ROAD IMAGE SEGMENTATION FOR AUTONOMOUS CAR

Project report submitted in partial fulfillment of the requirement for the degree of Bachelor of Technology

in

Computer Science and Engineering/Information Technology

By

Bharat Bhatia (191415)

Under the supervision of

Mr. Praveen Modi

to

Department of Computer Science & Engineering and Information Technology

Jaypee University of Information Technology Waknaghat, Solan-173234, Himachal Pradesh

CERTIFICATE

I hereby declare that the work presented in this report entitled "Road Image Segmentation for Autonomous Car" in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science and Engineering/Information Technology submitted in the department of Computer Science & Engineering and Information Technology**,** Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from January 2023 to May 2023 under the supervision of Mr. Praveen Modi, Assistant Professor (Grade-I), Department of CSE Jaypee University of Information Technology. The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Bharat Bhatia,
191415

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Mr. Praveen Modi
Assistant Professor (Grade-I)
Computer Science & Engineering
and Information Technology
Dated:

PLAGIARISM REPORT

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date: ………………………….

Type of Document (Tick): PhD Thesis | M.Tech Dissertation/ Report | B.Tech Project Report | Paper

Name: ____________________________ __**Department:** _________________ **Enrolment No** _________

Contact No. ______________________________**E-mail.** ______________________________________

Name of the Supervisor: ____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): _______________________

____

____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

– Total No. of Pages =
– Total No. of Preliminary pages =
– Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at ……………….(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor) **Signature of HOD**

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

Copy Received on	Excluded	Similarity Index (%)	Generated Plagiarism Report Details (Title, Abstract & Chapters)	
	• All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String		Word Counts	
Report Generated on			Character Counts	
		Submission ID	Total Pages Scanned	
			File Size	

Checked by
Name & Signature **Librarian**

……

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com

ACKNOWLEDGEMENT

Firstly, I express my heartiest thanks and gratefulness to almighty God for His divine blessing makes it possible for us to complete the project work successfully.

I am really grateful and wish my profound indebtedness to Supervisor Mr. Praveen Modi Assistant Professor (Grade-I), Department of CSE Jaypee University of Information Technology, Waknaghat. Deep Knowledge & keen interest of my supervisor in the field of "Data Science" to carry out this project. Her endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts and correcting them at all stages have made it possible to complete this project.

I would like to express my heartiest gratitude to Mr. Praveen Modi, Assistant Professor (Grade-I), Department of CSE, for his kind help to finish my project.

I would also generously welcome each one of those individuals who have helped me straightforwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non-instructing, which have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge with due respect the constant support and patience of my parents.

Name: Bharat Bhatia

Project Group No. 135

Roll No: 191415

TABLE OF CONTENT

LIST OF ABBREVIATIONS

ABBREVIATION	FULL FORM
ML	Machine Learning
CNN	Convolutional Neural Network
FCN	Fully Convolutional Network
ConvNet	Convolutional Network
ResNet	Residual Network
ADAS	Advanced Driving Assistance System
ILSVRC	ImageNet Large Scale Visual Recognition Challenge
AI	Artificial Intelligence
SVM	Support Vector Machine
PSPNet	Pyramid Scene Parsing Network
YOLO	You Only Look Once
VGG	Visual Geometry Group
LiDAR	Light Detection and Ranging
RADAR	Radio Detection and Ranging
GPS	Global Positioning System
SIFT	Scale-Invariant Feature Transform
API	Application Programming Interface
ReLU	Rectified Linear Activation Unit
DNN	Deep Neural Network
NAS	Network Architecture Search
CPU	Central Processing Unit

LIST OF FIGURES

ASTRACT

With production and utilization of resources and other general stuff, wasteful and other unneeded by products are bound to be generated. In India, especially the problem revolving around proper waste management sticks out like a sore thumb. When moving out of Delhi, the National Capital, the famous landfills, which are essentially massive mounds of waste and garbage cannot be missed. Not only in India, but rather globally landfills, ocean waste and debris, waste accumulation and improper waste disposal and management has become a severe problem.

Although the waste production is a problem that we may not be able to tackle with the current level of technology, we definitely can focus and improve the existing infrastructure around waste disposal and management. The unawareness regarding the same is alarming, even the lack of segregating and separating waste into the defined categories, at personal and household-level is obviously of the major component of this nuisance.

For that purpose, we aim to target the problem at the root and present a working solution towards proper segregation of waste before its disposal and subsequent management. Therefore, we propose a project that aims to assist the users in segregating and separating their waste produce in the most convenient way possible. The project proposes to leverage the techniques of Machine learning and Deep Learning through TensorFlow and Keras Convolutional Neural Networks Architecture to analyze the user input and then segregating the waste produce into its different categories. The software thus developed can be deployed using multiple hardware devices, depending on the ease of use and user experience and ubiquity.

Chapter – 1
INTRODUCTION

1.1 Introduction

In the late 1990s, the upgrade of general-purpose computers enabled the processing of vast amounts of data at high speeds. One efficient method was to extract image local features, also known as feature vectors, from an image and apply machine learning algorithms for image identification or segmentation. Unlike rule-based methods, supervised machine learning required a large amount of labeled training data, but did not require researchers to design specific rules. In the 2000s, researchers developed man-crafted features such as scale-invariant feature transform (SIFT) and histogram of oriented gradients (HOG) as image local features based on their knowledge. Combining image local features with machine learning advanced image recognition, particularly in face detection. After this, in 2010s, deep learning was introduced for feature extraction process with learning comes under consideration. Automating the feature extraction process, deep learning is an efficient method for image identification and segmentation. It has outperformed other methods in object recognition competitions and is necessary for autonomous driving with image segmentation. This paper aims to demonstrate the applicability of deep learning in image recognition and its superiority as a simpler approach for self-driving cars.

Self-driving means of transport is one of the famous technologies in today's modern era. Which was just a dream for humans is now a reality. An autonomous car (also known as self-driving car) can be drivable automatically without any human intervention and can understand its surrounding by senses. Different sensors are combined and used to distinguish road with barriers, different objects and pedestrians in the surrounding environment. This idea is also fuel efficient and also increased protection, mobility customer loyalty and many others are advantages of self-driving cars. We would see lower accident rate and insurance rate, less deaths and traffic accidents and so on are benefits for safety. By better navigation from source to destination location self-driving cars may improve traffic problem. Autonomous cars are also good for kids, the disabled, the elderly and the poor all would be able to control cars as they are not capable for non-autonomous vehicles. Travelers stress may be reduced from driving and navigation in any unfamiliar city, also reduce the requirement of parking space, fuel consumption and transportation will be more convenient by using services. Transforming existing vehicles like trains, taxis and buses to autonomous vehicles.

When we talk about India, number of accidents and people killed in 2013 was alone 1,37,000. Speeding is a critical problem. The basic reasons of these accidents are conversations on the phone, using intoxicated while driving and ignorance of traffic laws, and the count is regularly growing, posing a intense danger. Even, if we try to avoid this and raise awareness for traffic regulations and the importance of keeping a healthy riding environment, incidents retain to occur without any warning. Human mistakes cannot be totally eliminated, however accidents ought to be avoided. The improvement and enhancement of this technology have visible exponential increase in current years, beginning from the early radar-based collision detection to modern day technology. Auto-driving vehicles are one from the technology in spotlight in today's era. Which was just a imagination is now become a reality.

Semantic segmentation involves assigning predefined classes to each pixel in an acquired image, such as cars, traffic lights, signs, roads, and pedestrians. This pixel-wise classification is essential for understanding the scene, making it valuable in various fields, including auto-driving cars, satellite images, robotics, precision agriculture, medical images, and facial images. Autonomous driving relies on sensors to acquire data on the surrounding environment and create a comprehensive picture of the driving situation. Semantic segmentation is crucial for scene understanding as it provides rich visual information. Auto-driving depends on the statistics acquired with the aid of sensors of the nearby surroundings with a view to forming an entire photo of the riding state. Because the visible signal may be very wealthy in such statistics, doing semantic segmentation efficiently is essential for scene expertise. The more accurately and quickly we perform semantic segmentation, the better the ego vehicle understands the environment and makes appropriate decisions. However, semantic segmentation is complicated due to the complex relationships between pixels in each image frame and successive frames. Despite the advancements in technologies like deep learning, which have improved semantic segmentation, achieving real-time and accurate semantic segmentation remains a challenging research topic.

The development of autonomous cars has received significant attention in recent past years because of their potential to transform transportation and also to reduce accidents caused by human error. A key component of autonomous cars is their ability to perceive and understand the road environment, including identifying and segmenting the road surface from the surrounding environment. Road image segmentation is an important task that enables autonomous cars to navigate safely and make informed decisions. Road image segmentation is the process of breaking an input image into multiple segments, in which each segment classifies a different object or region in the image. For autonomous cars, the goal of road image segmentation is to identify and segment the road surface from the surrounding environment, such as buildings, trees, and vehicles, this allows the car to understand the road boundaries, detect lane markings, and plan a safe and efficient route.

Various computer vision techniques can be used for road image segmentation, including traditional image processing algorithms and deep learning-based approaches. Deep learning, has shown promising results in recent years, with convolutional neural network (CNN) being widely used for image segmentation tasks.

The purpose of this project is to investigate and implement a deep learning-based approach for road image segmentation in autonomous cars. The project will focus on developing a CNN model that can accurately segment road image in various environmental conditions, including different lighting and weather conditions. The results of the project will contribute to the development of safer and more reliable autonomous cars.

Road image segmentation is a fundamental problem in autonomous driving. An autonomous car needs to perceive and understand its environment to make decisions and navigate safely. Road segmentation is a crucial part of this perception pipeline as it enables the vehicle to distinguish the road from other objects and navigate accordingly.

The goal of road image segmentation is to classify each pixel in an image as either belonging to the road or not. This task is challenging due to the variability in lighting, weather conditions, and the presence of other objects in the scene. Traditional computer vision techniques for image segmentation rely on handcrafted features and heuristics, but these methods often struggle to handle the complexity of real-world scenarios. Recent advances in deep learning have led to the development of neural network models that can perform road image segmentation with high accuracy. These models are trained on large datasets of annotated images, enabling them to learn complex features and relationships between pixels. Road image segmentation is a critical component of autonomous driving systems, and continued research in this area is essential for improving the safety and reliability of autonomous vehicles.

Road image segmentation is a critical task in the development of autonomous cars. It involves dividing an image of the road into different segments or regions, each corresponding to a particular object or obstacle in the scene, such as vehicles, pedestrians, road markings, and traffic signs. This segmentation is necessary for the car to understand its environment and make decisions about how to navigate through it safely.

One approach to road image segmentation is to use deep learning techniques, such as convolutional neural networks (CNNs), which have been shown to be very effective at image segmentation tasks. To train a CNN for road image segmentation, a large dataset of labeled images is required, which can be obtained using various sources, such as cameras mounted on autonomous cars or publicly available datasets. Once the CNN has been trained, it can be used in real-time on images captured by the car's sensors, such as cameras and LIDAR, to segment the road into different regions. These regions can then be used by the car's algorithms to make decisions about how to navigate through the scene, such as adjusting the car's speed and direction to avoid obstacles or stay within the lanes.

In autonomous driving, accurately identifying and classifying various objects and regions in the road scene captured by sensors, such as cameras, is essential. Road image segmentation plays a critical role in this process by dividing an input image into multiple segments that correspond to specific objects or regions of interest. These may include the road surface, pedestrians, vehicles, traffic signs, and traffic lights.

Accurate road image segmentation is essential for autonomous vehicles to perceive their environment, make informed decisions, and safely navigate on the road. For example, by segmenting the road surface, an autonomous vehicle can determine its position and orientation, estimate the curvature and slope of the road, and detect obstacles and hazards. Similarly, by segmenting other objects and regions in the scene, the vehicle can understand the traffic situation, predict the behavior of other road users, and plan its trajectory accordingly.

Road image segmentation is a challenging task due to the complexity and variability of the road scene, which can be affected by lighting conditions, weather, occlusions, and other factors. Therefore, various computer vision techniques like methods on deep learning, have been introduced and applied to road image segmentation, aiming to achieve high accuracy and robustness in different scenarios.

However, it is important to note that road image segmentation is just one of the many challenges facing autonomous cars. Other challenges include object detection and tracking, decision-making, and control. As such, the development of autonomous cars is a complex and ongoing process that involves many different technologies and areas of expertise.

1.2 Problem Statement

Non-autonomous vehicles have been on the road for a while, and a recent online survey found that a significant portion of incidents involving these vehicles are due to human error. Around the world, almost 1.3 million people life are claimed by traffic accidents annually, or 3,287 people on average per day. In India, 1,37,000 persons lost their lives in traffic accidents in just one year (2013). A significant offence is speeding. Talking on the phone, driving while inebriated, and breaking traffic regulations are the main causes of these injuries, and their prevalence is gradually rising, posing a severe concern. Despite our best efforts to increase public understanding of traffic laws and the value of preserving a safe driving environment, incidents nonetheless happen suddenly.

The problem of road image segmentation for autonomous cars involves developing algorithms that can accurately identify and separate the different elements in an image captured by the car's camera, with a specific focus on identifying the regions of the image that correspond to the road surface. This is a typical task for autonomous cars as it allows them to navigate safely and efficiently by understanding the layout of the road ahead. The road image segmentation problem typically involves identifying different types of objects in the scene, for exp buildings, cars, pedestrians, and road markings, and accurately delineating their boundaries. The most critical part of this process is accurately identifying the road surface and distinguishing it from other objects in the scene.

This problem is typically addressed using machine learning techniques such as convolutional neural networks (CNNs) and semantic segmentation algorithms. These algorithms are trained using large datasets of labeled images to learn how to identify different elements in an image and accurately segment them. The goal of road image segmentation for autonomous cars is to develop algorithms that can accurately and reliably identify the road surface in real-time, under different lighting and weather conditions, and in complex urban environments. Achieving this goal will be a critical step towards developing fully autonomous cars that can operate safely and efficiently on our roads.

Although human error will always exist, injuries should be prevented. Technology has undoubtedly saved the day in this situation. From the very first radar-based collision detection to the current technology, the rise of this technology's development and improvement has been exponential in recent years. One of the most discussed technologies in the modern period is self-driving cars. What was once only a fantasy has come true.

An autonomous system's capacity for perception is its capacity to gather crucial information from its surroundings. Enabling autonomous driving is a critical task since it gives important details about the driving environment, such as the areas that are free to be driven in, the locations, the speeds, and a forecast of how the surrounding barriers will behave in the future. LiDAR and camera sensors are used by autonomous cars for perception, enabling them to accurately recognize obstacles and respond appropriately in various situations to prevent accidents. To ensure a safe driving experience, two critical tasks are semantic segmentation and object detection, which are summarized below.

The problem of road image segmentation for autonomous cars involves developing a computer vision system that can accurately identify and segment the road pixels in an input image. The goal of this task is to provide the autonomous car with a precise understanding of the road layout and enable it to navigate safely and smoothly.

The road image segmentation problem can be broken down into the following sub-tasks:

1. Image preprocessing: This step involves applying image enhancement techniques such as contrast adjustment, color space conversion, and noise reduction to improve the quality of the input image.

2. Object detection: The system needs to detect and identify all the objects in image, including road markings, curbs, and other obstacles. This step can be accomplished using techniques such as deep learning-based object detection models or traditional computer vision algorithms.

3. Semantic segmentation: In this step, the system needs to segment the road pixels from the background and classify them into different categories such as lane markings, road surface, and sidewalks. This can be achieved using deep

learning-based semantic segmentation models.

4. Post-processing: Finally, the system needs to refine the output segmentation mask and remove any inconsistencies or errors. This can be accomplished using techniques such as morphological operations and post-processing filters.

Overall, the road image segmentation problem is a critical component of autonomous car systems and requires advanced computer vision techniques to achieve accurate and reliable results.

Some of the major points to use self-driving cars are: -

- Reduced accidents - Vehicles are interconnected to each other with a constant communication, with the help of 360° vision, accidents will be efficiently reduced.

- Higher traffic efficiency - While it is expected that the speed of autonomous vehicles in urban areas will be lower, their traffic efficiency is anticipated to be higher.

- Access to the disabled and people with reduced mobility - Thanks to the fact that the automobile will be autonomous and will require practically no human interaction for its operation.

- Sustainable vehicles - The use of clean energy is anticipated for these vehicles, which would result in virtually zero carbon and greenhouse gas emissions.

This problem is crucial for the development of autonomous driving technology, as it requires the ability to process and understand complex visual scenes in real-time, and to make intelligent decisions based on that information. Effective road image segmentation algorithms can help autonomous cars to detect and avoid obstacles, stay within their lanes, and safely interact with other vehicles and pedestrians on the road.

The problem of road image segmentation for autonomous cars involves accurately identifying and segmenting the region of interest (ROI) in a given input image that represents the road surface. The goal is to distinguish between the road surface and the surrounding objects and background, such as buildings, trees, pedestrians, and other vehicles. This segmentation is critical for a self-driving car to perceive its environment accurately and make informed decisions about its motion planning, trajectory, and actions. The input to the road image segmentation problem is typically a high-resolution color image or video stream captured by one or more cameras mounted on the autonomous vehicle. The output is a binary mask or semantic segmentation map that highlights the pixels belonging to the road and those that do not.

Several challenges make this problem difficult, including variations in lighting conditions, changes in road markings and textures, occlusion by other vehicles, pedestrians, or objects, and the presence of shadows and reflections. In recent years, deep learning-based approaches using convolutional neural networks (CNNs) have shown remarkable success in solving this problem, achieving high accuracy, and robustness under different conditions. These methods typically involve training a CNN on a large annotated dataset of road images and labels, and then using the trained model to perform real-time segmentation on new input images.

The challenge of road image segmentation lies in the variability and complexity of the visual information present in road scenes, such as lighting conditions, weather, occlusions, and variations in object appearance and shape. Therefore, effective solutions require the use of advanced machine learning techniques, such as deep neural networks and computer vision algorithms, trained on large datasets of road images to accurately segment the different components of the scene.

