REPUBLIC OF TÜRKİYE

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Information Technologies

LANE SEGMENTATION AND ROAD DETECTION IN SMART CAR VISION USING DEEP CONVOLUTIONAL NEURAL NETWORK

Sarah Kadhim Hwaidi ALFADHLI

Master's Thesis

Supervisor

Asst. Prof. Dr. Timur İNAN

İstanbul, 2024

LANE SEGMENTATION AND ROAD DETECTION IN SMART CAR VİSİON USING DEEP CONVOLUTIONAL NEURAL NETWORK

Sarah Kadhim Hwaidi ALFADHLI

Information Technologies

Master's Thesis

ALTINBAŞ UNIVERSITY

2024

The thesis titled LANE SEGMENTATION AND ROAD DETECTION IN SMART CAR VISION USING DEEP CONVOLUTIONAL NEURAL NETWORK prepared by SARAH KADHIM ALFADHLI and submitted on x/x/2023 has been **accepted unanimously** for the degree of Master of Science in Information Technologies.

Asst. Prof. Dr.Timur INAN

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	_____

I hereby declare that this thesis meets all format and submission requirements of a Master's thesis.

Submission date of the thesis to the Graduate Education Institute: ___/___/___

I hereby declare that all information and 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.

Sarah Kadhim Hwaidi ALFADHLI

Signature

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my thesis supervisor, professor Timur Inan, for his invaluable guidance, patience, and expert knowledge throughout the course of this research. his insights and suggestions have been fundamental in shaping both the direction and execution of this thesis. professor Inan's unwavering support and encouragement have been a source of motivation and inspiration, making this journey both enlightening and enjoyable. I am also immensely grateful to the members of my thesis committee for their constructive feedback and essential advice. their expertise and perspectives have significantly contributed to the quality of this work. my sincere thanks also go to my colleagues and friends for their understanding and support during this challenging yet rewarding process. their camaraderie and encouragement have been a constant source of strength. I must also acknowledge the support of [Altinbas University department of Information Technologies], whose resources and facilities have been instrumental in the completion of this thesis. lastly, but most importantly, I would like to thank my family for their unwavering love, patience, and belief in me. their sacrifices and constant encouragement have been the bedrock of my perseverance and success. this thesis would not have been possible without the contributions and support of all these individuals. I am deeply thankful to each one of them

ABSTRACT

LANE SEGMENTATION AND ROAD DETECTION IN SMART CAR VISION USING DEEP CONVOLUTIONAL NEURAL NETWORK

ALFADHLI, Sarah Kadhim Hwaidi

M.Sc., Information Technologies, Altınbaş University,

Supervisor: Asst. Prof. Dr. Timur INAN

Date: 01/2024

Pages: 83

This thesis will delve into the intricacies of designing, training, and evaluating a deep CNN architecture for lane segmentation and subsequently harness the discriminative power of SVMs for road detection. Through a comprehensive investigation of these methodologies, this thesis aims to contribute to the advancement of smart car vision systems, ultimately paving the way for safer and more reliable autonomous driving solutions in an ever-evolving urban landscape. As society becomes increasingly reliant on digital communication and data, the ability to understand, process, and generate human language has become a critical component in various applications, from virtual assistants and sentiment analysis to machine translation and content recommendation systems

Keywords: AI, Segmentation, ML, Lane Detection, SVM, CNN.

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

ABBREVIATIONS

SVM	:	Support Vector Machine
ML	:	Machine Learning
RoI	:	Region Of Interest
RNN	:	Recurrent Neural Networks
CGM	:	Computer Graphics Metafile
CNN	:	Convolutional Neural Network

1. INTRODUCTION

1.1 BACKGROUND

A paradigm shift has occurred in the automotive industry in recent years as a result of the introduction of autonomous driving technology, which promises to make transportation networks safer and more efficient. The proper perception of the road and the infrastructure that surrounds it is one of the main issues that must be overcome in order to enable autonomous vehicles to navigate complex urban areas. Lane segmentation and road detection are two processes that play critical roles in this perception process. These processes give key information that is necessary for the control of vehicles and the making of decisions. Employing a synergistic mix of Deep Convolutional Neural Networks (CNN) and Support Vector Machine (SVM) classification approaches, this thesis investigates the essential problem of lane segmentation and road detection in the context of smart automobile vision. Specifically, the thesis focuses on the application of these techniques. Convolutional neural networks (CNNs) have shown exceptional capabilities in picture processing and feature extraction, making them an ideal candidate for extracting complicated spatial information from road scenes. This is due to the rise of deep learning methodologies.

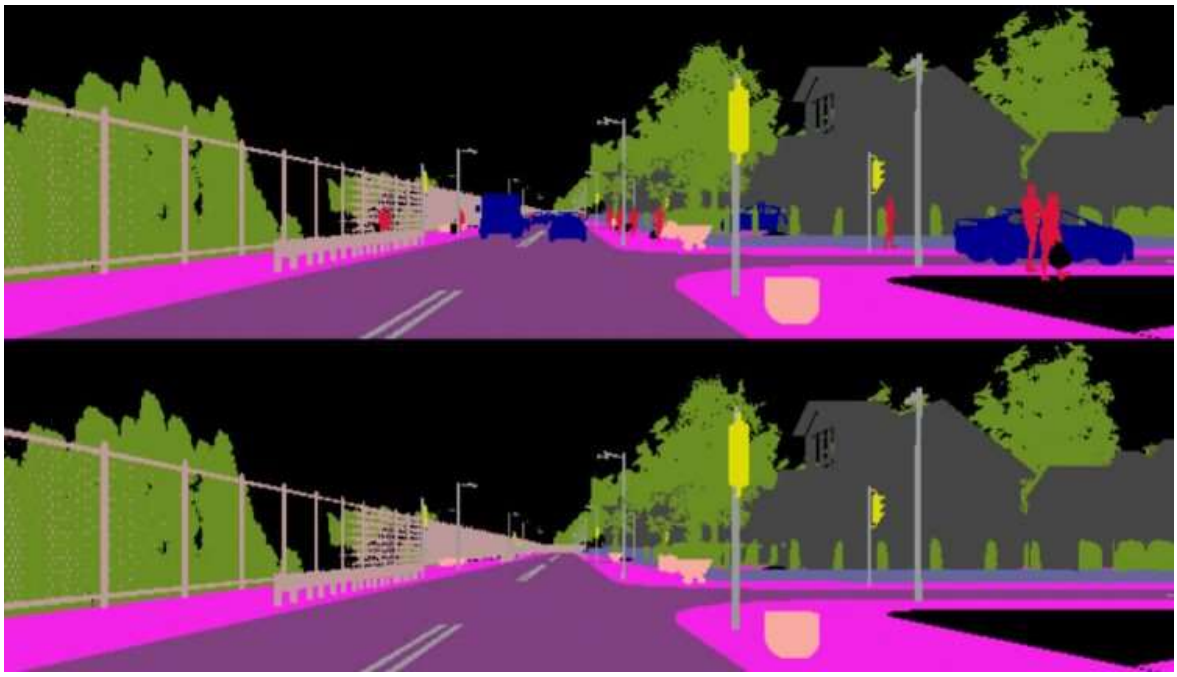

Figure 1.1: Example of Road Segmentation [1].

Concurrently, SVM classification offers a robust and versatile framework for decision-making, enhancing the precision and reliability of road detection.

The integration of these advanced technologies aims to address the complex challenges posed by varying environmental conditions, diverse road geometries, and the need for real-time processing. This thesis will delve into the intricacies of designing, training, and evaluating a deep CNN architecture for lane segmentation and subsequently harness the discriminative power of SVMs for road detection.

Through a comprehensive investigation of these methodologies, this thesis aims to contribute to the advancement of smart car vision systems, ultimately paving the way for safer and more reliable autonomous driving solutions in an ever-evolving urban landscape.

As society becomes increasingly reliant on digital communication and data, the ability to understand, process, and generate human language has become a critical component in various applications, from virtual assistants and sentiment analysis to machine translation and content recommendation systems.

Deep learning, particularly through the use of neural networks, has emerged as a powerful paradigm within NLP, revolutionizing the way we app roach language-related tasks.

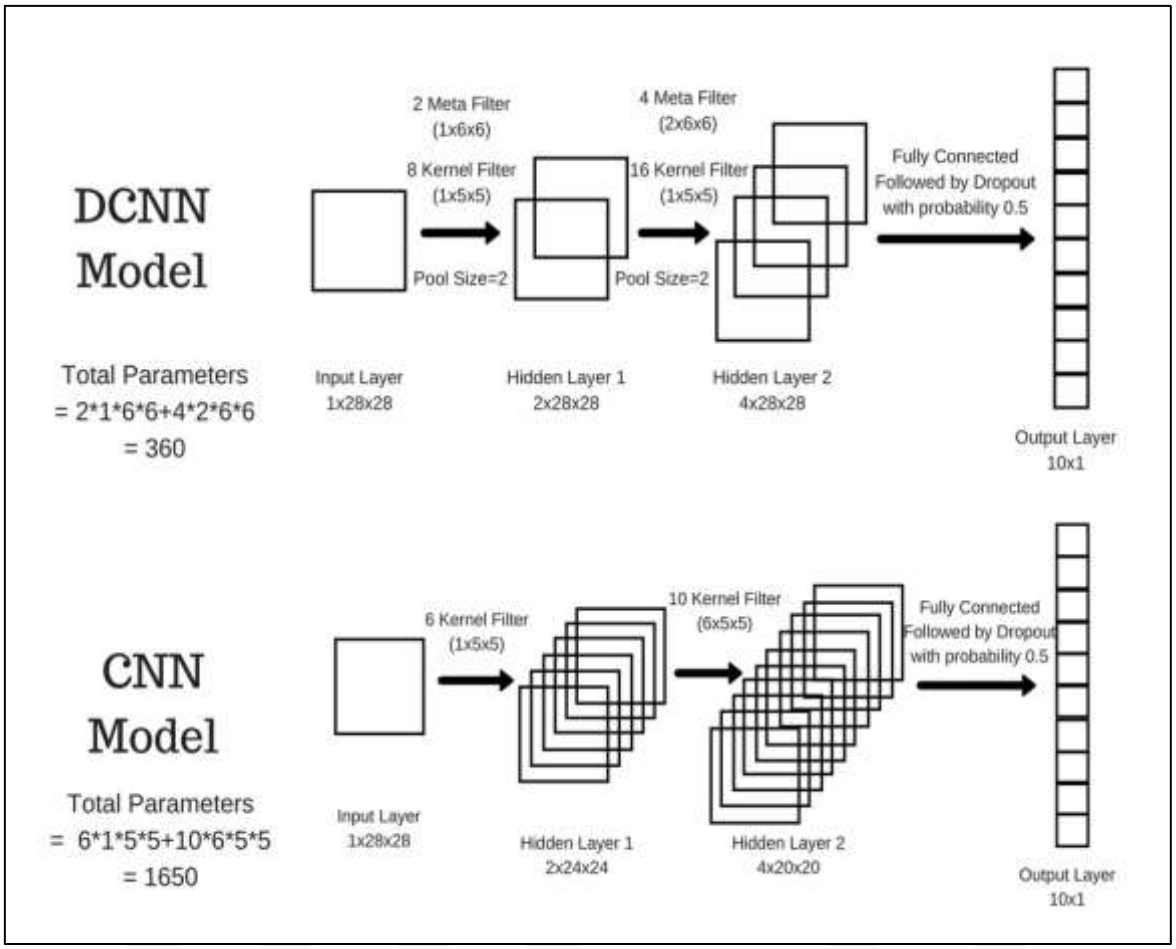

Figure 1.2: CNN vs DCNN Models [5].

This thesis delves into the multifaceted world of deep learning for natural language processing, aiming to explore the advancements and applications that have reshaped the landscape of this field. The proliferation of large-scale datasets and the development of increasingly sophisticated neural architectures have significantly improved our ability to model and understand natural language. These advancements have led to remarkable breakthroughs in areas such as machine translation, sentiment analysis, and question-answering systems. Moreover, pre-trained language models like BERT, GPT-3, and their successors have further pushed the boundaries of what is achievable in NLP. These models, trained on vast corpora of text data, have exhibited human-level performance on a variety of language tasks and have opened up new avenues for transfer learning and fine-tuning. This thesis aims to provide a comprehensive overview of the key deep learning techniques in NLP, highlighting their theoretical foundations and practical applications. We will delve into the intricacies of training deep neural networks for language understanding, explore the

challenges posed by different NLP tasks, and examine the ethical and societal implications of these advances. Through a critical examination of state-of-the-art research, real-world case studies, and potential future directions, this thesis seeks to shed light on the ongoing revolution in natural language processing powered by deep learning. As we journey through this exploration, we will gain valuable insights into how deep learning is not only reshaping NLP but also influencing how we communicate, interact with technology, and process the vast sea of textual information that surrounds us in the digital age.

1.2 PROBLEM STATEMENT

As the automotive industry continually evolves towards the realization of autonomous vehicles, the critical task of real-time lane detection and road segmentation using computer vision assumes paramount significance. Achieving reliable and accurate lane marking detection and road region segmentation represents a cornerstone for the safe and efficient operation of smart cars. This problem statement seeks to elucidate the multifaceted challenges encompassing this domain, taking into account the complexities of real-world driving scenarios, diverse environmental conditions, and the imperative need for robust and adaptive algorithms.

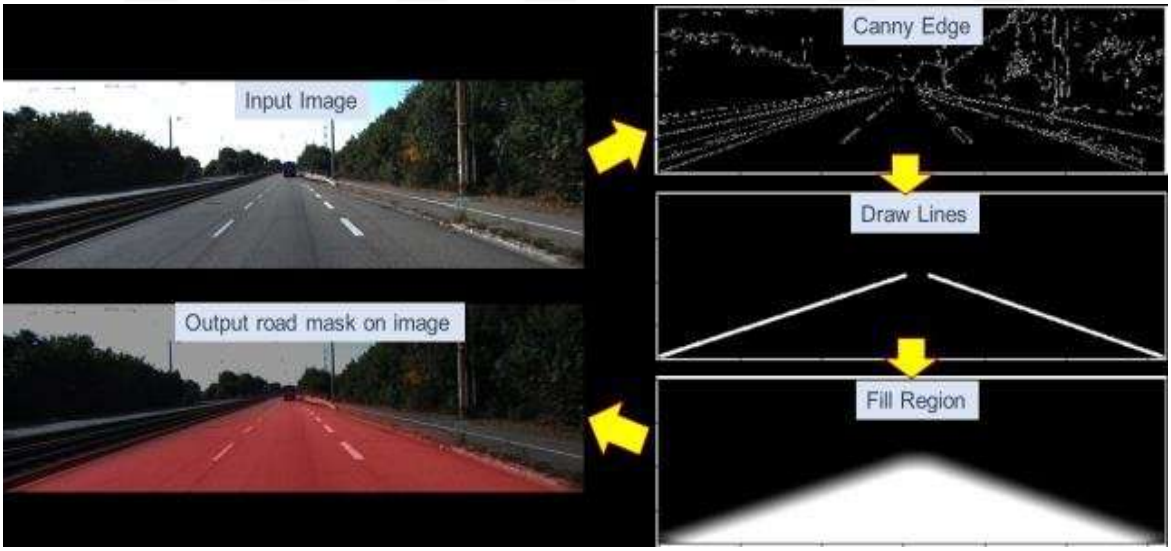

Figure 1.3: Lane Segmentation Using Canny Edge Method [6].

Modern smart cars are equipped with a plethora of sensors and technologies designed to emulate human perception and decision-making capabilities has emerged as a linchpin technology for scene understanding and navigation. In particular, lane detection and road

segmentation are indispensable components of this perception stack, as they provide crucial information for autonomous vehicle control, trajectory planning, and obstacle avoidance.

Challenges:

a. Variability in Lane Markings: Real-world Lane markings exhibit substantial variability in terms of color, width, type, and degradation. Consequently, developing algorithms that can robustly identify lane markings across different regions and conditions remains a formidable challenge.
b. Environmental Factors: Adverse weather conditions such as rain, snow, fog, or glare from sunlight can obscure or distort lane markings and road boundaries. Fog, in particular, poses a unique challenge due to its ability to limit visibility drastically.
c. Lane Change and Intersections: Smart cars must not only track lanes but also recognize when a lane change is necessary or when the vehicle approaches complex intersections. This requires the development of algorithms capable of understanding the dynamic nature of lane usage.
d. Infrastructure Variability: Roads across the world exhibit a wide range of infrastructure variations, including the presence of non-standard lane markings, diverse road geometries, and even the absence of lanes in some cases. Smart cars need to adapt to such diversity.
e. Real-time Processing: Achieving low-latency lane detection and road segmentation is paramount to ensure that smart cars can make real-time decisions in dynamic traffic scenarios. This necessitates the optimization of algorithms for efficiency without compromising accuracy.
f. Safety and Reliability: A single misjudgment in lane detection or road segmentation can have severe safety implications. Ensuring the reliability and fail-safety of computer vision-based perception systems is a critical concern.
g. Human Interaction: Smart cars must also account for human drivers and pedestrians, understanding their intentions and actions in relation to lane changes and road usage.

1.3 THESIS OBJECTIVES

To develop a comprehensive understanding of the state-of-the-art techniques and methodologies in computer vision, with a specific focus on lane segmentation and road detection in the context of smart car vision. To investigate the integration of Deep

Convolutional Neural Networks (CNNs) and Support Vector Machine (SVM) classification for enhancing the accuracy and robustness of lane segmentation and road detection in smart car environments.

1.3.1 Design and Implement a Deep CNN Architecture for Lane Segmentation

a. To design a deep CNN architecture optimized for the task of lane segmentation, incorporating appropriate convolutional layers, activation functions, and normalization techniques.
b. To implement the designed CNN model using popular deep learning frameworks, such as MATLAB r2022a and ensure its compatibility with real-time smart car vision systems.
c. To fine-tune hyperparameters and optimize the CNN model's architecture to achieve state-of-the-art performance in lane segmentation accuracy.

1.3.2 Explore SVM Classification for Road Detection

a. To investigate the feasibility and effectiveness of employing Support Vector Machine (SVM) classification for road detection based on features extracted from the CNN-processed image data.
b. To develop a robust feature extraction pipeline that captures discriminative information for road regions within the smart car's field of view.
c. To train and fine-tune the SVM classifier to accurately differentiate road regions from non-road regions in diverse environmental conditions and road types.

1.3.3 Integration and Fusion of Lane and Road Information

a. To explore methods for effectively combining lane segmentation and road detection outputs, ensuring seamless integration within the smart car's perception system.
b. To develop strategies for handling complex scenarios where lane markings might be ambiguous or temporarily obscured, relying on road information for enhanced situational awareness.

1.3.4 Real-world Testing and Evaluation

a. To conduct extensive real-world testing and evaluation of the proposed lane segmentation and road detection system using representative datasets and smart car platforms.
b. To assess the system's performance across a wide range of environmental conditions, including variations in lighting, weather, road geometry, and traffic density.

1.3.5 Optimization for Real-time Implementation

a. To optimize the computational efficiency and processing speed of the integrated system, ensuring that it meets the stringent real-time requirements of smart car vision applications.
b. To minimize latency and resource consumption while maintaining high accuracy in lane segmentation and road detection.

1.3.6 Comparative Analysis and Benchmarking

a. To compare the performance of the proposed integrated CNN-SVM system with existing lane detection and road segmentation approaches, highlighting its strengths and weaknesses.
b. To establish benchmarks and metrics for evaluating the system's contribution to the field of smart car vision.

1.4 THESIS CONTRIBUTIONS

1.4.1 Advancement of Smart Car Vision Technology

This thesis contributes to the field of smart car vision technology by proposing a novel and integrated approach to lane segmentation and road detection. It offers a comprehensive framework that combines the power of Deep Convolutional Neural Networks (CNNs) and Support Vector Machine (SVM) classification, thereby advancing the state-of-the-art in smart car perception systems.

1.4.2 Robust Lane Segmentation

The research conducted in this thesis results in the development of a deep CNN architecture optimized for lane segmentation. It contributes by demonstrating a highly accurate and

adaptable lane detection system that excels in real-world scenarios, even in challenging conditions such as varying lane markings and adverse weather.

1.4.3 Innovative Road Detection Methodology

This thesis explores the application of SVM classification for road detection, offering a unique and effective approach to recognizing road regions within the smart car's field of view. It contributes by providing a robust feature extraction pipeline and training methodology that enhances road detection accuracy across diverse environmental contexts.

1.4.4 Seamless Integration of Lane and Road Information

One of the significant contributions of this thesis is the investigation of strategies for effectively combining lane segmentation and road detection outputs. This integration enhances the smart car's situational awareness, especially in scenarios where lane markings are ambiguous or temporarily obscured, contributing to safer and more reliable autonomous driving.

