Fall 12-8-2023

Privacy Within Autonomous Vehicle Cameras

Joshua Montgomery

PRIVACY WITHIN AUTONOMOUS VEHICLE CAMERAS

by
Joshua Terrell Montgomery

A thesis submitted to the faculty of the University of Mississippi in partial fulfillment of the requirements of the Sally McDonnell Barksdale Honors College.

Oxford, MS
December 2023

Approved by

_____
Advisor: Dr. Charles Walter

_____
Reader: Professor Jeff Lucas

_____
Reader: Dr. Nathan Oakes

ABSTRACT

In recent years, cameras have become ubiquitous in daily life, constantly surveilling, and taking in information. This leads to a potential security risk of the invasion in one's privacy without their knowledge or any ability to prevent the privacy threat. While cameras alone are an issue, they are often only in locations where a user has some expectation of a loss of privacy, such as public locations with security systems. However, systems that rely on cameras to operate correctly, including autonomous vehicles, are becoming a more prominently used technology while often appearing in places where an average person has some expectation of privacy. With this technology, comes the use of many different sensors that collect numerous amounts of data, which can be used for malicious intent. As these technologies advance, security systems must advance to keep pace, protecting the security and privacy of users. In this thesis, I show that it is possible to increase security and, more specifically privacy, within autonomous vehicles. I used the Webots simulation platform, a well-known tool for simulation of autonomous robotics and autonomous vehicle systems. Webots allows a simulation that utilizes the same sensors autonomous vehicles currently use while allowing flexibility in the development of security and privacy methods. Using Webots, I created a system that uses a man-in-the-middle attack as a defense mechanism, intercepting data streaming from the camera, doing initial processing, and returning only textual information about what the camera sees, preventing any information that could cause a privacy violation from making it off the vehicle. I utilize an e-puck robot with controller code that represents the on-board computer used to make decisions within an autonomous vehicle and my

solution of a “privacy preserver” used to privatize information received from the camera sensor to help navigate through a maze with colors simulating objects to be categorized. I test this solution to ensure that all information utilized by the on-board computer has been sufficiently anonymized. I show that, through simulation, the “privacy preserver” concept is viable through simulation

TABLE OF CONTENTS

CHAPTER I: INTRODUCTION

Technology has always been a part of life. Whether it was during the Stone Age with the creation of stone tools, to the first Model T vehicle invented by Henry Ford in 1908, and finally, to the 21st century, where we have vehicles capable of driving themselves, we are living in a world where technology is all around us. While technology's rapid advancement is often seen as a benefit to society, these benefits come with a number of caveats. For example, there are cameras everywhere, each used for different purposes, including security, photogrammetry, and autonomous vehicle detection. Without properly handling the information received through these camera streams, several security and privacy risks arise.

Users have an expectation of privacy throughout their lives, with a low-level expectation in public and a higher level in places like our homes, restrooms, etc. However, with this increase in technology, the reality of user privacy is being impeded, whether intentionally or unintentionally. For instance, Google's use of imaging technology, such as their Street View cars, which "has helped…capture more than 10 million miles around the world," and their Trekker, which is a "portable camera system [that] can be used as a backpack…[enabling them] to collect imagery in narrow streets or in places [they] can only reach on foot," is an example of the large amount of data being captured through camera sensors [1].

Figure 1: Google's Street View Car [1].

Figure 2: Google's Trekker [1].

In addition, Google Earth is an open-source software that allows its users to search for different locations and see these places through the help of satellite and on-the-ground camera imaging technology. With this software, users can get a street view of any location they would like to look at closer. Years ago, when Google's Street View feature launched, most of the images were clear pictures of that location, which gave people the ability to see and possibly recognize people

or car tag numbers. This poses a very serious security problem and is an invasion of privacy. To mitigate this issue, Google, now, tries to blur images that contain this content of human faces, vehicle tags, or any personal identifiable information (PII). However, even this method of privacy protection has its flaws. Since the blur effect is added after the picture is stored, some images could have been missed or are not fully blurred, posing a very important security threat. The same goes for closed circuit television (CCTV) cameras used for surveillance and public safety. However, when they are placed in areas where they can see inside private locations, privacy risks begin to arise.

Unlike the previous examples with their efforts in privatizing image and video feed by blurring the input stream after being captured by the camera sensors, my solution to this ever-increasing problem is to hide what the camera sees before it is sent to its cloud storage destination, or any decisions are made. To show proof-of-concept, I examine this solution in the context of autonomous vehicles.

In this thesis, I discuss the concept of privatization within autonomous vehicles through a man-in-the-middle style of attack. Chapter 2 of this thesis consists of the background and related works. Chapter 3 goes into detail about the problem of privacy and my solution in fixing this issue. Chapter 4 discusses my choice of a simulator to aid in proving this concept of privacy maintenance and setting up of the experiment. Chapter 5 is about the experimentation. Finally, Chapter 6 is my conclusion and future work.

