REPUBLIC OF TÜRKİYE

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Electrical and Computer Engineering

IMPROVING LANE DETECTION AND STEERING CONTROL IN SELF-DRIVING VEHICLES THROUGH MACHINE LEARNING

Namariq Mohammed Swadi ALJAAFARI

Master's Thesis

Supervisor

Asst. Prof. Dr. Ayça KURNAZ TÜRKBEN

İstanbul, 2024

IMPROVING LANE DETECTION AND STEERING CONTROL IN SELF-DRIVING VEHICLES THROUGH MACHINE LEARNING

Namariq Mohammed Swadi ALJAAFARI

Electrical and Computer Engineering

Master's Thesis

ALTINBAŞ UNIVERSITY

2024

The thesis/dissertation titled IMPROVING LANE DETECTION AND STEERING CONTROL IN SELF-DRIVING VEHICLES THROUGH MACHINE LEARNING prepared by NAMARIQ MOHAMMED SWADI ALJAAFARI and submitted on …………………… (DATE) has been **accepted unanimously/by majority of votes** for the degree of Master of Arts/Master of Science/PhD in ……………………………

______________________________	______________________________
Academic Title – First and Last Name of the Co-Supervisor	Academic Title – First and Last Name of the Supervisor

Thesis Defense Committee Members:

Academic Title - First/Last Name	Department, University	___________________
Academic Title - First/Last Name	Department, University	___________________
Academic Title - First/Last Name	Department, University	___________________
Academic Title - First/Last Name	Department, University	___________________
Academic Title - First/Last Name	Department, University	___________________

I hereby declare that this thesis/dissertation meets all format and submission requirements of a …………………………… (Master's/PhD) thesis/dissertation.

I hereby declare that all information/data presented in this graduation project has been obtained in full accordance with academic rules and ethical conduct. I also declare all unoriginal materials and conclusions have been cited in the text and all references mentioned in the Reference List have been cited in the text, and vice versa as required by the abovementioned rules and conduct.

Namariq Mohammed Swadi ALJAAFARI

Signature

DEDICATION

I dedication my thesis to:

University of Altinbas

My very respected supervisor

All my family members, especially my very dear mother and father.

ABSTRACT

IMPROVING LANE DETECTION AND STEERING CONTROL IN SELF-DRIVING VEHICLES THROUGH MACHINE LEARNING

ALJAAFARI, Namariq Mohammed Swadi

M.Sc., Electrical and Computer Engineering, Altınbaş University

Supervisor: Asst. Prof. Dr. Ayça KURNAZ TÜRKBEN

Date: 07/2024

Pages: 72

Auto lane keeping, an increasingly prevalent driver assistance technology in modern vehicles, facilitates the accurate positioning of the vehicle within road lanes, a crucial aspect for subsequent lane deviation and trajectory planning in fully autonomous vehicles. Traditional lane detection methods have historically relied on sophisticated hand-crafted features and heuristics, which, while computationally efficient, face scalability challenges due to the diverse and dynamic nature of road scenes. However, recent advancements in machine learning, particularly with Convolutional Neural Networks (CNNs), have revolutionized this field by replacing hand-crafted feature detectors with deep networks capable of learning pixel-wise lane segmentations. In this thesis, we aim to address the lane detection problem using a variety of methods. To delve deeper, we utilize a dataset comprising highway lane images to conduct a comparative analysis of two distinct methods. Initially, we employ the traditional edge-detection method, featuring hand-crafted features. Subsequently, we explore various Deep Convolutional Network (CNN) architectures tailored to tackle the lane detection challenge. Our investigation culminates in a comparative assessment of these methods, leveraging images derived from their respective outputs.

Keywords: Lane Detection, CNN, Vision Transformer, Deep Learning, Semantic Segmentation, Computer Vision.

ÖZET

MAKİNA ÖĞRENİMİ YOLUYLA KENDİNDEN SÜRÜCÜ ARAÇLARDA ŞERİT ALGILAMA VE DİREKSİYON KONTROLÜNÜN GELİŞTİRİLMESİ

ELJAAFARI, Namariq Muhammed Swadi

Yüksek Lisans, Elektrik ve Bilgisayar Mühendisliği, Altınbaş Üniversitesi

Danışman: Dr. Öğr. ÜyesiAyça KURNAZ TÜRKBEN

Tarih: 07 /2024

Sayfa: 72

Modern araçlarda giderek yaygınlaşan bir sürücü destek teknolojisi olan otomatik şerit tutma, aracın yol şeritleri içinde doğru şekilde konumlandırılmasını kolaylaştırır; bu, tam otonom araçlarda sonraki şerit sapması ve yörünge planlaması için çok önemli bir husustur. Geleneksel şerit tespit yöntemleri tarihsel olarak karmaşık el yapımı özelliklere ve buluşsal yöntemlere dayanmıştır; bunlar hesaplama açısından verimli olsa da yol manzaralarının çeşitli ve dinamik doğası nedeniyle ölçeklenebilirlik zorluklarıyla karşı karşıyadır. Bununla birlikte, makine öğrenimindeki son gelişmeler, özellikle Evrişimli Sinir Ağları (CNN'ler) ile, el yapımı özellik dedektörlerinin piksel bazında şerit bölümlendirmelerini öğrenebilen derin ağlarla değiştirilmesiyle bu alanda devrim yarattı. Bu tezde şerit tespit problemini çeşitli yöntemler kullanarak çözmeyi amaçladık. Daha derine inmek için, iki farklı yöntemin karşılaştırmalı analizini gerçekleştirmek amacıyla otoyol şeridi görüntülerinden oluşan bir veri kümesi kullanıyoruz. Başlangıçta, el yapımı özelliklere sahip geleneksel kenar algılama yöntemini kullanıyoruz. Daha sonra, şerit algılama sorununun üstesinden gelmek için tasarlanmış çeşitli Derin Evrişimli Ağ (CNN) mimarilerini araştırıyoruz. Araştırmamız, ilgili çıktılardan elde edilen görsellerden yararlanarak bu yöntemlerin karşılaştırmalı bir değerlendirmesiyle sonuçlanıyor.

Anahtar Kelimeler: Şerit Tespiti, CNN, Görüş Transformatörü, Derin Öğrenme, Semantik Segmentasyon, Bilgisayarlı Görme.

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

ABBREVIATIONS

GPS	:	The Global Positioning System
V2V	:	Vehicle To Vehicle
VANET	:	Vehicular Ad Hoc Network
DM	:	Data Mining
NHTSA	:	National Highway Traffic Safety Administration
WLAN	:	Wireless Local Area Network

1. INTRODUCTION

1.1 GENERAL CONTEXT

Self-driving cars represent a groundbreaking advancement in automotive technology, allowing vehicles to navigate and operate without human intervention. While the concept of autonomous driving has been contemplated for years, the complexity of developing such systems has historically posed significant challenges. Recent strides in artificial intelligence have revolutionized the feasibility of this objective, attracting widespread attention and investment from leading automotive industries. The potential benefits of autonomous driving are immense, promising to enhance road safety by reducing accidents caused by human error. This transformative shift is propelled by the rapid evolution of artificial intelligence, which continues to undergo significant advancements year after year. Within the automotive sector, self-driving technology is increasingly regarded as indispensable for the vehicles of tomorrow, marking a paradigm shift in transportation towards safer and more efficient mobility solutionsIn this study, we extensively investigate a branch of artificial intelligence applied to autonomous driving, specifically focusing on end-to-end models utilizing imagery data. End-to-end models represent a departure from traditional approaches by taking raw data as input and directly mapping it to decision outputs, without incorporating domain-specific knowledge into the process. Historically, this approach was considered overly ambitious due to technological limitations and knowledge gaps. However, recent advancements have shifted this perception, with end-to-end models demonstrating promising results across various domains. In our research, we aim to develop a model that processes raw images captured by a front-mounted camera on a vehicle and translates them into direct decisions, specifically determining the steering wheel angle appropriate for the given situation.

The goal of this independent framework is to keep the vehicle inside the path keeping up with smooth driving way of behaving, similar to the one performed by a human. The models introduced are fit for figuring out how to drive a vehicle from human driving experience, arriving at critical exhibitions after an instructional meeting in light of a couple of long stretches of information. As a piece of the cutting edge, we present a speedy resume of the historical backdrop of start to finish models used to foster programmed controlling frameworks, beginning from pioneer structures like [1] created in 1989 with an early form of Multi-facet Perceptron, going on with more refined convolutional models like [2] in 2004 and [3] in 2016 , and closing with the Udacity Challenge #2 completely devoted to foster a path focusing highlight utilizing just profound learning.

1.2 AUTONOMOUS DRIVING

These days, perhaps of the most basic and convoluted task in the car business is planning and executing a computerized framework able to do totally controlling and driving a vehicle without the immediate communication of a human driver. This is a very hard endeavor, consolidating different know-hows from numerous specialized trains and using the most current innovation in the space of sensors, actuators, gadgets, PC vision, man-made brainpower, and numerous others.

A few instances of conceivable and functional frameworks as of now exist in the fields of railroad transportation (e.g., the Turin computerized underground) and flight (e.g., autopilot arrangements overseeing plane flying). Moreover, a few driver-less computerized applications will be presented without further ado with regards to self-driving cultivating work vehicles. Conversely, there is no veritable execution of a totally mechanized framework for cars and business vehicles available, and it is still in the examination stage. The justification for this divergence is the unpredictability of the different working circumstances. The course of a train or subterranean truck is confined by a railway, which is ideally free of any approaching barrier. In the case of an airplane, the flight route has already been calculated, and impediments (typically other planes) are detected at great distances and should not materialize abruptly in front of the system. Conversely, the street situation is comprised of different elements that act in an extremely unique and surprising way. For instance, in a metropolitan intersection, an independent driving framework ought to consider the traffic light, the path limits, the other vehicles' situation and conduct, plausible different entertainers, for example, walkers and cyclists who could out of nowhere go across the street, and so on. A framework working in such a climate should identify and separate each possible hindrance in a trustworthy and effective way, be responsive to outer risks, and have the option to change its conduct in response to those data sources progressively. Moreover, it should work in all circumstances, for example, during an extreme downpour or blizzard, in daytime or around evening time.

The motivations for pursuing the aim of autonomous driving are several. The most significant aspect is increased safety. As indicated by a new specialized examination incorporated by the US Public Thruway Traffic Security Organization (NHTSA), human blunder represents 94% of all street mishaps [4]. As per the Italian power ISTAT-ACI [5], there were 172,344 vehicle mishaps in Italy in 2018, bringing about 242,621 wounds and 3,325 passings. The significant reasons are distraction, failure to regard priority standards, and quick speed, which complete 40.8% of the cases. Moreover, most of deadly street mishaps happen around evening time, when drivers are bound to be exhausted or affected by opiates, with a pinnacle of 9 passings for each 100 episodes somewhere in the range of 5 and 6 a.m. These contemplations feature the requirement for a framework that might end up being useful to the driver stay away from hazardous conditions for himself/herself and other street clients, essentially bringing down the rate of car crashes.

One more issue that independent vehicles could assist with tending to is gridlock, which prompts the arrival of unsafe gasses [6]. As a matter of fact, an independent driving framework ought to have the option to gather traffic data from web applications and find the best technique to arrive at the client's ideal area while likewise bringing down trip time [7]. Moreover, straightforwardly dealing with the motor valve (on account of exemplary gas engines) may support eco-friendliness. This kind of gadget will likewise consider a decrease in the strain and exhaustion that are for the most part made while driving in rush hour gridlock or long trips [8]. The time customarily spent by drivers in such circumstances could be reused and allotted to additional useful exercises. Also, the reception of independent vehicles stretches out the idea of versatility to sections of the populace that might confront obstructions to driving, whether because of actual restrictions or different variables. This can possibly fundamentally upgrade the personal satisfaction and efficiency for these people and their networks. However, achieving the objective of fully automated driving presents considerable challenges, including the possibility of substantially increasing the development and production costs associated with such systems. Studies, such as those referenced in [10], have highlighted the potential costliness of this technology for manufacturers, prompting considerations of alternative business models. These discussions include the possibility of autonomous vehicles being primarily utilized in shared driverless taxi services rather than being sold directly to individual customers.

Furthermore, doubts surrounding this technology primarily stem from concerns about the potential for system failures. Various occurrences of mishaps coming about because of disappointments during testing or genuine utilization of semi-computerized driving frameworks currently available exist. For example, in 2016, a Tesla Model S furnished with a semi-computerized autopilot neglected to identify a white truck against the background of a brilliant sky, bringing about a lethal impact that guaranteed the driver's life [11]. Similarly, in 2018, an experimental Uber vehicle struck and killed a pedestrian crossing a road with a bicycle [12]. In this case, the system initially misclassified the pedestrian as an unknown object, then as a vehicle, and finally as a bicycle due to poor lighting conditions, causing a delay in emergency braking activation. These accidents underscore the fact that current systems are still not robust enough to handle the myriad situations encountered in real-world scenarios.

Indeed, the emergence of vehicles operating without direct human control has raised significant ethical and legal dilemmas. One pressing question pertains to responsibility in the event of an accident caused by a sudden system failure. Determining liability becomes complex when traditional notions of driver responsibility are blurred by automated systems. Additionally, scenarios involving potential harm, such as a choice between saving passengers or pedestrians in a dangerous situation, present ethical challenges. Decisions about how to program autonomous vehicles to prioritize lives or mitigate harm pose profound moral questions. Resolving these issues requires careful consideration of legal frameworks, ethical principles, and societal values to establish guidelines that prioritize safety, fairness, and accountability in the deployment of autonomous driving technology.

1.3 GENERAL STRUCTURE OF AN AUTOMATED DRIVING SYSTEM

The method involved with making and sending a utilitarian and reliable independent driving framework is very hard. A few framework plans have been created as of late to effectively address this challenge [13]. Among the best in class in building choices, we might distinguish two particular techniques to connectedness and two unique ways to deal with algorithmic plan.

For the main subject, we might recognize "self image as it were" and "associated" frameworks. Inner self just frameworks are predicated with the understanding that the entirety of the equipment and programming parts expected to finish the powerful driving task are now ready the vehicle. This is the most predominant procedure utilized in the best in class. The key advantage is the framework's autonomy from data given by different vehicles or foundation. Inner sensors give every one of the ecological data sources expected to follow through with a responsibility, making them more dependable than obscure outer sources. Besides, having an independent stage could help with the entire framework improvement and approval process. The downsides are associated with the way that some data won't be quickly removed without outside contact, making the failure of getting explicit ecological data sources or demonstrating a postpone in social event them.

Associated frameworks adopt an unmistakable strategy fixated on the trading of traffic and state data among peers, as well as correspondence with foundation components. Working with this correspondence is the usage of a VANET, which works as a remote transmission network utilizing either the 802.11p IEEE standard or utilizing the 4G/5G cell organization. Correspondence inside this structure can happen between at least two vehicles, taking into consideration the sharing of speed and course data, an idea named Vehicle-to-Vehicle (V2V) correspondence. Conversely, communication may also occur between a vehicle and an infrastructure element, such as a traffic light sharing information about upcoming signal changes, known as Vehicle-to-Infrastructure (V2I) communication. Collectively, these communication types are encompassed by the term Vehicle-to-Everything (V2X) when referring to connected vehicles. While this approach holds promise for enhancing the sensitivity of the system environment compared to ego-only systems, its operational implementation has yet to be realized due to the complexity of scenarios involving the exchange of data among hundreds or thousands of vehicles within urban settings.

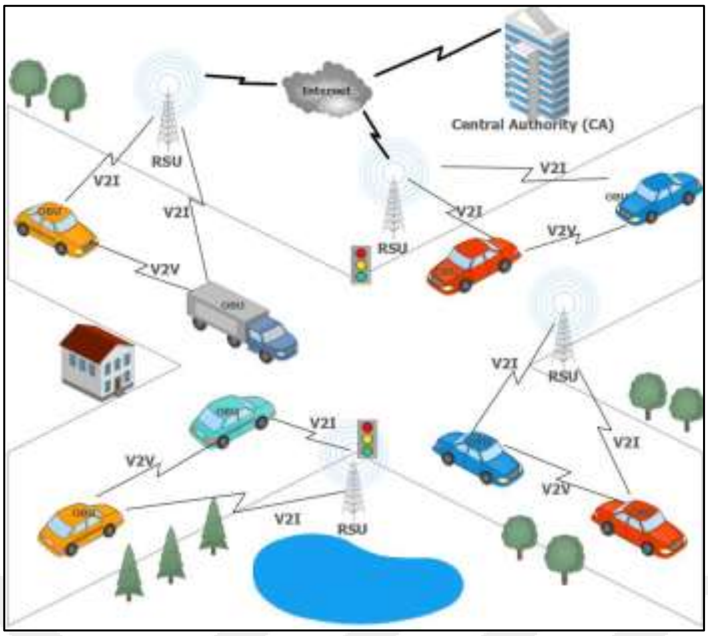

Figure 1.1: Example of a Vehicular Ad hoc NETwork (VANET)

There are two potential structural ways to deal with calculation plan. The first is the "secluded framework" engineering, which incorporates an organized pipeline of different parts that interfaces the climate, inner state information sources, and incitation yields. This is the most regularly used choice in the best in class.

