FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
TÜRKİYE

REAL TIME CAR MODEL AND PLATE DETECTION SYSTEM BY USING DEEP LEARNING ARCHITECTURES

Twana Saeed MUSTAFA

Doctoral Thesis

DEPARTMENT OF SOFTWARE ENGINEERING

NOVEMBER 2024

FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
TÜRKİYE

Department of Software Engineering

Doctoral Thesis

REAL TIME CAR MODEL AND PLATE DETECTION SYSTEM BY USING DEEP LEARNING ARCHITECTURES

Author
Twana Saeed MUSTAFA

Supervisor
Prof. Dr. Murat KARABATAK

NOVEMBER 2024
ELAZIG

FIRAT UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
TÜRKİYE

Department of Software Engineering

Doctoral Thesis

Title:	Real Time Car Model and Plate Detection System by using Deep Learning Architectures
Author:	Twana Saeed MUSTAFA
Submission Date:	08 October 2024
Defense Date:	11 November 2024

THESIS APPROVAL

This thesis, which was prepared according to the thesis writing rules of the Graduate School of Natural and Applied Sciences, Fırat University, was evaluated by the committee members who have signed the following signatures and was unanimously approved after the defense exam made open to the academic audience.

		Signature
Supervisor:	Prof. Dr. Murat KARABATAK Firat University, Faculty of Technology	Approved
Chair:	Assoc. Prof. Dr. Özal YILDIRIM Firat University, Faculty of Technology	Approved
Member:	Assoc. Prof. Dr. Fatih DEMİR Firat University, Faculty of Engineering	Approved
Member:	Prof. Dr. Fatih TALU İnönü University, Faculty of Engineering	Approved
Member:	Assist. Prof. Dr. Zeynep ÖZPOLAT Muş Alparslan University, Faculty of Engineering and Architecture	Approved

This thesis was approved by the Administrative Board of the Graduate School on

..... / ..... / 20 .....

Signature

Prof. Dr. Burhan ERGEN
Director of the Graduate School

DECLARATION

I hereby declare that I wrote this Doctoral Thesis titled “Real Time Car Model and Plate Detection System by using Deep Learning Architectures” in consistent with the thesis writing guide of the Graduate School of Natural and Applied Sciences, Firat University. I also declare that all information in it is correct, that I acted according to scientific ethics in producing and presenting the findings, cited all the references I used, express all institutions or organizations or persons who supported the thesis financially. I have never used the data and information I provide here in order to get a degree in any way.

11 November 2024

Twana Saeed MUSTAFA

PREFACE

As I stand at the culmination of this academic journey, I am deeply grateful to the many individuals whose wisdom and support have shaped this thesis. It represents the combined influence and support of everyone who has contributed.

My heartfelt thanks go to Firat University in Elazig, Turkey, for granting me the opportunity to study and grow within its esteemed institution. I am proud to have been a part of this community.

Foremost, I owe immense gratitude to my supervisor, Prof. Dr. Murat KARABATAK, for his invaluable guidance in the fields of deep learning and computer vision. His mentorship has been the cornerstone of this research and has profoundly impacted my academic development.

I am equally indebted to Assoc. Prof. Dr. ÖZAL YILDIRIM, whose critical insights and unwavering encouragement during challenging phases of my work were transformative. His support has truly been a beacon of hope throughout this journey.

To my family—my parents, sister, and brothers—your unconditional love and belief in me have been my unwavering foundation. You have been my source of strength, turning every challenge into an opportunity for growth.

To my beloved wife, your emotional and linguistic support have been invaluable. Your patience, understanding, and solid faith in me have been the pillars of my perseverance. This achievement is as much yours as it is mine.

I extend a special thanks to Engineer Marwa Farhad, my sister-in-law, for her linguistic expertise, which greatly enhanced the quality of my work.

To my friends and colleagues, your companionship and encouragement have made this journey all the more rewarding.

Lastly, to everyone who offered support, whether large or small, your contributions have been instrumental. While I may not have mentioned every name, my gratitude extends to each one of you.

The main contributions of this study include the development of an integrated system that combines vehicle make and model detection with Automatic Number Plate Recognition (ANPR) or Automatic License Plate Recognition (ALPR) technology at the entrance gate of Firat University in Turkey is motivated by the goal of enhancing security, improving access control, and optimizing campus operations. This innovative approach utilizes advanced technology to create a safer and more efficient environment on campus, serving multiple practical purposes and supporting the university's commitment to maintaining a secure and well-regulated atmosphere, achieving a significant improvement in accuracy and efficiency with a remarkable 97.5% accuracy rate. The system demonstrates exceptional robustness in various environmental conditions such as fog, rain, and low light. It was rigorously tested at Firat University's entrance with a thousand images, showcasing its practical applicability. Additionally, the use of GradCam technology provides valuable insights into the decision-making processes of neural networks, identifying areas for improvement, particularly in cases of misclassification. These contributions pave the way for advanced applications in autonomous driving, traffic management, stolen vehicle tracking, and security surveillance, setting a new standard for future research in the field.

For over a century, vehicle make model, and license plates have been utilized to uniquely identify vehicles, facilitating rapid tracking and identification for various purposes. This research developed and designed for real-time applications by leveraging Advanced deep learning algorithms(Utilizing MobileNet-V2 and YOLOx for vehicle identification, and YOLOv4-tiny, Paddle OCR, and SVTR-tiny for ANPR, the system leverages state-of-the-art technologies to achieve high accuracy and efficiency), particularly Convolutional Neural Network (CNN) architectures, the system is expected to deliver superior performance

compared to existing methods, offering enhanced tracking and monitoring, optimized campus traffic management, and overall security improvements

Finally, I am pleased to announce that our work has been approved and published by IEEE Access, IEEE Access one of the most important journals and institutions of technology today, indexed by (Science Citation Index Expanded (SCI) (Clarivate Analytics), ScopusQ1, Web of Science … etc.). IEEE Access was included in the 2023 Journal Citation Report (JCR) and received: 3.4-impact factor, 0.6 immediacy index, and 9.8 CiteScore.

Twana Saeed MUSTAFA
ELAZIG, 2024

TABLE OF CONTENTS

Page

ABSTRACT

Real Time Car Model and Plate Detection System by using Deep Learning Architectures

Twana Saeed MUSTAFA

Doctoral Thesis

FIRAT UNIVERSITY
Graduate School of Natural and Applied Sciences

Department of Software Engineering

November 2024, Page: xvi + 118

The advent of deep learning has revolutionized computer vision, enabling real-time analysis crucial for traffic management and vehicle identification. This dissertation introduces a novel system that integrates vehicle make and model detection with Automatic Number Plate Recognition (ANPR), achieving a remarkable 97.5% accuracy rate. By leveraging deep learning techniques, including MobileNet-V2, YOLOx, YOLOv4-tiny, Paddle OCR, and SVTR-tiny, the system effectively processes real-time video feeds, even under challenging conditions such as fog, rain, and low light. A key innovation of this dissertation lies in the integration of these diverse deep learning models into a unified framework. This integration enables the system to accurately identify vehicle makes and models, as well as recognize license plate numbers, with high precision. To further enhance the model's performance and gain deeper insights into its decision-making process, Gradient-weighted Class Activation Mapping (Grad-CAM) is employed. This technique highlights the most influential regions of the image, enabling the refinement of the model's predictions and identification of potential areas for improvement. The proposed system has significant implications for various applications, including autonomous driving, traffic management, law enforcement, and security surveillance. By advancing the state-of-the-art in vehicle identification, this dissertation paves the way for future innovations in computer vision. Specifically, this system can contribute to improved traffic flow, enhanced road safety, and more efficient law enforcement operations.

Keywords: Car model, Plate detection, Deep learning, computer vision, OpenCV, MobileNet-V2, YOLOv4, GradCam, Firat University,

ÖZET

Derin Öğrenme Mimarilerini Kullanarak Gerçek Zamanlı Araba Modeli ve Plaka Algılama Sistemi

Twana Saeed MUSTAFA

Doktora Tezi

FIRAT ÜNİVERSİTESİ
Fen Bilimleri Enstitüsü

Yazılım Mühendisliği Anabilim Dalı

Kasım 2024, Sayfa: xvi + 118

Derin öğrenmenin ortaya çıkışı, trafik yönetimi ve araç tanımlama için kritik öneme sahip gerçek zamanlı analize olanak sağlayarak bilgisayarlı görüyü devrimleştirmiştir. Bu araştırma, araç marka ve model tespitini Otomatik Plaka Tanıma (OPT) ile birleştiren ve çığır açan %97,5 doğruluk oranına ulaşan bir sistem sunmaktadır. Geleneksel yöntemlerin aksine, bu çalışma her iki yönü tek ve uyumlu bir sistemde bütünleştirerek, araç tanımlama için daha bütüncül ve verimli bir çözüm sunmakta ve olumsuz hava koşullarında bile sağlam bir performans sağlamaktadır.

Makale, Python programlama dili ile birlikte OpenCV dahil olmak üzere derin öğrenme tekniklerinin kullanımını araştırmaktadır. Araç tanımlama için MobileNet-V2 ve YOLOx (You Only Look Once), OPT için ise YOLOv4-tiny, Paddle OCR (optik karakter tanıma) ve SVTR-tiny kullanılarak geliştirilen sistem, Fırat Üniversitesi'nin girişinde sis, yağmur ve düşük ışık gibi çeşitli koşullar altında çekilen bin görüntü ile titizlikle test edilmiştir. Sistemin bu testlerdeki olağanüstü başarı oranı, sağlamlığını ve pratik uygulanabilirliğini vurgulamaktadır.

Ek olarak, deneyler sistemin doğruluğunu ve etkinliğini değerlendirmekte, sinir ağlarının karar verme süreçlerini anlamak ve özellikle yanlış sınıflandırmalarda iyileştirme alanlarını belirlemek için Gradyan Ağırlıklı Sınıf Aktivasyon Haritalama (GradCam) teknolojisini kullanmaktadır.

Bu araştırmanın bilgisayarlı görü için etkileri önemlidir ve otonom sürüş, trafik yönetimi, çalıntı araçlar ve güvenlik gözetimi gibi alanlarda ileri uygulamalara zemin hazırlamaktadır. Gerçek zamanlı, yüksek doğruluklu araç tanımlaması sağlayan entegre Araç Marka ve Model Tanıma (AMMT) ve OPT sistemi, alandaki gelecek araştırmalar için yeni bir standart belirlemektedir.

Anahtar Kelimeler: Araç modeli, plaka tespiti, derin öğrenme, bilgisayarlı görü, OpenCV, MobileNet-V2, YOLOv4, GradCam, Fırat Üniversitesi

LIST OF FIGURES

Page

LIST OF TABLES

Page

ABBREVIATIONS

DL	: Deep Learning
ML	: Machine Learning
MMR	: Make And Model Recognition
ANPR	: Automatic Number Plate Recognition
ALPR	: Automatic License Plate Recognition
VMMR	: Vehicle Make and Model Recognition
CV	: Computer Vision
IoT	: Internet of Things
ITS	: Intelligent Transportation Systems
CCTV	: Closed-circuit television (Video Surveillance)
Grad-CAM	: Gradient-Weighted Class Activation Mapping
AVI	: Automatic Vehicle Identification
CPR	: Car Plate Recognition
OCR	: Optical Character Recognition
COCO	: Common Objects in Context Dataset
YOLO	: You Only Look Once
ASCII	: American Standard Code for Information Interchange
CNNs	: Convolutional Neural Networks.
GANs	: Generative Adversarial Networks.
MLP	: Multi-layer Perceptron.
ReLU	: Rectified Linear Unit.
SPP	: Spatial Pyramid Pooling.
RoI	: Region of Interest.
RNNs	: Recurrent Neural Networks.
HoG	: Histogram of Oriented Gradient.
SVM	: Support Vector Machine
FC	: Fully Connected.
SGD	: Stochastic Gradient Descent.
AI	: Artificial Intelligence.
WSN	: Wireless Sensor Network
SPANS	: Smart Parking Service (SPANS)
TDNN	: Two Time-Delay Neural Networks
HOG	: Histogram of Oriented Gradients
CPU	: Central Processing Unit
GPU	: Graphical Processing Unit
TNPR	: Truck Number Plate Recognition
KNN	: K-Nearest Neighbors
LPL	: License Plate Localization
HSRP	: High Security Registration Plates
RNNs	: Recurrent Neural Networks
DQN	: Deep Q-Networks
DDPG	: Deep Deterministic Policy Gradient
ANN	: Artificial Neural Network
FNN	: Feedforward Neural Network

NLP	: Natural Language Processing
DNNs	: Deep Neural Networks
DPM	: Deformable Parts Model
RCNN	: Region-Based Convolutional Neural Network
MAP	: Average Precision
RPN	: Region Proposal Network
SSD	: Single-Stage Detectors
FPS	: Frames per Second
NAS	: Neural Architecture Search
HR	: High-Resolution
SL	: Supervised Learning
UNSL	: Unsupervised Learning

1. INTRODUCTION

Nowadays, as technology advances exponentially across all areas of life, it continues to enhance the quality of services provided [1]. This progress is especially significant in the field of vehicle transportation, where technology plays a crucial role. The automotive transportation system is essential for traffic monitoring [2], crime detection, stolen vehicle tracking, and various security applications.

The quantity of active vehicles has drastically increased, leading to more illegal activities. The rapid proliferation of vehicles makes it difficult to track them, highlighting the importance for relevant authorities to do so. As the flow of vehicles grows daily, there is a heightened need for automatic number plate detection and vehicle make and model identification. This tool is also valuable for monitoring vehicle speeds, especially in light of the serious accidents that occur each year. The UK police first recognized the Automatic Number Plate Recognition (ANPR) system in 1976, and its prevalence has significantly increased in the last decade [1].

Thousands of surveillance cameras are installed along highways, primarily to support traffic management and law enforcement. Continuous manual inspection of this footage is impractical, as it would require immense effort and substantial costs. Automated visual interpretation provides the capability to detect, track, and classify all traffic in real-time. A critical concept in this context is make and model recognition (MMR). The make and model information of a vehicle can be used to identify vehicles with stolen license plates by comparing the observed vehicle model with the registered details associated with the plate. Additionally, this capability is valuable in criminal investigations where a vehicle description—without a license plate number—is available. In such cases, visually obtaining the make and model of the vehicle is essential [3]. These challenges are central to this study.

Vehicle Make and Model Recognition (VMMR), also known as vehicle identification, tackles the challenge of determining a vehicle's make and model—identifying the manufacturer and specific model—based on an input image or video featuring the vehicle. Recently, VMMR has become an integral element of Intelligent Transportation Systems (ITS) and has garnered considerable interest from the research community [4,5].

Make and model recognition serves as a valuable tool in various applications, including traffic management, intelligent surveillance, traffic behavior analysis, and traffic monitoring [6]. For instance, VMMR technology is used in traffic cameras installed at toll stations to automatically detect and recognize passing vehicles. In traffic monitoring, vehicle model recognition assists Intelligent Transportation Systems (ITS) in generating vehicle flow statistics, which are essential for analyzing real-time traffic conditions. Additionally, in the realm of traffic security [2], make and model recognition aids in locating stolen vehicles [7].

The primary objective of this research is to enhance security and facilitate the recovery of stolen vehicles, as well as to address cases of mismatched or missing license plates through a highly accurate visual analysis system. To achieve this, we present a Vehicle Make and Model Recognition (MMR) system developed specifically for the Turkey National Police. This system utilizes cameras mounted on overhead highway sign structures to capture vehicle images, enabling precise extraction of make and model details. The extracted data can be integrated with existing Automatic Number Plate Recognition (ANPR) information [8]. This system is optimized to monitor a single lane (see Figure 1.1). The existing camera infrastructure supports the training of our recognition model, which has been trained on an extensive dataset of vehicle images labeled with corresponding make and model information. Given the bandwidth limitations between the online cameras and the offline training and testing facilities, optimizing the collection of training and testing samples is necessary. Additionally, a significant challenge involves the automatic processing of new or rare vehicle models registered in the vehicle database, for which acquiring sufficient training and testing images is difficult. To address these issues, we propose utilizing vehicle and ANPR datasets, such as the COCO and Stanford datasets. The sampling and annotation processes in this system are automated, though periodic manual intervention is required for training updates. This approach allows for the creation of an initial dataset and facilitates the incremental accumulation of new vehicle samples over time, ensuring the system's optimal performance at all times [9,10].

The VMMR system is specifically designed with detection and classification stages to localize vehicles within bounding boxes, focusing on a full-frontal view. This work aims to facilitate vehicle make and model identification without relying on an ANPR system. Our two-stage approach enables the detection of vehicles in each video frame and performs classification once a vehicle is identified. This study builds on our previous work by offering a comprehensive analysis of MMR classification performance and providing an in-depth evaluation of the MMR system. First, we compare various convolutional neural networks for vehicle model classification. Second, we provide detailed insights into classification performance by identifying the most informative regions for VMMR classification and assessing robustness against occlusions. Finally, we further investigate instances of misclassification within the VMMR system to identify potential weaknesses in the system and its data handling capabilities [10,11].

Deep Learning—now recognized as a powerful approach to managing large datasets—leverages complex algorithms and artificial neural networks to train machines in a manner that allows them to learn from experience, and to classify and recognize data or images similarly to the human brain. Within Deep Learning, Convolutional Neural Networks (CNNs) are widely used artificial neural networks specifically suited for image and object recognition and classification. In this context, CNNs enable the recognition of objects within an image. They are extensively applied across various domains, including image processing tasks, localization and segmentation for

computer vision, obstacle recognition in autonomous vehicles, and speech recognition in natural language processing. Due to their critical role in these rapidly evolving fields, CNNs are a fundamental component of Deep Learning [12].

Computer vision is an interdisciplinary scientific field dealing with the way machines understand digital photos from images or videos. Therefore, the computer vision tasks contain methods of acquiring and processing digital images, analyzing and understanding the different formats they have, such as single pictures and video sequences, or extraction of data from high-dimensional information out of the real world in order to produce numerical or symbolic descriptions that are useful for decision-making. Automatic number plate recognition (ANPR) has been a technology that uses optical character recognition on images to read vehicle registration plates. Modern advanced computer vision technology together with the falling prices of devices related to it in recent years makes it fairly possible to visually recognize vehicles automatically, irrespective of whether online or offline through video/CCTV. This research therefore aims to design a computer vision-based automatic vehicle identification system using optical character recognition (OCR) techniques in attaining vehicle identification as shown in Figure 1.1. The work then involves investigations on real-time automatic number-plate recognition and the possible extension of the above to other aspects of road traffic monitoring and control [13,14].

Grad-CAM was a key factor of both diagnosing and sense making the models' misclassifications. In that sense, when Grad-CAM highlighted the region of the image in which focal point the model gave predictions, it became possible to diagnose the patterns in misclassifications. For example, sometimes reflections or occlusions were the reason for which the model failed to classify a vehicle or its license plate correctly. Grad-CAM visualizations helped in tempering the model's behavior to be much robust towards such peculiarities [15].

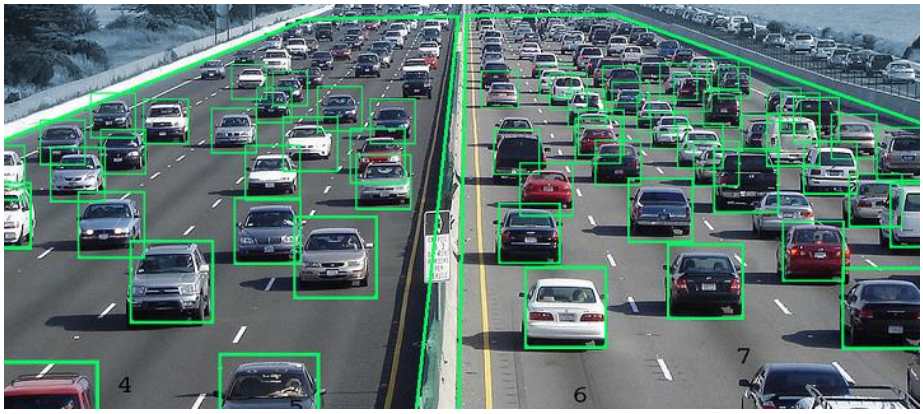

Figure 1.1. OCR technology used for object recognition [2]

Automatic License Plate Recognition (ALPR) is a technology that identifies the license plate numbers of vehicles from images captured by cameras, which may be in color, black and white, or infrared formats. This technology employs a mix of methodologies such as object detection, image processing, and pattern recognition. ALPR is referred to by various other terms including automatic vehicle identification (AVI), car plate recognition (CPR), automatic number plate recognition (ANPR), and optical character recognition (OCR) for vehicles [16,17].

Developing Automatic Number Plate Recognition (ANPR) systems presents a complex challenge, as it necessitates a sequence of interconnected steps, with the success of each step hinging on the one before it. Attaining absolute precision is unattainable at this point in time. Furthermore, an array of technical difficulties emerges due to diverse elements such as variations in lighting or background designs, the quality of images, and the processing speed of the machine learning algorithms. These elements significantly influence the effectiveness of number plate detection. In addition, the position, quantity, dimensions, typeface, color, and angle of the number plates introduce additional complications in the development of a reliable ANPR system. [18,19].

1.1. Motivation

The integration of Automatic Number Plate Recognition (ANPR) or Automatic License Plate Recognition (ALPR) technology with vehicle make and model recognition at the entrance gate of Firat University in Turkey is motivated by the goal of enhancing security, improving access control, and optimizing campus operations. This innovative approach utilizes advanced technology to create a safer and more efficient environment on campus, serving multiple practical purposes and supporting the university's commitment to maintaining a secure and well-regulated atmosphere.

For over a century, vehicle make, model, and license plates have been utilized to uniquely identify vehicles, facilitating rapid tracking and identification for various purposes. This research aims to develop an Automatic Number Plate Recognition (ANPR) system using OpenCV, coupled with car brand classification, to detect stolen and counterfeit vehicles, enhance security, and improve access control precision. The proposed system is designed for real-time applications, such as monitoring the entrance gate at Firat University, as well as other locations. By leveraging deep learning algorithms, particularly Convolutional Neural Network (CNN) architectures, the system is expected to deliver superior performance compared to existing methods, offering enhanced tracking and monitoring, optimized campus traffic management, and overall security improvements.

1.2. Significance and Applications of Vehicle Make Model and AN/LPR System

Both Vehicle Make and Model Recognition and Automatic Number/License Plate Recognition (AN/LPR) systems play a crucial role in enhancing security, optimizing traffic management, and streamlining various processes across multiple industries. Their diverse applications extend from supporting law enforcement efforts to providing convenience in everyday life.

Vehicle make and model recognition is essential for numerous applications, including identification and tracking, safety, and regulatory compliance. It is widely used in sectors including law enforcement, the insurance industry, the automotive industry, and traffic management. Similarly, Automatic Number/License Plate Recognition (AN/LPR) systems are significant for security and surveillance, law enforcement, and parking management. Their applications extend to security and surveillance, toll collection, and access control, making them essential tools across multiple domains [20,21].

1.3. Problem and Stamen of MMR and ANPR

This investigation delves into the significant obstacles faced by vehicle make and model license plate recognition systems. It particularly concentrates on the variability of license plates, the categorization of car brands, and the influence of environmental elements like lighting, angle, and diverse typography. Two primary types of adverse meteorological conditions hinder the effective deployment of ANPR systems: Static conditions, which include phenomena such as mist, haze, and fog, marked by the presence of water droplets in the air, and Dynamic conditions, which are characterized by high-speed events such as rain and snowflakes [46,47]. The identification of license plates under these conditions poses a considerable challenge in Figure 1.2 show how different weather conditions and lighting environments can affect image processing and recognition tasks in the context of vehicles. The study highlights the unique difficulties posed by static weather conditions, where the air is filled with water particles, for LPR systems. Moreover, dynamic weather conditions introduce further complications with fast-moving elements like rain streaks and snow, making the accurate detection of license plates more challenging [22,23]. The research underlines the importance of overcoming these obstacles to achieve efficient license plate recognition.

Practical Implications: This investigation centers on the crucial practical elements involved in image and video analysis for the purpose of license plate recognition. The efficacy of these methodologies is dependent on a variety of elements, including the specifications of the camera used and the prevailing conditions of illumination. Furthermore, the effectiveness of numerous

tasks within the domain of CV is significantly influenced by the accessibility of comprehensive training datasets.

Issues of Traffic Regulation and Monitoring: The increasing vehicular population in Turkey, compounded by its dense demographic composition, has led to a surge in traffic infractions. It is observed that a wide array of vehicles, including those of specialized nature, frequently violate traffic regulations, encroach upon restricted zones, and are implicated in incidents of theft. In light of these developments, the urgent need for an efficient license plate recognition system, coupled with a sophisticated car brand identification mechanism, becomes evident for the purpose of pinpointing vehicles implicated in traffic violations. This study sheds light on the complex challenges encountered by license plate recognition frameworks and vehicle identification, emphasizing the crucial role these systems play in mitigating traffic-related offenses. As illustrated in Figure **1.2**, the impact of dynamic and static weather conditions on vehicle imaging introduces significant challenges, particularly in plate variation and brand classification under diverse illumination settings. This figure highlights the complexities caused by different weather conditions, which can affect the visibility and clarity of vehicle plates and brands. Addressing these variations requires advanced imaging techniques to ensure accurate and reliable vehicle identification and classification. Through a thorough examination of license plate variability, automobile brand identification, and the impact of environmental factors, this research offers invaluable insights aimed at augmenting the precision and effectiveness of ALPR technologies. The ultimate objective of the proposed study is to catalyze technological advancements, thereby fostering a safer and more regulated traffic environment in Turkey.

Figure 1.2. Show some ANPR's problem [24]

1.4. Proposed Methodology

This section provides a comprehensive discussion of the methods and tools used for vehicle detection and tracking. Initially, the tools employed in the research are described in detail, highlighting their specific applications within the study. Following this, the implementation process is outlined, beginning with data cleaning and preprocessing, and proceeding to the application of algorithms. A summary of these topics is provided at the conclusion of the chapter.

This study presents a real-time system for the identification of car models and license plates utilizing cutting-edge deep learning methodologies. As is customary in such research, this work employed various tools and applications for model construction and experimental validation.
Figure 1.3 illustrates the flowchart of our system, delineating each step of its operation. This also addresses potential scenarios where the system may fail to detect ANPR and vehicle make and model recognition, possibly due to environmental factors such as complete darkness at night or the positioning of the vehicles.

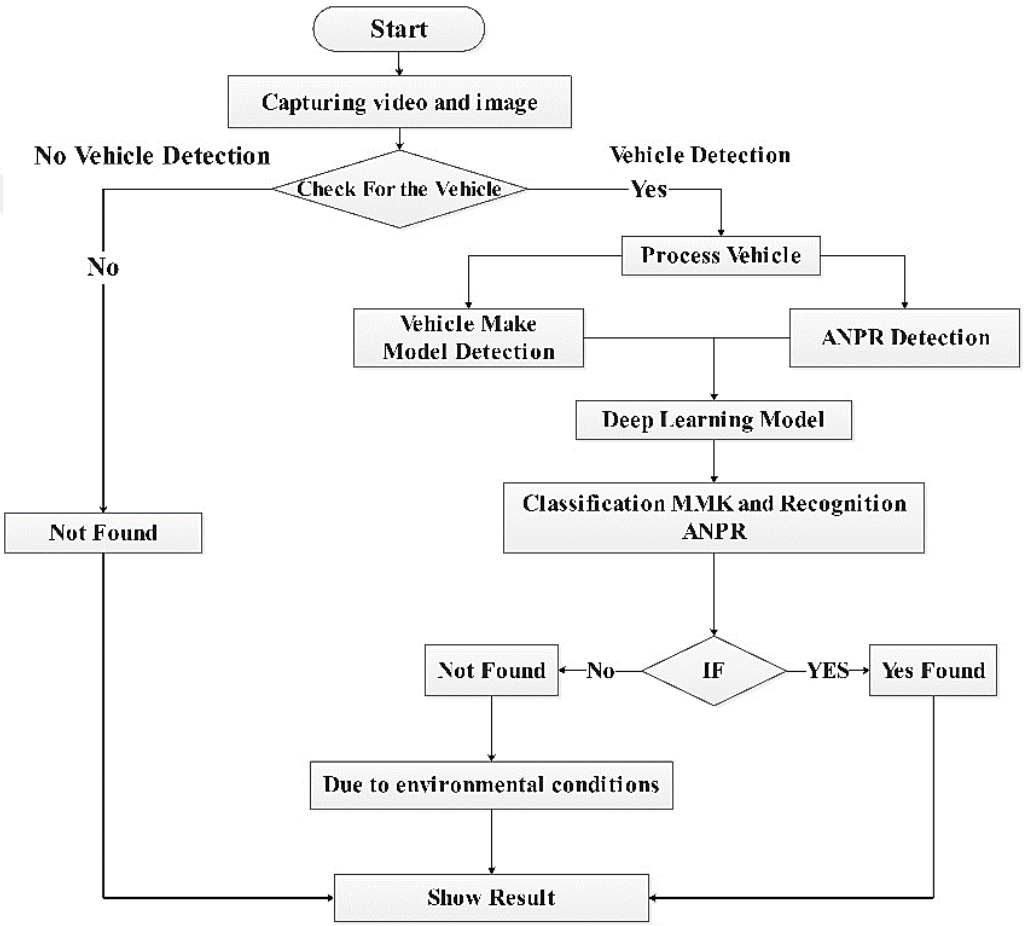

Figure 1.3. Show the system flow chart

In terms of Vehicle Detection and Classification, the system initiates by capturing images or videos of vehicles, which necessitates the use of specifically configured cameras. Following the successful identification of a vehicle in the visual data, the system advances to deduce the vehicle's brand and model. This process is facilitated by the deployment of the MobileNet-V2 Convolutional Neural Network (CNN) architecture, esteemed for its efficacy and precision in tasks related to image categorization. The system is capable of distinguishing and categorizing up to 30 distinct types of vehicles, including well-known manufacturers such as BMW, Audi, and Tesla. The

MobileNet-V2 CNN framework, which utilizes the COCO dataset for annotation and is pre-trained on the Stanford Car Dataset, is instrumental in the recognition of various vehicle attributes. For validation purposes, the system was tested at the vehicle entrance gate of Firat University, demonstrating its practical applicability.