1.4.5 Comparative Analysis and Benchmarking

The thesis offers a comparative analysis that benchmarks the proposed integrated CNN-SVM system against existing lane detection and road segmentation approaches. It contributes by shedding light on the system's strengths and weaknesses, providing valuable insights for further research and development in the field.

2. RELATED WORKS

2.1 INTRODUCTION

In this chapter, we will go further into the traditional approaches to lane identification, dissecting model-based and feature-based methods to investigate how they operate, how they differ from one another, and how they are comparable to one another. The first thing that is done in the model-based detection technique is a comparison of the extracted characteristics to the model of the lane that has been supplied. This model is used to determine whether or not a collision has occurred. This is accomplished by changing the lane identification process into one that involves the computation of model parameters. One of its many benefits is that it is not susceptible to noise, and in addition to that, it helps lessen the localization of blurring. The feature-based detection technique, on the other hand, classifies each pixel in the picture as either a lane point or a non-lane point based on specified lane features (such as edge gradient, width, intensity, and color). In other words, the feature-based detection method classifies each pixel as either a lane point or a non-lane point. This technique determines the presence of lanes by analyzing the information contained inside the picture. In order to shed light on the research paths that are currently being pursued as well as the efficacy of approaches that are considered to be state-of-the-art at the present time, the purpose of this chapter is to provide a comprehensive assessment of the published literature on lane detecting systems. In this study, not only the methods for gathering datasets but also the final dataset that was used for training, validating, and testing networks were investigated and discussed. This study establishes a solid basis for future research into the subject of automation by illuminating the present state of the art in lane detecting approaches, in addition to the difficulties and possibilities that lie in wait for the field in the years that are to come. This analysis was carried out to shed light on the current state of the art in lane detecting techniques.

2.2 AN OVERVIEW OF THE CURRENT BODY OF WORK

According to research that was issued by the World Health Organization in June 2022, it is expected that there be 1.3 million fatalities that occur annually as a consequence of accidents that are associated to traffic. This number is an increase from the previous prediction of 1.1

million deaths. It may be difficult for human drivers to remain in the correct lane and maintain a safe distance from the car in front of them since they are expected to pay constant attention to the road. Fatigue, tiredness, inattention, and sleepiness are all examples of human frailties; as a result, all of these conditions may be hazardous for drivers. The attention of the driver may be readily diverted away from the road and distracted by a number of electronic gadgets, such as cellphones, in-car entertainment systems, and navigation systems. This can lead to serious accidents. Road traffic accidents have a huge detrimental influence on society as a whole not just as a result of the emotional toll they take on people but also of the financial resources they drain from the economy. The creation of both active and passive safety measures for automobiles is a direct answer to the worry expressed in the previous sentence. When we hear the phrase "passive safety system," the first things that often spring to our minds are examples such as seat belts and airbags [1]. As a consequence of this, tracking could be able to help in mistakenly perceiving occlusion because to poor lane markers [12].

Gong et al. [34] preprocessed the road picture that had been captured separately and formed the ROI by making use of the double threshold methodology. We are able to increase the algorithm's real-time performance as well as lessen the noise that comes from the roadside thanks to the method of intercepting the area of interest that we are interested in, which contains information about the lane lines. After that, a technique for improving images called the exponential function transformation is applied to the picture, and the result is a change in the grayscale value of the image. When a nonlinear gray shift is applied, the region of the background that already had a low grey value becomes darker, and the area surrounding the lane lines becomes brighter. This not only makes the contrast better, but it also brings more emphasis to the outlines of the region that has substantial grey value concentrations.

After that, the straight-line fit capability of the Hough Transform was put to use in order to build a workable multi-layer evaluation function that could be put to use in order to perform online change of lane lines. In other words, the Hough Transform was used so that a functional multi-layer evaluation function could be created. Kasmi et al. [44] is an additional piece of study that suggested making use of the conventional method. The author started off by using an approach that has been proven effective in the past when it comes to lane identification. They chose the Region of Interest that had the highest potential.

Dhanashirur [95] suggests that a lane detection system might be built with the use of machine learning. The dataset was preprocessed for the purpose of this study using adaptive thresholding, commonly known as the Otsu approach for defining ROI in an image. This technique was used to determine the region of interest (ROI). In order to generate Bayesian inference, the Cascaded Dempster Schafer Combination Rule is then used. In the last phase of the post-process, the data are cleaned up by eliminating any remaining outliers by the sequential application of morphological operations on a very small kernel. These operations include erosion and dilation.

Later on, Feng and Werner Wiesbeek [89] argued that ML and DL should be combined in the study that they were doing. An early effort at finding a solution to the issue of lane identification was made by the author in the form of a semantic segmentation-based technique. This was done in the early stages of the project. A 5-layer SegNet segmentation neural network served as the foundation for the construction of this method, which was then split into an encoder and a decoder upon completion. However, there are certain segmentation ambiguities based on the results: in some single cycles, regions that do not belong to the lane will be separated into the lane, and vice versa. These ambiguities exist because the findings are not completely conclusive. These contradictions are a direct outcome of the imperfect nature of the findings.

In a similar fashion, we conducted research in order to create a reliable model for lane detection. This model was our goal. It is still difficult to find a solution to the issue of identifying lanes that have a sharp bend in them. Because of this, Fakhfakh [45] proposed a one-of-a-kind strategy for classifying and calculating curved lanes. This technique takes use of a Bayesian framework in order to estimate the parameters of many hyperbolas, and it is able to recognize curved lanes even when used in difficult contexts. A hyperbola is used in the initial phase of the method to mimic the trajectory across each segment. This is done by tracing the path of the object. In order to determine the parameters of the hyperbola, the hierarchical Bayesian model that has been provided is applied. Following the completion of some fundamental image processing in order to extract contours from the input picture, we go on to the next step of describing the recovered lanes by fitting them to the analytical

model that was selected. After we have finished the preceding phases in the picture processing, we will move on to this step.

An new network-based deep learning methodology for lane recognition has been developed by Sun et al. [71], who make use of methods such as atrous convolution and spatial pyramid pooling in their work. The LaneNet protocol is used to form the network; this protocol only requires one encoder, but it makes use of two distinct decoders throughout the building process. The two distinct kinds of decoders are referred to by the names Embedding Decoder and Binary Decoder, which are the phrases that are most often used to refer to them. In order to bring the encoder for LaneNet up to date, its inventor used a mix of the Atrous ResNet-101 network and the Spatial Pyramid Pooling (SPP) network in a sequential fashion. The combination was referred to as an SPP, which stands for spatial pyramid pooling. The only notable distinction between the Embedding Decoder and the Binary Decoder is the number of output dimensions; other than that, there is a lot of overlap between the two types of decoders.

In [80], Philion proposes a ground-breaking totally convolutional lane identification methodology. This method does not rely on post-processing to detect structure; rather, it uses convolutions to define lanes. Around the same time, Dawam and Feng presented their computer vision-based road surface marker recognition system in the publication [46]. This provided autonomous automobiles with an additional data source from which they may choose. The researchers were successful in teaching the detector to identify between 25 distinct types of road surface markings by using YOLOv3 in the cloud and a total of 25,000 pictures. The findings of the studies indicate that the method of detection has the potential to attain a good level of speed while simultaneously achieving a decent degree of accuracy in the future.

Stand-alone deep learning was proposed as a solution to this problem in Muthalagu et al.'s [35] article. They were able to do this by not only gaining an understanding of the segmentation of the lane markers, but also of the location and geometry of each lane in the form of critical points. This was made possible by using a Convolutional Neural Network (CNN) design that is both space-saving and highly effective over several stages. The technique that has been proposed combines two different types of networks: a lane mask

proposal network and a lane key-point determination network. When this is done, an exact calculation of the critical spots that display the left and right lane markings of the vehicle lanes may be obtained.

An encoder-decoder network was proposed as a strategy for semantic segmentation by Dewangan et al. [37]. This approach was described in their paper. It has been determined that the best way to achieve this objective is to use a model that combines UNet and ResNet. After the resolution of the picture was reduced, the image's characteristics were reconstructed with the aid of the segmentation technique developed by ResNet-50. UNet was then used in order to upsample and decode the picture segments once the features had been identified.

On the other hand, using fully supervised algorithms might be challenging in challenging road settings due to the absence of segmentation masks for host lanes. Yousri et al. [23] propose a solution to this obstacle by merging traditional computer vision methods with deep learning strategies in order to create a trustworthy benchmarking framework for lane identification tasks in intricate and ever-changing road circumstances. This allows the authors to overcome the challenge presented by the previous sentence. Traditional approaches to computer vision were combined with deep learning strategies by the researchers so that they could achieve this goal.

According to the findings of Kanagaraj and colleagues [25], increasing the usage of Convolutional Neural Networks in combination with Spatial Transformer Networks and real-time lane recognition boosted the effectiveness of autonomous autos. The process begins by grayscaling a live picture, which is followed by applying a Gaussian blur to the image in order to soften the edges while also minimizing the amount of noise. This is the first stage. The subsequent step that has to be accomplished is to make use of a Canny function so that aid with edge recognition may be provided. After the Canny procedure, which involves measuring the gradients of the pixels that are near to one another, the image's borders are formed. This occurs after the Canny technique has been carried out.

A lane identification system that was based on learning a complete reference quality-aware discriminative gradient deep model was proposed by J. Liu [72]. This technology was

developed specifically for use in autonomous cars. The author was able to determine the locations of the lanes by using a gradient-guided deep convolutional network. This is owing to the fact that the values of the gradient at the edges of the lanes are greater than those in the areas that surround them. The subsequent task is to locate additional gradient signals that are easily distinguishable via the use of geometric features and the Full Reference Image Quality Assessment (FR-IQA) approach. After that, a recurrent neural layer that uses ambiguous visual inputs to represent the location of specified lanes is used. This layer is used after the previous one has been employed.

In order to accomplish the task of lane identification, a deep hybrid architecture was created by combining convolutional neural networks (CNNs) with recurrent neural networks (RNNs). Zou et al. [126] were the ones who first described this design. Each frame is subjected to an analysis by a CNN block, and the results of those analyses are used to compile data that is significant. After that, the RNN block receives the features that were previously extracted by the CNN from a number of time-series-property continuous frames for the purposes of feature learning and lane prediction.

2.3 SUMMARY OF LITERATURE REVIEW

The section discusses various methodologies and approaches for lane detection and road recognition in the context of autonomous vehicles. Several research studies and techniques are summarized as follows:

a. In order to preprocess road photographs, Gong et al. [34] utilized a twofold threshold methodology. The primary aim of this methodology was to create a Region of Interest (ROI) for the purpose of obtaining lane line information. Image contrast was improved through the application of an exponential function transformation, with a special emphasis placed on regions that contained significant concentrations of grey value.
b. For the purpose of lane identification, Kasmi et al. [44] utilized conventional approaches, picking the Region of Interest that possessed the most potential.
c. • Dhanashirur [35] presented a lane detection system that is based on machine learning capabilities. The region of interest (ROI) was determined by Bayesian inference using the Cascaded Dempster Schafer Combination Rule, followed by a post-process phase

that involved morphological procedures for the removal of outliers. Adaptive thresholding, also known as the Otsu method, was used to define the ROI.

d. For the purpose of lane identification, Feng and Werner Wiesbeek [39] conducted experiments in which they examined the combination of Machine Learning (ML) and Deep Learning (DL). In the beginning, they utilized a SegNet segmentation neural network that came equipped with an encoder-decoder architecture. However, they ran into segmentation ambiguities as a result of inconclusive results.
e. Fakhfakh [45] presented a novel approach to the classification of curved lanes by making use of a Bayesian framework. It was possible to distinguish curved lanes even in difficult circumstances thanks to the approach that estimated hyperbola characteristics.
f. LaneNet is a network-based deep learning technology that was created by Sun et al. [51]. This learning approach makes use of atrous convolution and spatial pyramid pooling. The encoder-decoder structure of the network, which included both an embedding decoder and a binary decoder, was designed with the intention of enhancing the accuracy of lane recognition.
g. Philippion [50] proposed a solution for lane identification that is based on convolution and does not require any post-processing. For the purpose of defining lanes, it utilized convolutions, which provided a reliable solution.
h. Dawam and Feng [46] demonstrated a computer vision-based road surface marker recognition system that utilized YOLOv3. They were able to correctly differentiate between 25 distinct types of road surface markings, indicating the potential for both speed and accuracy.
i. Specifically, Muthalagu et al. [35] presented a Convolutional Neural Network (CNN) design that combines lane marker segmentation and key-point determination networks in order to properly identify essential spots of left and right lane markings.
j. An encoder-decoder network for semantic segmentation was constructed by Dewangan et al. [37], who used UNet with ResNet in order to reconstruct image features and segments.
k. Yousri et al. [23] developed a benchmarking framework for lane recognition in difficult road scenarios by combining classical computer vision approaches with deep learning. This framework was used to generate a benchmark.

l. By preprocessing live images, Kanagaraj et al. [25] were able to improve the efficiency of autonomous vehicles through the utilization of Convolutional Neural Networks (CNNs) and Spatial Transformer Networks.
m. J. Liu [52] presented a lane identification system that would locate lanes based on gradient values and important points. This system would make use of a gradient-guided deep convolutional network.
n. Both Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) were utilized by Zou et al. [26] in order to perform an analysis of continuous frames and to make predictions regarding the positions of lanes over time.
o. Pihlank and Riid [69] presented a unique method for lane detection that uses autoencoders, residual neural networks, and densely connected neural networks in conjunction with one another.
p. SCNN + RONELD and ENet-SAD + RONELD were reported as two unique methods by Z. M. Chng and colleagues. These methods brought attention to the significance of convolutional neural networks in modern lane detection algorithms, while also exposing the limitations of these networks in terms of their ability to handle unknown scenarios.

These studies encompass a wide range of techniques, from conventional computer vision to deep learning approaches, aiming to improve the accuracy and reliability of lane detection and road recognition for autonomous vehicles and can be summarized in Table 2.1:

Table 2.1: The Summary of the Mentioned Research Studies.

Study and Reference	Methodology and Key Points
Gong et al. [34]	- Preprocessed road images using double thresholding. - Formed the Region of Interest (ROI) for lane information. - Applied exponential function transformation for enhanced image contrast.
Kasmi et al.	- Utilized conventional methods for lane identification. - Chose ROI with the highest potential.
Dhanashirur	- Proposed a machine learning-based lane detection system. - Used adaptive thresholding (Otsu method) to define ROI. - Applied Bayesian inference with the Cascaded Dempster Schafer Combination Rule. - Performed post-processing with morphological operations for outlier removal.
Feng and Werner Wiesbeek	- Explored combining Machine Learning (ML) and Deep Learning (DL) for lane identification. - Initially used a SegNet segmentation neural network with encoder-decoder architecture. - Encountered segmentation ambiguities due to inconclusive results.
Fakhfakh	- Introduced a unique strategy for classifying curved lanes using a Bayesian framework. - Estimated hyperbola parameters for recognizing curved lanes even in challenging contexts.
Sun et al.	- Developed LaneNet with atrous convolution and spatial pyramid pooling. - Employed an encoder-decoder structure with Embedding and Binary Decoders. - Aimed at improving lane recognition accuracy.
Philion	- Proposed a convolution-based lane identification method without post-processing. - Relied on convolutions to define lanes, offering a robust solution.
Dawam and Feng	- Presented a computer vision-based road surface marker recognition system using YOLOv3. - Successfully distinguished 25 different types of road surface markings.
Muthalagu et al.	- Introduced a Convolutional Neural Network (CNN) design combining lane marker segmentation and key-point determination networks. - Precisely identified critical points of left and right lane markings.
Dewangan et al.	- Implemented an encoder-decoder network for semantic segmentation, combining UNet and ResNet. - Reconstructed image features and segments.

Table 2.1: The Summary of the Mentioned Research Studies “Table Continued” .

Yousri et al.	- Merged traditional computer vision methods with deep learning for lane identification in complex road scenarios.
Kanagaraj et al.	- Used Convolutional Neural Networks (CNNs) and Spatial Transformer Networks for preprocessing live images, enhancing autonomous vehicle effectiveness.
J. Liu	- Proposed a lane identification system utilizing a gradient-guided deep convolutional network to locate lanes based on gradient values and critical spots.
Zou et al.	- Combined Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) to analyze continuous frames and predict lane positions over time.
Pihlank and Riid	- Introduced a novel technique combining autoencoders, residual neural networks, and densely connected neural networks for lane detection.
Z. M. Chng et al.	- Presented SCNN + RONELD and ENet-SAD + RONELD methods, emphasizing the importance of convolutional neural networks in contemporary lane detection algorithms but noting limitations in handling unknown scenarios.

3. RELATED WORKS

3.1. INTRODUCTION

Image processing is a very vast field which has experienced significant growth in recent years. It is the application of a set of techniques on digital images to improve them or extract information from them. Images and videos can be collected from some platforms such as (Google, Facebook, etc.) which helps in the development of computer vision Through image processing, the computer can understand and analyze the world around us. Billions of dollars are being invested by developed countries to increase the knowledge of this technology, because the lower the error rate, the greater the gains, which gives better results. This technology can be used in many fields (the field of traffic, health, surveillance,

In the first part of this chapter we will define the digital image and these different Features, then the main image processing techniques.

In the second part, we will define computer vision, machine learning, and deep learning. In the deep learning overview, we mainly describe CNN convolutional neural networks. Then, we will define image classification and object detection.

Subsequently, we present the most used databases as a reference in the development and research of computer vision techniques.

In the last part, we show some areas where computer vision technology and image processing techniques are used.

3.2. IMAGE PROCESSING

3.2.1 Definition of Digital Image

The digital image is an image whose surface is separated into elements of fixed size called cells or pixels, each having Features at a gray or color level. Digitizing an image is the conversion of the image from its analog state into a digital image represented by a two-dimensional matrix of digital values f (x, y), as shown in Figure 1.3. where: x, y are the Cartesian coordinates of a point in the image and f (x, y) is its intensity level The value of each point expresses the measurement of light intensity perceived by the Sensor[1].

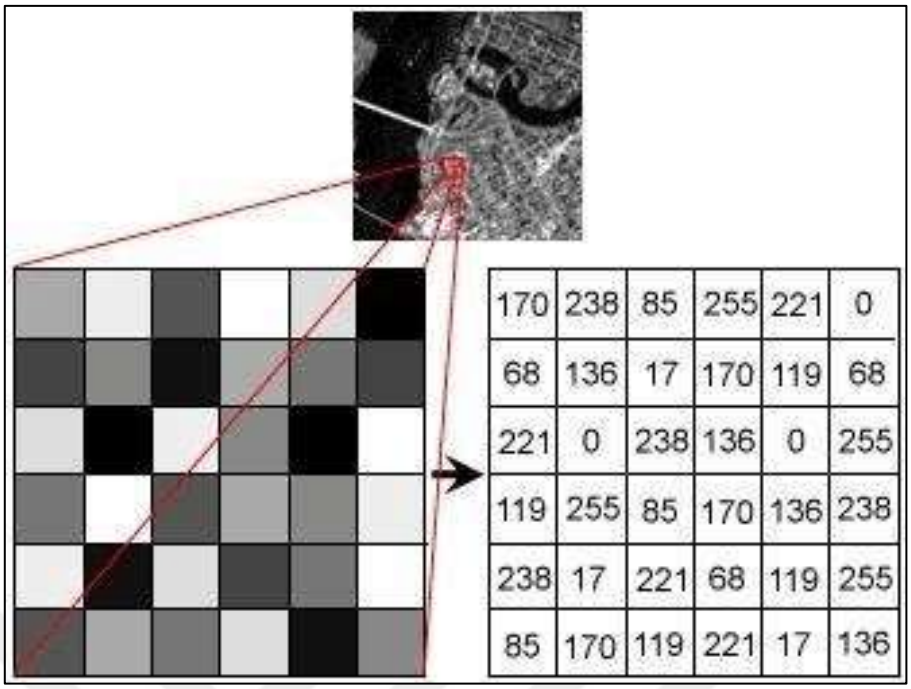

Figure 3.1: Digital Image Representation [1].

3.2.2 Features of Digital Images

a. The pixel

In order to create a digital image, it is necessary to create a collection of points known as pixels. Every pixel in a computer image has a color associated with it, and a pixel is the smallest constituent part of a digital image. A two-dimensional array that makes up the image has each and every one of these pixels stored within it. [1].

