A logo with green and black text

Description automatically generated

Self-Driving Vehicle

ECE 493 Progress Report #1 – Group 9

Fall 2023

Faculty Supervisor: Professor Sai Manoj

Team: William Davis, Upneet Singh, Daniel Bates, Bryan Benavides, Solanto Boolemu, Gerald Nkansah

# Task Progress

## Physical Car Design and Manufacturing

## Lane Detection Algorithm

## Object Detection Algorithm

# Requirements Specification

The requirements for this project are as follows:

1. The vehicle must stay within the provided lane lines with a settling time of 5% in less than two seconds.
2. The vehicle must reach any desired speed with a settling time of 5% in 5 seconds.
3. The object detection onboard the vehicle must be trained with a mean average precision (mAp) of 30%.
4. The vehicle must correctly identify obstructions on its left, right, and front with 95% accuracy.
5. The vehicle must correctly cross over into a neighboring lane, reaching a settling time of 5% in less than 5 seconds.
6. The vehicle must detect road signs in all lighting conditions with identical mean average precision.
7. The vehicle must send telemetry data and receive user input wirelessly.
8. The vehicle should weigh no more than 5 pounds.
9. The vehicle should have a maximum speed of 15 miles per hour.

# System Design/Architecture

## Physical Architecture

This project consists of a physical component and a heavy software component. The physical component is the structure of the vehicle and the necessary electronics to operate the vehicle. This self-driving car will have a single-motor drive with a front steering system powered by a servo. The primary board that handles all onboard vehicle functions is the Jetson Nano. The Jetson Nano will power up the drive motor, steering servo, and all sensors required for automated driving.

The sensor suite includes the following:

1. Dual Purpose Camera with IR-Cut module
2. IR sensors
3. LiDAR sensor
4. IMU module

The camera module is the main driver for lane detection and object detection. The exact algorithms used for these tasks are discussed in later sections. This camera is a dual-purpose camera with IR-CUT. This allows the camera to filter out IR light in the daytime and allows IR light to pass in low light to produce high-quality images in both day and night conditions. The IR sensors will be used on the sides of the vehicle to measure the distance from obstacles on each side. Additional cameras on the side of the vehicle will help improve accuracy and add a degree or redundancy for lane tracking. In the event the front camera fails, the vehicle will fall back on these cameras to keep itself aware of its location within a lane. While such a failure would reduce the overall system's effectiveness, it allows it to continue functioning even in the event of a sensor failure. The LIDAR (Light detection and ranging) sensor will be used to fall back to the exact positions of nearby obstacles to improve precision and introduce redundancy into the system’s map of the environment. The IMU module will be used to estimate the vehicle’s acceleration, velocity, and position.

The following is a breakdown of the physical components of the vehicle:

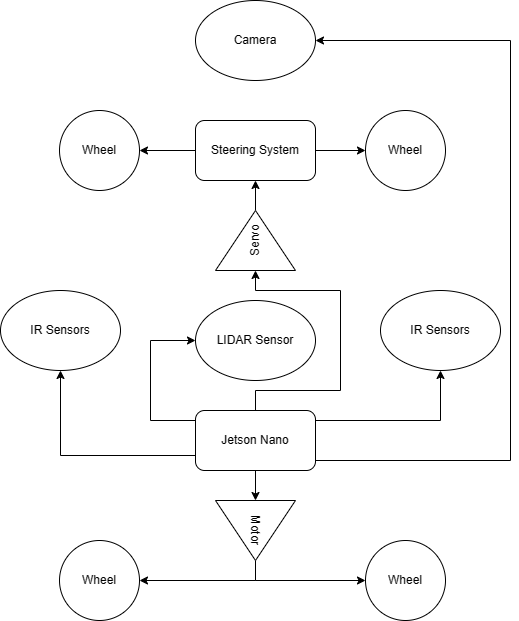


Figure 1: Physical Architecture

## Software Architecture

The vehicle software has three main parts: perception, decision-making, and controller.

A black background with white squares

Description automatically generated

Figure 2: Software Architecture

### Perception

The perception module utilizes the vehicle’s sensor suite to create a map of the environment and localize the vehicle within that map. This map will have the positions and distances relative to the vehicle, allowing the vehicle to determine if it needs to stop or if it is able to perform a lane change.

The perception module has the following functions:

1. Lane Detection
2. Vehicle Detection
3. Speed Sign Detection
4. Stop Sign Detection
5. Other Obstacles Detection

The following is a diagram of how the perception module operates:

A white square on a black background

Description automatically generated

Figure 3: Perception Model Operation

### Decision Module

The decision module examines the information about the surrounding environment from the perception module to decide what the vehicle should do. It also considers user input such as a command to change lanes to the left. This module will consider the user input and examine the environment to decide on what to do. The system examines information based on the urgency and priority of each potential situation:

1. Immediate obstacles that the vehicle may crash into.
2. Stop signs that the vehicle must stop for soon.
3. Vehicle is veering off to the side of the lane and corrective steering is needed.
4. User input to switch lanes. Determine steering and speed setpoints to perform the lane change.

After determining the appropriate next action for the vehicle, the decision module determines the exact setpoints for the steering angle and the speed of the back motor. These setpoints will be fed into the controller.

