Tesla's Autopilot: A Critical Look on its Formation, Components, Challenges and Future Directions.

By

Mukhesh Ravi

Final Project for Artificial Intelligence & Business Analytics

Submitted to the DeVos Graduate School,
Northwood University,
In Fulfillment of the Requirements,
For A Master's Degree
13th July 2024

Table of Contents

Tesla's Autopilot: A Critical Look on its Formation, Components, Challenges and Future Directions.

Mukhesh Ravi

Master of Science in Business Analytics
DeVos Graduate School, Northwood University

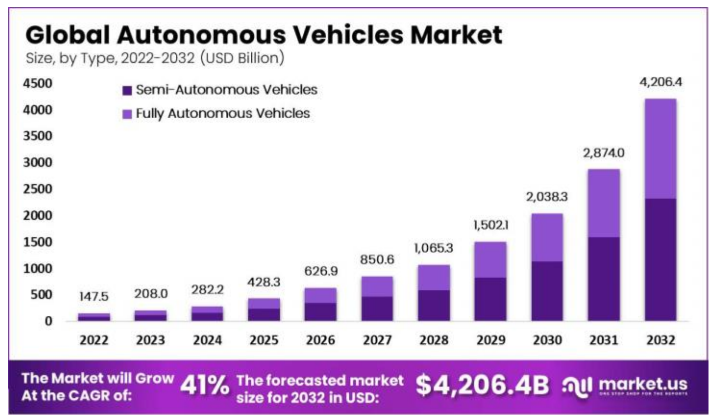
***Abstract* -** This paper delves into the development of Tesla's Autopilot system, a cornerstone of the company's commitment to autonomous driving technology. We explore the concept of Autopilot, tracing its formation from initial concept to real-world implementation. The paper then dissects the core components of Autopilot, including the sensor suite and the underlying Artificial Intelligence (AI) that drives decision-making. We critically evaluate the challenges inherent to Autopilot's technology, focusing on the limitations of camera-based vision systems, ensuring safety in complex environments, and navigating the regulatory landscape. Finally, the paper explores the potential future directions for Tesla, considering advancements in AI that could pave the way for more robust autonomous driving capabilities. In conclusion, by critically examining Tesla's Autopilot, we aim to inform the ongoing discussion about the viability and future of autonomous vehicles, a technology with the potential to revolutionize transportation and fundamentally alter our relationship with the road.

***Keywords* -** Tesla Motors, Autopilot, Electric Vehicles, Radar, Ultrasonic sensors, Autosteer, Autolane change, Autopark, Artificial Intelligence, Machine Learning

1. INTRODUCTION

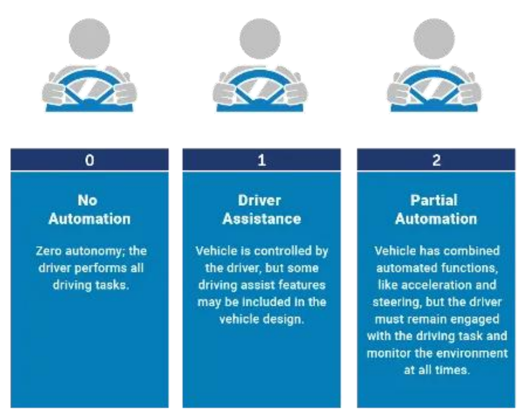
A self-driving car, also known as an autonomous car (AC), driverless car, robotaxi, robotic car, or robo-car, is a car that can operate with reduced or no human input (Rambus, 2022). Self-driving cars are responsible for all driving activities, such as perceiving the environment, monitoring important systems, and controlling the vehicle, which includes navigating from origin to destination. As per the chart below, the global autonomous vehicles market size was worth around USD 147.5 billion in 2022 and is predicted to grow to around USD 4,206.4 billion by 2032 with a compound annual growth rate (CAGR) of roughly 41% between 2022 and 2032. Today, over 40 autonomous car companies are vying for a space in the industry and manufacturers battle it out to claim as much market share as possible.