CHAPTER II: BACKGROUND & RELATED WORK

Autonomous vehicles traditionally utilize machine learning techniques to create a system to both recognize objects and make decisions on how to safely navigate the world. Looking at autonomous vehicle image classification, Ramakrishnan [2023], shows a powerful method for autonomous vehicle image classification. They utilized a convolutional neural network to categorize objects from a vehicle camera; meanwhile, they showed they could train the model to be 90% accurate at object identification in real time. While this is a very good result, a high accuracy at automatic identification can lead to potential privacy violations if the recognized objects include individuals. However, their model could be utilized as a recognition system for my proposed solution.

One major issue with recognition systems for autonomous vehicles is the risk of an adversarial attack against the recognition models. This can be done with as little as a custom created patch placed on objects in the environment. For example, DiPalma et al. [2021] showed this in a proof-of-concept attack against an autonomous vehicle in development in China. They were able to perform an attack that caused the vehicle to not recognize an obstacle and crash directly into it. This is a potentially devastating attack, though one that can be mitigated with additional work on recognition systems.

While limited, there has been some work on privacy in camera systems. Specifically, smart camera sensor networks, which autonomous vehicles are a part of, have significant privacy and trust concerns associated with their use. One solution [9] shows that blacking out pixels that

may contain sensitive information is enough to protect the privacy and increase trust of people who are recorded by these camera systems. While this is an effective strategy, it is difficult to implement in autonomous vehicles as it removes information from the input feed, limiting the effectiveness of the vehicle itself.

To attempt to alleviate this issue, Brea and Khandeparkar [2023] examine a method to separate, categorize, and privatize autonomous vehicle camera sensors to help protect data from adversarial attacks. Their method is conceptually similar to my method, as they are focused on ensuring that the data anonymizes all non-essential information from the camera feed. However, unlike my method, they focus on leaving some of the data in place and, while they attempt to prevent adversarial attacks through their method, their method is still susceptible to attacks against the anonymizer resulting in a lack of anonymization.

CHAPTER III: PROBLEM DESCRIPTION & SOLUTION CONCEPT

Technology is more than hardware, software, and computer concepts and theories. Now, in this day and age, technology is a way of life. Technology is everywhere, from the increased use of Artificial Intelligence to the use of cameras for security and autonomous vehicle environment detection. Although technology is meant to make life more efficient, there are still some unintended consequences that stem from its use. For example, cameras may be used to reduce crime by capturing criminals in the act to later be caught by law enforcement or in autonomous vehicles, which are becoming more prevalent, to guide the vehicle through its physical environment and to aid in making logical decisions based on what is seen. However, using these technologies can lead to many security issues, one of the most clear being an invasion or lack of privacy. Figure 3 displays a generalized topology of an autonomous vehicle, which consists of a collection of sensors, an on-board computer, a connection to the CAN network to control the vehicle, a cell service connection to send and receive data, and a connected cloud computing system, where it stores data and provides additional data for subsequent processing.

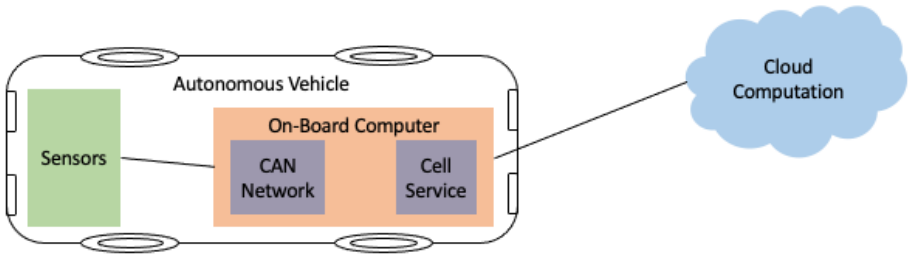

Figure 3: A simplified version of an autonomous vehicle topology.

While this is a powerful method of control for autonomous vehicles, it does come with several privacy concerns. First, if a malicious actor can gain physical access to the vehicle, they are capable of intercepting communication between the sensors and the on-board computer through either a Man-in-the-Middle (MitM) or eavesdropping attack. Using a MitM attack, they can tamper with sensor values, pass them to the on-board computer, and cause the computer to make decisions that are incorrect at best and harmful at worst.

Another possible privacy issue lies between the connection of the cell service, which allows for incoming and outgoing phone calls, location monitoring, and the ability for the cloud computing system to connect to the autonomous vehicle, and the on-board computer. There have been instances when a malicious actor was able to determine the phone number of the car, call it, and input a sequence of tones that caused the vehicle to accelerate uncontrollably and ultimately crash [3]. Lastly, the connection between the on-board computer and the cloud computing system contains both general and privacy interception risks.