### Controller

The controller module accepts the speed and steering angle setpoints and determines the exact inputs to apply to the steering servo and drive motor to reach those setpoints. Both the steering input and the drive motor input will be determined using a Linear Quadratic Regulator (LQR) controller.



Figure 4: Controller Breakdown

There will be two setpoints. The steering setpoint is the angle that the front wheels should face to best follow the desired trajectory. The velocity setpoint is the speed that the back motor must reach to follow the trajectory. The outputs will be the exact signals sent to the servo and motors ESC (Electronic Speed Controller) module.

### Schematics and Design

**Schematic:**

A computer screen shot of a circuit board

Description automatically generated

Figure 5: Organization of Connection to a HAT for Jetson Nano

**PCB Schematic:**

A diagram of a circuit

Description automatically generated

Figure 6: Ki-CAD Schematic for PCB HAT

Our project has more components that use I2C and SPI ports than the Jetson Nano offers so we will have to convert some of the GPIO pins to SDA, SCLA, SCLK, MISO, MOSI and CS using software. The PCB design organizes these converted ports for their matching components.

**PCB Design:**

A computer screen shot of a circuit board

Description automatically generated

Figure 7: PCB Design

**PCB 3D Render:**

A computer generated image of a circuit board

Description automatically generated

Figure 8: PCB 3D Render

A green box on a glass wall

Description automatically generated

Figure 9: Side view of PCB 3D Render

**PCB Analysis:**

The system requires more I2C and SPI ports to accommodate all necessary components than are available on the Jetson Nano. Therefore, software will be used to convert some of our GPIO pins into I2C and SPI ports. The PCB will be used to create organized connections for these converted pins.

**Components:**

A close-up of a computer chip

Description automatically generated

Figure 10: Jetson Nano

The Jetson Nano is our chosen microcomputer which has parallel processing which allows us to transmit and process more data from more components. It contains 4GB of RAM. The GPU (Graphical Processing Unit) uses NVIDIA Maxwell architecture. The CPU is a Quad Core-ARM Cortex –A57 64-bit @ 1.43GHz.

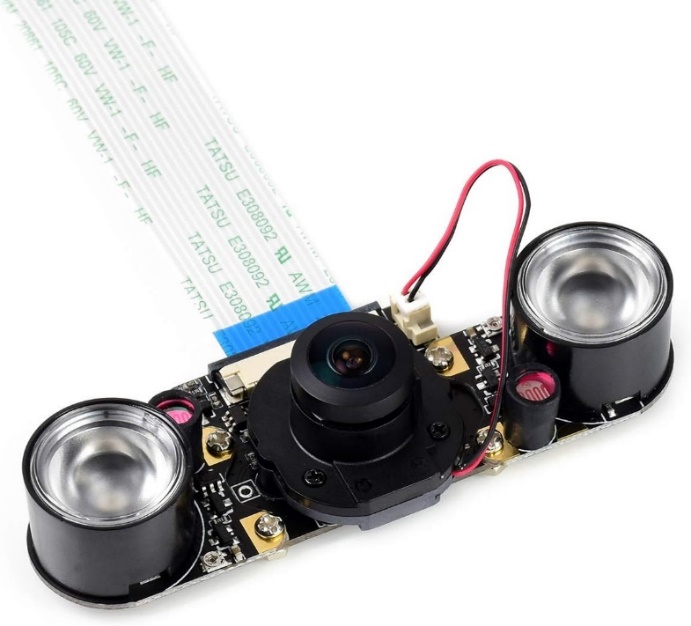


Figure 11: IR Camera

This component is the IMX219-160 8MP IR-cut infrared Night vision camera. This module has a field of view of 162 degrees and a resolution of 3280 x 2464. Will be using 2 IR cameras on the left and right side of the robot car. The purpose of the IR camera is to track lanes. The camera accomplishes this by detecting differences in intensity between light and dark colored surfaces. The IR camera ensures precision and accuracy lane tracking by having 5 IR sensors rather than one.



Figure 12: IR Sensor

This IR Sensor is called the Bolsen GP2Y0A21YK0F IR proximity distance sensor. We will use seven sensors for object detection (three on each side and one in the front). The voltage varies from 3.3 volts to 5 volts. The range of effectiveness varies from 2cm (about 0.79 in) to 80cm (about 31.49 inches).



Figure 13: Ultrasonic Sensor

This ultra sonic Sensor is called the HC-SR04 Ultrasonic Sensor Module. This module will be our back up sensor for object detection. Requires a DC 5 volts power supply. The range of effectiveness varies from 2cm (about 0.79 in) to 450cm (about 117. inches). The sensor uses sound waves to measure the distance to an object.

A black circular object with a white circle on it

Description automatically generated

Figure 14: LiDAR (Light detection and ranging) Sensor

The Slamtec RPLIDAR A1M8 is the LIDAR sensor we chose as a backup source for objection detection. It has 360-degree rotation and scan in a 2d plane. It has a 12-meter range. The LIDAR sensor works by sending a laser into the environment and measures the time it takes for the laser to be reflected to the scanner.