1.3 Objective

My main goal is to make my model as accurate as possible so that I can make it easier to use in everyday situations. To do this, I also intend to implement the project with high accessibility, meaning that everyone should be able to utilize it. I'll work to create a model that's simple to use and comprehend. The initiative emphasizes the significance of security precautions people should be conscious of and follow when using any technology.

The latest offerings from major automakers such as Audi, Volvo, Ford, Google, General Motors, BMW, and Tesla all incorporate autonomous technology. Nowadays, cities such as Columbus are teeming with autonomous vehicles, but a supervising driver is still required to assist the system.

The Society of Automotive Engineers (SAE) has classified self-driving cars into 5 tiers based on the amount of human intervention necessary which are:

Level 0: All manual operation is used to run this vehicle. Every choice is made by a person.

Level 1: ADAS (Advanced Driver Assistance System) The driver can get assistance from an ADAS with steering, braking, or speed. To warn drivers when they veer off the road, ADAS features configurable seat warning choices and rear-view cameras.

Level 2: Like ADAS, vehicles with this level of technology provide the capability for signaling vehicle motion, detection of nearby automobiles, etc.

Level 3: At this level, all driving functions are carried out solely by the vehicle. However, the human driver should be able to take over in constrained situations, such as car parking.

Level 4: Vehicles in this category are capable of all forms of driving and, in certain circumstances, environment control. These require relatively little human intervention and are often precise enough.

Level 5: The vehicles of the future or those included in this level act as virtual drivers and continue to move forward in all circumstances. They are capable of making independent decisions in situations that are known or unknown.

Our main objective is to make a level 5 cars for the improvement of society and make a comfortable car. Our method uses a deep learning model that is based on the idea of "Behavioral Cloning", where the agent picks up knowledge from how people behave. This algorithm is recognizable as being at level 5. This study uses behavioral cloning to create the CNN model and train it on the data. To provide the model more information, image pre-processing and image augmentation are also carried out.

The main objective of road image segmentation for autonomous cars is to identify and segment the road pixels in an input image, allowing the autonomous car to navigate safely and smoothly accurately and efficiently.

More specifically, the objectives of road image segmentation can be summarized as follows:

1. Accurate Road segmentation: The system should be able to accurately identify and segment all road pixels in an input image, including lane markings, road surface, and sidewalks.

2. Real-time performance: The system should be able to process images in real-time, allowing the autonomous car to react quickly to changes in the road environment.

3. Robustness: The system should be able to perform reliably under different lighting conditions, weather conditions, and road layouts.

4. Generalization: The system should be able to generalize well to different geographic locations and road types.

5. Low computational cost: The system should be able to run efficiently on the hardware platform of the autonomous car, without requiring significant computing resources.

6. Accurate detection of the road surface: The system should be able to identify the road pixels with high accuracy, even in challenging lighting and weather

conditions.

7. Precise localization of road markings and other features: The system should be able to detect and localize road markings, curbs, and other features accurately, allowing the autonomous car to navigate and stay within the lane boundaries.

8. Scalability: The system should be able to scale to handle large datasets and different road environments, enabling it to adapt to various use cases and scenarios.

By achieving these objectives, the road image segmentation system can enable autonomous cars to navigate safely and efficiently, leading to reduced accidents, increased efficiency, and improved transportation systems.

1.4 Methodology

Being a computer science engineer, I was considering artificial intelligence (AI), deep-learning, and machine learning in the data analytics space. I chose two-Dimensional Semantic Segmentation for Self-Driving Cars. The procedure for gathering data for two-dimensional image semantic segmentation. The perception stack estimates lanes, road limitations, dynamic objects, and various risks using language segmentation. A classification model is also used to detect intersection closeness. four the native planner employs a rule-based state machine to carry out simple, established rules that are tailored for urban settings. A PID controller that activates the steering, throttle, and brake performs continuous management. Currently, we tend to go into greater detail on the modules. Perception. The perception stacks we're going to talk about here is constructed using a language segmentation network. Every component of the image is classified by the network into one of the linguistic categories below. The network is trained using 2,500 annotated images that were victimized in the coaching environment.

Imitation learning includes both sensory activity and high-level commands. This method makes use of a dataset of driving routes that human drivers in the training city have reported. Drivers issue commands throughout the knowledge collection to express their intents, such as flip signals. Following the lane (by default), driving straight through intersections, turning left through intersections, and turning right through intersections are the common commands we utilize. The observations are captured on film by a camera that faces ahead.

With no human driving traces, reinforcement learning develops a deep network that supports a signal provided by the environment. On tasks like sports and three-dimensional maze navigation, this algorithmic program has demonstrated strong performance in simulated three-dimensional environments. Given the high sample quality of deep reinforcement learning, the asynchronous nature of the technique enables running numerous simulation threads in parallel, which is essential. Every coaching episode has an objective that the vehicle must accomplish under the direction of top-level orders from the topological planner.

The methodology for selecting and using sensors for road image segmentation in autonomous cars typically involves the following steps:

1. Identify the key requirements: The first step is to identify the key requirements for the sensor system, such as accuracy, range, resolution, and speed. This will depend on the specific application and environment in which the system will be used.

2. Evaluate sensor options: Once the key requirements have been identified, a range of sensor options can be evaluated based on their ability to meet these requirements. This may involve testing different types of sensors, such as cameras, LiDAR, and radar, and comparing their performance in different scenarios.

3. Select sensors: Based on the evaluation of different sensor options, the most suitable sensors can be selected for the specific application. This may involve using a combination of sensors, such as cameras for visual information and LiDAR for depth information.

4. Integrate sensors: Once the sensors have been selected, they need to be integrated into the overall system. This may involve developing custom

hardware and software to process and integrate the data from the different sensors.

5. Calibrate sensors: To ensure accurate and consistent performance, the sensors need to be calibrated regularly. This may involve adjusting parameters such as focal length, lens distortion, and sensor alignment.

6. Validate performance: Once the sensor system has been implemented, its performance needs to be validated in real-world scenarios. This may involve testing the system in different environments and conditions, and comparing the results to ground truth data.

1.4.1 Active and Passive Sensors

Road image segmentation for autonomous cars typically uses a combination of active and passive sensors to capture information about the driving environment.

Active sensors, such as LiDAR, emit pulses of light and measure the time it takes for the light to bounce back after hitting objects in the environment. This information is used to create a 3D point cloud representation of the environment, which can be used for object detection and localization. LiDAR sensors can also provide accurate depth information, making them particularly useful for obstacle detection and avoidance.

Passive sensors, such as cameras, capture images of the environment using visible or infrared light. These images can be processed using computer vision algorithms to extract features such as lane markings, traffic signs, and other road features. Cameras can also be used for object detection, although they may be less accurate than LiDAR in certain situations, such as low-light conditions. Other sensors commonly used for road image segmentation in autonomous cars include radar, which can provide information about the speed and distance of objects in the environment, and GPS, which can provide location and mapping information.

Overall, the use of a combination of active and passive sensors allows autonomous cars to capture a comprehensive view of the driving environment, enabling them to make accurate and safe driving decisions in real-time.

Basic Devices used in self-Driving car are: -

Camera**: -** Any autonomous vehicle must have cameras since they can gather the most detailed data about the surroundings and things around the vehicle. Monocular cameras are capable of detecting and categorizing lane geometry and color (such as broken white or double yellow), as well as classifying traffic light colors, recognizing traffic signs, and performing other object detection and classification tasks based on shape and texture information. However, they are unable to provide the depth information required to determine the position and size of the detected objects. Stereo cameras, on the other hand, can capture the relative depth of each point, thereby addressing this limitation.

GPS**: -** A portable GPS (Global Positioning System) tracking system lets you keep tabs on and trace the whereabouts of your car. A vehicle's speed and location can be displayed in real time by high-end trackers.

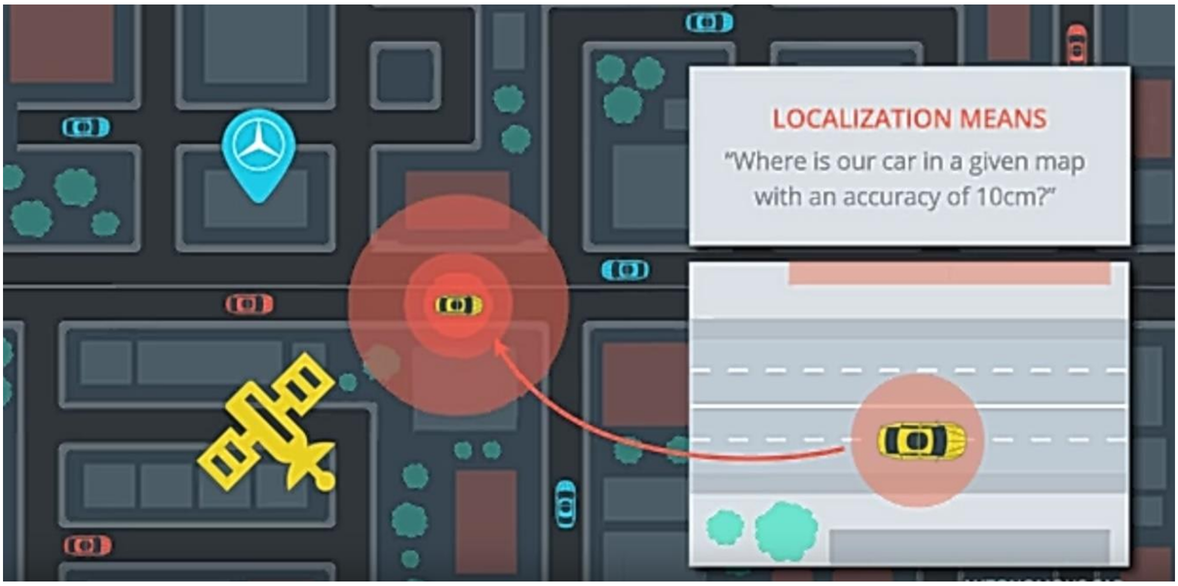

Figure 1: Localization using GPS.[8]

Cheaper trackers provide users the choice of saving these data for later use. When your automobile exceeds the posted speed limit or veers off the marked route, certain GPS trackers can also send you real-time alerts. On the other hand, a GPS tracker system is in charge of tracking the vehicle's actual location as well as keeping track of its speed and direction. These data are either broadcast in real-time or saved in the tracker for later use.

LiDAR**: -** LiDAR (Light Detection and Ranging) sensors emit laser pulses that are received by sensors to identify and classify objects, measure their distance and location precisely, and function well in various illumination and weather conditions. By measuring the time taken to send and receive the pulse in every direction, it accurately determines the distance to the object. It sends out thousands of pulses per second to create a point cloud map, providing a 360-degree view of the surroundings. However, since LiDAR is a depth-based sensor, it cannot be used alone to detect traffic light colors or interpret traffic signs' readings. Therefore, LiDAR sensors are always used in conjunction with camera sensors, which provide color and texture information. Because LiDAR is a depth-based sensor, it cannot be employed as a stand-alone sensor and cannot distinguish between the colors of traffic lights or interpret traffic signs' readings. LiDAR sensors are therefore always used in conjunction with camera sensors.

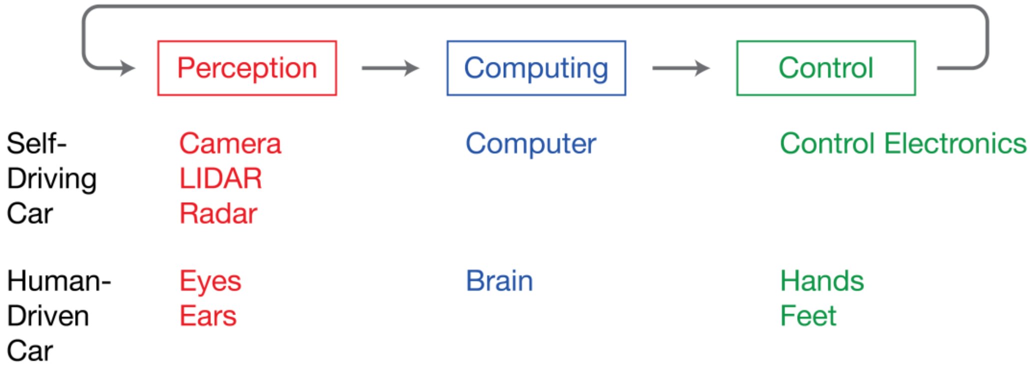

Figure 2: LiDAR in Self-Driving Cars.[8]

RADAR: - Radar sensors emit a radio signal in a particular direction, and the signal is reflected back to the receiver after hitting nearby objects. The distance between the antenna and the object is determined by calculating the travel time of the signal. Unlike other sensors, radars can detect objects in poor weather conditions such as snow, fog, and rain. However, radar is less precise than cameras and LiDAR and does not provide enough information for autonomous vehicle perception, making it unsuitable for detecting and classifying objects accurately. Due to their limitations, radars are only used in specific areas and are typically used in combination with cameras and LiDAR sensors. The information from the various sensors is then combined by the computer to create a coherent image of the immediate environment.

Ultrasonic Sensors: - Ultrasonic sensors transmit high-frequency sound waves to measure the separation between objects in close proximity, simulating the echolocation technique used by bats. Ultrasonic sensors have been a common feature of automobiles for many years. The Advanced Driver Assistance Systems (ADAS) that these sensors are a component of have mostly been employed for parking assistance and blind spot detection. However, with the development of autonomous vehicles and the rise of the automotive IoT, ultrasonic vehicle data is more important than ever.

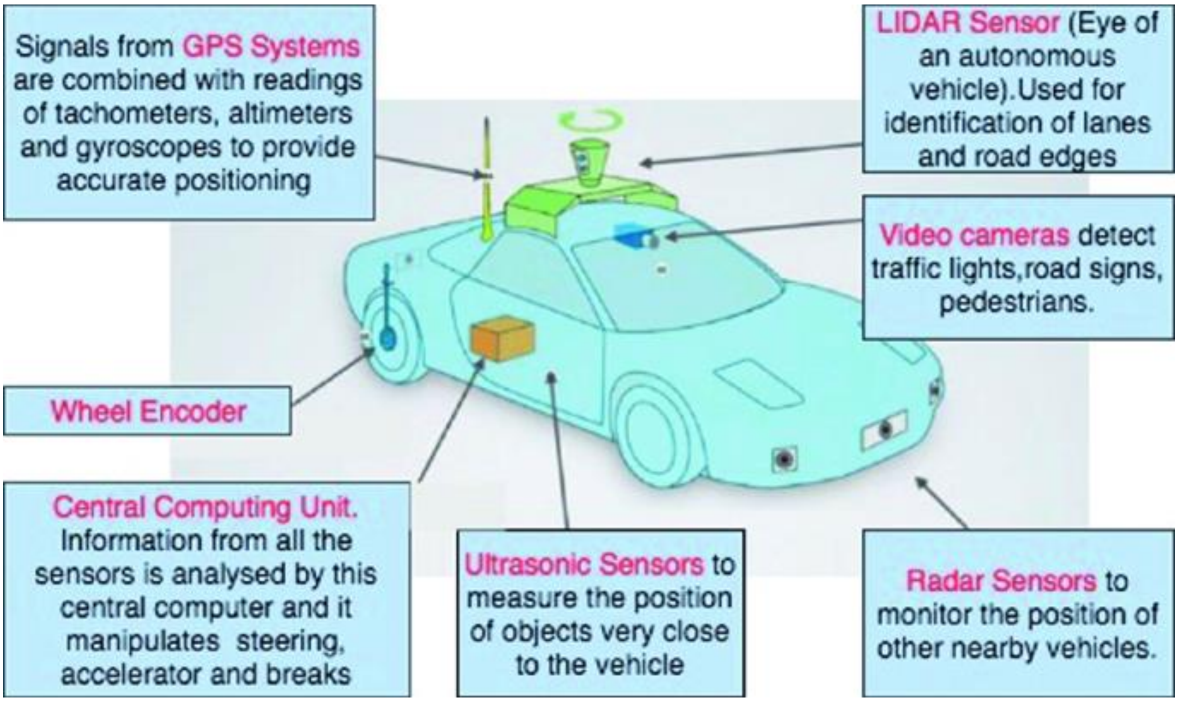

Figure 3: Complete Diagram with all Active and Passive Sensors.[9]

Wheel Encoder: - The ring magnet is also rotated by the motor as it turns the wheel. As the ring spins, the Hall effect sensor placed close by the ring measures changes in the magnetic field. This allows the sensor to determine how many times the motor has revolved. A wheel encoder is situated right behind each motor. The number of rotations of the motor (left or right) are tracked by each wheel encoder. This can be used to figure out how far the robot has travelled or how many turns it has made.

1.5 Organization

The project report is divided into 5 components. The first chapter's objectives are to discuss the problem's description, offer solutions or ideas for additional research, and demonstrate how well the suggested solution works. The literature review for the project is detailed in Chapter 2 with sources listed. Chapter 3 provides explanations of data preprocessing, the dataset, and the suggested framework. The methodology, tools, requirements, system performance criteria, and deadlines for software design are covered in Chapter 4. In Chapter 5, a summary of the project's evaluation, benefits, and future scope is provided.

The organization of a road image segmentation system for autonomous cars typically involves several key components, including:

1. Input Image Acquisition: The system first acquires input images from the cameras or sensors mounted on the autonomous car. These images serve as the input to the segmentation system.

2. Image Preprocessing: The acquired images are preprocessed to improve the image quality and remove noise. This step may include techniques such as contrast adjustment, color space conversion, and noise reduction.

3. Object Detection: The system uses object detection techniques to identify and classify objects in the input image. This step may involve detecting and classifying road markings, curbs, and other obstacles.

4. Semantic Segmentation: The system applies semantic segmentation techniques to segment the road pixels from the background and classify them

into different categories such as lane markings, road surface, and sidewalks.