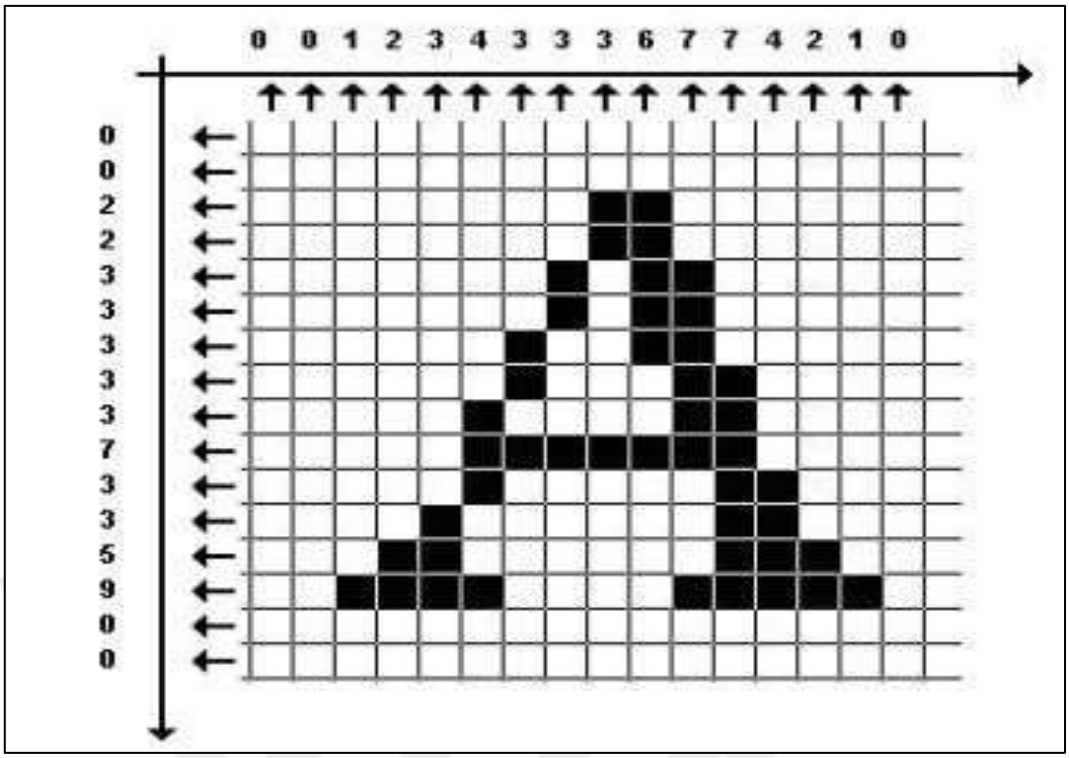

Figure 3.2: The Group of Pixels Forms the Letter A [2].

b. The resolution

We call resolution the number of pixels per unit area, it is most often expressed in dots per inch (DPI for Dots Per Inch), an inch represents 2.54 cm [3].

c. The dimension

The dimension of a digital image is the height and length of the image, and it is measured in pixels. It is provided in the form of a matrix that contains numerical values that are representative of the light intensities (pixels). The total number of pixels in an image can be calculated by multiplying the number of rows by the number of columns in this matrix [2].

d. The depth

Image depth represents the number of bits per pixel, this value reflects the number of grays or color levels in an image, for example [4]:

a. 32 bits/pixel = 1.07 billion colors

b. 24 bits = 16.7 million colors

c. 16 bits = 65,536 colors

d. 8 bits = 256 colors

e. The weight of the image

We can determine the weight of an image based on these two parameters: depth and dimension. The weight of the image is calculated by multiplying its dimension by its depth. For example, for a 640x480 image in true colors, we have the data below.

a. The number of pixels (dimension): 640 x 480 = 307200.
b. The weight of each pixel (depth): 24 bits = 3 bytes.
c. The weight of the image is thus equal to: 307200 x 3 = 921600 bytes [4].

f. The texture

A texture is a region in a digital image that has consistent Features
These Features are, for example, a basic pattern that repeats. The texture is composed of Texel, the equivalent of pixels [3].

g. The noise

Noise or parasite in an image is a phenomenon of sudden variation in the intensity of a pixel compared to its neighbors. Digital noise is a general concept for any type of digital image, regardless of the type of sensor at the origin of its acquisition (digital camera, scanner, thermal camera, etc.) [3].

h. Luminance

Luminance is a measurement of the total brightness of the pixels that make up an image. It is used to describe the overall brightness of the image. There is another way to think about it, and that is as the ratio of the apparent area of a surface to the amount of light that it is exposed to. It is possible to use the term "brilliance," which describes the degree to which an object is dazzling, as an alternative to the term "luminance" when the phenomenon is

observed from a greater distance. When photographs are taken in well-lit environments, the contrast is rather high, and the brightness of the images is reasonably high. When selecting images, it is not useful to choose those that have a very limited contrast range and a tendency to tilt too far toward black or white. The portions of these photos that are either dark or bright show where information is lost during the process [3].

i. Histogram

A statistical graph known as the histogram is utilized to depict the distribution of pixel intensities inside an image. More specifically, the histogram is used to describe the number of pixels that correspond to each light intensity. According to the standard, a histogram is a representation of the intensity level on the abscissa, with the darkest side (left) and the lightest side (right) being the extremes. [2].

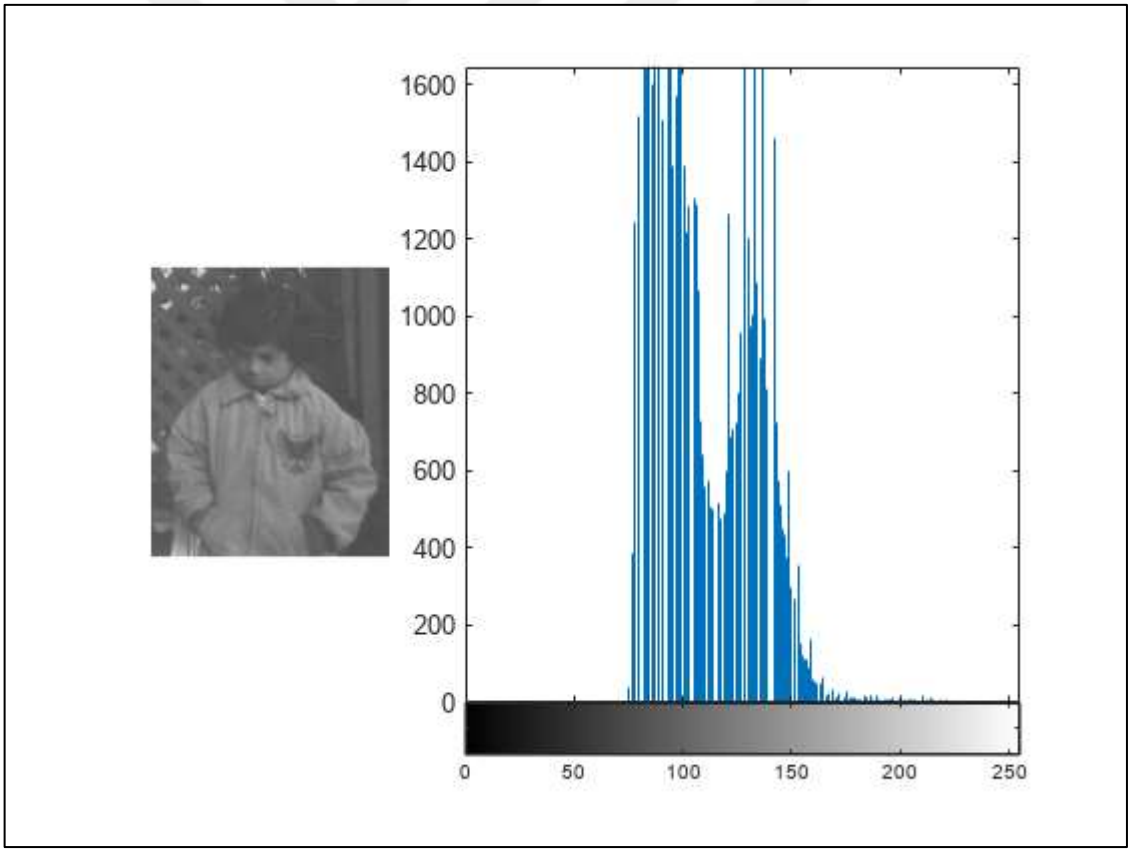

Figure 3.3: Representation of a Histogram of an Image in Matlab With H(X) is the Number of Pixels Whose Gray Level is Equal To x.

j. The contrast

The contrast between two sections of a picture, more specifically between the dark regions and the light portions of an image, is the marked opposition that exists between these two regions. It is the luminances of two different parts of the image that are used to determine the contrast [1].

3.2.3 Standard Image Formats

a. The raster images

A raster image (or bitmap) is made up of an array of points or pixels, each point contains position and color information.
The higher the density of the points, the greater the amount of information and the higher the resolution of the image. This type of image is suitable for display on screen but not very suitable for printing because very often the resolution is low (commonly 72 to 150 ppi for images on the Internet) [1].
Standard raster image formats are BMP (Windows Bitmap), PCX (PiCture eXchange), GIF (Graphic Interchange Format), and JPG or JPEG (Joint Photographic Expert Group) [1].

b. The vector images

The principle of vector images is to represent image data using mathematical formulas. This allows the image to be enlarged indefinitely without loss of quality and to obtain a small footprint. Standard formats for vector images are DXF (Data eXchange Format) and CGM (Computer Graphics Metafile) [24].
For example, to describe a circle in an image, it is enough to note the position of its center and the value of its radius rather than all the points of its contour.
This type is generally obtained from a synthetic image created by software (for example: Autocad) and not from a real object. This type is therefore particularly suitable for image resizing work, cartography or computer graphics [3].

3.2.4 Types of Images

a. The binary image

When each point in a M x N image may only take the value 0 or 1, the image is referred to as a binary image, sometimes known as a black and white image. Black (0) or white (1) is the value of each pixel. The gray level is encoded using a single bit, which is in binary format. Assuming that Ng is equal to two, the relationship between the gray levels can be expressed as follows: p (i, j) = 0 or p (i, j) = 1. [2].

b. The grayscale image

A grayscale image allows a gray gradient between black and white. In general, the gray level is encoded on one byte (8 bits) or 256 shades of gradient. The expression for the gray level value with Gray Level = 256 becomes: p (i, j)∈[0, 255] [2].

c. Color image

A color image is the composition of three (or more) grayscale images on three (or more) components. We therefore define three gray level planes, one red, one green and one blue. The final color is obtained by additive synthesis of these three (or more) components [2].

d. The real value image

For some calculations on images, the result may not be integer, so it is better to define the starting image and the result image as real-valued images. In general, a real-valued image is such that the gray level is a real value between 0.0 and 1.0.

3.2.5 The Main Image Processing Techniques

picture processing refers to the collection of processes and techniques that are applied to a picture with the purpose of either enhancing the image's visual appearance or extracting information that is taken into consideration to be pertinent. It is a collection of activities that are designed to extract qualitative and quantitative information from the image. According to its definition, [4].

a. Acquisition

To facilitate the manipulation of a picture within a computer system, it is essential to subject it to a transformation that renders it both legible and amenable to manipulation within that system. The act of transitioning from an external object, such as the original image, to its internal representation within the processing unit is accomplished by a digitization operation that involves sampling and quantification. Video cameras and digital cameras are frequently employed in contemporary society. Various imaging techniques such as Doppler echo, ultrasound, and scintigraphy are employed in the field of medicine. [4]

b. Filtering

Filtering is an operation which consists of reducing the noise contained in an image by means of algorithms from mathematics through the use of interpolation methods or mathematical morphology [26]. It aims to modify the content of a pixel by taking into account local information, that is to say information extracted from the more or less extended neighborhood of the pixel. Generally speaking, filtering is obtained by convolution of the image with a defined kernel. This kernel can be interpreted as a small image or thumbnail containing a transformation template (linear or non-linear) which is applied to each of the pixels of the image to be filtered to create a new image [2].

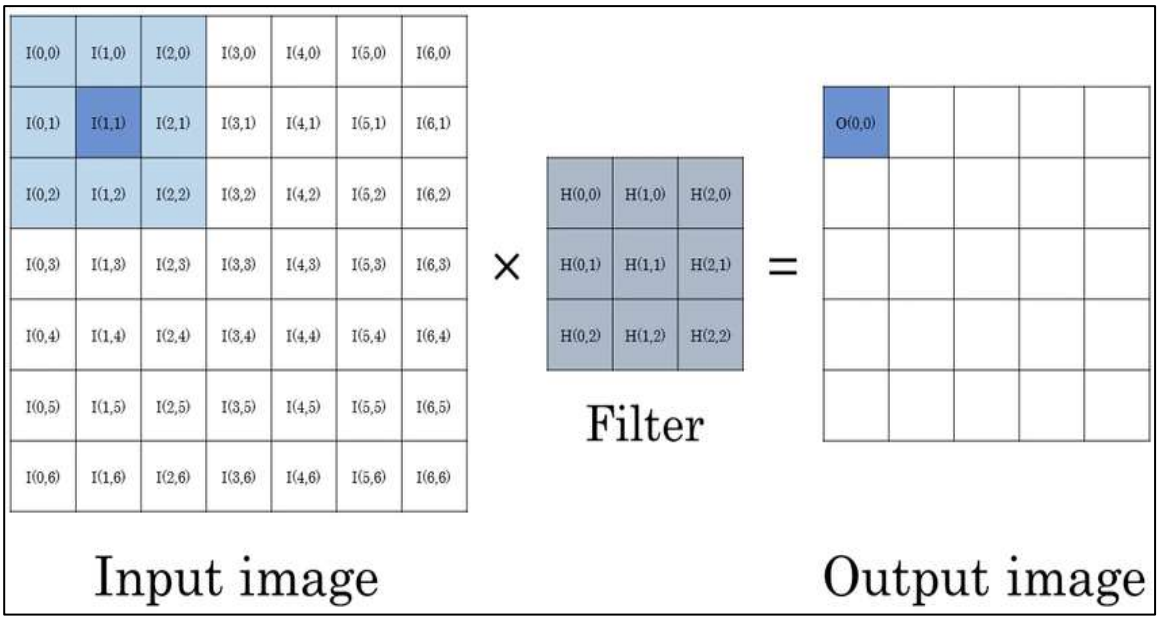

Figure 3.4: Filtering an Image by A Convolution Kernel [11].

c. Linear filtering

These operators are characterized by their impulse response h(x,y) (or h(i,j) in the discrete case, the input-output relationship being given by:

$$S = \Sigma(i = 1\ to\ N)\ \sqrt{Pi} * xi \tag{3.1}$$

For u, v varying from minus infinity to plus infinity.

Here h is a bounded support. A given linear filter will most often be characterized by its kernel, that is to say the matrix [h(i, j)] [5].

The figures below represent the results of linear filtering. Figure 3.5.a represents the original image, Figure 3.5.b represents the application of 7*7 averaging filter and Figure 3.5.c represents the application of 7*7 Gaussian filter.

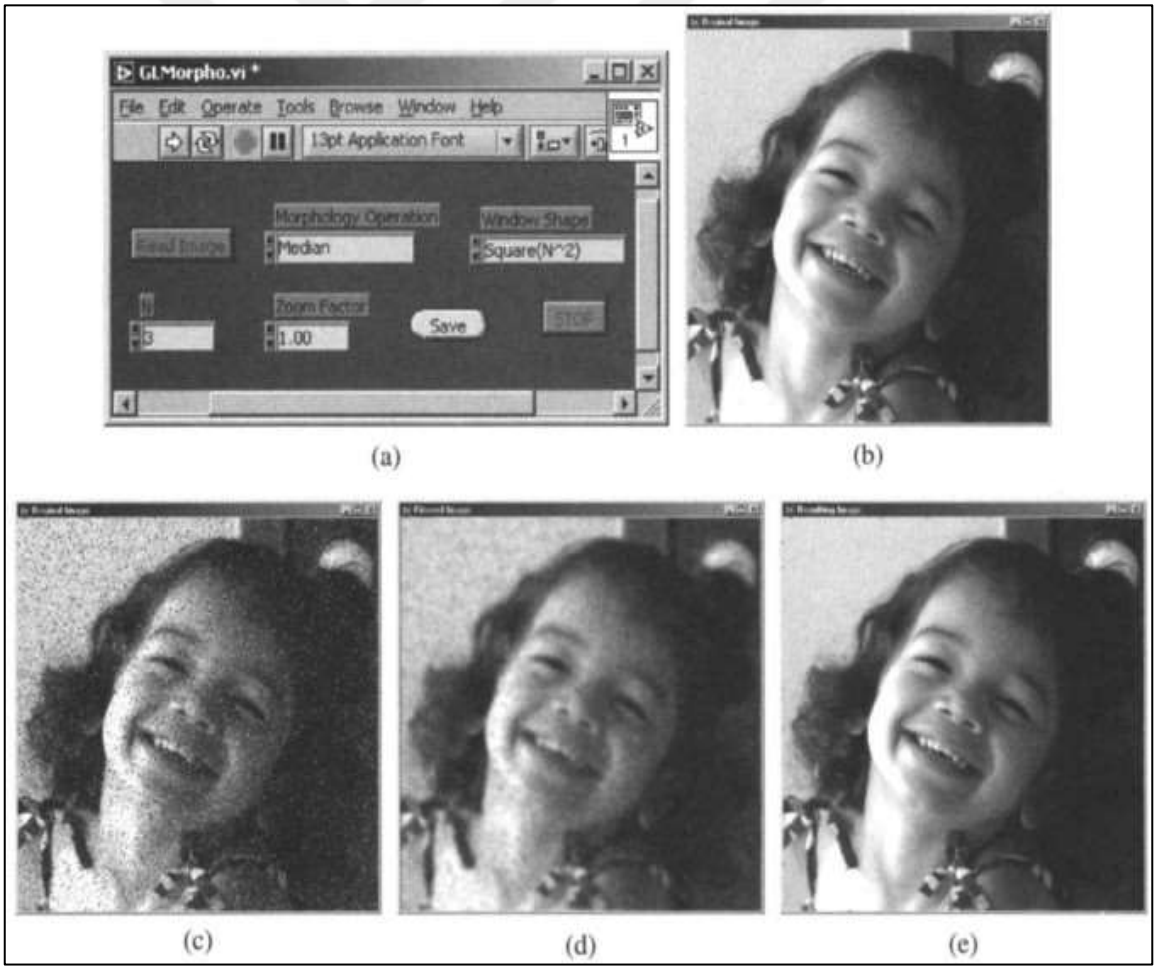

Figure 3.5: The Application of Linear Filtering [5].

d. Nonlinear filtering

Non-linear filtering is an operation which replaces the value of each pixel with a non-linear combination of the values of its neighboring pixels. This type of filter overcomes the major disadvantages of linear filters including the presence of aberrant values even after filtering and the poor conservation of transitions [5].

There are methods that apply nonlinear filtering like the median filter and the nagao filter. In figure 3.6.b, we used a median filter of size 7*7 and in figure 3.6.c, a nagao filter of size 9*9 on a gray level image.

Figure 3.6: The Application of Nonlinear Filtering [5].

e. Image segmentation

In order to divide the picture into distinct areas of interest, we employ segmentation. The first step in segmentation is to divide the target image into smaller parts, or regions. Connected sets of pixels that share characteristics (intensity, texture, etc.) that tell them apart from nearby pixels form regions [2]. Segmentation can be broadly classified into three types: pixel-based, region-based, and edge-based [2].

Figure 3.7: Segmentation of an Image [6].

f. Pixel-based segmentation

The principle is to group pixels according to their attributes without taking into account their location within the image. This allows you to construct pixel classes. Adjacent pixels, belonging to the same class, then form regions. There are methods that use this technique such as thresholding methods and classification methods (clustering). Figure 3.8 shows the results of pixel-based segmentation.