License Plate Detection and Recognition Following the vehicle classification, the system adopts the YOLOv4-tiny framework for the detection of license plates, chosen for its balance of speed and simplicity, thus ensuring swift execution times while preserving satisfactory precision levels. The system then progresses to decipher the characters on the license plates through Automatic License Plate Recognition (ALPR), unfolding in four principal stages: (1) Image Preprocessing, (2) License Plate Localization, (3) Character Segmentation, and (4) Recognition of the License

Plate's Actual Numbers. In this context, the ALNR (Automatic License Number Recognition) system employs Paddle OCR, synergized with YOLOv4, to achieve meticulous character segmentation. Paddle OCR is adept at predicting character sequences, including letters and digits, by scanning pixel clusters within the image to identify characters and convert them to their corresponding ASCII representations. The SVTR-tiny model, trained on a diverse array of license plates, further refines the recognition accuracy. The system's efficacy was validated using images from the vehicle entrances at Firat University, where it demonstrated an impressive OCR accuracy rate of 97.5%.

The operational workflow of the system is succinctly outlined in the accompanying flowchart. The process commences with the capture of an image or video, followed by the detection of vehicles within the frame. Should a vehicle not be detected, the system outputs a 'Not Found' message. Conversely, if a vehicle is identified, the system proceeds to analyze the make and model, and then moves on to the automatic detection of the number plate through ANPR. This is succeeded by the deployment of deep learning algorithms for the classification and recognition tasks, taking into account environmental variables that might affect the model's adaptability.

1.5. Thesis Structure

This Thesis is divided into five different chapters. Chapter 1 explains the introduction, and aim to this project in different area, motivation to the project. Chapter 2 background reviews the different existing Computer Vision techniques used in ANPR and The Vehicle Make and Model Recognition (VMMR) provides a deep explanation. Chapter 3 describes in great detail Deep Learning and the methodology carried out for developing the ANPR application. Chapter 4 analyses about the discussion and experimental results obtained with the developed system. Finally, Chapter 5 talked about conclusions provides some ideas for improving or increasing the functionalities of the developed system that could be implemented in the future.

2. RELATED WORK

This chapter presents a review of the literature related to this study. Over the past thirty years, significant efforts have been dedicated to addressing the challenges associated with (MMR) and Automatic License Plate Recognition (ALPR). As a result, a substantial body of research has emerged in these areas. This section provides an overview of the primary methodologies applied in MMR and the three key stages of ALPR: license plate detection, character segmentation, and character recognition. Recent advancements in deep learning have notably improved the effectiveness of real-time vehicle detection systems, particularly in the areas of car model recognition and license plate identification. Numerous studies have investigated various architectures and approaches aimed at enhancing detection accuracy and processing speed. For example, some research emphasizes frontal and rear vehicle detection techniques, detailing specific network architectures and training protocols to optimize overall performance.

Techniques such as MobileNet-V2, YOLO, and Paddle OCR are widely recognized for their high accuracy and efficiency in Vehicle Make and Model Recognition (VMMR) and Automatic Number Plate Recognition (ANPR). These systems excel at processing images in real-time, making them highly suitable for traffic management and security surveillance applications. Studies have demonstrated that the integration of VMMR and ANPR can achieve impressive accuracy rates, even in challenging conditions such as fog, rain, and low-light environments. This combined approach offers a comprehensive solution for vehicle identification, which is crucial in modern urban settings. Additionally, multi-stage systems and embedded solutions further enhance performance by incorporating multiple detectors and classifiers. Researchers have developed systems that utilize the redundancy of information in video streams to significantly improve recognition accuracy. Embedded systems, optimized for real-world limitations such as processing power and memory, ensure real-time performance without significant delays. These advancements are not only essential for traffic management and security but however play a pivotal role in the development of autonomous driving technologies. By continuously improving the accuracy and efficiency of these systems, deep learning architectures are laying the groundwork for smarter and safer transportation solutions. In addition, methods for locating license plates in images have been developed, providing a broad indication of the presence of vehicles. It is critical to explore the latest literature on specific algorithms and frameworks used in these systems, as approaches and effectiveness can vary widely.

2.1. Car Make Model Detection

Real-time vehicle makes and model detection systems, powered by deep learning architectures, have significantly advanced the field of computer vision, particularly in critical applications such as traffic management, security surveillance, and autonomous driving. These systems employ sophisticated neural networks to achieve high levels of accuracy and efficiency in identifying vehicle makes and models in real-time. Among the most widely adopted architectures for this task is YOLO (You Only Look Once), which is renowned for its exceptional speed and accuracy in object detection tasks. The ability of YOLO to process images in real-time is particularly advantageous in scenarios where rapid decision-making is essential, such as in dynamic traffic conditions or security monitoring. In conjunction with other state-of-the-art models like MobileNet-V2 and Paddle OCR, these systems ensure robust performance across various environments, enabling accurate vehicle detection even in complex and high-traffic situations. The continuous improvement of these deep learning architectures plays a pivotal role in pushing the boundaries of real-time vehicle detection, ensuring that the systems remain effective and reliable in practical applications [25].

Another widely utilized architecture in vehicle detection is MobileNet-V2, which is specifically optimized for mobile and embedded vision applications. MobileNet-V2 achieves an optimal balance between accuracy and computational efficiency, making it particularly suitable for use on devices with constrained processing capabilities, such as smartphones and embedded machines. A major advantage of MobileNet-V2 is its adaptability through transfer learning, which allows for fine-tuning to enhance specific tasks and applications, which allows the model to adapt to specific datasets and improve its performance in recognizing various car models. This adaptability is especially valuable in scenarios where environmental factors, such as lighting and weather conditions, may vary. By leveraging transfer learning, MobileNet-V2 can be refined to maintain high accuracy even in challenging environments, making it a robust solution for Vehicle Make and Model Recognition (VMMR) in real-world applications. As part of the broader field of deep learning architectures, MobileNet-V2's efficiency and adaptability continue to enhance the development of intelligent transportation systems and autonomous driving technologies [26].

A fundamental aspect of these systems is the use of high-quality, well-labeled datasets for both training and evaluation. One of the most prominent large-scale image datasets in computer vision is the COCO (Common Objects in Context) dataset. This dataset comprises over 330,000 images annotated with 80 object categories, including vehicles, making it an invaluable resource for training Vehicle Make and Model Recognition (VMMR) systems. The detailed annotations provided by COCO, such as bounding boxes, segmentation masks, and key points, are crucial for training deep learning architectures like MobileNet-V2, YOLO, and Paddle OCR. These detailed labels enable models to learn how to recognize and classify objects with high accuracy. The rich

variety of object types and contexts present in the COCO dataset allows models to generalize better to real-world scenarios, ensuring robust performance under varying conditions. Furthermore, using a well-annotated dataset like COCO not only improves the accuracy of detection and recognition tasks but also enhances the ability of these models to perform efficiently in real-time applications, such as traffic management and autonomous driving. The availability of such comprehensive datasets plays a critical role in advancing the development and refinement of deep learning architectures used in VMMR, helping to push the boundaries of accuracy and efficiency [27].

The integration of these deep learning models into real-time systems involves multiple stages, including image preprocessing, feature extraction, the use of the COCO dataset, and classification. Advanced techniques, such as Gradient-weighted Class Activation Mapping (Grad-CAM), are frequently employed to visualize and interpret the decision-making processes of neural networks, providing valuable insights into potential misclassifications. This capability is particularly useful for refining model accuracy and enhancing the interpretability of deep learning architectures like MobileNet-V2, YOLO, and Paddle OCR. The ongoing improvement and optimization of these architectures are crucial for developing more accurate and reliable Vehicle Make and Model Recognition (VMMR) systems, even in challenging environments such as poor lighting or adverse weather conditions. These advancements are vital not only for the evolution of intelligent transportation systems and the creation of smart cities but also for bolstering security surveillance efforts. As these systems become more refined, they play an increasingly significant role in ensuring safety and efficiency in modern urban infrastructure [28].

2.2. License Plate Detection

The license plate detection phase is pivotal in influencing the accuracy of subsequent stages in Vehicle Make and Model Recognition (VMMR) systems. The input is typically an image that may contain no license plates, one, or multiple plates, and the output is expected to isolate only the image region containing the license plate(s). In recent years, various algorithms have been developed to address this complex issue. Although detection accuracy has improved considerably, identifying license plates in images taken from diverse viewpoints remains a challenge. Factors such as occlusion and inconsistent lighting continue to impact the detection process. A brute-force method that analyzes every pixel in the image is computationally prohibitive due to high processing times. Consequently, a more efficient approach focuses on detecting prominent features of the license plate and processing only the relevant pixels, thereby significantly reducing processing time. Existing algorithms are generally categorized based on their methodologies. The five main approaches, which will be explored here, include edge-based, color-based, texture-based, character-based, and combinations of multiple features. A summary of the strengths and limitations of each method is presented below [29].

2.2.1. Edge-Based Approaches

Edge-based approaches detect license plates by leveraging the fact that license plates typically have a rectangular shape with a known aspect ratio. By utilizing the color contrast between the license plate and its background, the boundaries of the plate can be identified. A Sobel filter is commonly employed to detect potential edges, focusing on regions with a high density of edges. This method can be broken down into two categories: advantages and disadvantages [30].

Advantages:

1- Edge-based: Simple and fast in execution.

2- Color-based: Unaffected by inclined or deformed plates.

3- Texture-based: Does not rely on license plates having regular shapes.

4- Character-based: Achieves a high recall rate and is robust to rotation.

5- Combining two or more features: Highly robust and effective.

Disadvantages:

1- Edge-based: Returns too many candidates in images with numerous edges and struggles with blurry images.

2- Color-based: Produces limited results when used in isolation.

3- Texture-based: Computationally complex, particularly when the input image has many edges or varying illumination.

4- Character-based: Faces challenges when the image contains additional text.

5- Combining two or more features: Significantly increases computational complexity.

Regions identified as potential license plates undergo additional processing with both horizontal and vertical edge detection to pinpoint candidate areas. In contrast, Zheng et al. [31] utilize only vertical edge detection, as horizontal edge detection tends to increase error rates due to confusion between the horizontal edges of the license plate and those of the car bumper. Furthermore, only rectangular regions matching the aspect ratio of license plates are selected, which helps filter out irrelevant regions and improve detection accuracy. The Hough transform is advantageous in this context due to its capacity to identify straight lines with inclinations of up to 30 degrees, although this method is resource-intensive in terms of time and memory. Overall, edge-based techniques provide a straightforward and fast approach for license plate detection. Despite their efficiency, these methods encounter difficulties with complex input images, often producing excessive candidate regions in edge-rich images or struggling to detect license plates in blurry images [32].

2.2.2. Color-Based Approaches

Color-based approaches rely on the distinct color contrast between license plates and their background to detect the plates. Typically, the color combination of the plate and its characters is unique and rarely found outside the plate region. In [33], a genetic algorithm is applied to learn how to recognize these specific color combinations. Through training under varying illumination conditions, the algorithm adapts to identify the relevant colors. By analyzing the average brightness of each region, it can distinguish license plate regions from other areas [34]. Additionally, regions are segmented based on color differences. Once a candidate region is identified, edge-based methods are used to locate the license plate within that region. However, the execution time for this process is not specified. While color-based methods can be used independently, they are often combined with other approaches to enhance performance. One of their key advantages is that they are unaffected by inclined or deformed plates. Nevertheless, when used alone, color-based methods may yield limited results as they primarily depend on illumination conditions.

2.2.3. Texture-Based Approaches

Texture-based approaches identify license plates by leveraging the distinctive pixel density patterns within the plate area. The contrast between the plate's characters and the surrounding background facilitates the application of various texture-based detection methods. For instance, in [35], a technique using sliding concentric windows was introduced, treating license plates as texture anomalies within the image, with abrupt texture changes marking potential plate regions. To improve accuracy, [36] presented a method that combines sliding concentric windows with histogram analysis. Additionally, [37] introduced a technique that employs both global statistical features and local Haar-like features. The global features help eliminate most background noise, while the local Haar-like features enhance robustness against variations in brightness, color, size, and position of license plates. Texture-based methods offer the advantage of not depending on the plate’s regular shape and can operate even when the plate boundaries are deformed, making them more adaptable than color- and edge-based techniques. However, the primary limitation of texture-based approaches lies in their computational complexity, as processing time can increase substantially with images containing many edges or varying lighting conditions.

2.2.4. Character-Based Approaches

Character-based approaches identify license plates by analyzing images for character-like patterns, treating license plates as strings of characters. Regions where such patterns are detected are then proposed as potential license plate areas. For instance, in [38], a region-based approach utilizes a neural network to classify image segments based on their similarity to characters, and a

license plate is detected when a linear arrangement of character-like regions is identified. Building on this, [39] employs a 37-way output convolutional neural network (CNN) to detect all characters in the image, followed by a second CNN to filter out false positives, although this process takes approximately 5 seconds per image. Despite its slower speed, this approach eliminates the need for a separate character segmentation step. Additionally, in [40], a saliency-based technique is used, where characters are first detected and segmented using an intensity saliency map, achieving high recall. A sliding window is then applied to the segmented characters, with license plates identified based on saliency features. Overall, character-based methods generally provide high recall rates and exhibit robustness to rotated plates. However, these approaches may yield higher error rates when other text appears in the image, and they are often computationally intensive.

Advantages:

1- Projection-based: Simple and robust to rotation.

2- Pixel connectivity based: Simple and robust to rotation.

Disadvantages:

1- Projection-based Requires prior knowledge, is sensitive to font changes, and noise affects the results.

2- Pixel connectivity based: Broken or joined characters pose significant challenges.

2.2.5. Approaches Combining Two or More Features

When possible, integrating multiple features has proven to be an effective strategy for license plate detection. For instance, [41] utilizes a covariance matrix containing both statistical and spatial information related to the license plate. A neural network is then trained on this covariance matrix, alongside non-license plate regions, to enhance detection accuracy. In [42], two time-delay neural networks (TDNNs) are applied: one analyzes the license plate's color while the other examines its texture. The outputs of both networks are combined to pinpoint candidate regions, achieving a high detection rate on a dataset of 1,000 video sequences from toll gates. Additionally, convolutional neural networks (CNNs) used for general object detection also fall within this multi-feature category. Overall, combining two or more features generally yields more reliable and robust detection than using a single feature alone. However, these multi-feature approaches tend to be more computationally intensive than single-feature methods.

2.3. Literature Review

Over the past ten years, as information technology has advanced and the population has grown, various methods for creating license plate detection systems have been developed and explored. Machine learning, deep learning, artificial intelligence, and Internet of Things (IoT)

technologies are increasingly being employed to address the challenges people face in daily life, particularly within the transportation sector. Issues such as traffic congestion, parking difficulties, accidents, and safety concerns require smart solutions to mitigate their impact. In recent years, automated and intelligent electronic systems have been developed to tackle these problems. By utilizing advanced deep learning architectures such as MobileNet-V2, YOLO, and Paddle OCR, these technologies offer greater accuracy and efficiency in Vehicle Make and Model Recognition (VMMR) and other transportation applications. The integration of AI and IoT into transportation systems enables real-time data processing and decision-making, helping to alleviate these issues and improve overall safety and efficiency.

The author emphasizes that Automatic-Number-Plate Recognition (ANPR) systems are crucial in image processing, but they face challenges due to variations in license plate formats, lighting, scales, and colors across different regions. To improve detection and recognition rates, researchers have developed and tested various techniques, such as the Douglas Peucker Algorithm for shape approximation, which detects rectangular contours and extracts the most prominent one as the number plate. Connected component analysis is then used to segment characters, and optical character recognition (OCR) recognizes the number plate characters [43].

This paper provides a detailed survey of current ANPR systems and algorithms, comparing their performance in real-time testing and simulations. The goal is to advance ANPR technology built on computer vision (CV) algorithms by analyzing extraction, segmentation, and recognition techniques and providing guidelines for future trends. Additional hardware may be required to maximize accuracy even with the best algorithms [44].

The main objective of this research is to investigate the segmentation and recognition issues in the License Plate Recognition Framework and develop alternative solutions. This involves three phases: locating and extracting the license plate area from a larger scene image, removing the alphanumeric characters from the background using the license plate region, and then providing them for recognition in an OCR system. To successfully identify a vehicle through the license plate, the plate must be visible in the image captured by the acquisition system, such as a video or still camera. [45].

The use of an automated license detection system in today's busy traffic system is crucial. It automatically monitors traffic regulations and helps enforce them. In India, there are many incidents of rash driving and violation of traffic rules by drivers, making it difficult for traffic police officers to identify car details. To simplify and expedite traffic regulation monitoring on cars, an automated license detection system has been developed and implemented over time. This article provides a brief overview of various approaches for automatic license detection. [46].

The system for recognizing vehicles encompasses two primary phases: detection and classification, aimed at identifying and categorizing vehicles from a directly frontal perspective. In

the initial detection phase, various methodologies are employed. Ren and Lan [47] utilize frame differencing, while Prokaj and Medioni [48] employ background subtraction techniques to ascertain the complete dimensions of the vehicle. Extensions to these detection methods have been made by Siddiqui et al. [49] and Petrović and Cootes [50], who enhanced the system's capability through the integration of license-plate detection algorithms. Moreover, Wijnhoven and de with [51] introduced the use of a Histogram of Oriented Gradients (HOG) [52] for achieving detection that is robust to variations in lighting. Furthermore, Zhou et al. [53] have contributed to this field with their research on leveraging Convolutional Neural Networks (CNN) for enhancing the precision of vehicle detection. Upon successful detection of a vehicle, the identified vehicular section within the image is subsequently processed for classification, specifically within the context of Make and Model Recognition (MMR) [54].

Deep learning has become a fundamental approach in object detection, with Convolutional Neural Networks (CNNs) being among the most widely used architectures in this area. According to Jerry Wei, CNNs are highly effective models that are relatively easy to manage and train, demonstrating a low tendency to overfit when applied to large-scale image datasets. However, training CNNs on high-resolution images demands substantial computational resources [55].

Due to extensive research and practical applications, Convolutional Neural Networks (CNNs) have proven highly effective for vehicle detection and recognition tasks. For instance, Xingcheng Luo et al. [56] utilized a large image dataset and increased the depth of AlexNet to achieve recognition accuracies of 96.31% for vehicles and 91.22% for facial recognition. Similarly, Hyo Jong Lee et al. [57] extracted frontal views of vehicle images and employed SqueezeNet for training and testing. Despite relying on a desktop CPU with a high-performance GPU, the study achieved a recognition accuracy of 96.3%, with inference tasks averaging 108 ms. Notably, their model required under 5 MB of storage, enhancing its suitability for real-time inference applications.

The field of image classification has also been extensively documented. CNNs, or Convolutional Neural Networks, are currently the most advanced method for classifying images. They were first developed by LeCun et al. [58] and got widespread recognition when Krizhevsky et al. [59] utilized a CNN called AlexNet to obtain the highest performance in the 1000-class ImageNet Challenge. The user's text is [60]. MMR, a modified version of AlexNet, was developed by Ren and Lan [61] to reach a 96.7% accuracy. This was accomplished by using 233 vehicle models in a dataset of 42,624 pictures. Yang et al. [62] released the CompCar dataset, which comprises several perspectives of cars, including both internal and external components, as well as 45,000 frontal photos of 281 distinct car types. The study demonstrated that AlexNet [59] achieves similar performance to the more contemporary Overfeat[63] and GoogLeNet[64] CNN models (97.0% compared to 97.2% and 97.3%, respectively). Siddiqui et al. [49] demonstrated that the Bag

of SURF features method achieves a 94.8% accuracy on a vehicle dataset consisting of 29 classes and 6639 images, specifically for small-scale classification challenges.

Since its introduction by Krause et al. (2013), the Stanford Cars Dataset has been used as a standard for assessing MMR algorithms. It has a huge library of vehicle photos with thorough annotations, which makes deep learning model building and comparison easier [65].

Writer [66] Classification and Recognition of Car Images. This paper shows outcomes from experiments for the machine learning algorithm, Python modules, and deep neural network model built using Keras and TensorFlow for artificial vision classification of car pictures. Applications for image classification range from medical diagnostics to autonomous vehicles.

Writer [67] Convolutional Neural Network-Based Vehicle Classification Approach. For the purpose of identifying and classifying vehicles, they employ convolution. CNN is a neural network type that is utilized in deep learning. The driverless vehicle Developing a smart city with classification for traffic surveillance video systems is a challenging task for the Intelligent Transportation System (ITS). Three distinct categories of vehicles—motorbikes, cars, and trucks—are represented in this article by 3,000 motorcycles, 6,000 cars, and 2,000 pictures of trucks. CNN can automatically take in and extract the varied aspects of different vehicle datasets without requiring a manual selection of criteria. CNN's accuracy is evaluated using the detected object's confidence values. The highest confidence rating in the vehicle classification for the bike category is roughly 0.97.

Writer [68] CNN-Based Real-Time Vehicle Classification. This paper demonstrated the use of convolutional neural networks for the classification of four types of common vehicles in our country. They used two methods to classify the vehicle: feature extraction and classification. These are two tasks that convolutional neural networks can perform with ease.

The author highlights the significance of Automatic-Number-Plate Recognition (ANPR) systems in image processing. However, these systems encounter difficulties arising from the diverse license plate formats, lighting conditions, scales, and colors found in different locations. In order to enhance the accuracy of detecting and identifying objects, scientists have devised and evaluated different methodologies, including the Douglas Peucker Algorithm for form approximation. This algorithm specifically identifies rectangular outlines and selects the most prominent one as the number plate. Subsequently, connected component analysis is employed to partition characters, while optical character recognition (OCR) identifies the characters on the number plate. The number [69].

This research offers a thorough analysis of the state-of-the-art ANPR systems and algorithms, contrasting how well they function in simulations and real-time testing. Through analysis of extraction, segmentation, and recognition methods and the provision of suggestions for future trends, the goal is to enhance ANPR technology, which is based on computer vision (CV)

algorithms. Even with the finest algorithms, maximizing accuracy could require additional technology [70].

The primary aim of this research is to examine the problems related to segmentation and recognition in the License Plate Recognition Framework and devise alternate solutions. The process consists of three stages: identifying and isolating the license plate section from a wider image, eliminating the alphanumeric letters from the surrounding back-ground using the license plate region, and subsequently feeding them into an OCR system for identification. In order to accurately determine the identity of a vehicle using its license plate, it is necessary for the plate to be clearly visible in the image obtained by the acquisition system, such as a video or still camera. The user's text is [71].

In today's congested traffic system, the automated license detecting system is essential. It assists in enforcing traffic laws by automatically monitoring them. India has a high rate of driver infractions and reckless driving, which makes it challenging for traffic police officials to determine specific car specifications. Through development and implementation throughout time, an automatic license detecting system has been created to streamline and accelerate traffic regulation surveillance on automobiles. An overview of the several methods for automatic license detection is given in this article [72].

The real-time Automatic Number Plate Recognition (ANPR) system has become an essential tool for access control and traffic management, with applications extending from traffic regulation to vehicle parking and data management in smart surveillance. Earlier ANPR systems employed a range of localization techniques, such as edge detection and contour analysis, to accurately pinpoint the vehicle’s license plate area. These foundational image processing methods enable ANPR systems to perform effectively under challenging conditions, including variations in lighting, plate orientation, and resolution, as detailed in [73]. Previous studies have implemented various localization algorithms, incorporating preprocessing techniques like edge detection and contour analysis for reliable number plate detection.

Furthermore, two distinct variations of the ANPR system are discussed in [74], online and offline. The primary difference between these two types lies in their processing approach. In an online ANPR system, incoming video frames are processed in real-time, allowing for continuous tracking. In contrast, offline ANPR systems capture images for subsequent processing. The OpenCV library, which offers numerous predefined methods, is particularly suited for real-time implementation, making it a valuable tool in both variations.

In 2018, Fakhar et al. proposed an affordable ANPR system using Raspberry Pi [75], where the model processes real-time images captured from a camera. The image undergoes denoising, filtering, and segmentation, after which the characters on the license plate are recognized. The Raspberry Pi manages all computational tasks, though there is a noticeable 3-second delay before

the result is displayed. The resulting label is also stored in a database. Additionally, an ANPR system for smart check-in and check-out was introduced in [76], aimed at reducing waiting times and monitoring vehicle entry and exit. Images are captured from CCTV cameras, enabling the ANPR system to instantly detect the license plate number through image processing, while vehicle registration and security details are saved in a database via a web application. This approach is designed to decrease check-in times and alleviate parking and traffic congestion.

Virakwan and Nui Din in [77] proposed an ANPR system for POLIMAS, designed to ensure that only registered vehicles are granted access to specific areas. A webcam is installed with four different orientations: front, back, front top, and rear top. The captured images are first converted to grayscale, and the intensity and contrast are enhanced using the histogram equalization technique. The bounding box method is then employed to identify and crop the character regions, utilizing a combination of Sobel edge and Laplacian edge detectors. This approach achieved a 95.83% accuracy rate in plate localization. For Optical Character Recognition (OCR), eigenvector and correlation methods are applied to compare each character, resulting in a character recognition accuracy of 87.41%. The recognized characters are assembled into a string and matched against the database's reserved entries for verification.

Recognizing number plates remains a challenge due to the diverse formats used in different regions. In response, Chou and Liu [78] introduced a real-time Truck Number Plate Recognition (TNPR) system aimed at reducing labor and the time required for identifying number plates. Their system not only enhances automation and efficiency but also improves transparency and reduces the risks associated with crime. By employing YOLO and CNN-based deep learning architectures, the system achieved a single character identification rate of 97.39%, an overall recognition rate of 93.73%, and an inference time of 0.3271 seconds per image.

Chen and Hu introduced an Intelligent Transportation System (ITS) in [79], focusing on video-based vehicle identification and classification techniques. Their approach leverages both static and motion features to enhance accuracy. The proposed method successfully localizes the vehicle number plate area and classifies the characters on the plate with 95% confidence, even under varying environmental conditions such as different levels of illumination.

A vehicle plate recognition system based on Optical Character Recognition (OCR) and Wireless Sensor Network (WSN) is described in [80]. The proposed system utilizes the Smart Parking Service (SPANS) framework to capture images of parking spaces and recognize the number plates of moving or parked vehicles. Additionally, the system's performance is evaluated using real-time images. In [81], R. N. Babu et al. implemented state-of-the-art deep learning (DL) techniques for number plate recognition. Their dataset consisted of 6,500 Indian car number plates, divided into 90:10 ratios for training and testing. The images were captured using three different cameras with varying specifications, such as bit rate and focal length. A 37-class Convolutional

Neural Network (CNN) model was trained for character recognition, utilizing 126 filters. YOLOv3 was employed to detect the vehicle number plates, and the system achieved a 91% accuracy rate for number plate character recognition.

In [82], character recognition was achieved with 97% accuracy using a K-Nearest Neighbors (KNN) based technique. The images, captured with a webcam featuring a resolution of 640 × 480 pixels, underwent several preprocessing steps, including gray-scaling, inversion, thresholding, edge detection, and the application of morphological techniques.

[83] This article explains how to use artificial neural networks to create an automatic number plate recognition system for crowded areas and tollways. The system uses OpenCV, a programming library for real-time computer vision, to improve traffic flow and reduce congestion at tollways.

In [84] utilized template matching algorithm with normalized cross correlation and phase correlation method to test 90 patterns under different conditions. The paper compared the accuracy of normalized cross correlation method and phase correlation method in identifying license plates. The normalized cross correlation method achieved an accuracy of 67.98%.

in [85]. This article presents an algorithm for detecting number plates and identifying letters, which is based on a combined feature extraction model and BPNN. The algorithm achieves an efficiency of 97.1% and enhances time performance.

The article [86] discusses an automated parking fee calculation system that is capable of detecting 41 out of 50 THAI number plates with an accuracy rate of 93.42%. The system helps reduce manual identification of license plates. However, the accuracy of the system can reach 100% only if the number plates are bright, luminous, and free from noise.

The researchers developed a method to recognize license plates in images using deep neural networks. They processed the license plates and used LSTM Tesseract OCR Engine for recognition, achieving a 97% precision with 95% LPR accuracy. They plan to increase the size of the image dataset in the future [87].

In [88], the authors describe an automatic gate control system based on license plate recognition to enhance security and convenience at important locations. The system uses a PIC microcontroller and a regular PC with a video camera to detect visible license plates and process them. The system operates automatically and can recognize license plates to allow or deny entry. The proposed system was implemented using MATLAB, Proteus, and Micro C, and achieved a recognition accuracy of about 96%.

The author of [89] suggested an optimal approach for the ALPR system for Bangladeshi vehicles using deep neural network models. They trained and evaluated these models on their datasets of Bangladeshi vehicles and license plates. They also introduced an algorithm that eliminates the need for the segmentation phase and generates output efficiently. The proposed

system offers 97.37% accuracy in license plate localization and 96.31% accuracy in text recognition.