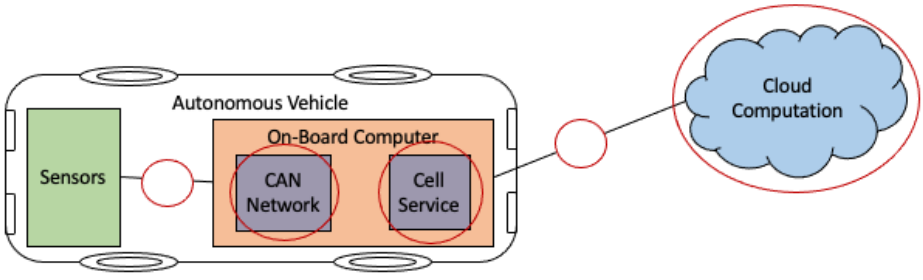

Figure 4: Points in the autonomous vehicle topology where privacy concerns pose a possible issue.

Finally, when sensor data is stored in the cloud for later analysis, it is often not modified to protect privacy, allowing an attacker with access to the cloud storage facility the ability to view and steal information about those captured by autonomous vehicles.

What are some ways to mitigate these privacy concerns within autonomous vehicles? One possible solution is allowing the on-board computer to analyze, make decisions on, and anonymize the sensor data. While this does protect the data from interception to the cloud and when stored on the cloud itself, it does not provide a guarantee of actual privatization of sensor values, but more of a "I will try" approach. Also, this approach of allowing the on-board computer to analyze, decide, and anonymize will require a high computational process power, which is not an efficient method both computationally and for privatizing the sensor data. Finally, any direct on-board eavesdropping/MitM approach will result in un-anonymized data being collected/analyzed by an attacker.

To attempt to solve the issue of physical access for an attacker, the anonymization of sensor data would need to take place before the sensor data is placed on the CAN network and, thus, before it is received by the on-board computer. This method pushes the responsibility of the privacy preservation off the main computer system and onto a connection from the sensors. This method is similar to a MitM attack, placing an invisible computation system between the sensors and the on-board computer whose job is to collect sensor data, anonymize it, and pass that data to the on-board computer.

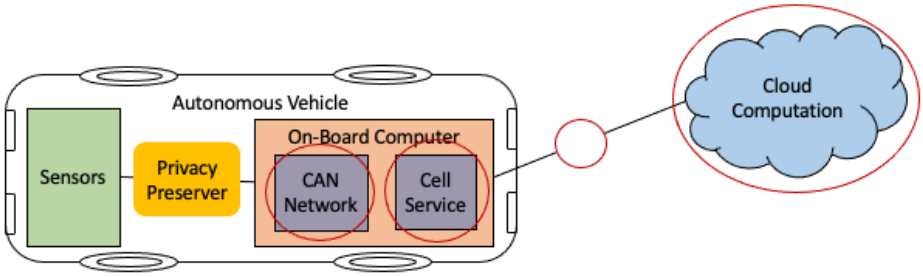

Figure 5: The autonomous vehicle topology with an added “privacy preserver”

Using this method allows for a one-way transit of information to the on-board computer cutting and reducing the privacy issues in the connections of the cell service, CAN network, and cloud computing system to the on-board computer. However, this man-in-the-middle raises the issue of the “privacy preserver” now having to be a perfect implementation, where we have to verify the accuracy of the results. While this is a major drawback of this security measure, it is out of scope for this project.

Utilizing the MitM method to provide additional privacy is good in theory, but there are a number of possible methods to produce an anonymized image. One option is to blur the camera input data. This method would mean that the on-board computer could, in theory, not need any modification to its general function, as it would receive a video feed it could analyze, just with some information blurred. Unfortunately, this is not enough for processing and determining what an object is, for example a blurred person in a red shirt looks a lot like a blurred stop sign. Another privacy option is to use Adversarial Machine Learning (AML) techniques to add small amounts of noise to the image to limit automatic recognition of users in images. However, this also requires high computational power and is not applicable in real time, which is not viable. In addition, it does not solve the privacy problem in a long-term scope.

One more privacy option and an important one to note is to cut the video entirely, performing object recognition on the privacy preserver, and returning the values of the objects, including the position, coordinates, and category of the object possibly as a string or integer value. For example, in the field of view, the rectangle of top left (X, Y) and bottom right (X, Y) contains the object "Person." This option of privacy reduces the load of the on-board computer by letting the "privacy preserver" handle the analysis and anonymization of the sensor data and only allowing the on-board computer to make a decision based on the received privatized or categorized information. Since, one cannot access the raw information within the man-in-the-middle from the cloud computing system, cell service, and CAN network, the privacy issues previously mentioned are either eliminated or reduced.