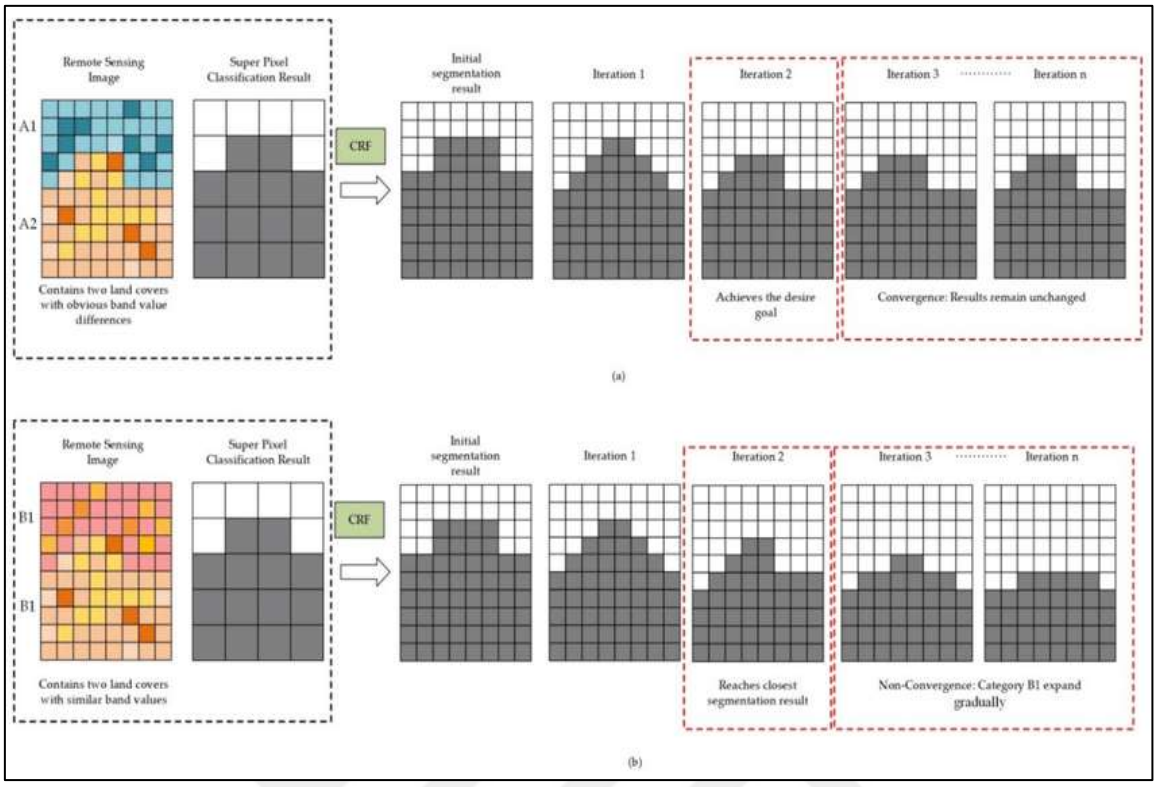

Figure 3.8: Pixel-Based Segmentation [2].

g. Segmentation based on regions

Region-based segmentation consists of partitioning the processed image into homogeneous regions. Each object in the image can thus be made up of a set of regions. In order to produce voluminous regions and in order to avoid a fragmented division of regions, a criterion of geographical proximity can be added to the criterion of homogeneity. Ultimately, each pixel of the image receives a label indicating its belonging to this or that region. There are two families of algorithms for the regional approach:

i Region growing methods that aggregate neighboring pixels (bottom-up methods) according to the homogeneity criterion (intensity and attribute vector).

ii Methods which merge or divide regions according to the chosen criterion (so-called top-down methods) [2].

h. Edge-based segmentation

The segmentation is based on the outline of the object in the image. Most of the algorithms associated with it are local, that is to say they operate at the pixel level.

Edge detection filters are applied to the image and generally give a result that is difficult to use unless the images are very contrasty.

The extracted contours are most of the time cut out and not very precise, it is then necessary to use contour reconstruction techniques by interpolation or to know a priori the shape of the desired object [5].

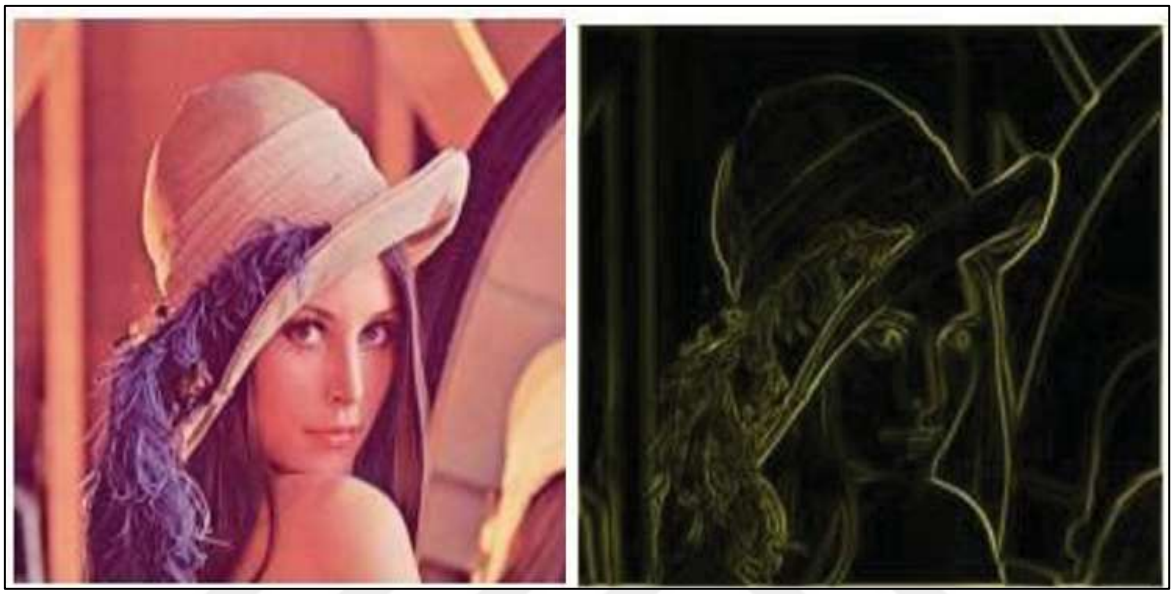

Figure 3.9: Edge Detection on Lena [5].

3.3 COMPUTER VISION

3.3.1 Definition

The goal of computer vision research is to give computers the ability to see, recognize, and process images in a manner similar to human vision, with the goal of producing accurate results. This is the principle that allows AI systems to perceive and comprehend their physical environment. This multidisciplinary area develops and implements techniques for machine learning, computers, and sensors (such as cameras) to mimic and automate these aspects of human visual systems [7]. The main objective of computer vision is to comprehend visual data and transform it into a format that can be utilized in other operations.

3.3.2 Machine Learning

Computers may learn from data, or increase their performance in doing tasks without being explicitly programmed for each one—this is called machine learning. Machine learning is a

subfield of artificial intelligence that relies on statistical methodologies. There are typically two stages to machine learning. One is learning, which is the process of calculating a model from data. The second step is to generate fresh data based on the identified model. This data may then be used to produce the desired task (prediction) outcome [8]. When it comes to picture categorization,

3.3.3 Deep Learning

Machine learning techniques belonging to the "deep learning" category aim to model data at a high level of abstraction by means of various non-linear transformations articulated in architectural frameworks. A description that may be used to describe it would be a multi-layered neural network. [9].

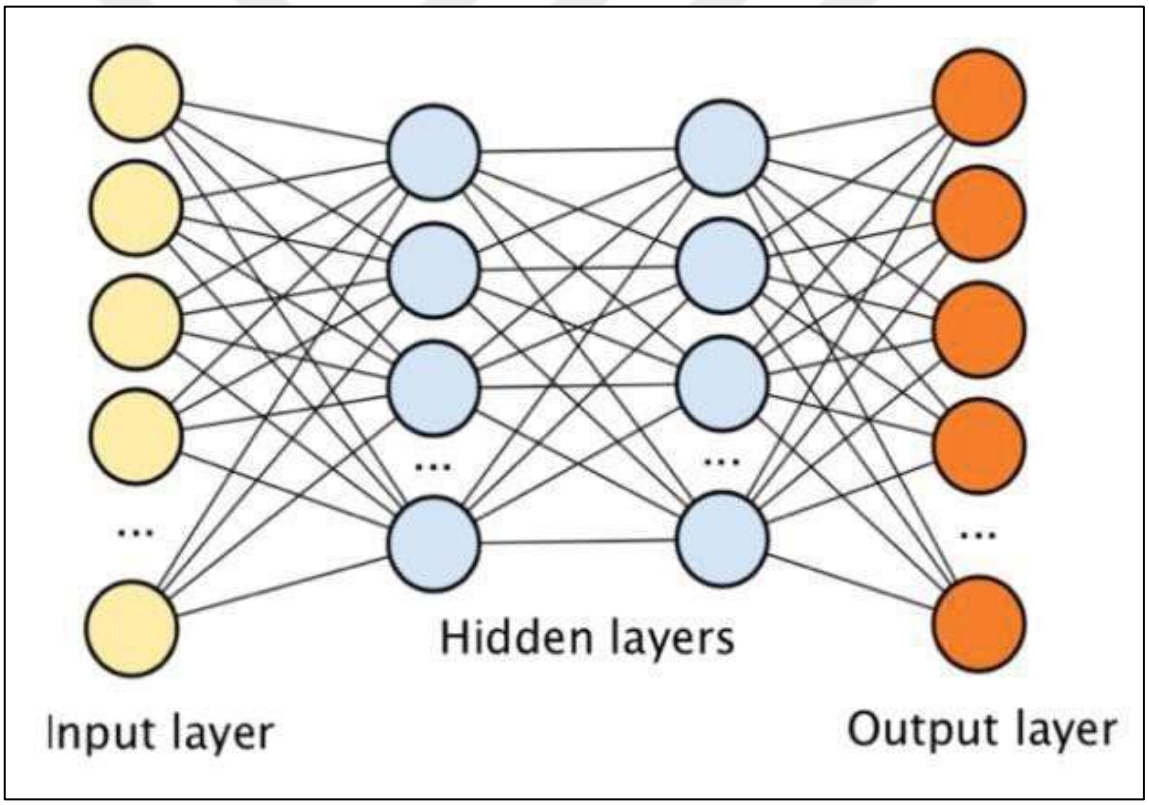

Figure 3.10: Illustration Shows the Deep Learning Architecture [10].

Images that have H rows, W columns, and three channels (R, G, and B color channels) are typical examples of the kind of tensors that convolutional neural networks (CNNs) accept as input. CNNs are used to process images. In order to process images, CNNs are utilized. Conversely, CNNs are able to apply the same logic to higher-order tensors, which is a

significant advantage. A procedure that is sequential is utilized in order to carry out the subsequent processing of the information. Throughout the processing procedure, there are a wide variety of layers that can be utilized that are of different types. Convolution, pooling, normalizing, completely connected, loss, and other layers are some examples of the types of layers that fall under this category.

For the time being, it will be sufficient to merely provide a high-level overview of the architecture of CNN.

For example, in the process of x1 → w1 → x2 →, It is demonstrated that the output of the left-hand function is of the form xL-1. The equation wL−1 → xL → wL gives rise to z.

As was seen in the equation that came before it, a convolutional neural network (CNN) is capable of advancing in a forward pass in a variety of different ways.

In most cases, the input (x1) is an image, which is a tensor of order 3. This is the case in the majority of instances. The first box, which is connected to the first layer, constitutes the location where the processing that takes place takes place. The parameters that are utilized in the processing of the first layer throughout its entirety are represented by a w3 tensor, which is used to represent the parameters.

It is important to note that the value x, which is one of the outputs of the first layer, is also utilized as an input by the second layer. Up until the point where it is completely created, CNN will continue to handle things in this manner. In contrast, the backward error propagation technique requires the insertion of an additional layer in order to acquire the appropriate CNN parameter values. This is done in order to learn the algorithms. Suppose for the sake of argument that there is a problem with the classification of pictures that includes C classes and that it needs to be addressed. A common illustration of a representation of x is the ith element of a C-dimensional vector that encodes the prediction (posteriori). This is an example of a typical representation of x. The posterior probability of x1 is the responsibility of the i-th class, which is responsible for determining it.

When referring to the processing that takes place at the (L - 1)-th layer, the phrase "softmax transformation of xL-3" is a descriptive term that can be utilized. Within the context of this transformation, xL is transformed into a probability mass function.

It is possible for the output of the xL to take on a variety of forms and have a range of meanings depending on the context in which it is applied.

It is a loss layer that makes up the final layer in the overall structure. Utilizing a loss or cost function, we are able to ascertain the degree to which the CNN prediction xL deviates from the ground truth (t). This can be accomplished by applying the function. In this case, the value that corresponds to the input x1 is denoted by the letter t.

Here is just one illustration of what a fundamental loss function may look like:

$$z = \Sigma(i = 1\ to\ xl)\ hi\ \sqrt{Pi}\ xi + N \tag{3.2}$$

There are different Deep Learning algorithms.

i Deep neural networks: These networks are similar to ANN networks but with more hidden layers.

ii Recurrent neural networks.

iii Convolutional neural networks.

In this thesis we were interested in the study of Deep convolutional neural networks, because this deep learning technique is most used in the fields of image classification and object detection [8].

a. Convolutional neural networks

Convolutional Neural Networks (CNN), which are a type of multi-layer neural network, perform incredibly well when it comes to processing inputs in two dimensions. CNN performs exceptionally well in this regard. As a source of inspiration for these networks, the study that Hubel and Wiesel carried out on the visual capacity of mammalian brains served as the source of inspiration. It was in the year 1989 that Yann Lecun and his colleagues were the ones who initially conceived of the notion [11].

There are two distinct components that make up a typical CNN design [12]. The term "convolutional" refers to one of these functions. Performing its duty as a feature extractor for photos, it is effective. Creating convolution maps from an input image is illustrated in Figure 3.11, which shows the technique that must be followed. This process entails putting the image through a series of filters, which are also known as convolution kernels. The outcome of this operation is the image being transformed.

The resolution of the image is decreased by certain intermediary filters because they perform a local maximum operation. This causes the resolution to decrease. In order to finish the process of constructing a CNN code, the convolution maps must first be flattened,

and then they must be concatenated into a feature vector. This ensures that the process is completed successfully.

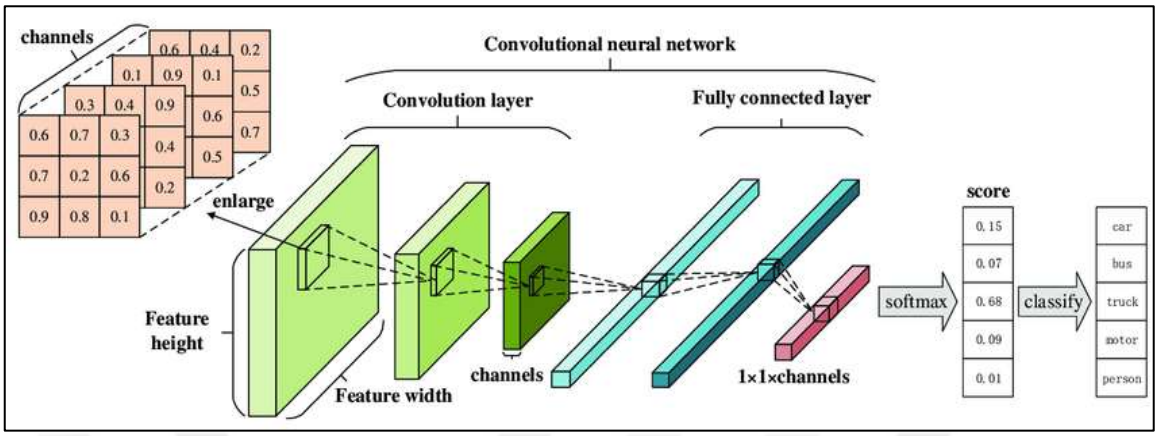

Figure 3.11: Architecture and Composition of A Convolutional Neural Network [7].

A second component, consisting of fully linked layers (multilayer perceptron), takes this CNN code as its input from the convolutional part. This section is responsible for classifying the image by combining the features of the CNN code. One neuron per type makes up the output layer. In order to generate a probability distribution over the categories, the numerical values that are produced are typically normalized between 0 and 1, with total 1.

b. The layers of a Convolutional Neural Network

The construction of a convolutional neural network (CNN) is detailed in this part; it consists of a series of separate processing layers [13] [14] [15]

c. The convolution layer (CONV)

A convolutional neural network (CNN) begins with the convolution layer, which is responsible for processing receptive field data. You can adjust the convolution layer's volume using three parameters. Margin, depth, and pitch.

i Layer depth: number of convolution kernels (or number of neurons associated with the same receptive field).

ii Step: control of overlap of receptive fields. The smaller the step, the more the receptive fields overlap and the greater the output volume.

iii The margin (at 0) or “zero padding”: This margin allows you to control the dimension.

Specifically, maintaining the same surface area as the input volume is sometimes desired. Displayed in is the convolution operation (figure 3.12).

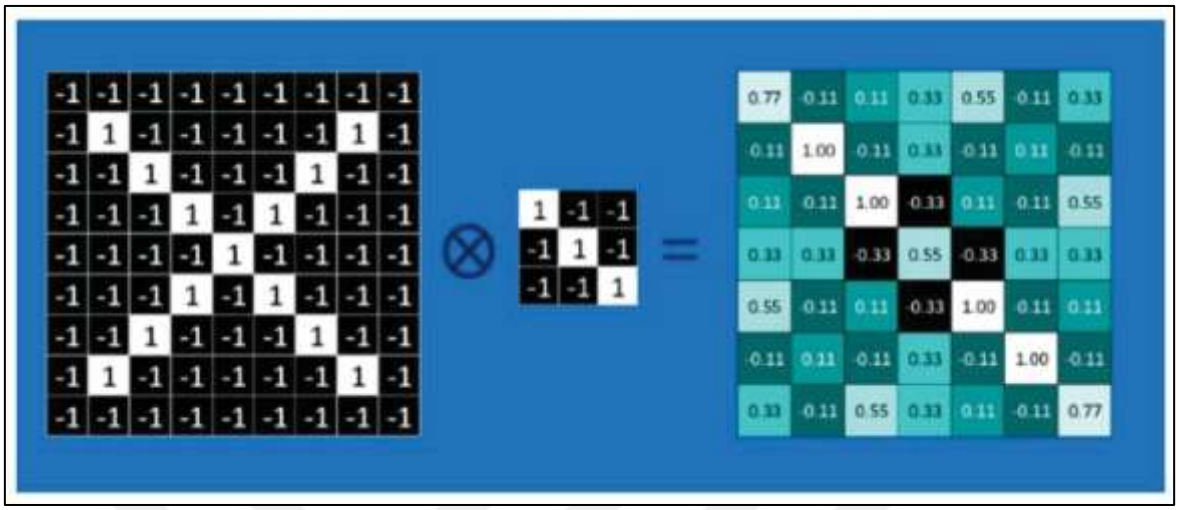

Figure 3.12: Illustration of the Convolution Operation Between an Image and A Filter [13].

d. The correction layer (Relu)

Interposing a layer that applies a mathematical function (activation function) to the output signals between processing layers might increase processing efficiency. With this function, neurons are compelled to provide favorable results.

In a Relue layer, the input dimensions (x and y) remain unchanged. One way to look at the Rectified Linear Unit (ReLU) is as a truncation that is done for each input element separately:

$$0 \leq i < Hl = Hl + 1, 0 \leq j < Wl = Wl + 1 \quad (3.3)$$

$$0 \leq d < Dl = Dl + 1 \quad (3.4)$$

.

There is no parameter inside a ReLU layer, so no need for training parameter in this layer. $di = di + ei$ (3.5)

This formula is the indicator function, being 1 if its argument is true, and 0 otherwise. Therefore, we have

$$T = \Sigma(i = 1\ to\ N)\ log2(1 + SINRi) \quad (3.6)$$

One thing that you want to be aware of is that y can be used as a substitute for x L+1. The fact that the function max(0, x) cannot be differentiated at the point where x equals zero is one of the reasons why Equation B raises a variety of theoretical concerns. Due to the fact that this does not create any issues when it is put into practice, it is safe to make use of ReLU. The major objective of the ReLU algorithm is to achieve an increase in the nonlinearity of the CNN function. For instance, a person and a Husky dog sitting on a garden bench together is an example of the semantic information that may be obtained in an image. There is no doubt that this data represents a highly nonlinear mapping of the pixel values that are present in the original photograph. Consequently, we would like it if the mapping from the CNN input to its output continued to be extraordinarily nonlinear. This is because of the fact that this is the case. It is important to note that the ReLU function is not linear, despite the fact that it seems to be straightforward. The HL WL DL feature, which can be either positive, zero, or negative, is one of the features that can be retrieved by the CNN layers at L−1. This feature can be extracted by the CNN layers. xL i,j,d will be taken into consideration as one of these characteristics. It is possible that xL i,j,d will be positive in the event that a portion of the input picture has specific patterns, such as the head of a dog or cat. If this is not the case, then it is possible that it will be negative or even zero, depending on the conditions. The ReLU layer will ensure that any negative values are set to 0 in order to restrict the activation of yL i,j,d to only those images that contain these patterns in that particular region. This will confirm that the activation was successful. It would appear that this feature is particularly remarkable when it comes to the ability to recognize intricate patterns and objects. When for example "the input image contains a cat," the presence of a feature whose pattern resembles a cat's face, when enabled, gives evidence that is not particularly strong. This is because the pattern of the feature is similar to the pattern of the cat's face. We are able to claim with greater certainty (at level l + 1) that the input image most certainly contains a cat if numerous activated features that follow the ReLU layer have target patterns that match the cat's head, torso, fur, paws, and other elements of the cat's body. This is because the target patterns match the cat's physical characteristics.

e. The pooling layer (POOL)

The practice of pooling, which is a subsampling technique for images, is one of the fundamental ideas that CNNs are founded on. When the rectangles in the input image are constructed, each tile is a representation of one of the n pixels that are utilized to generate the rectangles. The values that are measured for each individual pixel in the tile are what determine the output signal of each tile. Individual tiles are measured individually.