5. Post-processing: The output segmentation mask is refined using techniques such as morphological operations and post-processing filters to remove inconsistencies or errors.

6. Output: The system outputs the segmented image, providing the autonomous car with an accurate understanding of the road layout and enabling it to navigate safely and effectively.

The organization of the system may vary depending on the specific techniques and algorithms used. However, the key components listed above provide a general overview of the process of road image segmentation for autonomous cars.

Chapter – 2
LITERATURE SURVEY

In [1], the authors have analyzed and discussed the theoretical and practical challenges faced in the implementation of autonomous vehicles. Topics discussed range from special driving licenses to accident and privacy issues. This paper provides an overview of the current state of the art in the key aspects of autonomous driving. Diakaki et al. [2] have given an overview of proposed and available VACS and discuss their perspectives from the motorway traffic management point of view. Classifications of the different systems in this respect are also provided, while SWOT (Strengths, Weaknesses, Opportunities, Threats) analyses are used to identify specific exploitation ways. Gupta et al. [3] propose an environment perception framework using state representation learning (SRL). The proposed and existing models are compared through simulations run on DonKey simulator by examining the variations in policy loss, value loss, reward function and cumulative reward over the learning process. In [4] the authors design a reward function for the Markov Decision Process, implemented on the Deep Neural Network through the Maximum Entropy Principle (MEP). Zhu et al. [5] propose an autonomous car framework based on deep reinforcement learning, where the model learns through trial-and-error interactions based on reward functions. In [6], the authors attempt to tackle the current challenges encountered in designing AI architectures for autonomous driving. The paper follows both the approaches, convolutional and recurrent neural networks and reinforcement learning paradigm. Rasouli et al. [7] discuss various methods of studying pedestrian behavior and identify how the factors are interrelated. They review the practical applications aimed at solving the interaction problem, including design approaches for autonomous vehicles that communicate with pedestrians and visual perception and reasoning algorithms tailored to understanding pedestrian intention.

To gain insight into the advancements made in autonomous driving research over the past few years, it is crucial to perform a literature review that encompasses the various domains of application where autonomous driving has progressed, and also to determine the research voids. For a self-driving car to operate at high speeds, it must be able to detect drivable regions more precisely and with sufficient time.

Road image segmentation is a important task for autonomous cars as it helps the vehicle to identify and differentiate different parts of the road such as lanes, curbs, and sidewalks. There have been several approaches proposed in the literature to tackle this problem. In this survey, we will discuss some of the recent approaches to road image segmentation for autonomous cars.

1. Fully Convolutional Networks (FCN): FCN is a popular approach for semantic segmentation. In the context of road image segmentation, FCN has been used to classify every pixel of the image as belonging to the road or not. Several variations of FCN have been proposed in the literature, such as SegNet, U-Net, and DeepLabv3+. These approaches have been shown to achieve high accuracy in road image segmentation.

2. Encoder-Decoder Networks: Encoder-Decoder Networks are also popular for semantic segmentation. They consist of an encoder that extracts features from the input image and a decoder that reconstructs the output image. Examples of encoder-decoder networks for road image segmentation include the DilatedNet and RefineNet.

3. Multi-Task Learning: Multi-Task Learning (MTL) is a technique where a single model is trained to perform multiple tasks simultaneously. In the context of road image segmentation, MTL can be used to predict both the road and the lane markings. This approach has been shown to improve the accuracy of road image segmentation.

4. Graph-Based Approaches: Graph-based approaches use a graph to model the relationship between pixels in an image. The nodes in the graph represent pixels, and the edges represent the relationship between them. Examples of graph-based approaches for road image segmentation include the GraphCut and GrabCut algorithms.

5. Deep Reinforcement Learning: Deep Reinforcement Learning (DRL) is a technique that combines deep learning and reinforcement learning. In the context of road image segmentation, DRL can be used to learn a policy that maximizes the accuracy of road image segmentation. This approach has been shown to achieve high accuracy in road image segmentation.

In conclusion, road image segmentation is a crucial task for autonomous cars, and several approaches have been proposed in the literature to tackle this problem. These approaches include fully convolutional networks, encoder-decoder networks, multi-task learning, graph-based approaches, and deep reinforcement learning.

A key element required in any self-driving system is the ability to perceive and comprehend the environment. To gain public acceptance of this new technology, it is crucial to have accurate real-time visual signal processing that produces pixelwise categorized images, known as semantic segmentation, for scenario comprehension. Until recently, deep learning algorithms were not capable of achieving the required level of processing efficiency and accuracy due to the complex interactions between pixels in each frame of the camera data. This paper presents a successful approach to semantic segmentation for self-driving cars. Our model combines state-of-the-art techniques such as feature pyramid networks and bottleneck residual blocks with deep learning architectures like convolutional neural networks and autoencoders. The CamVid dataset, which has undergone extensive data augmentation, was used to train and test the model. We compare our results with various published baseline models to validate our proposed model.

Computer vision has made significant strides in the past decade, largely due to the rapid progress in deep learning research. Convolutional neural networks (CNNs) have played a key role in this success, significantly improving the accuracy and speed of tasks such as object recognition and detection. The success of early CNN models such as LeNet [9] and AlexNet [10] (released in 2017) led to a proliferation of proposed CNN works. VGG [11] was successful on the ImageNet dataset [12] due to its numerous configurations and increased the number of weight layers to 16 or 19.

Similar to other areas of computer vision, the field of semantic segmentation has made significant progress in the deep learning era. In addition to CNNs, AEs have been utilized to create more effective semantic segmentation models than previous ones. Recent research on semantic segmentation has focused on convolutional autoencoders (CAEs), which use convolutional and deconvolutional layers as the encoder and decoder components. These CAEs are based on CNN models initially developed for object recognition and detection.

To achieve semantic segmentation, FCN [13] introduced a fully convolutional architecture with a large number of parameters, and it was among the first to remove fully connected layers. SegNet [14] and SegNet-Basic [15] employed the VGG design as the basis for their encoder and decoder, using the encoder's pooling indices for the decoder's upsampling process. Other methods, such as UNet [16], used skip connections between the encoder and decoder and data augmentation to improve segmentation accuracy.

In this article titled "Review of Real-Time Object Detection in Remote Sensing Image Based on Visual Perception and Memory Reasoning" by the College of Field Engineering, PLA Army Engineering University in Nanjing, China, they used a cascaded convolution neural network for object detection. In contrast, I chose semantic segmentation for object detection using a u-net convolution neural network. Another helpful resource for object detection is the Cityscape Dataset for Semantic Urban Scene Understanding, which was reviewed in a paper by DaimlerAG&D MPI Informatics. To improve object detection accuracy, large-scale datasets like Cityscape are often used in conjunction with convolutional neural networks. Semantic segmentation involves grouping together image components that belong to the same class, and is a key component of visual perception for self-driving cars that rely on camera sensors to perceive their environment. In this paper, I utilized the Cityscape dataset and a u-net convolution neural network to focus on object detection in 2D images captured by the self-driving car's camera.

Despite the advancements made in models and architectures such as PSPNet, Dilated, and DeepLab, the development of real-time semantic segmentation remains a popular area of research, particularly in industries like autonomous driving and robotics, where highly accurate semantic segmentation is required with minimal processing time. Creating lightweight segmentation models without sacrificing accuracy is particularly challenging since images contain an abundance of semantic information that requires a large number of trainable parameters to capture their complexity. While some models like FPN have fewer parameters, the encoder architecture used in the original FPN model has a structure that is similar to ResNets, which may pose challenges when applied in real-time scenarios. To achieve real-time or embedded semantic segmentation, models such as ApesNet, Enet, ESPNet, and ESCNet were created with fewer parameters, but these models may not provide the necessary segmentation accuracy for critical applications like road scene interpretation in autonomous vehicles.

Many datasets have been gathered over the past few years in a variety of cities to increase the variety and complexity of urban street views for self-driving applications. The first dataset containing videos that have been semantically annotated is the Cambridge-driving Labelled Video database (CamVid) [26]. The collection is only tiny, with 701 hand annotated photos divided into 32 semantic classifications. Later, a collection of computer vision tasks, including stereo, optical flow, 2D/3D object detection and tracking, are included in the KITTI vision benchmark suite [28]. The main semantics focus is on detection, with up to 15 autos and 30 pedestrians in each of the 7,481 training and 7,518 test photos that are annotated with 2D and 3D bounding boxes. However, the benchmark for semantic segmentation is relatively poor because very few images have pixel-level annotations. Most recently, the Cityscapes dataset [31], which contains 30 semantic classes, was specifically gathered for 2D segmentation. Specifically, 20,000 images have coarse annotations while 5,000 have detailed annotations. Despite the availability of video frames, only one image is manually annotated for each video. As a result, tasks like video segmentation cannot be carried out. Like this, the Mapillary Vistas dataset [29], which contains 25,000 photos with 66 object types, offers a larger collection of photographs with fine annotations. The Toronto City benchmark [27] gathers LIDAR data and photos from moving cars and drones, including stereo and panoramas. Despite the size of the dataset, which includes the Toronto region, as stated by the authors, it is not possible to manually label each frame's individual pixels. Building footprints and roadways are the only two semantic classes offered as segmentation benchmarks. In the BDD100K database [30], there are 100K raw video sequences that include more than 1000 hours of driving time and more than 100 million pictures. Like the Cityscapes, one image is chosen for annotation from each video clip. Ten thousand photos are annotated at the pixel level and one hundred thousand images at the bounding box level.

These techniques may succeed in situations with distinct feature points, but they are still impractical in environments with billions of points at the city scale. They may also fail in situations with low texture, repeating structures, and occlusions. So, recently, hierarchical deep learning features have been proposed for localization. PoseNet [35], [33] can estimate pose in 10 ms w.r.t. a feature-rich environment made up of notable landmarks using a low-resolution image as input. Consider incorporating semantic hints as a more reliable localisation representation [32]. But in a street-view scenario, leaving a road with trees aside, it's common for no noteworthy landmark to show up, which could undermine the visual models. Therefore, GPS/IMU signals are essential for reliable localization in these scenarios [37], while the challenge now is determining the relative pose between the camera view from a noisy pose and the actual pose. Recently, researchers [39], [40] suggested stacking the two images as a network input for relative camera posture of two perspectives. Our method, which produces better results in our testing, combines the real image with an online-rendered label map from the noisy pose.

S. NO.	AUTHORS	APPROACH	DATASET
1	Diaz et al. (2018) [1]	Analysis of current systems was done	Existing infrastructure and systems were analyzed
2	Diakaki et al. (2015) [2]	SWOT analysis	Existing infrastructure and systems were analyzed
3	Gupta et al. (2021) [3]	Variational autoencoder (VAC), Deep deterministic policy gradient (DDPG), soft actor critic (SAC), State representation Learning (SRL)	Simulations on DonKey simulator were run.
4	You et al. (2019) [4]	Markov Decision Principle (MDP), Deep Neural Network (DNN), Reinforcement Learning	Traffic simulator using Pygame was developed and simulations were run
5	Zhu et al. (2018) [5]	Deep Reinforcement Learning (DRL)	Historical driving data is recorded and fed to the model
6	Grigorescu et al (2019) [6]	Convolutional Neural Networks, Recurrent Neural Networks, Reinforcement Learning	Scale, a self-driving dataset with over 1.4 million images
7	Rasouli et al. (2019) [7]	Vision-based intention estimation algorithm, intention prediction algorithm	Historical driving dataset was created through laser scanning and fed to model

Chapter – 3
SYSTEM DESIGN AND DEVELOPMENT

The system design and development for road image segmentation for autonomous cars involves the following steps:

1. System requirements: The first step is to define the requirements for the road image segmentation system based on the objectives and use cases. This step involves identifying the input and output requirements, performance metrics, and constraints.

2. Data collection and preprocessing: The next step is to collect a large dataset of road images with labeled ground-truth segmentation masks. This dataset needs to be preprocessed to enhance the image quality and remove any noise or artifacts.

3. Model selection and development: Once the dataset is prepared, the next step is to select a suitable model architecture for the road image segmentation task. This may involve using deep learning-based models such as U-Net, SegNet, or DeepLab, or traditional computer vision algorithms such as thresholding or edge detection. The model is then developed and trained on the labeled dataset using techniques such as supervised learning or transfer learning.

4. Model evaluation and optimization: Once the model is developed and trained, it is evaluated on a validation dataset to measure its performance and identify any issues such as overfitting or underfitting. The model is then optimized by tuning its parameters and architecture to improve its performance and reduce its computational requirements.

5. Real-time implementation: Once the model is optimized, it is deployed to perform road image segmentation in real-time. This may involve integrating the model into an autonomous car system and optimizing its performance for low-

latency inference.

6. Testing and validation: The system is tested and validated to ensure that it meets the performance metrics and requirements defined in the first step. This may involve testing the system on a variety of road environments, lighting conditions, and weather conditions to ensure that it is robust and can handle various scenarios.

7. Maintenance and updates: Regular maintenance and updates are essential to ensure the accuracy and robustness of the system over time. This may include retraining the model on new data or upgrading the model architecture to enhance its performance.

By following these steps, it is possible to design and develop an accurate and reliable road image segmentation system for autonomous cars that can enable them to navigate safely and efficiently.

We must recognize the four basic components of self-driving cars in order to comprehend how they operate.

1. Perception: Autonomous vehicles rely on perceptual sensing to gather input from their sensors and understand their surroundings, much like a human driver's sense of sight. To achieve this level of perception, self-driving cars must have three essential sensors: RADAR, LiDAR, and a camera.

2. Localization: Through the use of high-precision sensors such as RADAR, LiDAR, and cameras, autonomous vehicles can perceive their surroundings and gain an understanding of the road and lane configurations. This level of precision enables the vehicle to plan lane changes, identify lane paths, and determine if a lane is merging or branching. Localization algorithms are used to calculate the positioning and orientation of the vehicle as it travels. To enhance efficiency and classify various objects, deep learning is commonly utilized. Point data

is relied upon by some neural network frameworks to estimate the 3D position and orientation.

3. Prediction: In autonomous driving, prediction aims to predict the path of an opposing vehicle and take appropriate action to avoid collision. This is a challenging task due to the complexity of human drivers. The autonomous vehicle can use its 360-degree view of the surroundings to sense, collect, and interpret all the information. Deep learning algorithms can model this data during training. During inference, the same approach can help the vehicle be prepared for any maneuvers that may require it to brake, slow down, stop, change lanes, or take other actions.

4. Decision-Making: A thorough understanding of the overall framework of the decision-making system in autonomous driving technology is necessary for designing effective methods in order to conduct specific research for decision-making. The automobile should have access to sufficient data to choose the appropriate course of action. We employ deep reinforcement learning for decision-making (DRL). DRL is more explicitly based on a decision-making mechanism known as the Markov decision process (MDP).

Four goals to achieve in our model are: -

- ROAD SEGMENATATION
- 2D OBJECT DETECTION
- OBJECT TRACKING
- 3D DATA VISUALIZATION

3.1 How Self-Driving car works?

Self-driving cars, also known as autonomous cars, work by using a combination of sensors, software, and algorithms to perceive the environment, make decisions, and control the car's movements. self-driving cars use advanced technologies to perceive the environment, plan a route, and safely navigate through the environment without human intervention. The system needs to be highly accurate, reliable, and robust, and it needs to be optimized for real-time performance to ensure that it can react quickly to changes in the environment. The realization of a self-driving car in real life requires the implementation of three essential technologies.

3.1.1 Sensors

A unified system with an interior display panel connects all of these sensors, including LIDAR, camera, collision warning, ultrasonic, and blind spot monitoring, to facilitate navigation. Autonomous cars typically use a combination of these sensors to perceive their environment accurately and make decisions in real-time. The data from these sensors is processed by the autonomous car's software to enable safe and efficient navigation.

A common approach is to integrate these sensors into a sensor fusion system that combines data from multiple sensors to generate a more complete understanding of the surrounding environment. The sensor fusion system is used to perceive the environment and make decisions based on the data. The data is processed in real-time, and the system's output is used to control the car's movements.

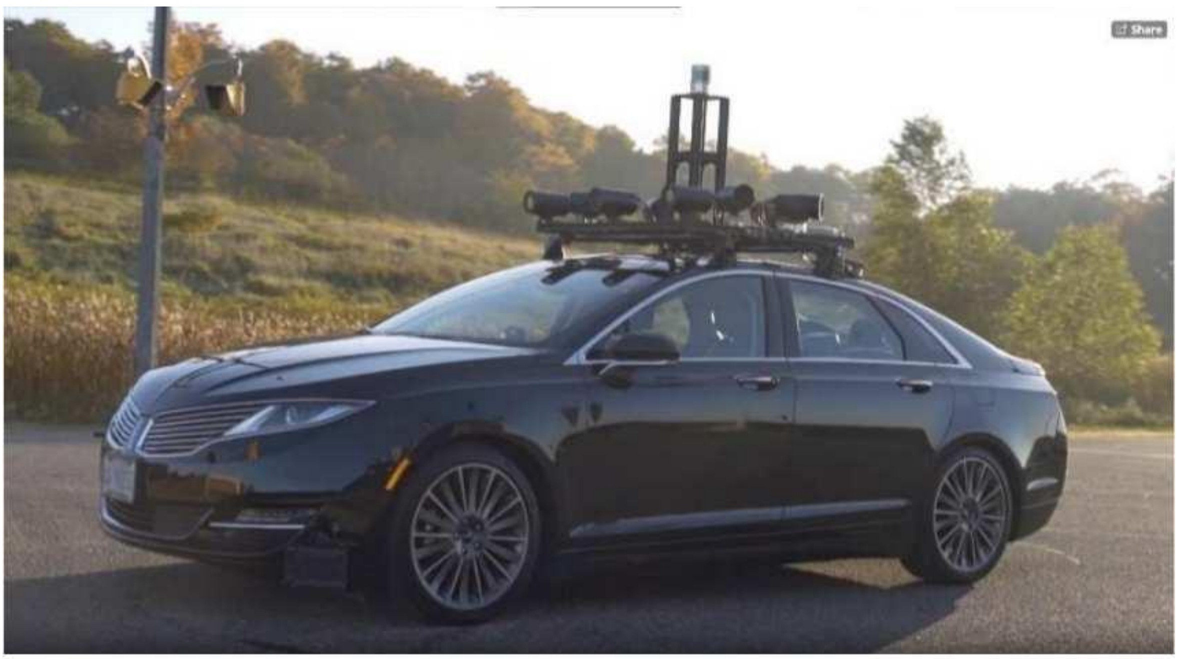

Figure 4: Sensors on Self-Driving Cars. [10]

3.1.2 IOT Connectivity

The automated self-driving car manufacturer develops a software program that utilizes cloud computing to connect with other IoT devices that provide information such as real-time traffic updates, weather conditions, maps, and road surface conditions. This enables the vehicle to analyze its surroundings more effectively and make more informed decisions.