In the Table **2.1** below show Comparison between our system and other study were conducted

Table 2.1. Show comparison research with proposed method

Ref	Methodology	Dataset	Accuracy
[47]	CNNs - five convolutional layers	ILSVRC 2012	91.6%
[49]	Bag of SURF Features, CNNs, Multi-class SVM.	NTOU-MMR Dataset	94.8%
[50]	NN Classification with Feature Extraction	Parking Lot Dataset	93%
[51]	Unsupervised Sub-categorization with Iterative Split-and-Merge	USC Multi-View Car Dataset	93%
[53]	DAVE Framework (Fast Vehicle Proposal Network (FVPN) and Attributes Learning Network (ALN))	CompCars Dataset and UTS Vehicle Dataset	97.1%
[54]	AlexNet	Custom vehicle dataset	97.3%
[56]	Nine-Layer CNN	Vehicle Dataset (90,000 images)	91.22%
[57]	Residual SqueezeNet Architecture	Custom Korean Vehicle Dataset	96.3%
[59]	AlexNet	ImageNet (ILSVRC 2010 and 2012 subsets).	97.0%
[61]	CNN (5 Convolutional Layers and 2 Fully Connected Layers with Batch Normalization).	CompCars Dataset.	96.7%
[62]	Overfeat CNN	CompCar dataset	92.7%
[63]	ConvNet	ImageNet	97.2%
[64]	GoogLeNet	ILSVRC 2014	97.3%
[66]	CNN- SSD (Single Shot MultiBox Detector)	Stanford Car Dataset	88%
[67]	CNN based on extracted features	Custom vehicle dataset	97%
[68]	CNN - 10-layer architecture	Standard Dataset	97%
[69]	Douglas Peucker Algorithm	Custom vehicle dataset	95.5%
[77]	Sobel edge and Laplacian edge	Custom vehicle dataset (Malaysian)	95.8%
[78]	YOLOv3, C-CNN-L3, and R-CNN-L3	Heping River Dredging Operation Dataset	95%
[81]	37-class (CNN), YOLOv3	Indian car number plates	91%
[82]	K-Nearest Neighbors (KNN)	Custom vehicle dataset (Indonesian vehicle license plates)	97%
[84]	Normalized Cross Correlation and Phase Correlation using Template Matching	Custom vehicle dataset (Nepali vehicle number plates)	67.98%
[85]	Feature Extraction Model and Backpropagation Neural Network (BPNN)	Custom vehicle dataset (Chinese license plates)	97.1
[86]	CNN- 2D Correlation for Character Recognition	Custom vehicle dataset (Thai license plate images)	93.42%
[87]	Faster R-CNN	Kaggle's Vehicle dataset	95%
[88]	PIC microcontroller	Custom vehicle dataset (Inda)	96%
[89]	YOLOv4, Resnet50	Bangladeshi vehicles LP	96.31%

2.4. Style of Turkish License Plate

The vehicle number plate in Turkey is rectangular and made of aluminum with a white background and black or yellow font letters. On the left side of the plate, there is a blue bar with the country code "TR" in the size of 4x10 cm. Two plates are mandatory for all vehicles except for motorcycles and tractors, with one placed at the front and the other at the rear. The text format on the plates follows a pattern of either "00 X 0000", "00 X 00000", "00 XX 000", "00 XX 0000" or "00 XXX 00". The first two digits indicate the province code number of the car owner's residence [90].

2.5. Automatic Number Plate Recognition

Automatic Number Plate Recognition (ANPR) is a system that uses optical recognition to identify license plate numbers after detecting them in an image. The accuracy of the system depends largely on the quality of the acquired images. Images are two-dimensional visual perceptions that are critical to the algorithm's functionality. The car can be captured in this frame, and the plate number can be extracted provided the image is not blurry, obscured or defective. A car can be identified uniquely by its number plate. The system processes the image and returns the vehicle's data, which includes the plate number. Figure 2.1. show lock diagram ANPR framework [91].

ANPR is a system that utilizes Optical Character Recognition (OCR) to recognize characters from various sources like surveillance cameras or other cameras. It is crucial to position the camera at the correct angle to capture the best and most precise image. OCR analyzes each character separately. OCR is a method of transmitting written or printed information from any source, including written or printed documents and images, to a text decryption machine to obtain the desired source. It is a complex process that involves multiple algorithmic steps, such as uploading the image, detecting characters, adjusting them on the page, eliminating blurriness, and producing a final editable format [91].

The OCR system is utilized in various domains, including business, industries, research, security, literature, and even medicine, to develop devices for individuals with vision impairments. This technology is widely employed in applications such as LP recognition, passport scanning in airports, institutional barcode reading, and digitizing handwritten text into electronic documents.

The ANPR system has different algorithms or rules for processing each part of the license plate image. However, there are several challenges in processing the image due to factors such as blurriness, insufficient lighting, obstructive objects, poor angles, font differences, and inconsistencies between states [92].

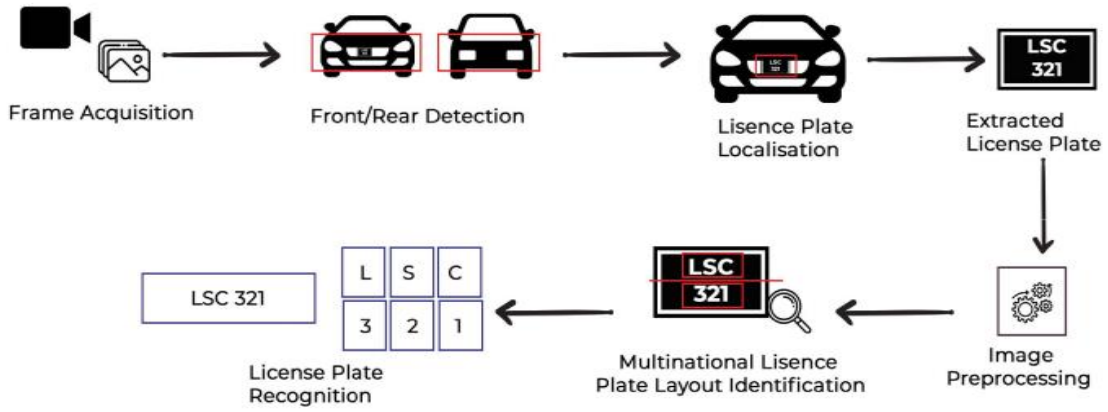

Figure 2.1. Block diagram of real-time ANPR framework [92].

ANPR is an optical system that identifies car license plates from images. It is a crucial tool used for traffic management, security, parking, and speed control. ANPR is implemented in various areas such as parks, highways, by police, customs, and private companies [93]. It is possible to extract data about a driver with just one image. The quality of the acquired images is crucial for the ANPR system to function correctly. The system uses Optical Character Recognition (OCR) to recognize characters on number plates from images acquired by surveillance cameras. The number plate uniquely identifies a car and can only be extracted if the image is clear and not defective. For accurate license plate recognition, it is important to position the camera at the right angle to capture a clear image. The characters on the license plate are analyzed individually using Optical Character Recognition (OCR). OCR is a technology used to convert written or printed text into digital format and is utilized in various fields such as business, industries, research, security, literature, and even medicine, to develop devices for individuals with visual impairments [94]. The ANPR system is commonly used for various applications, including recognizing plate numbers, passports at airports, barcodes in institutions, and handwritten text in electronic documents. The system has algorithms or rules for processing images and license plates, but there are challenges due to blur, insufficient lighting, objects obstructing the license plate's visibility, poor angles, different fonts, and inconsistencies between states [94].

Traffic rule violations are a major cause of accidents, and ALPR plays a significant role in restricting them, as well as being necessary for intelligent transportation systems. The ALPR system consists of four main parts, below is a Figure 2.2. that displays the minimum components required for an ANPR (Automatic Number Plate Recognition) system: image acquisition, license plate localization (LPL),

character segmentation, and character recognition. The accuracy of each stage affects the overall accuracy of the system. The first part involves inputting high-resolution digital images of the car from the front and rear, followed by image processing techniques such as Gaussian and

Prewitt filters to extract the license plate in the second part. In the third part, the characters and numbers in the extracted license plate are segmented, and finally, both ANNs and SVM are used for character recognition [95].

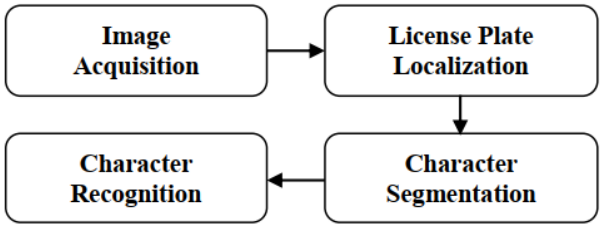

Figure 2.2. Flow diagram of ALPR system [95].

The accuracy and efficiency of an Automatic License Plate Recognition (ALPR) system rely heavily on its capability to manage varying conditions at each operational stage. However, numerous challenges within the ALPR domain can impact both its precision and processing speed. These challenges arise from both external and internal factors related to the system [95]. External factors can be grouped into three primary categories:

- Plate Variation
- environmental variation
- camera Mounting variation

Challenges stemming from the ALPR system's internal factors can be divided into two principal categories:

- Technique/Algorithm Used in ALPR system
- Hardware used in ALPR system

These factors will be discussed in detail in the following sections.

2.5.1. External Factors

A. Pate Variation

Plate size: The size of High Security Registration Plates (HSRP) or Turkish License Plates (LP) is standardized, but some people may use unstandardized LPs of varying sizes and dimensions, which can be seen in Figure 2.3 Although unstandardized LPs are not allowed legally in Turkey, they do exist, presenting a challenge for the ALPR system to be able to handle this variation in plate size, Figure 2.3 displays license plates of different sizes [44].

Figure 2.3. LPs with different plate sizes [44]

Plate position: In Turkey, there are regulations for the placement of license plates on vehicles in terms of their horizontal and vertical positions and height from the ground. However, in practice, license plates may be mounted anywhere on the vehicle, as illustrated in Figure 2.4. This presents a challenge for the ALPR system to accurately locate these plates [44].

Figure 2.4. LP with different plate positions [44]

Plate color [9]: In Turkey, the HSRP has standardized colors for the background and text on the LP. However, in Turkey, LPs come in various color combinations. Private vehicles have black text on a white background, commercial vehicles have black text on a yellow background, rental vehicles have yellow text on vehicles have white text on a light blue background, and electric vehicles have white text on a green background. This variety of colors poses a challenge for the ALPR system to accurately recognize LPs, Figure 2.5 displays license number plates of various colors [97].

Figure 2.5. LPs with different plate colors [97]

Number of LPs in an Image: When you take a picture with a camera, you might see one or more license plates (LPs) in the picture, as shown in Figure 2.6. This can happen because of the camera's field of view or other reasons. It is also difficult for the automatic license plate recognition (ALPR) algorithm to detect and recognize all LPs and their registration numbers in the image [44].

Figure 2.6. Number of plates in an image [44]

Plate Character Font: HSRP has standardized the fonts used in license plates (LP), including character width, height, and stroke width. However, there are also license plates that are not standardized, as seen in Figure 2.7. This poses a challenge for the ALPR system to locate the LP and recognize characters with varying font parameters [98].

Figure 2.7. LP with different fonts [98]

Plate Occlusion: there is a lot of dust on internal or local roads, which can accumulate and stick to license plates (LP). This can cause the characters on the LP to become partially or fully invisible, as depicted in Figure 2.8. This makes it difficult for the ALPR system to recognize characters from such images [99].

Figure 2.8. LP with different plate occlusions [99]

Noise: LP and vehicle bodies frequently contain extraneous or unwanted elements, such as screws, frames, pipes, text-like structures, stickers, and stamps, as shown in Figure 2.9. These elements present a challenge for automatic license plate recognition (ALPR) systems, which must differentiate these components from the actual LP to avoid misidentification [88].

Figure 2.9. LP with different noises [88]

Physical Obstructions: License plates (LP) can have unwanted obstructions, such as frames, pipes, and ribbons, as demonstrated in Figure 2.10. These obstructions can partially or completely hide the LP, which presents a challenge for the automatic license plate recognition (ALPR) system to extract the characters from such LPs [100].

Figure 2.10. LP with different physical obstruction [100]

Damaged Pate: License plates (LP) can become partially damaged for various reasons, as illustrated in Figure 2.11. Additionally, the stroke of a character may be broken, making it difficult for the automatic license plate recognition (ALPR) system to detect such LP and recognize its characters, Damaged License Plates (LP) [100].

Figure 2.11. Damaged LP [100]

Skew: The skew of a license plate (LP) can be caused by incorrect mounting of the plate and band or by the curvature of the road. This presents a challenge for the automatic license plate recognition (ALPR) system to detect the LP from such images, Figure 2.12. displays the skew of a license plate (LP) [98].

Figure 2.12. Skew LP [98]

2.5.2. Environmental Variation

Lighting Conditions: Images taken by a camera can vary in lighting conditions depending on the time of day, including sunlight, evening, night, shadow, glare, and cloudy conditions, as illustrated in Figure 2.13. Furthermore, license plate images can be impacted by both one's own vehicle's headlights and those of other vehicles, which can pose a challenge for automatic license plate recognition (ALPR) systems [101].

Figure 2.13. LP with different lighting conditions [101]

Surrounding effects: Other patterns and textures, such as characters written on the background, that resemble the license plate can impact license plate detection, as shown in Figure 2.14[101].

Figure 2.14. LP with different surrounding effects [101]

2.5.3. Camera Mounting Variation

The performance of the ALPR system can also be affected by the camera mounting technique. The camera can be either fixed or moving, with the former being mounted on a stationary surface and the latter being mounted on a moving surface, such as a patrolling vehicle. Relative velocity between the camera vehicle and the target vehicle can be a challenge for a moving ALPR camera. In most ALPR systems, fixed camera mounting is used, which cannot be mounted with zero horizontal and vertical angles [102].

Angle [88]: Three main parameters related to angle of Mounting of camera affects the accuracy of ALPR system as shown in Figure 2.15.

Pan: The angle between LP and camera pan

Tilt: Elevation angle of the LP image

Projection Orientation: The angle at which LP is rotated left or right.

Figure 2.15. Angle associated with camera mounted [102]

The ALPR system may face issues depending on the angle between the camera and the plate axes. A high angle can make it difficult to detect and extract characters from the license plate as shown in Figure 2.16 [102].

Figure 2.16. LP with different camera angles [102]

Plate distance from camera: The performance of the ALPR system is affected by the distance between the license plate and the camera, as depicted in Figure 2.17. This poses a challenge for the ALPR system to be robust for varying distances between the camera and the license plate [102].

Figure 2.17. LP with different distance from camera [102]

Area ratio: To improve the ALPR system's accuracy, the camera must be positioned to focus mainly on the LP region, as illustrated in Figure 2.18, while minimizing the non-LP region captured [103].

Figure 2.18. LP with different area ratio [103]

2.5.4. Internal Factors

The ALPR system consists of both hardware and software components, and it is essential to choose the best combination of these two for optimal performance. The following are the primary internal factors that can impact the system's performance [104].

A. Technique/ Algorithm

Various algorithms present limitations, including specific processing time requirements that can pose challenges in meeting the demands of real-time applications [105].

B. Hardware

Camera shuttering speed (Motion Blur): The speed of the vehicle and the camera's shutter speed [12] affect the amount of motion blur in the Figure 2.19. A larger shutter speed can reduce the open time and decrease motion blur, which presents a challenge for reducing motion blur in ALPR system [106].

Figure 2.19. LP with different camera shuttering speed [106]

Camera Resolution (Quality of Image): The resolution of the camera is a challenging factor for the ALPR system. Although high resolution can provide good image quality, it can also increase the overall processing time of the system. Figure 2.20 illustrates this issue by showing images with varying resolutions [106].

Figure 2.20. LP with different camera resolution [106]

Camera focus length (Maximum distance): The maximum distance between the camera and LP is determined by the camera's focus length, as demonstrated in Figure 2.21. Automatically adjusting the focus length for the ALPR system is also a challenge [107].

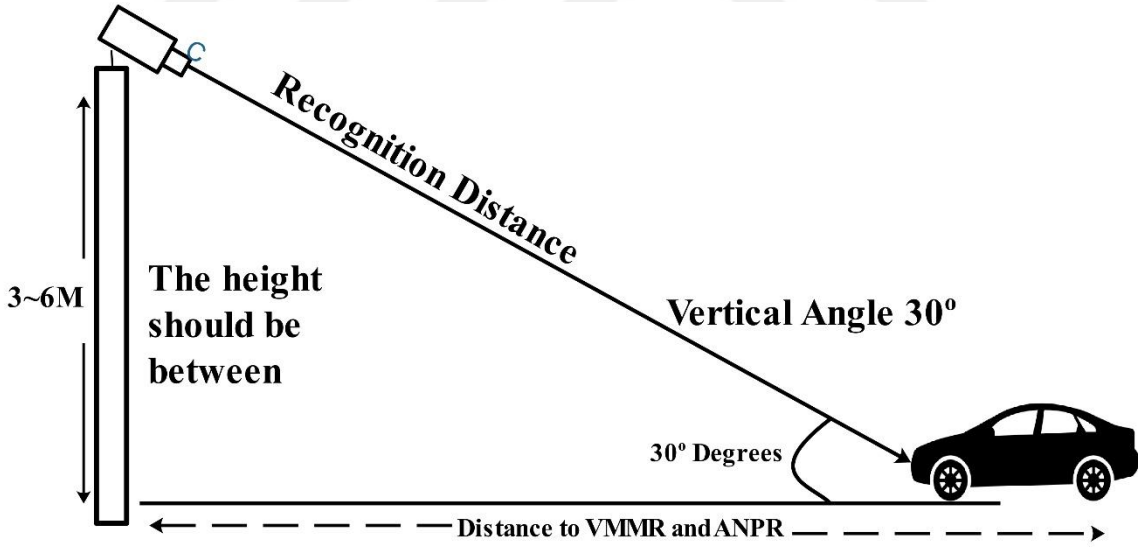

Figure 2.21. Maximum distance and view angle between LP and camera

Camera view angle (Area of Interest): The camera's view angle, denoted as 'Φ' in Figure **2.21**, determines the area on the ground/road that is of interest. Adjusting or deciding the appropriate camera view angle is a challenging task for the ALPR system [108].

System internal RAM and Processor (Fast Processing): The internal RAM and processor specifications of the ALPR system are also crucial factors that can impact its performance and cost. Choosing the right specifications for both is also a challenging task, Figure 2.22 Show different classification challenges by Diagram [108].

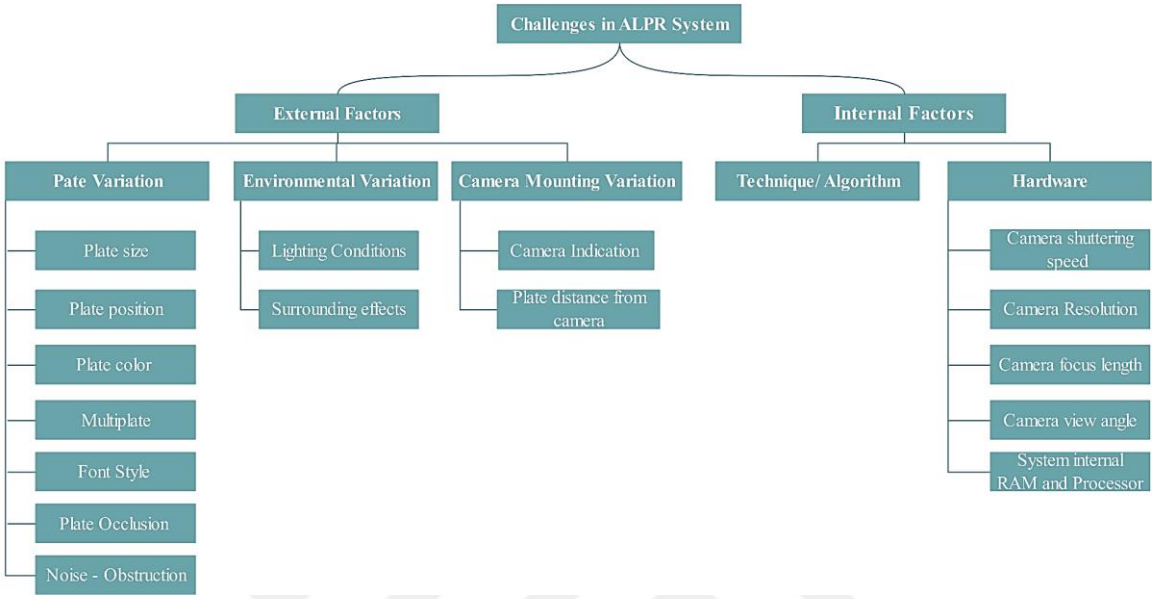

Figure 2.22. Classification of different ALPR challenges

2.6. Limitations of Existing System

Below are some significant constraints of the current system. The differences in plate types or surroundings make it difficult to identify and recognize license plates. These can be summarized as follows: [109]:

1- Location: License plates can be found in various parts of an image.

2- Poor resolution: The plate may be too far away, or the camera used may have low-quality black and white settings.

3- Blurry images: Images with motion blur are hard to process.

4- Quantity: An image may contain no plates or multiple plates.

5- Size: The plates can have different sizes based on the camera's distance and zoom level.

6- Poor lighting and low contrast: Overexposure or shadows can cause difficulties in reading the plate.

7- Different font: Custom or vanity plates can have unique fonts.

8- Lane changes: Vehicles may change lanes during plate recognition, affecting the camera's angle of view.

9- Obstructions: Objects like tow bars or dirt can obstruct parts of the plate.

10- Lack of coordination between countries or states: Plates from different states or countries may have the same number but different designs.

11- Color: Plates can have different background colors and characters due to capturing devices or plate types.

12- Font: Plates from different nations may have different fonts and languages.

13- Inclination: Plates may be tilted.

14- Other elements: Plates may contain frames and screws in addition to characters.

2.7. Car Make and Model

The automotive industry frequently uses various acronyms and terms that may be unfamiliar to many individuals. Terms such as "make" and "model" can initially seem confusing. If you are uncertain about the distinction between these two, you are not alone; most people only learn the difference between make and model when purchasing their first vehicle.

In everyday life, makes and models are more prevalent than you might realize. The automotive brands featured in TV commercials, along with the vehicles they advertise, serve as tangible examples of these terms. Essentially, "make" and "model" classify cars within a recognizable system, much like how various coffee or soda brands and their flavors are categorized [110,111].

The terms "make" and "model" are used to describe a specific vehicle and are often used together, though they can sometimes be mentioned separately. When you think of a vehicle’s name, you likely associate it with both its make and model. For example, if someone asks what kind of car you drive, you might respond with a description such as "a small blue car," or more specifically, by stating the make and model, such as "Toyota Camry." In some cases, you may simply refer to the model, saying "Civic" instead of "Honda Civic," as it is generally understood that all Civics are made by Honda. However, it is essential to recognize the distinction between make and model, particularly when buying or selling a vehicle.

2.7.1. Make

Simply put, the "make" refers to the brand of the vehicle, while the "model" is the specific product. For example, Nabisco is the brand, and Oreos are the specific product. According to DifferenceBetween.net, popular car makes include Ford, Chevrolet, Nissan, Honda, Toyota, and Volkswagen. Each make produces multiple models. For instance, Toyota offers a variety of models, such as the Camry, Corolla, 4Runner, and Tundra. Car makes are typically global companies with operations in multiple countries, though they usually have a specific country of origin where their

main headquarters are located. For example, Toyota is based in Japan, Volkswagen in Germany, and Ford in the United States.

Additionally, some car makes are subsidiaries of parent companies that are also considered makes. For instance, Fiat Chrysler Automobiles (FCA) serves as the parent company for makes such as Dodge and Jeep, while Fiat and Chrysler remain their own distinct brands. Similarly, Acura is a luxury make owned by Honda, and Lexus is a luxury brand under Toyota.

2.7.2. Model

Once you know the make or brand, the model further specifies the vehicle. According to Pediaa.com, models are typically distinguished by names or initials. BMW, for instance, primarily uses initials and numbers to differentiate its models. The company offers SUV models ranging from X1 to X7, sedan models numbered 2 through 8, as well as the i3 and i8. In contrast, companies like Ford assign distinct names to each model, such as the Ford Explorer or Expedition for SUVs and the Ford Fusion or Fiesta for sedans. Adding the model year to the description completes the vehicle's identification, such as "I drive a 2018 Ford Explorer." Additionally, trim levels exist within each model, representing another factor that determines the vehicle's specifications and price [110,111].

2.8. The Usefulness of the System in Various Areas of Life

2.8.1. Security Enhancement:

The integration of ANPR/ALPR with Vehicle Make and Model Recognition (VMMR) significantly strengthens campus security at Firat University. This system not only captures the license plate information but also identifies the make and model of each vehicle entering the campus, thereby providing an additional layer of security. By incorporating both ANPR/ALPR and VMMR, the system offers a more comprehensive approach to vehicle monitoring. This dual functionality enables the creation of a detailed database that facilitates the identification of authorized vehicles, streamlining campus access control. Moreover, the system improves the ability to detect anomalies, such as unauthorized vehicles or discrepancies in vehicle identification, further enhancing security measures. The integration of these advanced technologies, including MobileNet-V2, YOLO, and Paddle OCR, ensures high accuracy and efficiency in real-time vehicle recognition, making it a vital tool in modern campus security infrastructure [112].

2.8.2. Access Control Precision:

The integration of VMMR adds a higher level of precision to access control systems. By identifying authorized vehicles not only through their license plates but also by their make and model, the system ensures a more accurate and secure entry process. This dual-layered verification reduces the risk of unauthorized access, as the combination of license plate data with vehicle make and model information offers an additional safeguard. Such specificity enhances the overall security framework, creating a more robust access control mechanism that contributes to a safer campus environment. Moreover, the use of advanced deep learning architectures like MobileNet-V2, YOLO, and Paddle OCR ensures high accuracy and efficiency in real-time recognition, further strengthening the system's reliability in security applications [112].

2.8.3. Enhanced Tracking and Monitoring

The integration of AN/LPR with Vehicle Make and Model Recognition (VMMR) significantly enhances the tracking and monitoring of vehicular movements on campus. By combining these technologies, the system generates a comprehensive log of the vehicles entering and exiting, capturing both license plate information and detailed vehicle characteristics such as make and model. This detailed data allows for more effective management of parking spaces, improved regulation of traffic flow, and streamlined campus logistics. Furthermore, the use of advanced deep learning architectures, including MobileNet-V2, YOLO, and Paddle OCR, ensures high accuracy and efficiency in real-time data capture and analysis. This comprehensive monitoring system not only supports better operational management but also contributes to a more secure and organized campus environment [113].

2.8.4. Preventing Vehicle Misuse

By recognizing the make and model of vehicles, the system helps prevent the misuse of university-owned vehicles. Any unauthorized use or attempts to introduce non-compliant vehicles can be quickly detected, ensuring that campus transportation resources are managed responsibly. This capability enhances security by allowing for the prompt identification of vehicles that do not meet campus regulations, contributing to the overall efficient and responsible use of university transportation assets [113].

2.8.5. Campus Traffic Optimization

By leveraging AN/LPR technology alongside Vehicle Make and Model Recognition (VMMR), the university can effectively optimize campus traffic management. The data collected through these systems can be analyzed to pinpoint peak traffic hours, popular parking locations,

and areas prone to congestion. This information allows the university to implement strategic measures to improve traffic flow, designate specific parking zones, and enhance overall transportation efficiency. These insights contribute to a more organized and efficient campus transportation system [114].

2.8.6. Emergency Response Preparedness

In emergency situations, the integration of (ANPR/ALPR) with Vehicle Make and Model Recognition (VMMR) provides crucial data for emergency response preparedness. The system's ability to rapidly identify and categorize vehicles enables a more efficient response to incidents, such as medical emergencies or security threats. By leveraging this technology, responders can make quicker and more informed decisions, improving overall response times and effectiveness [114].

2.8.7. Database Integration for Accountability

Integrating vehicle make and model data with existing university databases enhances accountability. This system allows the university to maintain a detailed record of authorized vehicles, track usage patterns, and generate reports for administrative purposes. Such integration adds an additional layer of control and transparency to campus operations, improving overall management efficiency [115].

2.8.8. User-Friendly Experience

For authorized users, such as students and faculty, the integration of ANPR/ALPR with Vehicle Make and Model Recognition (VMMR) significantly enhances the campus entry experience. The automated entry process, which relies on both license plate data and vehicle characteristics, offers greater convenience by reducing wait times and streamlining access. This system improves efficiency while ensuring that only authorized vehicles are granted entry [115].

2.8.9. Technological Advancement and Image

Firat University's motivation to integrate advanced technologies reflects a commitment to technological advancement and innovation. The adoption of cutting-edge solutions not only enhances security but also contributes to the university's image as a modern and forward-thinking institution [115].

2.8.10. Record Keeping and Accountability

ANPR/ALPR systems generate a detailed log of all vehicles entering and exiting the campus, thereby establishing a robust accountability mechanism. This record is highly valuable for administrative purposes, such as tracking the use of university vehicles, managing parking allocations, and generating reports on campus transportation patterns. Such data provides insights that can improve the efficiency of campus operations and transportation management [112].

2.8.11. Integration with Campus Management Systems

The deployment of ANPR/ALPR systems at Firat University aligns with the broader trend of integrating advanced technologies into campus management systems. By linking license plate recognition data with existing campus databases and security systems, the university is able to implement a holistic approach to security and access control, enhancing both efficiency and effectiveness.

Finally, the integration of ANPR/ALPR technology with Vehicle Make and Model Recognition (VMMR) at the entrance gate of Firat University in Turkey is driven by a comprehensive strategy focused on improving security, precision in access control, traffic optimization, and emergency response preparedness. This forward-thinking initiative reflects the university's commitment to utilizing technology to enhance campus safety, operational efficiency, and the overall experience for its community members [114].

3. INTRODUCTION TO DEEP LEARNING

Today, as technology advances rapidly across all aspects of life, it continuously enhances the quality of services. This progress is particularly significant in the field of vehicle transportation, where technology plays a crucial role. The automobile transport system is essential for traffic monitoring, crime detection, stolen vehicle tracking, and various security applications. This system can be visualized as a complex network of interconnected nodes shown in Figure 3.1

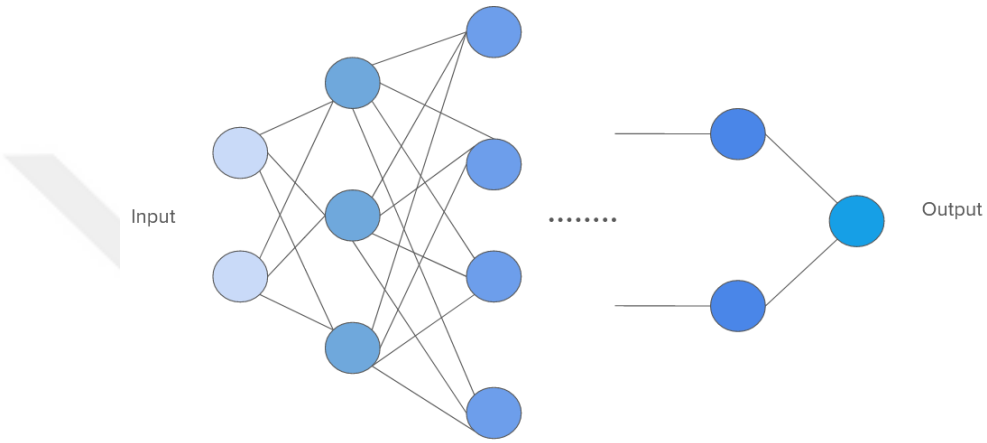

Figure 3.1. Connection of Neurons [114]

The functioning of the human brain was used as the model for a dense network made up of many neurons piled on top of one another. Every node is a neuron, and it is connected to every other neuron in the hierarchy below. This represents the way that information is transferred among neurons.