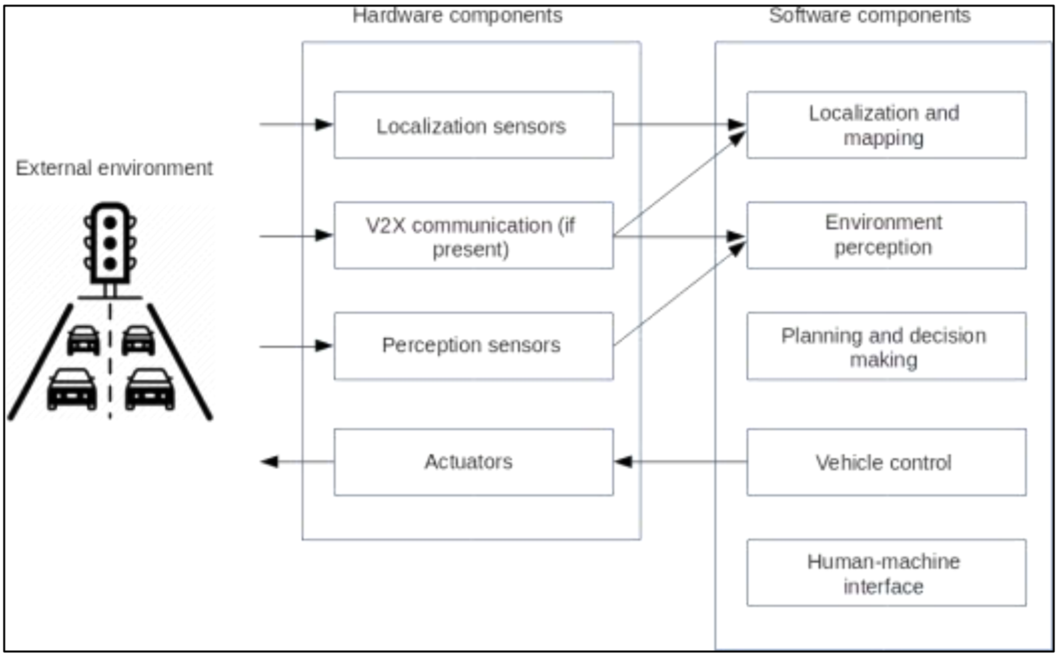

Figure 1.2: Architecture of a Modular Autonomous Driving System.

The key advantage of this technique is the ability to create and test each module individually, splitting the dynamic driving job into multiple relatively simple sub-tasks. The vital downside concerns the possibility of blunder engendering: in case of a slip-up in the lower modules of the pipeline, the accompanying module will figure its result starting from a wrong information, delivering a framework disappointment while impelling the vehicle control.

The "start to finish" plan idea spins around creating movement control activities straightforwardly from inputs utilizing a solitary module. This approach can be acknowledged through three unmistakable AI procedures:

i. Direct Supervised Deep Learning: This method involves training the system to learn appropriate actions based on inputs, mimicking examples provided by a ground truth source, typically an expert driver. Training occurs offline, without the need for real-world deployment. However, due to its reliance on limited supervised training data, this approach often suffers from poor generalization capacity.

ii. Neuro-evolution Learning: Similar to supervised deep learning, neuro-evolution learning employs evolutionary algorithms to train neural networks without relying on backpropagation or explicit supervision. This method offers an alternative route to training without the need for labeled datasets.

iii. Deep Reinforcement Learning: Unlike the previous approaches, deep reinforcement learning adopts a different paradigm. Here, the system learns optimal actions based on inputs by maximizing cumulative reward functions. This method doesn't rely on human models for imitation but rather learns independently through trial and error. While offering superior generalization capacity compared to supervised methods, deep reinforcement learning requires online training in real-world scenarios to continuously refine its decision-making abilities.

These methodologies represent different approaches to realizing the "end-to-end" design concept, each with its unique strengths and considerations regarding generalization capacity, training requirements, and real-world applicability.

The main issue with employing this type of solution is that these networks must interact with the environment on a large scale and fail several times before achieving a functioning and operational implementation. Moreover, in case of disappointment, it is very challenging to appreciate which deficiencies happen and why, making it extreme to cure the issues.

As far as key equipment parts, an independent driving framework requires an assortment of sensors to gather data from the climate, a bunch of actuators to direct the ideal activities as result, and a PC stage to carry out the recently depicted framework engineering. The activation is led on the directing framework to control the horizontal elements, on the slowing mechanism.

1.4 MOTIVATION

Many individuals throughout the world spend a significant amount of time driving, and they want to do it safely. Automobile accidents represent a grim reality, standing as the foremost cause of death and injury in many nations worldwide. According to statistics from the World Health Organization (WHO), traffic accidents claim the lives of over one million individuals annually. Consequently, there has been a concerted effort by both academic institutions and automotive companies in recent years to prioritize the advancement of Advanced Driver Assistance Systems (ADAS) and autonomous vehicles. This collective focus underscores the urgent need to mitigate the devastating toll of traffic accidents and enhance road safety through innovative technological solutions.

Autonomous driving vehicles epitomize a transportation mode devoid of human intervention, employing algorithms executed by an onboard computer to replicate driver behavior and render decisions. To enable autonomous operation, it is essential to gather environmental data from various sensors, including LiDAR, radar, cameras, IMUs, and GPS. The information is consolidated and overseen by a cohesive system termed sensor fusion, streamlining the creation of perception algorithms. These perception algorithms form the bedrock for path and trajectory planning, as well as the control phase of autonomous vehicles. They fulfill diverse functions including lane keeping, emergency braking, parking maneuvers, speed regulation, and obstacle avoidance. These functionalities generate signals that direct the actions of actuators, enabling precise control and navigation of the autonomous vehicle.

1.5 PROBLEM STATEMENT

Perceiving lanes under various weather conditions and settings is imperative to ensure the vehicle remains centered on the road during maneuvers like lane changes. Challenges arise from factors such as overexposure, shadows, worn-out lane markings, occlusion caused by leading vehicles, and wet or reflective roads. Consider the scenario of an autonomous car traversing through a tunnel: sudden shifts in lighting conditions can momentarily disrupt lane detection, prompting the vehicle to prioritize maintaining its position on the road. Similarly, when large vehicles obstruct the view ahead, lane lines may become partially or entirely obscured, complicating the recognition process. False lane detections pose a significant risk, potentially leading to unintended route alterations and endangering occupants. Failure in lane detection could result in the vehicle veering off the road, leading to accidents.

To overcome these obstacles and minimize false positive rates, extensive research has been conducted, yielding a plethora of proposed techniques. These include advancements in sensor fusion, leveraging multiple data sources to enhance lane perception accuracy. Additionally, robust computer vision algorithms capable of adaptively adjusting to changing environmental conditions have been developed. Machine learning approaches, particularly deep learning models, offer promise in effectively distinguishing genuine lane markings from spurious ones, thereby reducing false positive rates. Furthermore, real-time feedback mechanisms and redundant sensor systems can provide fail-safe measures to ensure reliable lane detection performance under diverse circumstances. Overall, a multidisciplinary approach combining advanced sensor technologies, sophisticated algorithms, and rigorous validation processes is essential to address these challenges and achieve robust lane detection capabilities in autonomous vehicles.

1.6 OBJECTIVES

The main objectives of the research are as follows:

i. Evaluate Traditional Lane Detection Methods: Assess the effectiveness of traditional edge-detection methods with hand-crafted features in accurately detecting lane markings on highway images.

ii. Compare Deep Learning Approaches: Investigate and compare the performance of various Deep Convolutional Network (CNN) architectures for lane detection, considering factors such as accuracy, robustness, and computational efficiency.

iii. Assess Robustness to Road Scene Variability: Evaluate the robustness of the lane detection algorithms to variations in road scenes, such as changes in lighting conditions, road surface texture, lane markings quality, and presence of occlusions.

iv. Explore Generalization Abilities: Investigate the generalization abilities of the deep learning models trained on highway lane images to detect lanes in diverse real-world driving scenarios beyond the training dataset.

v. Consider Real-time Implementation: Assess the feasibility of real-time implementation for both traditional and deep learning-based lane detection methods, considering factors such as processing speed and hardware requirements.

vi. Contribute to Advancements in Autonomous Driving: Contribute to the broader goal of advancing autonomous driving technology by improving the accuracy and reliability of lane detection systems, ultimately enhancing vehicle safety and user experience

1.7 CONTRIBUTION

The contribution of this thesis lies in its comprehensive investigation and comparison of traditional edge-detection methods with hand-crafted features and various Deep Convolutional Network (CNN) architectures for lane detection in modern vehicles. By evaluating the performance, robustness, and real-time feasibility of these methods using a dataset of highway lane images, this research provides valuable insights into the strengths and limitations of different approaches. Furthermore, the analysis of model outputs sheds light on the effectiveness of traditional versus deep learning-based techniques in accurately detecting lane markings under diverse road conditions and scenarios. This research not only advances our understanding of lane detection technologies but also contributes to the broader goal of enhancing autonomous driving systems' capabilities, ultimately paving the way for safer and more reliable transportation solutions in the future.

1.8 THESIS OUTLINE

The second chapter of this thesis paper discusses related studies on current lane detecting algorithms. The implementation approach and network training options are covered in Chapter 3. Chapter 4 elaborates on the experiments and their outcomes. The experimental results are thoroughly explored. Chapter 5 presents the conclusions and recommendations for further study.

2. LITERATURE REVIEW

2.1 INTRODUCTION

Many studies have been conducted in recent years on autonomous driving, and as a result, an overview of existing initiatives is provided at the start of this work. After establishing the significance of lane recognition for automobiles in the Introduction chapter, the primary objective is to consistently and coherently identify lane lines across all weather conditions and driving scenarios. This chapter delves into pertinent strategies for tackling the challenging task of lane detection. Lane detection techniques are broadly categorized into classical and deep learning methods. Classical lane detection systems rely on human mapping functions or algorithms to establish the relationship between observable road characteristics and lane lines. In contrast, deep learning approaches eschew the creation of a mapping function between observed road characteristics and lane lines. Instead, these approaches train the mapping function by analyzing extensive datasets containing a variety of attributes. Another distinction between classic and deep learning approaches lies in the input methodology. Spatial approaches process individual images, while spatio-temporal methods analyze a sequence of images and incorporate temporal information between consecutive time steps. Figure 2.1 provides an overview of the approaches outlined in both classic lane detection and deep learning methods.

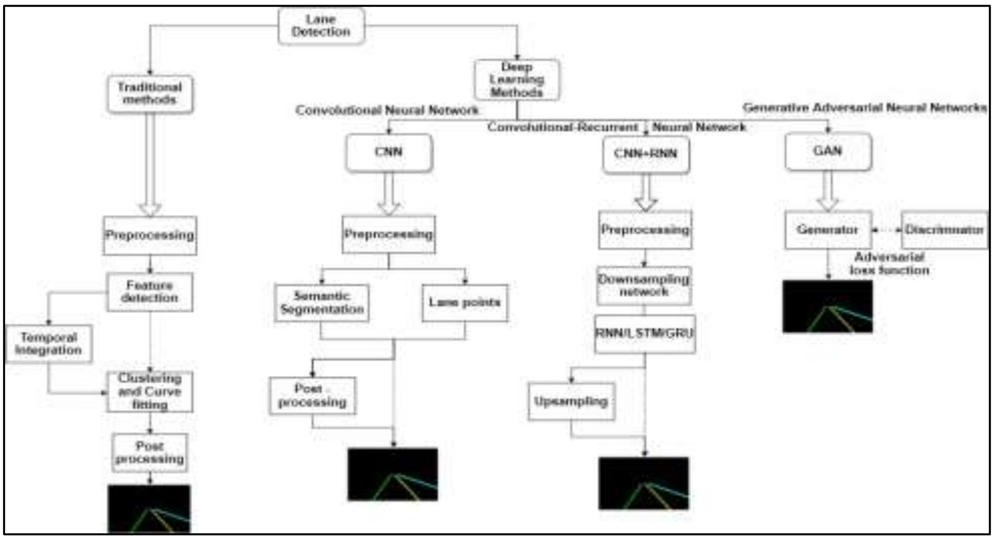

Figure 2.1: Overview of Traditional Lane Detection Method and Deep Learning Method of Lane Detection.

Traditional lane detection (figure 2.1) uses a broad technique that includes picture preprocessing to denoise and remove the camera viewpoint, detecting important lane points, clustering the lane points, and fitting a curve to create lane lines. A postprocessing stage is likewise wanted to diminish bogus positive path discovery. A few methodologies consider worldly information and integrate it into the strategy in the wake of identifying significant path focuses. The following segments give a nitty gritty clarification of these activities. In profound learning draws near (figure 2.1), convolutional brain networks preprocess the image and either give semantic division (pixel-wise arrangement) or conjecture the place of path focuses by contracting or downsampling the pictures. These outcomes are postprocessed to dispose of anomalies. Intermittent brain network-based approaches utilize indistinguishable preprocessing and downsampling stages, however they likewise add fleeting data from earlier pictures through Lengthy Transient Memory (LSTM) or Gated Repetitive (GRU) units. To perceive paths, the photographs are resized or upsampled to their unique aspects. Generative ill-disposed brain networks have a generator that makes an identified path picture and advances by recognizing the created and unique pictures. These methodologies are depicted top to bottom in the accompanying segments.

Traditional lane identification algorithms adopt a heuristic approach to detect and model lanes. Utilizing a mono-camera sensor, a sequence of road images is captured, which can subsequently undergo processing using matrix operations. Lane points are identified and consolidated into lane line instances through the application of filters and algorithms. The typical system comprises several stages, including image preprocessing, feature extraction, integration of temporal information, clustering, curve fitting, and postprocessing. Figure 2.2 below provides a visual representation of these sequential stages.

The calculations inspected in the writing address a subset of the summed up framework and may not consolidate each of the stages illustrated previously. For example, the strategy proposed by Low et al. doesn't use worldly information from picture groupings however sticks to different stages recorded in Figure 2.1. Subsequently, each move toward the framework is completely investigated, and a correlation of different strategies is given.

Figure 2.2: Outline of Traditional Method For Lane Detection Task.

2.2 LANE DETECTION

In general, lane detection does not use a single technique. The following studies have been researched in order to gain a more comprehensive understanding of the many ways that may be used: these provide various distinguishing characteristics. [15] presents a vision-based technique for lane detection. Their technique enables real-time operations while being resilient to illumination changes and shadows. Figure 2.3 depicts the block plan of the computation made during this audit. As displayed in the block plan, the image from a camera is changed over totally to grayscale and dealt with to lessen noise. Following these cycles, the Watchful calculation was rushed to look through the edges of the paths.

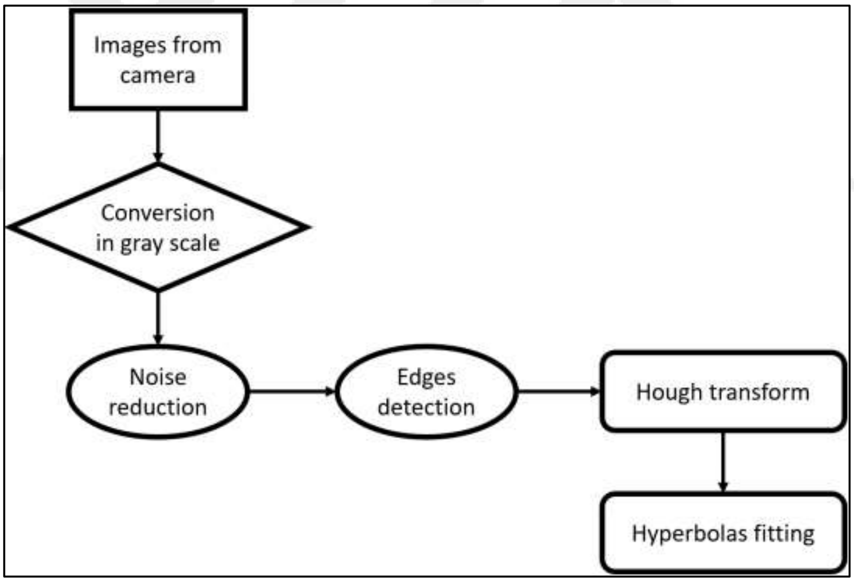

Figure 2.3: Block Scheme.

According to the SAE definition, one of the most important tasks that an autonomous driving system must do is environmental monitoring. This occupation is basic since it permits the vehicle to appropriately control its longitudinal and sidelong elements as well as answer outside boosts continuously. Presently, the essential bottlenecks being developed and research on this occupation are addressed by the impediment identification and characterization, as well as the street and path discovery, for example the acknowledgment of a locale wherein the vehicle might travel (known as the drivable region).

Both sub-tasks within the environment monitoring task face a common critical challenge: they must operate effectively under diverse environmental conditions, including scenarios with highly variable brightness such as those influenced by shadows, as well as extreme weather conditions. Achieving this level of generalization remains an ongoing research challenge. Furthermore, the errands should fight with the need to distinguish a shifting number of paths across various street geographies, alongside the fluctuation in path markings' size and variety, which entangles path discovery. Moreover, these undertakings assume a vital part in shutting the circle of sidelong control calculations.