3.1.3 Software and Algorithm

The primary purpose of control algorithms is to analyses the ideal path of action. Semantic segmentation is used to categories drivable regions, people, trucks, and other objects in 2D images of objects that have been detected by cameras.

Combining the three aforementioned stages creates the fundamental prerequisites for turning a real car into a self-driving car. In the paper, we'll talk about the Software & Algorithm portion, and I've chosen Semantic Segmentation to explore identifying objects in 2D camera-captured images.

3.2 Semantic Segmentation

3.2.1 (scene Understanding) Semantic segmentation

Semantic segmentation is one of the major issues facing computer vision today. Semantic segmentation is one of the high-level tasks that, when taken as a whole, lay the road to thorough scene comprehension. The fact that more applications depend on drawing conclusions from pictures highlights the significance of scene understanding as a fundamental computer vision problem. Some of these applications include driverless cars, virtual reality, and human-computer interaction. Convolutional neural networks, which outperform other methods in terms of accuracy and efficiency, are often used in deep architectures to solve various semantic segmentation problems as a result of deep learning's recent surge in popularity.

Clustering elements of an image that belong to the same object class together is what semantic segmentation is. Due to the categorization of every pixel in an image, it is a type of pixel-level prediction. In addition to self-driving cars, other fields can benefit from semantic segmentation.

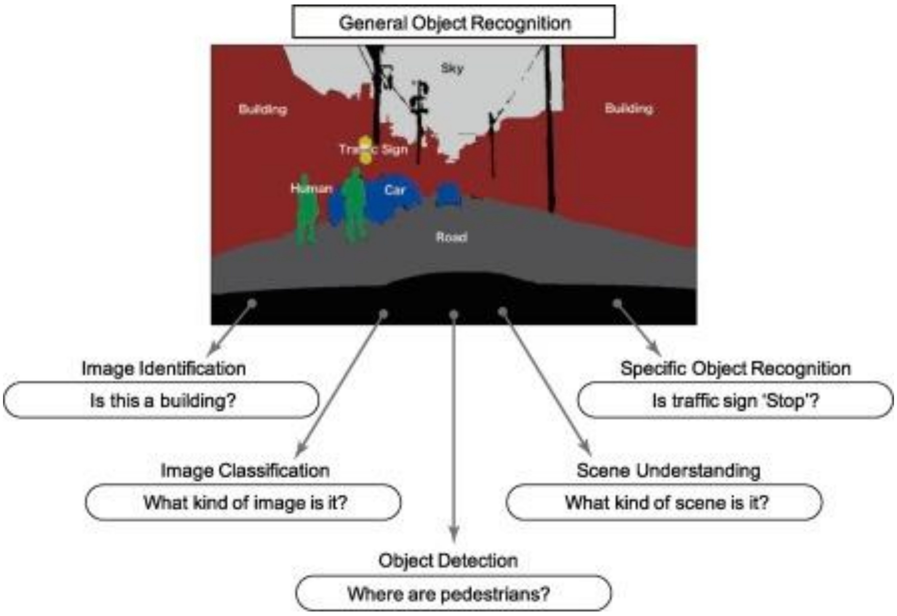

Figure 5: Segmentation of general object recognition. [11]

3.2.2 Specific Object Recognition

Recognizing a particular object is referred to as specific object recognition, which involves assigning attributes to objects using proper nouns. The use of SIFT for feature point extraction from images and a voting mechanism based on measuring the distance between feature points of reference patterns helps accomplish specific object detection. Though machine learning is not directly employed in this case, the performance was enhanced by deep learning each SIFT operation using the learnt invariant feature transform (LIFT) developed in 2016.

3.3 Deep Learning-Based Image Recognition

3.3.1 ConNet for Semantic Segmentation

A type of hierarchical model called a convolutional neural network is capable of taking raw data inputs, such as unprocessed audio or RGB images. This type of network employs operations such as convolution, pooling, and mapping of non-linear activation functions, stacked in layers to extract high-level semantic information from the raw data input. This process is commonly referred to as a "feed-forward operation," with each layer consisting of different types of operations such as pooling and convolutional layers. The final layer of a convolutional neural network typically converts the target task's goal function, such as classification or regression.

One common problem that can be addressed using convolutional neural networks is semantic segmentation, which involves taking camera images as input and producing a class classification for each pixel within the image as output. This problem can be formulated as a function approximation problem, with a ConvNet model used for object detection. The model typically consists of a feature extractor followed by an output layer.
By convoluting kernel (weight filter) on to the input image, CNN calculates the feature map corresponding to the kernel. Given that there are several kernels, feature maps corresponding to the various kernel types can be generated. The pooling feature map then reduces the size of the feature map.

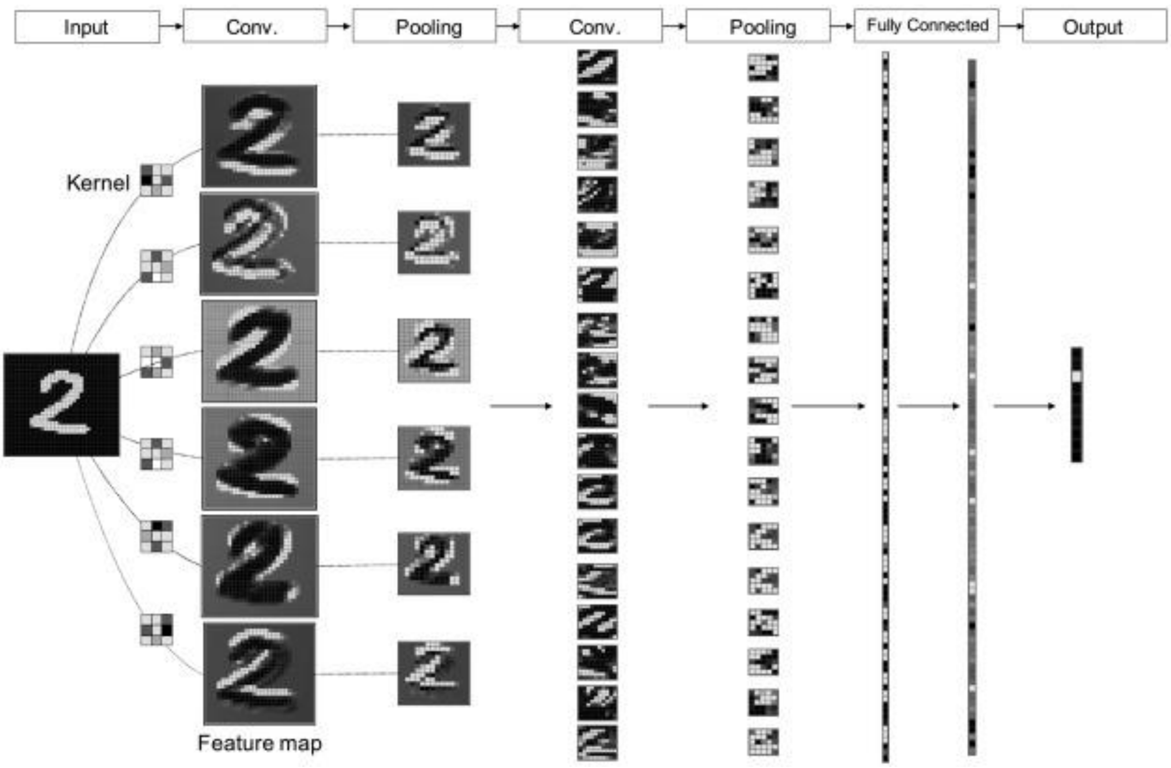

Figure 6: Basic Structure of CNN. [12]

Hence, minor geometric changes in the input image such as translation or rotation can be accommodated. Convolution and pooling operations are typically employed to extract feature maps. The resultant feature map is then fed into fully connected layers, which produce the probability of each class as output.

3.3.2 Advantages of ConvNN compared to conventional machine learning.

Samples of the first convolution layer kernels of AlexNet, which was developed for the 2012 ILSVRC (ImageNet Large Scale Visual Recognition Challenge) to classify 1000 objects, can be visualized to understand its operations. AlexNet comprises five convolutional layers and three fully connected layers, and its output layer has 10,000 units representing the classes. The filters in AlexNet extract directional information from edge, texture, and color information automatically. Comparing the CNN filter's performance as a local image feature to the HOG in the human detection task was explored.

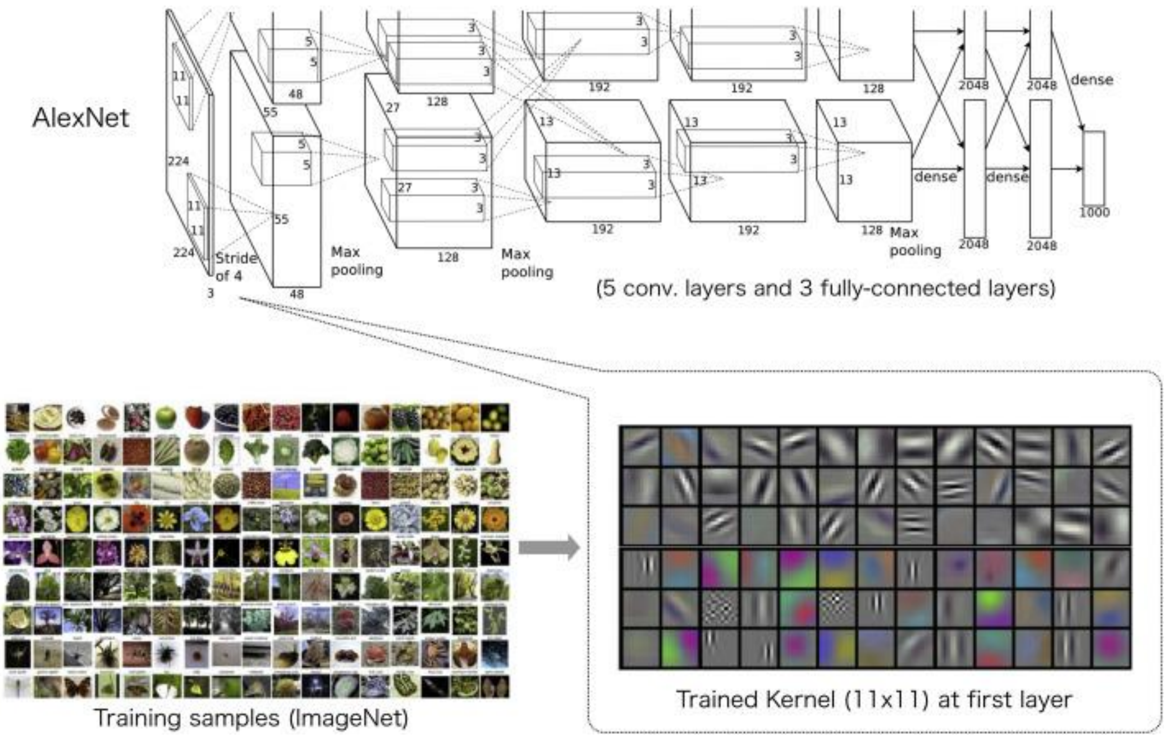

Figure 7: Network Structure of AlexNet and Kernels. [11]

CNN can perform various image recognition tasks, including object detection, image classification, and semantic segmentation, by designing an appropriate output layer for each task. When the output layer is configured to produce the probability of each class and detection region for each grid, it can be transformed into a network structure that performs object detection. On the other hand, the output layer for semantic segmentation should be created to output the class probability for each pixel. In both cases, convolution and pooling layers are commonly used as modules.

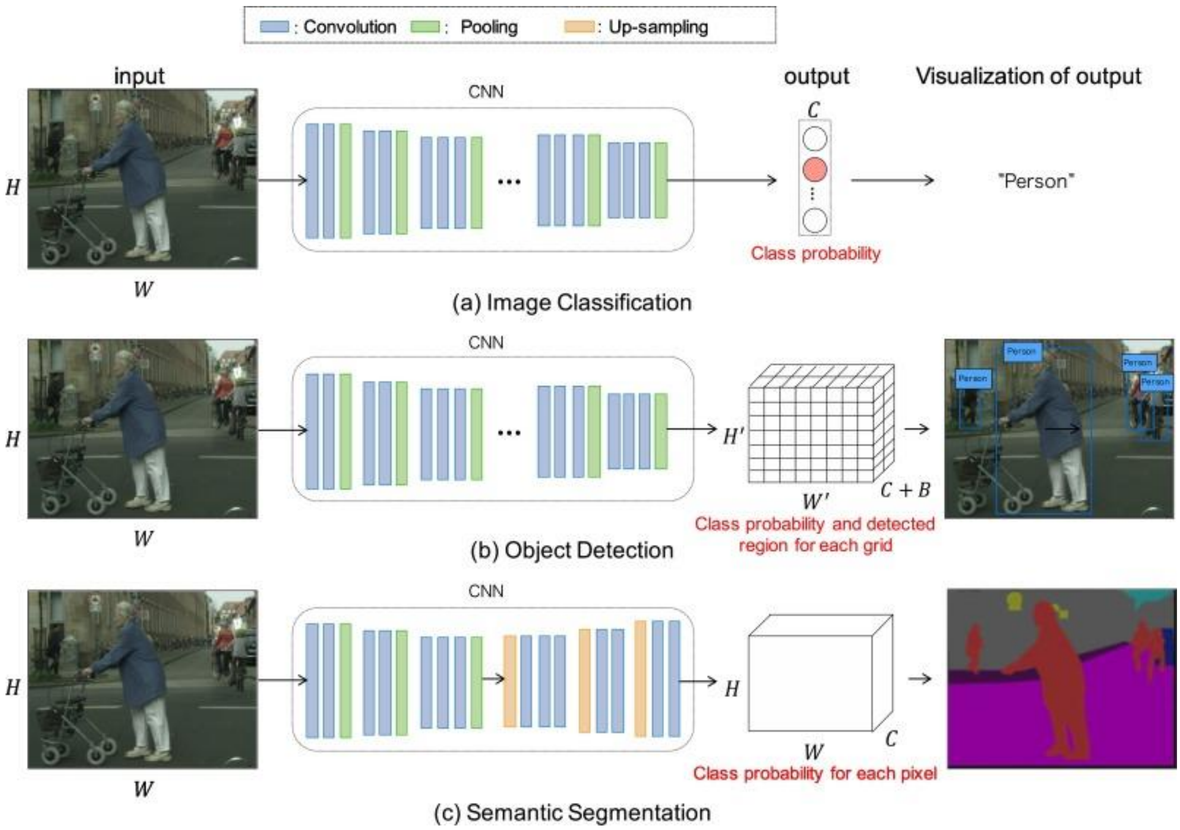

Figure 8: Semantic Segmentation. [11]

3.4 Layer for ConNet

3.4.1 Feature Extractor Layer

A convolutional neural network utilizes a linear process that involves multiplying input data with a set of weights, similar to a conventional neural network. However, in a convolutional neural network, this process takes place in the context of a convolution, which is designed for two-dimensional input. The multiplication occurs between an array of input data and a filter or kernel, which is a two-dimensional array of weights.

For object detection, we can use a VGG design feature but with reduced resolution of 0.5. However, convolution layers may cause the resolution to increase. The output will undergo a sixteen-fold down sampling, and the features will be labeled as code classes 1, 2, and 3 as early as possible to simplify coding and memory management.

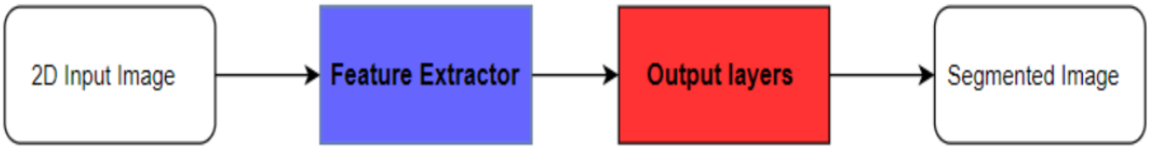

Figure 9: Feature Extractor Layer ConvNet. [10]

3.4.2 Upsampling Layer

In deep learning, upsampling layers are used to increase the spatial resolution of an image or feature map. They are commonly used in convolutional neural networks (CNNs) for tasks such as image segmentation, where the goal is to classify each pixel in an image.

There are several types of upsampling layers, including:

1. Nearest neighbor upsampling: This is the simplest form of upsampling, where each pixel in the original image is replicated to increase the image's size. This method does not involve any computation and is fast, but it can result in blocky artifacts in the image.

2. Transposed convolution: Transposed convolution, also known as deconvolution, involves creating a learnable kernel that is used to upsample the feature map. The kernel is applied to the feature map, and the resulting output is the upsampled feature map.

3. Subpixel convolution: Subpixel convolution is a type of transposed convolution that is specifically designed for upsampling. It involves rearranging the channels of the feature map and then applying a regular convolution to

increase the spatial resolution.

4. Pixel shuffle: Pixel shuffle is another type of transposed convolution that involves rearranging the channels of the feature map and then reshaping the feature map to increase the spatial resolution.