Pooling makes it possible to reduce the number of parameters and processing that are carried out by the network. This is made possible by the fact that it reduces the spatial dimension of an intermediate image (also known as an intermediate image). For the purpose of preventing overfitting, a pooling layer is often placed between two convolutional layers in a CNN design at regular intervals. This action is taken in order to remedy the problem of overfitting an individual.

The pooling layer performs separate processing on each of the input depth slices, and the only thing that is involved in this processing is the reduction of the slices' dimensions at the surface level. A pooling layer that is formed of tiles that are 2x2 in width and height is utilized in the usual setup. This layer is responsible for providing the maximum value that was provided as an output (for reference, see figure 3.13). Within the scope of this discussion, the phrase "Max-Pool 2x2" is utilized".

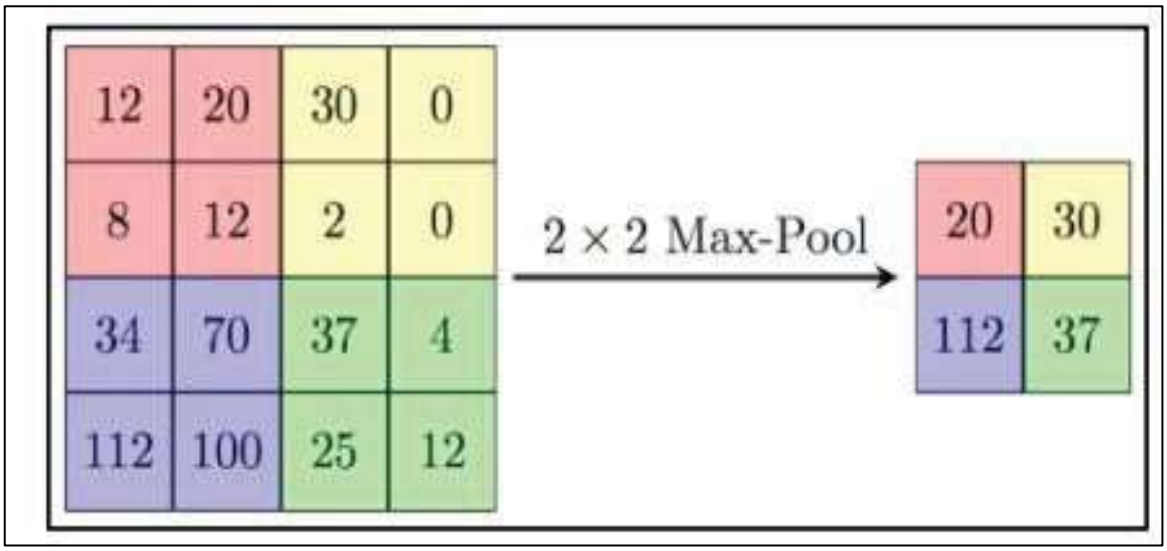

Figure 3.13: Illustration of the "POOL 2×2" Pooling Operation [15].

f. The fully connected (FCN) layer

Fully connected layers perform high-level reasoning in the neural network after multiple layers of max-pooling and convolution. Every output from the layer below is accessible to neurons in a completely linked layer.

g. The loss layer (LOSS, Softmax)

When training a network, the loss layer determines how much of a penalty to apply for each discrepancy between the expected and actual output. Typically, it sits at the very bottom of the network stack. You can utilize different loss functions for different jobs there. To determine the distribution of output classes' probabilities, you can use the most popular "Softmax" function.

h. CNN architectures

More popular CNN architectures are described below [9][15].

i. LeNet

Originally created in 1990 and then enhanced in 1998, LeNet was the prototypical convolutional neural network created by Yann LeCun et al. The most famous and effective LeNet architecture is the one trained to read digits, zip codes, and the like.

j. AlexNet

It was AlexNet, the first famous CNN architecture, that brought convolutional neural networks into the mainstream for use in computer vision. Alex et al. created it. It wasn't until 2012 that AlexNet competed in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), when it finished ahead of the second-place finisher. He finished in the top five with a 16% mistake rate, whilst the runner-up in second place had a 26% one.

k. ZFNet

This convolutional neural network, developed by Rob Fergus and Matthew Zeiler, took second place at the 2013 ILSVRC. Zeiler & Fergus Net, or ZFNet for short, was its original moniker. Its improvements over AlexNet are the result of tweaks to the architecture's

hyperparameters, most notably larger core convolution layers and smaller primary layer strides and filters.

l. GoogleNet

The 2014 ILSVRC award went to this design by Google's Szegedy et al. With a top-five mistake rate of 6.67 percent, it performed almost as well as a human. Although it comprised 22 layers, GoogLeNet's 4 million hyperparameters were far fewer than AlexNet's 60 million).

m. VGGNet

The VGGNet network, created by Andrew Zisserman and Karen Simonyan, was a finalist in the 2014 ILSVRC. The most important thing it did was prove that system depth is a key component of high performance. The most recent top-performing network, VGG16, has 16 CONV/FC layers overall and used a consistent architecture to complete all tasks with 3x3 convolutions and 2x2 pooling.

n. ResNet

With the goal of As a group, they created the ResNet, or residual network. The single-hop connections and batch normalization are key components of this convolutional neural network architecture. Due to the high variety of parameters, evaluating this network is quite costly, which is its primary drawback. But up until this point, ResNet has been the go-to model for ConvNets because it is the most advanced convolutional neural network model. The 2015 ILSVRC was won by him.

3.3.5 Image Classification

The goal of image classification is to identify and assign a specific object's class from a given image. This method typically takes an image of an item as input and returns predicted classes that describe and match the input objects as output.

When we are presented with an image of a dog or any other object, we would like to know which elements are most prominent, which is an example of a classification problem. This means that a classification system should always put this picture in the "dog" category, regardless of the exact placement of the dog, so long as it's the main focus. When the dog

is no longer the main focus, the system should update the image's label to reflect the next main subject [16].

3.3.6 Object Detection

Here, we'll lay out the rules for this area, go over some of the defining characteristics of object detection algorithms that use traditional machine learning techniques, and then show you the pipeline for these algorithms as well as some groups of CNN-based object detection methods.

a. Definition

Finding and labeling items in images are two components of object detection, a computer vision job. In order to accomplish this, we use the object's expected class to create a bounding box around it. That is, unlike in image classification tasks, the system is not merely tasked with predicting the image's class. In addition, it can estimate the location of the object's bounding box.

Object localization—which is necessary for finding and drawing a bounding box around each object in an image—and object classification—which is necessary for predicting the right class of the object that has been located—make this a difficult computer vision task. [7].

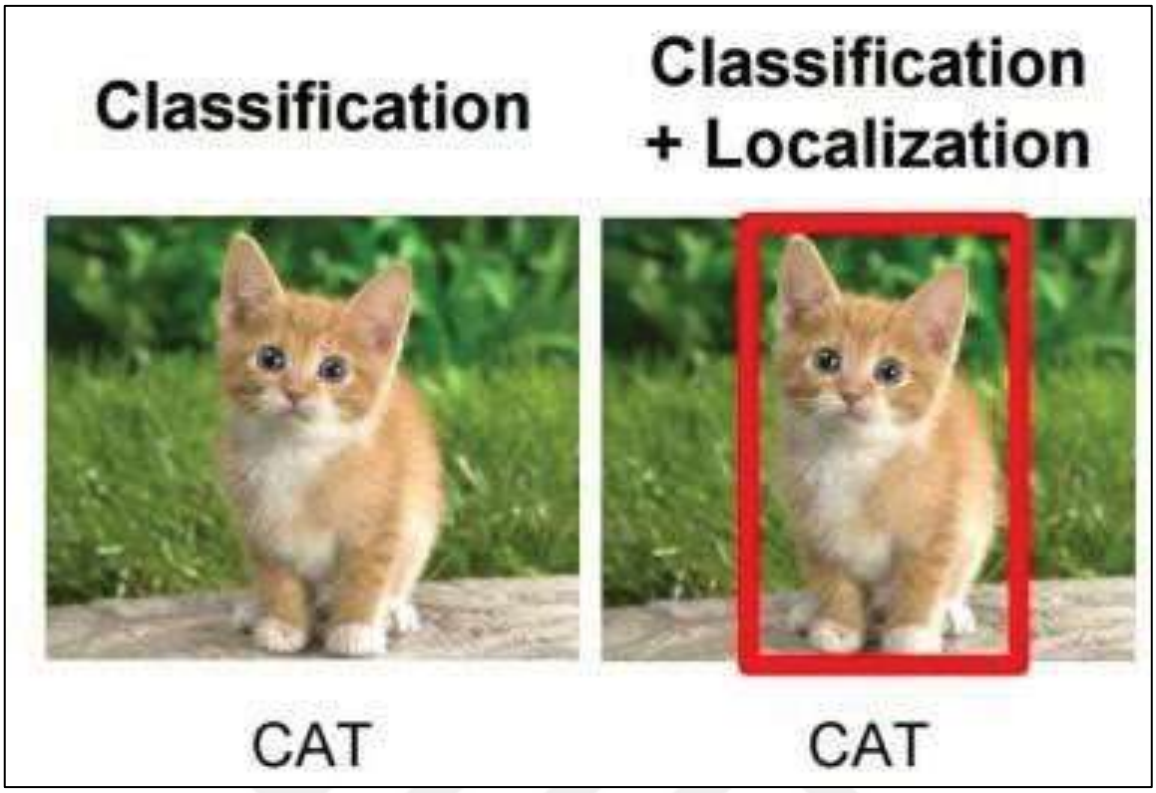

Figure 3.14: Illustration of the Difference Between Classification and Detection [7].

b. Features descriptors

Any piece of data that is useful in completing the computing task associated with a particular application is called a feature. Particular picture structures, like points, edges, or objects, might be used as examples. An picture is processed by an algorithm known as a feature descriptor, which then generates feature vectors. Feature descriptors serve as a "digital fingerprint" for distinguishing one feature from another by encoding relevant information into a string of numbers. For the element to remain recoverable regardless of picture alteration, this information ideally has to be invariant. [17].

c. Histogram Features of oriented gradients

After Robert K. McConnell of Wayland Research Inc. described the histogram of oriented gradients (HOG feature) in 1996, it became widely employed in 2005 in the "pedestrian detection" project by Navneet Dalal and Bill Triggs in a static image. Computer vision and image processing utilize it as a feature descriptor for object detection. This method finds all the instances of gradient orientation in specific regions of a picture [17].

d. Transformation Features of a scale invariant

It was D. Lowe of UBC who initially introduced the ScaleInvariant Feature Transform (SIFT feature) in 2004. Image tracking, object detection, and identification (even while partially obscured) are all made possible by SIFT, which remains constant regardless of the image's size or rotation. [18].

e. Robust accelerated Features

A robust, accelerated function Researchers from ETH Zurich and the Catholic University of Leuven were the first to create the algorithm and feature description descriptor known as "accelerated robust features" (SURF features). It finds application in computer vision tasks including 3D reconstruction and object detection. It takes some cues from the SIFT descriptor, which the authors claim is more resilient and faster across a variety of picture modifications. [19].

f. Features of pseudo-Haar

In their 2001 article published in the International Journal of Computer Vision (IJCV), Paul Viola and Michael Jones introduced a novel approach to facial recognition that they called Pseudo-Haar features. Featuring complete picture sets. Integral pictures can be used to compute pseudo-Haar features rapidly. A 2D lookup table that is built from the original image and is the same size as it is is called an integral image. Every one of its nodes contains the total of all the pixels immediately above and to the left of the current pixel. [20].

Traditional Object Detection Algorithms

a. The selection of regions

Conventional object detection approaches computationally intensively scan the whole image using a series of sliding windows of varying sizes and scales to produce smaller image clips that are subsequently examined independently to ascertain whether an object is contained within the sliding window. [21].

b. Feature extraction

Visual elements that give us useful picture information are necessary for analyzing each image crop produced by the sliding window technique. Features similar to Haar's employed in face recognition and HOG's employed in human detection are two such examples. The problem is that illumination can impact the performance of feature descriptors, and most of them are only meant to recognize a certain sort of item. [21].

c. The classification

Classifying picture elements into a target object class and background follows the acquisition of the feature descriptor vector for each scrolling window [21].
Categories of convolutional neural network (CNN) based object detection techniques
Structures with two stages for sensing entail keeping tabs on a two-step procedure. Prior to classifying each region of interest into specified object classes, the algorithm concentrates on producing a region of interest, which is a recommended area of the original image [21]. Included in this category are the following instances.

d. Faster Regional Convolutional Network (Faster R-CNN)

Using a neural network instead of an algorithm to propose regions of interest is the fundamental premise of a Faster Region Neural Network [24]. neural convolutional network of interest. The Regional Proposals Network (RPN) was one of his notable presentations. Following the collection of regional recommendations, they are immediately input into a largely fast RCNN. Lastly, we incorporate a bounding box rectifier and a softmax classification layer, following by fully connected layers and a pooling layer (see to Figure 3.16 for visual representation). Quicker R-CNN is, in a way, just RPN plus Fast R-CNN.

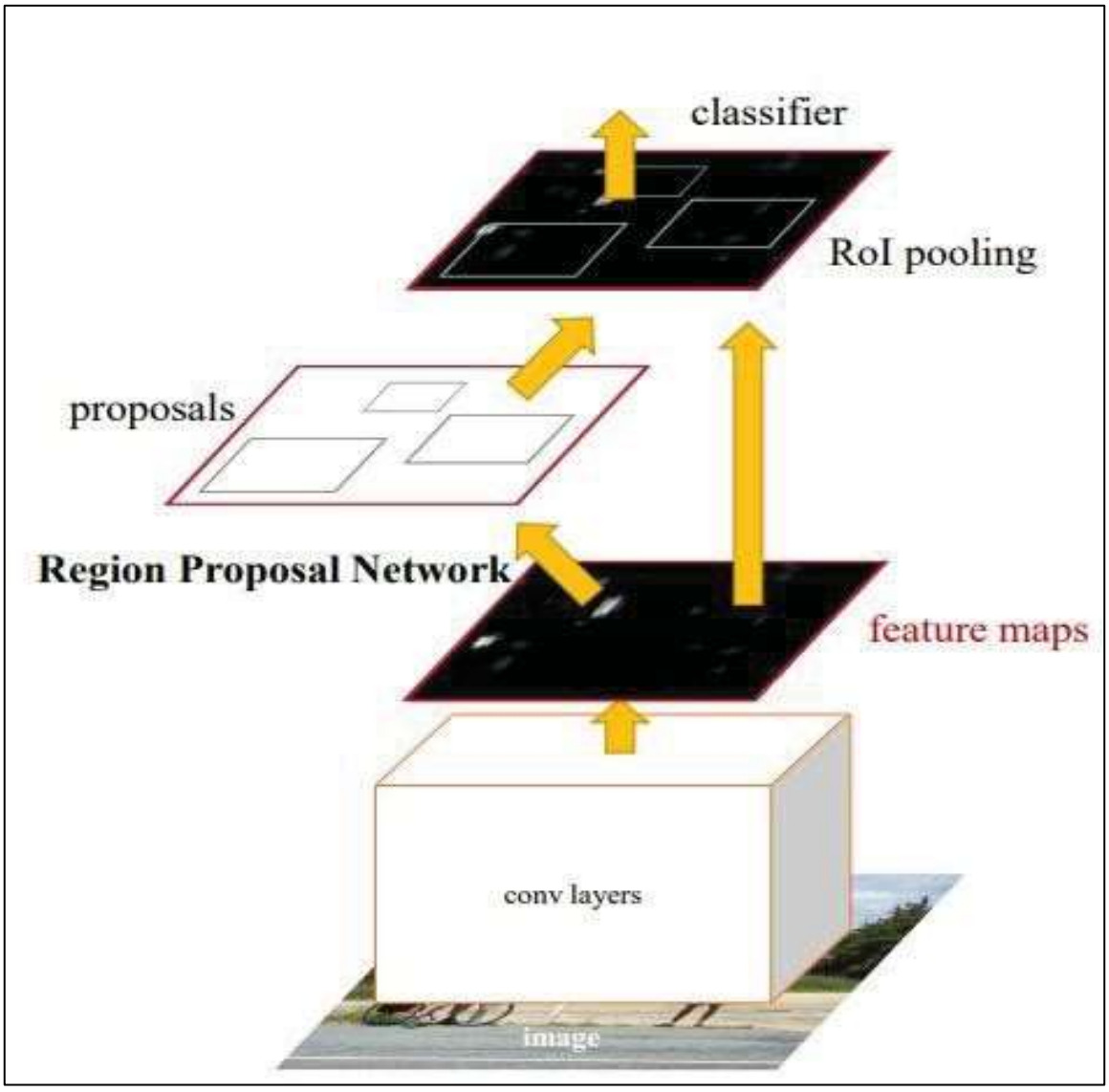

Figure 3.15: Faster R-CNN Model Architecture [24].

e. Regional Convolutional Network Masks (Mask R-CNN)

One such two-stage detector is the mask RCNN, which stands for "mask region neural network" [25]. The initial step involves scanning the image and producing proposals, which are regions that are likely to contain an object. In the second stage, bounding boxes and masks are generated and the proposals are categorized. The fundamental framework is connected to both stages. As a feature extractor, the fundamental building block is a regular convolutional neural network, often ResNet50 or ResNet101. The lower-level components (corners and edges) are detected in the first levels, while the higher-level elements (vehicle, person, sky) are detected in the subsequent levels. (see figure 3.16).

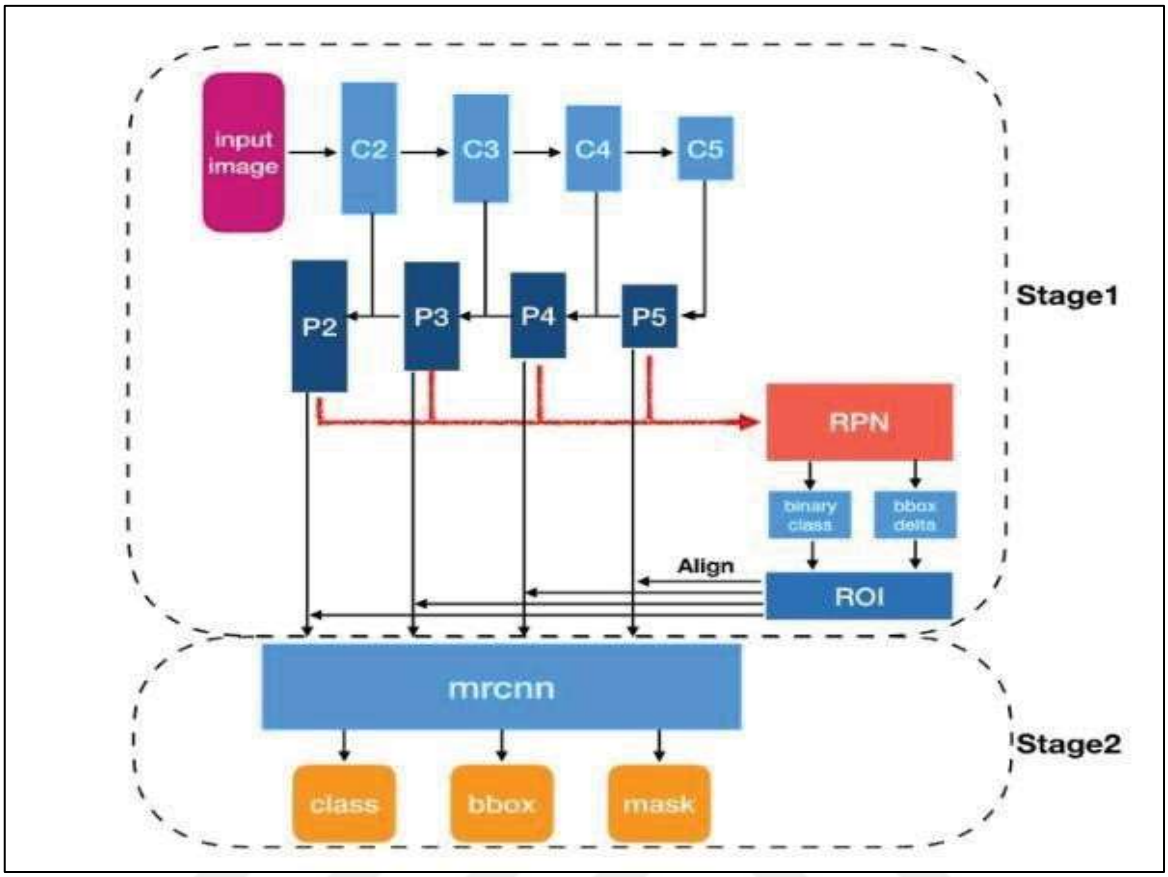

Figure 3.16: Architecture of Mask R-CNN [25].

f. The single-stage detector

With only one convolutional neural network, the single-stage detector can identify every object in a picture with only one pass [21]. The network then uses this information to make predictions about the objects' classes and locations. Compared to two-stage detectors, the accuracy of single-stage detectors is lower, but their primary goal is to increase detection speed. Included in this category are the following instances.

g. You Only Look Once (YOLO)

The YOLO model [49] uses a single network assessment to directly forecast the bounding boxes and probability of each class. The YOLO model's simplicity makes it possible to make predictions in real-time.

