be compared to determine if the increase in accuracy of a two stage layer could offset the increased processing time. The CNN with the highest average accuracy of object detection across all three (3) objects in the set will be selected for further development.

In parallel with the NN discovery phase, the project will be implemented on prototyping hardware. For this purpose, a Raspberry Pi and Pi camera have been selected. The Raspberry Pi will run Ubuntu[8] like the Google Cloud server. The prototype will aid in verifying the system in a real-time environment. Furthermore, the system will be validated to determine the remaining requirements to be fulfilled. This will involve black box testing of the system in a real world environment. To do so, the system will be mounted on a road-safe vehicle with a human operator. The human operator will drive on the Carleton University campus and the surrounding streets of Ottawa. While driving, the system will pass the video through the neural network where each identified subject will be masked and categorized. The masked video stream will be recorded to an MP4 file alongside the feature extraction data the NN produced. Real-world testing will determine the accuracy of the system in its ability to identify objects. Multifactor statistical analysis will be applied to the data gathered from the testing.

After the NN has been trained to 70% accuracy in object identification, the system will be put into a simulated environment using CARLA simulation software. This testing will act as a verification of the NN before it is implemented on the final project hardware. This implementation in software alone will rule out the possibility of hardware incompatibility or malfunction. The model will test in a number of realistic simulated environments whose challenges coincide with the project requirements. This verification of the NN will determine which requirements may have been overlooked and which should be improved upon. Should there be any unsatisfactory functionalities, the NN will continue training until all requirements are met or exceeded.

Once the model has been trained up to 80% accuracy in object classification, the NN will be implemented on the Nvidia Jetson. To maintain consistency in the integrated application environment, Ubuntu will be installed on the Jetson. The system will then be validated in a real world test environment involving challenge scenarios similar to those presented to the NN in the simulations. The system will undergo the same process involved in testing the prototype hardware; mounting the system to a road-safe vehicle, operated by a human. The operator will drive around and record the masked output and metadata. The testing will act as further verification of the system in a real-time environment on the final hardware. The system will be adjusted as necessary if any bugs or undesired functionalities become present during these tests.

The final objective of the project will be to test the system in the final computing environment. This will involve mounting the camera forward facing on the project vehicle's chassis and mounting the Jetson in a secure manner. The system will be tested in this environment to ensure that all functions are operational and all requirements have been met or exceeded.

Phase	**Hardware**	**Software**
Data Collection	Desktop	CARLA, Ubuntu
Design	Google Cloud Space	Python, TensorFlow library for Python, Ubuntu
Prototype	Raspberry Pi 4, Raspberry Pi High Quality Camera	Ubuntu
Simulation	Desktop	CARLA, Ubuntu
Implementation	daA1440-220uc (CS-Mount) - Basler dart, Nvidia Jetson	Ubuntu
Testing	daA1440-220uc (CS-Mount) - Basler dart, Nvidia Jetson	Ubuntu
Delivery	daA1440-220uc (CS-Mount) - Basler dart, Nvidia Jetson, Project Vehicle	Ubuntu

Additional Optional Objectives for the Project

The solution to the given problem as stated above is the main objective of the project within the given time frame. However, there are many other useful applications of a working neural network model trained for object detection that are applicable when it comes to driving an autonomous vehicle in the real world. These optional objectives will be discussed as follows and what needs to occur in what order before these objectives can become a realization in the delivery of this project.

Road signage plays a key role in the behavior of a vehicle in the real world. Its role is to protect every party utilizing the roadwork infrastructure including your own vehicle, other vehicles, and pedestrians. Being able to detect road signage with our neural network would be a great benefit to keeping the passengers and other people safe, as well as adding more complexity to our object detection neural network. The potential of recognizing signs such as stop signs, railway crossings, do not enter, yield, and the various other signs that are present at lights such as no right turn, and left turn, would be a useful addition to the neural network training dataset.

The other groups involved with the Autonomous Vehicle Initiative project would require a LiDaR (Light Detection and Ranging) camera for their respective tasks. This could benefit our group as the potential use of LiDaR in our project could help improve the accuracy in recognizing the objects we are detecting due to the ability to gauge distance with a LiDaR camera. Being able to inform the central system that there is an object on the road with high accuracy can be supplemented with data points regarding its height and distance relative to the vehicle.

While a vehicle is on a public road it is never the only vehicle there, therefore, being able to detect other vehicles going in the same direction of travel, opposite direction, or perpendicular trying to merge into the road would also be a useful addition to the neural network. Tying the recognition of vehicles into the concept of judging distance to the main objects we are detecting as well as other vehicles can aid in the self driving accuracy.