Deep learning, a subset of machine learning (ML), focuses on training artificial neural networks with extensive datasets to discern patterns and generate predictions or decisions. This innovative approach is essential for tackling challenging problems across diverse domains, including computer vision, natural language processing, speech recognition, and autonomous vehicles. By utilizing the capabilities of deep neural networks, deep learning facilitates the analysis of complex data structures and relationships, thereby driving significant advancements and innovations within these fields. The hierarchical architecture of deep learning models enables the extraction of high-level abstractions from raw data, leading to improved performance and accuracy in various applications [114].

For example, by analyzing medical pictures, such as X-rays, CT scans, and MRIs, deep learning algorithms have proven to be highly useful in the healthcare industry by improving the precision and speed of disease diagnosis and treatment. One noteworthy application is the training

of deep learning models to accurately identify cancerous growths in these images. These models have the ability to identify cancers early on, which raises the possibility of a successful intervention and may even save lives. This application demonstrates how deep learning has the potential to revolutionize medical diagnostics by giving doctors the tools they need to identify key health issues early and accurately, improving patient outcomes [116].

Another prominent example of deep learning application is in the domain of autonomous vehicles. Deep learning algorithms are crucial in enabling self-driving cars to navigate intricate environments and make informed decisions based on real-time data collected from sensors and cameras. By training neural networks on extensive datasets encompassing diverse driving scenarios, these vehicles can learn to accurately identify and interpret objects such as pedestrians, other vehicles, and traffic signals. This capability allows autonomous systems to make judicious decisions about safe navigation and maneuvering. The integration of deep learning in this context not only enhances the vehicle's ability to operate autonomously but also contributes to the broader goal of improving road safety and advancing transportation technologies [114].

However, we should first have a general grasp of what ML is before delving further into the specifics of DL. A clear outline of Machine Learning Time is shown in Figure 3.2 that illustrates the connection between Artificial Intelligence, Machine Learning, and Deep Learning.

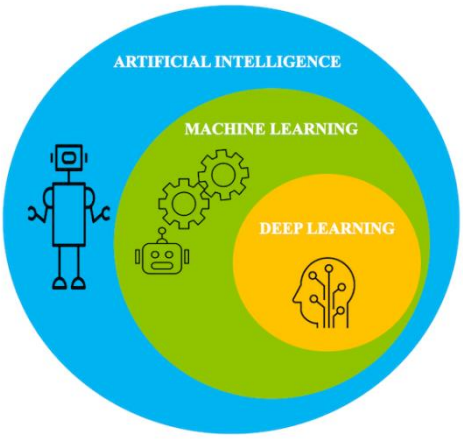

Figure 3.2. Everything is a subset of AI [116]

The field devoted to creating machines that can simulate human behavior is known as artificial intelligence (AI). A particular aspect of artificial intelligence called machine learning (ML) is concerned with teaching algorithms to make predictions or judgments by examining input data. These algorithms learn from the data itself to enhance their performance, which increases precision over time without the need for particular reprogramming [109].

At its core, Machine Learning (ML) fundamentally relies on principles of linear algebra and calculus. These mathematical frameworks are crucial for developing and understanding the algorithms that drive ML models. Linear algebra provides the tools for representing and manipulating data, while calculus is essential for optimizing these algorithms to enhance their performance [117]. Thus, despite the complex applications and outcomes of ML, its foundational techniques are rooted in these fundamental mathematical disciplines shown in Figure 3.3.

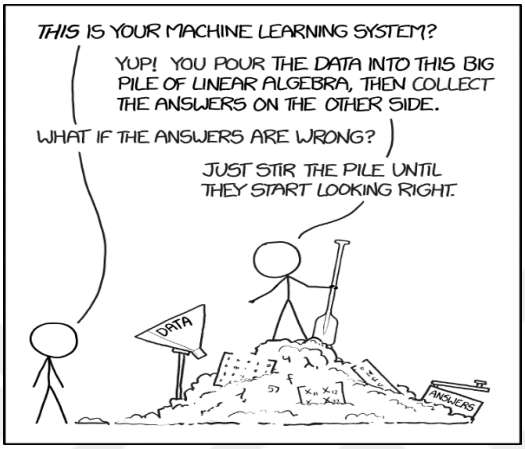

Figure 3.3. Machine Learning [117]

A dataset containing input data and matching output labels is used to train an algorithm in machine learning. The algorithm gains the ability to recognize relationships and patterns in the data through this training process. After being trained, the algorithm can use what it has discovered to forecast or decide on the basis of brand-new, untested data. Developing models that perform well on training data and effectively generalize to new data is the main goal of machine learning. This allows the models to provide precise predictions and well-informed decisions in a variety of scenarios [117].

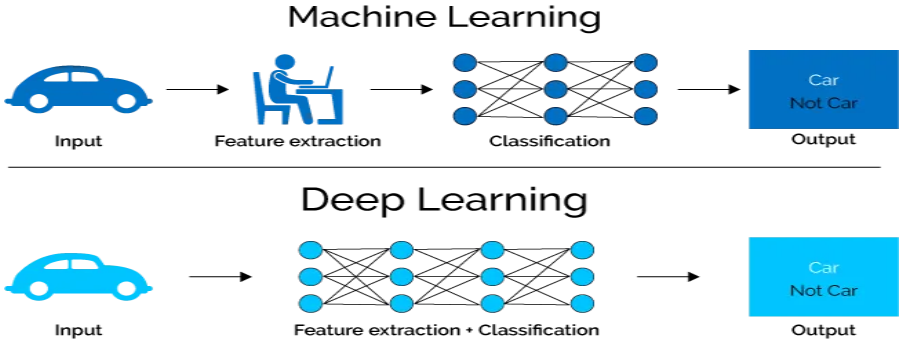

Figure 3.4. Machine learning vs Deep learning [117]

A fundamental difference between ML and DL models lies in how feature extraction is conducted shown in Figure 3.4. In traditional machine learning, humans usually perform feature extraction manually by identifying and selecting pertinent attributes from the data. Conversely, deep learning models are structured to autonomously discover and learn features from raw data through multiple layers of abstraction. This ability to self-learn enables deep learning models to identify intricate patterns and representations without explicit human input, making them well-suited for complex datasets and tasks where manual feature engineering may be impractical [119].

There are three main types of machine learning:

supervised learning

unsupervised learning

reinforcement learning

Supervised Machine Learning: Supervised machine learning is a technique where a neural network learns to make predictions or classify data using labeled datasets. In this approach, both input features and corresponding target variables are provided. The neural network improves its predictions by minimizing the error between its predicted outputs and the actual target values, a process known as backpropagation. During training, the algorithm is exposed to labeled data, this process allows the model to establish a mapping between input data and the appropriate output labels, enabling it to generalize and make accurate predictions on new, previously unseen data. Deep learning algorithms, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), are commonly employed in supervised learning tasks, including image classification and recognition, sentiment analysis, and language translation. These advanced models are particularly effective at handling complex data and extracting meaningful patterns from large datasets [118].

Unsupervised Machine Learning: Unsupervised machine learning is a technique in which neural networks learn to identify patterns or cluster data without the use of labeled datasets. In this approach, there are no predefined target variables; instead, the model must autonomously uncover hidden structures or relationships within the data. During training, the algorithm operates on unlabeled data and strives to detect patterns or groupings independently. This approach is widely utilized for tasks like clustering and dimensionality reduction. Deep learning algorithms, such as autoencoders and generative models, are often applied to unsupervised learning tasks, such as identifying clusters, reducing the dimensionality of data, and detecting anomalies. These advanced models enable the extraction of meaningful insights from complex and high-dimensional datasets [116].

Reinforcement Machine Learning is a methodology wherein an agent learns to make optimal decisions within a specific environment to maximize a reward signal. The agent engages with the environment by performing actions and receiving feedback in the form of rewards or penalties.

Over time, this interaction enables the agent to develop policies, or sequences of actions, aimed at maximizing cumulative rewards. This learning approach is driven by trial and error, with the feedback—positive or negative—directing the agent's decision-making process. Deep learning methods can be applied to reinforcement learning to enhance the development of these policies. For instance, Deep Q-Networks (DQN) and Deep Deterministic Policy Gradient (DDPG) are advanced algorithms used in deep reinforcement learning to tackle complex tasks such as robotics and game playing. These techniques enable the agent to handle high-dimensional state spaces and learn sophisticated strategies through extensive interaction with the environment [118].

In recent years, deep learning—a specialized subfield of machine learning—has significantly transformed the landscape of artificial intelligence, offering the potential to revolutionize a multitude of industries and academic disciplines. Deep learning is distinguished by its ability to automatically learn and model complex patterns in data, achieving remarkable performance improvements across diverse applications. These include tasks such as image and speech recognition, natural language processing, and autonomous driving, where deep learning has set new benchmarks in accuracy and efficiency. Its advancements continue to drive innovation and progress in these and other fields, highlighting its transformative impact on technology and society.

Deep learning techniques are fundamentally based on artificial neural networks, which are modeled after the intricate network of neurons in the human brain. In contrast to traditional machine learning algorithms that necessitate manual feature extraction, deep learning models have the capability to automatically learn and extract hierarchical features directly from raw data. This ability allows them to identify and capture complex relationships and patterns that were previously difficult to represent with conventional methods. Consequently, deep learning offers a more nuanced and effective approach to understanding and interpreting data, significantly advancing the field of machine learning [114].

This research focuses on investigating the fundamental principles, recent advancements, and practical applications of deep learning. By analyzing the foundational architecture of deep neural networks, the optimization techniques that facilitate their training, and the diverse array of neural network architectures, this study seeks to offer a comprehensive overview of this dynamic and swiftly advancing domain. Through a detailed examination of these elements, we aim to illuminate the key developments and emerging trends within deep learning, thereby contributing to a deeper understanding of its current and future impact in various field [119].

Gaining an understanding of how deep learning models work requires an understanding of the training process. The main optimization methods covered in this study include stochastic gradient descent and backpropagation, which are fundamental to neural network training. This section also covers regularization techniques that aim to reduce overfitting, a deep learning concern that frequently arises. Through our examination of various regularization methods and optimization

tactics, we hope to offer a thorough grasp of the mechanics behind efficient training in deep learning models.

From convolutional neural networks (CNNs) to recurrent neural networks (RNNs) that rule sequential data to transformers that revolutionize natural language processing, the landscape of deep learning architectures is vast. Our investigation aims to disentangle these designs' underlying mechanics and clarify their areas of use [119].

Furthermore, this research examines the recent development of transfer learning and its significant impact on addressing the issue of data scarcity. Pretrained models, such as BERT and GPT, have demonstrated exceptional performance improvements across various domains, creating new opportunities for research and innovation across diverse industries. By leveraging the capabilities of these advanced models, transfer learning facilitates the effective application of learned knowledge to new, data-limited tasks, thus enhancing the versatility and efficiency of deep learning solutions [116].

This thesis also explores the ethical implications associated with deep learning, recognizing the potential biases, privacy issues, and ethical dilemmas that emerge as these models become increasingly embedded in critical decision-making processes. By addressing these concerns, the research aims to highlight the importance of developing and implementing robust frameworks to ensure the responsible use of deep learning technologies. This includes examining strategies to mitigate bias, safeguard privacy, and navigate the ethical challenges inherent in deploying such advanced models in sensitive and impactful areas [117].

The goal of this thorough examination of deep learning is to give readers a fundamental grasp of the field background, functioning, and future prospects of the field. Through an exploration of the historical background, fundamental ideas, and real-world implementations, we hope to illuminate the prospects and difficulties that await in the ever-changing field of deep learning.

Parking management, tolling, law enforcement enforcement, and border control are just a few of the many uses for AN/LPR systems. Services in many facets of life are getting better as a result of technological advancements. The vehicle transport system is one important sector in which technology is playing a significant part. In addition to other security applications, this system is crucial for traffic monitoring, crime detection, detecting stolen cars, and other security applications. An automatic license plate recognition system's original concept was created by a police station in the UK. With the goal of increasing its accuracy, this technology has seen improvements in both hardware and software throughout time. The system's present accuracy level, however, is still below the traffic standards because to issues including dim lighting, an excessive amount of plates, and unreadable plates [120].

Using image processing, Automatic License Plate Recognition (ALPR) technology recognizes cars by their license plates. Law enforcement on public roadways and automated traffic

monitoring depend on it. The article talks about an LPR algorithm for a smart parking system that recognizes license plates automatically for entry using images. By improving image quality, retrieving license plate information, and separating characters, the system seeks to increase accuracy. The door will automatically open if a car's license plate is recognized; otherwise, it will stay closed.

The number of active automobiles has increased dramatically recently, which has also increased illegal activity. Because of their quick proliferation, it is hard to keep track of every vehicle, hence it is vital that the appropriate authorities do so. Automatic number plate recognition is becoming increasingly essential as the number of vehicles increases. In regard to the large number of incidents that happen every year, it is very crucial to monitor vehicle speeds with this technology [120].

3.1. Artificial neural networks

The structure and functionality of artificial neural networks are fundamentally inspired by the principles governing human neurons. Commonly referred to as neural networks or neural nets, these systems begin with an input layer, which receives data from external sources and relays it to the subsequent hidden layer. In this hidden layer, each neuron calculates a weighted sum of inputs from the preceding layer and passes the result to neurons in the next layer shown in Figure 3.5. These connections carry specific weights, allowing the model to optimize the influence of inputs from prior layers. During the training phase, these weights are adjusted to enhance the overall performance of the model [121].

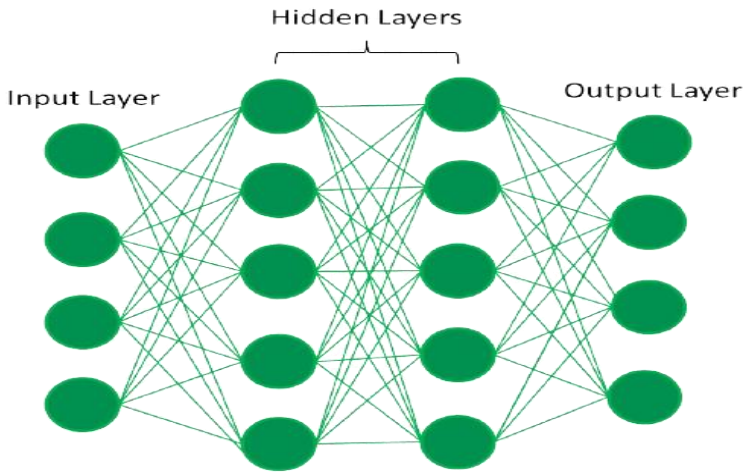

Figure 3.5. Fully Connected Artificial Neural Network [121]

Artificial neural networks contain artificial neurons, sometimes referred to as units. These artificial neurons are layered and make up the whole Artificial Neural Network. The complexity of neural networks is intricately linked to the complexity of the underlying patterns within the dataset, regardless of whether a layer comprises a dozen units or millions. Typically, an artificial neural network (ANN) is structured with an input layer, one or more hidden layers, and an output layer. The input layer is responsible for receiving data from external sources, which the neural network subsequently analyzes or learns from. This architecture allows the network to process and interpret data through its hidden layers, ultimately generating predictions or classifications in the output layer [121].

In a fully connected artificial neural network, the architecture includes an input layer followed by one or more sequentially connected hidden layers. Each neuron within the network receives input from the preceding layer's neurons or directly from the input layer. The output of each neuron in one layer serves as the input for neurons in the next layer. This transformation continues throughout the network, refining and processing data as it moves through the hidden layers. Ultimately, the output layer produces the network's final response based on the thoroughly processed data.

In typical neural networks, units are linked across various layers, with each connection carrying a weight that determines the influence one unit exerts over another. As data flows through these connected units, the network gradually learns and refines its comprehension of the information. This iterative process, regulated by the connection weights, allows the network to produce an output from the final layer. During training, weight adjustments are made, improving the network's capacity to accurately predict or classify based on the input data [121].

3.2. Deep Learning vs. Traditional Machine Learning

Fundamentally, both traditional machines learning and deep learning are approaches to discovering the underlying patterns in data and finding solutions.

In traditional machine learning, algorithms are developed to learn from and forecast data. Labeled data, or previously classed or categorized data, can be used to train these algorithms, which are generally based on statistical models. Decision trees, random forests, and support vector machines are a few examples of conventional machine learning techniques.

Conversely, artificial neural networks are employed in deep learning—a specialized subset of machine learning—to learn from data and make predictions. These networks are modeled after the human brain's structure, consisting of layers of interconnected nodes that process information hierarchically. Deep learning algorithms can be trained on substantial volumes of either labeled or unlabeled raw data, enabling them to uncover complex relationships and patterns within the data [122].

DL offers several advantages, chief among them being its ability to automatically extract features from raw data—a task that traditional ML finds both challenging and time-intensive. Moreover, deep learning models are highly suitable for applications in image and speech recognition due to their scalability and capacity to handle high-dimensional data.

However, deep learning models often necessitate larger training datasets and specialized hardware compared to traditional machine learning models. They also tend to be more complex and computationally intensive. Additionally, interpreting deep learning models can be challenging, which makes it difficult to understand the rationale behind their predictions [123].

3.3. Difference between Machine Learning and Deep Learning

Both machine learning and deep learning are subsets of artificial intelligence, sharing several similarities while also exhibiting distinct differences explained in Table 3.1. [120].

Table 3.1. Difference between machine learning and deep learning [120]

	Machine Learning	Deep Learning
1	Employs statistical techniques to reveal hidden correlations and patterns within the dataset	Reveals the hidden correlations and patterns within the dataset through the use of artificial neural network architecture.
2	Able to operate on a reduced dataset size	Demands a significantly larger dataset compared to ML
3	Better for the low-label task.	Better for complex task like image processing, natural language processing, etc
4	Model training takes a shorter amount of time.	Increases the model's training time.
5	Important features are manually extracted from an image and used to build a model that can identify an object in it.	Images undergo automatic processing to extract essential features, making it a thorough learning process
6	Less complicated and simpler to understand the outcome.	It functions more intricately, making it difficult to understand the outcome in black box interpretations.
7	In contrast to deep learning, it requires less computational power and can efficiently run on a CPU.	A high-performance computer with GPU capabilities is required

3.4. Types of neural networks

Deep learning models excel in applications such as natural language processing, speech recognition, image detection, and image classification due to their ability to automatically extract features from raw data. The most commonly used architectures in deep learning include feedforward neural networks, convolutional neural networks (CNNs), and recurrent neural networks (RNNs). A feedforward neural network (FNN) is a type of artificial neural network (ANN) characterized by a linear flow of information, making it suitable for various applications, including natural language processing, audio recognition, and image classification.

CNNs are specifically designed for image and video recognition tasks. Their ability to automatically extract features from images makes CNNs highly suitable for applications such as object detection, image segmentation, and image classification

Recurrent Neural Networks (RNNs) are a type of neural network designed to process sequential data, such as time series and natural language. Their ability to maintain an internal state that captures information from previous inputs makes RNNs particularly effective for tasks such as speech recognition, natural language processing, and language translation [117].

3.5. Applications of Deep Learning

Deep learning has three main usages reinforcement learning, natural language processing (NLP), and computer vision [122].

3.5.1. Computer Vision

Machines can recognize and comprehend visual data by means of deep learning models in computer vision. Among the principal uses of deep learning in computer vision are:

- Object detection and recognition: Robotics, surveillance, and self-driving cars are just a few of the duties that robots can now accomplish according to deep learning models that can locate and recognize things in images and videos.
- Image classification: To categorize images into groups like plants, animals, and buildings, deep learning models can be employed. This finds value in quality control, image retrieval, and medical imaging applications.
- Image segmentation: By segmenting images into distinct regions, deep learning models can be used to recognize particular features throughout images.

3.5.2. Natural Language Processing (NLP)

The Deep Learning Model in NLP can help machines comprehend and produce human language. Among the principal uses of deep learning in NLP are [122]:

- Automatic Text Generation: DL models can be trained on large text corpora, enabling them to generate new text such as summaries and essays automatically based on the learned patterns.
- Language Translation: Deep learning models facilitate communication across different languages by automatically translating text from one language to another.
- Sentiment Analysis: Deep learning models can determine whether a text expresses a positive, negative, or neutral sentiment. This capability is widely used in applications such as social media monitoring, customer service, and political analysis.

- Speech Recognition: Deep learning models can recognize and transcribe spoken language, enabling advancements in voice-activated devices, voice search, and speech-to-text conversion applications.

3.5.3. Reinforcement Learning

In reinforcement learning, deep learning is employed to train agents to act in ways that maximize a reward signal. Key applications of deep learning within reinforcement learning include the following [122]:

- Game playing: Deep reinforcement learning models have shown the capability to outperform human experts in games such as Atari, Chess, and Go.
- Robotics: Deep reinforcement learning models can be utilized to train robots for complex tasks, including object manipulation, navigation, and grasping.
- Control systems: Deep reinforcement learning models can be applied to manage complex systems, such as supply chain optimization, traffic control, and power grid management.

3.5.4. Challenges in Deep Learning

Though there are still certain issues that need to be resolved, deep learning has made tremendous strides in many different domains. The following are some of the primary obstacles in deep learning [124]:

1. Data availability: To learn from it, a lot of data is needed. Obtaining as much training data as possible is crucial for deep learning applications.
2. Computational Resources: It is computationally expensive to train the deep learning model because it needs specialized hardware, such as GPUs and TPUs.
3. Time-consuming: Working with sequential data might require a significant amount of time, potentially months or days, depending on the available computer resources.
4. Interpretability: Models for deep learning are intricate and operate like a mysterious black box. Reading the results is quite difficult.
5. Overfitting: Repeated training of the model causes it to become excessively specialized for the training data, which results in overfitting and subpar performance on fresh data.

3.5.5. Advantages of Deep Learning

1. High accuracy: Deep learning algorithms achieve state-of-the-art performance across various domains, including natural language processing and image recognition.
2. Automated feature engineering: Deep learning algorithms can autonomously identify and learn relevant features from data, eliminating the need for manual feature engineering.

3. Scalability: Deep learning models are capable of handling complex and large-scale datasets, allowing them to be trained on vast amounts of data.
4. Flexibility: Deep learning models are adaptable and can be applied to a range of tasks and data types, such as text, speech, and images.
5. Continual improvement: Deep learning models can continuously enhance their performance as more data becomes available [125].

3.5.6. Disadvantages of Deep Learning

1. High computational requirements: Training and optimizing deep learning models require substantial data and computational resources.
2. Data Requirements: Deep learning models often need large volumes of labeled data, which can be costly and time-intensive to acquire.
3. Interpretability: Understanding deep learning models can be challenging, making it difficult to discern how they reach specific conclusions.
4. Overfitting: Deep learning models may perform poorly on new, unseen data due to overfitting to the training dataset.
5. Black-box nature: Deep learning models are frequently considered "black boxes," as their internal processes and prediction mechanisms can be hard to interpret.

In summary, while deep learning offers numerous advantages, such as high accuracy and scalability, it also presents several challenges, including high computational costs, extensive data requirements, and issues with interpretability. Deciding whether to utilize deep learning for a specific task requires a careful evaluation of these limitations alongside its benefits [125].

3.5.7. Deep Learning vs. Traditional Machine Learning

Fundamentally, the goals of both traditional machines learning and deep learning are to identify underlying patterns in data and provide remedies for issues.

Within artificial intelligence, traditional machine learning develops algorithms to learn from and forecast data. Usually built on statistical models, these algorithms can be trained on labeled data that is, data that has already been assigned a classification or categorization. Traditional machine learning techniques include random forests, decision trees, and support vector machines.

On the contrary, artificial neural networks are used in deep learning, a subset of machine learning, to learn from and forecast data. These networks are inspired by the structure of the human brain, which is made up of layers of connected nodes that process information in a hierarchical manner. Large volumes of unlabeled or labeled raw data can be used to train deep learning algorithms to discover complicated relationships and patterns.

DL has several benefits, chief among them the capacity to automatically extract features from unprocessed data, something that classical ML finds challenging and time-consuming. In addition to their high dimensionality and scalability, deep learning models are extremely suitable for image and audio recognition applications.

Deep learning models, nonetheless may also need more training data and specialized technology than conventional machine learning models. They might also be increasingly complex and computationally demanding. Furthermore, it can be more difficult to interpret DL models, which makes it challenging to comprehend how they generate predictions [126].

3.6. Introduction to the YOLO Family

In computer vision, object detection is one of the most important topics. Identifying several categories of visual objects, such as faces, cars, buses, and pedestrians, is a common task in computer vision problems. It is one of those fields that has potential applications in the business sector outside of academics, including autonomous driving, in-vehicle sensing, healthcare, and video monitoring [127].

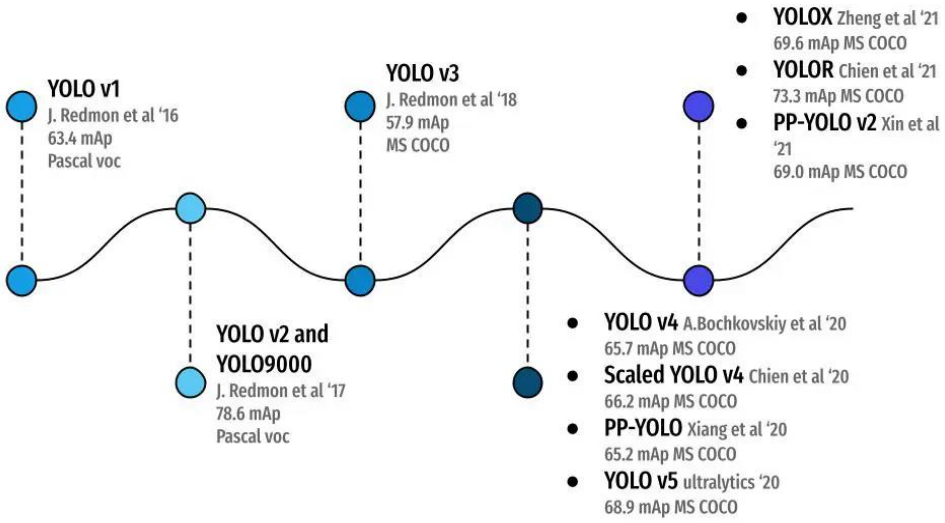

Figure 3.6. Introduction to the YOLO Family [127]

Several use cases necessitate great accuracy and real-time inference speed, particularly autonomous driving. It is therefore crucial to select an object detector that meets the requirements for accuracy and speed. A single-stage object detector called YOLO (You Only Look Once) was developed to meet both objectives (i.e., speed and accuracy). Additionally, we will cover all of the YOLO versions, such as YOLOv1, YOLOv2, YOLOX, and YOLOR, in order to introduce the YOLO family shown in Figure 3.6.

The object detection field has evolved greatly since its inception, and the state-of-the-art structures perform rather well in terms of generalization on a variety of test datasets. However, in order to really appreciate the wonder of these state-of-the-art buildings today, one must first comprehend where the YOLO family originated and how far we have come. There are three classes of algorithms in object detection:

- Based on Traditional Computer Vision
- Two-Stage Deep Learning based algorithms
- The third one is Single-Stage Deep Learning based algorithms

Furthermore, we will talk about the Single-Stage Deep Learning algorithms that are part of the YOLO object detection family. It is our opinion that this blog article is unique since it covers all of the YOLO variations in one place. It will provide you with valuable information about each variation and possibly even help you choose the ideal YOLO version for your project [127].

3.6.1. Introduction to Object Detection

Recall that the objective of image classification is to respond to the query, "What is present in the image?" by designating a particular label for the image in an effort to help the model understand the complete picture. Typically, we address situations in which the image's classification contains a single object. The Santa Claus and a few other items are displayed in Figure 3.7, but Santa is the major object and has a 98% likelihood of being properly categorized. Classification is sufficient to address our query in some situations, such as this one, where an image represents a single object.

Figure 3.7. Image classified as Santa with 98% confidence [128]

Nevertheless, there are numerous situations in which a single label cannot accurately describe what is in the picture, and image categorization is insufficient to address these issues. Consider Figure 3.8, for instance, where the model not only recognizes the presence of three objects

two people and one baseball glove—but also pinpoints their locations. This is called object detection [128].

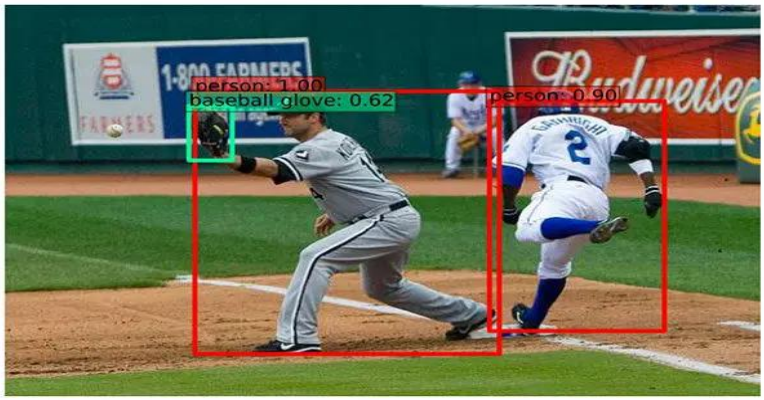

Figure 3.8. Examples of object detection using Single Shot Detectors (SSD) [129]

Figure 3.9 illustrates an additional crucial application of object detection: Automatic License Plate Recognition. Now pose this question to yourself: if you did not know where a car number plate is located, how would you recognize the letters and numbers on it? Naturally, in order to recognize the numbers, you must first use an object detector to locate the number plate. Only then can you use a second method.to identify the numbers.

Figure 3.9. An example of a real-time Automatic Number Plate Recognition system [129]

Object detection analyzes more realistic scenarios where more than one object may present in an image, combining tasks from both classification and location. As a result, object detection is a two-step procedure, where item locations are determined initially. Classifying these bounding boxes into distinct classes is the next stage, as object identification is hindered by all the issues related to image classification. It also has complications with localization and execution speed [129].

3.6.2. Object Detection Challenges

A- Crowded or Cluttered Scenario: Figure 3.10 is a very crowded image due to an excessive number of items. This poses a number of difficulties for the object detect model, including the possibility of massive occlusions, small objects, and uneven scales [130].