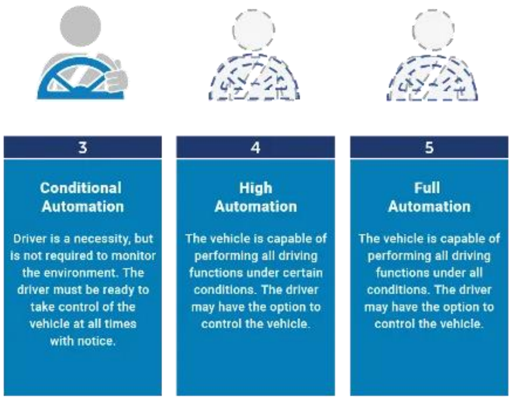
Image 1: *Global Autonomous Vehicles Market Targeted Sales*



The Society of Automotive Engineers (SAE) categorizes the level of vehicle autonomy in six levels. As the levels progress, the need for human intervention gradually decreases until the final level, in which full automation is achieved.

Image 2: *The Six Levels of Vehicle Autonomy*





1.1 - Chosen Organization - Tesla

Tesla Motors is a well-known automobile company in the automotive industry, particularly known for its innovations in electric vehicles (EVs) and autonomous driving technology. Founded in 2003 by a group of engineers, Tesla's mission is to accelerate the world's transition to sustainable energy through its product offerings. Under the leadership of Elon Musk, Tesla has revolutionized the perception and capabilities of electric cars with its lineup, including the Model S, Model 3, Model X, Model Y, and Cybertruck. Tesla's vehicles are renowned for their convenience, long-range batteries, sleek designs, smart features, and powerful performance.

Mission Statement: Accelerating the world's transition to sustainable energy.

Vision Statement: To create the most compelling car company of the 21st century by driving the world's transition to electric vehicles.

1.2 – Problem Statement

Tesla aims to create more sustainable vehicles by developing electric cars that reduce carbon emissions and autonomous driving technologies that enhance safety and efficiency. Their goal is to lead the transition towards a greener and smarter transportation future.

1.3 – Alternative Solution

Before adopting AI as the primary technology for autonomous vehicles, Tesla explored several alternative solutions:

a) *Enhanced Advanced Driver Assistance Systems (ADAS)*

Tesla initially considered relying on pre-programmed rules for its driver assistance systems (Ojango, 2023). This is similar to the first-generation GPS systems where data is fed into a device and sold. It will not have the capability to update itself. This approach offered some benefits like lane-keeping and automatic braking, but it had limitations. The pre-set rules couldn't adapt to ever-changing situations like construction zones, road closures, or unusual traffic patterns. In addition, programming for every possible scenario globally would be incredibly challenging, complex and time-consuming. This lack of adaptability and scalability ultimately steered Tesla away from this approach and towards exploring solutions with more learning capabilities.

b) *Connected Vehicle Technologies*

Tesla also tried implementing Vehicle-to-Everything (V2X) communication, where cars talk to each other (V2V) and to traffic lights and signs (V2I) (U.S. Department of Transportation, 2020). This approach allows for smoother traffic flow and collision avoidance. However, V2X relies on almost all vehicles and infrastructure being equipped with this technology around it. Since most cars currently lack V2X capabilities and upgrading roads worldwide would be expensive, this solution wasn't practical in the short term. Besides that, V2X just provides information exchange, and Tesla might have desired more control over the car's behavior for its autonomous driving goals.

c) *Infrastructure Improvements*

Upgrading roads with smart technology like embedded sensors and dedicated lanes for autonomous vehicles seemed like a good fit for Tesla's self-driving ambitions. This "smart infrastructure" could improve traffic flow and safety. However, the idea faced major backlash as upgrading the entire infrastructure would be incredibly expensive and require cooperation with government agencies, which would be a lengthy process. These issues pushed Tesla to explore other solutions.