Figure 15: High Speed IR Camera

The B0205 Arducam 1080P component is a day and Night vision USB Camera that will be used for detecting objects, traffic sign identification and lane tracking with the aid of algorithms. It has a 100-degree Field of Vision which is more than adequate for observing the front field of view. It also has the resolution 1920(H) x 1080(V). The motorized (switchable) IR-CUT filter and infrared LEDs can be triggered by the photoresistor according to the light environment to be ON/OFF to block or pass the infrared light.

A close-up of a circuit board

Description automatically generated

Figure 16: Wireless Serial Port Module

The 2Pcs HC-12 433MHZ SI4463/SI4438 wireless serial port Module 1000M component is a radio module and will be implemented for transmitting information between components as well it uses UART Serial for transmitting data. It has an operating frequency Range of 433.4-473.0MHZ and up to 100 communication channels. The maximum voltage is 100mW transient power.

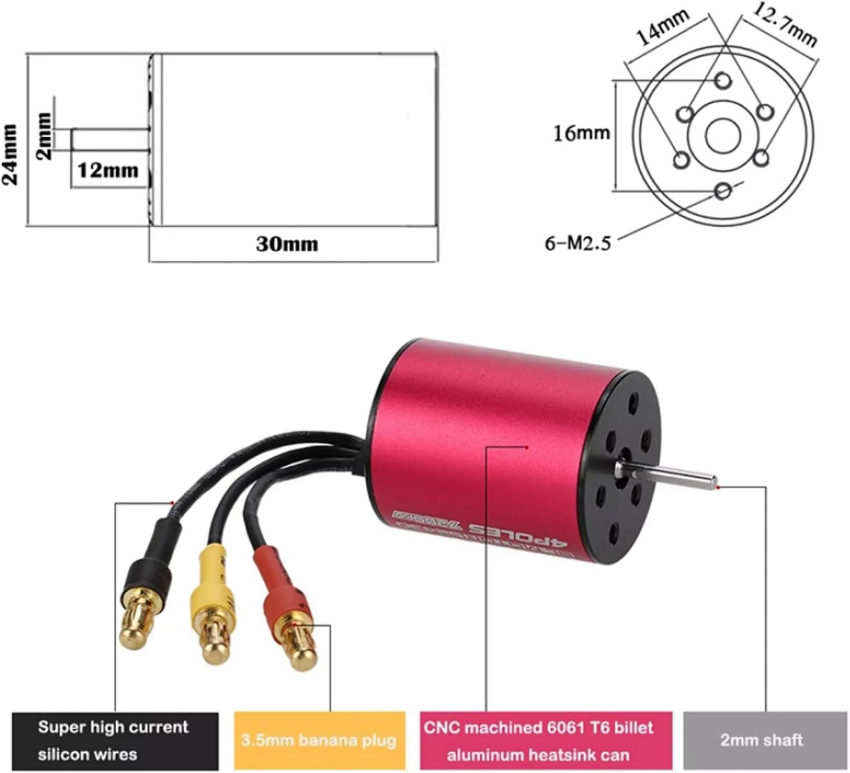


Figure 17: Brushless DC Motor

The component we are using is called a Gool RC S2430 7200KV brushless Motor. It's maximum of 600000 RPM. It requires 5 volts to operate. The Brushless DC Motor will be used exclusively for the vehicle's two back wheels.

A blue circuit board with white text

Description automatically generated

Figure 18: IMU Sensor

This IMU sensor is called a Sun Founder MPU6060 Module. The responsibility of this component is to always determine the car’s orientation, it has a 6-axis gyroscope which gives angular velocity. It also comes with a 3-axis accelerometer that converts mechanical energy to electrical to give a measure for acceleration. This component also has a magnetometer.



Figure 19: Servo Motor

This component is a Gool RC Servo Motor which oversees controlling the two front wheels of the car. The Servo Motor also has the following specifications: 4.8-6.0 Volts for voltage, a plastic gear, rotates 0 to 180 degrees, and has an approximate size of 43 x15 x 30 mm. These motors are typically self-contained electrical devices that rotate parts of a machine (in this case a car) with a good amount of efficiency and great precision which we will need for steering.



Figure 20: LIPO Battery

This component is called the OVONIC 2S LIPO Battery, and its main purpose is to power the jetson nano which will power all the components and motors. One of the best features of this battery is the fact that it is rechargeable so there will not be any need to replace batteries. It also weighs around 98.9 grams. It has a JST-XHR\_3P charging plug. It has a 2200mAh capacity.

# Alternate Designs/Backup

## Distance Detection Sensors

Our current plan for sensors is to use IR sensors on both sides of the vehicle to determine the distance of objects from the vehicle. In the event that the IR sensors prove to be too noisy or inaccurate for our purposes, we plan to use ultrasonic sensors to determine distance. We chose IR sensors for our original plan because of their lower cost along with their reasonable accuracy. However, as we progress into actually building our finalized product, we may choose to switch to the HC-SR04 Ultrasonic Module if the IR sensors do not meet our standards:

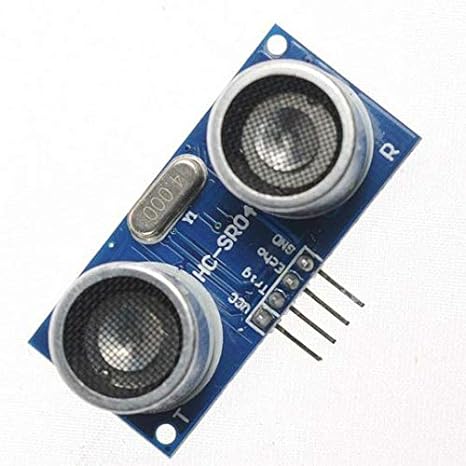


Figure 21: HC-SR04 Ultrasonic Sensor

## Lane Detection Sensors

We have also currently opted to use a single front camera with IR-CUT capabilities for both day and night vision. Extensive preliminary research has shown that this should work excellently and will allow for a single camera to provide all front-facing vision needs. This reduces cost and overall complexity of the system. However, if the single camera fails to perform well, we will switch our design to two camera systems. One camera will be strictly for daytime settings, while the other will be strictly for low-light scenarios. The daytime camera we have selected is the IMX-219 with motorized focus from Arducam:

A camera with a paper strip

Description automatically generated

Figure 22: IMX-219 with motorized focus

The night vision camera we have selected is the IMX-219 IR camera from Arducam:

A close up of a device

Description automatically generated

Figure 23: IMX-219 IR Camera

In addition, the design entails using one small camera with IR-CUT on each side of the vehicle for additional lane detection redundancy. Research has shown that this would be effective. However, in the event that this choice becomes too computationally expensive for the Jetson Nano to handle, we may switch to a Thermal Imaging Camera with much lower resolution. This will significantly reduce the computational power needed to determine the location of lanes on the side of the vehicle.

A blue circuit board with a black circle and white connectors

Description automatically generated

Figure 24: Thermal Imaging Camera

## Control Algorithm

The current plan is to design an LQR controller based on a linearized model of the vehicle. However, if the LQR controller does not perform up to our expectations, we will instead create a PID controller. The benefit of using a PID controller is that it is not based on a linearized model, giving us increased flexibility in how it operates.

A PID controller has three terms: Proportional, Integral, and Derivative.

The Proportional term computes the error between the desired state and the current state. This error is then multiplied by a constant of proportion, which is a parameter that must be tuned.

The Derivative term computes the rate at which the system is approaching the desired state, the derivative of the error. This difference is then multiplied by a constant of derivation, another tunable parameter. This creates some negative feedback to prevent overshoot.

The Integral term computes the sum of all error over time. This accumulated error is then multiplied by a constant of integration. This term ensures that there is no steady-state error and that the system does indeed reach its desired state.

The outputs from each term are added to create the final value that is sent to the motor and servo.

# Background Knowledge

**Infrared Sensors:**

Infrared sensors use infrared light to determine distance from objects. Infrared light is sent from an infrared emitter and then reflected from an object. The time it takes for the light to return to the sensor can be used to determine the distance of an object from the sensor.

**IMU:**

A 6-axis gyroscope calculates the angular velocity of each axis by using the Coriolis effect. A 3-axis accelerometer uses fixed springs, fixed capacitors and a proof mass that will convert mechanical energy into electrical energy. The magnetometer measures and monitors magnetic fields by using polarized light to control the spin of rubidium atoms.

**Dual vison IR Camera with IR-CUT:**

Camera uses the light entering through the lens strikes an image sensor. A camera lens takes all the light rays bouncing around and uses glass to redirect them to a single point, creating a sharp image. The signal output by the image sensor is processed within the camera to create image data, which is stored on the memory card. The photoresistor changes resistance based on the number of lumens received which in turn changes the voltage which can be used to activate the use of IR cut filter which can block or deliver the IR waves depending on external lighting. The focal length and field of view are inversely proportional. FOV=2\*arctan(x/2f), where x is the width of the film.

**Radio Module:**

The radio module transfers data from one object to another as it is a UART (Universal Asynchronous Receiver Transmitter). It has two signals, one for the object receiving and transmitting. It contains a power supply voltage of 3.2 volts to 5.5 volts, and it operates at these frequency ranges: 433.0 to 473.0 Megahertz.

**Servo Motor:**

The servo motor has the following parts: Rotor, Bearings, Shaft, and controller. The Servo Motor also has a PWM signal used for determining the torque, speed, and direction of the car. They typically only have three connections, the PWM as previously mentioned, voltage, and ground. The servo motor moves at specified iteration from 0 degrees to 180 degrees. Servo motors have tachometer sensor which measure the RPM which ensures precision.

**Brushless Motor:**

The brushless motor has three main connections the PWM, Voltage, and Ground. The voltage oversees supplying power to the Motor, the ground is a necessary connection to prevent electrical Finally, the PWM connection is used for controlling the torque, speed, and direction of the car to be mobile. This component will be used for the speed as previously mentioned.

**LiDAR Sensor:**

The lidar sensor works by sending out a light pulse that reflects off objects in the environment and then calculating how long it takes to reflect to the scanner, considering the speed of light. It also measures the angle of reflection. This will generate a point cloud with the photo of all objects around the vehicle. This will be used in conjunction with other sensors to accurately determine the positions of all obstacles around the vehicle.