Figure 3.10. A large group of people watching a match in a stadium [130]

B- Intra-Class Variance: Accurately identifying objects in the same class, which can have significant variance, presents another significant obstacle to object detection. Identifying objects of the same class can be difficult because, as Figure 3.11 illustrates, there are six distinct dog breeds with varying sizes, colors, fur lengths, ears, and other characteristics [131].

Figure 3.11. Six breeds of dogs [131]

C- Class Imbalance: In particular, image classification has significant challenges in the image domain, and object identification is no exception. This problem affects nearly all modalities, including text, images, and time-series. As seen in Figure 3.12, we refer to it as an imbalance between the foreground- background classes in object detection.

Consider an image with a small number of primary objects to get an idea of how class imbalance could cause issues with object detection. The background fills the remaining space in the image. Consequently, the model would examine numerous places within the image (dataset), with the majority of those parts being regarded as negatives. These drawbacks result in the model not learning any relevant information and have the potential to overwhelm the model during training.

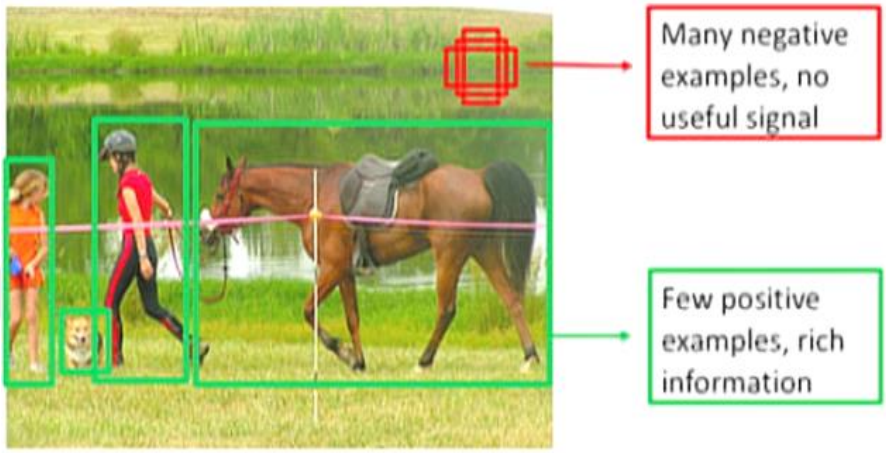

Figure 3.12. Many negative background examples and few positive foreground examples [132]

Object detection faces a range of additional challenges, including occlusion, deformation, viewpoint variation, and varying illumination conditions. These factors can significantly complicate the accurate detection and classification of objects in real-world scenarios. Moreover, the need for real-time detection, particularly in industrial applications, adds another layer of complexity, as object detection models must not only be accurate but also fast enough to process images instantaneously. Balancing accuracy and speed are critical for applications such as autonomous vehicles, security systems, and manufacturing processes, where timely and precise detection is essential [132].

3.6.3. History of Object Detection

Object detection remains one of the most critical and challenging problems in the field of computer vision, and its development over the past decade has been remarkable. The progress in this domain has been substantial, with the research community continuously pushing the boundaries and achieving new state-of-the-art benchmarks year after year. This rapid advancement has been largely fueled by the power of deep neural networks (DNNs) and the significant computational capabilities provided by NVIDIA GPUs, which have enabled the training and deployment of

increasingly sophisticated models. Without these technological innovations, many of the breakthroughs in object detection would not have been possible [127].

The history of object detection can be divided into two distinct eras:

1- The first era, lasting until 2010, was dominated by traditional computer vision approaches,

2- From 2012, a new era began with the rise of convolutional neural networks (CNNs), marked by the groundbreaking victory of AlexNet, an image classification network, in the ImageNet Visual Recognition Challenge. This event ushered in a paradigm shift in object detection techniques, leading to significant advancements in the field.

The distinction between these two eras is also illustrated in Figure 3.13, which outlines the roadmap of object detection, beginning with the Viola–Jones [133] Detector in 2004 and extending to EfficientDet in 2019 [134]. It is important to note that deep learning-based detection methods are further categorized into two-stage detectors and single-stage detectors, each offering unique approaches to object detection tasks.

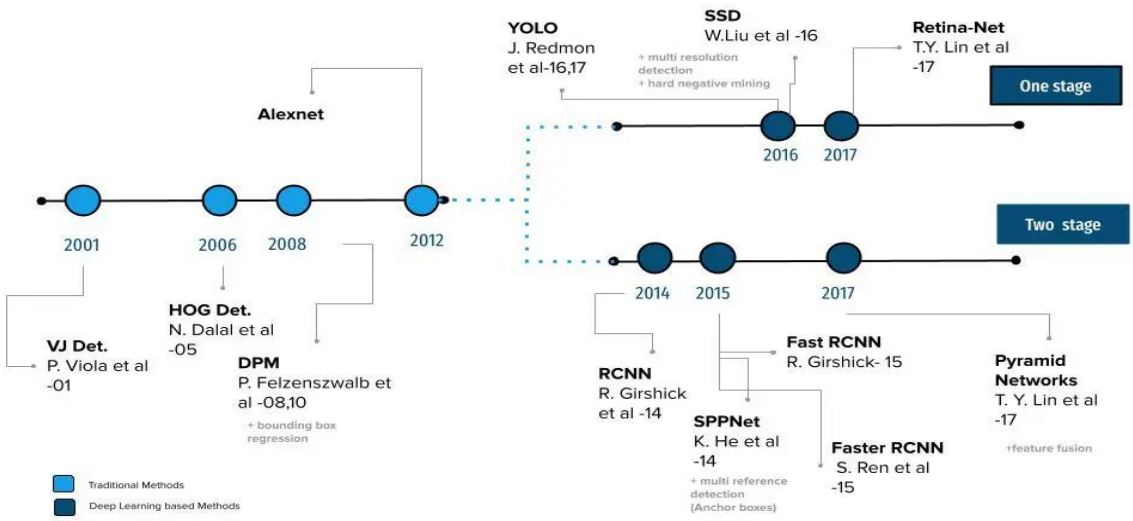

Figure 3.13. A road map of Object Detection [134]

Most traditional object detection algorithms, such as the Viola–Jones detector, Histogram of Oriented Gradients (HOG), and Deformable Parts Model (DPM), relied on manually engineered features like edges, corners, and gradients, which were then processed using classical machine learning techniques. For instance, the Viola–Jones detector, one of the earliest object detection algorithms, was specifically designed to detect frontal human faces. However, its performance was limited when applied to faces in different orientations, such as sideways or upside-down views, highlighting the constraints of early object detection methods in handling variations in pose and appearance.

In 2012[59], The advent of AlexNet, an innovative deep convolutional neural network (CNN) architecture, marked the beginning of a new era in computer vision. Designed to improve performance in the ImageNet competition, AlexNet made a transformative impact by achieving an impressive 84.7% accuracy in the 2012 ImageNet LSVRC-2012 competition significantly surpassing the second-highest entry, which achieved 73.8% accuracy. The success of AlexNet highlighted the potential of deep learning for complex visual recognition tasks, setting a new benchmark in computer vision research and spurring further advancements in CNN development.

It was only a matter of time before these state-of-the-art image classification architectures began to be employed as feature extractors in the object detection pipeline, given the interrelated nature of these tasks and their reliance on learning robust high-level features. In 2014, Girshick et al [135]. demonstrated how convolutional features could be effectively utilized for object detection by introducing the R-CNN (Region-based Convolutional Neural Network) model, which applies a CNN to region proposals. This innovation marked a significant breakthrough, as it showcased the potential of leveraging deep learning for object detection. Since then, the field of object detection has advanced at an unprecedented pace, with continual improvements in accuracy, speed, and efficiency.

As illustrated in Figure 3.13, deep learning-based object detection methods can be categorized into two primary groups. The first group comprises two-stage detection algorithms, which generate predictions through multiple stages. This category includes well-known networks such as R-CNN, Fast R-CNN, and Faster R-CNN. These models typically involve an initial stage where region proposals are generated, followed by a second stage where these proposals are refined and classified, enabling more precise object detection [129].

The second category of object detectors is known as single-stage detectors, which include models such as SSD [136], YOLO, and EfficientDet. As depicted in Figure 3.14, the YOLO (You Only Look Once) family of object detection algorithms has undergone significant evolution since its inception. Starting with YOLOv1 in 2016, which achieved a mean Average Precision (mAP) of 63.4 on the Pascal VOC dataset (consisting of 20 classes), the series has progressed to YOLOR in 2021, attaining an impressive 73.3 mAP on the more complex MS COCO dataset, which contains 80 classes. This evolution exemplifies the remarkable advancements made in academia, demonstrating how persistent research and innovation have propelled the YOLO series to achieve state-of-the-art performance in object detection.

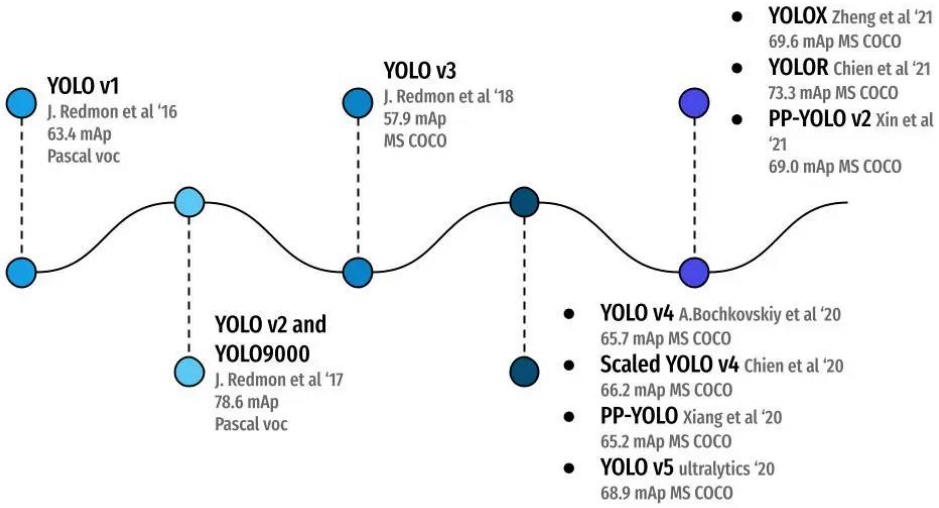

Figure 3.14. History of YOLO [136]

3.6.4. Single-Stage Object Detectors

Single-stage object detectors represent a class of object detection architectures that streamline the detection process into a single stage. These models treat object detection as a straightforward regression problem, where the input image is fed directly into the network, which then outputs both class probabilities and bounding box coordinates simultaneously. Unlike two-stage detectors, single-stage models bypass the region proposal phase, typically found in architectures like Faster R-CNN, which first identify regions of interest that may contain objects. By eliminating the Region Proposal Network (RPN) step, single-stage detectors achieve faster inference speeds, making them more suitable for real-time applications where rapid processing is essential.

Figure 3.15 illustrates the workflows of single-stage and two-stage object detectors differ significantly. In single-stage detectors, the detection head is applied directly to the feature map, allowing the model to predict object classes and bounding boxes in one pass. Conversely, two-stage detectors utilize a region proposal network (RPN) to generate candidate object regions from the feature map before applying the detection head to classify and refine these regions. This two-step approach enables more accurate localization and classification but generally results in higher computational complexity and slower inference speeds compared to single-stage models [127].

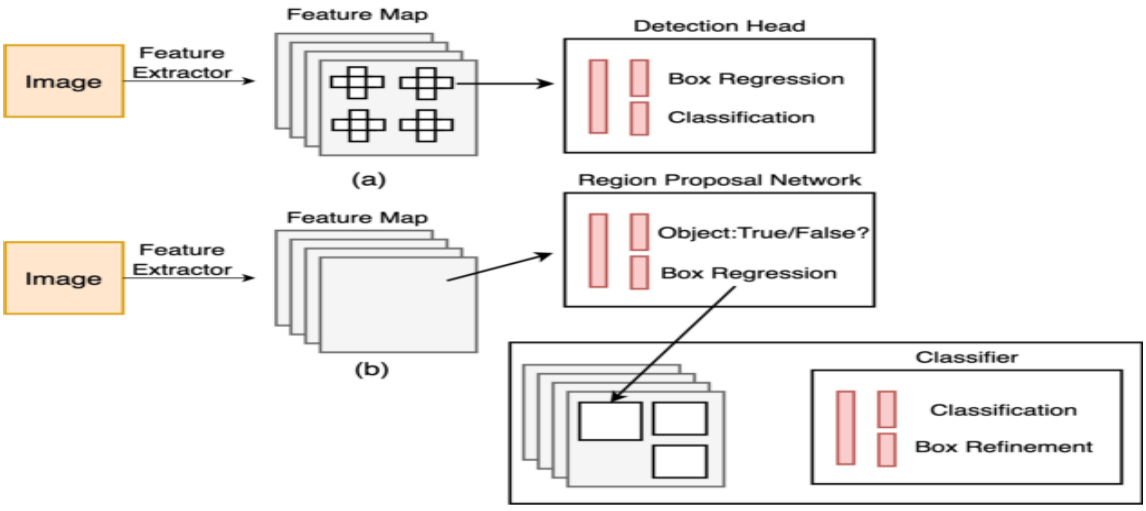

Figure 3.15. Single-Stage (top) and Two-Stage Detector [136]

In the second stage of two-stage object detectors, these proposed regions are further processed to make predictions for each individual region. Models such as Faster R-CNN and Mask R-CNN are among the most widely used two-stage detectors, known for their high accuracy in object detection tasks. Although two-stage detectors generally achieve superior precision compared to single-stage models, their multi-step process results in slower inference speeds. Conversely, single-stage detectors, such as YOLO and SSD, prioritize speed by performing detection in a single pass, making them more suitable for real-time applications despite a potential trade-off in accuracy [136].

3.6.5. YOLOv1

In 2016, Joseph Redmon and colleagues introduced the first single-stage object detector, You Only Look Once [137], Unified, Real-Time Object Detection, at the CVPR conference. YOLO (You Only Look Once) marked a significant breakthrough in the field of object detection, as it was the first approach to treat detection as a regression problem in a single-stage process. The YOLO architecture processes the entire image in a single pass, predicting the location of objects and their class labels simultaneously.

Unlike two-stage detector approaches such as Fast R-CNN and Faster R-CNN, YOLOv1 does not include a proposal generation and refinement stage. Instead, it utilizes a single neural network to predict class probabilities and bounding box coordinates for the entire image in one pass. Since the detection pipeline operates as a unified network, it can be optimized end-to-end, functioning similarly to an image classification network.

The YOLO network is designed for end-to-end training, similar to image classification tasks, which makes the architecture highly efficient. The base YOLO model is capable of processing

images at 45 frames per second (FPS) when benchmarked on a Titan X GPU. To further enhance speed, the authors introduced a lighter version, known as Fast YOLO, which has fewer layers and processes images at an impressive 155 FPS. As shown in Table 3.2., YOLO achieved a mean average precision (mAP) of 63.4, more than doubling the performance of other real-time detectors. This significant performance boost underscores the uniqueness of the model. Both YOLO and Fast YOLO outperform real-time object detection variants of the Deformable Part Models (DPM) by a substantial margin, nearly doubling their mean average precision and FPS, highlighting YOLO's superiority in terms of both accuracy and speed [127,137].

Table 3.2. Real-Time Detectors Quantitative Benchmarks [137]

Real-Time Detectors	Training Data	Evaluation (mAP)	FPS
100Hz Deformable Parts Model	2007	16.0	100
30Hz Deformable Parts Model	2007	26.1	30
Fast YOLO	2007+2012	52.7	155
YOLO	2007+2012	63.4	45

The generalizability of YOLO was evaluated on a diverse range of images, including both artwork and natural images sourced from the internet. The results demonstrated that YOLO significantly outperformed traditional detection methods, such as the Deformable Parts Model (DPM) and Region-Based Convolutional Neural Networks (RCNN). YOLO’s ability to generalize across various types of images, while maintaining both speed and accuracy, highlights its robustness and superiority over these earlier detection models. This performance reinforces YOLO’s effectiveness in handling real-world object detection tasks across different domains.

3.6.6. YOLOv2

In 2017, Redmon and Farhadi introduced their groundbreaking work, YOLO9000 [138]: Better, Faster, Stronger, at the CVPR conference. In this paper, the authors proposed two advanced variants of the YOLO framework: YOLOv2 and YOLO9000. While both variants share an identical architecture, they differ in their training strategies. YOLOv2 was trained on a standard object detection dataset, whereas YOLO9000 was trained using a combination of detection and classification data, enabling it to recognize over 9,000 object categories. This innovative approach demonstrated the scalability and enhanced performance of the YOLO framework, pushing the boundaries of real-time object detection.

YOLOv2 was trained on well-known object detection datasets, such as Pascal VOC and MS COCO, to achieve high accuracy in detecting standard object categories. In contrast, YOLO9000 was designed to predict over 9,000 different object categories by employing a more sophisticated

joint training strategy. This approach involved simultaneously training the model on both the MS COCO dataset for object detection and the ImageNet dataset for object classification. By combining these two datasets, YOLO9000 demonstrated an unprecedented ability to generalize across a vast range of objects, significantly expanding the model's versatility and real-world application potential.

The improved YOLOv2 model introduced several novel techniques that allowed it to surpass state-of-the-art methods such as Faster R-CNN and SSD in both speed and accuracy. One of the key innovations was multi-scale training, which enabled the network to predict at varying input sizes, providing flexibility in the trade-off between speed and accuracy. At a 416×416 input resolution, YOLOv2 achieved 76.8 mAP on the VOC 2007 dataset while maintaining a processing speed of 67 FPS on a Titan X GPU. When using a higher input resolution of 544×544, YOLOv2 reached 78.6 mAP with a reduced speed of 40 FPS.

Figure 3.16 illustrates the performance benchmark of YOLOv2 across different resolutions on a Titan X GPU, compared with other detection architectures like Faster R-CNN, YOLOv1, and SSD. The results clearly show that YOLOv2 consistently outperforms these frameworks in either speed or accuracy, demonstrating a distinct trade-off between mean average precision (mAP) and frames per second (FPS) depending on the input resolution. This balance between speed and accuracy makes YOLOv2 a versatile choice for various real-time object detection tasks [127,138].

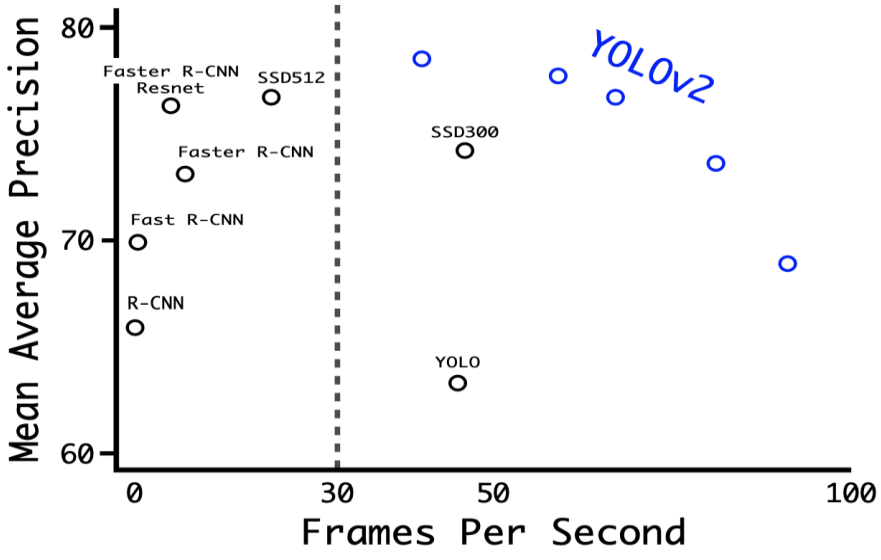

Figure 3.16. Accuracy and speed on Pascal VOC 2007 [138]

3.6.7. YOLOv3

In 2018, Redmon and Farhadi published the paper YOLOv3[139], An Incremental Improvement on arXiv. This paper introduced several significant design changes to the network architecture, building on the foundation laid by YOLOv1 and, more notably, YOLOv2. The authors adapted many of the techniques from these earlier versions while refining and optimizing the model's architecture to enhance both performance and efficiency. These improvements allowed YOLOv3 to better balance speed and accuracy, further solidifying its place as a leading framework in real-time object detection.

This paper introduced a new network architecture known as Darknet-53. Darknet-53 is a significantly larger and more advanced network than its predecessors, offering both increased accuracy and speed. The model was trained at various image resolutions, including 320×320 and 416×416, to optimize performance across different use cases. At a resolution of 320×320, YOLOv3 achieves a mean average precision (mAP) of 28.2 and runs at 45 frames per second (FPS) on a Titan X GPU. Notably, at this resolution, YOLOv3 matches the accuracy of the Single-Shot Detector (SSD321) but operates three times faster, as demonstrated in Figure 3.17. This combination of speed and accuracy makes YOLOv3 a highly efficient model for real-time object detection tasks [127][139].

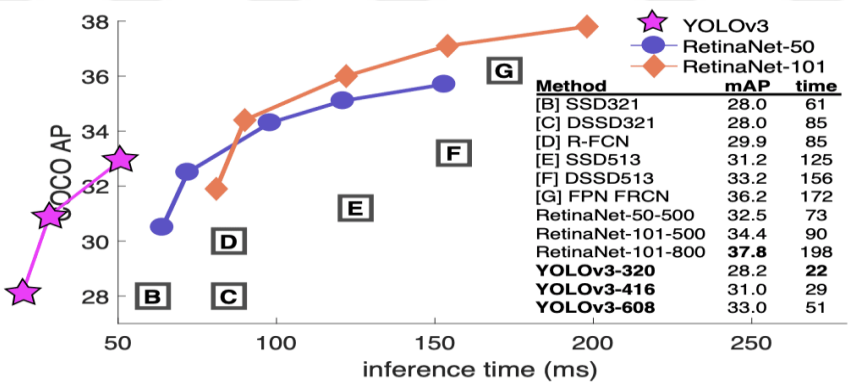

Figure 3.17. Quantitative Benchmark showing accuracy (COCO mAP) vs. inference time (ms)[139]

3.6.8. YOLOv4

YOLOv4 emerged from extensive research and experimentation, incorporating a range of innovative techniques designed to enhance both the accuracy and speed of Convolutional Neural Networks (CNNs).

The paper [141], The study YOLOv4, Optimal Speed and Accuracy of Object Detection conducted extensive experiments across various GPU architectures, revealing that YOLOv4 surpassed other object detection models in both speed and accuracy.

In 2020, Bochkovskiy et al., who also maintain the renowned GitHub repository Darknet, published this paper on arXiv [141]. As shown in Figure 3.18, YOLOv4 operates twice as fast as EfficientDet while delivering comparable performance. Moreover, YOLOv4 improved upon YOLOv3 by enhancing the mean average precision (mAP) by 10% and the frames per second (FPS) by 12%, making it a significant advancement in the field of real-time object detection.

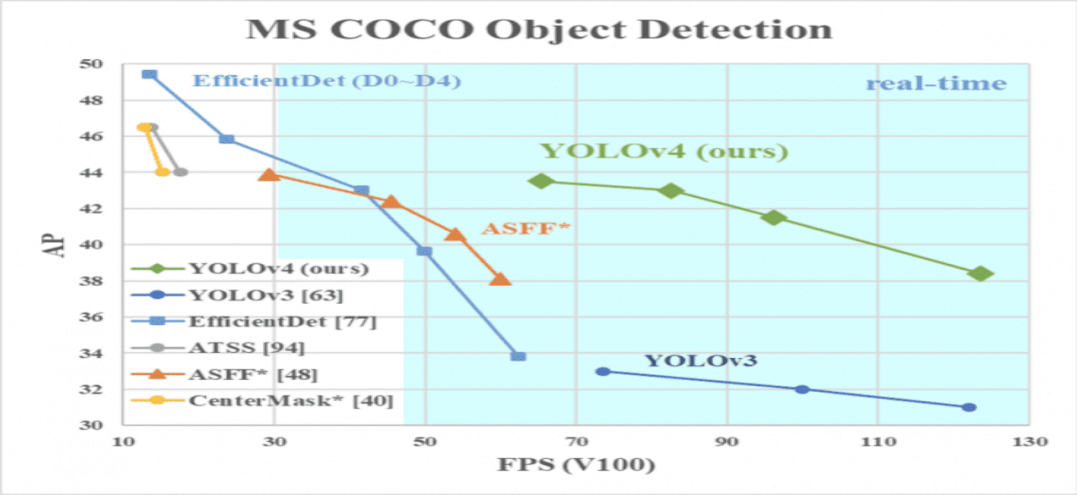

Figure 3.18. Comparison of the proposed YOLOv4 and other state-of-the-art object detectors [141]

The performance of Convolutional Neural Networks (CNNs) is heavily influenced by the features that are utilized and how they are combined. Some features may only work effectively with specific models, problem statements, or datasets. However, certain features, such as batch normalization and residual connections, are applicable across a wide range of models, tasks, and datasets, making them universal.

In their paper, YOLOv4, Optimal Speed and Accuracy of Object Detection, Bochkovskiy et al. leveraged this idea by incorporating several universal features, including:

- Weighted-Residual-Connections (WRC)
- Cross Stage Partial connections (CSP)
- Cross mini-Batch Normalization (CmBN)
- Self-adversarial training (SAT)
- Mish activation
- Mosaic data augmentation
- DropBlock regularization
- CIoU loss

These features were strategically combined to achieve state-of-the-art results, including a 43.5% mAP (65.7% mAP50) on the MS COCO dataset at a real-time speed of approximately 65 FPS on a Tesla V100 GPU.

The YOLOv4 model classified these features into two categories: the "Bag of Freebies," which enhanced the model's training, and the "Bag of Specials," which improved the accuracy of the object detector. This combination of techniques allowed YOLOv4 to set a new benchmark in real-time object detection.

3.6.9. YOLOv5

In 2020, Soon after YOLOv4's release, Glenn Jocher, founder and CEO of Ultralytics, launched an open-source version of YOLOv5 on GitHub [141]. Within two months, YOLOv5 garnered attention for providing a series of object detection architectures pre-trained on the MS COCO dataset, although it was introduced after the release of EfficientDet [134] and YOLOv4, YOLOv5 stood out due to its performance and ease of use. Notably, YOLOv5 is the only YOLO object detector that lacks an accompanying research paper, which initially sparked some controversy. However, this skepticism was quickly dispelled as the model's capabilities and effectiveness demonstrated its value, silencing the initial doubts.

Today, YOLOv5 stands as one of the leading state-of-the-art models, widely supported and highly efficient for production use. One of its key advantages is its native implementation in PyTorch, which removes the limitations of the Darknet framework, which is based on the C programming language and was not originally designed with production environments in mind. While the Darknet framework has evolved over time and remains a valuable research tool—enabling training, fine-tuning, and inference with TensorRT—it has a smaller community and, as a result, offers less support compared to PyTorch. YOLOv5's seamless integration with PyTorch, combined with its ease of use and robust community backing, has made it an ideal choice for production-level object detection tasks.

The transition of YOLO to PyTorch significantly simplified the process for developers to modify the architecture and seamlessly export it to various deployment environments. This ease of customization and deployment has greatly enhanced YOLOv5's adaptability in real-world applications. Furthermore, YOLOv5 is recognized as one of the official state-of-the-art models featured in the Torch Hub showcase, further solidifying its position as a highly valuable and versatile tool for object detection tasks.

3.6.10. PP-YOLO

Thus far, YOLO has been implemented in two primary frameworks, Darknet and PyTorch. However, there is also a third framework where YOLO has been developed, known as the PaddlePaddle framework, leading to the creation of PP-YOLO [142]. PaddlePaddle, developed by Baidu, is a deep learning framework that boasts an extensive repository of models, particularly in the domains of Computer Vision and Natural Language Processing (NLP). This alternative implementation further broadens the scope of YOLO, offering enhanced versatility across different deep learning ecosystems.

An Effective and Efficient Implementation of Object Detector. Like YOLOv4, the PP-YOLO object detector is also built upon the YOLOv3 architecture, incorporating various enhancements to improve both efficiency and accuracy. This development reflects the continuous evolution of object detection models, with PP-YOLO offering an alternative that balances performance and computational cost effectively [127,142].

3.6.11. Paddle Detection

PP-YOLO is integrated into PaddleDetection, an end-to-end object detection development kit built on the PaddlePaddle framework [143]. This toolkit offers a wide range of object detection architectures, backbone networks, data augmentation techniques, and components, such as loss functions and feature pyramid networks. These elements can be configured in various combinations to design and optimize object detection networks, making PaddleDetection a versatile and powerful resource for developing high-performance models tailored to specific tasks.

In summary, PaddleDetection offers a comprehensive suite of image processing capabilities, including object detection, instance segmentation, multi-object tracking, and keypoint detection. These features streamline the process of constructing, training, optimizing, and deploying object detection models, enabling faster and more efficient development. This versatility makes it an invaluable tool for improving the accuracy and performance of object detection systems across a range of applications.

The primary objective of the PP-YOLO paper was not to introduce a novel object detection model but rather to present an object detector that achieves a balanced trade-off between effectiveness and efficiency, making it suitable for direct application in real-world scenarios. This goal aligns with the overarching motivation behind the development of the PaddleDetection toolkit. The novelty of PP-YOLO lies in demonstrating how the combination of various techniques and optimizations leads to improved performance while maintaining computational efficiency. The paper also provides an ablation study to quantify the contribution of each individual step in enhancing the overall performance of the detector.

Similar to YOLOv4, the PP-YOLO paper aimed to combine various existing techniques that enhance the detector's accuracy without increasing the model's parameters or FLOPs, ensuring that the detector's speed remains largely unchanged. However, unlike YOLOv4, this paper did not delve into exploring different backbone networks, such as Darknet-53 or ResNeXt50, nor did it experiment with diverse data augmentation methods. Additionally, the authors did not employ Neural Architecture Search (NAS) to optimize model hyper parameters, focusing instead on refining existing techniques to strike a balance between accuracy and computational efficiency [127,143].

3.6.12. PP-YOLOv2

In 2021, Baidu released the second iteration of PP-YOLO, titled PP-YOLOv2: A Practical Object Detector, authored by Xing Huang and colleagues [144], and published on arXiv. This version achieved significant advancements in the object detection field, setting new benchmarks in both accuracy and efficiency, and further solidifying its position as a highly practical and effective object detection model.