1.4 – Tesla's Autopilot System driven by AI (chosen solution)

Tesla claims that Autopilot gives you more confidence behind the wheel, increases your safety on the road, and makes highway driving more enjoyable (Tesla, 2015). While truly driverless cars are still a few years away, Tesla Autopilot functions a lot like the systems that airplane pilots use when conditions are clear. The driver is still responsible for, and ultimately in control of, the car. What's more, Tesla gives the driver intuitive access to the information that it is using to control its actions. Along with the usual combination of accident prevention technology such as Advanced Driver Assistance Systems(ADAS), which actuates emergency steering and braking, the autopilot technology that powers their electric vehicles enables cars to autonomously steer, change lanes, follow vehicles and curves, park automatically and many more.

1.5 – Why this decision?

Back in 2010 when the idea of machine learning and Artificial Intelligence were just being explored on a small scale, Tesla was still reluctant to approve this idea. Despite acknowledging potential challenges with AI, they see it as a more promising path for autonomous vehicles compared to the alternatives explored. AI offers significant benefits. First, AI can learn and adapt to new situations on the road, unlike pre-programmed rules that struggle with the unexpected. Second, AI algorithms can be constantly improved and deployed to a large fleet of vehicles, making them more scalable for diverse driving conditions worldwide. Lastly, Tesla might have more control over a car's behavior using AI compared to relying on external factors like communication with other vehicles (V2X) or waiting for widespread infrastructure upgrades. These potential benefits of adaptability, scalability, and control make AI an attractive option for Tesla's vision of self-driving cars.

2. TESLA'S AUTOPILOT FEATURES

2.1 – Traffic-Aware Cruise Control

Tesla's Traffic-Aware Cruise Control (TACC) uses a combination of machine learning and sensor data to automatically adjust your car's speed and maintain a safe following distance from the vehicle ahead.

Image 3: *Traffic-Aware Cruise Control activated in a moving Tesla*

2.1.1 - Machine Learning Techniques

a) Computer Vision

Tesla's TACC system relies heavily on computer vision to understand the road environment. Cameras act as the eyes of the system, capturing real-time video. Those videos are then transferred to the powerful algorithms called Convolutional Neural Networks (CNNs), where it takes center stage. These CNNs are like expert image analysts, having been trained on massive datasets to identify specific objects on the road. They sift through the camera footage, effectively recognizing and pinpointing lane lines and surrounding vehicles. This ability to "see" the road is crucial for TACC to function properly.

b) Object Detection

Tesla's TACC system also relies on object detection with Convolutional Neural Networks (CNNs). Cameras capture real-time video, which is then analyzed by CNNs. These CNNs, trained on massive datasets of vehicles, can effectively distinguish and classify cars, trucks, and motorcycles based on their visual features. This allows the system to understand the type and presence of surrounding traffic.

c) Sensor Fusion

Besides the camera, Tesla also utilizes sensor fusion. This can be defined as combining data from multiple sensors for a more comprehensive picture. Radar sensors emit radio waves to precisely estimate the distance of the vehicle ahead, even in low-light conditions. Tesla's are also equipped with ultrasonic sensors that provide short-range detection for objects very close to the car, crucial for stop-and-go traffic or parking situations. By combining camera data with the precision of radar and short-range awareness of ultrasonic sensors, TACC gains a richer understanding of the surrounding environment, enabling it to accurately estimate vehicle distance and ensure safe control of your car's speed.

2.2 – Autosteer

Image 4: *Autosteer mode activated in a moving Tesla*

2.2.1 – Machine Learning Techniques

a) Lane Detection

Lane detection is a critical component of Tesla's Autopilot, leveraging computer vision to identify lane markings on the road. The system uses a combination of cameras and deep learning algorithms to process the visual data captured in real-time. Convolutional Neural Networks (CNNs), a type of deep learning model particularly effective in image recognition tasks are trained to recognize and distinguish lane markings from other road features. These networks are fed with large datasets containing diverse road scenarios to improve accuracy and robustness. By continuously analyzing the visual input, the lane detection system can accurately determine the position and orientation of lane boundaries, even in challenging conditions such as poor lighting, worn-out markings, or adverse weather

b) Path Planning

Once the lane markings are detected, path planning algorithms take over to calculate the optimal path for the vehicle to follow within the identified lane. This involves creating a safe and efficient trajectory that takes into account the current speed, road curvature, and positions of other vehicles. The path planning system uses a combination of rule-based logic and machine learning models. Rule-based logic provides the fundamental guidelines for driving, such as maintaining a safe distance from other vehicles and adhering to traffic rules. Machine learning models, particularly reinforcement learning, are used to optimize these rules based on real-world driving data. This ensures that the vehicle can adapt to dynamic driving environments, such as merging traffic or sudden lane changes by other drivers, and choose the best possible path to maintain smooth and safe driving.