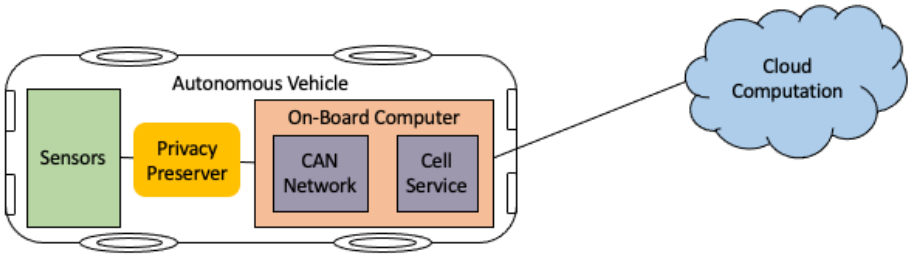

Figure 6: The autonomous vehicle topology, where the points where privacy was a possible issue, is now either reduced or eliminated due to the "privacy preserver."

To get access to the raw sensor data, one must be physically connected to the autonomous vehicle making this solution a viable method in maintaining the concept of privacy within this technology. With this high level of abstraction in a possible solution, how are we able to show its viability? We can use the robot software and simulator Webots in displaying this concept.

CHAPTER IV: CHOOSING A SIMULATOR & SETTING UP THE EXPERIMENT

When first starting my research, my focus was on tampering with autonomous vehicle sensors and functionality. While performing experiments on real autonomous vehicles would be ideal to show proof-of-concept, it was infeasible for this work. This led me to focus on finding autonomous vehicle simulators to show that the concept of ensuring privacy was able to be implemented. Initially, I examined a number of different simulation software that allows for robot and autonomous vehicle testing. I first examined using the Robot Operating System (ROS) paired with Gazebo to help simulate these adversarial attacks and some possible solutions.

Figure 7: The Robotic Operating System logo [4].

ROS is an industry standard, “open-source development kit… [used by] developers across industries [for] research and prototyping [2].” In addition, ROS contains different tools and libraries that allow you to create different robotic simulations with the ability to run on

multiple different systems. Gazebo is another open-source software that allows one to create 3D robotic simulations with great accuracy and efficiency. It is made to have very good ROS integration, making it the simulator of choice for ROS simulations in 3D environments.

Figure 8: Gazebo simulator logo [5].

For the initial experimentation, the idea was to build a simulated world within Gazebo and use ROS as the medium of communication for the man-in-the-middle sensor input and output. However, upon trying to download the ROS software, I was unable to get ROS to run on my local system and had to find another simulator for my research.

Fortunately, after browsing for different simulators, I found Webots, "a professional mobile robot simulation software package [that] offers a rapid prototyping environment for [creating] 3D virtual worlds with robots [equipped] with a number of sensors and actuator devices…[and] is well suited for research and educational projects related to mobile robotics," for instance "mobile robot prototyping in academic research and the automotive industry [11]."

Figure 9: Webots robot simulator logo [6].

After successfully installing Webots, I started my initial experimentation on using the sample autonomous vehicle world that simulated an autonomous vehicle in freeway traffic using the Simulation of Urban MObility (SUMO) package software, that is an "open-source traffic simulation suite [that] allows modeling of intermodal traffic systems - including road vehicles, public transport, and pedestrians [12]." In this simulation, I tried to show viability for my man-in-the-middle defense system by conducting a man-in-the-middle attack that would force the vehicle to crash; however, after attempting this attack, because of the processing power needed to take account of the user input, the simulation would often freeze, not allowing me to see the results of my efforts. Because SUMO required additional input to allow for the man-in-the-middle attack to run, it was deemed non-viable for my solution.

Since SUMO did not work, I made a shift to create my own simulated world, and I chose to use the e-puck robot with a recognition camera attachment. In this new simulation, I created 6 solid spheres, 3 red and 3 green, and placed them in a circle around the e-puck robot. I chose the color blue for the walls because at that moment, it was the color used in the tutorials while learning

how to use Webots, and in the case of the spheres, I opted for the complementary colors, red and green, so the camera could distinguish the difference between the two and perform the specified actions. With the added recognition camera attachment, I had it set to only detect the color of the centermost pixel to save on computational time for testing the viability of privatizing the input stream.