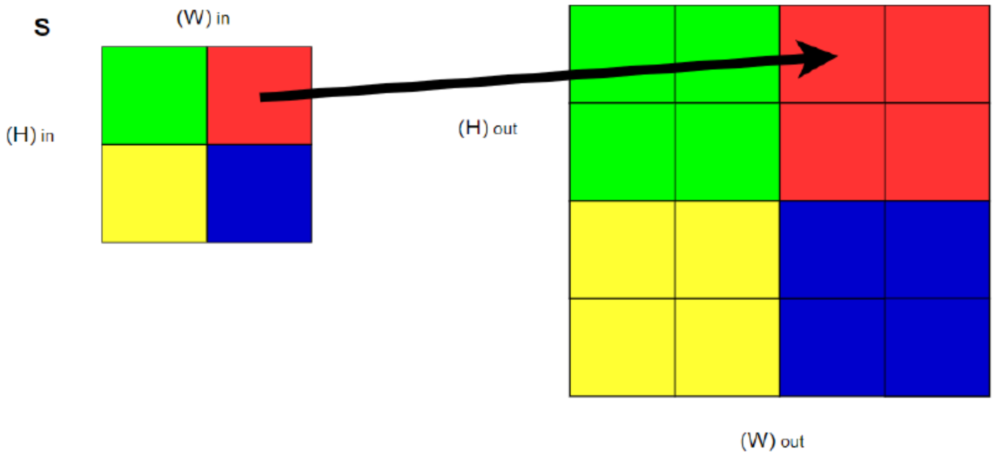

Figure 10: Upsampling Explanation. [10]

Upsampling layers are typically used in conjunction with downsampling layers, such as max pooling or stride convolution, which reduce the spatial resolution of the feature map. The combination of downsampling and upsampling layers allows the network to extract high-level features from the input image and then use them to produce a pixel-wise classification of the image.
Upsampling involves injecting artificially produced data points (corresponding to the minority class) into the dataset. The counts for each label are nearly equal after this procedure. The model won't tend to favor the class with the majority of members thanks to this equalization process. Additionally, the border line (interaction) between the target classes is unaffected. Additionally, because of the extra information, the upsampling technique introduces bias into the system. By using closest neighbor upsampling on a 2x2 picture pixel, upsampling will improve the image resolution. Each picture element in the image patch has a completely separate value represented by its color. It has upsampling multiplier factor S as a result.

- (W)out= S x (W)in
- (W)out= S x (H)in
- (D)out= (D)in x (W)

In 2D Image Semantic Segmentation for Self-Driving Car Using Convolution Neural Network

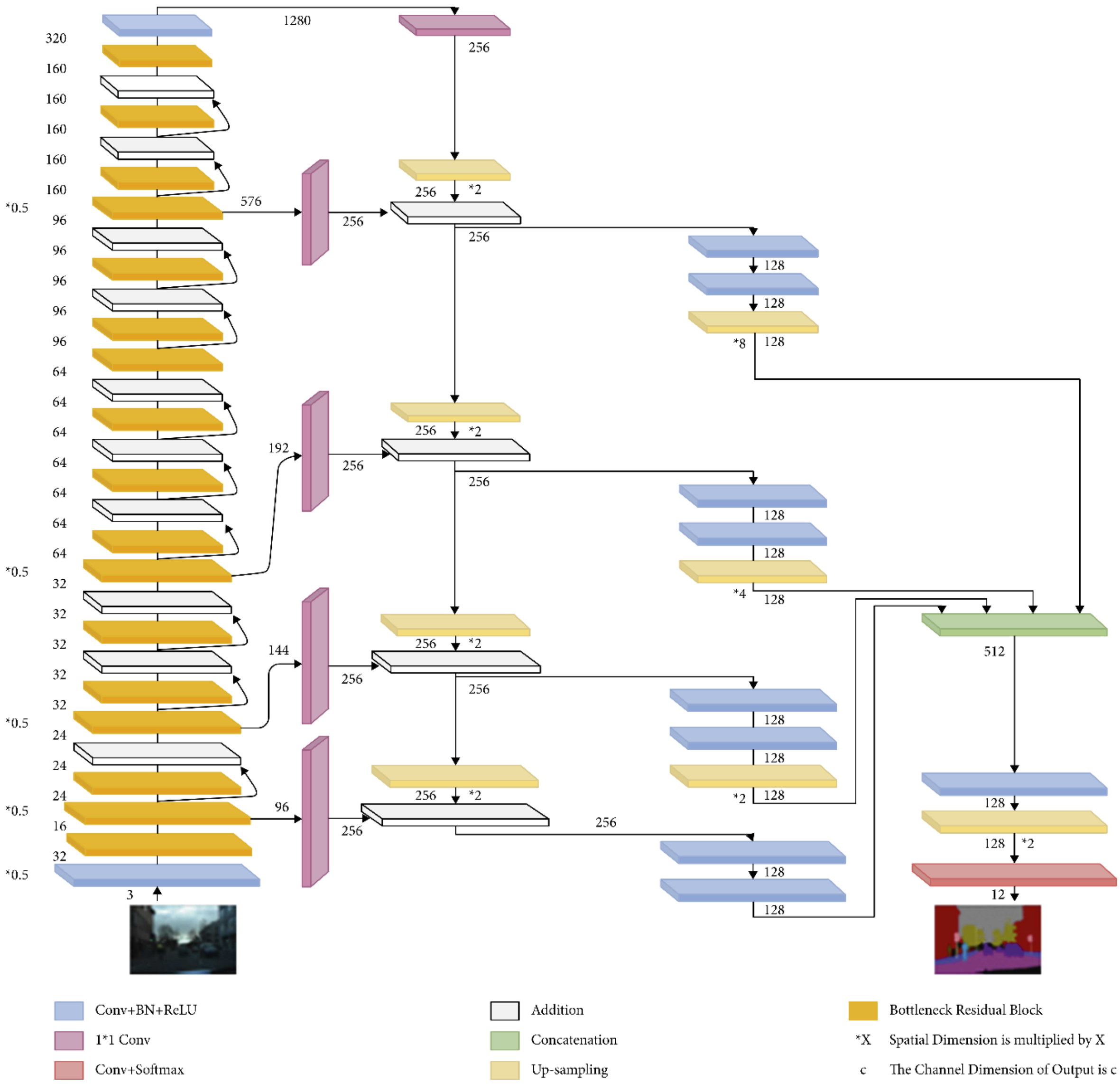

Figure 11: Architecture of bottleneck residual block. [12]

3.4.3 Feature Decoder Layer

Semantic segmentation is a challenging task, especially when it comes to obtaining precise boundaries around objects. The use of upsampling layers may result in sharp boundaries in the output, which can pose a challenge. To overcome this issue, we employed a feature decoder that utilizes the classes generated by feature extraction to create clear boundaries, thereby improving the accuracy of semantic segmentation.

Figure 12: Feature Decoder Layer in ConvNet. [10]

3.4.4 Softmax Layer

To reduce complexity, the depth of the map in a 2D image is often reduced. One of the simplest segmentation techniques is upsampling, which is followed by a linear output and then a Softmax layer to perform segmentation. In order to prevent depth reduction in the output of a segmented image, the Softmax layer can be used.

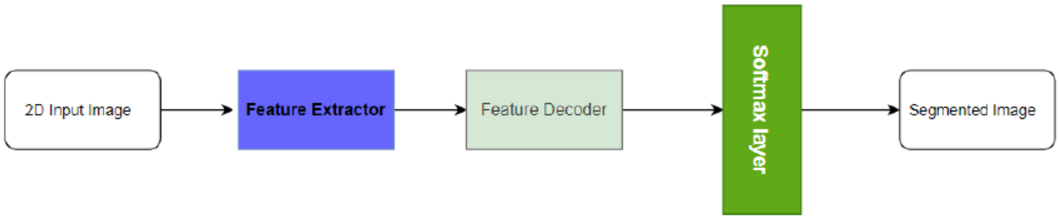

Figure 13: Softmax Layer in ConvNet. [10]

3.5 Object Detection Task in CNN.

Traditional machine learning-based object detection involves the use of two-class classifiers to scan an image. However, due to the constant aspect ratio of the object to be identified, only a specific type of object detection can be learned as a positive sample. In contrast, CNN-based object detection can detect parts of the object proposal with different aspects, enabling multiclass object detection through the region proposal approach. This approach employs CNN to perform multiclass classification for each discovered region. In the Region Proposal Network (RPN), an item is located by raster scanning the detection window on the acquired feature map. In this process, k-shaped detection windows are applied with their centers centered on the anchor-focused regions.

RPN takes the area defined by the anchor as input and outputs the detected coordinates on the input image, as well as the object's likeness score. The second fully connected network receives the region specified by the anchor, and when the RPN determines it to be an object, object recognition is performed.

In 2016, a new multiclass object detection approach called single-shot methodology was introduced. This technique involves providing the entire image to CNN without prior raster scanning in order to detect multiple objects. One notable method of this approach is called YOLO (You Only Look Once), which outputs an object rectangle and category for each local region in a 7 by 7 grid. The process begins with convolution and image pooling to create feature maps, and the output layer unit consists of the sum of the number of categories (20 categories) and the quantity of grids (7 by 7) added to the position, size, and reliability ((x, y, w, h, reliability) 2) of two object rectangles. YOLO does not require object region candidate detection like Faster R-CNN, allowing for real-time object detection. Please see figure for a YOLO-based multiclass object detection example.

Detecting and localizing objects of interest in images or videos is a computer vision task known as object detection. Convolutional Neural Networks (CNNs) have gained popularity as an effective approach for this task, as they have the ability to automatically extract important features from input images.

There are several methods for performing object detection using CNNs, including:

1. Region-based CNNs: This approach involves first generating region proposals (potential object locations) using an algorithm such as Selective Search, and then using a CNN to classify and refine these proposals.

2. Single-shot detectors (SSDs): SSDs are a type of CNN that can simultaneously predict the location and class of multiple objects within an image.

3. Faster R-CNN: Faster R-CNN is a CNN-based object detection method that uses a Region Proposal Network (RPN) to generate region proposals and a Fast R-CNN network to classify and refine these proposals.

4. YOLO (You Only Look Once): YOLO is a real-time object detection algorithm that uses a single CNN to predict the location and class of objects within an image.

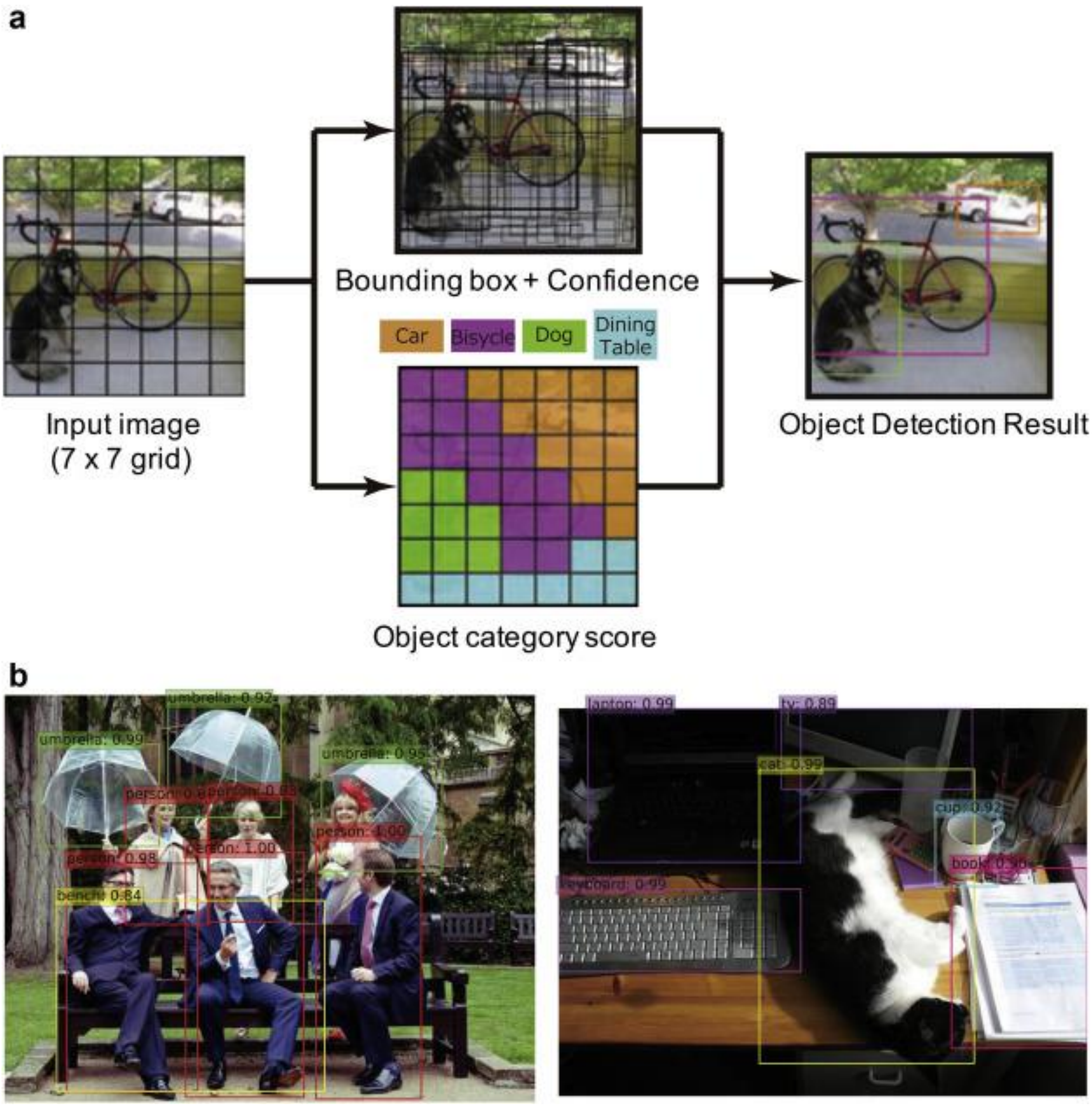

Figure 14: YOLO structure and examples of multiclass object detection. [11]

These methods typically involve training the CNN on a large dataset of annotated images, where the objects of interest have been manually labeled.

During training, the CNN learns to identify the relevant features and patterns within the input images that correspond to the objects of interest. Once the CNN has been trained, it can be used to automatically detect and localize objects within new, unseen images.

YOLO (You Only Look Once) is a popular real-time object detection algorithm that uses a single convolutional neural network (CNN) to predict the location and class of objects within an image. YOLO divides the input image into a grid of cells, and each cell is responsible for detecting objects that are centered within it.

YOLO is known for its speed and efficiency, as it can process images in real-time on a single GPU. However, it may have lower accuracy compared to other object detection algorithms such as Faster R-CNN or Mask R-CNN.

3.6 Application of CNN to Semantic Segmentation

Convolutional Neural Networks (CNNs) have been highly successful in the field of semantic segmentation, which is the task of assigning a label to each pixel in an image. CNNs have been used for a wide range of applications, including medical image analysis, autonomous driving, and natural language processing.

The basic architecture of a CNN for semantic segmentation involves a series of convolutional layers followed by upsampling layers, which are used to increase the spatial resolution of the feature map. The final layer of the network is a softmax layer, which produces a probability distribution over the labels for each pixel in the image.

One of the key advantages of using CNNs for semantic segmentation is their ability to learn features directly from the input image. Traditional image segmentation algorithms rely on hand-crafted features, which can be time-consuming and error-prone. In contrast, CNNs can learn features from the input data, which can improve the accuracy and generalization of the segmentation.

Another advantage of CNNs for semantic segmentation is their ability to handle complex input data, such as 3D medical images or RGB-D images. CNNs can be extended to handle multiple modalities, which can provide additional information to improve the accuracy of the segmentation.

Finally, CNNs can be trained end-to-end using backpropagation, which allows the network to learn the optimal weights for the task of semantic segmentation. This training process is computationally efficient and can be parallelized across multiple GPUs, which enables the training of large-scale models.

CNNs have emerged as a highly promising technology in the field of semantic segmentation and are expected to remain a major player in this area going forward. While the challenge of semantic segmentation has been extensively studied in computer vision for many years, deep learning-based methods have demonstrated significantly better performance than traditional machine learning approaches, as with many other tasks. A technique that enables end-to-end learning and can produce segmentation results using only CNNs is the fully convolutional network (FCN), which lacks a fully connected layer in its architecture. By repeatedly applying convolutional and pooling layers to the input image, the size of the resulting feature map is reduced. The structure of the FCN is illustrated below.

The feature map is enlarged 32 times by the final layer and then subjected to convolution processing to match the size of the original image, which is known as deconvolution. This layer produces a probability map for each class, resulting in an output size of (w, h, number of classes), where the probability of the class in each pixel is obtained. Since the feature map of the middle layer of CNN captures more detailed information than other layers, it is often utilized to achieve greater accuracy. However, detailed information is lost during the pooling process, resulting in coarse segmentation outcomes. To address this issue, FCN integrates the middle layer's feature map into the network's core. During the convolution process, the mid-feature maps are merged in the channel direction, resulting in segmentation results that are the same size as the original image.