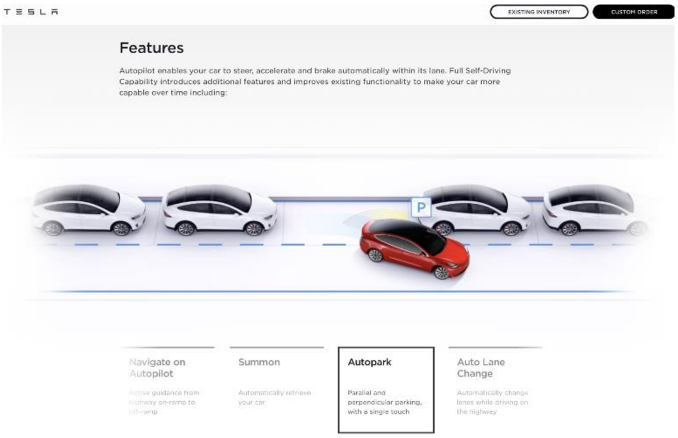
c) Control Systems

Control systems are responsible for executing the planned path by making continuous adjustments to the vehicle's steering, acceleration, and braking. Machine learning models play a vital role in these control systems by processing real-time sensor data and ensuring that the vehicle remains centered in the lane. Proportional-Integral-Derivative (PID) controllers, often enhanced with machine learning techniques, are used to maintain precise control over the vehicle's movements. These models are trained to handle various driving scenarios, such as straight roads, curves, and lane merges, ensuring a stable and responsive driving experience. By continuously adjusting the steering and speed, the control systems can react to any deviations from the planned path, compensating for factors like road surface conditions, wind, and vehicle dynamics. This results in a smooth and safe driving experience that mimics the attentiveness and adaptability of a human driver.

2.3 – Autopark

Tesla's Autopark feature provides automated parallel and perpendicular parking assistance. It uses ultrasonic sensors and cameras to detect suitable parking spaces, then employs machine learning algorithms to generate and follow an optimal path to park the car precisely. The system continuously monitors for obstacles and adjusts the maneuver to avoid collisions, ensuring a smooth and safe parking experience.

Image 5: *Autopark feature of a Tesla*



2.3.1– Machine Learning Techniques

a) Space Detection

Space detection is the first step in Tesla's Autopark feature, where the system identifies suitable parking spaces for parallel or perpendicular parking. This process involves using a combination of ultrasonic sensors and cameras to scan the surroundings. Ultrasonic sensors provide precise measurements of the distance to nearby objects, while cameras capture visual data of the parking area. Machine learning algorithms process this sensor data to detect and evaluate potential parking spaces. These algorithms are trained on a vast dataset of various parking scenarios, allowing them to accurately recognize spaces that are appropriate for the vehicle's size and parking type. The system can distinguish between occupied and unoccupied spaces and assess the dimensions and orientation of available spots to ensure the vehicle can fit safely.

b) Path Planning and Control

Once a suitable parking space is detected, the path planning and control system takes over to guide the vehicle into the spot. Machine learning algorithms generate an optimal path that considers the vehicle's current position, the dimensions of the parking space, and the surrounding environment. This involves calculating the precise steering angles and speed adjustments required to maneuver the car into the space smoothly and accurately. The algorithms use real-time data from the vehicle's sensors to continuously update the planned path and control actions, ensuring that the vehicle follows the intended trajectory. The system's ability to adapt to dynamic conditions, such as slight misalignments or variations in the parking space, ensures a reliable and precise parking performance.