**Infrared Camera:**

All objects radiate energy at a wavelength. Some of the wavelengths are invisible to the naked eye such as infrared and ultraviolet. Infrared cameras receive energy waves with wavelengths between 780-50 nm. Infrared camera contains an optical system that focuses infrared energy onto a special sensor array that contains thousands of detector pixels arranged in a grid. The pixels express each heat value or wavelength to its corresponding color. The produced image is called a thermography.

**LIPO Battery:**

One of the main characteristics of this battery is that it has an LP503759 cell. LiPos operates based on lithium intercalation and de-intercalation from a positive electrode material and a negative electrode material, with the liquid electrolyte acting as a conducting medium. The S number tells us how many cells are in series with each other. The nominal voltage is the default resting voltage (without charge) of the battery pack that is represented on the battery like 7.4V. The capacity of the battery is a measure of how much power the battery can hold and is measured in milliamp hours(mAh). The bigger the capacity the bigger the battery. The discharge rating is a measure of how fast the battery can discharge safely without harming the battery. It is represented by a c rating. to calculate the maximum continuous load multiple the c rating times the capacity in amps. The charge rate uses the same unit as the discharge rate. To find the wattage of the battery the multiple voltage by amperage.

**Object Detection:**

This vehicle uses YOLOv7 (You Only Look Once Version 7) for its object detection, which is an evolution of the YOLO family of object detection models. These models are known for their high speed and accuracy in detecting objects in images and videos, making them well-suited for real-time applications. The use of YOLOv7 in a small-scale autonomous vehicle can bring several benefits:

* YOLOv7, like its predecessors, is designed for real-time object detection, which is crucial for autonomous vehicles that must make immediate decisions based on their surroundings.
* YOLOv7 is optimized for efficient computation, which means it can run on hardware with limited resources, such as the computer systems available in small-scale autonomous vehicles.
* YOLO models have improved in terms of their accuracy and the ability to minimize false positives. This improvement is critical for autonomous vehicles to reliably detect and classify objects in various environmental conditions.
* YOLOv7 can detect a wide range of objects, which can be essential for autonomous vehicles that need to be aware of other vehicles, pedestrians, traffic signs, and more. However, in our case, we will be using it for traffic signs.
* YOLOv7 can be fine-tuned on a custom dataset. This means it can be trained to recognize objects that are specific to the environment where the autonomous vehicle will operate.
* YOLO models, including YOLOv7, can use multi-scale prediction, allowing them to detect objects at various distances and sizes, which is beneficial for autonomous navigation where obstacles can vary greatly.
* YOLOv7 can be integrated with other sensors and systems in the vehicle, such as LIDAR or Infrared, to improve the robustness and accuracy of object detection under various conditions which will be applied to our use case.

**Control Theory:**

Multiple areas of expertise are required to implement a controller. The first is an understanding of basic Newtonian mechanics which must be applied to the vehicle. This includes basic free-body diagrams and an understanding of forces, torques, and other areas of classical physics. The purpose of this is to develop an accurate state space model that correctly represents the system and the interaction between system states and system inputs. The next area of knowledge which builds upon the state space model is the design of a linear quadratic regulator (LQR) which correctly augments the inputs to the system so that the desired states are observed. This requires an understanding of tuning the cost function associated with the system to develop solutions for optimal values – such as the solution to the Ricatti equation.

# Detailed Design

## Lane Detection Algorithm:

Lane detection for this project uses the following steps:

1. Generate bird’s eye view.
   1. This uses inverse perspective mapping, a type of homography transformation.
2. Grayscale image
3. Identify lane markers.
   1. This uses a Ridge filter.
4. Determine equations of best fit
   1. This uses the RANSAC algorithm.

The following is a breakdown of the lane detection algorithm:

A black and white text

Description automatically generated

Figure 25: System for Lane Tracking

A view from the window of a road

Description automatically generated

Figure 26: Original Image Captured

A blurry image of a person walking on a road

Description automatically generated

Figure 27: Bird's Eye View

A white lines on a black background

Description automatically generated

Figure 28: Bird's Eye View with Ridge Filter

A red and blue line

Description automatically generated

Figure 29: Bird's Eye View with Lane Overlays

## Object Detection Algorithm:

The vehicle will be exposed to environments where there may be multiple objects in the camera module’s field of view at any given time. This implies that the vehicle must be capable of detecting road signs and cars simultaneously. Additionally, the vehicle must be capable of detecting objects in real-time. The current industry standard object detection algorithm which meets both needs is known as ‘You Only Look Once’ version 7 (YOLOv7). This object detection algorithm is computationally cheaper for inference and training and faster for real-time applications compared to its predecessors. However, even this model is too computationally heavy for the limited hardware used in this project. For this reason, a variant will be used, known as YOLOv7-tiny. This model is a downscaled version that is much faster at detecting objects but with less precision. The precision here is sufficient for the requirements of a model vehicle, meaning far greater processing speeds can be achieved with minimal costs. For comparison, both these models were originally trained and tested on the COCO dataset as a benchmark performance. The standard YOLOv7 model has 36.9 million parameters, while the tiny version has only 6.2 million parameters. This results in an inference time almost 600% faster.

The architecture, in both models, is made up of three key components: The Backbone, the Neck, and the Head. The Backbone is what extracts the features of the image, across multiple levels of the image, to extract information for objects large and small. The Neck is a collection of neural network layers that combine and mix image features to pass to the next stage for predictions. This next stage, the Head, takes the features of the network and generates predictions.