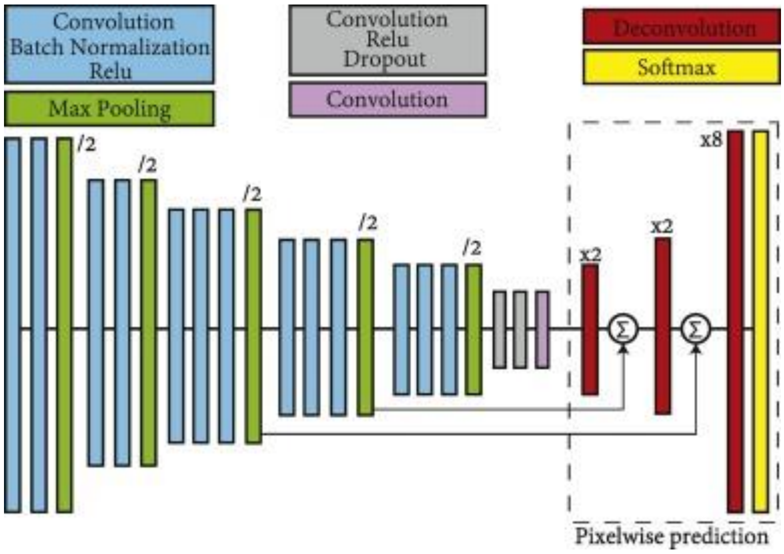

Figure 15: Fully Convolutional Network (FCN) Structure. [11]

PSPNet utilizes the Pyramid Pooling Module to capture information of different scales from the feature map obtained on the encoder side. This is achieved by pooling feature maps with different sizes, such as 1x1, 2x2, 3x3, and 3x6x6. This reduces the vertical and horizontal sizes of the original image to 1/8 on the encoder side. Each feature map is then subjected to the convolution process, and after the feature maps have been enlarged and linked to the same size, probability maps for each class are produced using convolution. In the "Scene parsing" category of the 2016 ILSVRC, PSPNet was the successful technique.

3.7 CNN for ADAS Application

Convolutional Neural Networks (CNNs) have become a popular technique for Advanced Driver Assistance Systems (ADAS), which aim to improve driving safety and comfort using system intelligence. While CNN-based systems have shown significant contributions to pedestrian detection, lane detection, and redundant object detection at moderate distances, ADAS primarily uses radar and sonar for long-range detection. The core elements of autonomous driving are perception, planning, and control. Perception involves recognizing and classifying objects based on their semantic labels, locating barriers, and recognizing road signs and markings. Localization refers to the ability of the autonomous vehicle to determine its location in the environment. Planning is the act of selecting actions that will help the vehicle achieve its objectives, usually involving getting the vehicle from one point to another while avoiding obstacles and maximizing its trajectory. Control is the ability of the vehicle to carry out the intended actions.

Object detection is a key application of CNNs in ADAS. The goal of object detection is to detect and classify objects in the driving environment, such as other vehicles, pedestrians, and traffic signs. CNNs can be used to process input from various sensors, such as cameras and LiDAR, and classify objects based on their shape, size, and motion.

Lane detection is another important ADAS application that uses CNNs. The goal of lane detection is to detect the boundaries of the road and determine the position of the vehicle within the lane. CNNs can be used to process input from a camera and detect the edges of the road, which can then be used to estimate the vehicle's position within the lane.

Pedestrian detection is another ADAS application that uses CNNs. The goal of pedestrian detection is to detect pedestrians in the driving environment and estimate their trajectory. CNNs can be used to process input from a camera and detect the presence of pedestrians based on their shape, size, and motion.

In addition to these applications, CNNs can also be used for other ADAS tasks, such as traffic sign recognition, collision detection, and autonomous driving. CNNs are well-suited for these tasks because they can learn complex patterns from the input data and make accurate predictions in real-time.

Overall, CNNs are a powerful tool for ADAS applications and are likely to play an increasingly important role in enhancing driving safety and comfort in the future.

Chapter – 4
EXPERIMENTS AND RESULT ANALYSIS

There are several publicly available datasets for road image segmentation that can be used to train and evaluate segmentation algorithms. Some popular datasets include:

1. CamVid: CamVid is a dataset of urban street scenes captured from a moving vehicle. It contains 701 images with corresponding high-resolution ground truth annotations, and is commonly used for evaluating road and object segmentation algorithms.

2. Cityscapes: Cityscapes is a large-scale dataset of street scenes from 50 cities around the world. It contains over 5,000 high-resolution images with pixel-level annotations for 19 different object classes, including roads, vehicles, and pedestrians.

3. KITTI: The KITTI dataset is a popular benchmark for evaluating autonomous driving algorithms. It contains a large number of high-resolution images and point cloud data captured from a moving vehicle, along with ground truth annotations for road segmentation and other tasks.

4. Mapillary Vistas: Mapillary Vistas is a large-scale dataset of street-level imagery with pixel-level annotations for 37 object classes, including roads, buildings, and traffic signs. It contains over 25,000 images captured from around the world, and is useful for evaluating segmentation algorithms in diverse environments.

5. ApolloScape: ApolloScape is a dataset of autonomous driving scenes captured by Baidu's Apollo platform. It contains over 140,000 images with annotations for road segmentation, object detection, and other tasks, and is commonly used for evaluating autonomous driving algorithms.

These datasets provide a wide range of images and annotations that can be used to train and evaluate road image segmentation algorithms.

In this project I train a Fully Convolutional Network (FCN) to classify each pixel of an image as road or no road.

I use KITTI Dataset.

The dataset consists of 289 training and 290 test images. It contains three different categories of road scenes:

- uu - urban unmarked (98/100)
- um - urban marked (95/96)
- umm - urban multiple marked lanes (96/94)
- urban - combination of the three above

Ground truth has been generated by manual annotation of the images and is available for two different road terrain types:

- road - the road area, i.e., the composition of all lanes, and
- lane - the ego-lane, i.e., the lane the vehicle is currently driving on (only available for category "um").

Ground truth is provided for training images only.

In road image segmentation, there are typically three types of images that are used for evaluation: the original image, the ground truth segmentation, and the predicted segmentation.

The ground truth segmentation is an image that contains the correct segmentation of the original image. It is typically a binary image where each pixel is assigned a label indicating whether it belongs to the road or not. The ground truth segmentation is manually annotated by human experts, or sometimes generated automatically using other sensors such as LiDAR or GPS.

The predicted segmentation is the output of the segmentation algorithm being evaluated. It is also a binary image where each pixel is assigned a label indicating whether it belongs to the road or not. The predicted segmentation is generated automatically by the segmentation algorithm using the original image as input.

The original image, ground truth segmentation, and predicted segmentation are often displayed side-by-side to facilitate visual comparison and analysis.

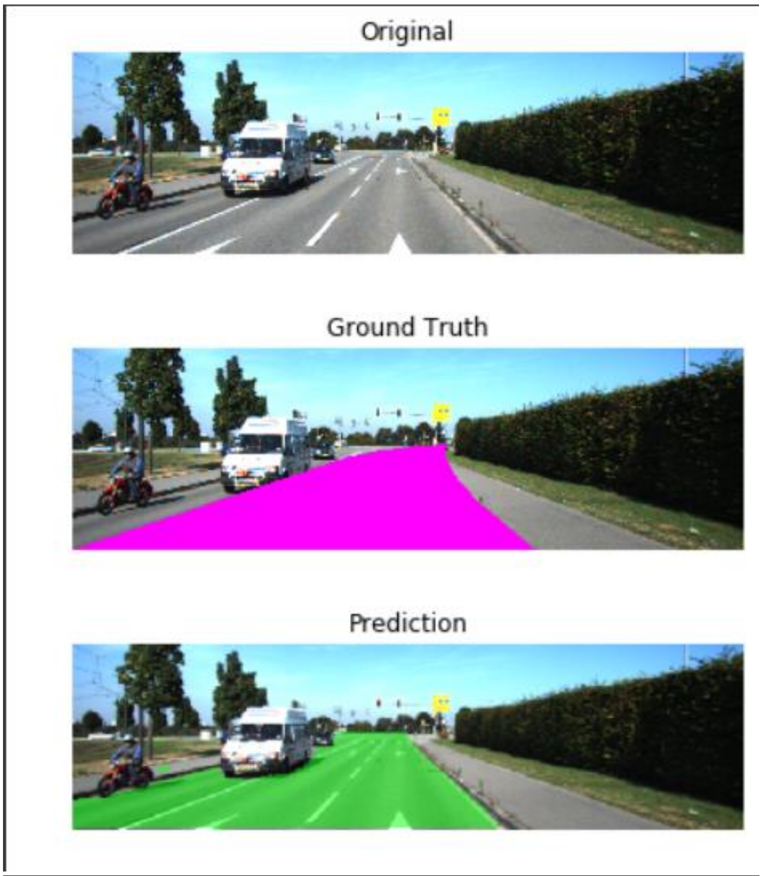

Figure 16: Result

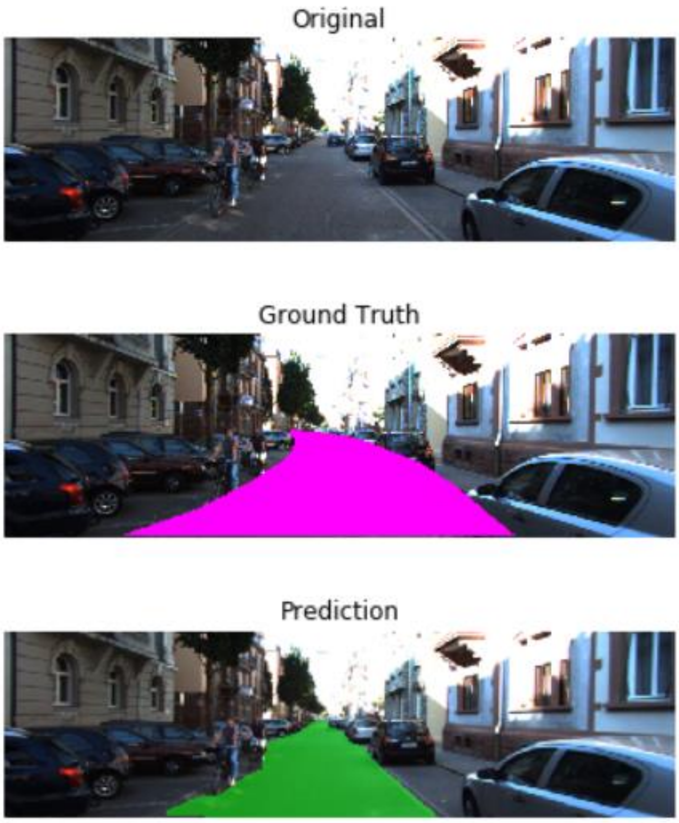

Figure 17: Result

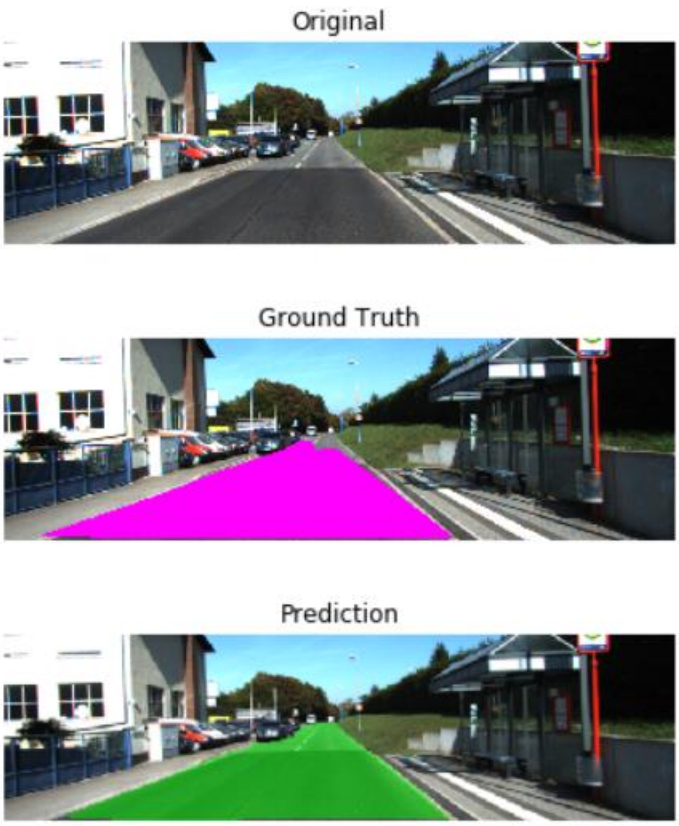

Figure 18: Result

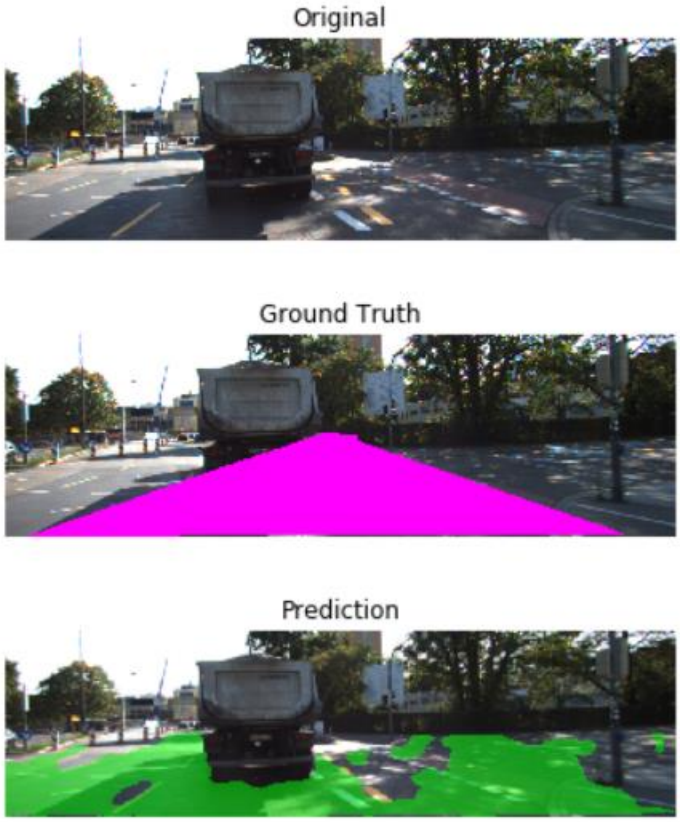

Figure 19: Result

In road image segmentation using deep learning, the Softmax function is commonly used as the activation function in the last layer of the neural network. The Softmax function is used to produce a probability map of the input image, where each pixel in the output image corresponds to the probability of that pixel belonging to each of the possible classes. In road image segmentation, the two possible classes are typically the road and the background.

The original image is the input image that is being segmented. It is usually a color or grayscale image that represents a scene from a camera mounted on a vehicle or a drone.

To visualize the road image segmentation process, the original image, Softmax image, and result image are often displayed side-by-side. This allows for easy visual comparison and analysis of the segmentation process.

Figure 20: original image.

The Softmax image is the output of the neural network's final layer, where the Softmax function is applied to the neural network's activation map. This produces a probability map of the input image, where each pixel in the output image corresponds to the probability of that pixel belonging to each of the possible classes. In road image segmentation, the two possible classes are typically the road and the background.

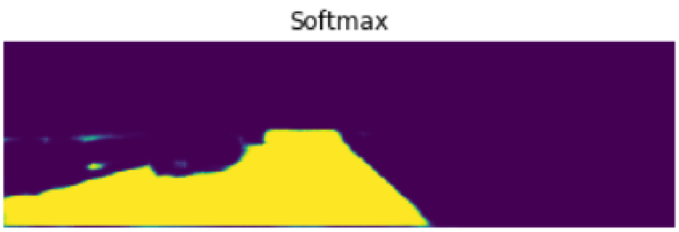

Figure 21: Result

The result image is the final output of the road image segmentation algorithm. It is a binary image where each pixel is assigned a label indicating whether it belongs to the road or not. This binary image is produced by thresholding the Softmax image, where pixels with high probabilities of belonging to the road class are labeled as road pixels and pixels with low probabilities are labeled as background pixels.

Figure 22: Result

Deep Learning for Road and Building Segmentation in satellite imagery.

Road image segmentation can also be applied to satellite images, which are images captured by satellites orbiting the Earth. Satellite images are useful for mapping large areas of the Earth's surface and monitoring changes in land use and land cover over time.

The methodology for road image segmentation in satellite images is similar to that for road image segmentation in other types of images, but there are some additional challenges and considerations.

One challenge is the resolution of the satellite image. Satellite images typically have lower spatial resolution than images captured by ground-based cameras. This can make it more difficult to accurately segment roads and distinguish them from other features in the image.
Another challenge is the variability in lighting and weather conditions. Satellite images can be affected by cloud cover, shadows, and atmospheric interference, which can affect the quality and accuracy of the segmentation results.

To address these challenges, techniques such as image enhancement, feature extraction, and data fusion may be used to improve the quality and accuracy of the segmentation results.

Road image segmentation for satellite images can be a valuable tool for mapping and monitoring road networks over large areas, and can provide important information for urban planning, transportation management, and

disaster response.
Deep learning is used during semantic segmentation to categories each pixel of an image into several classes. This makes it easier to locate areas in an image where specific things are present.

I utilized the Massachusetts Roads Dataset for this task. Along with the target masks, this dataset also includes aerial photos. To download every image mentioned on this website, use download images.py. Using a torrent client to get the material might be a wise move if you have unstable internet connections. The dataset is available here, and you can obtain the photos from academic torrent sites.

The steps of pre-processing involved:

- pictures where the map was missing more than 25% were deleted.
- extracted 256x256 pictures from the original images. Consequently, more than 22,000 photos are included overall.
- The mask was binarized so that each pixel's value always falls between 0 and 1.

I utilized an Unet, a fully convolutional network with three cross-connections, to fix this issue. Dice loss and the Adam optimizer with a learning rate of 0.00001 were applied (because of the unbalanced nature of the dataset.) Before Earlstopper intervened and stopped the training process, the model trained for 61 epochs. The achieved validation dice loss was 0.7548.

Video segmentation, or the division of video frames into several segments or objects, is crucial in a range of real-world applications, including the development of virtual backgrounds for video conferences, autonomous driving scene interpretation, and visual effect help in movies. In this essay, we will examine what video object segmentation is and how it is applied in order to properly understand this idea.

Road image segmentation is a computer vision task that involves separating the pixels of an image that correspond to roads from those that correspond to other objects in the image. The goal of road image segmentation is to accurately identify the location and boundaries of roads in an image, which can be used for a variety of applications such as autonomous driving, road maintenance, and urban planning.