The title of the paper clearly indicates that the primary motivation was to develop an object detector that not only achieves high accuracy but also performs inference at a faster speed, thus making it a practical solution for real-world applications. Since this paper builds on the previous work, PP-YOLO, the authors aimed to create a detector that strikes an optimal balance between effectiveness and efficiency. To achieve this, they employed a similar approach of combining various techniques and optimizations, with a particular emphasis on conducting ablation studies to assess the contribution of each technique to the overall performance.

By incorporating multiple effective refinements, PP-YOLOv2 achieved a substantial improvement in performance, increasing the mean average precision (mAP) from 45.9% to 49.5% on the MS COCO 2017 test set. In addition to its enhanced accuracy, PP-YOLOv2 demonstrated impressive speed, running at 68.9 FPS with a 640×640 image resolution, as illustrated in Figure 3.19. Furthermore, unlike its predecessor PP-YOLOv1, which was released with only the ResNet-50 backbone, PP-YOLOv2 introduced two backbone architectures: ResNet-50 and ResNet-101, offering greater flexibility and potential for further performance optimization [144].

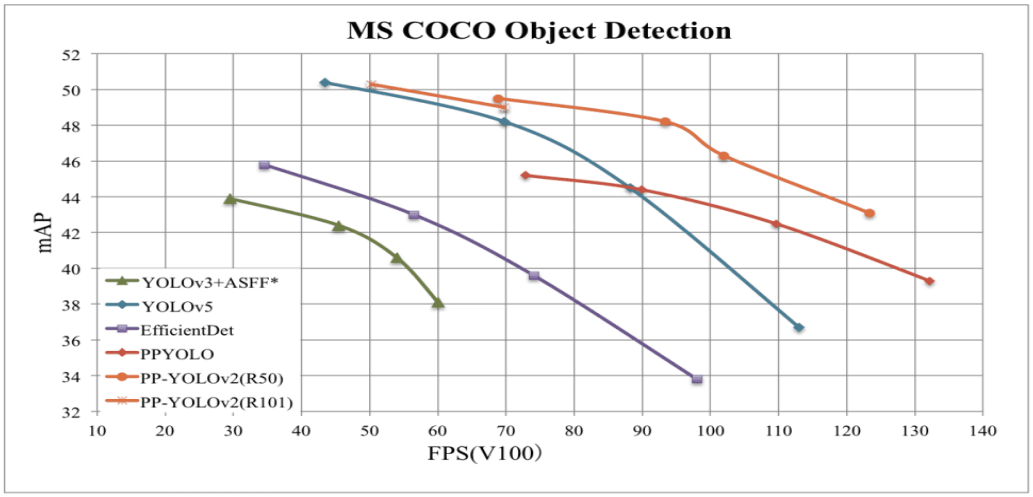

Figure 3.19. Comparison of the proposed PP-YOLOv2 and other object detectors [144]

3.6.13. YOLOX

In 2021, Ge et al. published a technical report titled YOLOX: Exceeding YOLO Series in 2021 on arXiv [145], introducing another advancement in the YOLO family of object detectors. Up until that point, YOLOv1 was the only anchor-free YOLO model, but YOLOX also adopted an anchor-free approach to object detection. In addition to this, YOLOX incorporates several advanced detection techniques, including the use of a decoupled head, robust data augmentation strategies, and the cutting-edge label assignment strategy SimOTA. These advancements allow YOLOX to deliver cutting-edge performance in object detection tasks, further pushing the boundaries of the YOLO series.

3.7. Future of Deep Learning

The future of deep learning is highly promising, as it remains one of the fastest-evolving fields within artificial intelligence (AI) research. Several potential advancements are on the horizon, including improved interpretability of deep learning models, enhanced transfer learning, greater efficiency, the integration of multimodal learning, reinforcement learning, and edge computing. One of the primary challenges facing deep learning models is their lack of interpretability, this opacity frequently makes it challenging to understand how deep learning models reach specific decisions. To address this issue, researchers are increasingly focused on creating more interpretable deep learning models to enhance transparency and build greater trust in these systems. Such efforts are essential for addressing concerns regarding the "black-box" nature of many deep learning models. Another area of potential advancement is enhanced transfer learning. While transfer learning has already shown success in certain domains, ongoing research seeks to improve its

scalability and applicability across a wider range of fields. This progress would allow deep learning models to learn effectively from smaller and more diverse datasets, thereby reducing the reliance on large volumes of labeled data.

Efficiency is another critical area of focus for future advancements in deep learning. Ongoing improvements in both hardware and software, including the development of specialized hardware for deep learning and the creation of more efficient algorithms, will significantly reduce the computational demands of deep learning models. These advancements will make deep learning more accessible and scalable, facilitating broader adoption across various industries and applications. As efficiency improves, the ability to deploy deep learning models on a larger scale, including on resource-constrained devices, will also expand, further enhancing their practical utility in real-world scenarios [116].

Multimodal learning represents another promising avenue for the future of deep learning. This approach enables deep learning models to simultaneously process and learn from multiple data modalities, such as audio, text, and images. By integrating information from diverse sources, multimodal learning has the potential to power more sophisticated and comprehensive AI applications, broadening the scope of AI's capabilities. In addition, reinforcement learning continues to show significant promise, particularly in areas such as robotics and gaming. As research in this domain advances, it is anticipated that more robust and adaptable AI systems will emerge, capable of managing increasingly complex tasks and adjusting to dynamic environments.

Edge computing is an emerging trend that enables computations to be performed directly on devices located at the network’s edge, rather than depending on centralized cloud servers. This approach supports more efficient, low-latency deep learning applications, which is especially advantageous for fields like the Internet of Things (IoT) and autonomous systems, where real-time processing is essential.

In conclusion, Deep learning is a rapidly advancing field with vast potential for the future. Ongoing progress in research and development is expected to lead to the creation of more efficient, interpretable, and adaptable deep learning models, capable of tackling a wider range of real-world challenges and applications [146].

3.8. Convolutional Neural Network

Artificial Intelligence (AI) has made significant strides in narrowing the gap between human capabilities and machine potential. Data enthusiasts and researchers worldwide continue to explore various facets of AI, transforming ambitious visions into reality. One of the most remarkable areas of progress has been in the field of Computer Vision, which seeks to enable machines to perceive and interpret the world in a manner similar to humans. This technology is leveraged for numerous tasks, including image recognition, analysis, and classification, among others. The advancements

in Computer Vision, particularly through the use of Deep Learning, have achieved substantial success, with Convolutional Neural Networks (CNNs) playing a pivotal role in driving these breakthroughs [147].

3.8.1. Introduction to CNN

Yann LeCun, director of Facebook's AI Research Group, is widely recognized as the pioneer of convolutional neural networks (CNNs). In 1988, he introduced the first CNN, known as LeNet, which was specifically designed for character recognition tasks, such as reading handwritten digits and zip codes. This foundational work set the stage for contemporary deep learning applications in computer vision, transforming fields like image recognition and optical character recognition (OCR) [148].

A Convolutional Neural Network (CNN) is a neural network model that integrates essential concepts such as receptive fields, weight sharing, and subsampling. It is a specialized form of multilayer perceptron trained in a supervised manner using backpropagation. CNNs have become one of the most successful machine learning architectures in computer vision, achieving state-of-the-art results in tasks like character recognition, object recognition, face detection, pose estimation, speech recognition, license plate recognition, image preprocessing, and segmentation. CNNs are particularly adept at learning complex, high-dimensional features from extensive datasets, making them ideal for pattern recognition tasks. Their ability to automatically extract relevant features allows them to tackle a diverse range of challenging tasks with high accuracy. CNN architectures also offer a degree of shift, scale, and distortion invariance through features such as local receptive fields, subsampling, and shared weights. Unlike traditional pattern recognition methods, one of the primary advantages of CNNs is their capacity to automatically extract image features during both training and testing [149]. The network constructs progressively complex features by processing a vast set of examples, with feature complexity increasing as additional layers are incorporated. This process involves convolving the input image with filters, thereby building a hierarchical series of feature maps. Consequently, the network learns intricate features while preserving translational and distortion invariance. A CNN can be viewed as a score function, taking raw image pixels as input and producing a category or class score as output. Have you ever wondered how facial recognition functions on social media, how object detection is vital for self-driving car development, or how visual imagery supports disease detection in healthcare. Convolutional neural networks (CNNs) power all of these advancements. The following example shown in Figure 3.20. demonstrates the functionality of CNNs [150].

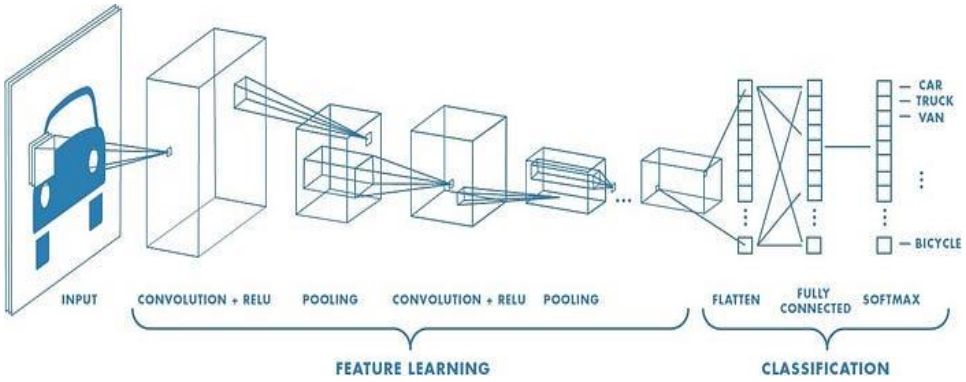

Figure 3.20. Convolutional Neural Network — CNN architecture [150]

Consider an image of a bird, and the task is to determine whether it is indeed a bird or some other object. The first step is to input the pixel values of the image, arranged in arrays, into the neural network's input layer (a multi-layer network used for classification tasks). The hidden layers then perform feature extraction through various calculations and manipulations. These hidden layers include the convolutional layer, the ReLU layer, and the pooling layer, all of which contribute to extracting features from the image. Finally, the fully connected layer processes the extracted features to identify the object in the image [150,151].

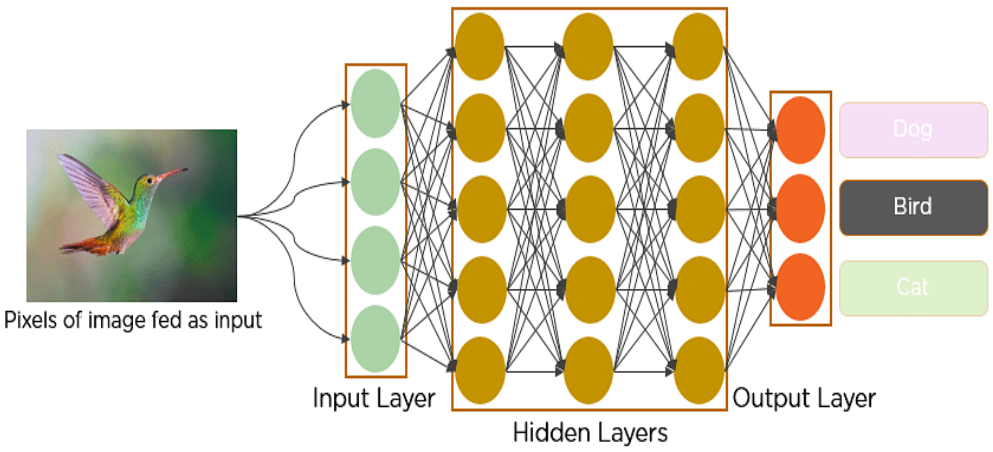

Figure 3.21. Convolutional Neural Network to identify the image of a bird [151]

1- Input Layers: This is the initial layer where data is fed into the model. The number of neurons in this layer corresponds to the total number of features in the dataset, which, in the case of an image, equates to the number of pixels.

2- Hidden Layer: The data from the input layer is passed to one or more hidden layers. The number of hidden layers and neurons within each layer varies based on the model architecture and dataset size. Typically, the hidden layers contain more neurons than the input layer. The output of

each layer is calculated by performing matrix multiplication between the output of the preceding layer and the learnable weights of the current layer, followed by the addition of learnable biases. This result is then processed through an activation function, introducing nonlinearity to the network.

3- Output Layer: The final output from the hidden layers is passed through a logistic function, such as sigmoid or softmax, which converts the results into probability scores for each class [149].

3.8.2. Explain Convolutional Neural Network

A Convolutional Neural Network (CNN) is a feed-forward neural network architecture primarily used for visual image analysis, processing data structured in a grid-like format. Commonly known as a ConvNet, this architecture is highly effective for tasks such as object detection and image classification. CNNs are specifically designed to automatically and adaptively learn spatial feature hierarchies through successive convolutional layers, making them particularly well-suited for image recognition applications

For example in Figure 3.22., a neural network can be trained to identify and classify objects, such as flowers. In this case, the network could be designed to distinguish between two types of flowers: orchids and roses. By feeding labeled images of these flowers into the network, the CNN can learn the distinguishing features of each flower type and accurately classify new images based on those learned patterns [149].

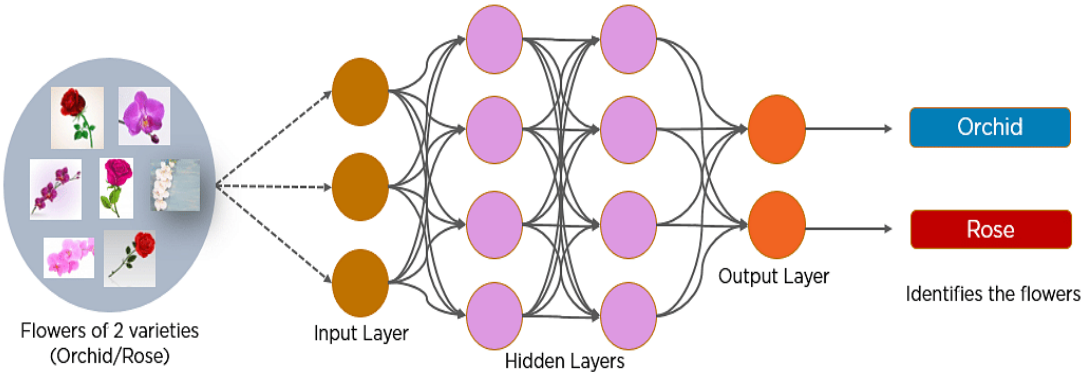

Figure 3.22. Neural network trained to identify and classify types of flowers orchids and roses [151]

In a Convolutional Neural Network (CNN), an image is represented as an array of pixel values. This pixel matrix serves as the foundation for the network to process and interpret visual data, allowing for the detection and classification of distinct features within the image.

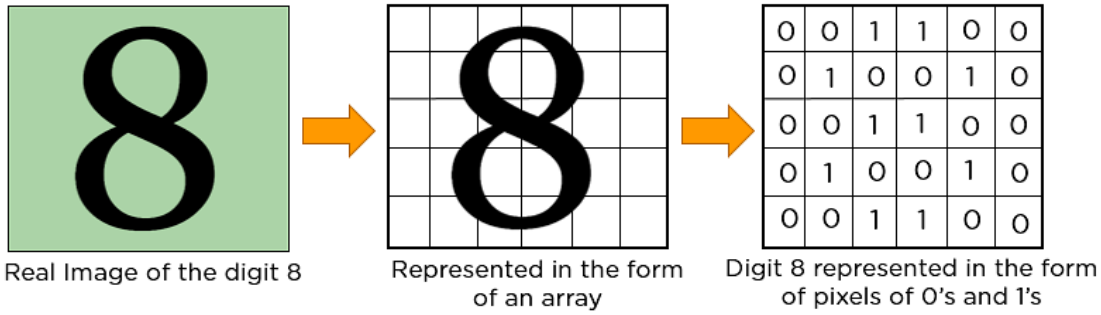

Figure 3.23. Convolutional Neural Network (CNN) image is represented as an array of pixel values [149]

The convolution operation is fundamental to the functioning of any convolutional neural network. To illustrate this operation, consider two one-dimensional matrices, a and b:

a = [5,3,7,5,9,7]

b = [1,2,3]

In a convolution operation, the elements of the arrays are multiplied element-wise, and the resulting products are summed to create a new array, which represents the convolution of a and b (denoted as a*b) shown in Figure 3.24. The operation begins by multiplying the first three elements of matrix a with the elements of matrix b, and the sum of these products forms the first value in the resultant array [150].

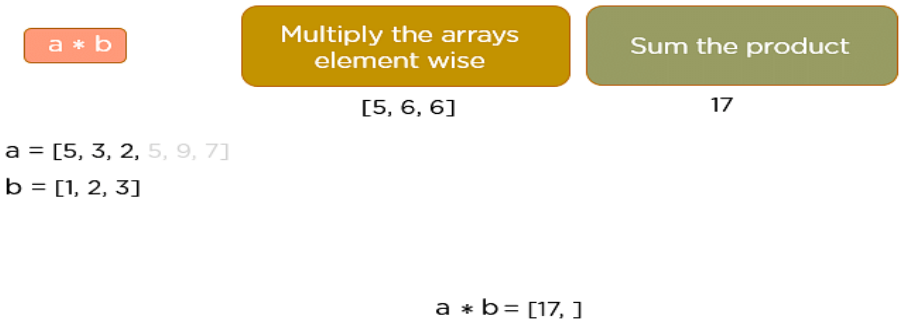

Figure 3.24. Convolutional Neural Network (CNN) of matrix a and b (denoted as a*b) [149]

The next three elements of matrix a are then multiplied by the corresponding elements of matrix b, and the resulting products are summed shown in Figure 3.25.

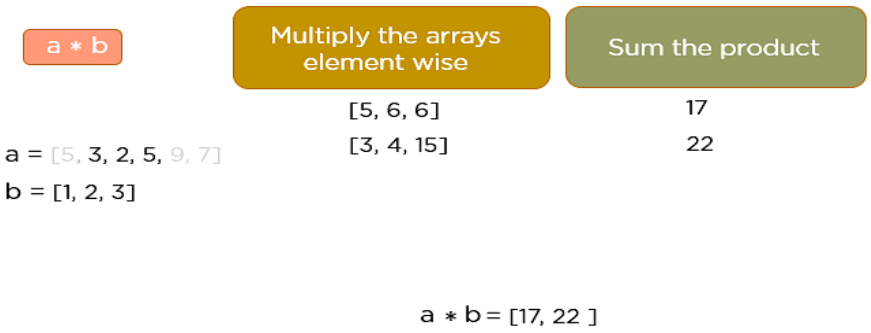

Figure 3.25. Convolutional Neural Network (CNN) of matrix a and b [149]

This process is repeated until the convolution operation is fully completed. At each step, a new set of elements from the input matrix is multiplied by the corresponding elements of the filter matrix, and the resulting products are summed to generate the next value in the output matrix.

3.8.3. CNN Recognize Images

Consider the following images shown in Figure 3.26.

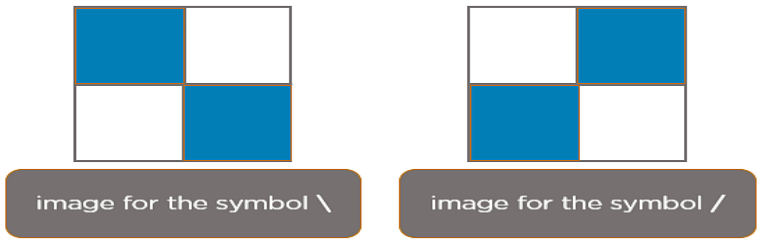

Figure 3.26. CNN Recognize Images [152]

The colored boxes represent a pixel value of 1, while uncolored boxes represent a value of 0. When the backslash (\), is pressed, the image below is processed accordingly shown in Figure 3.27.

Figure 3.27. CNN Recognize Images when you press \ [152]

When the forward slash (/) is pressed, the image below is processed shown in Figure 3.28.

Figure 3.28. CNN Recognize When the forward slash (/) is pressed [152]

Here is another example shown in Figure 3.29. That demonstrates how a Convolutional Neural Network (CNN) processes and recognizes an image. This example highlights the step-by-step process through which CNNs extract features and patterns from the image data to achieve accurate classification or detection.

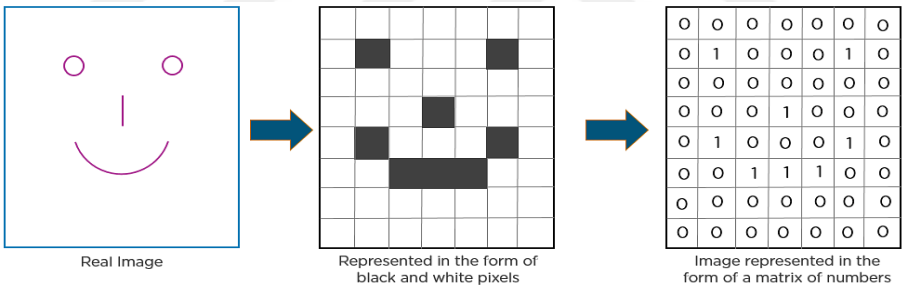

Figure 3.29. Convolutional Neural Network (CNN) processes and recognizes an image

As illustrated in the diagram above, only the values with a value of 1 are highlighted [152].

3.8.4. Convolutional Neural Network Layers

Convolutional Neural Networks (CNNs) are widely used to process 2D images, learning complex, high-dimensional, nonlinear mappings from large datasets. This capability makes CNNs highly effective for various computer vision tasks. The network receives an image as input and applies a 2D kernel with adjustable weights across different regions, a technique known as weight sharing, which reduces the number of free parameters in the model. The results of these convolutions are combined with a trainable scalar, known as a bias, and passed through an activation function, producing a 2D plane referred to as a feature map. Depending on the number of kernels, multiple feature maps are generated, as determined by the network's architecture. Each

feature map is then sent through a subsampling layer to reduce its dimensionality. After subsampling, the maps proceed through a ReLU layer and a local normalization layer, preserving the data's nonlinear characteristics. The number of 2D planes after subsampling, ReLU, and normalization remains consistent with the number of feature maps generated by the convolutional layer. These layers—convolutional, subsampling, ReLU, and normalization—are stacked to form a block, as shown in Figure 3.30. The composition of this block may vary based on the application, with the number of blocks corresponding to the depth of the learning process [149].

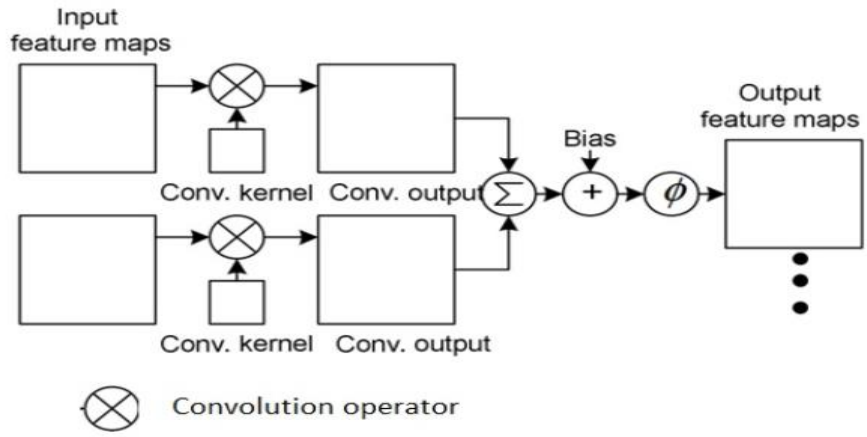

Figure 3.30. Example of Convolutional layer

Layers in a Convolutional Neural Network

A Convolutional Neural Network (CNN) consists of multiple hidden layers that facilitate the extraction of information from an image. The four key layers in a CNN are:

1- Convolution layer

2- ReLU layer

3- Pooling layer

4- Fully connected layer

A- Convolution Layer

As previously discussed, a convolutional layer processes an input image by convolving it with a set of kernels, producing multiple two-dimensional neuron arrays known as feature maps. Each element within a feature map is generated by applying the respective kernel to neighboring units in the preceding layer. The outputs from these convolutions are then combined with a trainable bias term and passed through an activation function to compute each unit within the feature map [151]. While the input image may contain hundreds or thousands of pixels, convolutional layers function as feature extractors, identifying components such as edges, corners, endpoints, or even non-visual features through weight-based kernel convolution [152]. The weight-sharing approach reduces the number of parameters needed for training, thereby decreasing the memory required to

store these parameters during processing [152]. This convolution operation within each layer also grants CNNs translational and distortion invariance, ensuring that any shift in the input image results in a corresponding shift in the feature map. The number of kernels in each convolutional layer, which determines the number of feature maps, depends on the specific architecture of the network. Figure 3.31 illustrates these concepts within a convolutional layer.

The initial phase of extracting meaningful features from an image involves the convolution layer, which comprises multiple filters that carry out the convolution operation. Each image is represented as a matrix of pixel values. For instance, consider a 5x5 image matrix with pixel values of either 0 or 1, accompanied by a 3x3 filter matrix. By sliding the filter across the image and calculating the dot product at each position, a matrix of convolved features is produced [153].

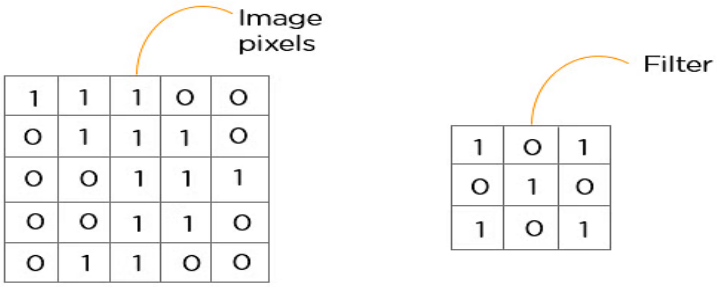

Figure 3.31. Convolution Layer [153]

B- ReLU layer

After the feature maps are extracted, they are passed through a Rectified Linear Unit (ReLU) layer. The ReLU function applies an element-wise operation, converting all negative pixel values to zero, which introduces non-linearity into the network. The output is a rectified feature map. A graph of the ReLU function is shown in Figure 3.32. [153,154].

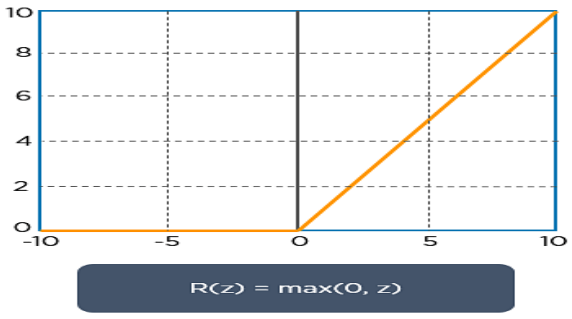

Figure 3.32. Rectified Linear Unit (ReLU) activation function, which is zero when $x < 0$ and then linear with slope 1 when $x > 0$ [154]

The original image is processed(scanned) through multiple convolutions and ReLU layers to detect and locate features is shown in Figure 3.33 and Figure 3.34.

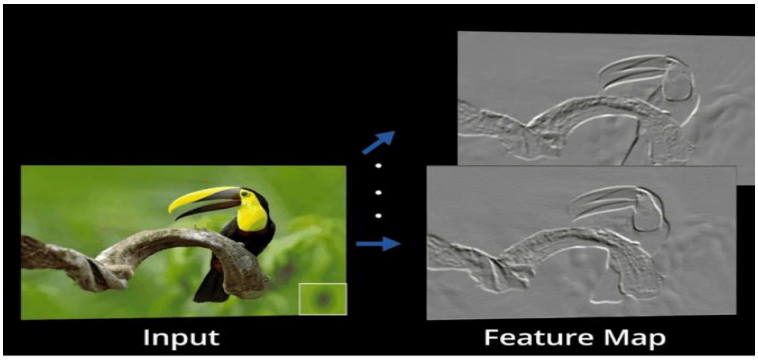

Figure 3.33. ReLU Layer scanned with multiple convolutions [154]

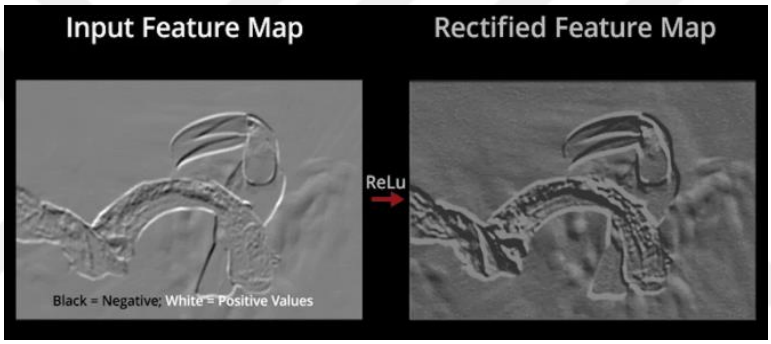

Figure 3.34. ReLU layers for locating the features [154]

C- Pooling Layer

Similar to the convolutional layer, the pooling layer serves to reduce the spatial dimensions of the convolved features. This dimensionality reduction is critical for minimizing the computational resources needed to process the data. Additionally, pooling is beneficial for capturing prominent features that remain consistent despite variations in rotation and position, which aids in the efficient training of the model.

There are two primary types of pooling: Max Pooling and Average Pooling is shown in Figure 3.35. Max Pooling selects and returns the highest value within the region covered by the kernel, while Average Pooling computes and returns the average of all values within that same region.

Max Pooling also acts as a noise suppressor by eliminating noisy activations, effectively performing both de-noising and dimensionality reduction. In contrast, Average Pooling primarily emphasizes dimensionality reduction with a limited impact on noise suppression. As a result, Max Pooling typically outperforms Average Pooling in most applications [152].

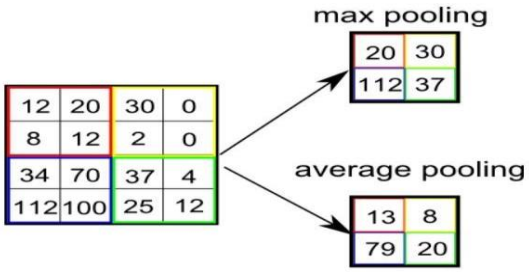

Figure 3.35. Types of Pooling [152]

The convolutional and pooling layers together comprise the i-th layer of a Convolutional Neural Network (CNN). The number of these layers can be increased depending on the complexity of the images being processed, enabling the network to capture finer, low-level details. However, adding layers results in higher computational demands and longer processing times. As the model grows in depth, it becomes adept at recognizing more complex patterns, but this requires increased resources in terms of memory and computational power [152].