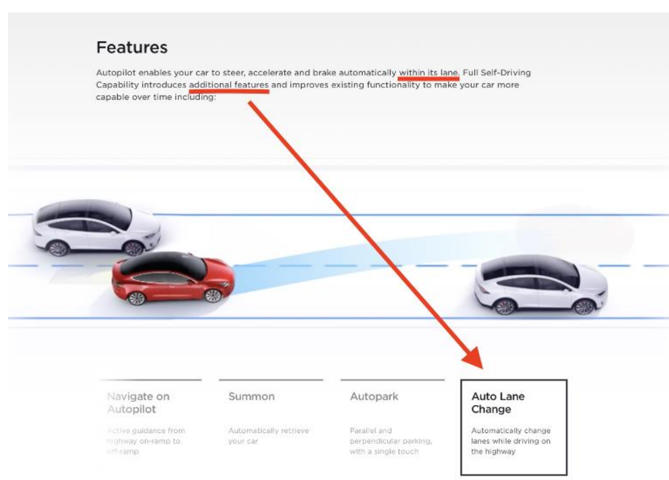
c) Object Avoidance

Object avoidance is a critical component of the Autopark feature, ensuring that the vehicle does not collide with any obstacles during the parking maneuver. The system uses the data from ultrasonic sensors and cameras to detect obstacles in the vehicle's path, such as other parked cars, pedestrians, or curb edges. Machine learning models analyze this data to predict the potential risks and adjust the vehicle's movements accordingly. If an obstacle is detected, the system can modify the path, slow down, or stop the vehicle to prevent collisions. These adjustments are made in real-time, providing a responsive and safe parking experience. The continuous feedback loop between the sensors, machine learning models, and control systems enables the vehicle to navigate complex parking environments with high precision and safety.

2.4 – Auto Lane Change

Tesla's Auto Lane Change feature allows the vehicle to change lanes automatically when the driver activates the turn signal. It uses sensors and computer vision to detect gaps and the positions of other vehicles, performs a risk assessment using machine learning models to ensure the safety of the maneuver, and then smoothly executes the lane change by adjusting the vehicle's speed and steering. This feature enhances driving convenience and safety by automating the lane change process.

Image 6: *Auto Lane Change of a Feature of Tesla*



2.4.1 – Machine Learning Techniques

a) Surrounding Awareness

Surrounding awareness is the initial step in Tesla's Auto Lane Change feature, where the system gathers information about the vehicle's environment to detect gaps and the positions of other vehicles. This involves using an array of sensors, including cameras, radar, and ultrasonic sensors, to capture comprehensive data around the car. Computer vision algorithms process the visual data from the cameras to identify the lanes and surrounding vehicles, while radar and ultrasonic sensors provide precise measurements of distances and relative speeds. The integration of these sensor inputs allows the system to create a detailed and real-time map of the vehicle's surroundings, identifying suitable gaps in adjacent lanes for a potential lane change.

b) Risk Assessment

Before initiating a lane change, the system conducts a thorough risk assessment to evaluate the safety of the maneuver. Machine learning models analyze the data from the surrounding awareness step to determine the feasibility of the lane change. These models are trained on extensive driving datasets, enabling them to understand various traffic scenarios and predict the behavior of nearby vehicles. The risk assessment involves checking for sufficient space in the target lane, the speed and trajectory of other vehicles, and any potential hazards. The system ensures that the lane change can be completed safely without causing disruptions or collisions, prioritizing the safety of the vehicle and its occupants.

c) Execution

Once the surrounding awareness and risk assessment steps confirm that it is safe to change lanes, the execution phase begins. This involves controlling the vehicle's speed and steering to perform the lane change smoothly and efficiently. Algorithms calculate the precise adjustments needed to transition from the current lane to the target lane, taking into account the vehicle's current speed, the distance to the target lane, and the positions of surrounding vehicles. The control systems use this information to manage the steering angle and acceleration or deceleration required for a smooth lane change. Throughout the maneuver, the system continuously monitors the environment and makes real-time adjustments to ensure that the lane change is completed safely and seamlessly.