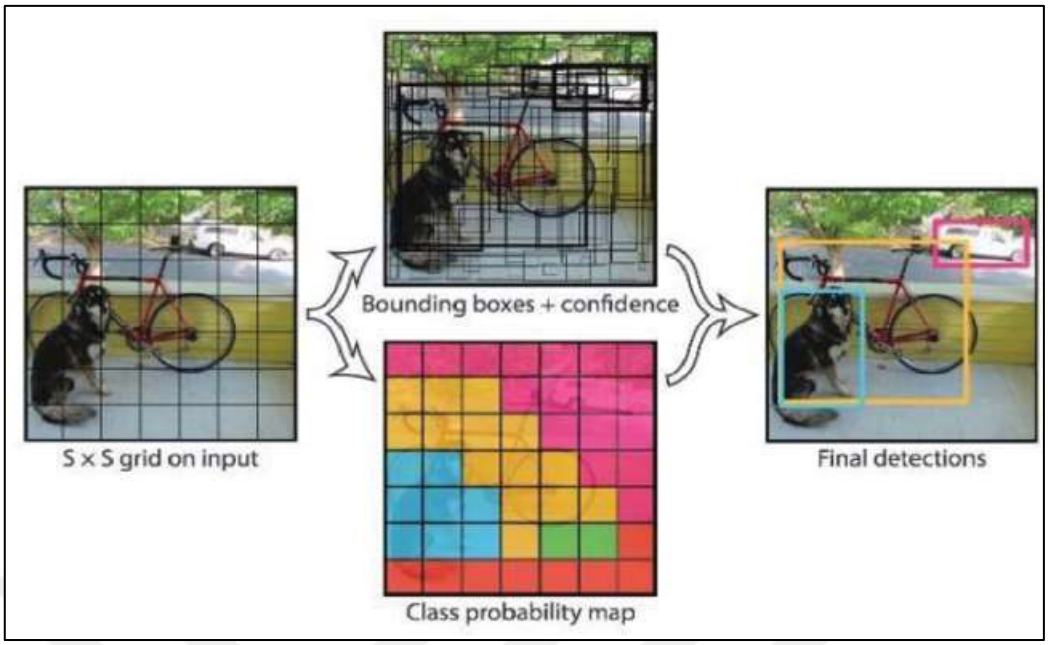

Figure 3.17: Model Yolo.

The model yolo starts by taking an input image and dividing it into a SxS grid. There are B confidence zones surrounding each cell in this grid. To get this confidence score, we only multiply the likelihood of recognizing a certain object class by the "Intersection Junction Overlap" (IoU) index, which is the difference between the predicted constraint zones (the bounding boxes B utilized in each cell) and the actual (marked) images.

h. The Single Shot Detector (SSD)

Created in December 2016 by Wei Liu and his research team, SSD (short for “Single Shot Multibox Detection”) [27] is an object detection algorithm. The name SSD means that the object location and classification tasks are carried out in a single pass through the network, this is achieved using a Multi Box regression technique. Multi Box is a method that provides fast class-independent bounding box coordinates with multiple bounding boxes.

4. PROPOSED METHOD

4.1 INTRODUCTION

It is of the utmost importance for each and every human being to do the necessary precautions to ensure that everyone is safe. When it comes to this matter, the general public has a legitimate expectation that they would arrive at their destination without any issues taking place. Automobile accidents are responsible for tens of thousands of deaths and millions of injuries each year in the majority of nations with considerable traffic, notably those in Asia, the United States of America, and the United Kingdom [1]. These countries are also responsible for the bulk of motor vehicle accidents. When it comes to the transportation industry, the vast majority of injuries and fatalities that take place are documented on routes that are a part of the Interstate system. It is possible that the deployment of better driver assistance systems has the ability to greatly minimize the likelihood of being involved in a collision that involves a motor vehicle.

In order for the SegNet, which acts as the primary CNN for our pipeline, to achieve its maximum level of performance potential, it goes through a process of self-directed training. The vast majority of our hyper-parameters were obtained from earlier implementations of comparable designs that were discovered in the body of published research, such as those that were carried out by T. Chen et al. (2020). The following parameters were applied in the training of the model: an initial learning rate of 102, a batch size of 4, and augmentations as detailed in section 4.2.1. BCEWithLogitsLoss is the name of the loss function that will be utilized by us. This function is part of the optimization package that is included with Pytorch. The "WithLogits" section of the "BCE" program demonstrates that the function is willing to take raw logits rather than probabilities as input. The fact that the function takes the output from the very top layer of the model is a strong indicator that this is the case. The lane class weight in our loss function is set to 38 in order to account for the large size difference that exists between lane pixels and non-lane pixels. This was done so that we could take into account the existence of both types of pixels. In order to assess the significance of lanes, the total number of training set pixels that are not lanes is divided by the entire number of training set lanes. In addition to this, we make use of the ReduceLROnPlateau learning rate scheduler as well as the Stochastic Gradient Descent (SGD) optimizer that is contained

within the PyTorch package. When the validation loss is getting close to a plateau, a learning rate scheduler can automatically slow down the learning rate to prevent the model from getting stuck in a suboptimal local minimum. This is done to prevent the model from becoming unstable. This prevents the model from becoming stagnant and failing to advance. In particular, we keep a close eye on the validation loss, and if it does not improve after four epochs in a row, we decrease the learning rate by a factor of 0.05. This is done so that we can continue to train the model. This is done in preparation for the possibility that the validation loss will not get better. The SegNet model that we are using consists of a total of 29443587 trainable parameters and has been trained for a total of 30 epochs. Both the SegNet foundation and the full pipeline can make use of the training-time loss and assessment metric charts that are provided in Appendix A. In addition, in order to train our SegNet backbone and pipeline models, we made use of Google Collab's NVIDIA V100 GPU RAM, which is outfitted with 32 gigabytes (GB) of VRAM memory. This was done so that we could train our models more quickly. In the following part, under the heading "Results and Analysis," you will find a comprehensive explanation of the training timetable for our pipeline, which takes into account all of our research. In this area, you will also discover descriptions of the many hyper-parameters that were used throughout the testing, in addition to the outcomes of those studies. You can access this section by clicking on the link provided. The following diagram, which will be referred to as Fig. 4.1, provides a visual representation of the full process flow of our proposed pipeline for lane detection.

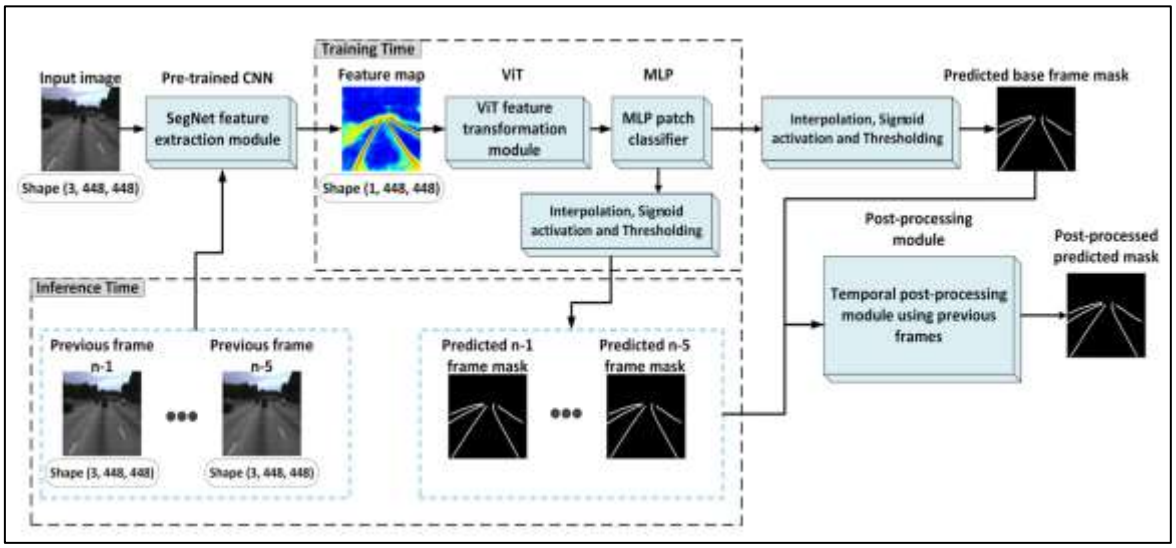

Figure 4.1: The Proposed Lane Detection Method.

Among the many applications of the vision-based lane detection principle is lane detection. Just like its name suggests, it's a way to identify and track the lanes that ground traffic uses. Scientists and engineers from all around the globe have been focusing on this practical technology for the past few decades.

4.2 SYSTEM OUTLINE

With just one dashboard-mounted camera, a lane-position detection system can mainly use image processing methods to pull lane markers from recorded video. We are excited to use these image processing techniques for our project so that we can: · Determine the car's current lane of travel. Show the part of the road directly in front of the vehicle Assume you are driving in a certain lane and try to anticipate the car's next move. Calculate the required steering angle. The MATLAB software was crucial in the development of the autonomous driving system project.

4.3 SYSTEM CONSTRAINTS

Our project is limited in its breadth due to some restrictions, such as the fact that the program can only conduct its operations with a pre-recorded video.

a. The application could run into problems if the lane markers are worn or nonexistent.
b. If the curvature of a curve is really high, the software might not be able to detect it.
c. When the application is running, changing lanes could cause software issues.

4.4 DESIGN AND TESTING:

It is necessary to extract the image frames before lane detection in a video may begin. The next step is to analyze the pulled frame for lane boundaries and steering/turn prediction information.

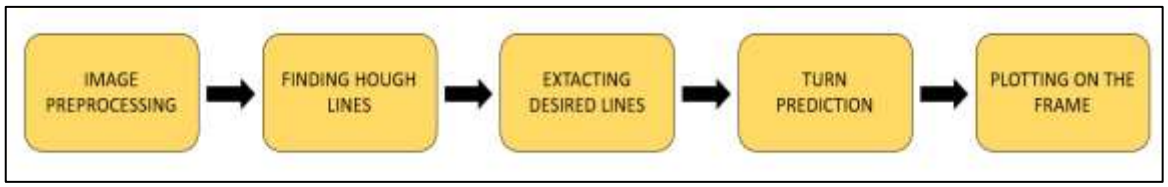

Figure 4.2: Flowchart of Image Processing.

A reference was made to the GitHub URL provided under References [42] in order to gain a better understanding of the necessary procedural processes. The methods for calculating and visualizing the Hough Lines were lifted from the aforementioned GitHub repository. It should be noted that the referred code and this project employ different approaches to finish the integral tasks, such as obtaining the needed lines, calculating the turn prediction, and steering angle.

There are five main phases of the image processing:

4.4.1 Image Pre-processing

Pre-processing the frame is necessary for line detection because it removes things that aren't needed. To eliminate noise, the image is initially twisted using a gaussian kernel.

Figure 4.3: First Frame of the Video (Left) Is Processed to the Gaussian Filtered Frame (Right).

We create two binarized images by determining the RGB threshold values of the yellow and white lines, and then we utilize these to isolate the yellow and white thresholds, respectively. The color picker feature on Paint was used to establish the threshold values.

Figure 4.4: Binarizing for Yellow (Left) and White (Right) RGB Thresholds.

By utilizing Canny edge detection, the pixels with the highest gradients were identified as having the edges.

Figure 4.5: Edges for Yellow (Left) and White (Right) Binarized Image.

In order to separate the target area, two masks were made. The yellow and white lane edges make up the project's zone of interest. The roipoly function was used to extract the masks. To further isolate the lanes, we multiply the edge pictures (using dot multiplication) with the mask image. This yields two edge images, one each of the yellow and white lane edges.

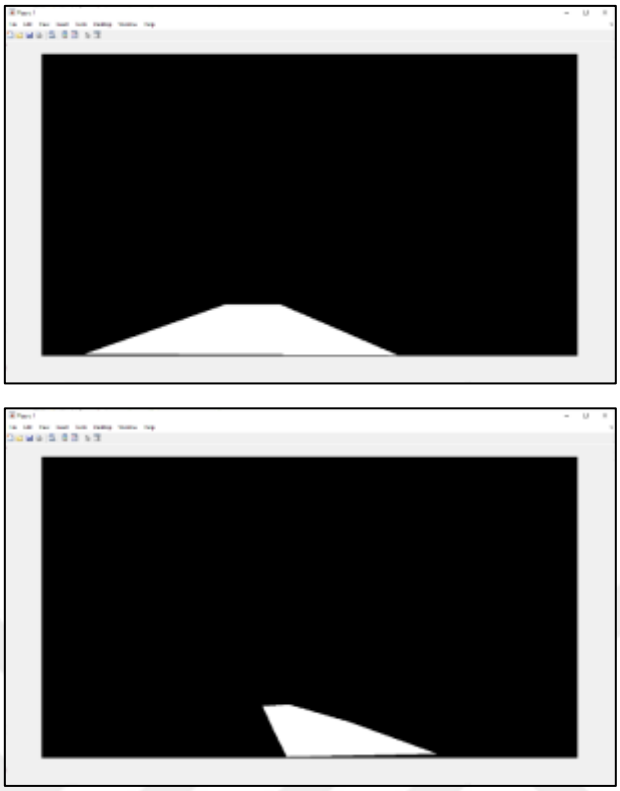

Figure 4.6: Region of Interest Masks, to Extract Yellow (Left) and White (Right) Lane. The ROI is Determined From the First Frame (Figure 1).

To further isolate the lanes, we multiply the edge images by the mask image; this yields two edge images, one each of the yellow and white lane edges.

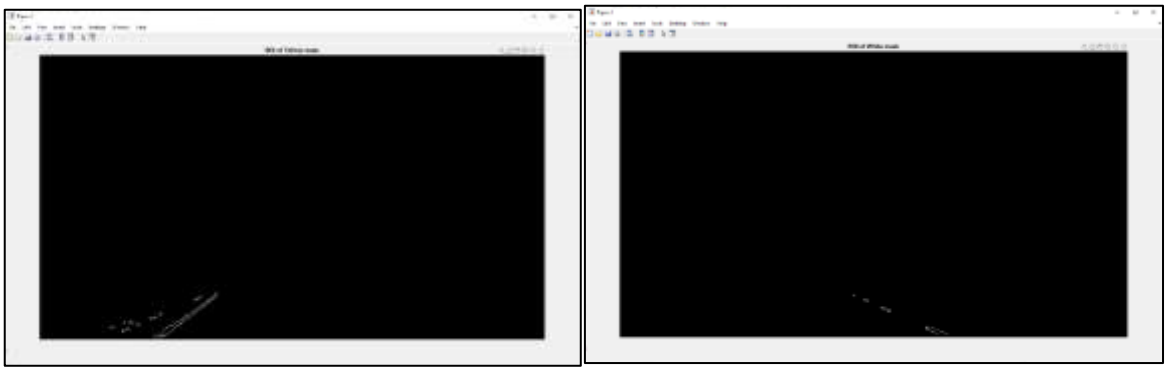

Figure 4.7: Isolated Edges of the Lanes. Yellow (Left) White (Right).

4.4.2 Finding Hough Lines

By finding the equation of a line in polar coordinates, the Hough lines are utilized to extract the lane margins.With the x-axis representing theta and the y-axis rho, the hough function

depicts the values of the Hough transform. We may see the rho value for various theta values plotted for each x-y coordinate.

Next, we get the intersection points of the Hough lines from the houghpeaks function. We choose two intersections/Hough summits. The purpose of this is to guarantee that the very edge of the lane is constantly picked up. The beginning and ending points of the line, as well as their related rho and theta values, are then determined using the houghlines function.

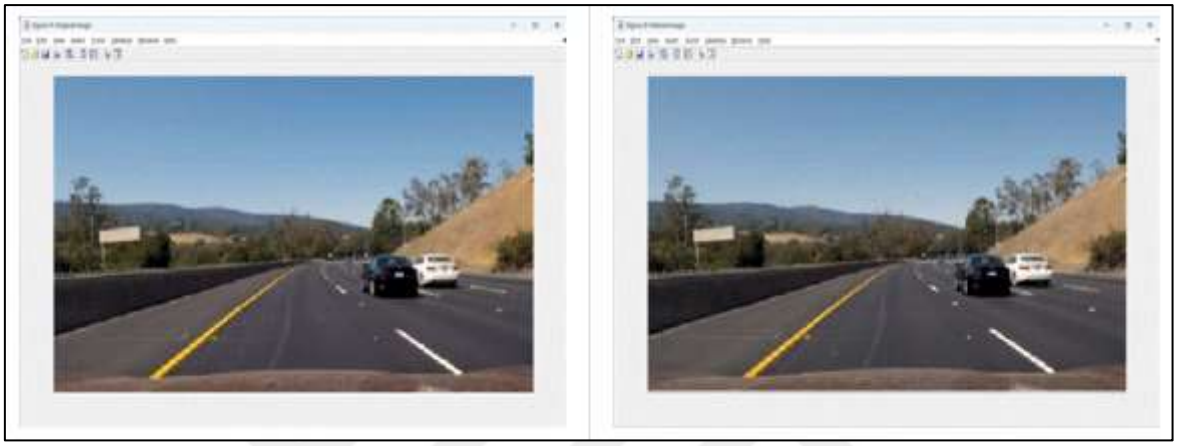

Figure 4.8: Hough Lines For Yellow and White Lane Edges.

4.4.3 Extracted Lines

There are two lines that are provided for each lane edge, and once the lines for each edge have been obtained, it is important to select one of the two lines that are available. Performing this action is required for every lane advantage.

Instead of changing the values of the coordinates, we are extending the lines to the bottom of the frame and selecting the lower coordinates of the lines based on the coordinate that is closest to the inner lane boundary. This is being done in place of changing the coordinate values. What we are doing is exactly this.

Following a comparison of the other coordinates, the top location is the one that depicts the line that is the longest. This position is picked after further consideration.

The coordinates that were chosen for the edge of the lane were acquired from the two sites that were discussed earlier in the conversation. One is able to determine the equation that characterizes the line by making use of the coordinates. Following that, the equation that is positioned at the line is employed in order to determine the coordinates that are situated at the bottom of the frame.

What we do is to compare the top y locations of the two new lines that have been added. One of the y-coordinates that is utilized for both of the locations is the one that represents the greatest distance that exists between the two locations. It is necessary to compute the x-value of the lower point of the two that correlates in order to ascertain the value.

According to this, the line that is supposed to be drawn along each lane edge has the same height at its bottom coordinate and the same height at its top coordinate in regard to the frame. This is the case since the frame is of the same height.

4.4.4 Plotting and Determination of Direction and Steering Angle

In order to determine the direction in which the steering wheel of the vehicle will turn, the turn prediction is utilized. Finding the center coordinate between the bottom coordinates and the point where the lines cross is the approach that is applied to determine the direction in which the steering is to be performed. This enables the steering to be performed in the desired direction.

In order to create an accurate prediction regarding a straight turn, it is essential that the intersection point be located in close proximity to the geographic center. In the event that the intersection point travels to the left or right of the center coordinate, the direction that is created is either left or right, depending on the direction in which the point traveled. It is also possible to see it from a different angle by taking into consideration the ratio of the x-coordinates of the junction point to the central coordinate that is shown in red and is situated at the bottom of the frame.

The steering ratio is considered to be 12:1, which indicates that the angle of the car in proportion to the steering angle [3] is taken into consideration. For the purpose of determining the turning angle of the automobile, trigonometric ratios are applied. The turning angle is the first angle that needs to be determined before the steering ratio can be applied to that angle in order to compute the steering angle.

It is discovered that the straight thresh hold can be anywhere between 0.9 and 1.1 when all of the frames are taken into consideration. Any figure that is lower than the threshold limits or that is closer to the grates is indicative of a turn to the left or right. However, a turn to the right is also possible. It is also possible to plot a polygon by making use of the coordinates that were extrapolated during the third step of the production process. Because to the

employment of the patch function, all of this is kept in good condition. The next step is to save the rendered frame to the object that represents the output file. This is the next stage.

Figure 4.9: Plotted Frame. This Frame Will Be Written to the Output File.