Upon completing the preceding steps, the model is equipped to identify and extract essential features from the input data. The subsequent phase involves flattening the final output of the Convolutional Neural Network into a one-dimensional vector, which is then fed into a fully connected neural network for classification. This shift from feature extraction to classification enables the model to assign labels or categories to the input data based on the learned features, facilitating tasks such as object detection and image classification

Pooling is a down-sampling technique that decreases the dimensionality of the feature map, thereby reducing the computational burden. The rectified feature map is subsequently passed through a pooling layer is shown in Figure 3.36, resulting in a pooled feature map. This process preserves crucial features while minimizing the spatial dimensions of the data, allowing for more efficient processing [155].

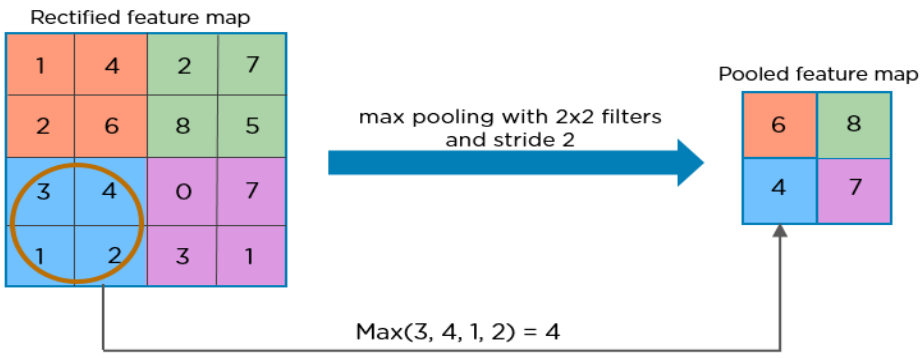

Figure 3.36. Pooling Layer [155]

The pooling layer utilizes various filters to detect distinct parts of the image is shown in Figure 3.37., such as edges, corners, and more specific features like the body, feathers, eyes, and beak. These filters help the model focus on key elements of the image while reducing its spatial dimensions, ensuring that the most important details are preserved for subsequent layers. This process enhances the model's ability to recognize patterns and make accurate classifications while improving computational efficiency.

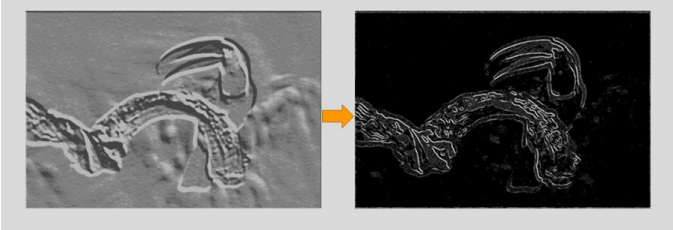

Figure 3.37. The pooling layer uses various filters [155]

The structure of the Convolutional Neural Network (CNN) up to this point is shown in Figure 3.38.

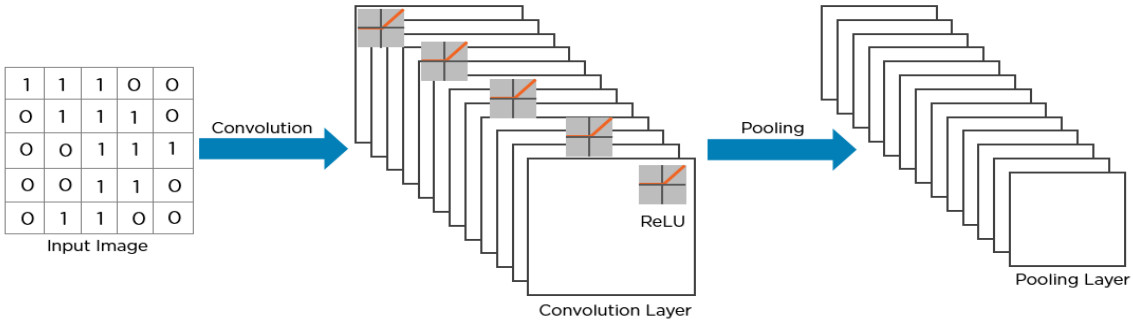

Figure 3.38. Structure of the convolution neural network [155]

The following step in the process is called flattening is shown in Figure 3.39. This operation transforms all the resulting two-dimensional arrays from the pooled feature maps into a single, continuous linear vector.

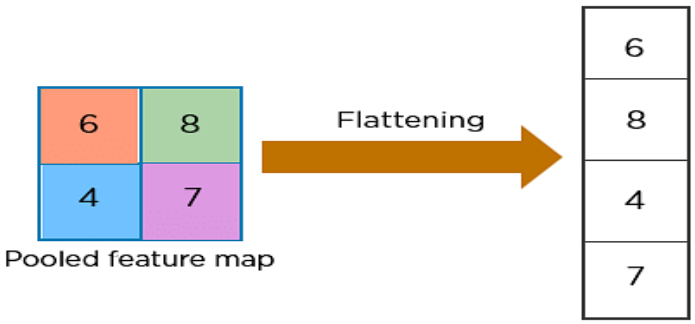

Figure 3.39. Show the flattening process [155]

The flattened matrix is then fed into the fully connected layer, which is responsible for classifying the image is shown in Figure 3.40. [152,155].

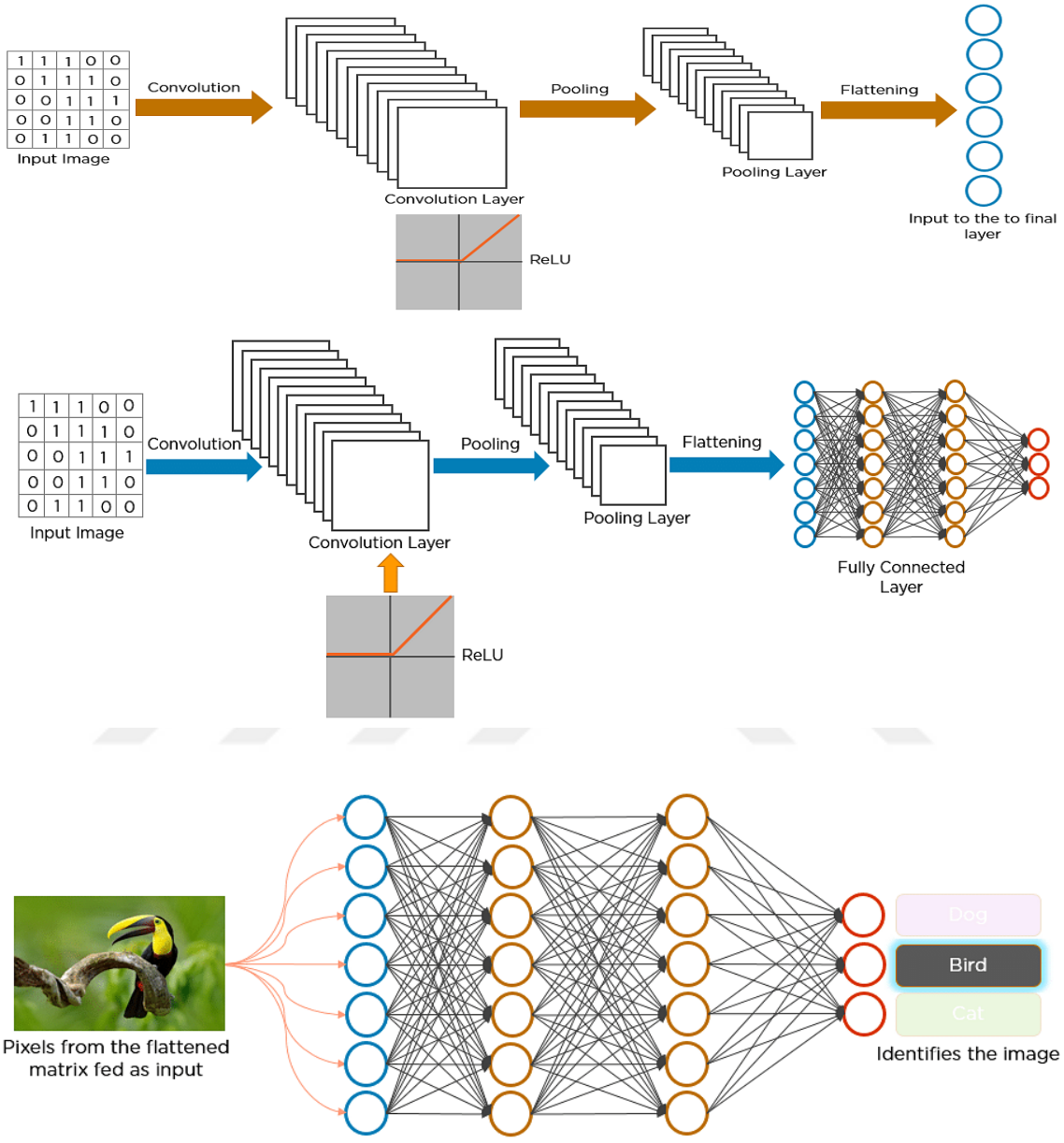

Figure 3.40. Convolutional Neural Network (CNN) architecture [155]

D- Fully Connected Layer

The final layer in a Convolutional Neural Network (CNN) architecture is the fully connected layer. In this layer, each unit is fully connected to every unit from all feature maps of the preceding layer. The fully connected layer functions as a classifier, assigning a confidence score or class label to the input image. These layers handle classification or regression tasks by taking the high-level features extracted in previous layers and mapping them to output class scores [152,156].

The process of CNN recognizing an image, such as a bird, can be summarized as follows is shown in Figure 3.41.

- The image's pixel data is inputted into the convolutional layer, where the convolution operation is applied.
- This produces a convolved feature map.
- The convolved feature map is then processed through a ReLU function, resulting in a rectified feature map.
- The image is subjected to multiple convolution and ReLU layers to detect specific features.
- Various pooling layers, each with different filters, are applied to identify distinct parts of the image.
- Finally, the pooled feature map is flattened and passed through the fully connected layer, generating the final output.

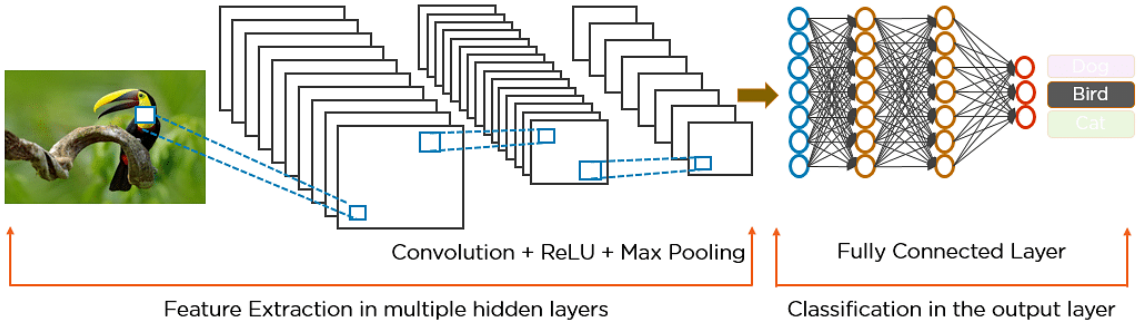

Figure 3.41. Convolutional Neural Network (CNN) architecture is the fully connected layer [156]

3.8.5. Benefits of using CNNs for deep learning

Deep learning, a subset of machine learning, uses neural networks with a minimum of three layers. Compared to single-layer networks, multi-layer networks can achieve higher accuracy. Depending on the application, deep learning incorporates models such as Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs).

CNNs are especially effective for tasks such as image recognition, image classification, and computer vision (CV) applications, as they deliver high accuracy, particularly with large datasets. As data progresses through the multiple layers of a CNN, the network incrementally learns object features in successive iterations, removing the need for manual feature extraction, or feature engineering [152,157].

CNNs can be retrained for new recognition tasks and further developed using pre-existing networks. These capabilities enable CNNs to be effectively applied to real-world tasks without substantially increasing computational complexity or cost

As previously discussed, CNNs are more computationally efficient than traditional neural networks due to their use of parameter sharing. These models are also highly versatile, easy to deploy, and can run on various devices, including smartphones [157].

3.8.6. Applications of convolutional neural networks

Convolutional Neural Networks (CNNs) are already widely employed in various computer vision (CV) and image recognition applications. Unlike basic image recognition, CV allows computing systems to not only identify visual inputs (such as digital images) but also extract meaningful information from them and take appropriate actions based on that information [152,158].

The most prevalent applications of CV and CNNs are typically found in the following fields:

Image Classification: Identifying and classifying objects within images is a fundamental application of Convolutional Neural Networks (CNNs). This includes tasks such as recognizing animals, vehicles, and everyday objects in photos.

Healthcare. CNNs are widely used to analyze thousands of medical images, such as X-rays or MRI scans, to detect anomalies like malignant cancer cells or other abnormal conditions in patients.

Video Analysis: CNNs are applied to video data for tasks like action recognition, tracking objects or individuals in videos, and identifying patterns or anomalies in surveillance footage.

Automotive. CNN technology drives research into autonomous vehicles and self-driving cars, enabling these systems to recognize road signs, pedestrians, and other vehicles in real-time.

Social media. Social media platforms utilize CNNs to automatically identify people in user-uploaded photographs, assisting users in tagging their friends or recognizing individuals.

Retail. E-commerce platforms that use CNNs for visual search can recommend products to customers based on visual similarity, helping brands suggest items likely to appeal to a shopper's preferences.

Gesture Recognition: CNNs are employed to recognize and interpret hand gestures, facilitating human-computer interaction in areas such as virtual reality, gaming, and smart devices.

Facial recognition for law enforcement. CNNs are used to detect and identify individuals based on facial features. They are capable of recognizing faces in images, even under varying lighting conditions and angles, and comparing those faces to stored profiles for identification.

Audio processing for virtual assistants. CNNs used in virtual assistants detect and learn user-spoken keywords, allowing the system to process audio inputs and provide responses or perform actions based on user commands.

Document analysis: CNNs are applied in handwriting recognition, allowing the system to read, interpret, and categorize documents. They are essential for tasks such as categorizing historical manuscripts or interpreting handwritten forms in industries like banking and finance.

Object detection: CNNs are used to identify objects in a variety of images by categorizing them based on their shapes and patterns. Models developed with CNNs can recognize a wide range of objects, from everyday items like food, animals, or people, to specialized objects like firearms and currency. Object detection techniques such as semantic or instance segmentation are used for more precise identification. CNNs are also applied in areas like drones or self-driving cars, enabling them to locate and recognize objects in images and generate different perspectives of those objects.

Auto translation: CNNs are utilized in deep learning for automatic translation between language pairs, such as English and French. Traditionally, translation relied on word-for-word methods or the assistance of multilingual human experts. However, CNNs have significantly improved the accuracy of automatic translation, replacing these older approaches. With CNNs, language pairs like Chinese and English can now be translated with high precision, making the process more efficient and less dependent on human intervention.

3.8.7. Advantages of a convolutional neural network.

The fundamental advantage of CNNs over their predecessors is that they use machine learning to automatically identify key features without the need for human intervention. For instance, when provided with numerous images of cats and dogs, CNNs can independently learn the distinguishing characteristics of each class. In addition to this autonomous feature extraction, CNNs are highly computationally efficient. They utilize parameter sharing, specific convolution, and pooling operations, which reduce the computational load. This efficiency makes CNN models widely adaptable and allows them to run on various devices.

CNNs are particularly effective when working with data that exhibits spatial relationships, making them the preferred approach for prediction tasks involving image input data. Their ability to generate an internal representation of two-dimensional images enables the model to capture information about the location and size of features within different data structures. This capability is essential when processing images, as it allows the model to maintain context while interpreting complex visual data [152,159].

Aim and objectives

The primary goal of this project is to develop a system capable of automatically recognizing Turkish license plates from high-resolution (HR) digital images by integrating advanced computer vision techniques with powerful deep learning algorithms.

The key objectives of this project are as follows:

- To conduct a comprehensive analysis of existing computer vision techniques applied in Automatic Number Plate Recognition (ANPR) and to identify the most effective and suitable approaches. Additionally, the objective is to gain proficiency in utilizing these techniques to their fullest potential.
- To assess various machine learning methods and determine those best suited for ANPR, with a focus on understanding their functionality and adapting them specifically for this application.
- To design and implement a fast, efficient, and reliable ANPR system capable of competing with recently developed systems of similar attributes.
- Beyond the development of the ANPR system, an additional aim is to create a separate application for quantitatively assessing the system's performance.

To create a comprehensive dataset of vehicle images for both training and testing the machine learning algorithms, ensuring a thorough evaluation of the system's performance [152][159].

3.9. MobileNet Deep Learning Techniques

MobileNet represents a suite of advanced deep learning architectures developed and openly shared by Google, tailored specifically for mobile and embedded vision applications in real-life scenarios. These models are meticulously crafted to strike a balance be-tween efficiency and accuracy, making them well-suited for training classifiers on devices with limited computational resources. The core innovation in MobileNets is the implementation of depthwise separable convolutions, which drastically cut down the model's parameter count when contrasted with conventional neural networks, thereby rendering it a more compact and efficient deep learning solution. The essence of depthwise separable convolutions lies in their two-fold approach: initiating with a depthwise convolution that assigns a unique filter to every input channel, followed by a pointwise convolution that applies a 1x1 convolution across every channel output from the depthwise convolution. This methodology not only enhances computational efficiency but also maintains a competitive level of accuracy comparable to traditional convolutional methods. Empirical evidence has demonstrated that MobileNets deliver top-tier performance across a diverse array of tasks such as image classification, object detection, and semantic segmentation. De-spite this high degree of accuracy, what sets MobileNets apart is their unparalleled efficiency, which renders them particularly well-suited for deployment on mobile devices, where computational resources are often at a premium [161].

MobileNetV2 is an advanced neural network architecture optimized for mobile and edge devices. Developed by Google, it improves upon its predecessor, MobileNetV1, enhancing

efficiency and performance for tasks like image classification. It's ideal for use on devices with limited computational power, such as smartphones and IoT devices, making it popular in computer vision applications. Figure 3.42 illustrates the MobileNetV2 architecture, emphasizing its efficient design and high accuracy. Below is a breakdown of its architecture:

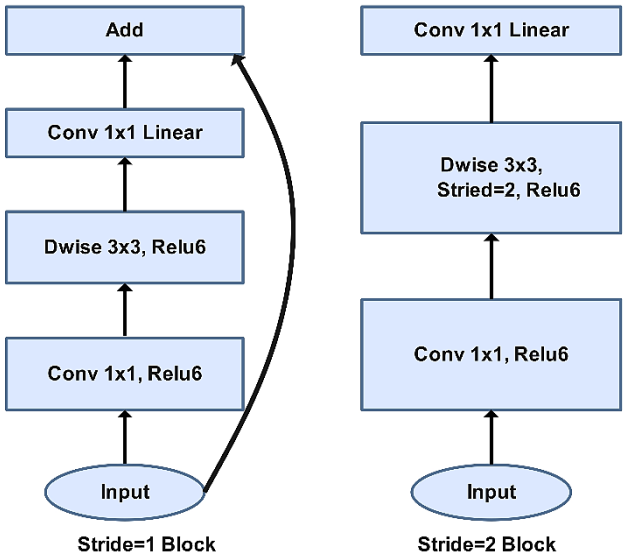

Figure 3.42. MobileNetV2 architecture.

1. Inverted Residual Blocks: These blocks expand, process, and then compress the data to preserve important information efficiently.

2. Linear Bottlenecks: This technique processes data in a compressed format to maintain detail without heavy computation.

3. Depthwise Separable Convolutions: This approach separates the filtering and combining steps of convolution, reducing computational demands.

4. Skip Connections: These helps speed up training and improve information flow by allowing some layers to bypass others.

5. ReLU6 Activation: It limits activation to a maximum of 6, optimizing the network for mobile devices with limited capacity for handling large numerical data.

This architecture allows MobileNetV2 to achieve a balance between latency and ac-curacy, making it highly suitable for real-time applications on devices with limited computational resources.

Here are some of the benefits of using MobileNet:

Efficiency: MobileNets are designed to be lightweight, making them exceptionally suitable for mobile environments where resources are limited.

Accuracy: They achieve leading-edge accuracy across various tasks, showcasing their robustness.

Flexibility: The architecture is adaptable, capable of handling tasks like image classification, object detection, and semantic segmentation with ease.

Here are some of the challenges of using MobileNet:

Complexity: Understanding and implementing MobileNet architectures can be intricate due to their advanced design.

Data requirements: They generally necessitate substantial datasets for effective training.

Computational cost: Despite their efficiency, the training phase of MobileNets can demand significant computational resources.

Limited Resources: On mobile or edge devices, constraints on computational power and memory can reduce the model's performance.

Environmental Variability: The model may struggle with images that vary from its training data, such as different lighting or backgrounds.

Real-Time Processing: While MobileNetV2 is fast, meeting the real-time processing needs of certain applications like video streaming can still be demanding.

Accuracy Limits: For tasks requiring high precision, such as medical imaging, MobileNetV2 might not provide sufficient accuracy.

Model Drift: In dynamic environments, the model's performance can degrade unless it is regularly updated.

MobileNet stands out as a potent solution for a myriad of mobile-based challenges, yet it's crucial to consider its complexities prior to initiating a project involving its implementation.

3.9.1. MobileNet is Actively Employed in Real-World Scenarios Such As

Image Classification: It is utilized for various recognition tasks, from identifying products to aiding in medical diagnoses and enhancing security measures.

Object detection: MobileNets are used to detect objects in images for applications such as self-driving cars and robotics.

Semantic Segmentation: It is applied in segmenting images into distinct objects or areas, useful in medical imaging and autonomous navigation systems.

As MobileNet continues to develop, we can expect to see even more applications for this powerful technology on mobile devices [162-163-164].

3.10. Dataset

3.10.1. COCO Dataset:

We have used coco dataset for notation and labeling vehicle make model(VMM) also used coco dataset as a database to compare actual model with prediction model by our system, coco dataset trained with YOLOx for labeling and coco dataset trained with stanford car dataset to recognize vehicle, please write about this to me.

COCO, which stands for Common Objects in Context, is a large-scale dataset widely used in computer vision tasks. Developed by Microsoft, it offers a variety of annotations for object detection, segmentation, image captioning, keypoint detection, and more. What makes COCO unique is its focus on real-world scenarios, providing images that include objects in natural and context-rich environments. This makes it ideal for training models that need to understand complex scenes.

The dataset includes 80 object categories, such as animals, vehicles, household items, and people. It contains over 330,000 images, with more than 2.5 million labeled instances, and includes annotations for bounding boxes, segmentation masks, keypoints, and captions. For instance, bounding boxes are used to locate objects, while segmentation masks provide pixel-level annotations. Key points help in human pose estimation, and captions describe the overall scene in images. the Figure 3.43 show COCO datasets [165].

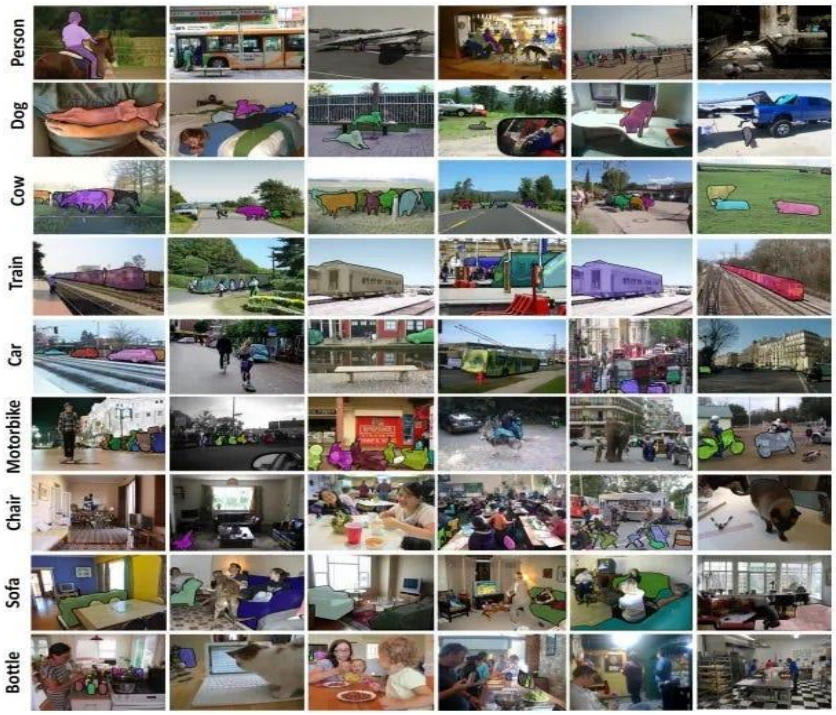

Figure 3.43. Examples of Object Annotations in the COCO Dataset Across Multiple Categories[165]

One of the main strengths of COCO is the quality and diversity of its annotations. Each image is densely annotated, often featuring multiple objects interacting naturally. This richness in data makes it a standard benchmark for training and evaluating computer vision models. The dataset is divided into training, validation, and test sets, with around 118,000 training images, ensuring a robust setup for model development and testing.

COCO has become a cornerstone for a range of applications, including object detection, instance segmentation, human pose estimation, and image captioning. Models like YOLOx, MobileNetv2 have used COCO for training and benchmarking. Additionally, its comprehensive labeling allows researchers to explore tasks like panoptic segmentation, which combines instance and semantic segmentation for a full understanding of an image.

The importance of COCO lies in its ability to represent real-world diversity, making it a critical tool for developing generalizable models. By supporting multiple tasks in one dataset, it enables researchers to tackle complex scene understanding problems effectively. Its wide adoption and role as a benchmark dataset have solidified its position as a key resource in computer vision research [165].

The COCO dataset was used in this study for both annotation and benchmarking in Vehicle Make and Model (VMM) recognition. It served as a tool for labeling vehicles using the YOLOx model and as a database for comparing actual vehicle models with system predictions. YOLOx was employed to generate high-quality annotations, including bounding boxes and segmentation masks, ensuring precise detection data for training.

To enhance recognition capabilities, the COCO dataset was combined with the Stanford Car dataset, which provides fine-grained classification for over 196 car models. This integration enabled the system to effectively recognize specific vehicle makes and models by leveraging COCO's diverse real-world images and Stanford Car's detailed classification data.

The system's performance was evaluated by comparing its predictions with actual vehicle models, ensuring a robust assessment framework. This approach demonstrates COCO's versatility in both annotation and recognition tasks, supported by advanced training methods like YOLOx and the integration of complementary datasets.

3.10.2. Stanford Cars Dataset

The Stanford Cars Dataset is a well-known dataset used for fine-grained classification of vehicles. It was developed by researchers at Stanford University to advance research in tasks such as vehicle make and model recognition. The dataset focuses on distinguishing vehicles at a granular level, providing annotations for specific car models, rather than just general vehicle types, making it ideal for applications that require detailed classification.

The dataset contains a total of 16,185 images of cars, divided into 196 car classes, representing different makes, models, and years. These images are captured from various angles, including side, front, and rear views, to reflect the diversity of real-world scenarios. The dataset is split into 8,144 images for training and 8,041 images for testing, ensuring a balanced distribution for model development and evaluation. Figure 3.44 show example of datasets [166].

Figure 3.44. Sample Images from the Stanford Cars Dataset Representing Various Vehicle Makes, Models [166].

Each image in the dataset is annotated with bounding boxes to locate the car within the image, along with labels specifying the car's make, model, and manufacturing year. For example, a car in the dataset may be labeled as "Tesla Model S 2015" or "Ford Mustang 2014." This level of detail enables researchers to develop fine-grained classification models that can differentiate between visually similar vehicles.

The Stanford Cars Dataset is widely used for training deep learning models, particularly convolutional neural networks (CNNs), for tasks such as vehicle identification and fine-grained recognition. It serves as a benchmark for evaluating model performance in car classification tasks. Due to its detailed annotations and high-quality images, the dataset is also used in applications like traffic monitoring, automatic vehicle identification, and fleet management. the Stanford Cars Dataset is a specialized and comprehensive resource for fine-grained vehicle classification. Its detailed annotations and diverse image set make it an essential dataset for advancing research in vehicle recognition and related computer vision applications.

4. CAR PLATE DETECTION AND CLASSIFY MODEL

This research introduces a novel system that leverages deep learning architectures for simultaneous vehicle and license plate detection, a pioneering approach in vehicular technology. By integrating vehicle make and model detection with Automatic Number Plate Recognition (ANPR), this system offers a more holistic and efficient solution for vehicle identification, especially in real-time applications like traffic surveillance and automated toll collection.

The objective of this research is to develop a real-time system capable of detecting and recognizing vehicles, including their make, model, and license plate number. To achieve this, we leverage state-of-the-art deep learning architectures, including Convolutional Neural Networks (CNNs), to process images and video feeds. The system was rigorously tested on a dataset of 1,000 real-time images captured under various environmental conditions, such as rain, snow, and low light. The system achieved a remarkable classification accuracy of 97.5%, demonstrating its robustness and potential for real-world applications (correctly identifying 975 images and misclassifying 25).

4.1. The Car Dataset

The Firat University Image Dataset was developed and used to evaluate a real-time car model and plate detection system utilizing advanced deep learning architectures. Firat University, located in Elazig, Turkey, served as the site for data collection. Images were captured at the university's three entrance gates under various challenging conditions, including rain, snow, fog, mist, and sunny weather. This diversity ensured that the dataset tested the robustness of the proposed system across real-world scenarios.

In this dataset, a total of 1,000 images were taken from the entrance gates. These images, captured with visible license plates and vehicles under different lighting and weather conditions, were used specifically to test the accuracy and performance of the system. The uploaded sample image demonstrates examples of vehicles entering the campus, with variations in brightness, shadows, and headlight interference, reflecting real-world complexities.