A screenshot of a computer

Description automatically generated

Figure 26: YOLO Architecture

A new problem arises. To train a model of this size, hundreds of thousands of images are required to have satisfactory results. Current relevant datasets have nowhere near this number of images, the necessary hardware is unavailable, and it is not feasible to capture and label such a large quantity of images from scratch. For this reason, a method known as transfer learning will be used.

Transfer learning is the process of taking weights that have previously been successfully trained on the images from a different dataset and training those preexisting weights on a new dataset. This method is incredibly attractive for several reasons. The first is that if the dataset being trained on is like the dataset the pre-trained weights were used with, transfer learning can yield incredibly reliable results. This is because the weights have already been established, as opposed to beginning from nothing, and therefore only minor tuning is required. This solves many of the issues mentioned above including meeting minimal hardware requirements, training time, and operation on minimal dataset image quantities.

For the system design, the vehicle must be able to identify several speed signs, and stop signs, and recognize humans. To recognize these objects, two separate datasets will be used: German Traffic Sign Recognition Benchmark (GTSRB), and a ‘Human Dataset.’ GTSRB is a dataset with over 50,000 images which are divided into 43 different classes that represent all German street signs is dataset consisting of over 17,000 images of people. Using several classes from the GTSRB dataset and several thousand images from the Human Dataset, the YOLOv7-tiny model can be trained using transfer learning. An initial test was done using transfer learning on all 43 of the GTSRB classes, whose results can be seen below.

Examples of the signs that our trained model detected:

Inserting image...Inserting image...

Figure 27: Our Model's Tracking of Traffic Signs

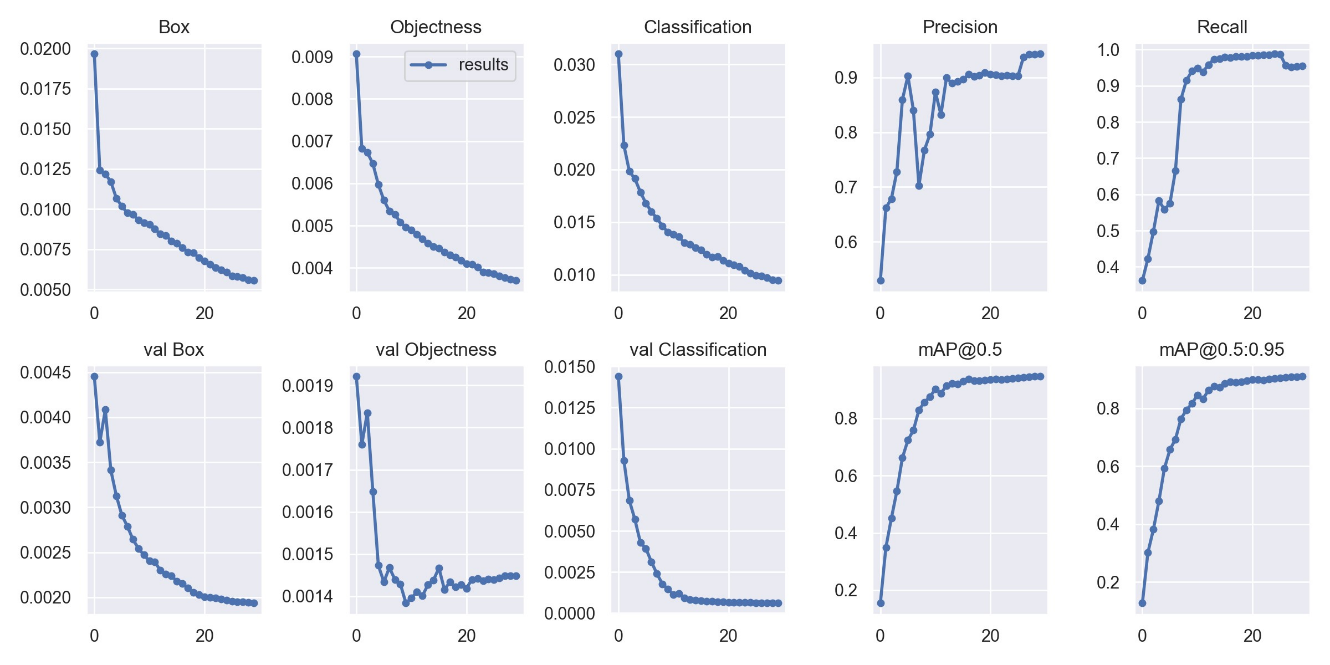


Figure 2830: Calculations produced by Trained Model

* Precision is calculated through dividing the true positives over the true positives and false positives.
* Recall is calculated through dividing the true positives over the true positives and false negatives.