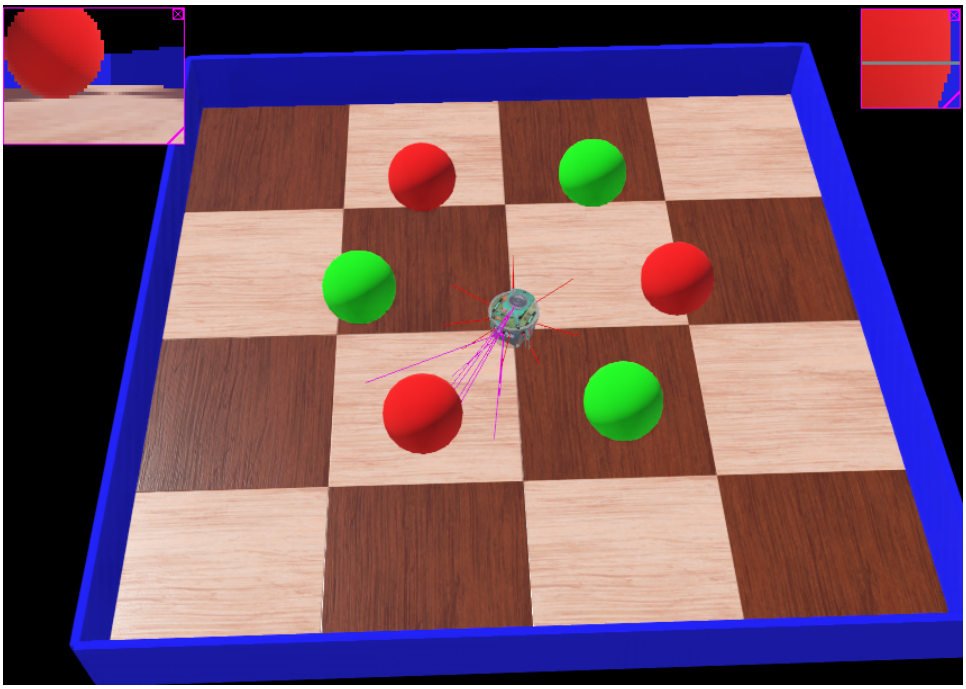

Figure 10: Webots simulation of the e-puck and three red and green spheres.

The controller code consisted of some decision structures, that if the e-puck camera saw the green balls, it would behave normally, turning in place and avoiding any objects it detects. However, if it saw the red balls, it would ultimately crash into them. At the beginning of the simulation, the e-puck would rotate until a red ball was spotted, and then, it would begin to move toward that ball to crash into it. After the ball rolled away and the camera no longer recognized the red color, the e-puck would behave normally, dodging the blue walls and green balls, once the

distance sensors determined it was too close to that object. From this simulation, I translated to using 6 balls of different colors: red, orange, yellow, green, light blue, and purple.

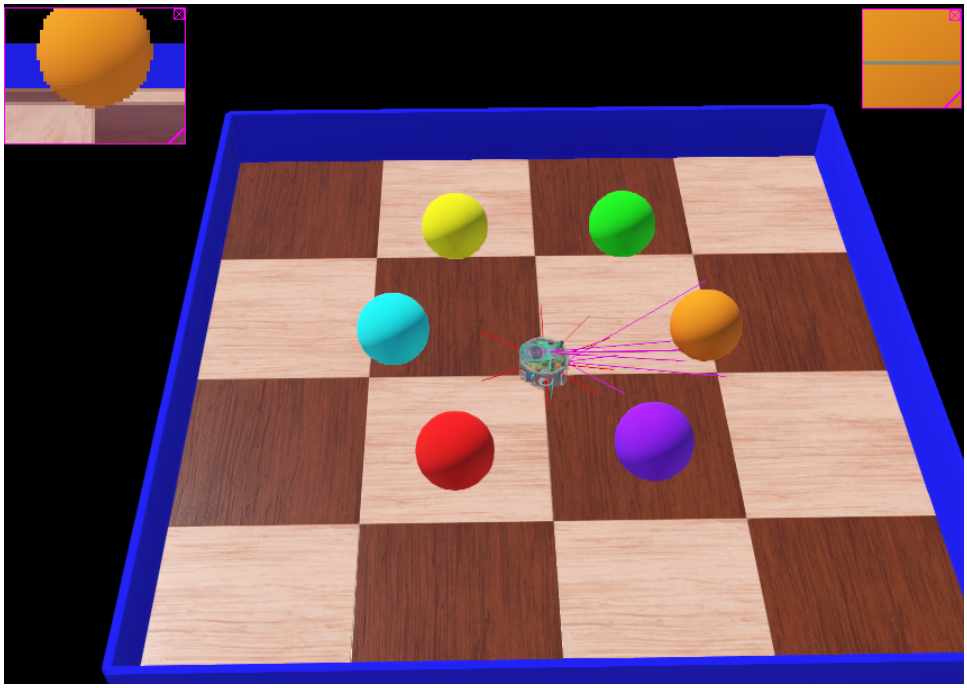

Figure 11: Webots simulation consisting of an e-puck surrounded by six different colored spheres (red, green, purple, yellow, orange, and light blue).

In this case, some of the colors made the robot perform specific tasks, such as, red and green made the robot turn left, while yellow and purple made it move in reverse. It was from this point forward that the focus of my thesis was beginning to change from an adversarial attack on an autonomous vehicle to categorizing objects and privatizing the information received by the recognition camera.