2.5 – Summon

Tesla's Summon feature allows the vehicle to move in and out of tight spaces remotely, providing added convenience and safety in crowded parking areas. This feature can be controlled via a mobile app, enabling the driver to move the car without being inside it. Summon is particularly useful for parking in tight spaces, garages, or retrieving the car from a spot where it is difficult to open the doors.

I***mage 7:*** *Summon Feature of Tesla*

2.5.1 – Machine Learning Techniques

a) Obstacle Detection

The obstacle detection system uses an array of sensors, including ultrasonic sensors and cameras, to detect and navigate around obstacles. These sensors provide real-time data on the vehicle's surroundings, which are processed by machine learning models. Neural networks, specifically trained for 3D object detection and localization, identify and classify surrounding objects in three-dimensional space. This ensures that the vehicle can safely maneuver around obstacles, avoiding potential collisions during the summoning process.

b) Path Planning

Path planning is a critical component of the Summon feature, where machine learning models are used to plan a safe and efficient path to the desired location. Algorithms consider the vehicle's current position, the detected obstacles, and the layout of the parking area to generate an optimal path. These models continuously update the path based on real-time sensor data, ensuring safe navigation out of the parking spot. The path planning system ensures that the vehicle can handle complex parking environments and tight spaces with precision.

c) Remote Control

The Summon feature allows the car's movements to be controlled remotely via a mobile app, leveraging the onboard machine learning systems. The app communicates with the vehicle, sending commands to initiate the summoning process. The machine learning systems onboard the vehicle then execute these commands, using the obstacle detection and path planning models to safely navigate the vehicle. This remote control functionality adds a layer of convenience and safety, enabling the driver to summon the vehicle from a distance without needing to be physically present.

3. CHALLENGES

3.1 – Technical Challenges

3.1.1 - Sensor Integration and Accuracy

In 2016, a Tesla Model S was involved in a fatal accident when Autopilot failed to recognize a white truck crossing the road against a bright sky background. The car's radar and camera sensors did not accurately detect the truck, highlighting the difficulties in sensor fusion and accuracy under challenging conditions (Goodall, 2014).

3.2 – Regulatory Challenges

3.2.1 - Compliance with Laws and Regulations

Tesla faced regulatory scrutiny in Germany, where the Federal Motor Transport Authority (KBA) ordered Tesla to stop using the term "Autopilot" in advertising, arguing it could be misleading. This highlights the complexity of complying with varying international regulations (Van Eperen, 2016).

3.3 – Safety Challenges

3.3.1 - Ensuring Robust Safety

The 2016 fatal crash in Florida, where a Tesla on Autopilot failed to detect a crossing truck, underscores the importance of ensuring robust safety. The incident prompted investigations by the National Highway Traffic Safety Administration (NHTSA) and led to calls for improved safety measures and more rigorous testing (Kalra & Paddock, 2016).

3.4 – Public Perception, Trust and Liability Issues

3.4.1 - Building Public Trust

High-profile accidents involving Tesla's Autopilot, such as the 2018 fatal crash in Mountain View, California, have damaged public trust in the technology. These incidents have led to skepticism about the safety and reliability of autonomous driving, posing a significant challenge for Tesla in gaining public trust (Bansal et al., 2016).

4. IMPLEMENTATION OF IMAGE CLASSIFICATIONS THROUGH JUPYTER

For this research paper, I have created a basic python code to recognize and classify traffic signs using deep learning and Convolution Neural Networks. The complete python code is attached in the Appendix.