Precision graph displaying the precision of each class that was trained:

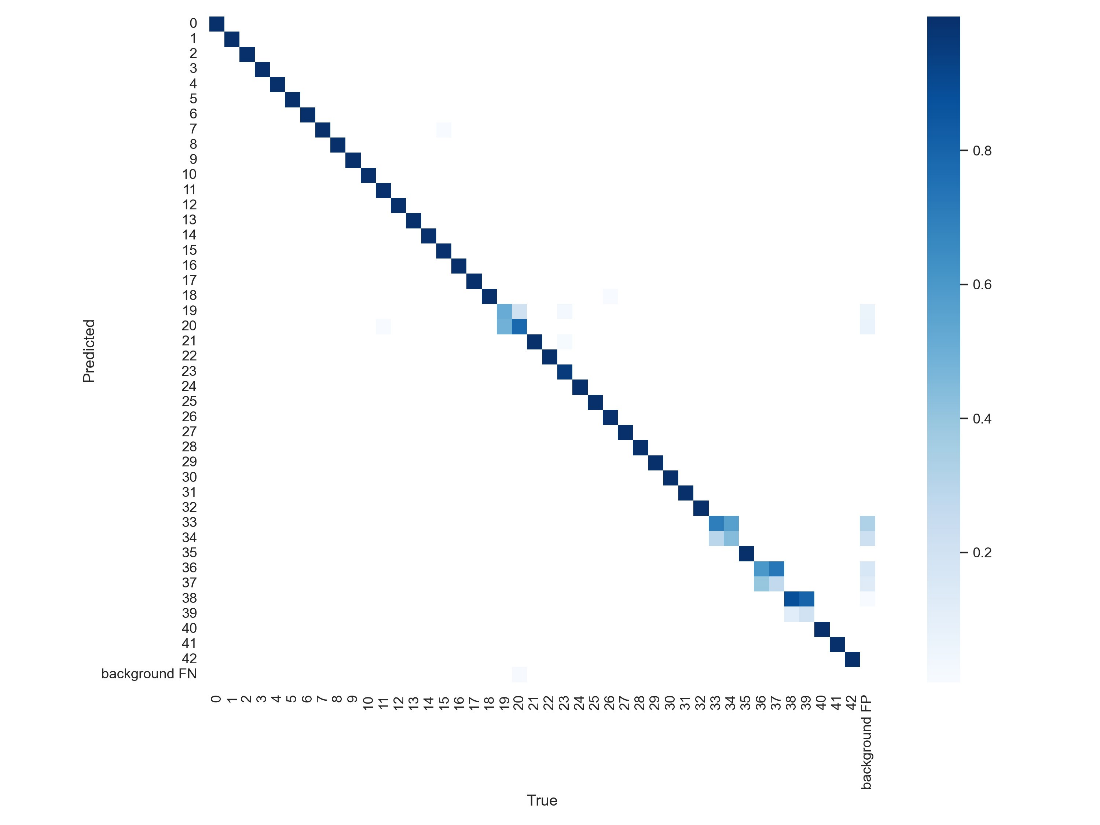


Figure 29: Confusion Matrix Produced by Trained Model

This figure shows the confusion matrix that is produced after the training phase. This is a good representation of how the model behaved when tested on the testing dataset. The more concentrated blue down the diagonal of the matrix, the more accurate the model was (i.e. no false positives or false negatives).

**Object Detection Model:**

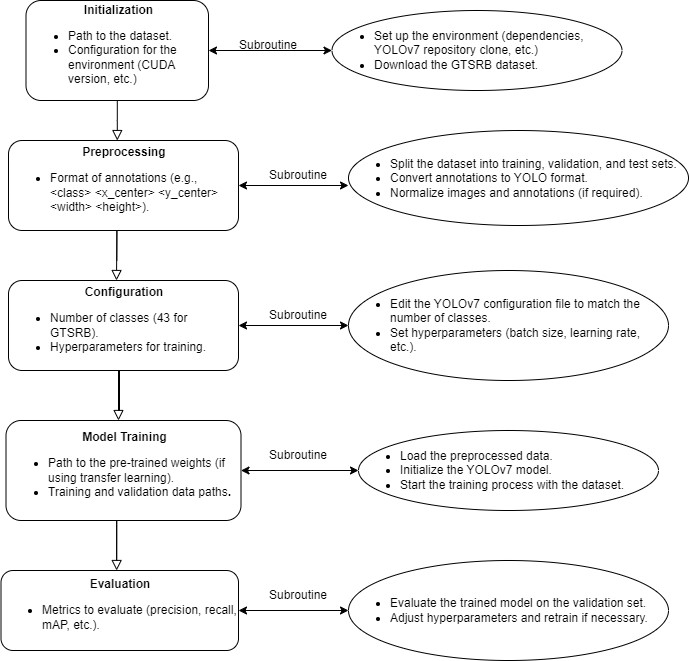


Figure 30: Flow Diagram for Object Detection

## System Controller Design:

The vehicle must control its angular position and rate, velocity, acceleration, and lateral position. This has many state variables to keep track of. Also, these state variables affect each other. For example, steering in the left direction changes the angular rates and positions of the vehicle, which in turn changes the forward velocity. For this reason, a linear quadratic regulator (LQR) will be used. The LQR takes the developed state space model of the vehicle and alters the applied input vector such that the system maximizes stability and reaches the desired response. Specific parameters can be tuned so that the response is altered to the desired transient.

We start by modeling the state space of the vehicle. There are several assumptions to consider before state derivations, which are the following:

1. The system is simplified into a ‘bicycle model’ where the front wheels are treated as a single wheel and the back wheels are also treated as a single wheel.
2. There is no lateral slipping of the tires, especially around turns. Accounting for kinetic friction and the forces and torques involved would increase the mathematical complexity of the model.