The focal point of this proposition will be on the path location task, which is the most common way of perceiving a district in the street space that has a place with a path and is characterized by its path markers. Numerous procedures have been introduced in the writing to resolve this issue, a large number of which depend on similar perceptual data sources used by human drivers, including street tone and surface, street cutoff points, and path markings. In any case, it ought to be feasible to coordinate different data sources, like vehicle-to-vehicle (V2V) or vehicle-to-foundation (V2I) correspondences, however, particularly for the subsequent sort, it can't be ensured that this sort of administration would be executed and kept up with on each street, because of the enormous required costs. Past the domain of completely mechanized driving, path location fills in as the essential perceptual contribution for different high level driving help frameworks. Among the most huge are:

i. Lane Departure Warning (LDW): This system provides a simple alert to the driver when the vehicle is on the verge of exiting the current lane, thereby helping to prevent unintentional lane departures.

ii. Lane Keeping Assist (LKA): LKA takes a step further by actively correcting the vehicle's lateral direction within certain limits to ensure it remains within the current lane. This assistance system provides continuous support to the driver in maintaining proper lane positioning.

iii. Automatic Lane Keeping Control (ALKC): Offering full authority over the vehicle's lateral control, ALKC functions similarly to LKA but with the ability to autonomously maintain the vehicle within the lane without requiring driver intervention. This system represents a more advanced level of assistance, providing enhanced convenience and safety by actively controlling the vehicle's trajectory.

In these examples, the framework should have the option to recognize the current path region from a brief distance (40 to 50 meters). In additional modern frameworks, for example, Tesla Autopilot, the path recognition module is likewise used to independently perceive adjoining paths and switch to another lane. In this present circumstance, the framework ought to have the option to perceive a few paths from an impressive distance (roughly 150 meters).

The initial stage in every lane detecting system is to collect inputs from the external environment. Several sensors have been proposed in literature to gather this information, including cameras with single or stereo vision.

The most ordinarily used detecting methodology in writing is camera-based. The fundamental explanations behind this decision are two: the first is that way markings and road limits are expected for human drivers' vision, so they should be observable in essentially every condition for cameras; and the second is the negligible cost of such sensors, which, got together with their power, ensures a fair clever plan. With a solitary camera, the execution cost is negligible, yet no 3D portrayal of the environmental factors is doable. Such a goal can be accomplished by using a sound system vision framework; be that as it may, this subsequent choice can prompt more prominent processing costs and an expanded error likelihood.

Another crucial component is LiDAR, a sensor capable of efficiently mapping the vehicle's surroundings in 3D. When used in conjunction with cameras, LiDAR can effectively address scenarios where conventional sensors struggle to provide accurate results. For instance, in situations where road markings are unclear or absent, LiDAR can accurately detect the boundaries and their distances relative to the vehicle. Moreover, this setup remains unaffected by typical lighting issues or low-visibility conditions. Additionally, LiDAR can aid in estimating approaching gradients and facilitate smoother image adjustments, such as rearview mirror calibration. Despite its effectiveness, the primary drawback of LiDAR sensors is their high cost. However, their deployment can lead to a more precise and detailed 3D representation of the environment compared to stereo vision systems.

At long last, it is conceivable to orchestrate GPS sensors data with IMU evaluations to get the nonstop vehicle position with an accuracy of 1 meter. Consequently, it is feasible to facilitate the vehicle excusing a visual portrayal of the ceaseless way markings yet utilizing the electronic aides depiction of the climate. The really squeezing worries for this present circumstance is the low unflinching quality, because of the need of a consistent correspondence with GPS satellites and the alignment mistake likelihood.

Within the lane detection problem, literature presents two primary approaches stemming from the same inputs:

i. Traditional Methods: These techniques rely on handcrafted features and heuristic algorithms to detect lanes. This conventional approach is commonly utilized in industry and has been refined over time to effectively identify lane markings based on predefined rules and patterns.

ii. Machine Learning-Based Methods: In contrast, machine learning-based approaches, particularly those leveraging convolutional neural networks (CNNs), have emerged as a novel alternative. These methods eschew explicit rule-based programming in favor of training deep learning models to automatically learn relevant features from raw input data, such as images. By leveraging CNNs, researchers aim to develop more adaptive and robust lane detection systems capable of handling diverse road conditions and scenarios.

2.3 LANE KEEPING

An investigation of the potential techniques for creating path keeping framework has been led, as finished for the path discovery issue. Probably the most fascinating methodologies are accounted for underneath. Crafted by [15] is centered around the acknowledgment of a regulator to execute a path keeping framework utilizing Model Prescient Control (MPC) hypothesis. Figure 2.4 shows how the controller is made in this investigation. The outcome is the ideal directing mark of the front wheel figured restricting the cost ability of the MPC controller. The cost capacity is made from the coordinating places and the mix-up between the reference and the judicious course. The age of the reference heading is performed fitting five audit centers coming from sensors.

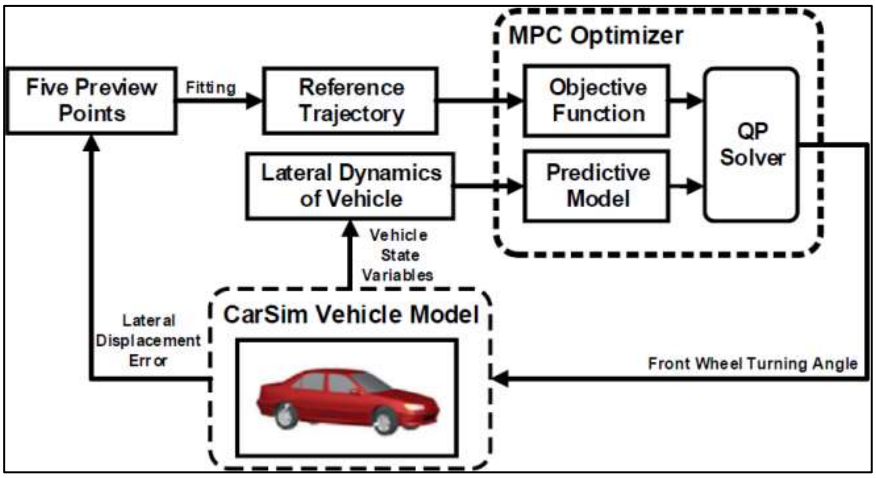

Figure 2.4: Block Scheme Developed by [15].

In their work, [17] introduced a linear Model Predictive Control (MPC) controller designed to manage lane maintenance and obstacle avoidance systems specifically tailored for low curvature roadways. The control mechanism developed in this study is structured into two distinct stages: the underlying stage involves processing slowing down or choke profiles using a forecast skyline, while the resulting stage includes carrying out MPC utilizing straight time-shifting models of the vehicle's sidelong elements got from the profiles created in the principal stage. Through MPC, the directing point guidance is resolved in light of the ideal slowing down or choke orders.

In a separate study conducted in 2009, [18] proposed an innovative approach to route following for autonomous vehicles. Their strategy revolves around fostering a stable Proportional-Integral-Derivative (PID) steering control system that leverages a vision-based framework. While the primary control is utilized to counteract constant disturbances and variations in boundary during the calculation of the steering point, the secondary control aims to mitigate disturbances during cornering.

2.4 DEEP LEARNING BACKGROUND

In this section, the background of the techniques, methodologies and deep learning architectures that are used throughout this work is covered. This section is meant to provide the reader with a clear understanding of the techniques discussed in the Related Work chapter and provide the motivation behind the design choices of our own proposed work.

2.4.1 Convolutional Neural Networks

Image data analysis has been massively affected by the introduction of the CNN, a deep learning model architecture based on the mathematical linear operation of convolution. In order to perform image analysis with deep learning, image data are usually represented as a fixed uniform grid of parameters representing each pixel and their respective colors. The groundbreaking difference of a CNN network from an Artificial Neural Network (ANN) with fully-connected layers, is the way the connections between neurons of successive layers takes place. Instead of just connecting every layer's output neurons to the next layer, a CNN network makes local connections, using convolution. This way, the number of generated network parameters is drastically reduced [19].

A CNN network typically consists of three types of layers: the convolutional layer, the non-linear layers referring to activation functions (e.g., sigmoid, tanh, Rectified Linear Unit(ReLu) and the pooling layers. The convolutional layers apply different regional matrix kernels to the input image, acting as feature ex- tractors, to identify and extract valuable features from the input image. The use of different convolutional kernels, acting as filters, for each convolutional layer allows the network to learn different features from the image. Two additional ways to decrease the number of parameters while also mitigating the possible loss of information side effects, are stride manipulation and padding. Stride dictates the size of the spatial step a convolution kernel makes over an image grid. A larger stride can prevent the filter from revisiting parts of an image, reducing the number of new parameters this way. However, the aforementioned manipulation of the stride parameter can shrink the output of a layer potentially causing information loss. An answer to that is the use of some kind of padding (e.g., zero padding) around the borders of the images in order to prevent the output of the network from shrinking and to make sure that important semantic information that exists near the outer regions of an image is preserved.

In most CNN architectures, a non-linear layer follows the convolutional layer in order to adjust the out- put of the preceding layer using some kind of activation function. Finally, another important operational component of a CNN network is the pooling layer. Its main contribution is to preserve high-value features while down sampling the feature map, thus reducing the complexity of the CNN architecture. Addition- ally, according to the CNN's task at hand (e.g.,, classification, regression), an output is generated by one or more fully connected layers and their respective activation functions.

2.4.2 Transformer Networks

In an effort to push the boundaries of the at the time state-of-the-art models for sequential data, such as Recurrent Neural Networks (RNNs) or Long-short term memory (LSTM) neural networks, the Transformer architecture was proposed [20]. The Transformer architecture immediately offered in- creased parallelization capabilities, compared to the RNNs, as it relies solely on attention mechanisms to extract global dependencies between input and output [20]. An attention mechanism is able to identify correlations between an input and an output sequence, by mapping a query and a set of key-value vector pairs to a set of output vectors. Consequently, these are used to create the attention matrices, which describe the correlation degree between features from the input feature matrices, across the different layers of the network. Transformer networks typically follow an encoder/decoder architecture, which consists of several stacked attention layers as well as some fully connected layers. The Transformer's encoder receives an embedding vector as an input along with a positional embedding representing the position of each embedding in the original sequence. Respectively, the decoder of the network is tasked to create an output embedding, exploiting the features learned from the encoder.

The Transformer architecture originally became popular due to its success in various Natural Language Processing (NLP) related tasks such as, machine translation, sentiment analysis or named entity recognition. This success is credited to the effectiveness of the aforementioned attention mechanism capturing the complex characteristics of natural language. Following the success of the Transformer architecture in NLP related tasks, a Transformer architecture, the Vision Transformer (ViT), was designed to fit the needs of image data [21]. ViT networks fragment the input images into fixed-size rectangular patches, generate vector embeddings for each of those and feed those to the encoder of a Transformer architecture along with the respective positional embeddings. In ViTs, the positional embedding holds information about the position of a given patch in the original image. After being introduced by ([22], ViT networks have been researched quite extensively as a possible replacement of CNN-based networks due to their ability to mitigate some of the downsides of the local convolutional filters used in CNNs. The attention mechanism, along with the patch embedding architecture, offers an alternative to the standard convolutional-based approach, as ViTs manage to capture global characteristics of an image quite successfully. Nevertheless, training ViT networks usually comes with a considerable computational cost, as ViTs need to be pre-trained on large amounts of data before fine-tuning them for downstream tasks [22].

2.4.3 Semantic Segmentation

A task with numerous applications in image analysis-related works is semantic segmentation. Semantic segmentation refers to the process of assigning a class label to each pixel of an image from a predefined set of classes. This way, different entities in an image can be identified. In semantic segmentation, multiple occurrences of the same classes are treated as a single entity. The most common evaluation metrics for semantic segmentation algorithms are the Interesection over Union (IoU) [23] metric, pixel-level accuracy, or the F1 Score.

2.5 RELATED WORKS

2.5.1 Traditional Algorithms

Traditional lane recognition algorithms rely on handcrafted feature selection, which is grounded in prior knowledge and understanding of the problem and its observable properties. These selected characteristics undergo low-level pixel processing, leading to predictable final outputs. Not at all like brain network-based approaches, these calculations are intensely reliant upon the nature of information pictures, which should display the ideal qualities with high perceivability. Nonetheless, due to the predefined highlight choice, the picture might be divided into more modest locales of interest (returns for capital invested), bringing about a lighter and quicker handling.

The majority of traditional lane detection algorithms outlined in the literature adhere to a common workflow, as documented in various studies ([24], [25], [26]). This workflow typically comprises several distinct steps:

i. Pre-processing (Image Cleaning) : This initial stage involves preparing the input image data by applying various techniques to enhance image quality, remove noise, and improve overall clarity. Common pre-processing steps may include noise reduction, contrast enhancement, and color space conversion.

ii. Low-level Processing (Feature Extraction): In this phase, the algorithm identifies and extracts relevant features from the pre-processed image data. These features often include edge points, color gradients, or other distinctive visual cues indicative of lane markings.

iii. Post-processing (Lane Model Fitting): Once features have been extracted, the algorithm proceeds to fit a mathematical model to the detected lane markings. This typically involves curve fitting or line fitting techniques to delineate the trajectory of the lanes within the image.

iv. Temporal Integration: To enhance robustness and stability, many algorithms incorporate temporal integration mechanisms that leverage information from previous frames in a video sequence. This allows for smoother and more consistent lane detection across successive frames.

v. Image to World Correspondence: Finally, the algorithm maps the detected lane markings from the image space to the corresponding real-world coordinates. This step is crucial for accurately estimating the vehicle's position within the lane and facilitating higher-level decision-making processes.

By following this systematic workflow, traditional lane detection algorithms aim to effectively identify and characterize lane markings in input images, laying the groundwork for subsequent analysis and decision-making tasks in autonomous driving systems.

The initial stage of the typical workflow, known as the image cleaning process, focuses on enhancing the input image to accentuate characteristics crucial for subsequent feature extraction. Several image cleaning operations are commonly employed, including obstacle detection and removal, shadow elimination using color-space transformations, and the exclusion of irrelevant image regions. The latter operation, based on the selection of regions of interest (ROIs), ensures that only pertinent areas of the input image are processed by subsequent modules. For instance, in [16], a dynamically computed vanishing line is used to discard pixels above this line, while other approaches such as those in [13], [14], and [15] involve applying image transformations to mitigate perspective effects and concentrate on a limited remapped area. These operations collectively refine the input image, optimizing it for effective feature extraction and subsequent processing in the lane detection algorithm.

The component extraction process in path identification frameworks is the module answerable for identifying significant attributes, to be specific path borders. The elaboration thinks about the picture's low-level pixel depiction. By and large, these limits are addressed by path markers, which can have various actual properties like tone (going from shades of white to yellow or orange), width, and structure (constant lines, ran lines, etc. Nonetheless, as the path markings are expected to be recognizable by people, a few presumptions can be made. The least complex supposition that will be that the markers are more brilliant than the street, inciting a few proposed calculations (e.g., [13] and [15]) to search for dim to light to dim pixel splendor power spikes, which are then sifted utilizing versatile or fixed limits. In different conditions, for example, [14], Gaussian channels are utilized on the cleaned picture. [16] expects that path markers are lines that meet onto the perspective centering point.

The lane model fitting stage involves integrating the extracted features from the preceding module into predefined lane geometric models. These models typically fall into one of three categories:

i. Parametric Models: These models are characterized by a set of parameters that define the shape of the lane, such as curvature or slope. Examples include polynomial functions or line equations, where the parameters determine the shape and position of the lane markings.

ii. Semi-parametric Models: These models combine parametric and non-parametric elements, allowing for greater flexibility while still maintaining some predefined characteristics. They may incorporate features like splines or piecewise functions to capture complex lane geometries.

iii. Non-parametric Models: In contrast to parametric models, non-parametric models do not rely on predefined parameters. Instead, they directly map the extracted features to the lane geometry without explicit parameterization. Techniques such as clustering or graph-based methods may be employed to infer the lane structure directly from the data.

Each type of model offers distinct advantages and trade-offs in terms of flexibility, computational complexity, and robustness to variations in lane markings and road conditions. The selection of an appropriate model depends on factors such as the level of detail required, the complexity of the lane geometry, and computational constraints.

Parametric models are many times straight lines, illustrative bends, or round curves. These are the easiest models, in light of the possibility that a respectable control calculation requires moderately little distances, with assessed path borders. The Hough change is the most frequently involved approach for fitting highlights to a direct model portrayal, as it looks for the most regular line tendency point. A few models are: [13] thinks about the street markings as equal lines after a converse point of view planning change; [27] arrangement utilizes a clothoid, a bend whose ebb and flow is relatively corresponding to its bend length; [28] utilizes the RANSAC strategy to wipe out exceptions and fit to a direct or exaggerated model.

Semi-parametric models, for example, spline bends, are more definite and make no suspicions about the anticipated path shape. Moreover, dissimilar to parametric models, unobtrusive boundary changes bring about a little bend shape change. The burden is the probability of overfitting, bringing about ridiculous bends as results. In [29], the RANSAC approach is utilized to fit the elements to a third-degree Bezier spline. In [30], b-splines, which have more nearby parametrization control than Bezier bends, are used. At long last, non-parametric models depend on a consistent portrayal of their limits. This is a less well known mathematical model.

2.5.2 Neural Network-based Algorithms

In recent years, literature has offered a novel technique to lane identification that differs significantly from existing algorithms. Rather than working at low-level with pixels and contingent upon deduced information on the discernible characteristics of the path markers and the design of the path, various scientists have proposed an AI based other option.

As elucidated, lane detection systems encounter significant challenges stemming from highly variable environmental factors. These include fluctuating daylight conditions, particularly during sunrise and sunset, as well as artificial light interference at night. Additionally, the diversity of road types and structures, shadows cast onto the road surface, and obstacles obstructing lane markers further compound the complexity. Moreover, the presence of faint or scarcely visible lane markers and adverse weather conditions pose additional hurdles for accurate lane detection. Traditional lane recognition algorithms may be quite successful in confined driving contexts, such as highways, if they are designed with adequate feature selection. However, the efficacy of these algorithms might be significantly diminished due to their low generalization capacity in relation to the aforementioned external circumstance changes.