After making great strides with the previous simulation, it was time to create a world where I could demonstrate categorization and privatization of objects as a proof of concept. For this world, I created a maze with dark blue walls, and at specific instances, a portion of the wall would be one of the six colors: red, orange, yellow, green, light blue, and purple.

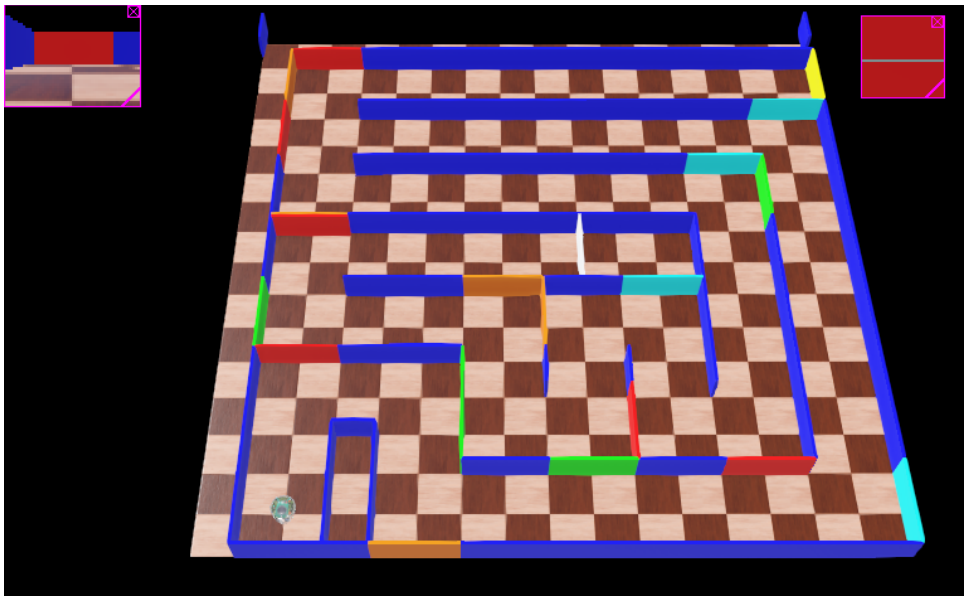

Figure 12: Webots simulation of the maze I created to test the theory of categorization and privatization.

Red and green indicated a right turn, and the robot's speakers would play Pachelbel's "Canon in D," orange and light blue indicated a left turn, and the robot would play Tchaikovsky's "Danse Des Petits Cygnes," and yellow and purple indicated a 180-degree turn, and no music would be played. After completing this sequence of turns, the robot will eventually and ideally finish in the center of the maze where a white wall will indicate that it has reached the end. While designing the maze, I created the path in a way where the robot has a clear route to the end, but the actions and functionality of completing the maze are determined through the man-in-the-middle style of attack. The controller code for the robot consisted of two Python files. The main controller only allowed the robot to perform specific actions such as playing music or not, and turning left, right, or around. The second file was a collection of user-defined functions that determined what color the robot was looking at, categorized that object based on the color, and then, sent that category to the main controller, that determined the specific action for the robot to take.

This implementation of code represents the man-in-the-middle approach of protecting the object seen by the robot camera. From the robot's perspective, it knows it sees an object that has some color, but it doesn't know what that specific color is. This even translates to real world applications, where we have cameras like CCTV everywhere and autonomous vehicles becoming more prominent in our society. This expectation of privacy is impeded when nothing is done to separate the nonessential information from the input image. To a degree, my simulation worked well in functionality for the purpose of my research; however, it was limited in complete success in that I could not figure out how to make the robot turn exactly 90 degrees because of a few discrepancies within the calculation of angular rotation and simulation time.

CHAPTER V: EXPERIMENTATION

To test this idea of privacy within autonomous vehicle cameras, I ran the simulated maze containing an e-puck that uses my proposed solution of a type of man-in-the-middle attack, which consists of two Python files, one that only the robot uses for actions to take (the on-board computer of an autonomous vehicle) and the other for color recognition and category return (proposed MitM/"privacy preserver"). I ran this world multiple times each resulting with the robot in a different position on the maze; however, the privatization of the camera input stream remains fairly consistent. Figure 13 shows the optimal path the robot should take to reach the end (center) of the maze. In Figure 14, the image displays the actual path the robot is supposed to take based on the colors that will be seen by the recognition camera. Figure 15 shows the best run and distance, as well as two other runs and distances that the robot traveled through the maze despite the inaccuracies and inconsistencies present.