Our endeavor is only a stepping stone on the route to our long-term goal of developing code that will enable vehicles to drive themselves, but our long-term objective is to write code that will enable cars to drive themselves. The detection of lanes, which is an early component of autonomous driving, is the primary focus of this research being conducted. It was necessary to make use of the difficult adjustment in order to accomplish all of the goals that were outlined. When evaluating each frame, the typical amount of time that is required is around 0.1639 seconds. Because the video contains a total of 790 frames, the processing of those frames took roughly 129.5 seconds (790 equals 0.1639). This is due to the fact that the movie contains a total of those frames. In the course of the work that will be done in the future, it is recommended that a formal evaluation of the algorithm's performance be included. Furthermore, by applying the algorithm to a number of different video sequences, we are able to evaluate the dependability of the approach. We are of the opinion that the identification method that was utilized in the automotive study is sufficiently reliable; however, we are of the opinion that developing more sophisticated tracking systems could be beneficial for conducting research in the future.

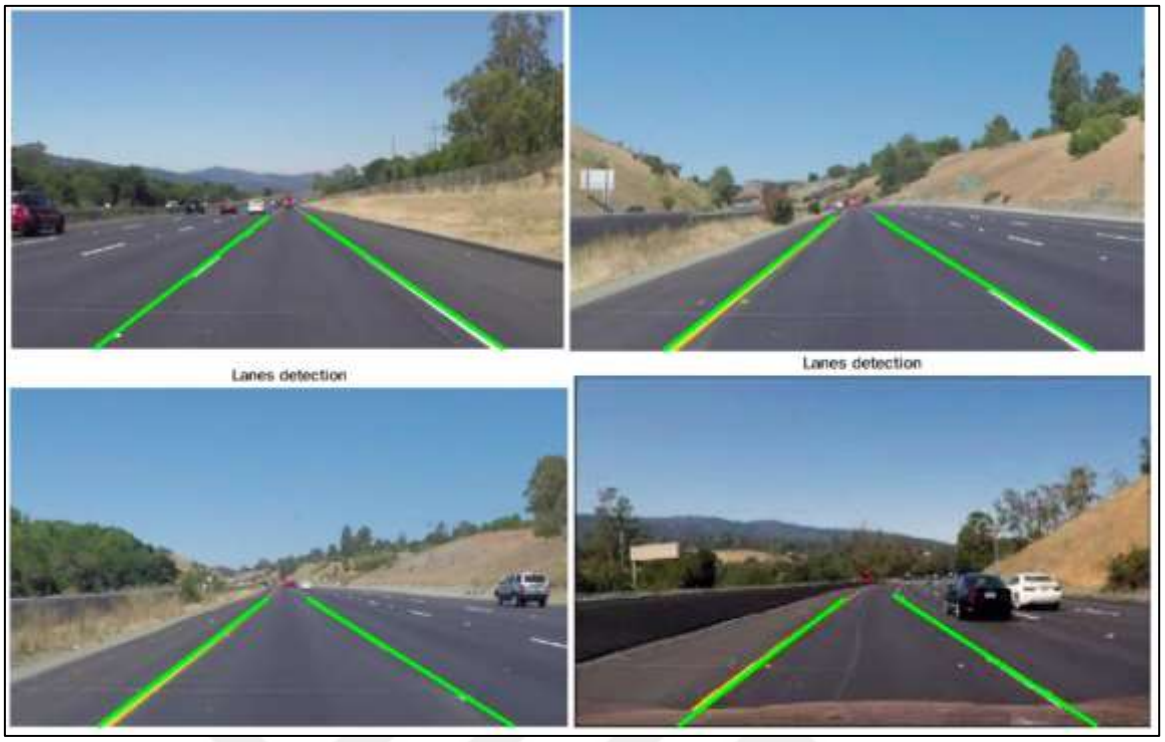

Figure 4.10: Segmented Output of the Lane Sides.

4.4.5 CNN Turn Prediction

The pipeline begins with a single training session for the CNN backbone model, which is known as SegNet. This session is performed in isolation. This topic was discussed in the part that came before this one. After the initial training has been completed, our post-processing system is evaluated in a variety of settings to determine how well it performs under those conditions. As a consequence of this, the response to the second research question that we had is pertinent to the discussion of this. In this section, we study the outcomes of putting our model through its paces with and without making use of our temporal post-processing technique. Specifically, we compare the results of these two scenarios. Table 4.1 is where all of the different measures that were gathered from our SegNet model can be found presented. We put our temporal postprocessing mechanism through its paces by using a test set, and the findings reveal that we get the best results from it when we give it between three and five frames of previous data to deal with. This was determined by the results of the test set, which showed that we get the best results from it when we provide it with between three and five frames of previous data to deal with. It is vital to keep in mind that each frame is separated from the next by a time interval of 100 milliseconds; as a result, accessing 5 frames

in the past will take you back half a second. It is important to keep in mind that each frame is separated from the next by a time gap of 100 milliseconds. When it comes to recognizing the lanes of traffic on a road, the optimal length of time that has elapsed between the base frame and the most recent frame that preceded it may differ from one circumstance to the next and from one application to the next. As a consequence of this, we think that it is appropriate to choose a time range that falls between 0.3 and 0.5 seconds for the frame that comes before it. This is because we believe that it is appropriate. There is a good chance that drivers require only a half second of additional temporal context in order to perceive the lanes more clearly or to make a turn while there are automobiles going around them.

Table 4.1: FNR and PNR in Each Image Frame.

	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

Now main part after all processing we have to predict so I use built in function name cross to write left right point also consider all value which we calculate like theta rho and also value of saturation hue etc.Here we calculate slope for better results

slope = (y-2-y-1)/(x-2-x-1).

Where x1 x2 y1 y2 are the dot product of xy with

a. x-1 = xy(1,1).
b. y-1 = xy(1,2).
c. x-2 = xy(2,1).
d. y-2 = xy(2,2);

If we get negative slope, then our left prediction code is execute otherwise right prediction code is running

Figure 4.11: Turn Prediction of the CNN-SVM.

4.5 RESULT AND ANALYSIS

So, as you see after the all processing we can say that our algorithm is properly working it also detect the lane properly. It only works on highways as we discuss in introduction our proposed system is only for lane detection not for proper driving assistant system we further working on it also the one more thing we add is detect lane on first because our algorithm made for first lane but it also work on second lane of road just by few changes.

The results of the experiments are shown in Table 4.2, which shows how the overall performance of the SegNet backbone model may be significantly enhanced by making use of temporal information. The best results are obtained by using temporal information from 5 different frames, with an F1 score of 0.52, an IoU score of 0.355, and an accuracy of 98.7%. The accuracy of the results is taken into consideration while assigning these scores. This demonstrates that making use of a more extensive window of frames enables the model to better capture the temporal dynamics of the input data, which in turn ultimately leads to more accurate predictions. In spite of the fact that the frames per second (FPS) will be decreased, the performance hit will still be manageable for a wide variety of applications.

Table 4.2: Parameters for the SVM Classification.

Sr. No.	Parameters			Percentage Accuracy (SVM)		Percentage Accuracy (Hybrid CNN-SVM)	
	Gamma	Degree	Decision Function	Training	Testing	Training	Testing
1	1	3	one-vs-one	97.80	97.52	98.80	98.82
2	1	3	one-vs-rest	97.38	97.74	98.48	98.74
3	1	5	one-vs-one	96.18	96.37	97.38	97.39
4	0.1	3	one-vs-one	97.95	97.85	98.95	98.95
5	0.1	3	one-vs-rest	98.93	97.84	98.93	98.84
6	0.1	5	one-vs-one	98.35	97.68	**99.28**	98.88

Although it work properly but it not detect lane on highways where tree or more traffic is here but we are working on it as you see in above image they properly show the direction where to turn the car. Its accuracy rate is 98%, it work properly on every input video where we just want to detect lane and show on screen.

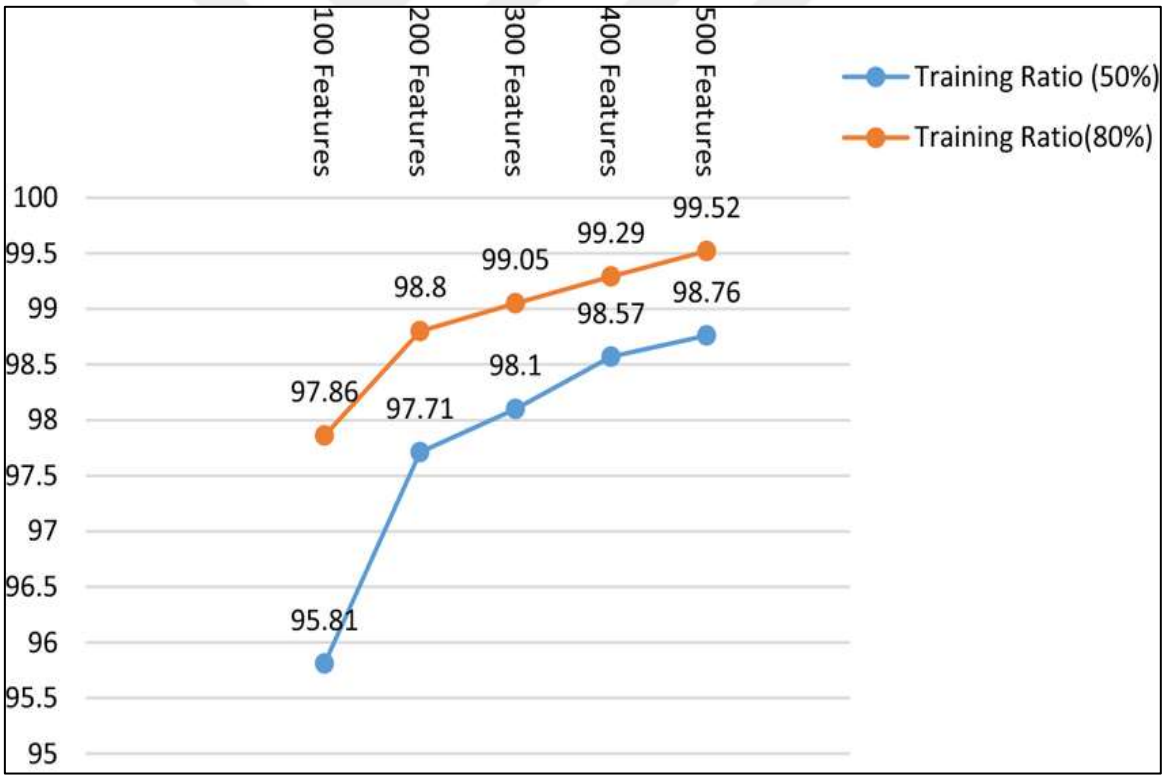

Figure 4.12: Training Accuracy Ratios.

According to the findings of our experiments, there was a cluster of points located in the vicinity of the vanishing point, which is also referred to as the horizon line's apparent

convergence point. When the lanes are packed together too closely, there is also a significant rise in the amount of uncertainty that occurs.

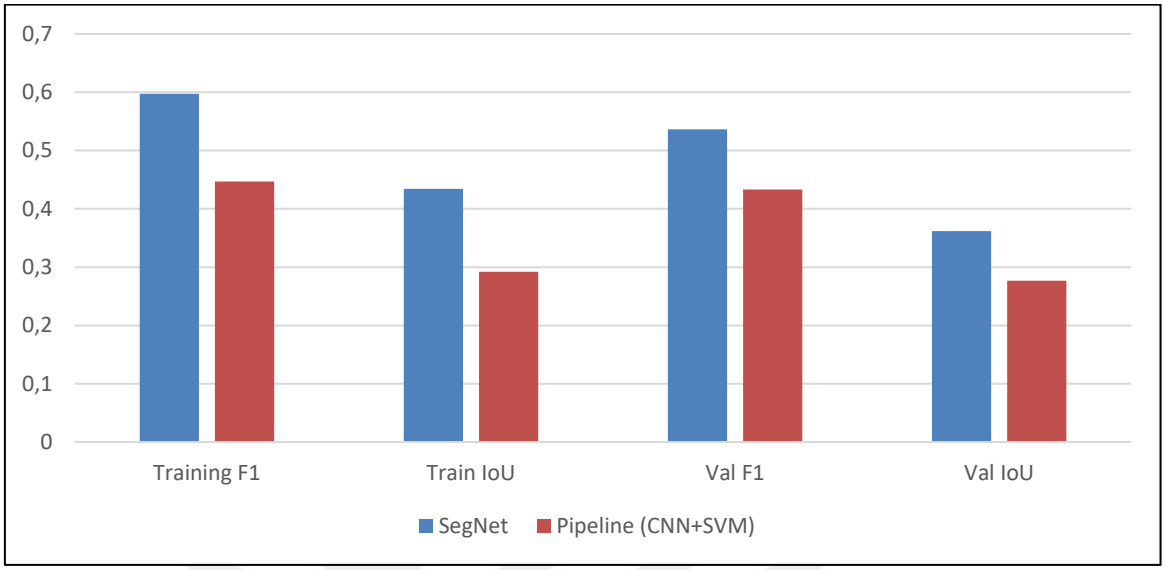

Figure 4.13: Training and Validation Accuracies.

The key factor that contributes to this behavior is the fact that our backbone model, in contrast to the ground truth or state-of-the-art models, is unable to provide forecasts that are almost flawless. Utilizing additional noise reduction and post-processing approaches, such as a higher threshold or ROI (Region-of-interest) segmentation, may have enabled our model's performance to be enhanced while at the same time minimizing the consequences of its inherent constraints. This was possibly the case.

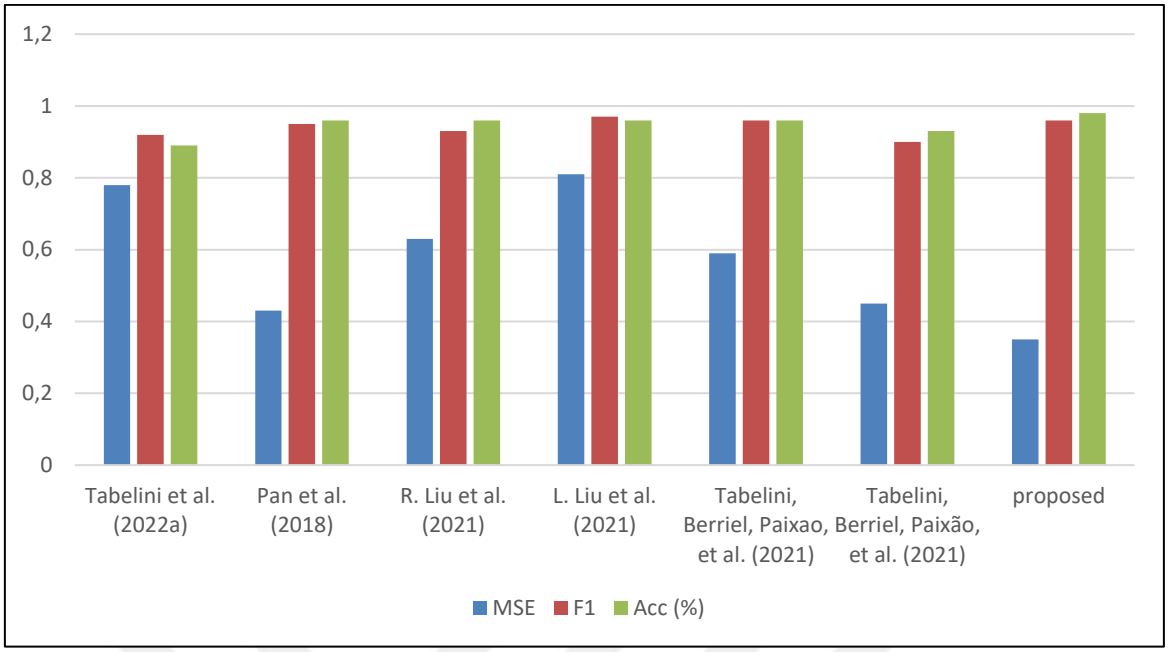

Figure 4.14: Comparing the Accuracies With State of the Art Methods.

5. CONCLUSION AND FUTURE WORK

We established a full lane identification pipeline as we structured our experiment to test the two assumptions that were stated in the Scientific Method section. This pipeline was created by merging a CNN architecture known as SegNet with a SVM model. The hypotheses that were being tested were presented in the section on the Scientific Method. Because of this, we were able to detect all lanes at the same time. After testing the first hypothesis using our pipeline with a variety of hyper-parameters, settings, and iterations, we came to the conclusion that the primary research question does not provide sufficient evidence to support the first hypothesis. Because of the results of our quantitative analysis of the model, we are unable to reach the conclusion that the SVM design, either on its own or in conjunction with a CNN, is advantageous for lane detection. This holds true whether we consider the architecture on its own or in connection with a CNN. Because, as far as we are aware, this is the only piece of study that has examined the viability of the TIT architecture as a solution to the problem of lane detection, we believe it is vital to highlight the significance of this discovery.

We only used the established mechanism in conjunction with our baseline CNN model in order to test our second hypothesis, which is that temporal information can be used as a post-processing solution. This hypothesis states that temporal information can be used. This was done in order to verify that our findings may be trusted as accurate. Because the pipeline that we had designed from the ground up expressly for this experiment was unable to fulfill the requirements of the lane detection challenge, we came to the conclusion that an other approach was necessary. Earlier works did make some attempts to use temporal data in their lane detecting algorithms, as was mentioned. These attempts did not yield the desired results. However, it becomes abundantly evident that there is a knowledge gap when it comes to implementing such a strategy as a post-processing answer. The findings that correlate to each conclusion generated from the tests show that the performance of our backbone model is much enhanced when tested using the suggested post-processing approach. This is proven by the fact that the correlation between the findings and the conclusions is clear. that there is still a need to find a balance between the two competing objectives of inference speed and detection performance. This is because these two requirements are sometimes at odds with

one another. As a consequence of the investigation that we conducted, we have arrived at the verdict that the second hypothesis is correct. In spite of the fact that there is a trade-off between the quality of the prediction and the speed of the inference, one of the most important things that we stress is the requirement of employing temporal information from a scene as a post-processing technique. This is one of the most important things that we stress since it is one of the most important things that we stress.

We believe there are a few attractive avenues to look further in terms of prospective work that could be done in the future. It is vital to study various backbone architectures prior to digging into temporal post-processing because this step comes first. These architectures ought to have been created from the bottom up specifically for lane recognition and ought to have exhibited state-of-the-art performance on benchmark lane detection datasets. It is feasible to increase the model's precision as well as its performance in the kinds of activities that need lane recognition by moving to an architecture that is more specialized for the types of tasks involved in lane recognition.

In addition, there is the potential to examine a wide variety of alternative hyper-parameter settings for our post-processing module, which is a course of action that is available for investigation. The capability of the module to make use of additional temporal information without "correcting" the forecasts for the base frame should be increased, and this should be the focus of a particular amount of attention when the process of improvement is being carried out. Experimenting with various parameters, such as the size of the temporal window, the weighting system for aggregating information across previous frames, and the fusion strategy for merging temporal and spatial cues, can help improve the consistency and smoothness of the final binary mask that is used for lane detection. Other examples of parameters that can be manipulated include the weighting system for combining temporal and spatial cues.

In spite of the fact that the findings of this research suggest that the SVM architecture is not an ideal fit for lane identification, it has been shown that making use of Transformers or other attention-based modules can result in considerable improvements in performance. This is the case even though the results of this research indicate that the SVM architecture is not an appropriate fit for lane detection. As a direct result of this, we have good cause to be

optimistic about the prospect of a wide variety of attention approaches, such as channel attention modules within the CNN network or a pipeline that takes use of typical transformer layers in order to improve feature quality.

In conclusion, it is essential to keep in mind that even if lane detection is successful with one dataset, this does not guarantee that it will be successful with any other dataset. We were unable to investigate the efficiency of our post-processing module in relation to a variety of other datasets as a result of the time constraints imposed by this investigation. Because of this, we were unable to arrive at any definitive findings regarding the efficiency of our module. As a consequence of this, we are unable to place enough emphasis on the significance of conducting additional study.

REFERENCES

[1] C. Y. Chan, ‘‘Trends in crash detection and occupant restraint technology,’’ Proc. IEEE, vol. 95, no. 2, pp. 388–396, Feb. 2007, doi: 10.1109/JPROC.2006.888391.

[2] Z. Sun, G. Bebis, and R. Miller, ‘‘On-road vehicle detection: A review,’’ IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 5, pp. 694–711, May 2006, doi: 10.1109/TPAMI.2006.104.

[3] K. A. Brookhuis, D. De Waard, and W. H. Janssen, ‘‘Behavioural impacts of advanced driver assistance systems-an overview,’’ Eur. J. Transp. Infrastruct. Res., vol. 1, no. 3, pp. 245–253, 2001,

[4] K. Grove, J. Atwood, P. Hill, G. Fitch, A. DiFonzo, M. Marchese, and M. Blanco,‘‘Commercialmotorvehicledriverperformancewithadaptive cruise control in adverse weather,’’ Proc. Manuf., vol. 3, pp. 2777–2783, Jan. 2015, doi: 10.1016/j.promfg.2015.07.717.