4.1 – Stop Sign

Image 8: *A Stop sign is being detected by the camera*

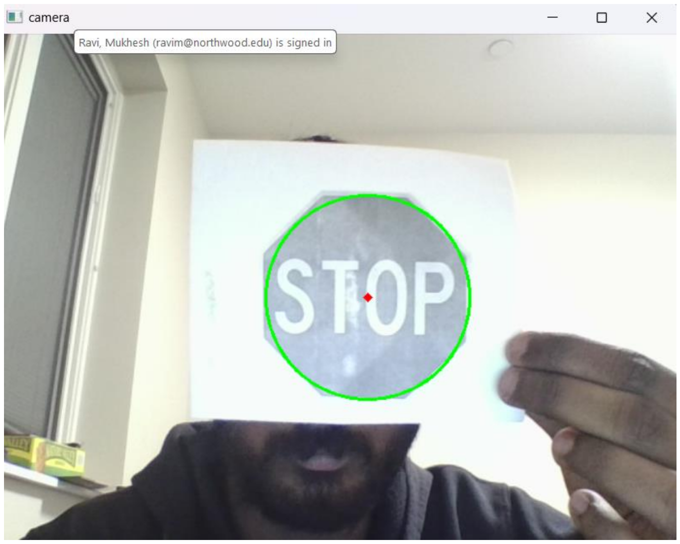


Image 9: *The algorithm detects the image and responds to it*

```
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
STOP
```

In this experiment, I showed the camera a STOP sign and it recognized and classified it.

4.2 – Go Forward and Turn Right Sign

In this experiment, I showed the camera a STOP sign and it recognized and classified it.

Image 10*: A move forward and turn Right sign being detected by the camera*

Image 11*: The algorithm detects the image and responds to it*

```
FORWARD AND RIGHT
FORWARD AND RIGHT
FORWARD AND RIGHT
FORWARD AND RIGHT
FORWARD AND RIGHT
FORWARD AND RIGHT
FORWARD AND RIGHT
FORWARD AND RIGHT
FORWARD AND RIGHT
FORWARD AND RIGHT
FORWARD AND RIGHT
FORWARD AND RIGHT
FORWARD AND RIGHT
FORWARD AND RIGHT
FORWARD AND RIGHT
FORWARD AND RIGHT
FORWARD AND RIGHT
```

5. Literature Review

I have reviewed multiple papers that have spoken about Tesla's Autopilot and the mechanisms behind it. For instance, I have screened through the paper titled "Tesla Autopilot : Semi-Autonomous Driving, an Uptick for Future Autonomy" by Shantanu Ingle and Madhuri Phute. They wrote this paper in 2016 when Tesla slowly started to penetrate the automobile industry. Many elements are similar to the structure of my paper. However, I have emphasized the machine learning techniques used by Tesla's Autopilot that are not explained in their paper. I have particularly explained the process of each machine learning and how they contribute towards the feature. Besides that, I have also developed an image classification system where I can show road signs to my webcam, and the system recognizes and responds to it. Lastly, I have also emphasized the challenges Tesla faced after implementing the Autopilot in their vehicles.

6. Conclusion

Upon completing this research paper, I feel Tesla's direction towards developing a fully autonomous vehicle is achievable, despite many challenges. There will be many technical, regulatory, and safety challenges that Tesla needs to overcome. However, the future of Tesla's

Autopilot is promising. With continued advancement in AI and machine learning in the transportation industry, the goal of a fully autonomous vehicle that is safe can be achieved.

7. References

Anderson, J., Kalra, N., Stanley, K. D., Sorensen, P., Samaras, C., & Oluwatola, O. A. (2018). *Autonomous Vehicle Technology: A Guide for Policymakers*. RAND Corporation.

Bansal, P., Kockelman, K. M., & Singh, A. (2016). Assessing public opinions of and interest in new vehicle technologies: An Austin perspective. *Transportation Research Part C: Emerging Technologies, 67*, 1-14.

Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., ... & Zieba, K. (2016). End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*.

Goodall, N. J. (2014). Machine ethics and automated vehicles. In *Road vehicle automation* (pp. 93-102). Springer, Cham.

Kalra, N., & Paddock, S. M. (2016). Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part A: Policy and Practice, 94,* 182-193.

Levinson, J., Askeland, J., Becker, J., Dolson, J., Held, D., Kammel, S., ... & Thrun, S. (2011). Towards fully autonomous driving: Systems and algorithms. *2011 IEEE Intelligent Vehicles Symposium (IV)*, 163-168.

Ojango, A. (2023). Pre-programmed rules vs. AI in autonomous driving systems. *Journal of Advanced Transportation*, 123-136.