Being able to complete all of these optional objectives within the given time frame of this project is not possible, however, an opportunity might arise as the project develops to tackle one of the problems. Based on the challenges the team faces throughout the development of our main objective, object recognition of traffic cones, barricades, and potholes, the incorporation of detection for road signage, LiDaR assisted distance calculation, or other vehicles may become a reality. As of writing this proposal it is far too early to tell how many, if any, of these optional objectives will be accomplished, however stating them in this proposal keeps the need for these challenges within scope of the project.

Design Decision and Costs

For the design we require 3 fundamental tools. Firstly, a virtual machine with enough computing power to be able to train the neural networks to be able to detect specific objects with thousands of sample images for each object, obtained from the internet. Second, a processor is needed that will be able to use the neural network created as well as other neural networks from other teams. Lastly, a high frame rate camera is needed, that is compatible with the processor.

For the virtual machine there are two ways of obtaining it. The first way is potentially cost free, as it is the new virtual machines that are being used at Carleton. However, the man in charge of the VMs, Patrick Fairs, had said they are only in the testing phase. The only other option, besides training our networks locally on a computer, is the Google Cloud virtual machines. The Google Cloud's VM comes with your first $300 of processing for free and after that it is pay as you go. It is very difficult to determine how much will be needed, but a best guess is an amount of $1000.

The primary testing will be done using a laptop as the processor, just so it is possible to see the real time decision making that the neural network model is doing. Afterwards, there are three options for a processor. The first option is the Raspberry Pi Model 3 which is $46.95, this would be only able to run a sole neural network and not be able to be used by any other teams. The second option is using a headless computer as the main training board. It would be able to handle the training for a large number of neural networks and would be easily accessible to the other groups. This option will be one of the more expensive options but it would build a solid foundation for the future years. We estimate the cost to be approximately $2000 when factoring in the high cost of premium graphics cards in the market. This option is highly dependent on the amount of funding received from CUESEF. The third option is the Nvidia Jetson Nano which is currently being sold at $500. This would be able to run multiple neural networks for the various teams but sharing one of these between all the teams would be a nuisance as there would need to be booking times to use it. So, it would be ideal to buy both the Nvidia Jetson and the Raspberry Pi. The Raspberry Pi will be used to run our own tests until the Neural Networks are prepared to be combined with the other groups onto the Nvidia Jetson, where everything can be run together. There is a similar solution for the camera as well. Research is currently being done for a LIDAR camera that will be shared between the teams and for our own test purposes we will get a raspberry pi camera that is $35.95.

Hardware	**Cost**
Google Cloud VM	Estimate: $1000
Raspberry Pi Model 3	$46.95
Raspberry Pi Camera	$35.95
Nvidia Jetson Nano	$500
Final Camera (Undecided)	-
Total Cost:	$1582.90

How Each Students Knowledge Gained in Degree Aids Solving the Problem

As each group member is a Software Engineer most, if not all, experiences and knowledge gained throughout the degree will be the same. This section will outline the general knowledge gained throughout the four (4) years of the Software Engineer degree and will also contain sections for professional knowledge gained through co-op / internships for each group member.

Throughout the software engineering course tree there have been many courses that have supplied the group with knowledge and experience to facilitate the successful delivery of the project. ECOR 1051 introduced group based python programming in a project format that ensured Software Engineers were placed as lead members while the other members were of varying engineering disciplines. This allowed the software engineering team leads to be exposed to a group structure where every member may not have the same level of experience or knowledge when it comes to the CI/CD practices that are involved in software projects. SYSC 2004 and 2006 introduced the group to fundamental concepts of memory management and test-driven development using Java and C languages. SYSC 3303 demonstrated the use of real-time concurrent systems as well as performance based programming fundamentals to ensure time and memory complexity is at its lowest for quick functioning applications. Lastly, SYSC 3120 and SYSC 4106 taught us on defining engineering requirements in a software based setting as well as software project management and economics, learning about software lifecycle, cost value comparisons and Agile workflows. All of these courses will enable the group to perform the project in ways demonstrated through our learning within the last three (3) years.

Professional knowledge and experience gained through co-op / internship terms will also aid the group throughout the project. Below we discuss each of our professional experiences in detail.