Road image segmentation for video involves segmenting each frame of a video in real-time or near real-time. This is important for applications such as autonomous driving and traffic monitoring, where real-time segmentation of road and traffic-related features is critical for making timely decisions.

The methodology for road image segmentation in video is like that for road image segmentation in single images, but there are some additional challenges and considerations.
Road image segmentation can also be applied to videos, which are a sequence of frames captured by a camera over time. Video segmentation can be useful for applications such as autonomous driving, traffic monitoring, and surveillance.

The methodology for road image segmentation in videos is similar to that for road image segmentation in static images, but there are some additional considerations.

One consideration is the temporal coherence of the segmentation results. Since videos are a sequence of frames, the segmentation results for each frame should be consistent with the results for the adjacent frames. This can be achieved using techniques such as optical flow, which estimates the motion of objects in the video and can be used to propagate segmentation results from one frame to the next. One challenge is the computational complexity of processing multiple frames of a video in real-time. To overcome this challenge, techniques such as parallel processing, GPU acceleration, and optimized algorithms may be used to speed up the segmentation process. Another challenge is the temporal consistency of the segmentation results. Since adjacent frames in a video are usually correlated, the segmentation results for one frame should be consistent with the segmentation results for the neighboring frames. This can be achieved through techniques such as temporal filtering, motion estimation, and tracking.

To address these challenges, specialized techniques such as real-time segmentation algorithms, motion estimation, and optical flow analysis may be used. Additionally, deep learning techniques such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) can be used to exploit the temporal information in video data and produce more accurate segmentation results.

To evaluate the performance of a road image segmentation algorithm for video, metrics such as frame rate, segmentation accuracy, and temporal consistency can be used. Real-time video segmentation requires a high frame rate, typically 30 frames per second or more, and a high segmentation accuracy, with minimal false positives and false negatives. Temporal consistency can be evaluated by measuring the consistency of the segmentation results between adjacent frames, and by analyzing the smoothness of the segmentation boundaries over time.

Overall, road image segmentation for video is an important task for a variety of applications, and requires specialized techniques and evaluation metrics to achieve accurate and efficient results.

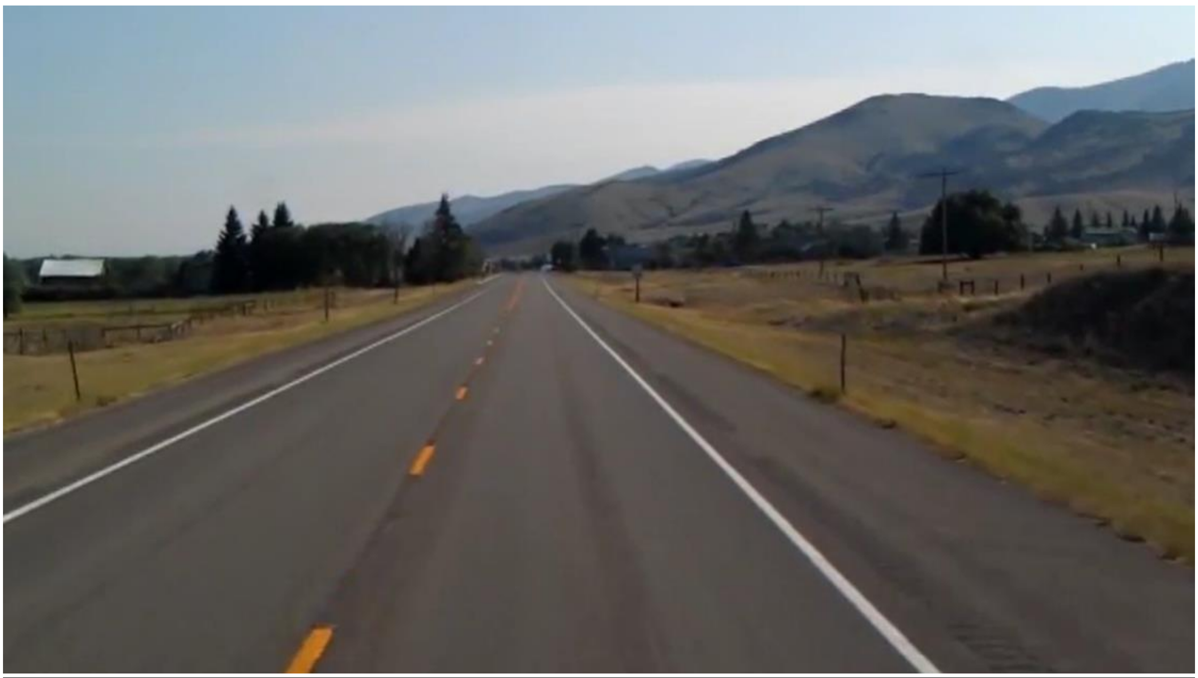

Figure 23: Training video of segmentation.

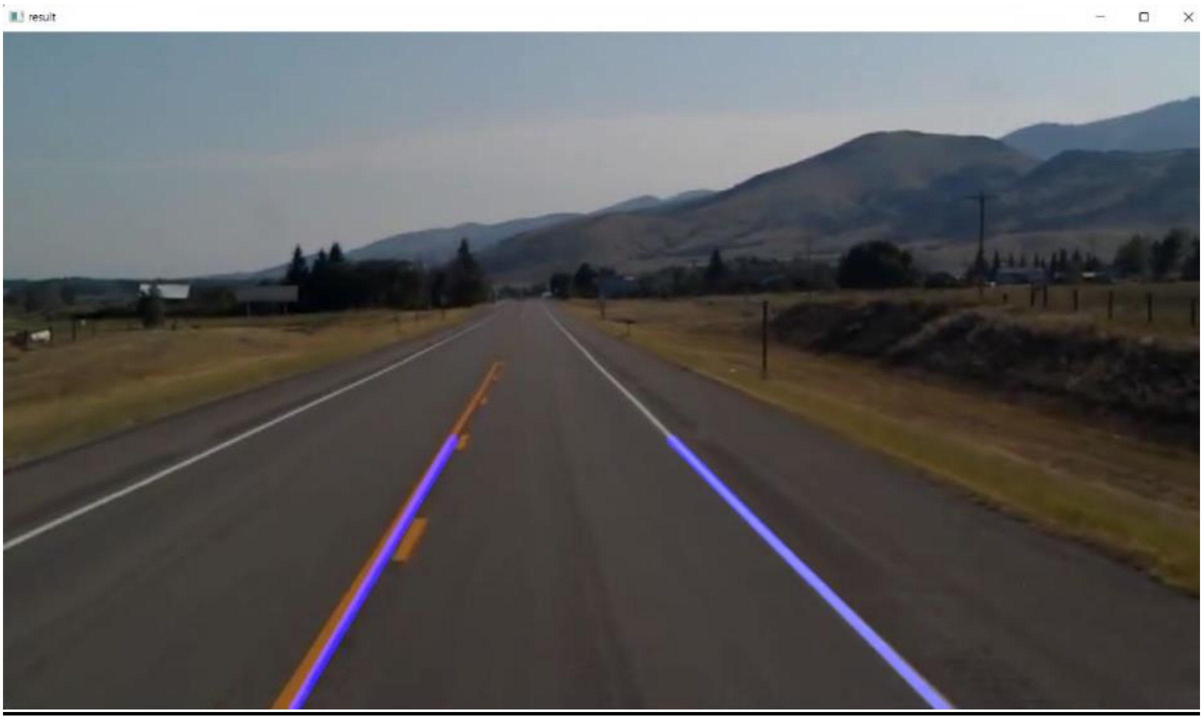

Figure 24: Output for video segmentation.

Chapter – 5
CONCLUSIONS

5.1 Conclusion

In conclusion, road image segmentation is a crucial task in the development of autonomous cars. Accurate identification of the location and boundaries of roads in an image is essential for safe and efficient autonomous driving. Various techniques and algorithms can be used for road image segmentation, with deep learning-based approaches such as CNNs being particularly effective.

However, there are still challenges and limitations that need to be addressed, such as the need for large and diverse datasets, real-time processing requirements, and the variability of environmental conditions. Despite these challenges, road image segmentation for autonomous cars is a rapidly evolving field, and advancements in technology and algorithms are constantly improving the accuracy and efficiency of the segmentation process. With continued research and development, road image segmentation will play a key role in the future of autonomous driving.

Road image segmentation for autonomous driving still faces some challenges, such as variability in lighting and weather conditions, occlusions, and complex road geometries. Addressing these challenges requires the development of more robust and adaptable segmentation algorithms, as well as the integration of multiple sensors and data sources to provide a more comprehensive view of the driving environment.

Overall, road image segmentation is an exciting and rapidly evolving field, with significant potential for improving the safety and efficiency of transportation systems. As the technology continues to advance, we can expect to see more sophisticated and capable autonomous driving systems that are capable of navigating complex and dynamic environments with greater precision and reliability. Road image segmentation is a critical task for autonomous cars, as it enables them to accurately identify the location and boundaries of roads in their environment. This information is essential for safe and effective navigation, and can also be used for applications such as traffic monitoring and urban planning.

Convolutional Neural Networks (CNNs) have proven to be effective for road image segmentation, as they can learn to extract relevant features from images and accurately classify each pixel as either belonging to a road or not. Additionally, techniques such as data augmentation and transfer learning can be used to improve the performance of CNN-based segmentation models.

There are also various challenges to consider when performing road image segmentation for autonomous cars, such as the need for real-time processing and the variability in lighting and weather conditions. However, with the development of advanced sensors and algorithms, road image segmentation for autonomous cars is becoming increasingly accurate and reliable, bringing us one step closer to the widespread adoption of autonomous vehicles on our roads.

Autonomous vehicles are already available, and prominent automakers are stepping into this market. Vehicle manufacturers are embracing this technology. To successfully navigate and comprehend the area they are in, these vehicles rely on on-board sensors and technical equipment. For successful route planning, emergency movements, and route calculations, precise sensor data is essential. The main success criteria and barriers for autonomous vehicles have been covered in this essay. The study also highlights why safety is important and a crucial part of the security of autonomous cars. Human involvement in the control process has diminished as vehicle automation has increased. The system performs some tasks, and the driver's job is to manage the traffic situation and react appropriately (and as quickly as possible) in case of an emergency. The degree of road safety may increase if the human aspect is reduced or eliminated from the vehicle control process. Knowing how the system functions and what it is capable of can help to achieve the maximum level of safety. Designing systems that provide the maximum level of knowledge therefore appears vital.

Therefore, the three primary technologies—IoT sensor, IoT connectivity, software & algorithm—that were explained in the introduction section can be used to create a self-driving car in real life. Semantic segmentation, a topic under the heading of "Software & Algorithm," is the one I've chosen for my research. In which I developed a Convolution Neural Network using a cityscape dataset as a 2D input image. This idea can be applied to the analysis of road scenes in self-driving cars, which would aid in visual perception, or the ability to interpret the environment using camera sensors.

This paper provides the most recent deep learning-based image identification technology and discusses how deep learning is used in picture recognition applications. Finding a suitable mapping function from a huge amount of data and teacher labels is an issue in image recognition technology employing deep learning. Furthermore, multitask learning can be used to tackle multiple issues at once. For the perception of autonomous vehicles, semantic segmentation is a crucial procedure. It is crucial to comprehending how the ego automobile functions in a road setting. The model for achieving driverless autos was presented in this paper. Although research is ongoing, we anticipate using this computerized model as software in actual automobiles in the not-too-distant future. Millions of data scientists and artificial intelligence researchers are attempting to translate this software model into practical applications so that automobiles can become driverless, there will be fewer rule violations, and hopefully fewer accidents on the road. As deep learning techniques have advanced over the past ten years, more and more research is focusing on how to take advantage of deep learning to increase perception and decision-making, among other autonomous process characteristics. In this study, we developed a hybrid model that is constructed and trained using the design principles of two deep learning models. To improve the training quality, we applied data augmentation approaches. The developed architecture was intended to be as realistic as real-time concerns would allow.

5.2 Future Scope

The field of road image segmentation for autonomous cars is rapidly evolving, and there are several exciting areas of future research and development.

One area of future scope is the integration of multiple sensors for more accurate segmentation results. While cameras and LiDAR are currently the most used sensors for road image segmentation, other sensors such as radar and sonar could potentially be used to improve segmentation accuracy and robustness.

Another area of future scope is the development of more advanced segmentation algorithms that can handle more complex driving scenarios. For example, algorithms that can segment roads in crowded urban environments, where there are numerous other objects such as pedestrians, bicycles, and vehicles.

Furthermore, there is a need to develop more efficient and lightweight algorithms that can be deployed on low-power embedded systems such as those used in autonomous vehicles.

The availability of larger and more diverse annotated datasets will continue to be important for the training and validation of road image segmentation algorithms.

Overall, road image segmentation for autonomous cars is an active area of research, and there are numerous exciting opportunities for future developments that could help advance the field and enable the deployment of safe and reliable autonomous driving systems.

The field of road image segmentation for autonomous cars is rapidly evolving, and there are several areas of future research and development that could lead to significant improvements in the performance and capabilities of self-driving cars. Here are a few potential areas of future scope:

1. multi-sensor fusion: Currently, most autonomous cars rely on a combination of cameras and LiDAR for perception tasks such as road image segmentation. However, there is potential for integrating other sensors such as radar, sonar, and GPS to provide a more comprehensive and robust view of the environment.

2. Real-time optimization: Although CNNs are effective for road image segmentation, they can be computationally expensive, making it difficult to achieve real-time performance in embedded systems. There is a need for efficient algorithms and hardware architectures that can achieve high accuracy while also optimizing for real-time processing.

3. Unstructured environments: While road image segmentation is useful for autonomous driving on structured roads, it becomes challenging in unstructured environments such as off-road terrain, construction sites, and rural areas. Developing algorithms that can accurately segment roads in these environments is an area of future research.

4. Generalization: Current Road image segmentation algorithms are typically trained and tested on specific datasets and environments, which limits their ability to generalize to new and unseen scenarios. Developing algorithms that can adapt to new environments and datasets is an area of future research that could significantly improve the performance of autonomous driving systems.

More research can be done in the future to improve the performance of the suggested hybrid design and fix its current flaws. The following are the key research axes that can be tested. Despite being viable for real-time applications due to current computer capabilities, the number of parameters in our model can be decreased by reducing the model complexity and by paying attention to the design concepts of certain other cutting-edge models with fewer parameters.

The development of end-to-end learning and deep reinforcement learning technologies for "judgement" and "control" of autonomous cars is expected to meet high expectations in the future. Future prospects include "recognition" for input photos as well. It is also desirable to move beyond visual explanation to verbal explanation through integration with natural language processing. Citing judgement grounds for output of deep learning and deep reinforcement learning is a big issue in practical application. This study will contribute to a decrease in traffic accidents brought on by human mistake, which results in needless fatalities and lives that could be spared by safer driving. being immune to issues like driver weariness, mood, or disease makes them safer than human-driven vehicles.

5.3 Applications Contributions

Road image segmentation has a wide range of applications and contributions, including:

1. Autonomous driving: Road image segmentation is a critical component of autonomous driving systems, enabling vehicles to understand and navigate their environment.

2. Road maintenance: Road image segmentation can be used to identify areas of damage and deterioration on road surfaces, allowing for targeted maintenance and repair.

3. Urban planning: Road image segmentation can provide valuable information about traffic patterns, pedestrian movement, and urban infrastructure, helping

to inform the design and planning of cities.

4. Traffic analysis: Road image segmentation can be used to analyze and understand traffic patterns, including the volume of traffic on different roads, the speed of vehicles, and the behavior of drivers. This information can be used to optimize traffic flow, reduce congestion, and improve safety.

5. Emergency response: Road image segmentation can be used to quickly identify road closures and obstructions in emergency situations, enabling first responders to navigate to their destination quickly and safely.

6. Environmental monitoring: Road image segmentation can be used to monitor the impact of urbanization on natural habitats, such as by tracking changes in road networks and traffic patterns.

7. Augmented reality: Road image segmentation can be used in augmented reality applications to overlay virtual objects onto real-world road scenes, enhancing the user experience.

Overall, road image segmentation has numerous applications and contributions across a variety of fields, making it a critical area of research and development for advancing the safety and efficiency of our transportation systems.

REFERENCES

[1] Margarita Martínez-Díaz, Francesc Soriguera, Autonomous vehicles: theoretical and practical challenges, Transportation Research Procedia, Volume 33, 2018, Pages 275-282, ISSN 2352-1465, https://doi.org/10.1016/j.trpro.2018.10.103. (https://www.sciencedirect.com/science/article/pii/S2352146518302606)

[2] Diakaki, Christina & Papageorgiou, Markos & Papamichail, Ioannis & Nikolos, Ioannis. (2015). Overview and analysis of Vehicle Automation and Communication Systems from a motorway traffic management perspective. Transportation Research Part A Policy and Practice. 75. 147-165. 10.1016/j.tra.2015.03.015.

[3] A. Gupta, A. S. Khwaja, A. Anpalagan and L. Guan, "Safe Driving of Autonomous Vehicles through State Representation Learning," 2021 International Wireless Communications and Mobile Computing (IWCMC), 2021, pp. 260-265, doi: 10.1109/IWCMC51323.2021.9498960.