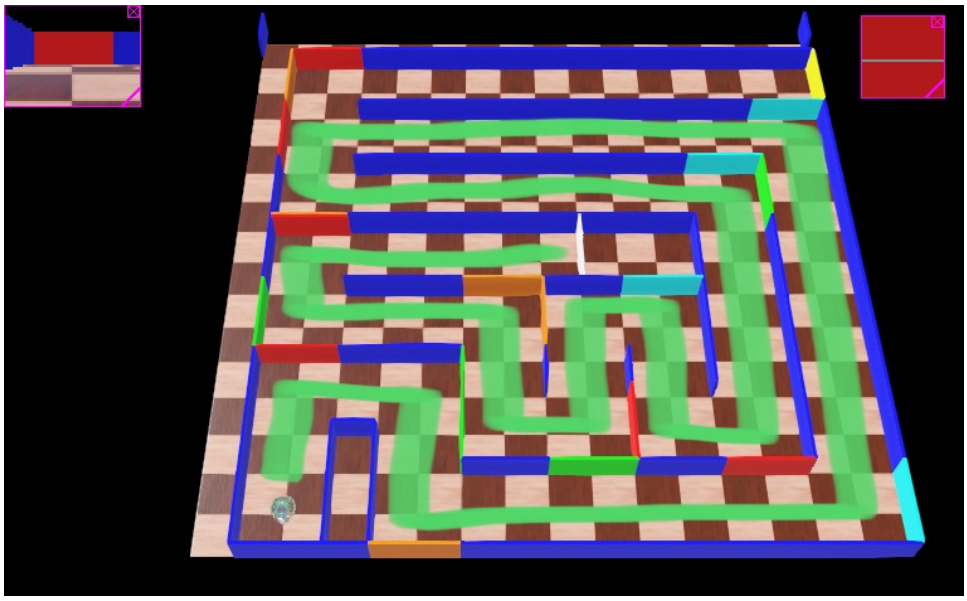

Figure 13: The green line represents the optimal path the robot should follow to reach the end of the maze.

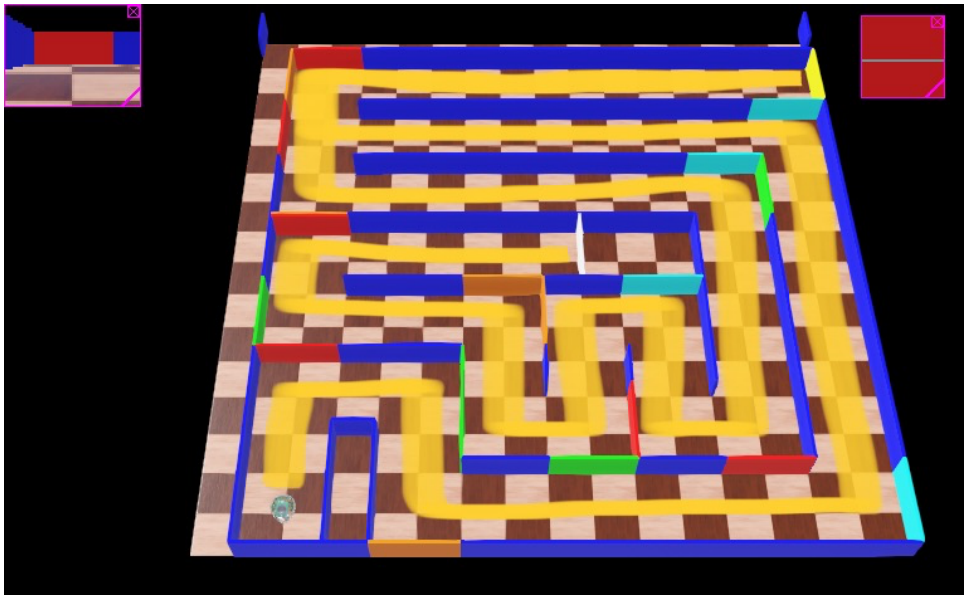

Figure 14: The yellow line represents the actual path, based on the colored portions of the wall, the robot will follow to complete the maze.