Kareem has had four (4) past internships in a professional setting. He interned twice at Bell Canada, once at Ciena, and once at Nokia. Throughout his time at Bell and Ciena, Kareem worked primarily in Python, developing and designing a REST API for automation. He learned how to design, test, and build a Kafka-integrated system from the ground up. At Nokia, Kareem worked as a Data Engineer. His role involved a large amount of SQL and Python. He learned how to work with industry-sized databases and perform operations on them effectively. He was responsible for migrating various databases to OLAP (online analytical processing) systems such as Clickhouse. The teams incorporated an Agile Methodology with daily stand-ups and weekly sprints. Throughout all his internships, he used version control software(Git) as well as Issue and Project Tracking software(Jira). Kareem hopes to integrate the theoretical knowledge acquired from his classes with the practical experience gained during his internships to ensure a successful project.

Gilles has had three (3) past internships in a professional environment. Firstly, he worked on a financial reporting system developed in Python. This system was designed to be expanded as business spending and budget habits changed over each financial quarter. Maintaining a well functioning CI/CD workflow and base layer of programming ensured other employees took over the project with ease as the internship term ended. Using libraries such as Pandas Dataframes and NumPy within a Jupyter Notebooks framework translates over well as the group is planning on training the neural network within a Jupyter framework due to its ease in visualizing data. Furthermore, he worked on a SQL heavy database using Microsoft Azure to manipulate data points coming from a data lake producing over one million data points per day to distribute them to various systems and visualize the data points in a manner that made it easy to track and analyze. Lastly, he worked on in-house tool development at an automotive fleet management solutions company using Python and C++ languages. In a collaborative environment with many developers working on the same thing, Git was used for CI/CD purposes while Jira was used for project management using a 2 week Agile sprint development strategy. All of his past professional experiences will translate over well for this project when it comes to professional development practices and project management strategies.

Frank has previously worked on a team to create a text-based monopoly game. This has given Frank the experience of working on a programming team collaboratively using Github. Github will be an essential tool for the team, working together and having prior knowledge of this platform will be of great benefit. Frank has also had two (2) previous internships. For his first internship he was delegated tasks weekly to update changes on a school website using HTML and manage the student data for the website. These tasks were always completed on a weekly basis which benefits Frank with time management skills in order to help with the challenges faced in order to stick to a proposed timeline. Frank's second internship's focus was on project management and database management. Project management will play an important part of the project as a whole, with the delegation of tasks, setting clear and concise objectives, risk management, and quality assurance. Database management will also be an important part of the project as managing the input data and output data of the system is vital.

Joshua has gained knowledge relevant to the Autonomous Vehicle Initiative Obstacle Detection System project by working on other software projects at Carleton including a web store, monopoly game and elevator logic system. These projects involved using and modifying software design patterns to fulfill the requirements, ensuring the developed software is robust in nature. Joshua also learned about project management in his learning and experiences. Proper task delegation and group member management are vital to keeping a successful project's timeline. Joshua has years of experience in testing code and can apply modern standards in industry. Joshua is aware of safety related design practices and the consequences of not following basic safety standards.

Project Risks & Mitigation Strategies

Object detection within an autonomous vehicle platform presents many risks that can easily be overlooked or forgotten, leading to significant financial, property, trust factor, and in extreme scenarios, human life. Object detection using a deep learning approach - when multiple hidden layers of neurons are incorporated in the neural network model - must be treated as a test platform, unsafe for real world use until thousands of hours of testing, performance tuning, and loss measurements have taken place. The following sections will highlight the risks associated with the undertaken project, and how the team will maneuver them to mitigate not only these risks, but also the losses involved.

All failed attempts and success stories involving object detection using a deep learning neural network all have one thing in common, the accuracy measurement issue. The complexity of a neural network model greatly depends on the given input(s) and expected output(s) of the system. There are many options to choose from when designing a neural network model, such as what firing function, optimizer, loss calculation model, number of hidden layers, and number of neurons in each hidden layer to use. All of these options will in essence decide the level of accuracy the neural network can associate with its decisions, and subsequently will determine the amount and types of risk the project will create and encounter.

With semi and fully autonomous vehicles being the new rage of the 21st century the world has seen a lot of stories about said vehicles not avoiding a fender bender, swerving out of the lane, and even mistaking pedestrians and blockades as something else or simply non-existent. In most of these cases there is one thing to blame, and it’s the level of accuracy the automotive manufacturer has deemed “acceptable” for real world use. For example, in the image below [4, Fig. 7], you will see that the object detection neural network has detected with “good” accuracy, 93.3% accuracy to be exact, that the front bumper of the oncoming car is a pothole, as well as it recognizing an actual pothole on its right.