Given the intricacies of the aforementioned challenges, novel approaches leveraging neural networks have emerged. These methods, rooted in deep learning, possess the capability to automatically select features, thus streamlining the complex process and yielding a more resilient solution. Such approaches exhibit enhanced robustness, enabling better generalization of the algorithm across highly variable environmental conditions.

A neural network comprises interconnected elements known as neurons, which process inputs (such as individual pixels in an image) through computational operations. Each neuron typically computes a weighted sum of its inputs and produces an output, often normalized between 0 and 1, which is transmitted to other neurons. These neurons are organized into layers, including an input layer, an output layer, and intermediate layers called hidden layers. In a "fully connected" network, every neuron in a layer is linked to every neuron in the preceding layer.

To train a network to produce certain outcomes when given comparable inputs, the network parameters, or weights utilized in the neuron sum process, must be appropriately configured. Typically, the train procedure begins with random weights and is followed by inputs with known predicted results. This set of inputs is known as a train set. Following computation, the network's outcomes are contrasted with the anticipated values, and the weights are adjusted accordingly. This iterative process, known as back-propagation, is repeated until satisfactory results are achieved. Subsequently, the network's performance is evaluated using inputs that were not utilized during training. This collection of inputs is referred to as the "test set," allowing for an assessment of the network's effectiveness across unseen data.

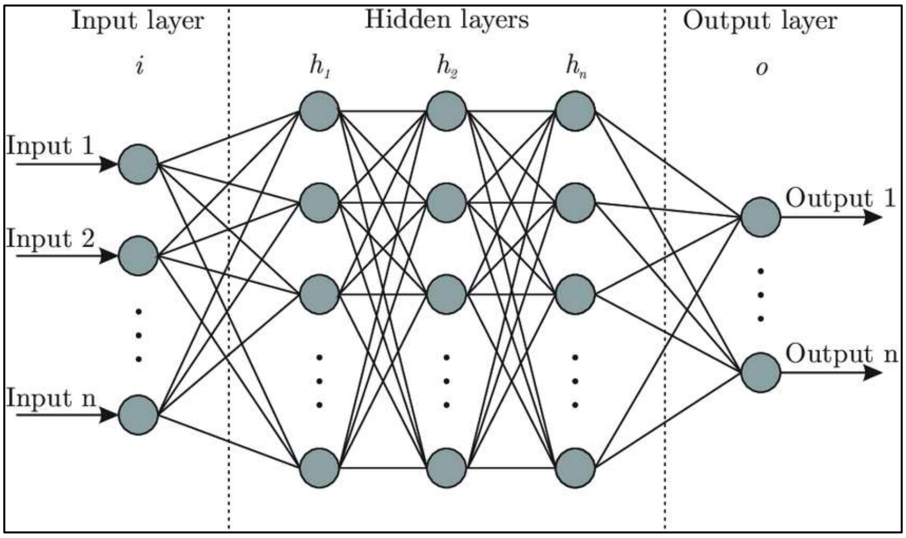

Figure 2.5: Structure of a Generic Neural Network.

In a convolutional neural network (CNN), various types of layers are employed, with the most common being the convolutional layer. Within a convolutional layer, the neurons are organized in three dimensions: width, height, and depth. Unlike traditional fully connected layers, where each neuron is connected to all neurons in the preceding layer, convolutional layer neurons are only linked to a subset. This design reduces the number of connections, leading to decreased computation time and fewer weights required for training the algorithm.

The convolutional brain network approach is especially appropriate to the path acknowledgment issue on the grounds that the design of the convolutional layer considers the synchronous assessment of various different picture qualities to recognize a path structure. Moreover, the preparation stage considers the robotized determination and exact setting of this multitude of perspectives through the task of loads, thus advancing decision in muddled conditions. The fundamental path discovery issue is hence improved to a picture binarization, which perceives regardless of whether a pixel has a place with the path.

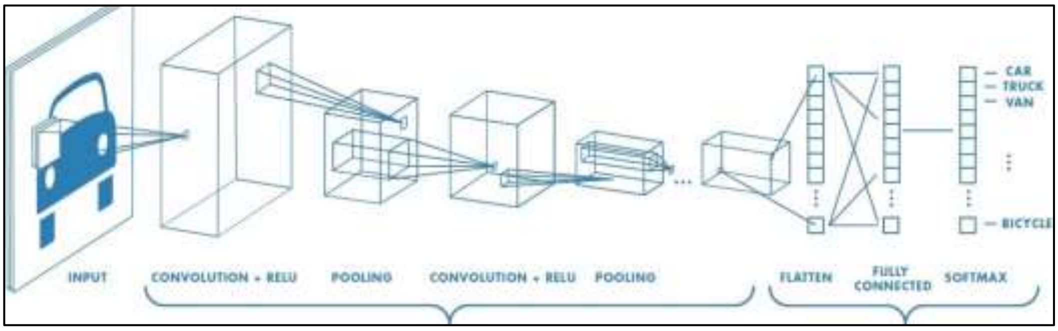

Figure 2.6: Structure of a Generic Convolutional Neural Network.

Several instances of lane detection using convolutional neural networks have been reported in the literature. One of the earliest instances of this approach is provided in [31], which combines a convolutional neural network with a RANSAC algorithm, beginning with an edge detection operation on the input picture. Be that as it may, in this execution, the AI strategy is utilized essentially for picture expansion and, for execution purposes, just when the gathered scene is muddled, including hindrances and path intersections.

End-to-end techniques are used in a new proposal, which means that the convolutional neural network performs the whole identification process beginning from raw pictures and without any previous pre-processing. For instance, in [32], convolutional brain networks are utilized to perceive paths and hindrances on a street. In [33], a start to finish discovery was done both in the front view and in the top view, to coordinate the information and prohibit wrong identifications; this model is named Double View CNN.

Alternate ways have been created to keep away from the particular issues brought up in the past cases. For instance, [34] introduced a Spatial CNN (SCNN) to forestall execution misfortunes brought about by hindrances blocking path markers, copying the human capacity to fill the hole in the impeded region. This point is accomplished by permitting correspondence between neurons inside the equivalent convolutional layer, as opposed to simply between neurons in discrete layers. This extra degree of correspondence considers perfection in the resultant paths, without breaks from obstructions.

Another issue emerges from the need to perceive a few paths in a similar street picture. The primary clear strategy is to design the organization to execute multi-class characterization, with every path relegated to a class. The essential drawback of this method is the prerequisite to execute location on a set number of paths, which lessens framework flexibility. In [35], the recommended engineering is a perform multiple tasks CNN with two branches: division and implanting. The first does double division of the image into two classes: path and scenery. The second, starting with the binarized picture, doles out every path pixel a path distinguishing proof number, dividing the unmistakable paths. In this proposal, profound learning is additionally utilized in the model fitting step. To advance define the resultant path portrayal bend, an opposite point of view planning is utilized. As recently expressed, this sort of change is subject to the presumption that the street is level, and will give erroneous mappings assuming the street has a steady incline. To dodge this trouble, a brain network is prepared to find the poor change boundaries, considering a good viewpoint view in any street conditions.

In [35], a novel approach to lane detection using multi-task Convolutional Neural Networks (CNNs) is presented, offering a comprehensive solution to the lane detection problem. This architecture comprises multiple branches, each dedicated to distinct tasks critical for accurate lane detection. The binary segmentation branches effectively distinguish between lane pixels and background, as well as drivable areas and background. This multi-task CNN framework not only streamlines the lane detection process but also enhances its robustness and accuracy by simultaneously addressing various aspects of lane detection.

Despite the potential benefits of a deep learning-based strategy, there are several issues that must be addressed. The major issue is the intricacy of the convolutional brain network model, which might struggle with the constant elaboration needs of control circle strategies. As a general rule, the quantity of neurons and the sort of tasks they perform on the information (e.g., lattice convolution) bring about an enormous computational expense, which might be diminished by executing the calculation on particular equipment (for instance, a specific GPU) or by easing up the organization. In this situation, there is a compromise between network execution concerning exact identifications and handling time.

Finally, like with any deep learning-based technique, lane detecting systems based on convolutional neural networks require a large quantity of data to be used during the training and testing stages. Furthermore, these pictures must be as diverse as possible, because the system might meet extremely dynamic scenarios and varied road constructions, thus it must be robust in every case.

2.5.3 Combination of Methods

Lane detection has been approached from various different perspectives, ranging from the use of traditional computer vision techniques to end-to-end convolutional or transformer-based solutions amongst others. In addition to these, a number of recent research works attempted to approach lane detection combining different techniques and methods. In this section, some of the most relevant and recent approaches combining different network architectures or techniques will be discussed.

An interesting graph-embedded solution for lane detection was also recently proposed [36]. At first, they use a variation of ResNet [36] as a backbone network, which hierarchically, in four steps, performs semantic segmentation on a road scene image. Thereafter, the extracted feature map from the semantic segmentation network is transformed to the Bird's Eye View (BEV), which refers to a panoramic view of the road scene image, exploiting a computer vision technique named In- verse Perspective mapping (IPM). After transforming the extracted feature maps into the two dimensional BEV representations, the authors introduce a novel graph-based solution as a mean to reduce the dependency on model-specific annotated data or strong lane geometry assumptions. They perform different clustering and sampling techniques on the two dimensionally represented features in order to create an initial graph representation of the extracted lane points. Afterwards, they apply corrections to the initial graph by performing node merging, to reduce redundancy, and reassessing the clustering centers for each detected lane. Following the correction of the constructed graph, the proposed model outputs the polynomial parameters of the lane along with other information like the lane type or viewable distance for each lane. The proposed graph embedded model was compared against various state-of-the-art methods, using the TuSimple dataset [37], and performed similarly to existing state-of-the-art. Discussing the findings of their work, the authors argue that graph embedded learning or using Graph Convolutional Networks (GCNs) [38] could be the next step towards models, not prone to losing important semantic information due to complex road scene environments, with better generalization capabilities.

Another creative approach using a generative adversarial network (GAN) [39] has caught our attention [41]. The proposed pipeline learns to model the data distribution of a road scene image and thereafter generates a representation with distinct semantic entities like road signs or lanes contextually. Evaluation results from the BDD100K [41 and [42] Apollo datasets highlight the efficiency of this approach. Using the adversarial loss, a conditional GAN network (cGAN) can effectively learn and reproduce an input image with distinct annotated semantic entities like the road lanes. However, this is not a trivial task, since as generative models usually have a hard time in learning and maintaining local semantic information from complex images, as those are usually used to create samples closely similar to the learned distributions. In order to tackle this, the authors altered the loss function and structure of the network's discriminator and used the mean square error loss to minimize the difference between the ground truth's representation and the learned feature maps. The proposed model outperformed a number of state-of-the-art segmentation tasked networks using metrics like accuracy and IoU. Another promising finding from this approach is that the authors managed to deliver a well performing model with a considerably smaller number of parameters. This thesis, to the extent of our knowledge, is the only one having used generative models to provide a solution for efficient real-time lane detection and could be used as basis for further research as an alternative to conventional segmentation solutions.

2.5.4 Data Augmentations

Most deep learning models rely on extremely large amounts of data to avoid overfitting and to achieve a reliable and accurate performance. However, in many cases existing datasets are not always guaranteed to include the data diversity or volume of samples necessary for training a model effectively. Data augmentation is a set of techniques tasked to address these limitations by manipulating the original input image data in different ways [43]. Traditional data augmentation techniques are based on either geometric transformations of an image (e.g., cropping, flipping, rotation, scaling etc.), color manipulation (e.g contrast, brightness, saturation etc.), and transforming parts of an image or mixing images together. More recent approaches make use of generative models, like Variational Auto-Encoders (VAE) [44] or Generative Adversarial Networks (GANs) [45], to generate additional synthetic samples based on learned data distributions.

All the aforementioned latest research contributions on deep learning-based lane detection, introduce different model architectures, technical approaches, or propose performance enhancement modules. Nevertheless, only a few studies have attempted to explore data augmentation or other pre-processing techniques for the existing datasets. This observation could indicate a possible research gap in the field.

In a recent research work [46] the authors proposed an interesting data augmentation technique for augmentation of 3D lane detection datasets, which does not contain a good balance between lanes with different height sizes. Such an imbalance could potentially prevent models from effectively learning to detect the lane instances in uphill or downhill scenarios [47]. In their work, the authors approached the lane detection task as a 2D to 3D reconstruction problem, leveraging the geometrical information between the two representations of a road scene. Using an encoder/decoder architecture, they transform the input 3D RGB images into 2D top-view lane masks. They also introduce a novel loss function, called geometry prior loss, which they use to train the 3D lane reconstruction regression net- work. The proposed loss function is built to optimize the preservation of the properties of the original 3D shape of the lane during the reconstruction from the extracted 2D masks process. Additionally, the authors highlight that conventional augmentation methods like translation or center rotation could alter the geometric properties of the 3D lanes. Therefore, such augmentations would generate invalid data which would reduce the reliability of their proposed mechanism. Instead, the proposed augmentation involves roll, pitch and yaw rotations of the 3D lanes.

Their model was tested with and without the proposed augmentation on the Apollo 3D synthetic dataset [48] with promising results against similar state-of-the-art approaches. The experiment consisting the augmented data yielded the best results, highlighting the effectiveness of their method. Additionally, the authors argue in favor of their method’s effectiveness in detecting the lane instances at extra-long distances, after testing the model with an extra-long range lanes data split from the same dataset.

A dimensionality reduction-based technique, named Fast-ICA, was recently proposed as an alternative to traditional data augmentation methods [49]. The proposed method was based on a statistical method called Independent Component Analysis (ICA), a dimensionality reduction statistical method used to represent multivariate data as a set of independent components. Fast-ICA outperformed different geometrical transformation-based data augmentation techniques, for the lane detection task, as the method was tested using the mean percent error metric (MPE). The MPE metric indicates the average percentage of the difference between predicted and actual values [50]. The proposed Fast-ICA method yielded an impressive 42.2 % increase in performance of a model trained with data making use of it, in comparison to the original and data augmented dataset. However, it has not been tested using images with complex weather and road scene conditions.

2.6 RESEARCH GAPS

The identified research gaps center around two fundamental challenges in lane detection systems for modern vehicles. Firstly, there's a need to address the extraction of spatial-temporal information effectively and the capture of relevant features over time. Traditional methods often struggle to adapt to dynamic changes in the environment, such as varying road conditions and vehicle movements. Thus, there's a gap in understanding how to integrate temporal context into the lane detection process to improve accuracy and robustness. Secondly, false positive lane detections remain a significant issue, particularly in complex or cluttered road scenes. Existing algorithms may produce erroneous detections due to factors like shadows, road markings, or other objects resembling lanes. Bridging this gap involves exploring techniques to mitigate false positives, such as refining feature extraction methods, enhancing model learning capabilities, or incorporating contextual information to better distinguish true lane markings from noise or artifacts. Addressing these research gaps is crucial for advancing lane detection technology and enhancing the safety and reliability of autonomous driving systems.

2.7 CONCLUSION

Despite the potential advantages offered by a deep learning-driven approach, several challenges need to be addressed. This computational overhead can be mitigated by employing specialized hardware, like dedicated GPUs, or by optimizing the network architecture. However, this optimization introduces a trade-off between detection accuracy and processing speed. Moreover, similar to other deep learning-based methodologies, CNN-based lane detection systems necessitate extensive datasets for effective training and testing. Furthermore, these pictures must be as diverse as possible, because the system might meet extremely dynamic scenarios and varied road constructions, thus it must be robust in every case.

3. PROPOSED SYSTEM

Lane detection is crucial in autonomous driving because lanes can serve as important cues for restricting vehicle maneuvering on roads. The goal in autonomous driving is to gain a complete understanding of the environment around the car by employing a variety of sensors and control modules. Camera-based Lane recognition is a significant step toward such environmental awareness because it helps the car to align itself correctly inside the road lanes.

3.1 TECHNICAL APPROACH

In deep learning, a pipeline typically refers to a sequence of models or processing steps that are applied one after the other to accomplish a specific task. A pipeline is typically comprised of various different modules, each designed to perform a specific sub-task that contributes to the overall goal of the pipeline. For our hybrid model approach , the developed pipeline consists of three different neural models, a CNN backbone network. Finally, an additional module for post-processing using temporal information is also introduced as the last module of our lane detection pipeline. Each unique module of our pipeline has a specific role in the pipeline and is explained in detail, in this section of the report, along with argumentation for our design choices. Additionally, the dataset used for our experiment along with the necessary pre-processing steps are presented.

3.2 METHODOLOGY

3.2.1 Dataset and Pre-Processing

For our experimentation we have chosen to use the TUSimple dataset (Tomatosliu et al., 2020). As previously mentioned in the Scientific Method section, TUSimple is one of the most widely used datasets for the lane detection task. The full dataset consists of 6408 images, divided in approximately 320 one- second long clips of 20 frames each. The original resolution for each frame image of the clip is 1280x720 pixels. For every clip, only the 20th frame comes with annotated lane markings for a non-standard number of lanes (2 to 5 lanes). According to the creators of the dataset, the clips were recorded under good or mediocre weather conditions, during different daytime and traffic conditions. Moreover, the dataset is already split into test and training sets by the creators. Out of the total 6408 clips, 3626 are given as the train and validation split whereas 2782 are provided for testing purposes.