A diagram of a mathematical equation

Description automatically generated

Figure 31: Free body diagram

A close-up of a mathematical equation

Description automatically generated

Figure 32: Vehicle State Space Representation

The preceding figures describe the state-space model for perpendicular velocity and for yaw rate and acceleration. The forward velocity can be derived by the following equations.

A math equations on a white background

Description automatically generated

Figure 33: Velocity and acceleration and mass multiplied by acceleration.

A car with text and numbers

Description automatically generated

Figure 34: Free Body Diagram of a Car

An initial linear quadratic controller for single direction velocity is modeled by the Simulink diagram below. The response to a velocity setpoint is then given in the figure below the Simulink diagram.

A diagram of a vehicle system

Description automatically generated

Figure 35: Horizontal LQR design follows MATLAB Simulink Diagram

A graph of a vehicle velocity response

Description automatically generated

Figure 31: Velocity Response for Single Dimensional Velocity Control

A math equations with black text

Description automatically generated with medium confidence

Figure 32: System Transfer Function

A white background with black text

Description automatically generated

Figure 38: System Variables

However, having information to derive a value for each variable in this system transfer function is rare. Therefore, the MATLAB ‘System Identification Toolbox’ will be used. Within this toolbox is the functionality to plot the voltage vs response of the servo and motor. This time domain analysis derives a transfer function for each device which will be used in simulation and physical implementation.

# Prototyping Progress Report

## Simulation Progress

This project involves the use of MATLAB/Simulink to design and test a perception system. To date, we have successfully created a lane detection algorithm using MATLAB’s Automated Driving Toolbox. The lane detection algorithm has been described in earlier sections. The algorithm works well for environments where there are lane markers on both sides. However, the algorithm detects lane markers in places where they do not exist when crossing intersections where lane markers are absent. This is one key issue with the algorithm that needs to be improved with further testing and simulation.

This lane detection system has been tested in multiple different scenarios and has performed well in these scenarios. Because of this experimentation in different environments, we have understood how to tune the parameters of the camera to obtain the best performance.

## Testing Plan

Requirements specified in section 2 must be met, which includes the following:

1. The vehicle must stay within the provided lane lines with a settling time of 5% in less than two seconds.
2. The vehicle must reach any desired speed with a settling time of 5% in 5 seconds.
3. The object detection onboard the vehicle must be trained with a mean average precision (mAp) of 30% and have a high inference percentage of 95%.
4. The vehicle must correctly identify obstructions on its left, right, and front with 95% accuracy.
5. The vehicle must correctly cross over into a neighboring lane, reaching a settling time of 5% in less than 5 seconds.

# Future Plan

The following is our plan for the remainder of the project duration. According to our plan, we will finish the project before the end of the semester. However, this is intentionally the case to account for unforeseen issues. This will provide us with several weeks of cushion time in the likely event that a task does not go as planned and takes longer than would have been expected.

A screenshot of a computer

Description automatically generated

Here is a detailed list of all remaining tasks:

|  |  |
| --- | --- |
| Create Chassis/Skeleton | Develop the entire frame for the vehicle. Attach all motors, sensors, axles, wheels, differentials, etc. and Jetson Nano on the vehicle. Test thoroughly and ensure the drive works well. |
| Develop Code for LiDAR | Generate a 2D map of the surroundings of the LiDAR sensor. |
| Create Code for User Interface | Develop code that shows what the vehicle is seeing and its perception of the environment. Also, should allow the user to specify what the vehicle should do - change lanes, stop driving, etc. |
| Finalize algorithm for lane boundary detection | Write final code to determine the location of the lane boundaries from the vehicle's current location using both the front and side cameras |
| Combine LiDAR and IR Sensor Data | Create a simplified map of the environment using data from both LiDAR and IR sensors. This should specify the locations of objects in the front and both sides of the vehicle with relatively high accuracy. |
| Develop Night-time Road Sign Detection | Develop code for sign detection in low light conditions. Determine the processing that is necessary to make low-light images look good, and test low-light sign detection |
| Test Sign Detection of Front Camera (Day and Night) | Fully test all sign detection and make sure that the vehicle can accurately detect signs and their positions in both day and night. |
| Test Lane Detection (Day and Night) | Thoroughly test determining lane boundaries in both day and night conditions. Ensure that the camera can track lanes very effectively. |
| Test Collision Detection (Day and Night) | Test how well the vehicle automatically stops for obstacles in the front of the vehicle. |
| Develop Code for a Voter/Decision Module | Write the code that determines what the vehicle should do based on its surroundings and user input. It should do the following: check if there is an object in front of the vehicle check if there is a sign to indicate a stop or a speed change take in user input check if user wants vehicle to stop check if object to left or right. Then if user requests lane change, grant if no objects, deny if there ARE objects not changes, then stay centered in lane output is a setpoint (or series of states) |
| Create Controller (test linear and nonlinear control) | Create all code that determines the inputs for steering servo and drive motor. Test the controller to ensure the vehicle can stay within its lane |
| MATLAB Simulations | Test everything in MATLAB |