[4] Changxi You, Jianbo Lu, Dimitar Filev, Panagiotis Tsiotras, Advanced planning for autonomous vehicles using reinforcement learning and deep inverse reinforcement learning,
Robotics and Autonomous Systems, Volume 114, 2019, Pages 1-18, ISSN 0921-8890,
https://doi.org/10.1016/j.robot.2019.01.003.
(https://www.sciencedirect.com/science/article/pii/S0921889018302021)

[5] Meixin Zhu, Xuesong Wang, Yinhai Wang, Human-like autonomous car-following model with deep reinforcement learning, Transportation Research Part C: Emerging Technologies, Volume 97, 2018, Pages 348-368, ISSN 0968-090X, https://doi.org/10.1016/j.trc.2018.10.024.
(https://www.sciencedirect.com/science/article/pii/S0968090X1830055X)

[6] Grigorescu, S, Trasnea, B, Cocias, T, Macesanu, G. A survey of deep learning techniques for autonomous driving. *J Field Robotics*. 2020; 37: 362–386. https://doi.org/10.1002/rob.21918

[7] A. Rasouli and J. K. Tsotsos, "Autonomous Vehicles That Interact With Pedestrians: A Survey of Theory and Practice," in IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 3, pp. 900-918, March 2020, doi: 10.1109/TITS.2019.2901817.

[8] Stilgoe, Jack. "Machine learning, social learning and the governance of self-driving cars." *Social studies of science* 48.1 (2018): 25-56.

[9] Khan, Firoz, et al. "Autonomous vehicles: A study of implementation and security." *International Journal of Electrical & Computer Engineering (2088-8708)* 11.4 (2021).

[10] Choudhary, Nilesh K., Nikhil C. Gavali, and Jashwant Kumar M. Bind. "SELF DRIVING CAR USING CONVOLUTIONAL NEURAL NETWORK." *IJRAR-International Journal of Research and Analytical Reviews (IJRAR)* 7.1 (2020): 320-323.

[11] Fujiyoshi, Hironobu, Tsubasa Hirakawa, and Takayoshi Yamashita. "Deep learning-based image recognition for autonomous driving." *IATSS research* 43.4 (2019): 244-252.

[12] Sellat, Qusay, et al. "Intelligent semantic segmentation for self-driving vehicles using deep learning." *Computational Intelligence and Neuroscience* 2022 (2022).

[13] Brust, Clemens-Alexander, et al. "Efficient convolutional patch networks for scene understanding." *CVPR Scene Understanding Workshop*. 2015.

[14] A. G. Howard, M. Zhu, B. Chen et al., "Mobilenets: efficient convolutional neural networks for mobile vision applications," 2017, https://arxiv.org/abs/1704.04861.

[15] L. Sifre and S. Mallat, "Rigid-motion scattering for image classification," Ecole Polytechnique, Palaiseau, France, 2018, Ph. D. thesis.

[16] M. Tan and Q. V. Le, "Efficientnet: rethinking model scaling for convolutional neural networks," 2014, https://arxiv.org/abs/1905.11946.

[17] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, Boston, MA, USA, June 2015.

[18] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: a deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.

[19] V. Badrinarayanan, A. Handa, and R. Cipolla, “Segnet: a deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling,” 2015, https://arxiv.org/abs/1505.07293.

[20] O. Ronneberger, P. Fischer, and T. Brox, “U-net: convolutional networks for biomedical image segmentation,” in *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 234–241, Springer, Munich, Germany, October 2015.

[21] J. D. T. D. a. J. M. Ross Girshick, " Rich feature hierarchies for accurate object detection and semantic segmentation," CVPR, 2014.

[22] -T. L. E. H. a. Q. V. L. Qizhe Xie, "Self-training with noisy student improves imagenet," M. 2019.

[23] F. L. L. D. a. B. G. Qing Lian, ". Constructing Self-Motivated Pyramid Curriculums for," in IEEE International Conference on Computer Vision (ICCV), 2019.

[24] S. Y. a. G. Peng, "Attention to refine through multi scales for semantic segmentation," in Pacific Rim Conference on Multimedia, 2018.

[25] A. V. a. A. Z. Karen Simonyan, Deep fisher networks for large-scale image, NIPS, 2013.

[26] X. Z. S. R. a. J. S. Kaiming He, " Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016.

[27] Suresh Babu Rajasekaran, "A Self-Adaptive Chimp-Driven Modified Deep Learning Framework for Autonomous Vehicles to Obtain Autonomous Object Classification", *Electric Power Components and Systems*, pp.1, 2023.

[28] Qingyu Wang, Dihua Wu, Wei Liu, Mingzhao Lou, Huanyu Jiang, Yibin Ying, Mingchuan Zhou, "PlantStereo: A High Quality Stereo Matching Dataset for Plant Reconstruction", *Agriculture*, vol.13, no.2, pp.330, 2023.

[29] Wentian Xin, Ruyi Liu, Yi Liu, Yu Chen, Wenxin Yu, Qiguang Miao, "Transformer for Skeleton-based action recognition: A review of recent advances", *Neurocomputing*, vol.537, pp.164, 2023.

[30] Nikita Singhal, Lalji Prasad, "Comparative Analysis of Deep Learning based Vehicle Detection Approaches", *International Journal of Next-Generation Computing*, 2023.

[31] Zhouyan Qiu, Joaquín Martínez-Sánchez, Pedro Arias-Sánchez, Rabia Rashdi, "External multi-modal imaging sensor calibration for sensor fusion: A review", *Information Fusion*, pp.101806, 2023.

[32] Martí Caro, Hamid Tabani, Jaume Abella, Francesc Moll, Enric Morancho, Ramon Canal, Josep Altet, Antonio Calomarde, Francisco J. Cazorla, Antonio Rubio, Pau Fontova, Jordi Fornt, "An automotive case study on the limits of approximation for object detection", *Journal of Systems Architecture*, pp.102872, 2023.

[33] Ihssane Bouasria, Walid Jebrane, Nabil El Akchioui, "The Role of Ground Truth Annotation in Semantic Image Segmentation Performance for Autonomous Driving", *Proceedings of the 6th International Conference on Big Data and Internet of Things*, vol.625, pp.263, 2023.

[34] Jingjing Tao, Zhe Chen, Zhongchang Sun, Huadong Guo, Bo Leng, Zhengbo Yu, Yanli Wang, Ziqiong He, Xiangqi Lei, Jinpei Yang, "Seg-Road: A Segmentation Network for Road Extraction Based on Transformer and CNN with Connectivity Structures", *Remote Sensing*, vol.15, no.6, pp.1602, 2023.

[35] Daniel Steininger, Andreas Kriegler, Wolfgang Pointner, Verena Widhalm, Julia Simon, Oliver Zendel, "Towards Scene Understanding for Autonomous

Operations on Airport Aprons", *Computer Vision – ACCV 2022 Workshops*, vol.13848, pp.153, 2023.

[36] Jincao Zhou, Xin Bai, Wenjie Hu, "Recognition and Prediction of Pedestrian Hazardous Crossing Intentions in Visual Field Obstruction Areas Based on IPVO-LSTM", *Applied Sciences*, vol.13, no.5, pp.2999, 2023.

[37] Francisca Rosique, Pedro J. Navarro, Leanne Miller, Eduardo Salas, "Autonomous Vehicle Dataset with Real Multi-Driver Scenes and Biometric Data", *Sensors*, vol.23, no.4, pp.2009, 2023.

[38] Lei Chen, Jiazhen Chang, Jinli Xu, Zuowei Yang, "Automatic Measurement of Inclination Angle of Utility Poles Using 2D Image and 3D Point Cloud", *Applied Sciences*, vol.13, no.3, pp.1688, 2023.

[39] Akhil M. Kurup, Jeremy P. Bos, "Winter adverse driving dataset for autonomy in inclement winter weather", *Optical Engineering*, vol.62, no.03, 2023.

[40] Rolf Ernst, Dominik Stöhrmann, Alex Bendrick, Adam Kostrzewa, "Application-centric Network Management - Addressing Safety and Real-time in V2X Applications", *ACM Transactions on Embedded Computing Systems*, vol.22, no.2, pp.1, 2023.

[41] Yuxiao Zhang, Alexander Carballo, Hanting Yang, Kazuya Takeda, "Perception and sensing for autonomous vehicles under adverse weather conditions: A survey", *ISPRS Journal of Photogrammetry and Remote Sensing*, vol.196, pp.146, 2023.

[42] Junyi Gu, Mauro Bellone, Raivo Sell, Artjom Lind, "Object Segmentation for Autonomous Driving Using iseAuto Data", *Electronics*, vol.11, no.7, pp.1119, 2022.

[43] Junbo Yin, Jin Fang, Dingfu Zhou, Liangjun Zhang, Cheng-Zhong Xu,

Jianbing Shen, Wenguan Wang, "Semi-supervised 3D Object Detection with Proficient Teachers", *Computer Vision – ECCV 2022*, vol.13698, pp.727, 2022.

[44] Siyuan Huang, Limin Liu, Xiongjun Fu, Jian Dong, Fuyu Huang, Ping Lang, "Overview of LiDAR point cloud target detection methods based on deep learning", *Sensor Review*, vol.42, no.5, pp.485, 2022.

[45] Filip Rezabek, Marcin Bosk, Thomas Paul, Kilian Holzinger, Sebastian Gallenmüller, Angela Gonzalez, Abdoul Kane, Francesc Fons, Zhang Haigang, Georg Carle, Jörg Ott, "EnGINE: Flexible Research Infrastructure for Reliable and Scalable Time Sensitive Networks", *Journal of Network and Systems Management*, vol.30, no.4, 2022.

[46] Zhe An, Yang Liu, "Tracking registration of optical see-through augmented reality based on the Riemannian manifold constraint", *Optics Express*, vol.30, no.26, pp.46418, 2022.

[47] Xavier Dauptain, Aboubakar Koné, Damien Grolleau, Veronique Cerezo, Manuela Gennesseaux, Minh-Tan Do, "Conception of a High-Level Perception and Localization System for Autonomous Driving", *Sensors*, vol.22, no.24, pp.9661, 2022.

[48] Yilin Liu, Liqiang Lin, Yue Hu, Ke Xie, Chi-Wing Fu, Hao Zhang, Hui Huang, "Learning Reconstructability for Drone Aerial Path Planning", *ACM Transactions on Graphics*, vol.41, no.6, pp.1, 2022.

[49] Trong-Lanh R. Nguyen, Thierry Chateau, Guillaume Magniez, "VampNet: Unsupervised Vampirizing of Convolutional Networks", *Advances in Visual Computing*, vol.13598, pp.249, 2022.

[50] Jiyao An, Wei Liu, Qingqin Liu, Liang Guo, Ping Ren, Tao Li, "DGInet: Dynamic graph and interaction-aware convolutional network for vehicle trajectory prediction", *Neural Networks*, vol.151, pp.336, 2022.

[51] Saurabh Jain, Adarsh Kumar, Keshav Kaushik, Rajalakshmi Krishnamurthi, "Autonomous driving systems and experiences: A comprehensive survey", *Autonomous and Connected Heavy Vehicle Technology*, pp.65, 2022.

[52] Weiqi Wang, Xiong You, Lingyu Chen, Jiangpeng Tian, Fen Tang, Lantian Zhang, "A Scalable and Accurate De-Snowing Algorithm for LiDAR Point Clouds in Winter", *Remote Sensing*, vol.14, no.6, pp.1468, 2022.

APPENDICES

Few snippets of important section of codebase are as follows:

```
import numpy as np
import matplotlib.pyplot as plt
import cv2
import tensorflow as tf
import scipy.misc
from moviepy.editor import VideoFileClip

%matplotlib inline

def show_image(img, title):
    plt.imshow(img)
```

```
    plt.axis('off')
    plt.title(title)
    plt.show()


img = cv2.imread('/content/drive/MyDrive/data_road/testing/image_2/umm_0
00001.png')
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

label = cv2.imread('/content/drive/MyDrive/data_road/training/gt_image_2/u
mm_road_000041.png')
label = cv2.cvtColor(label, cv2.COLOR_BGR2RGB)
gray = cv2.cvtColor(label, cv2.COLOR_RGB2GRAY)
ret, mask = cv2.threshold(gray, 76, 255, cv2.THRESH_BINARY)
mask_inv = cv2.bitwise_not(mask)
img1_bg = cv2.bitwise_and(img, img, mask = mask_inv)
# label = only_road(label)
img2_bg = cv2.bitwise_and(label, label, mask = mask)

img_overlay = cv2.add(img1_bg, img2_bg)

prediction = cv2.imread('/content/drive/MyDrive/data_road/testing/image_2/u
mm_000041.png')
prediction = cv2.cvtColor(prediction, cv2.COLOR_BGR2RGB)

show_image(img, 'Original')
show_image(img_overlay, 'Ground Truth')
show_image(prediction, 'Prediction')
```

```python
plt.figure(figsize=(20,16))
x, y = 5,4
for i in range(y):
    for j in range(x):
        plt.subplot(y*2, x, i*2*x+j+1)
        pos = i*120 + j*10
        plt.imshow(images[pos])
        plt.title('Sat img #{}'.format(pos))
        plt.axis('off')
        plt.subplot(y*2, x, (i*2+1)*x+j+1)

        #We display the associated mask we just generated above with the training image
        plt.imshow(masks[pos])
        plt.title('Mask #{}'.format(pos))
        plt.axis('off')

plt.show()
```

```python
img = cv2.imread('/content/drive/MyDrive/data_road/testing/image_2/uu_000093.png')
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)

label = cv2.imread('/content/drive/MyDrive/data_road/training/gt_image_2/uu_road_000093.png')
label = cv2.cvtColor(label, cv2.COLOR_BGR2RGB)
gray = cv2.cvtColor(label, cv2.COLOR_RGB2GRAY)
ret, mask = cv2.threshold(gray, 76, 255, cv2.THRESH_BINARY)
mask_inv = cv2.bitwise_not(mask)
img1_bg = cv2.bitwise_and(img, img, mask = mask_inv)
# label = only_road(label)
img2_bg = cv2.bitwise_and(label, label, mask = mask)

img_overlay = cv2.add(img1_bg, img2_bg)

prediction = cv2.imread('/content/drive/MyDrive/data_road/testing/image_2/uu_000093.png')
```

```
prediction = cv2.cvtColor(prediction, cv2.COLOR_BGR2RGB)

show_image(img, 'Original')
show_image(img_overlay, 'Ground Truth')
show_image(prediction, 'Prediction')
```

```
ix = random.randint(0, len(predictions))
num_samples = 10

f = plt.figure(figsize = (15, 25))
for i in range(1, num_samples*4, 4):
  ix = random.randint(0, len(predictions))

  f.add_subplot(num_samples, 4, i)
  imshow(test_images[ix][:,:,0])
  plt.title("Image")
  plt.axis('off')

  f.add_subplot(num_samples, 4, i+1)
  imshow(np.squeeze(test_masks[ix][:,:,0]))
  plt.title("Groud Truth")
  plt.axis('off')

  f.add_subplot(num_samples, 4, i+2)
  imshow(np.squeeze(predictions[ix][:,:,0]))
  plt.title("Prediction")
  plt.axis('off')

  f.add_subplot(num_samples, 4, i+3)
  imshow(np.squeeze(predicton_threshold[ix][:,:,0]))
  plt.title("thresholded at {}".format(thresh_val))
  plt.axis('off')

plt.show()
```

```
import cv2
import numpy as np

def make_coordinates(image, line_parameters):
  try:
    slope, intercept = line_parameters
```

```
    except TypeError:
        slope, intercept = 0.001,0
    y1 = image.shape[0]
    y2 = int(y1*(3/5))
    x1 = int((y1 - intercept)/slope)
    x2 = int((y2 - intercept)/slope)
    return np.array([x1, y1, x2, y2])

def average_slope_intercept(image, lines):
    left_fit = []
    right_fit = []
    for line in lines:
        x1, y1, x2, y2 = line.reshape(4)
        parameters = np.polyfit((x1, x2), (y1, y2), 1)
        slope = parameters[0]
        intercept = parameters[1]
        if slope < 0:
            left_fit.append((slope, intercept))
        else:
            right_fit.append((slope, intercept))
    left_fit_average = np.average(left_fit, axis = 0)
    right_fit_average = np.average(right_fit, axis = 0)
    left_line = make_coordinates(image, left_fit_average)
    right_line = make_coordinates(image, right_fit_average)
    return np.array([left_line, right_line])

def canny(image):
    gray = cv2.cvtColor(image, cv2.COLOR_RGB2GRAY)
    blur = cv2.GaussianBlur(gray, (5, 5), 0)
    canny = cv2.Canny( blur, 50, 150)
    return canny

def display_lines(image, lines):
    line_image = np.zeros_like(image)
```

```
    if lines is not None:
        for line in lines:
            x1, y1, x2, y2 = line.reshape(4)
            cv2.line(line_image, (x1, y1), (x2, y2), (255, 0, 0), 10)
    return line_image

def region_of_interest(image):
    height = image.shape[0]
    triangle = np.array([[200, height], [1100, height], [550, 250]])
    mask = np.zeros_like(image)
    cv2.fillPoly(mask, pts=[triangle], color=[255, 0, 0])
    masked_image = cv2.bitwise_and(image, mask)
    return masked_image

cap = cv2.VideoCapture("test2.mp4")
while(cap.isOpened()):
    _, frame = cap.read()
    canny_image = canny(frame)

     lines = cv2.HoughLinesP(region_of_interest(canny_image), 2, np.pi/100,
100, np.array([]), minLineLength = 40, maxLineGap = 5)
    averaged_lines = average_slope_intercept(frame, lines)
    line_image = display_lines(frame,  averaged_lines)
    combo_image = cv2.addWeighted(frame, 0.8, line_image, 1, 1)
    cv2.imshow("result", combo_image)
 cv2.waitKey(1)
```

road image segmentation

ORIGINALITY REPORT

16%	10%	12%	6%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	coek.info Internet Source	2%
2	www.hindawi.com Internet Source	2%
3	www.iaeme.com Internet Source	1%
4	Hrag-Harout Jebamikyous, Rasha Kashef. "Autonomous Vehicles Perception (AVP) Using Deep Learning: Modeling, Assessment, Challenges", IEEE Access, 2022 Publication	1%
5	www.researchgate.net Internet Source	1%
6	www.mdpi.com Internet Source	1%
7	paperswithcode.com Internet Source	1%
8	Hironobu Fujiyoshi, Tsubasa Hirakawa, Takayoshi Yamashita. "Deep learning-based	1%