Van Eperen, L. (2016). The regulation of self-driving cars: A preliminary analysis. *Journal of Law and Mobility, 12*(2), 34-45.

U.S. Department of Transportation. (2020, February 27). How connected vehicles work. https://www.transportation.gov/research-and-technology/how-connected-vehicles-work

Ojango, B. (2023, October 23). *Here's why Tesla's autopilot isn't the best self-driving tech currently*. TopSpeed. https://www.topspeed.com/teslas-autopilot-the-best-adas/

Rambus. (2022a, June 9). *Autonomous vehicles: Everything about self-driving cars explained*. https://www.rambus.com/blogs/autonomous-vehicles-explained/

8.0 – APPENDIX

8.1 – Python Code used for Image Classifications

```python
import cv2
import numpy as np

def get_dominant_color(image, n_colors):
    pixels = np.float32(image).reshape((-1, 3))
    criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 200, .1)
    flags = cv2.KMEANS_RANDOM_CENTERS
    _, labels, centroids = cv2.kmeans(pixels, n_colors, None, criteria, 10, flags)
    palette = np.uint8(centroids)

    _, counts = np.unique(labels, return_counts=True)
    return palette[np.argmax(counts)]

clicked = False

def onMouse(event, x, y, flags, param):
    global clicked
    if event == cv2.EVENT_LBUTTONUP:
        clicked = True

cameraCapture = cv2.VideoCapture(0)
cv2.namedWindow('camera')
cv2.setMouseCallback('camera', onMouse)

# Read and process frames in loop
success, frame = cameraCapture.read()

while success and not clicked:
    cv2.waitKey(1)
    success, frame = cameraCapture.read()

    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    img = cv2.medianBlur(gray, 37)
    circles = cv2.HoughCircles(img, cv2.HOUGH_GRADIENT, 1, 50, param1=120, param2=40)
```

```python
    if circles is not None:
        circles = np.uint16(np.around(circles))
        max_r, max_i = 0, 0
        for i in range(len(circles[:, :, 2][0])):
            if circles[:, :, 2][0][i] > 50 and circles[:, :, 2][0][i] > max_r:
                max_i = i
                max_r = circles[:, :, 2][0][i]
        x, y, r = circles[:, :, :][0][max_i]
        if y > r and x > r:
            square = frame[y-r:y+r, x-r:x+r]

            dominant_color = get_dominant_color(square, 2)
            if dominant_color[2] > 100:
                print("STOP")
            elif dominant_color[0] > 80:
                zone_0 = square[square.shape[0]*3//8:square.shape[0]*5//8, square.shape[1]*1//8:square.shape[1]*3//8]
                cv2.imshow('Zone0', zone_0)
                zone_0_color = get_dominant_color(zone_0, 1)

                zone_1 = square[square.shape[0]*1//8:square.shape[0]*3//8, square.shape[1]*3//8:square.shape[1]*5//8]
                cv2.imshow('Zone1', zone_1)
                zone_1_color = get_dominant_color(zone_1, 1)

                zone_2 = square[square.shape[0]*3//8:square.shape[0]*5//8, square.shape[1]*5//8:square.shape[1]*7//8]
                cv2.imshow('Zone2', zone_2)
                zone_2_color = get_dominant_color(zone_2, 1)
```

```python
                if zone_1_color[2] < 60:
                    if sum(zone_0_color) > sum(zone_2_color):
                        print("LEFT")
                    else:
                        print("RIGHT")
                else:
                    if sum(zone_1_color) > sum(zone_0_color) and sum(zone_1_color) > sum(zone_2_color):
                        print("FORWARD")
                    elif sum(zone_0_color) > sum(zone_2_color):
                        print("FORWARD AND LEFT")
                    else:
                        print("FORWARD AND RIGHT")
            else:
                print("N/A")

        for i in circles[0, :]:
            cv2.circle(frame, (i[0], i[1]), i[2], (0, 255, 0), 2)
            cv2.circle(frame, (i[0], i[1]), 2, (0, 0, 255), 3)
    cv2.imshow('camera', frame)

cv2.destroyAllWindows()
cameraCapture.release()
```