The lane annotations for all the labeled frames are provided in JSON format as one pixel thick poly- lines, comprised of X and Y Cartesian coordinates for each lane present. As a poly-line, we refer to a continuous line composed of piece-wise straight line segments. In order for these to be useful inputs to any deep learning segmentation model, additional pre-processing is required. The first step, in order to train a segmentation model, is to generate the binary segmentation masks from the raw annotations. A binary segmentation mask is used as the target output for a deep learning model during training or testing and refers to a black and white or grayscale image of the same dimensionality as the input image. The black pixels refer to the background class, whereas the white pixels refer to the target class, which in our case was the road lane.

However, as the annotations are provided as one pixel thick poly-lines , further thickening of the poly- lines was necessary. In their repository, the creators provide the community with a pre-built evaluation method where they state that any predicted pixel within 20 pixels radius from the original annotated points is considered as an accurate prediction. Therefore, a thickening value of 5 pixels in each dimension was selected, as there does not seem to be a generally preferred value for this present in the literature. This choice was made by us after experimentations during the pre-processing, as a thickening value of five pixels in each dimension seemed to generate the most realistic and accurate ground truth lane markings. Even though an even thicker poly-line could potentially increase our model's performance, we made this design choice, so our model's evaluation metrics comparison with other models is valid. Additionally, we decided to resize the input images and binary masks to a fixed 448x448 size for computational reasons and compatibility with the filters of our ViT module.

In order to identify the new coordinates of the lane ground truth points in the resized masks, we firstly calculate the scaling factor for height and width respectively according to:

$$Scaling\ Factor\ (SF) = \frac{new\ dimension\ size}{old\ dimension\ size} \tag{3.1}$$

Consequently, each original lane point's coordinates are re-calculated for the new image size as:

$$\text{New x} = \frac{x}{\text{Width SF}} \tag{3.2}$$

$$\text{New y} = \frac{y}{\text{Width SF}} \tag{3.3}$$

As already mentioned, the input images are also resized to a fixed 448x448 size using bicubic interpolation, a commonly used technique for image resizing . Following the generation of the segmentation masks and image and ground truths resizing, both are converted to PyTorch tensors in order to be compatible with our model's framework. Furthermore, in our test set generator, we also include the option to add a user-given number of previous frames along with the annotated one. These previous frames are later used by our temporal post-processing mechanism. In Figure 3.1, a sample of our resized images and pre-processed ground truths is depicted.

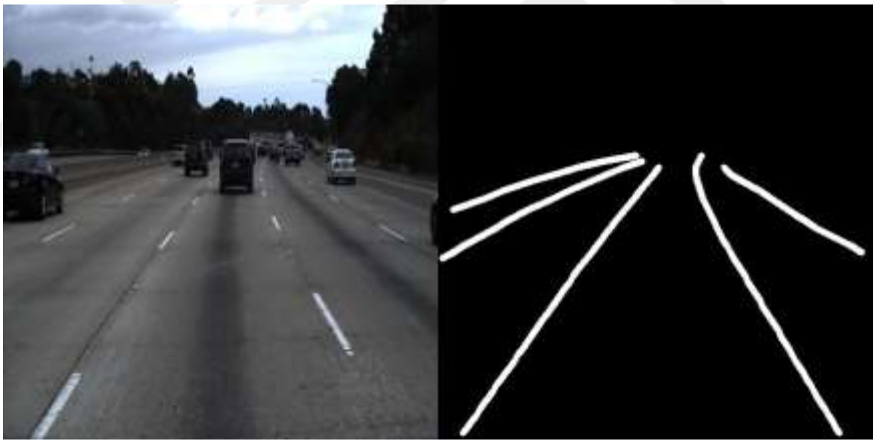

Figure 3.1: Resized Input Image and Respective Ground Truth With Poly-Line Markings.

When training deep learning models, a pivotal step is data augmentation. Applying data augmentations is proven to increase the learning capabilities of deep learning models and increase generalization on unseen data by increasing the diversity of the training data. Some augmentations (e.g., rotations, flip) need to be applied both to the ground truths and input images, while others are only applied to the input images (e.g., blurring, color jittering). The input images used for training of both our backbone CNN network and the rest of our pipeline's modules were augmented using the following augmentations:

i. Random rotations between 10 and 30 degrees
ii. Random horizontal flipping
iii. Gaussian blurring
iv. Color jittering

On the one hand, gaussian blurring is an augmentation technique which blurs an image according to a Gaussian function. Color jittering, on the other hand, is used to alter the original image's brightness, contrast, hue and saturation. By applying these augmentations, more variance is introduced in our training data, resulting in better feature extraction performance.

3.2.2 Image Segmentation

After applying the pre-processing techniques on the TUSimple dataset, we incorporate the SegNet back- bone network. SegNet is a CNN model designed for semantic segmentation of images, whose goal is to assign a class label to each pixel in the input image. As previously mentioned. Their model consists of an encoder and a decoder, with skip connections between them. Typically, the input data is transformed into a higher-dimensional representation by the encoder block, sequentially and repeatedly from the previous layer while down-sampling in size. Once the input data has been processed by the encoder, the resulting highest-dimensional representation is sent to the decoder that performs the reverse process. This output of the encoder, which is way smaller in size than the original input, is in turn sequentially and repeatedly reduced, by the decoder block, to lower-dimensions and up-sampled to the original input size.

The encoder extracts high-level features from the input image, and the decoder uses these features to produce a pixel-wise segmentation map. The skip connections are used to connect corresponding encoder and decoder layers, enabling the decoder to use higher resolution features to refine the segmentation map. What's more, the encoder comprises of pooling layers, for instance max-pooling. Max-pooling is a pooling activity that separates the info picture into non-covering rectangular districts and taking the most extreme worth inside every locale. This cycle decreases the spatial size of the contribution, while holding the main data.

There is an existing issue where spatial information is always lost in an image auto-encoder network during down-sampling in the encoder (via max-pooling). In order to mitigate this problem, the novelty the SegNet is offering is that it utilizes a modified form of max-pooling called max-pooling with indices to preserve the location information that is lost in traditional max-pooling. The max-pooling indices are stored during the encoder stage and used in the corresponding up-sampling layers in the decoder stage to produce a more accurate segmentation map.

In order to answer the first research question, at first, we do train the SegNet separately for the road lane segmentation task, evaluate it on the TUSimple test set, save the pre-trained model and then use the pre- trained model with frozen weights to train the rest of the pipeline. This choice was made due to the need for reasonable inputs into our ViT module for it to be effectively trained, as training the pipeline in an end-to-end fashion did not seem to be realistic according to the resources we had at hand for this work. Our pipeline's next module is the ViT network which is fed a 64 channel feature map with identical spatial dimensions as the input image. These feature maps are the output of the last decoder block of our SegNet model. A visual illustration of the SegNet architecture is depicted in Figure 3.2 below.

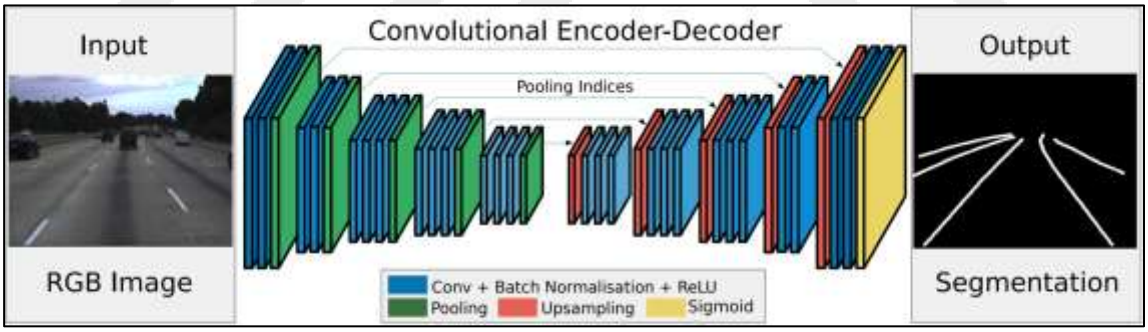

Figure 3.2: Our SegNet Model's Architecture.

3.2.3 ViT Module and Feature Transformation

After passing the input images from our backbone SegNet model and extracting the respective 64 channel feature maps, the next module in our pipeline is a Vision Transformer (ViT). ViT models are typically used for image classification and usually require large volumes of data to be effective in this task. However, recent works have shown that with a few configurations in the architecture, a ViT model can be trained or fine-tuned for segmentation tasks. Also, another reason we moved forward with the CNN feature extraction backbone selection was to mitigate the specific training requirement of a ViT for large data volume, large training batch sizes and time.

When an image is passed through a ViT model, the first procedure that takes place is splitting the original image into non-overlapping patches of a given size (16x16 in our case), with a sliding window approach. As the input feature maps are of shape 64x448x448, we end up with a patch grid of shape 28x28, and a total of 784 patches per channel. A visualized example of an input 64 channel feature map is depicted in figure 9. Each patch then is linearly projected, using a 2D Convolutional kernel, into a single channel vector representation called a patch embedding. Another learnable parameter, called a positional embedding, is also added to each of the resulting patch embeddings to provide spatial information about the relative position of each patch in the original image. Finally, an extra learnable parameter called class embedding is also added to the patch embeddings and it serves as a learnable representation of the class labels.

Consequently, the patch embeddings with positional information are then passed through a number of transformer layers. The number of transformer layers depends on the specific ViT variant used. In each transformer layer, a series of multi-head attention and feed-forward operations take place, for hierarchical feature extraction or transformation. The total number of transformer layers used, form the transformer encoder of the ViT architecture. The multi-head attention modules of each transformer block in the network allow the model to capture global correlations between features, unlike the CNN architectures which use local kernels to extract information. Finally, the output patch encodings from the transformer encoder, can be fed into a prediction head for downstream tasks like segmentation or image classification.

To summarize, we use the extracted feature maps from our SegNet backbone as inputs for this module and refine the extracted feature maps by exploiting the benefits of the ViT architecture. Our ViT module’s architecture is depicted in Figure 3.3 . In our implementation, we decided to use the smallest ViT variant (ViT-Ti). We have chosen this architecture as transformers are typically very demanding to train in terms of computational power and training time.

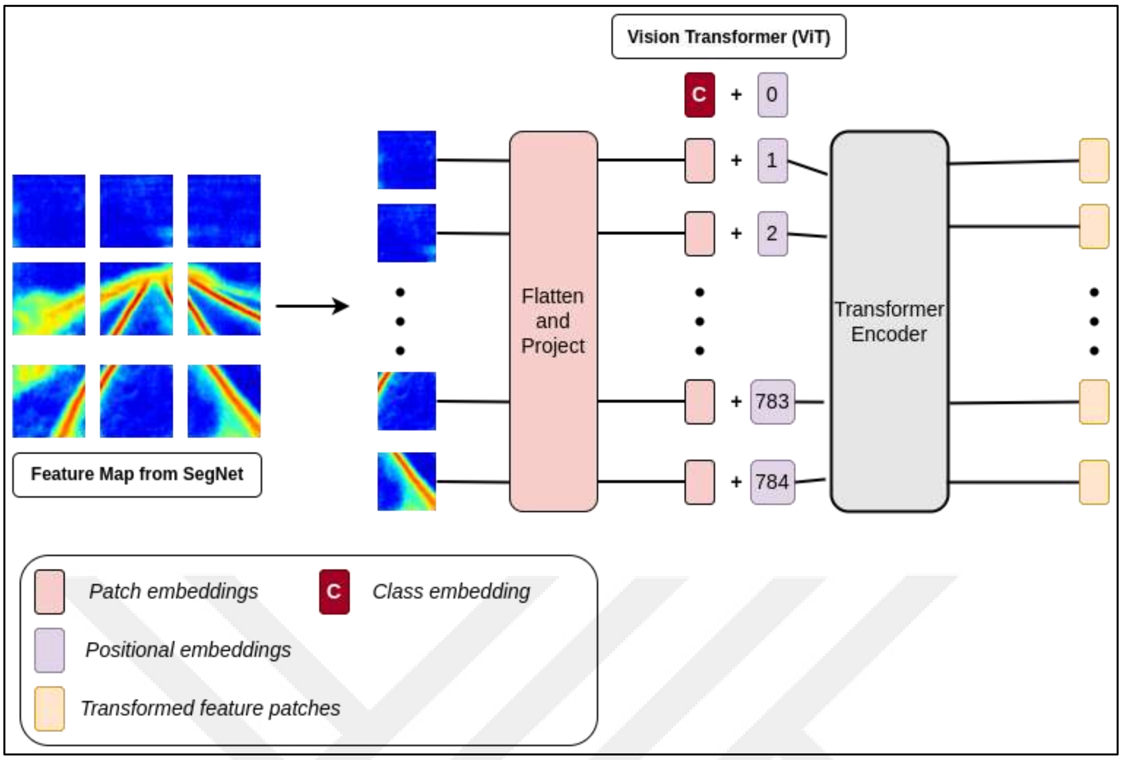

Figure 3.3: Our Vision Transformer (ViT) Module's Architecture.

3.2.4 Canny Edge Detection

After feature transformation, in order to extract the lanes of an image we need to detect their edges. One of the most popular edge detection methods is the Canny edge detector. It is a multi-stage algorithm, and it is one of the best methods for detecting consistent edges. The algorithm can be broken down into five steps.

3.2.5 Non-Maximum Suppression

Lower bound cut-off suppression is a technique employed to identify areas exhibiting the most significant changes in pixel values. This method assesses the edge strength of the current pixel in relation to adjacent pixels in both positive and negative gradient directions. This process enables the identification and preservation of pixel values indicative of strong edges while suppressing those associated with weaker or less significant edges, thereby enhancing the clarity and accuracy of edge detection algorithms.

3.2.6 Double Threshold

Non-maximum suppression gets a much more adequate detection of actual edges in an image. However, noise and color variance continue to affect certain edge pixels. Selecting high and low threshold values achieves the solution. In edge detection processes, pixels are classified based on their gradient values relative to predefined threshold values. Pixels with gradient values lower than the low threshold are designated as suppressed. This classification system aids in distinguishing between strong, weak, and suppressed edge pixels, facilitating precise edge detection and subsequent image analysis tasks.

3.2.7 Hysteresis Thresholding

Even after suspending the weak pixels two times in the row, variance and noise may still affect the determination if an edge exists. A weak edge pixel resulting from weak edges is usually connected to a strong edge pixel, whereas noise responses are not. By admitting this reality, Blob analysis then is used to track the edge connection by looking at a weak edge pixel and its eight connected neighbor pixels. There will be certainly an edge product as long as there exists a strong value pixel. Figure 3.5 illustrates the hysteresis thresholding.

3.3 HOUGH TRANSFORM

The Hough transform is a technique used to extract features. The technique's primary objective is to use a voting procedure to find imperfect instances of objects within a given class of shapes. The lane on the road is an imperfect instance of a shape in our case. The edge detector generates points, which should be used to obtain lines. The Hough transform is used in many model cases. The simplest one is the detection of straight lines. To begin with we need to model our shape in a set of parameters.

The linear Hough transform algorithm is used to estimate the two straight-line parameters. Each point in the transform space serves as an accumulator to detect or classify a line described by this, and the transform space has two dimensions. At each step, the algorithm iterates over all of the detected edges in the image, adding to the accumulator. The most probable lines can be obtained and their parameters can be collected by finding the containers with the highest values in the accumulator space. Depending on the nature of the problem an optimized voting process may be utilized, however, we usually search for maxima in the accumulator space. The final result of the transform is the parameters that most likely form a line on the corresponding edge pixels.

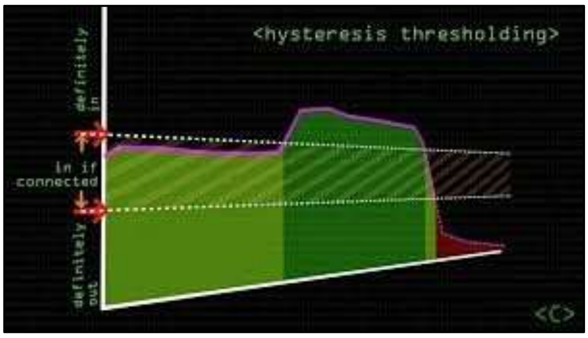

Figure 3.4: Hysteresis Thresholding.

3.4 DEEP LEARNING MODELS

3.4.1 Convolutional Neural Networks

Deep learning stands as a transformative force across various scientific and technological domains, showcasing its prowess through applications ranging from natural language processing to image recognition. In CNNs, layers comprising multiple 2D convolutions operate on input data, with trainable parameters encapsulated in filter kernels. This amalgamation of convolutional operations and non-linear activations empowers deep CNNs to excel in tasks spanning image analysis, natural language understanding, and beyond, thereby solidifying their status as indispensable tools in modern computational paradigms.

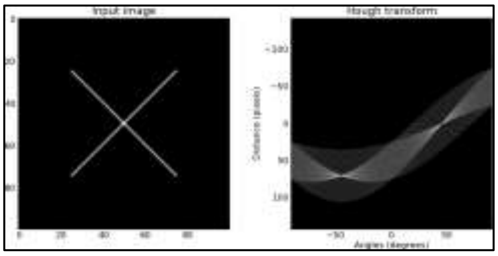

Figure 3.5: Hough Transform Accumulator Space.