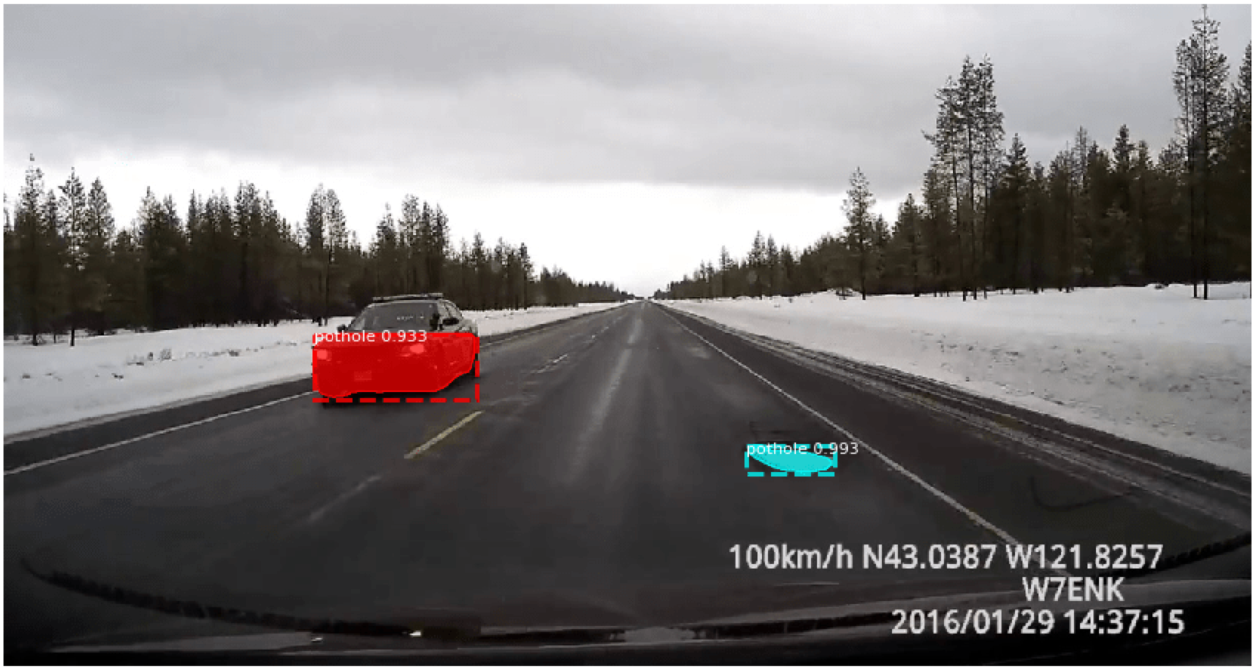

Figure 7: False accuracy with pothole detection neural network.

In this case the car using object detection would either have to swerve right into the shoulder or between the two potholes putting the oncoming car into danger, or even worse come to an emergency stop putting the driver and all following cars at risk. This complete failure of object detection can be a result of multiple decisions made during the development stages of this product. When training a neural network, the size, realism/condition, and labeling method of the data set(s) you expose the neural network to will impact the way it perceives real time data in the future. For example, if you train a model with a data set of thousands of images, taken at different angles, lighting conditions, and visibility (or lack thereof), it will have been exposed to so many different scenarios, similar to the saying “wisdom comes with age”, where age in this case is a large and diverse data set during training.

This project is about taking what the world has publicly made available through years of research, trial & error, and success and culminating it into the use case of this project. Risks will be involved and actively discussed throughout this project, and more importantly how the team can mitigate these risks to create a test ready platform for future years. In the early stages of this project the team has researched various methods on reducing the risk for false positive detections. Labeling, masking, and the creation of an accurate, diverse, and sizable dataset are all methods the team plans on utilizing to mitigate the risks discussed. Firstly, the creation of an accurate, diverse, and sizable dataset takes a lot of time and precision. Creating and finding real world pictures of the obstacles we want to look out for is a lengthy process that is required to build said dataset, it involves having pictures of the obstacles at various angles, road conditions, and lighting conditions, and more importantly is challenging the neural network to be able to locate these objects in adverse image conditions. Furthermore, labeling the images by hand is mandatory when it comes to training the neural network. Labeling is often done by encircling the desired object with a frame and title to show the neural network what it can expect as inputs. The image below [5, Fig. 8] gives an example of label framing on a set of balloons.