[5] U. Zakir, U. Z. A. Hamid, K. Pushkin, D. Gueraiche, and M. A. A. Rahman, ‘‘Current collision mitigation technologies for advanced driver assistance systems—A survey,’’ Perintis eJ., vol. 6, no. 2, pp. 78–90, 2016, doi: 10.1105/tpc.15.01050.

[6] A. H. Eichelberger and A. T. McCartt, ‘‘Toyota drivers’ experiences with dynamic radar cruise control, pre-collision system, and lanekeeping assist,’’ J. Saf. Res., vol. 56, pp. 67–73, Feb. 2016.

[7] N. S. A. Rudin, Y. M. Mustafah, Z. Z. Abidin, J. Cho, and H. F. M. Zaki, ‘‘Vision-based lane departure warning system,’’ J. Soc. Automot. Eng. Malaysia, vol. 2, no. 2, pp. 166–176, 2018.

[8] G. Kaur and D. Kumar, ‘‘Lane detection techniques: A review,’’ Int. J. Comput. Appl., vol. 112, no. 10, pp. 1–5, 2015.

[9] L. Zhang, F. Jiang, B. Kong, J. Yang, and C. Wang, ‘‘Real-time lane detectionbyusingbiologicallyinspiredattentionmechanismtolearncontextual

information,'' Cogn. Comput., vol. 13, pp. 1333–1344, Sep. 2021, doi: 10.1007/s12559-021-09935-5.

[10] R. Zhang, Y. Wu, W. Gou, and J. Chen, ''RS-lane: A robust lane detection method based on ResNeSt and self-attention distillation for challenging traffic situations,'' J. Adv. Transp., vol. 2021, pp. 1–12, Aug. 2021.

[11] F. Munir, S. Azam, M. Jeon, B.-G. Lee, and W. Pedrycz, ''LDNet: Endto-end lane marking detection approach using a dynamic vision sensor,'' 2020, arXiv:2009.08020.

[12] S. Ghanem, P. Kanungo, G. Panda, S. C. Satapathy, and R. Sharma, ''Lane detection under artificial colored light in tunnels and on highways: An IoT-based framework for smart city infrastructure,'' Complex Intell. Syst., pp. 1–12, May 2021, doi: 10.1007/s40747-021-00381-2.

[13] D. Moher, A. Liberati, J. Tetzlaff, and D. G. Altman, ''Preferred reporting items for systematic reviews and meta-analyses: The PRISMA statement,'' BMJ, vol. 339, Jul. 2009, Art. no. b2535, doi: 10.1136/bmj.b2535.

[14] C. Lee and J. H. Moon, ''Robust lane detection and tracking for realtime applications,'' IEEE Trans. Intell. Transp. Syst., vol. 19, no. 12, pp. 4043–4048, Dec. 2018, doi: 10.1109/TITS.2018.2791572.

[15] J. Li, F. Jiang, J. Yang, B. Kong, M. Gogate, K. Dashtipour, and A. Hussain, ''Lane-DeepLab: Lane semantic segmentation in automatic driving scenarios for high-definition maps,'' Neurocomputing, vol. 465, pp. 15–25, Nov. 2021, doi: 10.1016/j.neucom.2021.08.105.

[16] D. Kavitha and S. Ravikumar, ''Designing an IoT based autonomous vehicle meant for detecting speed bumps and lanes on roads,'' J. Ambient Intell. Hum. Comput., vol. 12, no. 7, pp. 7417–7426, Jul. 2021.

[17] N. M. Gohilot, A. Tigadi, and B. Chougula, ‘‘Detection of pedestrian, lane and traffic signal for vision based car navigation,’’ in Proc. 2nd Int. Conf. for Emerg. Technol. (INCET), May 2021, pp. 1–6, doi: 10.1109/INCET51464.2021.9456137.

[18] S. Swetha and P. Sivakumar, ‘‘SSLA based traffic sign and lane detection for autonomous cars,’’ in Proc. Int. Conf. Artif. Intell. Smart Syst. (ICAIS), Mar. 2021, pp. 766–771, doi: 10.1109/ICAIS50930.2021.9396046.

[19] B. Akbari, J. Thiyagalingam, R. Lee, and K. Thia, ‘‘A multilane tracking algorithm using IPDA with intensity feature,’’ Sensors, vol. 21, no. 2, p. 461, 2021.

[20] S. Lu, Z. Luo, F. Gao, M. Liu, K. Chang, and C. Piao, ‘‘A fast and robust lane detection method based on semantic segmentation and optical flow estimation,’’ Sensors, vol. 21, no. 2, pp. 400, 2021.

[21] S.-W. Baek, M.-J. Kim, U. Suddamalla, A. Wong, B.-H. Lee, and J.-H. Kim, ‘‘Real-time lane detection based on deep learning,’’ J. Electr. Eng. Technol., vol. 17, no. 1, pp. 655–664, Jan. 2022.

[22] H. Park, ‘‘Implementation of lane detection algorithm for self-driving vehicles using tensor flow,’’ in Proc. Int. Conf. Innov. Mobile Internet Services Ubiquitous Comput. (Advances in Intelligent Systems and Computing), vol. 773, 2019, pp. 438–447, doi: 10.1007/978-3-319-935546_42.

[23] R. Yousri, M. A. Elattar, and M. S. Darweesh, ‘‘A deep learning-based benchmarking framework for lane segmentation in the complex and dynamic road scenes,’’ IEEE Access, vol. 9, pp. 117565–117580, 2021, doi: 10.1109/ACCESS.2021.3106377.

[24] A. M. Alajlan and M. M. Almasri, ‘‘Automatic lane marking prediction using convolutional neural network and S-shaped binary butterfly optimization,’’ J. Supercomput., vol. 78, pp. 3715–3745, Aug. 2021.

[25] N. Kanagaraj, D. Hicks, A. Goyal, S. Tiwari, and G. Singh, ‘‘Deep learning using computer vision in self driving cars for lane and traffic sign detection,’’ Int. J. Syst.

Assurance Eng. Manage., vol. 12, no. 6, pp. 1011–1025, Dec. 2021, doi: 10.1007/s13198-021-01127-6.

[26] S. Ghanem, P. Kanungo, G. Panda, and P. Parwekar, ‘‘An improved and low-complexity neural network model for curved lane detection of autonomous driving system,’’ Wasser und Abfall, vol. 21, no. 4, p. 61, 2019, doi: 10.1007/s35152-019-0027-x.

[27] Q. Huang and J. Liu, ‘‘Practical limitations of lane detection algorithm based on Hough transform in challenging scenarios,’’ Int. J. Adv. Robot. Syst., vol. 18, no. 2, pp. 1–13, 2021, doi: 10.1177/17298814211008752.

[28] P. Lu, C. Cui, S. Xu, H. Peng, and F. Wang, ‘‘SUPER: A novel lane detectionsystem,’’IEEETrans.Intell.Vehicles,vol.6,no.3,pp. 583–593, Sep. 2021, doi: 10.1109/TIV.2021.3071593.

[29] Z. Wu, K. Qiu, T. Yuan, and H. Chen, ‘‘A method to keep autonomous vehicles steadily drive based on lane detection,’’ Int. J. Adv. Robot. Syst., vol. 18, no. 2, pp. 1–11, 2021, doi: 10.1177/17298814211002974.

[30] S. Samantaray, R. Deotale, and C. L. Chowdhary, ‘‘Lane detection using sliding window for intelligent ground vehicle challenge,’’ in Innovative Data Communication Technologies and Application. Singapore: Springer, 2021, pp. 871–881, doi: 10.1007/978-981-15-9651-3_70.

[31] M.Liu,X.Deng,Z.Lei,C.Jiang,andC.Piao,‘‘Autonomouslanekeeping system: Lane detection, tracking and control on embedded system,’’ J. Electr. Eng. Technol., vol. 16, no. 1, pp. 569–578, Jan. 2021.

[32] O. Rastogi, ‘‘Color masking method for variable luminosity in videos with application in lane detection systems,’’ in Proc. Int. Conf. Mach. Intell. Data Sci. Appl., 2021, pp. 275–284, doi: 10.1007/978-981-334087-9.

[33] D. K. Dewangan and S. P. Sahu, ‘‘Lane detection for intelligent vehicle system using image processing techniques,’’ in Data Science. Singapore: Springer, 2021, pp. 329–348, doi: 10.1007/978-981-16-1681-5_21.

[34] J. Gong, T. Chen, and Y. Zhang, ‘‘Complex lane detection based on dynamic constraint of the double threshold,’’ Multimedia Tools Appl., vol. 80, no. 18, pp. 27095–27113, Jul. 2021, doi: 10.1007/s11042-02110978-x.

[35] R. Muthalagu, A. Bolimera, and V. Kalaichelvi, ‘‘Vehicle lane markings segmentationandkeypointdeterminationusingdeepconvolutionalneural networks,’’ Multimedia Tools Appl., vol. 80, no. 7, pp. 11201–11215, Mar. 2021, doi: 10.1007/s11042-020-10248-2.

[36] K. Ren, H. Hou, S. Li, and T. Yue, ‘‘LaneDraw: Cascaded lane and its bifurcation detection with nested fusion,’’ Sci. China Technol. Sci., vol. 64, no. 6, pp. 1238–1249, Jun. 2021, doi: 10.1007/s11431-0201702-2.

[37] D. K. Dewangan, S. P. Sahu, B. Sairam, and A. Agrawal, ‘‘VLD-

[38] Net: Vision-based lane region detection network for intelligent vehicle system using semantic segmentation,’’ Computing, vol. 103, no. 12, pp. 2867–2892, Dec. 2021, doi: 10.1007/s00607-021-00974-2.

[39] L. Review and L. L. Bachrach, ‘‘Lane and obstacle detection system based on single camera-based stereo vision system,’’ in Proc. Int. Conf. Adv. Syst. Control Comput., 2021, pp. 259–266, doi: 10.1007/978-98133-4862-2.

[40] A. N. Ahmed, S. Eckelmann, A. Anwar, T. Trautmann, and P. Hellinckx, ‘‘Lane marking detection using LiDAR sensor,’’ in Proc. Int. Conf. P2P, Parallel, Grid, Cloud Internet Comput., in Lecture Notes in Networks and Systems, 2021, pp. 301–310, doi: 10.1007/978-3-030-61105-7_30.

[41] Y. Wu, F. Liu, W. Jiang, and X. Yang, ‘‘Multi spatial convolution block for lane lines semantic segmentation,’’ in Proc. Int. Conf. Intell. Comput., in Lecture Notes in Computer Science, vol. 12837, 2021, pp. 31–41.

[42] C. Y. Lee, J. G. Shon, and J. S. Park, ‘‘An edge detection–based eGAN model for connectivity in ambient intelligence environments,’’ J. Ambient Intell. Hum. Comput., vol. 13, no. 10, pp. 4591–4600, Oct. 2022, doi: 10.1007/s12652-021-03261-2.

[43] L. Zhang, B. Kong, and C. Wang, ‘‘LLNet: A lightweight lane line detection network,’’ in Proc. Int. Conf. Image Graph., 2021, pp. 355–369, doi: 10.1007/978-3-030-87355-4_30.

[44] Y. Qin, J. Peng, H. Zhang, and J. Nong, ‘‘Lane recognition system for machine vision,’’ in Proc. 10th Int. Conf. Comput. Eng. Netw., 2020, pp. 388–398, doi: 10.1007/978-981-15-8462-6_44.

[45] A. Kasmi, J. Laconte, R. Aufrere, R. Theodose, D. Denis, and R. Chapuis, ‘‘An information driven approach for ego-lane detection using lidar and OpenStreetMap,’’ in Proc. 16th Int. Conf. Control, Autom., Robot. Vis. (ICARCV), Dec. 2020, pp. 522–528, doi: 10.1109/ICARCV50220.2020.9305388.

[46] M. Fakhfakh, L. Chaari, and N. Fakhfakh, ‘‘Bayesian curved lane estimation for autonomous driving,’’ J. Ambient Intell. Hum. Comput., vol. 11, no. 10, pp. 4133–4143, Oct. 2020, doi: 10.1007/s12652-020-01688-7.

[47] E. S. Dawam and X. Feng, ‘‘Smart city lane detection for autonomous vehicle,’’ in Proc. IEEE Int. Conf Dependable, Autonomic Secure Comput., Int. Conf Pervasive Intell. Comput., Int. Conf Cloud Big Data Comput., Int. Conf Cyber Sci. Technol. Congr. (DASC/PiCom/CBDCom/CyberSciTech), Aug. 2020, pp. 334–338, doi: 10.1109/DASC-PICom-CBDCom-CyberSciTech49142.2020.00065.

[48] K. C. Bhupathi and H. Ferdowsi, ‘‘An augmented sliding window technique to improve detection of curved lanes in autonomous vehicles,’’ in Proc. IEEE Int. Conf. Electro Inf. Technol. (EIT), Jul. 2020, pp. 522–527, doi: 10.1109/EIT48999.2020.9208278.

[49] R. Muthalagu, A. Bolimera, and V. Kalaichelvi, ‘‘Lane detection technique based on perspective transformation and histogram analysis for self-driving cars,’’ Comput. Electr. Eng., vol. 85, pp. 1–16, Jul. 2020, doi: 10.1080/15551020902995363.

[50] A. Evlampev, I. Shapovalov, and S. Gafurov, ‘‘Map relative localization based on road lane matching with iterative closest point algorithm,’’ in Proc. 3rd Int. Conf. Artif. Intell. Pattern Recognit., 2020, pp. 232–236, doi: 10.1145/3430199.3430229.

[51] G. Zhang, C. Yan, and J. Wang, ‘‘Quality-guided lane detection by deeply modeling sophisticated traffic context,’’ Signal Process., Image Commun., vol. 84, May 2020, Art. no. 115811.

[52] B. Dorj, S. Hossain, and D.-J. Lee, ‘‘Highly curved lane detection algorithms based on Kalman filter,’’ Appl. Sci., vol. 10, no. 7, pp. 1–22, 2020, doi: 10.3390/app10072372.

[53] J. Hu, S. Xiong, J. Zha, and C. Fu, ‘‘Lane detection and trajectory tracking control of autonomous vehicle based on model predictive control,’’ Int. J. Automot. Technol., vol. 21, no. 2, pp. 285–295, 2020, doi: 10.1007/s12239-020-0027-6.

[54] D. Liu, Y. Wang, T. Chen, and E. T. Matson, ‘‘Accurate lane detection for self-driving cars: An approach based on color filter adjustment and K-means clustering filter,’’ Int. J. Semantic Comput., vol. 14, no. 1, pp. 153–168, Mar. 2020, doi: 10.1142/S1793351X20500038.

[55] S. Shirke and R. Udayakumar, ‘‘A novel region-based iterative seed method for the detection of multiple lanes,’’ Int. J. Image Data Fusion, vol. 11, no. 1, pp. 57–76, 2019, doi: 10.1080/19479832.2019. 1683623.

[56] Z. M. Chng, J. M. H. Lew, and J. A. Lee, ‘‘RONELD: Robust neural network output enhancement for active lane detection,’’ in Proc. 25th Int. Conf. Pattern Recognit. (ICPR), Jan. 2021, pp. 6842–6849.

[57] R. Agrawal and N. Singh, ‘‘Lane detection and collision prevention system for automated vehicles,’’ in Applied Computer Vision and Image Processing. Singapore: Springer, 2021, doi: 10.1007/978-981-154029-5_5.

[58] Z. Wang, W. Ren, and Q. Qiu, ‘‘LaneNet: Real-time lane detection networks for autonomous driving,’’ 2018, arXiv:1807.01726.

[59] F. Pizzati, M. Allodi, A. Barrera, and F. García, ‘‘Lane detection and classification using cascaded CNNs,’’ in Proc. Int. Conf. Comput. Aided Syst. Theory, in Lecture Notes in Computer Science: Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics, 2020, pp. 95–103, doi: 10.1007/978-3-030-45096-0_12.

[60] S. Chen, B. Li, Y. Guo, and J. Zhou, ‘‘Lane detection based on histogram of oriented vanishing points,’’ in Pattern Recognition. Singapore: Springer, 2020, pp. 3–11, doi: 10.1007/978-981-15-3651-9_1.

[61] N. Ma, G. Pang, X. Shi, and Y. Zhai, ‘‘An all-weather lane detection system based on simulation interaction platform,’’ IEEE Access, vol. 8, pp. 46121–46130, 2020, doi: 10.1109/ACCESS.2018.2885568.

[62] C. Hasabnis, S. Dhaygude, and S. Ruikar, ‘‘Real-time lane detection for autonomous vehicle using video processing,’’ in ICT Analysis and Applications (Lecture Notes in Networks and Systems). Singapore: Springer, 2020, pp. 217–225, doi: 10.1007/978-981-15-0630-7_21.

[63] Q. Zou, H. Jiang, Q. Dai, Y. Yue, L. Chen, and Q. Wang, ‘‘Robust lane detection from continuous driving scenes using deep neural networks,’’ IEEE Trans. Veh. Technol., vol. 69, no. 1, pp. 41–54, Jan. 2020, doi: 10.1109/TVT.2019.2949603.

[64] T.Andersson,A.Kihlberg,A.Sundström,andN.Xiong,‘‘Roadboundary detection using ant colony optimization algorithm,’’ in Advances in Natural Computation, Fuzzy Systems and Knowledge Discovery, vol. 1074. New Zealand: Springer, 2020, pp. 409–416, doi: 10.1007/978-3-03032456-8_44.

[65] S. Hossain, O. Doukhi, I. Lee, and D.-J. Lee, ‘‘Real-time lane detection and extreme learning machine based tracking control for intelligent self-driving vehicle,’’ in Intelligent Systems and Applications. Cham, Switzerland: Springer, 2020, pp. 41–50, doi: 10.1007/978-3-03029513-4_4.

[66] V.-T. Luu, V.-C. Huynh, V.-H. Tran, T.-H. Nguyen, and T.-N.-H. Phu, ‘‘Traditional method meets deep learning in an adaptive lane and obstacle detection system,’’ in Proc. 5th Int. Conf. Green Technol. Sustain. Develop. (GTSD), Nov. 2020, pp. 148–152, doi: 10.1109/GTSD50082.2020.9303108.

[67] S. Shirke and R. Udayakumar, ‘‘Fusion model based on entropy by using optimized DCNN and iterative seed for multilane detection,’’ Evol. Intell., vol. 15, no. 2, pp. 1441–1454, Jun. 2022, doi: 10.1007/s12065020-00480-y.

[68] I. J. P. B. Franco, T. T. Ribeiro, and A. G. S. Conceição, ‘‘A novel visual lane line detection system for a NMPC-based path following control scheme,’’ J. Intell. Robotic Syst., vol. 101, no. 1, pp. 1–13, Jan. 2021, doi: 10.1007/s10846-020-01278-x.

[69] P. Subhasree, P. Karthikeyan, and R. Senthilnathan, ‘‘Driveable area detection using semantic segmentation deep neural network,’’ Computational Intelligence in Data Science. India: Springer, 2020, pp. 222–230, doi: 10.1007/978-3-030-63467-4_18.

[70] R. Pihlak and A. Riid, ‘‘Simultaneous road edge and road surface markings detection using convolutional neural networks,’’ in Proc. Int. Baltic Conf. Databases Inf. Syst., in Communications in Computer and Information Science, 2020, pp. 109–121, doi: 10.1007/978-3-030-57672-1_9.

[71] D. Rato and V. Santos, ‘‘Detection of road limits using gradients of the accumulated point cloud density,’’ in Proc. 4th Iber. Robot. Conf. (Robot), 2020, pp. 267–279, doi: 10.1007/978-3-030-35990-4_22.

[72] Y. Sun, L. Wang, Y. Chen, and M. Liu, ‘‘Accurate lane detection with atrous convolution and spatial pyramid pooling for autonomous driving,’’ in Proc. IEEE Int.

Conf. Robot. Biomimetics (ROBIO), Dec. 2019, pp. 642–647, doi: 10.1109/ROBIO49542.2019.8961705.

[73] J. Liu, ‘‘Learning full-reference quality-guided discriminative gradient cues for lane detection based on neural networks,’’ J. Vis. Commun. Image Represent., vol. 65, pp. 1–7, Dec. 2019, doi: 10.1016/j.jvcir.2019.102675.

[74] H. Zhan and L. Chen, ‘‘Lane detection image processing algorithm based on FPGA for intelligent vehicle,’’ in Proc. Chin. Autom. Congr. (CAC), Nov. 2019, pp. 1190–1196, doi: 10.1109/CAC48633.2019.8996283.

[75] S. Srivastava and R. Maiti, ‘‘Multi-lane detection robust to complex illumination variations and noise sources,’’ in Proc. 1st Int. Conf. Electr., Control Instrum. Eng. (ICECIE), Nov. 2019, pp. 1–8, doi: 10.1109/ICECIE47765.2019.8974796.