3.4.2 UNet Autoencoder

Convolutional networks are often used for classification operations, where the output to an image is a single class mark. However, in many visual tasks, especially image segmentation, the desired output should provide localization, i.e., each pixel should be given a class label. Some of the most recent approaches to image segmentation have attempted to apply deep algorithms optimized for class prediction to pixel-wise labeling like. The findings, although promising, are not conclusive. This is large since maximum pooling and sub-sampling limits the feature map resolution. The U-Net architecture includes an encoder for context-capturing and a symmetric decoder for approximating the location. The high-resolution functions from the encoder are linked to the upsampled output for localization. Each encoder's layer has 2 3x3 convolutions and a 2x2 max pooling layer for downsampling accompanied by a ReLu activation function. After 2 downsamplings, the number of function channels is doubled. Similarly, each layer of the decoder, that upsamples the feature map, consists of an up convolution 2x2 layer and then two 3 × 3 convolutions with a ReLu layer. After each upsampling, the number of functional channels is halved. Skip connections are formed between the encoder and decoder by concatenating the layers with the same number of features to propagate high-resolution features to the decoder, allowing precise localization. U-Net architecture is illustrated in Figure 3.7.

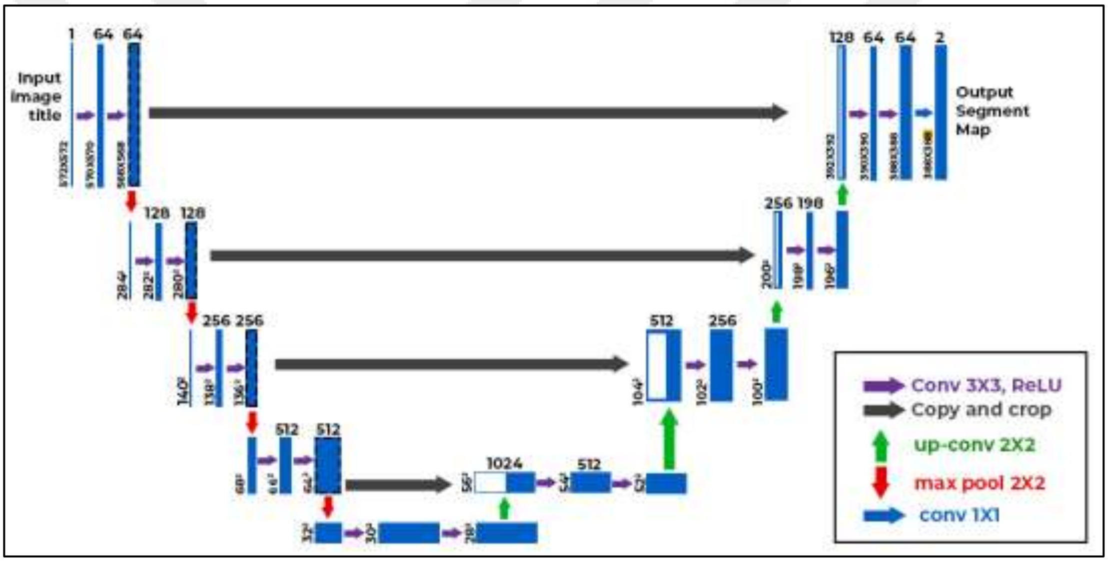

Figure 3.6: U-Net Architecture.

3.4.3 ResNet Model

A residual neural network (ResNet) represents a unique architecture within the realm of neural networks, distinguished by its utilization of skip connections or shortcuts to establish direct connections between layers. At the heart of ResNet lies the skip connection block, depicted in Figure 4, which serves as the fundamental building block of the network. Figure 3.8 visually illustrates the architectural framework of the residual model, showcasing the interconnected layers and the pathways facilitated by skip connections.

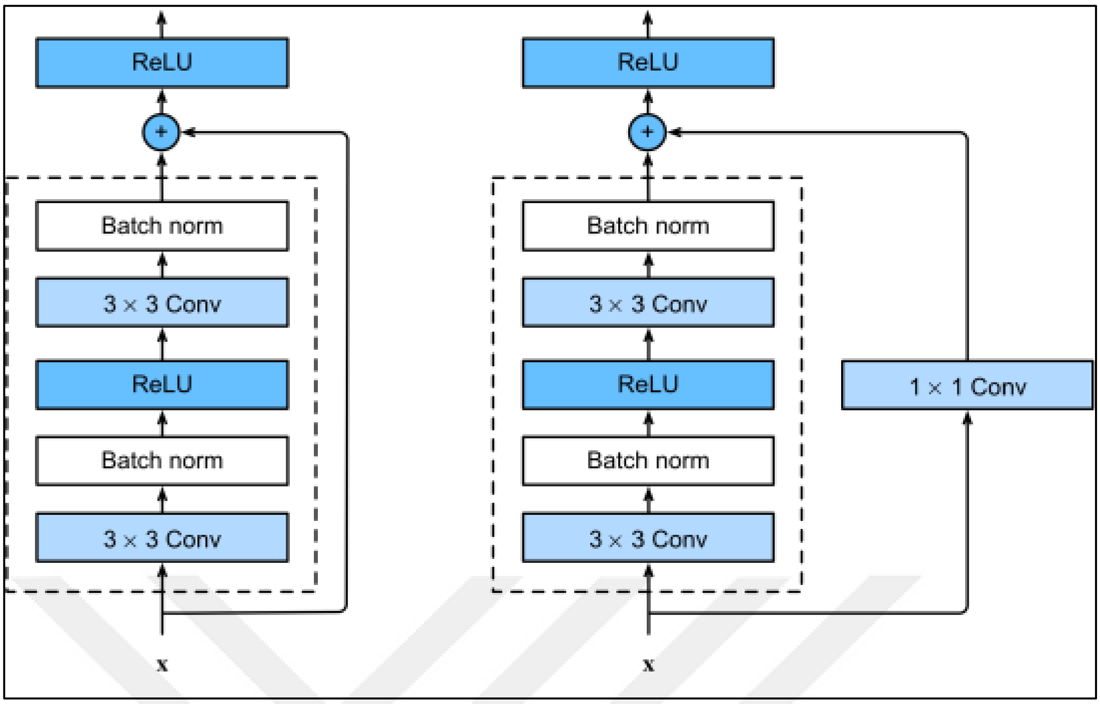

Figure 3.7: ResNet Convolution Block.

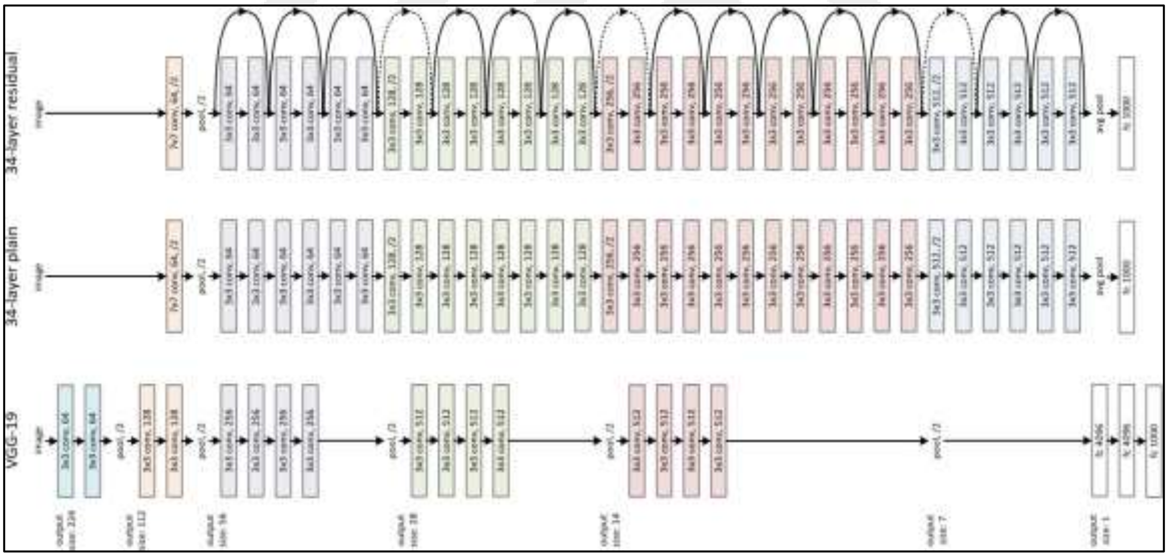

Figure 3.8: ResNet Architecture.

3.4.4 EfficientNet

Convolutional neural networks are typically designed with a limited resource budget and then scaled up for greater precision as more resources are available. EfficinetNet investigates model scaling in detail and discovers that carefully matching net- work depth, width, and resolution will contribute to improved results. Figure 6 illustrates the scaling of the network. It employs neural architecture search to develop a new base- line network and scale it up to produce a family of models known as EfficientNets, which outperform previous ConvNets in terms of precision and performance. On ImageNet and five commonly used transfer learning datasets, the EfficientNet model can be scaled up very efficiently, surpassing state-of-the-art Precision With An Order of Magnitude Fewer Parameters and FLOPS.

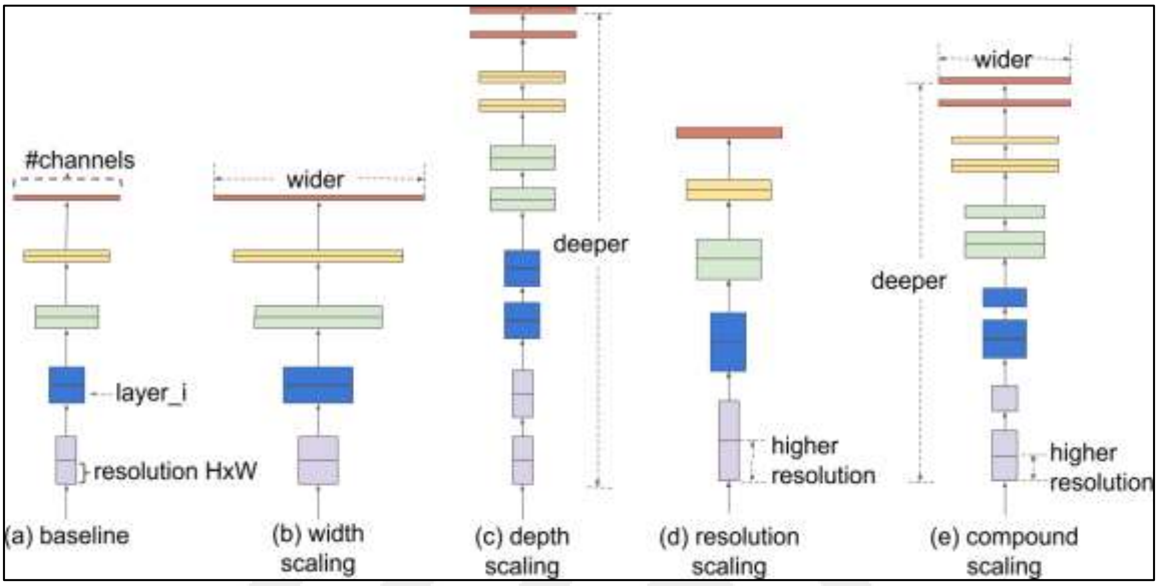

Figure 3.9: The Ways of Scaling A Model.

3.4.5 MobileNet

Convolutional neural networks advancements increase performance but do not actually make networks more effective in terms of size and time. Many real-world applications, such as self- driving cars, need processes to be completed quickly with limited resources. The MobileNet model is designed to be lightweight and suitable for use in smartphone applications. Profundity wise distinct convolutions are utilized by MobileNet. When contrasted with the organization with normal convolutions of similar profundity in the nets, it fundamentally diminishes the quantity of boundaries. We note that the channel used to have 9 boundaries, however presently it simply has 6. This was made conceivable by isolating the level and width estimations. A similar idea can likewise be utilized to isolate the profundity aspect from convolutions. Following that, we apply a 1*1 channel to cover the profundity aspect.

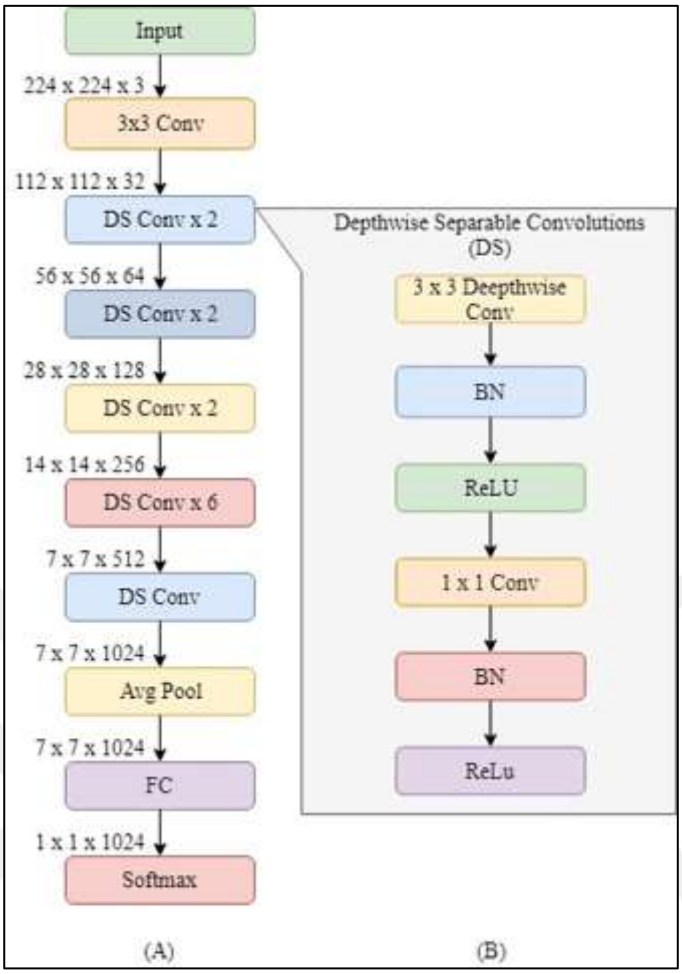

Figure 3.10: The Ways of Scaling A Model.

3.5 CONCLUSION

In this thesis, we addressed the problem of lane recognition on the road, which involves autonomous vehicles. We attempted to solve it using traditional techniques such as edge recognition and Hough transforms, as well as modern techniques such as Convolutional Neural Networks. We compared the methods using images of their predictions, and the results were very satisfactory, as we discovered that even with simple models, we could detect lanes on the road. There is also a clear distinction between classical and modern methods, with the latter producing significantly better results than the former

4. RESULTS AND DISCUSSION

4.1 DATASET

The experiments utilize the TuSimple dataset, accessible at https://github.com/TuSimple/tusimple-benchmark, which comprises 6,408 road images captured from US highways. Each image has a resolution of 1280x720 pixels. The dataset is partitioned into three subsets: 3,626 images for training, 358 for validation, and 2,782 for testing. These images are collected under diverse weather conditions, providing a comprehensive representation of real-world scenarios encountered on highways.

4.2 IMPLEMENTATION

During preparing, the pictures' goal was downscaled to 128x128, and for profound learning techniques, the quantity of variety channels was diminished from 3 to 1 because of equipment limitations. The training process took place within the Jupyter Notebook Editor, utilizing CPU runtime. Consistency was maintained across all models with identical parameters, including a learning rate set at 10−4. Batches were configured at a size of 32, and the models underwent training for 15 epochs, each iteration requiring approximately 7 hours. Throughout the training sessions, the Adam optimizer was employed alongside Binary Cross Entropy loss function.

4.3 RESULTS

4.3.1 Image Segmentation Using SegNet

As mentioned in the previous section, the first process of the pipeline is the separate training of the CNN backbone model (SegNet). Following the training of our model, we test its capabilities with different configurations, regarding the use of our post-processing mechanism. This is connected to the answer to our second research question. In this section, our model's performance results, with and without the use of our post-processing temporal mechanism, are analyzed and presented in detail. The different metrics from our SegNet model are depicted in Table 4.1. Regarding the use of our post- processing temporal mechanism, we try with a number of previous frames, ranging from 3 to 5 frames, and we obtain the respective metrics from our test set evaluation. It is worth noting that each frame has 100 ms temporal distance with the next, something that denotes that using 5 previous frames refers to 0.5 seconds into the past. The appropriate temporal difference between base and the last previous frame for road lane detection can vary depending on the specific characteristics of the scene and the requirements of the application. Thus, we believe that any time between 0.3 and 0.5 seconds is a reasonable amount of time to be used for previous frames. It could be the case that 0.5 seconds of temporal context can be sufficient to distinguish a turn or improve the visibility of the lanes in the presence of moving cars

Table 4.1: The different Metrics Obtained from the Segnet Backbone Model

	Without Temporal	**With Temporal**		
		3 frames	**4 frames**	**5 frames**
FPR	0.0706	0.0644	0.0636	**0.0631**
FNR	0.0524	0.0509	0.0503	**0.05**
F1 Score	0.493	0.515	0.518	**0.52**
IoU score	0.331	0.351	0.353	**0.355**
Accuracy (%)	93.0	93.6	93.7	**93.7**
FPS	**264.5**	34.8	28.4	23.9

The following images depict different random images from the TuSimple test set along with their binary ground-truth masks, the predictions from our backbone model without the temporal post-processing mechanism and finally the predicted masks exploiting our proposed temporal mechanism. Noteworthy, a total of five previous frames were used to generate these specific results.