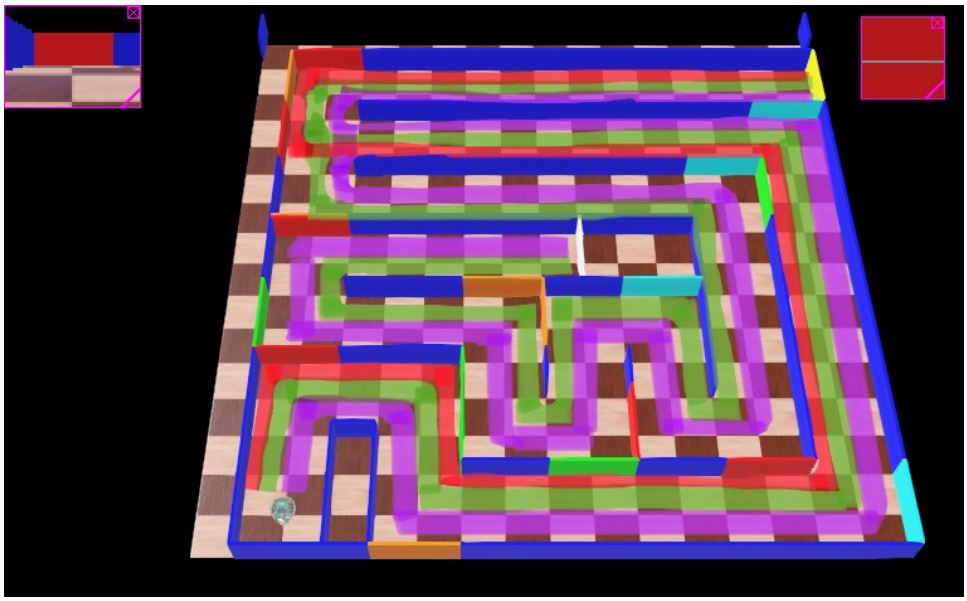

Figure 15: An image of the three best runs of the simulated maze. The red line ran into the robot making a continuous loop on the back corridor of the maze. Although the robot made it to the end of the maze (the green and purple lines), each run executed differently with the robot making loops in different areas of the maze.

These inconsistencies include the lack of a turn 90-degree function for the e-puck to make navigating the maze simpler; therefore, I had to calculate the angular rotation of the robot, but the simulation time led to some possible inaccurate calculations of angular rotation. In addition, there were times when the wall would cast a shadow on the colored portions making the recognition camera unable to determine the category of the object causing the simulation to run inconsistently. To mitigate this issue, I added directional light nodes to minimize the number of casted shadows by the wall, but even this led to parts of the wall that were supposed to be unrecognizable by camera to become recognized causing imprecise results. Although the robot never made it through the complete maze in most of the experiments, the concept of privatization through the man-in-middle attack is still proven viable. This can be seen in the functionality of the code. There is this "gap" between what the robot sees, and the actual action needed to be fulfilled, and to simulate

this, there are two separate Python files, one to control the robot and the other to categorize the colors that are seen by the recognition camera.

CHAPTER VI: CONCLUSION & FUTURE WORK

In conclusion, privacy is an important security factor that should be protected for all, as these technologies discussed in this thesis become more forefront in our society. With the large amounts of data being collected by cameras, we must make sure that we are maintaining privacy within these systems. By utilizing Webots, we were able to simulate the applicability of this high-level concept of privatization and categorization and have proven that this method of using a man-in-the-middle style of attack as a defense mechanism is a viable way in increasing privacy within camera sensors, specifically in autonomous vehicles.

Due to time constraints, we were only able to take this project and research to a certain level; however, with time, this research can be expanded by fine-tuning the robot's actions to make more precise turns or placing the lighting in the simulation in more precise locations. In addition, we could integrate this solution onto a system with a working version of ROS and Gazebo, further explore ways to "perfect" the MitM/ "privacy preserver" code and find other solutions to more security risks found within autonomous vehicle sensors.

LIST OF REFERENCES

[1] Anon. How street view works and where we will collect images next. Retrieved 2023 from https://www.google.com/streetview/how-it-works/
[2] Katherine Scott. 2021. Why Ros? (June 2021). Retrieved 2023 from https://www.ros.org/blog/why-ros/
[3] Andy Greenberg. 2015. Hackers remotely kill a Jeep on the highway-with me in it. (July 2015). Retrieved 2023 from https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/
[4] Anon. 2022. *The Robotic Operating System (ROS) logo*, Tknika.
[5] Anon. 2014. Open Source Robotics Foundation.
[6] Anon. *Webots simulator logo*, Cyberbotics.
[7] Ramakrishnan P, Dhanavel K, Deepak K, and Dhinakaran R. 2023. Autonomous Vehicle Image Classification using Deep Learning | IEEE ... (April 2023). Retrieved 2023 from https://ieeexplore.ieee.org/abstract/document/10104830
[8] Christopher DiPalma, Ningfei Wang, Takami Sato, and Qi Alfred Chen. 2021. Demo: Security of Camera-based Perception for Autonomous Driving under Adversarial Attack. (July 2021). Retrieved 2023 from https://ieeexplore-ieee-org.umiss.idm.oclc.org/document/9474317
[9] Michael Loughlin and Asma Adnane. 2015. Privacy and Trust in Smart Camera Sensor Networks. (October 2015). Retrieved 2023 from https://ieeexplore-ieee-org.umiss.idm.oclc.org/document/7299922
[10] Shagun Bera and Kedar Khandeparkar. 2023. AI Based Real-Time Privacy-Aware Camera Data Processing in Autonomous Vehicles. (June 2023). Retrieved 2023 from https://ieeexplore-ieee-org.umiss.idm.oclc.org/document/10155024
[11] Anon. Introduction to Webots. Retrieved 2023a from https://www.cyberbotics.com/doc/guide/introduction-to-webots
[12] Anon. About eclipse sumo. Retrieved 2023a from https://eclipse.dev/sumo/about/