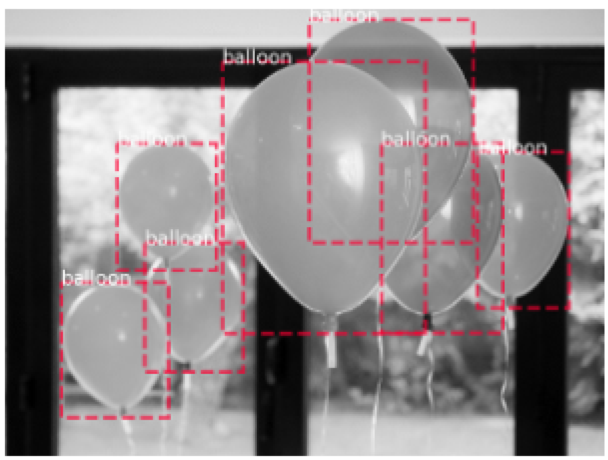

Figure 8: Labeled image of various overlapping balloons.

Lastly, masking is also a very effective way of training the neural network to be more precise with the exact geometry of objects it will encounter. This approach is typically paired with labeling to further enhance the neural networks learning. This is done by filling the object to detect in a specific color so that the neural network can get a scale of the object and its resolution. The image below [6, Fig. 9] also gives an example of masking on a group of people and aircraft in the background.

Figure 9: Labeled and masked image of a group of people and airplanes.

In conclusion, there are many risks involved with object detection on an autonomous vehicle platform as described above, and there are many methods that can be applied so that these risks can be mitigated. Labeling, masking, and the creation of an accurate dataset are all actions that can be taken during the training of the neural network, but this does not mean that the risks suddenly disappear. Performance tuning the neural network model architecture after training is complete can also help mitigate the risks of a false positive or lack in accuracy when it comes to detecting objects in a real world environment.

Funding Requests

Forecasted Equipment Expenditures

Please find attached a live-link to the Forecasted Equipment Expenditures document. We have also attached a PDF sample of the timeline but it is best viewed using the link below.
Link:
https://descriptive-binder-341.notion.site/Proposal-Forecasted-Equipment-Expenditures-d3e6d5d265f14ba59bf76b914db60dfa

Funding Information

Please find attached a live-link to the Proposal Timeline document. We have also attached a PDF sample of the timeline but it is best viewed using the link below.
Link:
https://descriptive-binder-341.notion.site/Proposal-Funding-189c465fd72a40ae8a0e29d378f4bacd

References

[1] “Leading causes of death, total population by age group,” *Statistics Canada*, 24-Jan-2022. [Online] Available: https://www150.statcan.gc.ca/t1/tbl1/en/tv.action?pid=1310039401. [Accessed: 1-Oct-2022]

[2] C. Pilet, C. Vernet, and J.L. Martin, “Estimated crash avoidance with the hypothetical introduction of automated vehicles: a simulation based on experts’ assessment from French in-depth data,” *European Transport Review*, 20-Dec-2021. [Online] Available: https://etrr.springeropen.com/articles/10.1186/s12544-021-00521-2#:~:text=We%20estimate%20that%20highly%2Dautomated,56%25%20of%20fatal%20crashes. [Accessed: 1-Oct-2022]

[3] S. C. Wang, “Artificial Neural Network,” Interdisciplinary Computing in Java Programming, 1-Jan-2003. [Online] Available: https://link.springer.com/chapter/10.1007/978-1-4615-0377-4_5. [Accessed: 3-Oct-2022]

[4] A. F. Gad, “Object detection using mask R-CNN with TensorFlow,” Paperspace Blog, 09-Apr-2021. [Online]. Available: https://blog.paperspace.com/mask-r-cnn-in-tensorflow-2-0/. [Accessed: 18-Oct-2022].

[5] B. Liu, “Using object detection to find potholes,” SpringML, Inc., 18-Aug-2022. [Online]. Available: https://www.springml.com/blog/using-object-detection-potholes/. [Accessed: 18-Oct-2022].

[6] W. Abdulla, “Splash of color: Instance segmentation with mask R-CNN and TensorFlow,” Medium, 10-Dec-2018. [Online]. Available: https://engineering.matterport.com/splash-of-color-instance-segmentation-with-mask-r-cnn-and-tensorflow-7c761e238b46. [Accessed: 18-Oct-2022].

[7] R. Chauhan, K. K. Ghanshala, and R. C. Joshi, “Convolutional Neural Network (CNN) for Image Detection And Recognition,” *IEEE Xplore*, 2018. [Online]. Available: https://ieeexplore.ieee.org/document/8703316. [Accessed: 19-Oct-2022].

[8] J. Bellows, “Comparing Linux Operating Systems for the Raspberry Pi 2 ,” *Winona State University: Department of Computer Science*, 27-Apr-2016. [Online]. Available: https://cs.winona.edu/cs-website/current_students/Projects/. [Accessed: 19-Oct-2022].