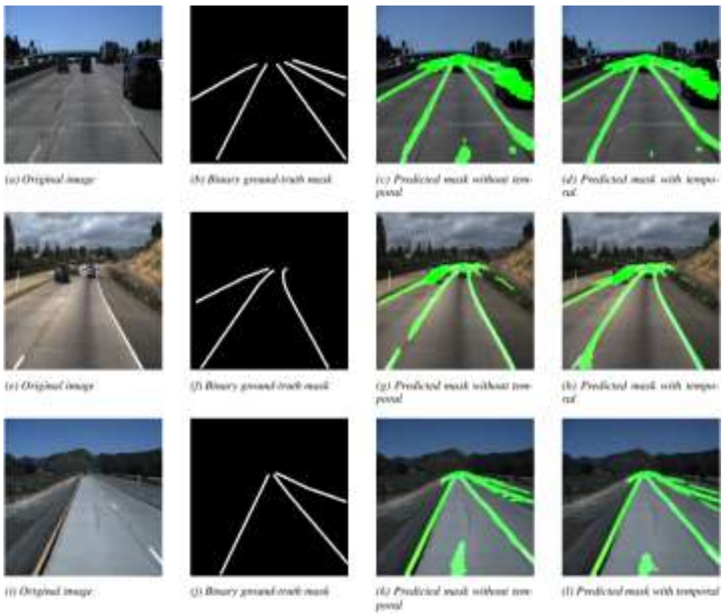

Figure 4.1: Lane Detection Results for Some Random Images from The TuSimple Test Set.

4.3.2 Classic Lane Detection

Upon inspecting the images, it becomes evident that the performance of the classic line detector without averaging the line calls leaves much to be desired. Figure 8 illustrates the resultant lines, which are characterized by their fragmented nature due to the presence of numerous individual lines. These disparate lines ultimately coalesce into two imaginary lines, representing the delineation of the road. Notably, this effect is confined to the designated area of line detection, typically a triangular region extending from the image's forefront. However, despite this localized focus, the method's accuracy appears compromised when confronted with diverse dynamic environments. However, when the brightness in the image is increased, these different threshold values are not enough to separate the road lines. This can be seen in Figure 4.2, where the brightness is so high that determining the best threshold pair makes little sense.

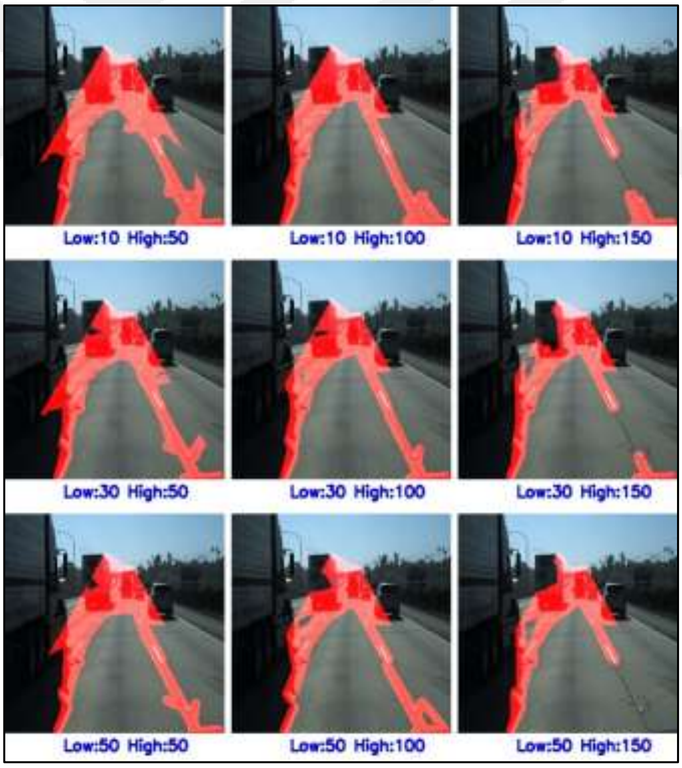

Figure 4.2: Classic Lane Detection Without Averaging the Slopes of the Detected Lines**.**

4.3.3 Average Slopes

To mitigate this issue to some extent, an approach involves computing the average of the slopes of detected lines, resulting in the creation of only two representative lines on the road ahead of the driver. Figure 8 displays improved results, with lines accurately delineating the road at locations typically traversed by the driver. Nonetheless, challenges persist, particularly in regions where the asphalt exhibits unique characteristics, rendering accurate forecasting problematic. Furthermore, Figure 4.3 accentuates another challenge encountered in the detection process, particularly evident during turns where suboptimal results are frequently observed across various threshold settings. This inconsistency poses a significant hurdle in determining an optimal parameter selection strategy. Consequently, a departure from traditional methods reliant on manual parameter tuning is warranted. Instead, an adaptive approach leveraging automated parameter adjustment mechanisms based on street image characteristics emerges as a viable alternative.

Figure 4.3: Classic Lane Detection with Average slopes.

4.3.4 Convolutional Neural Networks

Upon transitioning to the machine learning methods depicted in Figure 4.4, notable improvements in performance are readily apparent. In this figure, the results of all four distinct machine learning models are presented simultaneously. Additionally, the training curve can be observed in Figure 4.5. Eminently, it is obvious that, aside from the most straightforward model with the least boundaries, the excess models show an exceptional level of improvement, accomplishing essentially upgraded execution levels.

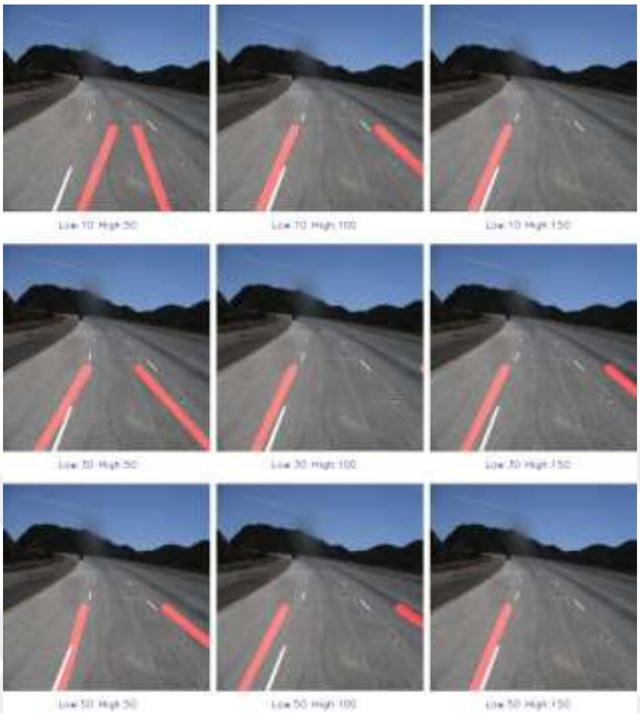

Figure 4.4: Classic Lane Detection With Average Slopes Missing The Curve

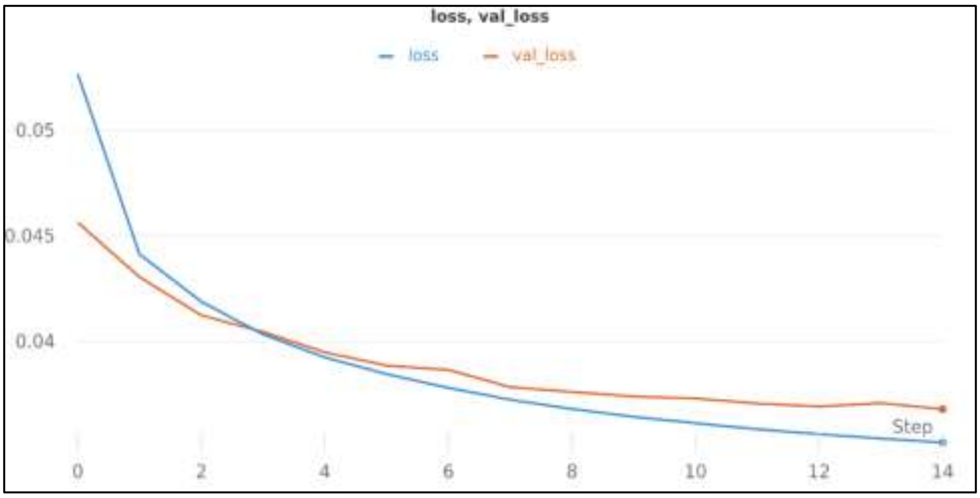

Figure 4.5: CNN Binary Cross Entropy Validation Error of Training.

4.3.5 Simple Autoencoder

In the simplest model, which lacks the residual connections and the structure of the U-Net, the results in the straight line are very good and even lines of the other strips are detected, which is particularly useful. However in some cases like Figure 4.6 the prediction is noisy.

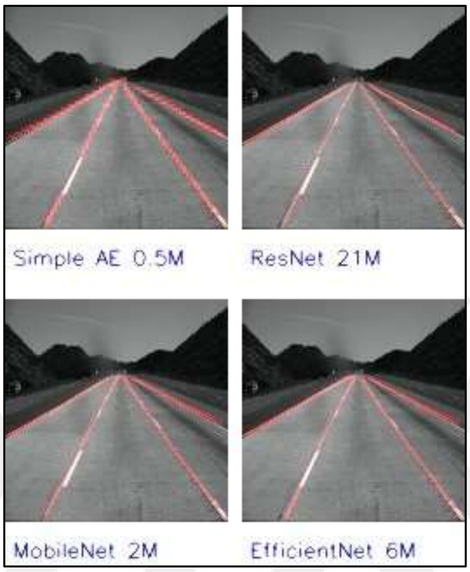

Figure 4.6: Autoencoder Models in Straight Lanes.

We have 4 images in the machine learning lane detection results. These are the 4 Profound Convolutional Organization models that we tried. The quantity of boundaries that the model has is demonstrated toward the finish of their name.

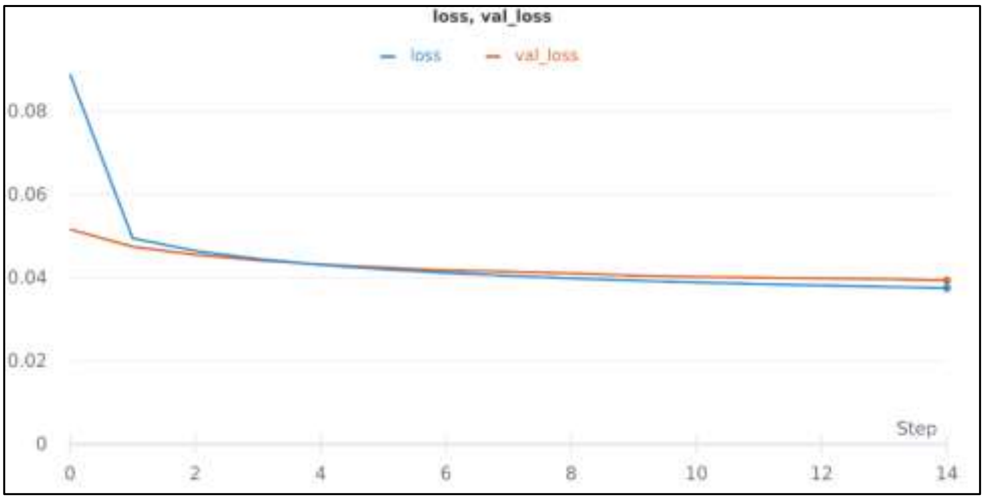

Figure 4.7: Simple Autoencoder Binary Cross Entropy Validation Error of Training.

4.3.6 ResNet Model

The ResNet model, having the largest set of parameters has the best performance of all the other models observing the characteristic forecast images. However, its superiority is not sufficiently justified by the number of its parameters as the other models achieve almost similar results with orders of magnitude smaller parameters.

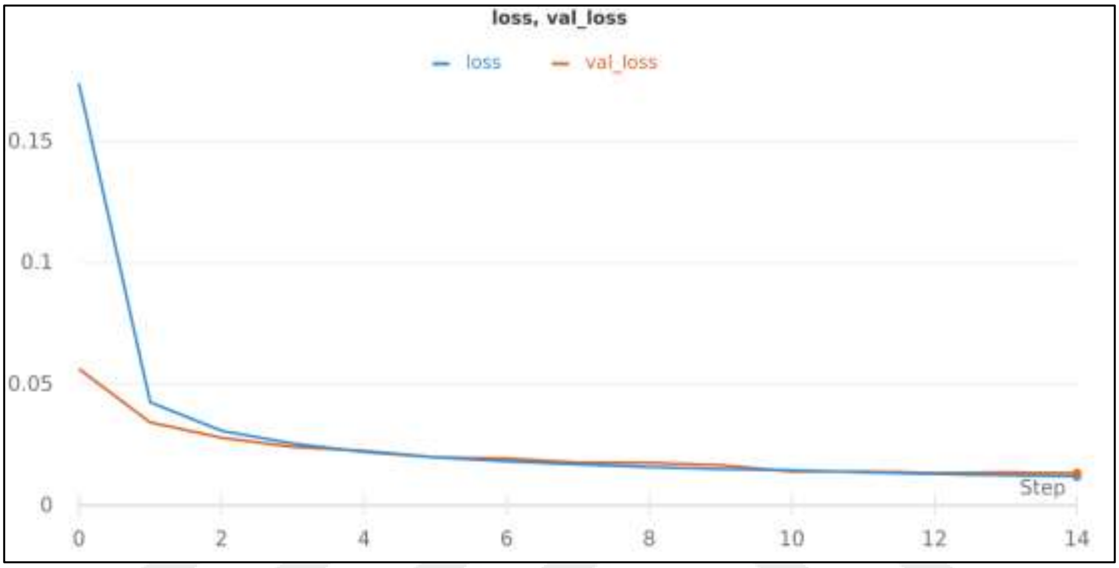

Figure 4.8: ResNet Binary Cross Entropy Validation Error of Training.

4.3.7 EfficientNet

The EfficientNet having fewer parameters but a more sophisticated combination of its architecture achieves particularly good results even in Figure 4.9 where it predicts the line in this difficult case among all the other models.

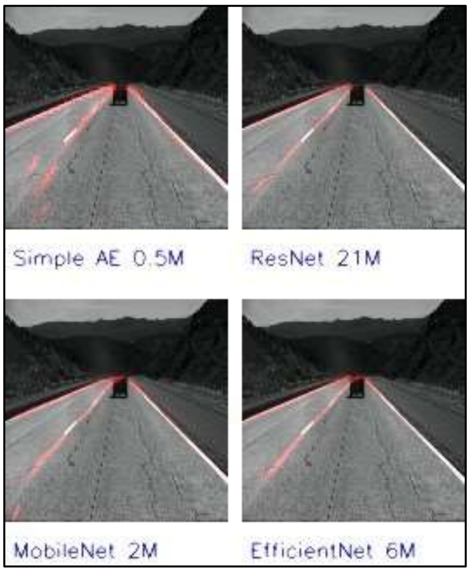

Figure 4.9: Autoencoder Models Having Noisy Prediction

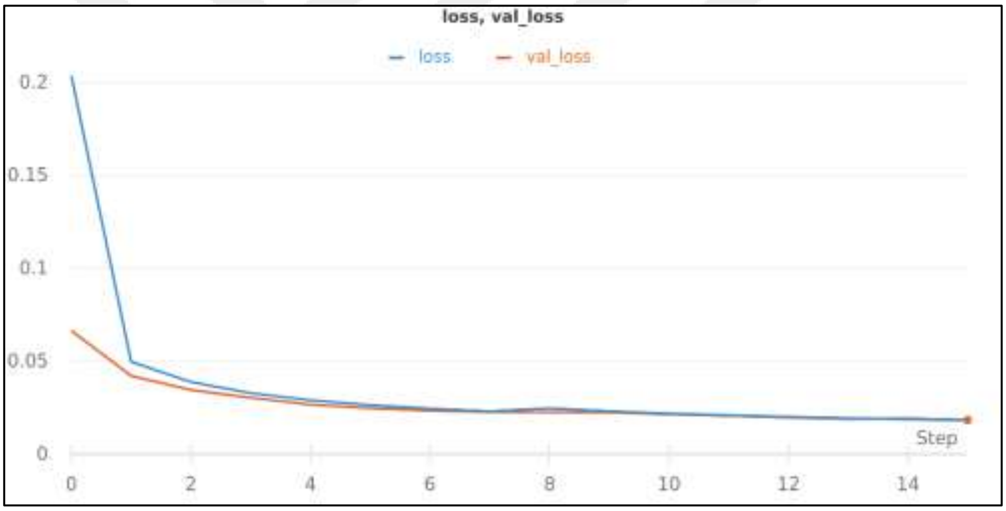

Figure 4.10: EfficientNet Cross Entropy Validation Error of Training.

4.3.8 MobileNet

Finally, MobileNet, having the smallest number of parameters has particularly good results. With 10 times fewer parameters than ResNet, its training is easier and the prediction of lines can be done from a small computer device. Even in difficult examples such as Figure 4.11or Figure 4.12, MobileNet can give quite good predictions.

These four images encapsulate the outcomes of our testing and analysis, offering insights into the efficacy and performance of these diverse CNN architectures in the context of lane detection tasks. Through this comparative assessment, we aim to elucidate the relationship between network complexity, parameter count, and the resulting accuracy and robustness of lane detection models.

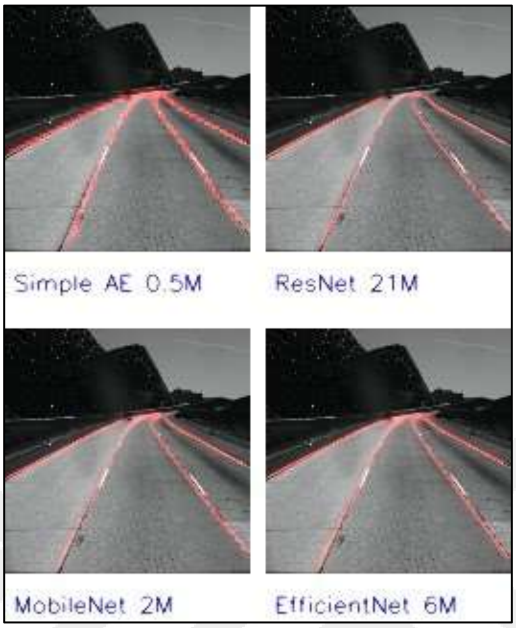

Figure 4.11: Autoencoder models detect curved lanes

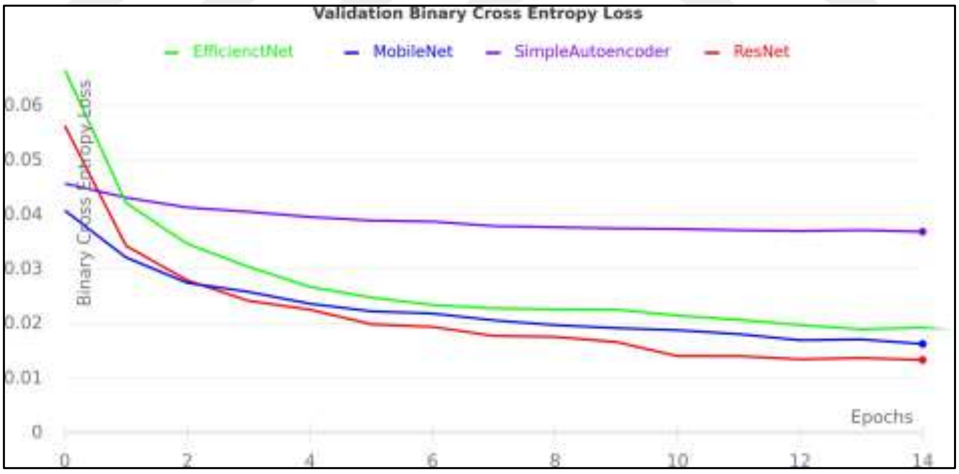

Figure 4.12: Binary Cross Entropy Validation Error of Training.

4.4 DISCUSSION

When comparing the outcomes of the two methodologies, it becomes evident that Convolutional Neural Networks (CNNs) exhibit superior performance in reliably predicting lane lines amidst varying environmental conditions and noise levels. However, it's worth noting that classical learning models could have yielded better predictions if supplemented with statistical techniques such as particle or Kalman filters. These approaches offer greater consistency and transparency, as they provide a clear understanding of the algorithm's decision-making process at each point. In contrast, CNNs often operate as black boxes, making it challenging to decipher their inner workings. Despite the computational resources required for training CNNs, recent advancements have led to the development of more efficient architectures, reducing both time and data requirements.

4.5 CONCLUSION

In this thesis, we addressed the problem of lane recognition on the road, which involves autonomous vehicles. We attempted to solve it using traditional techniques such as edge recognition and Hough transforms, as well as modern techniques such as Convolutional Neural Networks. We compared the methods using images of their predictions, and the results were very satisfactory, as we discovered that even with simple models, we could detect lanes on the road. There is also a clear distinction between classical and modern methods, with the latter producing significantly better results than the former

5. CONCLUSIONS AND FUTURE WORKS

5.1 CONCLUSION

In this thesis, we addressed the problem of lane recognition on the road, which involves autonomous vehicles. We attempted to solve it using traditional techniques such as edge recognition and Hough transforms, as well as modern techniques such as Convolutional Neural Networks. We compared the methods using images of their predictions, and the results were very satisfactory, as we discovered that even with simple models, we could detect lanes on the road. There is also a clear distinction between classical and modern methods, with the latter producing significantly better results than the former. Both methods, however, have advantages and disadvantages. The choice should consider the specifics of the problem as well as the environment in which we intend to use them.

Upon juxtaposing the outcomes of the two approaches, it becomes evident that Convolutional Neural Networks (CNNs) outperform classical learning models, offering more reliable predictions in dynamic environments characterized by diverse variations and noise levels. However, it's worth noting that classical learning models could potentially yield better predictions if a statistical approach utilizing particle or Kalman filters were employed. Moreover, these classical models often provide more consistent predictions as they offer a clear understanding of the algorithm's behavior at each point, unlike CNNs, which operate as black boxes. Despite the computational demands and training time associated with CNNs, recent advancements have led to the development of smarter and more efficient architectures, thereby reducing these challenges. Consequently, the choice between these two approaches may vary depending on the specific application requirements and constraints.

5.2 FUTURE WORKS

Lane detection stands as a crucial task for vehicles, underscoring the need for robust methods capable of adapting to diverse driving conditions with accuracy and adaptability. To this end, several future recommendations encompassing both network architecture and datasets are proposed:

The ongoing organizations are prepared and tried on picture groupings with a length of 5. Notwithstanding, growing testing to remember a larger number of pictures for the succession can yield benefits concerning generalizability and improved exactness. Also, preparing the organization with fluctuating spans between pictures inside the arrangement can actually recreate different driving circumstances, like quick or slow driving paces. By accommodating varying sequence lengths and intervals between images, the proposed network architecture demonstrates flexibility. Nevertheless, extensive testing is imperative to assess the network's generalizability comprehensively.

The proposed consideration modules are coordinated after the fourth downsampling layer and the first upsampling layer in the organization design. Nonetheless, it merits investigating elective positions of the consideration layer between various layers, for example, downsampling layers one and two or downsampling layers two and three. While carrying out consideration modules in these positions might expand the quantity of boundaries because of bigger component map measures, their viability warrants testing. In addition, the upsampling layer's usefulness vigorously depends on the last picture in the succession, which may not necessarily in every case contain the vital elements for exact path recognition. As a result, modifying the network architecture to prioritize upsampling the image with the most relevant features is essential. This adjustment ensures that the upsampling process effectively incorporates the critical information required for accurate lane detection, thereby improving the overall performance of the network.

REFERENCES

[1] Parekh, D., Poddar, N., Rajpurkar, A., Chahal, M., Kumar, N., Joshi, G. P., & Cho, W. (2022). A Review on Autonomous Vehicles: Progress, Methods and Challenges [Number: 14 Publisher: Multidisciplinary Digital Publishing Institute]. Electronics, 11(14), 2162.

[2] Soni, A., Padamwar, P., & Konda, K. R. (2021). Contextual road lane and symbol generation for autonomous driving. 2021 20th IEEE International Conference on Machine Learning and Applications (ICMLA), 865–872.

[3] Tabelini, L., Berriel, R., De Souza, A. F., Badue, C., & Oliveira-Santos, T. (2022a). Lane Marking Detection and Classification using Spatial-Temporal Feature Pooling [ISSN: 2161-4407]. 2022 International Joint Conference on Neural Networks (IJCNN), 1–7.

[4] Tabelini, L., Berriel, R., De Souza, A. F., Badue, C., & Oliveira-Santos, T. (2022). Lane marking detection and classification using spatial-temporal feature pooling. 2022 International Joint Conference on Neural Networks (IJCNN), 1–7.

[5] Tabelini, L., Berriel, R., Paixao, T. M., Badue, C., De Souza, A. F., & Oliveira-Santos, T. (2021). Keep your eyes on the lane: Real-time attention-guided lane detection. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 294–302.

[6] Wang, B., & Li, H.-s. (2021). Lane detection algorithm based on moblienet + unet lightweight network. 2021 3rd International Symposium on Robotics Intelligent Manufacturing Technology (ISRIMT), 352–356.

[7] Wang, S., Yue, W., Xue, N., Chen, Y., Ji, X., & Li, C. (2022). Mining Image Semantics via Deep Learning: A Robust Lane Detection Approach for Autonomous Driving [ISSN: 2577-2465]. 2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring), 1–6.

[8] Wu, Y.-H., Gao, S., Mei, J., Xu, J., Fan, D.-P., Zhang, R.-G., & Cheng, M.-M. (2021). JCS: An Explainable COVID-19 Diagnosis System by Joint Classification and Segmentation. IEEE Transactions on Image Processing, 30, 3113–3126.

[9] Zhang, Y., Zhu, L., Feng, W., Fu, H., Wang, M., Li, Q., Li, C., & Wang, S. (2021). Vil-100: A new dataset and a baseline model for video instance lane detection. Proceedings of the IEEE/CVF International Conference on Computer Vision, 15681–15690..

[10] Zou, Q., Jiang, H., Dai, Q., Yue, Y., Chen, L., & Wang, Q. (2019). Robust lane detection from continuous driving scenes using deep neural networks. IEEE Transactions on Vehicular Technology, PP, 1–1.

[11] S. D. Pendleton, H. Andersen, X. Du, X. Shen, M. Meghjani, Y. H. Eng, D. Rus and M. H. Ang. Perception, Planning, Control, and Coordination for Autonomous Vehicles. Machines, 2017.

[12] I. S. Kholopov, "Bird's eye view transformation technique in photogrammetric problem of object size measuring at low-altitude photography," in Proceedings of the International Conference "Actual Issues of Mechanical Engineering" 2017 (AIME 2017), Atlantis Press, 2017.

[13] Yuchi Tian, Kexin Pei, Suman Jana, and Baishakhi Ray. Deeptest: Automated testing of deep-neural-network-driven autonomous cars. CoRR, abs/1708.08559, 2017.

[14] Mengshi Zhang, Yuqun Zhang, Lingming Zhang, Cong Liu, and Sarfraz Khurshid. Deeproad: Gan-based metamorphic autonomous driving system testing. CoRR, abs/1802.02295, 2018.

[15] D. Neven, B. De Brabandere, S. Georgoulis, M. Proesmans, L. Van Gool, "Towards end-to-end lane detection: an instance segmentation approach.", arXiv preprint arXiv: 1802.05591, 2018.

[16] V. Kishore Ayyadevara. Recurrent Neural Network, pages 217–257. Apress, Berkeley, CA, 2018.

[17] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence, 39(12):2481–2495, 2017.

[18] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473, 2014.

[19] Aharon Bar Hillel, Ronen Lerner, Dan Levi, and Guy Raz. Recent progress in road and lane detection: A survey, 2014.

[20] Karsten Behrendt and Ryan Soussan. Unsupervised labeled lane marker dataset generation using maps. In Proceedings of the IEEE International Conference on Computer Vision, 2019.

[21] Jingwei Cao, Chuanxue Song, Shixin Song, Feng Xiao, and Silun Peng. Lane detection algorithm for intelligent vehicles in complex road conditions and dynamic environments. Sensors, 19(14):3166, 2019.

[22] Zhang, Y., Zhu, L., Feng, W., Fu, H., Wang, M., Li, Q., Li, C., & Wang, S. (2021).Vil-100: A new dataset and a baseline model for video instance lane detection. Proceedings of the IEEE/CVF International Conference on Computer Vision, 15681–15690

[23] Zhou, D., Fang, J., Song, X., Guan, C., Yin, J., Dai, Y., & Yang, R. (2019). Iou loss for 2d/3d object detection. 2019 International Conference on 3D Vision (3DV), 85–94.

[24] Zou, Q., Jiang, H., Dai, Q., Yue, Y., Chen, L., & Wang, Q. (2019). Robust lane detection from continuous driving scenes using deep neural networks. IEEE Transactions on Vehicular Technology, PP, 1–1

[25] Bezawada, V. V., Alapati, M. S., Nidamanuri, J., & Venkataraman, H. (2022). Modeling Lane and Road Characteristics for Drive Quality Rating [ISSN: 2155-2509]. 2022 14th International Conference on COMmunication Systems & NETworkS (COMSNETS), 865–870.

[26] Chen, L., Luo, R., Weng, X., Cao, Y., & Yilihamu, A. (2021). Lane Polynomial Regression with Dual-Dimension Attention Convolution. ICMLCA 2021; 2nd International Conference on Machine Learning and Computer Application, 1–7.

[27] Chen, T., Cai, Z., Zhao, X., Chen, C., Liang, X., Zou, T., & Wang, P. (2020). Pavement crack detection and recognition using the architecture of segnet. Journal of Industrial Information Integration, 18, 100144.

[28] Chng, Z. M., Lew, J. M. H., & Lee, J. A. (2021). RONELD: Robust Neural Network Output Enhancement for Active Lane Detection [ISSN: 1051-4651]. 2020 25th International Conference on Pattern Recognition (ICPR), 6842–6849.

[29] Dogru, H. B., & Zengin, A. T. (2022). Fast-ICA Based Lane Detection Method for Autonomous Vehicles. 2022 26th International Conference Electronics, 1–6.

[30] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale [arXiv:2010.11929 [cs] version: 2].

[31] Guo, Y., Chen, G., Zhao, P., Zhang, W., Miao, J., Wang, J., & Choe, T. E. (2020). Gen-lanenet: A generalized and scalable approach for 3d lane detection. CoRR, abs/2003.10656.

[32] Hafiz, R. I., Faruq, T. B., & Nyeem, H. (2021). Revisiting Deep Learning Models for Road Lane Detection. 2021 5th International Conference on Electrical Engineering and Information Communication Technology (ICEEICT), 1–6.

[33] Hou, H., Guo, P., Zheng, B., & Wang, J. (2021). An Effective Method for Lane Detection in Complex Situations [ISSN: 2378-8607]. 2021 9th International Symposium on Next Generation Electronics (ISNE), 1–4.

[34] Kumar, S., Jailia, M., & Varshney, S. (2022). A Comparative Study of Deep Learning based Lane Detection Methods. 2022 9th International Conference on Computing for Sustainable Global Development (INDIACom), 579–584.

[35] Lee, M., Lee, J., Lee, D., Kim, W., Hwang, S., & Lee, S. (2022). Robust Lane Detection via Expanded Self Attention [ISSN: 2642-9381]. 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 1949–1958.

[36] Li, C., Shi, J., Wang, Y., & Cheng, G. (2022). Reconstruct from Top View: A 3D Lane Detection Approach based on Geometry Structure Prior [ISSN: 2160-7516]. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 4369–4378.

[37] Liu, R., Yuan, Z., Liu, T., & Xiong, Z. (2021). End-to-end Lane Shape Prediction with Transformers. 2021 IEEE Winter Conference on Applications of Computer Vision (WACV), 3693–3701.

[38] Liu, Z., Li, L., Wang, H., Jiang, P., Fang, Y., & Wu, C. (2021). Vision transformer with convolutional encoder-decoder for hand gesture recognition using 24 ghz doppler radar. 2021 IEEE Radar Conference (RadarConf21), 1–5.

[39] Parekh, D., Poddar, N., Rajpurkar, A., Chahal, M., Kumar, N., Joshi, G. P., & Cho, W. (2022). A Review on Autonomous Vehicles: Progress, Methods and Challenges [Number: 14 Publisher: Multidisciplinary Digital Publishing Institute]. Electronics, 11(14), 2162.

[40] Wang, S., Yue, W., Xue, N., Chen, Y., Ji, X., & Li, C. (2022). Mining Image Semantics via Deep Learning: A Robust Lane Detection Approach for Autonomous Driving [ISSN: 2577-2465]. 2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring), 1–6.

[41] Yang, S., Gao, L., Yanan, Z., & Li, X. (2021). Research on the quantitative evaluation of the traffic environment complexity for unmanned vehicles in urban roads. IEEE Access, PP, 1–1

[42] Zhang, Y., Zhu, L., Feng, W., Fu, H., Wang, M., Li, Q., Li, C., & Wang, S. (2021). Vil-100: A new dataset and a baseline model for video instance lane detection. Proceedings of the IEEE/CVF International Conference on Computer Vision, 15681–15690.

[43] Zou, Q., Jiang, H., Dai, Q., Yue, Y., Chen, L., & Wang, Q. (2019). Robust lane detection from continuous driving scenes using deep neural networks. IEEE Transactions on Vehicular Technology, PP, 1–1.

[44] Seungwoo Yoo, Hee Seok Lee, Heesoo Myeong, Sungrack Yun, Hyoungwoo Park, Janghoon Cho, and Duck Hoon Kim. End-to-end lane marker detection via row-wise classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 1006–1007, 2020.

[45] Qin Zou, Hanwen Jiang, Qiyu Dai, Yuanhao Yue, Long Chen, and Qian Wang. Robust lane detection from continuous driving scenes using deep neural networks. IEEE transactions on vehicular technology, 69(1):41–54, 2019

[46] H. Zhou and H. Wang, "Vision-based lane detection and tracking for driver assistance systems: A survey," in Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM), 2017 IEEE International Conference on, pp. 660–665, IEEE, 2017.

[47] Z. Kim, "Robust lane detection and tracking in challenging scenarios," IEEE Transactions on Intelligent Transportation Systems, vol. 9, no. 1, pp. 16–26, 2008.

[48] Jigang Tang, Songbin Li, and Peng Liu. A review of lane detection methods based on deep learning. Pattern Recognition, page 107623, 2020.

[49] Johan Vos, Haneen Farah, and Marjan Hagenzieker. Speed behaviour upon approaching freeway curves. Accident Analysis & Prevention, 159:106276, 2021.

[50] Qin Zou, Hanwen Jiang, Qiyu Dai, Yuanhao Yue, Long Chen, and Qian Wang. Robust lane detection from continuous driving scenes using deep neural networks. IEEE transactions on vehicular technology, 69(1):41–54, 2019.