Figure 4.1. Sample Images from Firat University Dataset Captured for Real-Time Vehicle Make, Model, and License Plate Detection

The Figure 4.1. showcases several samples from the Firat University dataset, depicting vehicles entering and exiting the campus. These images highlight the diversity of vehicle makes, models, and plate configurations encountered in the dataset. The data captured represents a broad range of scenarios, making it suitable for comprehensive testing and evaluation of automated systems. The goal of this dataset was to create a realistic benchmark for analyzing the performance of vehicle detection and recognition technologies.

The dataset was employed as test data, while the COCO Dataset and Stanford Cars Dataset were used as the main training databases. The system was designed to detect vehicle make and model while performing Automatic Number Plate Recognition (ANPR). Several state-of-the-art deep learning models, including MobileNet-V2, YOLOx, YOLOv4-tiny, Paddle OCR, and SVTR-tiny, were utilized for this purpose. Each model contributed to vehicle detection and fine-grained classification of car makes and models, while Paddle OCR and SVTR-tiny focused on license plate text recognition.

For evaluation, the Grad-CAM technique was used to analyze and visualize misclassification cases, helping identify errors in the system's predictions. This approach provided insights into why certain predictions were incorrect, offering a pathway for further system improvement. The Firat University dataset proved invaluable in testing the system's ability to operate effectively under diverse and challenging conditions, making it a critical component of the project.

4.2. System setup

The experiments were conducted on a high-performance computing system equipped with an Intel Core i7 processor, 16 GB of RAM, and an NVIDIA GeForce RTX 2080 Ti GPU. The system utilized Python 3.8 as the programming language, with PyCharm as the integrated

development environment (IDE). TensorFlow served as the deep learning framework, while OpenCV, NumPy, Pandas, and Matplotlib were employed for image processing, numerical computations, data analysis, and visualization, respectively. The datasets used in this research included a diverse collection of vehicle images captured under various environmental conditions, such as rain, sunshine, and snow. The datasets comprised the COCO dataset for vehicle make and model labeling, the Stanford Cars dataset for vehicle make and model classification, and a custom dataset collected from Firat University for real-time system testing.

Figure 4.2. the structure of a study for a real-time car model and plate detection system using deep learning architectures

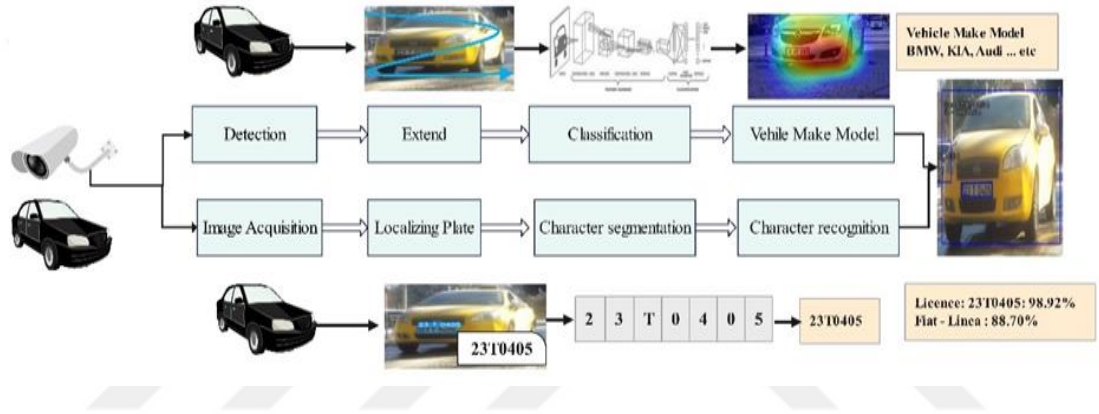

Figure 4.2. Integrated Process Flow for Automated Vehicle Identification and License Plate Recognition

The illustrated workflow Figure 4.2. depicts a sequential process, typical in computer vision tasks. It involves the following steps: image acquisition, vehicle detection, region extension, license plate localization, character segmentation, and character recognition. The figure also includes visual representations of these steps, such as a vehicle being captured by a camera, highlighted license plate regions, and segmented characters.

Heatmaps and overlays on vehicles suggest the use of techniques like Grad-CAM to visually explain the model's predictions. The side panel showcases an example of the system's output, displaying a recognized vehicle with a bounding box around the license plate. It provides the vehicle make and model (e.g., BMW, KIA, Audi), the license plate number with a high confidence score (e.g., "License: 23T0405: 98.92%"), and the make and model confidence score (e.g., "Fiat - Linea: 88.70%"). This structured approach indicates a sophisticated system that employs a combination of deep learning methods to achieve real-time detection and recognition of vehicles and their license plates. This structured approach is indicative of a sophisticated system that uses a combination of deep learning methods to achieve real-time detection and recognition of vehicles and their number plates.

The process begins by acquiring images or videos containing vehicles. Once a vehicle is detected within a frame, the system proceeds to identify its make and model. This task is accomplished using the MobileNet-V2 Convolutional Neural Network, a model renowned for its accuracy in image classification. Additionally, the YOLOx algorithm, trained on the extensive COCO dataset, is employed to further enhance the system's ability to recognize a wide range of vehicle makes and models.

The MobileNet-V2 model, pre-trained on the Stanford Cars Dataset, can accurately classify a wide range of vehicle makes and models, including BMW, Audi, KIA, Sedan, Volkswagen, Passat, and others. This model's high accuracy of 97.5% was demonstrated on a dataset of 1,000 real-time images captured under various challenging conditions, such as low-quality images, adverse weather, and suboptimal camera angles. This highlights the system's robustness and potential for real-world applications, including stolen vehicle tracking, traffic monitoring, and security surveillance.

4.3. Experiments

ANPR system was rigorously tested across a variety of environmental conditions to evaluate its robustness, particularly its sensitivity and accuracy during adverse weather situations and when analyzing low-resolution images from real-time vehicular university campus or traffic. For the detection of vehicles and their license plates, the system incorporates the Yolov4-tiny model, which is built upon a streamlined darknet architecture. This model is distinguished by its balance of efficiency and rapid processing capabilities, achieved through a simplified network structure that reduces the number of layers and computational requirements, yet manages to preserve a high level of accuracy essential for license plate detection. The YOLOv4-tiny, a more compact variant of the YOLOv4 architecture, distinguished by its reduced complexity and a more efficient backbone and fewer convolutional layers, thereby ensuring expedited performance without significantly impacting the accuracy of detection. For the recognition of license plate numbers, the system employs Paddle OCR in conjunction with YOLOv4 to achieve precise character segmentation. Paddle OCR to extract text from images and videos. It can handle over 80 languages, such as Chinese, Arabic, French, English, and Cyrillic, including both letters and numerals, making it particularly suitable for reading license plates. To further augment the recognition capabilities, SVTR-tiny model, which has been trained on a diverse dataset of license plates, is utilized, enhancing the system's proficiency in plate recognition tasks. The strategic integration of these advanced models culminates in achieving peak accuracy levels for the OCR component of the system, showcasing its effectiveness in accurately identifying license plate numbers under a wide array of operational scenarios. Figure 4.3. and Table 4.1 illustrate the final output of the system, showcasing a vehicle with a detected license plate and identified make and model. The license plate

number "01AG[redacted]" is recognized with a high confidence score of 96.24%, while the vehicle is identified as a " Volkswagen-Golf" with a confidence score of 96.23%. This demonstrates the system's ability to accurately detect and recognize vehicles in real-time.

Table 4.1. Model Predictions for a 'Volkswagen-Golf' vehicle

Actual Model	Make	Model	SoftMax prob	Prediction model
Volkswagen, Golf	Volkswagen	Golf	0.96236265	Volkswagen, Golf
		Golf Plus	0.014997232	
		Polo	0.0078093642	

Figure 4.3. Real-Time Automated License Plate and Vehicle Model Detection System Realtime

The image above demonstrates a real-world application of the system, showing a Volkswagen Golf captured in real-time at the entrance of Firat University. The system accurately detects the vehicle and its license plate, "01 AG[redacted]" with a high confidence score of 96.24%. This highlights the system's ability to[redacted] real-time vehicle identification under various conditions.

Grad-CAM, a technique used to visualize the decision-making process of CNNs, was employed to analyze the model's performance. By highlighting the most influential regions of the image, Grad-CAM helps identify areas where the model may be struggling, particularly in cases of misclassification. Grad-CAM was employed to analyze cases of incorrect classifications, as shown in Figure 4.4. This technique helps visualize the model's decision-making process and identify potential areas for improvement.

Figure 4.4. Grad-CAM technique Analysis for Vehicle Detection and License Plate Recognition Realtime

Gradient-weighted Class Activation Mapping (Grad-CAM) is a technique used to visualize the regions of an image that are most important to a deep learning model's decision-making process. In the case of vehicle classification, Grad-CAM can highlight areas of the vehicle, such as the shape, color, or specific features, that the model focuses on to determine the vehicle type or license plate information. This visualization is particularly useful for identifying potential biases or errors in the model's predictions. The color coding in the heatmap overlay typically represents the following:

- **Blue areas:** Regions with minimal influence on the model's decision.
- **Red areas:** Significant features that the model heavily focuses on.
- **Yellow areas:** Regions of intermediate importance.

By analyzing these color-coded heatmaps, we can gain valuable insights into the neural network's decision-making process and identify the key features that contribute to accurate vehicle recognition and classification.

The system has been rigorously tested under various conditions, including adverse weather and low-light scenarios. The results demonstrate the system's ability to accurately identify vehicle makes, models, and license plates. Figure 4.5. and Table 4.2 illustrate the system's performance under various conditions.

Table 4.2. Model Predictions for ‘Opel Insignia’ vehicle under normal condition

Actual Model	Make	Model	SoftMax prob	Prediction model
Opel, Insignia	Opel	Insignia	0.9921659	Opel, Insignia
		Astra	0.005185814	
		Meriva	0.00067606586	

Figure 4.5. Result of VMMR and detect ANPR Realtime

Table 4.2 provides a detailed breakdown of the model's predictions for a specific image. It shows that the model correctly predicted the vehicle make and model as "Opel Insignia" with high confidence, while also providing probabilities for other less likely options. This demonstrates the system's ability to not only identify the most probable class but also provide a measure of uncertainty in its predictions.

The image of Figure 4.5. showcases a real-world application of the system, demonstrating its ability to accurately detect and recognize vehicles in real-time conditions. The system successfully identifies the vehicle as an "Opel Insignia" with a high confidence score of 99.22%. Additionally, the license plate "06 AV [redacted]" is recognized with a confidence score of 97.57%. This highlights the system's effectiveness in both vehicle make and model identification, as well as license plate recognition.

Figure 4.6. illustrates the application of Grad-CAM to a real-world image. The heatmap overlay highlights the regions of the image that were most influential in the model's decision to classify the vehicle as an "Opel Insignia" and recognize the license plate "06 AV [redacted]." The red and yellow areas indicate the regions that the model focused on, such as the car's logo, shape, and license plate. This visualization provides valuable insights into the model's decision-making process and helps identify potential areas for improvement.

Figure 4.6. Apply Grad-Cam for analysis Real time

Figure 4.7. showcases the system's robustness under challenging lighting conditions, such as direct sunlight. Despite the lens flare and overexposure, the system accurately recognized the license plate "06 BIL [redacted]" with a high confidence score of 96.70%. However, the model fails to identify the vehicle as a Ford Focus as Table 4.3 demonstrates this issue.

Table 4.3 provides a breakdown of the model's predictions for a specific image. The actual model is a Ford Focus, and the system correctly identifies the make as "Ford." However, it incorrectly predicts the model as a "Kuga" with a higher confidence score compared to the correct "Focus" prediction. This highlights a potential limitation of the model, as it may struggle with distinguishing between similar models, especially when presented with limited visual information or challenging lighting conditions. It's important to note that while the system's overall performance is impressive, there are still areas for improvement, such as refining the model's ability to differentiate between similar vehicle models.

Table 4.3. Model Predictions for 'Ford Focus' vehicle under intensive sunlight condition

Actual Model	Make	Model	SoftMax prob	Prediction model
Ford, Focus	Ford	Focus	0.13321988	Ford, Focus
		Kuga	0.12182534	
		Civic	0.037777178	

Figure 4.7. Vision of Progress Harnessing Deep Learning for Precision in Vehicle Make and Model Recognition and Enhanced Automatic Number Plate Detection

Figure 4.8 illustrates a challenging scenario where the vehicle is partially obscured by a tree. The Grad-CAM have been utilized Despite the occlusion; the system successfully recognizes the license plate "06 BIL [redacted]" with a high confidence score. However, the model struggles to accurately identify the vehicle make and model due to the limited visibility. This highlights the importance of considering various environmental factors and potential occlusions when designing and training computer vision models.

Figure 4.8. Vision of Progress Harnessing Deep Learning for Precision in Vehicle Make and Model Recognition and Enhanced Automatic Number Plate Detection

Figure 4.9. demonstrates the system's ability to function in challenging low-light and rainy conditions. Despite the adverse weather, the system accurately recognized the license plate "23 BK [redacted]" with a high confidence score of 93.95%. Additionally, the model identified the vehicle as a

Volkswagen Polo with a confidence score of 81.73%. This highlights the system's robustness in handling various lighting conditions.

Table 4.4. Model Predictions for a ‘Volkswagen Polo’ vehicle under rainy condition

Actual Model	Make	Model	SoftMax prob	Prediction model
Volkswagen, Polo	Volkswagen	Polo	0.21734688	Volkswagen, Polo
		Golf	0.055434152	
		Amarok	0.046584003	

Figure 4.9. Low-Light License Plate Recognition and car model in Evening Realtime

Table 4.4. Model Predictions for a ‘Volkswagen Polo’ vehicle under rainy condition provides a breakdown of the model's predictions for a specific image. The actual vehicle is a Volkswagen Polo, and the model correctly identifies the make as "Volkswagen." The model predicts "Polo" as the most likely model (22%), while it assigns 5% and 4% for ‘Golf’ and ‘Amork’ models respectively.

Figure 4.10. illustrates the application of Grad-CAM to a vehicle image captured under challenging conditions, including rain and low light. The heatmap overlay highlights the regions of the image that were most influential in the model's decision to identify the vehicle's make and model and read the license plate. Despite the adverse weather conditions, the model successfully recognized the license plate "23 BK [redacted]."

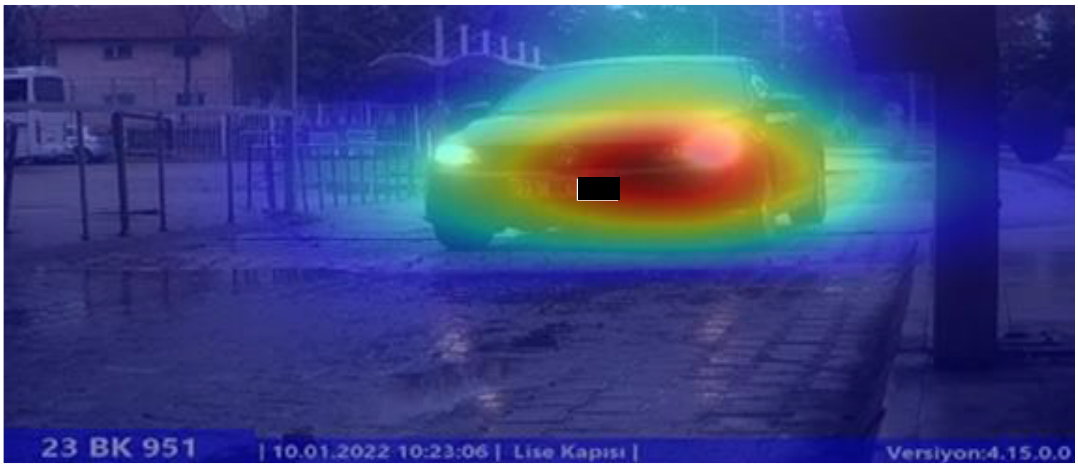

Figure 4.10. Cloudy with a Rainy day of Data Precise Vehicle Analytics

Figure 4.11. demonstrates the system's ability to function in challenging low-light conditions. Despite the darkness, the system accurately recognized the license plate "23 ABA [redacted]" with a high confidence score of 99.74%. While the model identified the vehicle as a Renault Fluence, the confidence score of 47% indicates potential uncertainty in this classification. This highlights the importance of considering lighting conditions when evaluating the performance of computer vision models.

Table 4.5 shows the model's prediction results for ‘Renault, Fluence’ vehicle under low-light conditions. The actual vehicle is a Renault Fluence, and the model correctly identifies the make as "Renault." However, the model predicts "Fluence" as the most likely model, even though the confidence score is relatively low (0.47). This indicates that the model may struggle to differentiate between similar models, especially in challenging conditions.

Table 4.5. Model Predictions for a ‘Renault, Fluence’ vehicle under low-light conditions

Actual Model	Make	Model	SoftMax prob	Prediction model
Renault, Fluence	Renault	Fluence	0.47003028	Renault, Fluence
		Sport	0.05977053	
		Clio	0.043012273	

Figure 4.11. Nighttime vehicle recognition result Realtime

Another example demonstrates the system's performance on a Mercedes-Benz vehicle captured from a front-side angle. While the model successfully detected the license plate "80 ABD [redacted]" with a high confidence score of 94.50%, it struggled to accurately identify the vehicle model due to the camera angle, which may have obscured key features. This highlights the importance of camera placement and image quality in achieving accurate vehicle identification.

Figure 4.12. demonstrates the application of Grad-CAM to visualize the model's decision-making process under challenging weather conditions, such as rain and cloudy skies. The heatmap highlights the regions of the image that were most influential in the model's prediction. This visualization helps to understand how the model handles adverse weather conditions and identifies potential areas for improvement.

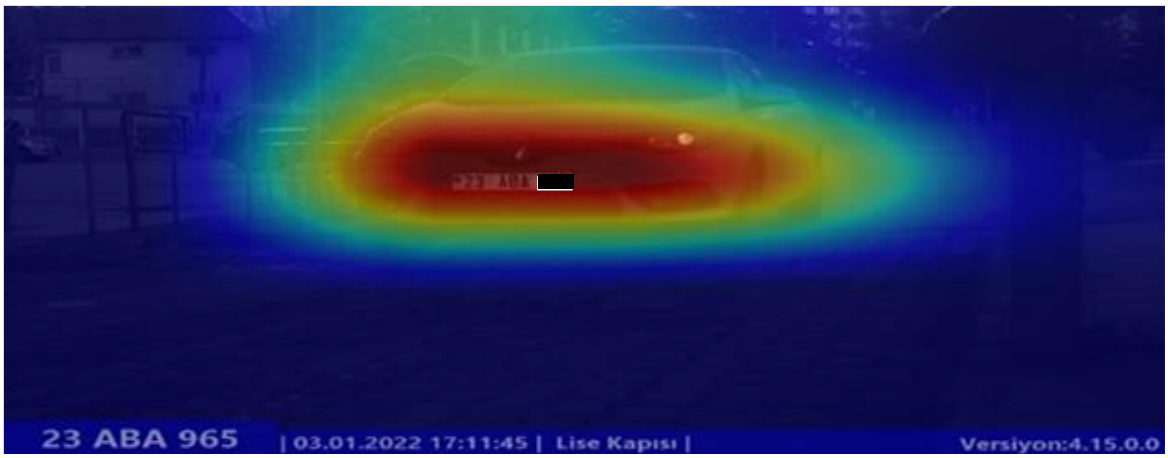

Figure 4.12. Midnight Trace: GradCam's Insight on Vehicles and Plates

Figure 4.13. illustrates a scenario where the camera angle hinders the model's ability to accurately identify the vehicle model. While the system successfully detects the license plate "80 ABD [redacted]" with a high confidence score, the model struggles to identify the vehicle model. This highlights the importance of camera placement and image quality in achieving accurate vehicle identification.

Figure 4.13. Result due to angle cam angle issue

Figure 4.14. shows the application of Grad-CAM to visualize the model's decision-making process for a vehicle captured from a challenging angle. The heatmap highlights the regions of the image that were most influential in the model's decision to recognize the license plate "80 ABD [redacted]." While the model successfully detected the license plate, the limited view of the vehicle hindered accurate vehicle make and model identification. This highlights the importance of camera placement and image quality in achieving optimal performance.

Figure 4.14. Intricacies of Camera Angles in Enhancing License Plate Recognition While Obscuring Vehicle Make and Model Identification

To evaluate the performance of our proposed system, we conducted a comparative analysis with existing state-of-the-art methods for vehicle make and model recognition and ANPR. Deep learning architectures, such as YOLOv4-tiny and MobileNet-v2, were chosen for their high performance and efficiency in real-time processing.

Our integrated system offers several advantages over traditional methods that focus solely on either vehicle make and model recognition or license plate recognition:

- **Efficiency:** By combining both tasks into a single process, our system reduces computational overhead and improves real-time performance.
- **Accuracy:** Our system achieves high accuracy in both vehicle detection and license plate recognition, even under challenging conditions.
- **Robustness:** The system is robust to variations in lighting conditions, weather, and camera angles.

We compared our system to existing methods using metrics such as accuracy, processing time, and ability to operate under various environmental conditions. The results demonstrate the superior performance of our proposed approach. The models were pre-trained on extensive datasets, including COCO and Stanford Cars, to learn a wide range of vehicle features. Additionally, real-world images from the entrance gate of Firat University were incorporated into the training and testing process to further enhance the models' robustness. Extensive hyperparameter tuning was performed to optimize the models' performance. The resulting system demonstrated a significant improvement in accuracy compared to single-task systems, particularly under adverse weather conditions. Furthermore, the system's reduced processing time makes it suitable for real-time applications.

Figure 4.15. presents Grad-CAM visualizations for several misclassified images. While no clear pattern emerges, a common trend is observed in images captured during evening hours. These images often exhibit challenges in lighting conditions, potentially affecting the model's ability to accurately identify vehicle features. Additionally, images that do not fully capture the entire vehicle may also contribute to misclassifications.

To address these limitations, future research could focus on improving the model's performance in low-light conditions and developing techniques to handle partially occluded vehicles. A potential approach could involve using a voting system that considers multiple frames to increase classification accuracy.

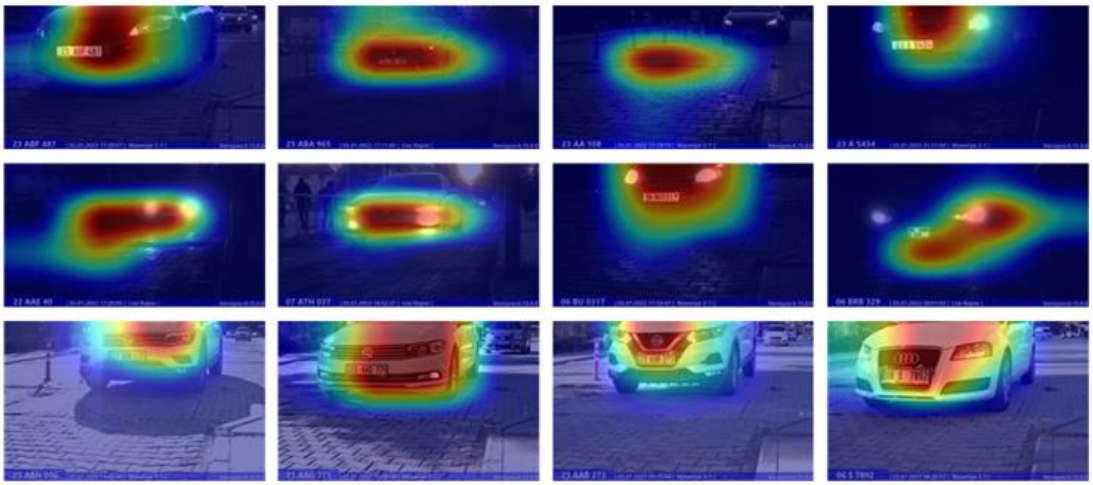

Figure 4.15. Grad-Cam outputs of some images that were misclassified (FP and FN).

Table 4.6. Below show Comparison between our study and other study were done

Ref	Methodology	Dataset	Accuracy
[47]	CNNs - five convolutional layers	ILSVRC 2012	91.60%
[49]	Bag of SURF Features, CNNs, Multi-class SVM.	NTOU-MMR Dataset	94.80%
[50]	NN Classification with Feature Extraction	Parking Lot Dataset	93.00%
[51]	Unsupervised Sub-categorization with Iterative Split-and-Merge	USC Multi-View Car Dataset	93.00%
[53]	DAVE Framework (Fast Vehicle Proposal Network (FVPN) and Attributes Learning Network (ALN))	CompCars Dataset and UTS Vehicle Dataset	97.10%
[54]	AlexNet	Custom vehicle dataset	97.30%
[56]	Nine-Layer CNN	Vehicle Dataset (90,000 images)	91.22%
[57]	Residual SqueezeNet Architecture	Custom Korean Vehicle Dataset	96.30%
[59]	AlexNet	ImageNet (ILSVRC 2010 and 2012 subsets).	97.00%
[61]	CNN (5 Convolutional Layers and 2 Fully Connected Layers with Batch Normalization).	CompCars Dataset.	96.70%
[62]	Overfeat CNN	CompCar dataset	92.70%
[63]	ConvNet	ImageNet	97.20%
[64]	GoogLeNet	ILSVRC 2014	97.30%
[66]	CNN- SSD (Single Shot MultiBox Detector)	Stanford Car Dataset	88.00%
[67]	CNN based on extracted features	Custom vehicle dataset	97.00%
[68]	CNN - 10-layer architecture	Standard Dataset	97.00%
[69]	Douglas Peucker Algorithm	Custom vehicle dataset	95.50%
[77]	Sobel edge and Laplacian edge	Custom vehicle dataset (Malaysian)	95.80%
[78]	YOLOv3, C-CNN-L3, and R-CNN-L3	Heping River Dredging Operation Dataset	95.00%
[81]	37-class (CNN), YOLOv3	Indian car number plates	91.00%
[82]	K-Nearest Neighbors (KNN)	Custom vehicle dataset (Indonesian vehicle license plates)	97.00%
[84]	Normalized Cross Correlation and Phase Correlation using Template Matching	Custom vehicle dataset (Nepali vehicle number plates)	67.98%
[85]	Feature Extraction Model and Backpropagation Neural Network (BPNN)	Custom vehicle dataset (Chinese license plates)	97.10%
[86]	CNN- 2D Correlation for Character Recognition	Custom vehicle dataset (Thai license plate images)	93.42%
[87]	Faster R-CNN	Kaggle's Vehicle dataset	95.00%
[88]	PIC microcontroller	Custom vehicle dataset (Inda)	96.00%
[89]	YOLOv4, Resnet50	Bangladeshi vehicles LP	96.31%
	Proposedmethod-MobileNet-V2, YOLOv4-tiny, Paddle OCR, SVTR-tiny	COCO, Stanford Car, Firat-University datasets	97.50%

Table 4.6 offers a comprehensive comparison of various methodologies, datasets, and their corresponding accuracies for vehicle-make model tasks such as detection, classification, and Automatic Number Plate Recognition (ANPR). It includes references to prior studies that utilized models like Convolutional Neural Networks (CNNs), AlexNet, YOLOv3, and others on datasets ranging from custom vehicle datasets to publicly available datasets like the COCO and CompCar datasets. The accuracy values vary, highlighting the effectiveness of different approaches and datasets in achieving high performance.

Table 4.6 is the dominance of deep learning methods, particularly CNN-based architectures, in achieving state-of-the-art results. These models have demonstrated superior performance compared to traditional methods, such as those based on feature extraction and classification. The choice of dataset also plays a crucial role in model performance. Large and diverse datasets, like COCO and CompCar, provide a rich source of training data, enabling models to learn robust features and improve generalization.

Another important observation is the wide range of accuracies reported in the table, ranging from 67.98% to 97.5%. This variation can be attributed to several factors, including the complexity of the task, the quality of the dataset, the model architecture, and the training techniques employed.

Table 4.6 highlights the significant role of deep learning models in achieving high accuracy in vehicle-make model recognition and ANPR tasks. Models like AlexNet, GoogLeNet, and Overfeat have consistently demonstrated impressive performance, often exceeding 96% accuracy on custom datasets. However, the highest accuracy reported in the table (97.5%) is achieved by a combination of models including MobileNet-V2, YOLOv4-tiny, Paddle OCR, and SVTR-tiny. This suggests that the use of multiple models, each specialized for specific tasks, can lead to significant improvements in overall performance. Additionally, leveraging diverse and comprehensive datasets, such as COCO, Stanford Car, and Firat University, plays a crucial role in training robust and accurate models.

By combining advanced deep learning techniques and large-scale datasets, this research contributes to the state-of-the-art in vehicle-make model recognition and ANPR, paving the way for further advancements in the field.

Compared to the existing studies, our proposed approach achieves a high accuracy of 97.5%, positioning it at the forefront of vehicle-make model recognition and ANPR. By leveraging a combination of advanced deep learning models and a diverse dataset, our method demonstrates superior performance in terms of both accuracy and generalization. This highlights the effectiveness of our approach in addressing the challenges associated with vehicle identification in real-world scenarios.

5. CONCLUSION AND FUTURE WORK

This research presents a novel system that integrates vehicle make and model detection with Automatic Number Plate Recognition (ANPR), achieving a remarkable 97.5% accuracy rate. By leveraging advanced deep learning techniques, including MobileNet-V2, YOLOx, YOLOv4-tiny, Paddle OCR, and SVTR-tiny, the system effectively processes real-time video feeds, even under challenging conditions such as fog, rain, and low light.

A key contribution of this work is the integration of multiple deep learning models into a unified framework, enabling the system to accurately identify vehicles in real-time. Additionally, the use of Grad-CAM provides valuable insights into the model's decision-making process, aiding in the identification of areas for improvement.

The proposed system has significant implications for various applications, including autonomous driving, traffic management, law enforcement, and security surveillance. By advancing the state-of-the-art in vehicle identification, this research paves the way for future innovations in computer vision.

Future research could explore further advancements, such as incorporating more sophisticated deep learning architectures, expanding the dataset to include more diverse scenarios, and developing real-time adaptive learning mechanisms to enhance the system's performance and adaptability. Additionally, deploying the system on edge devices and addressing ethical considerations, such as privacy and surveillance, are important areas for future exploration.

To further enhance the performance and capabilities of the proposed system, several promising avenues for future research can be explored:

1. **Real-time Adaptive Learning:**

Implementing real-time adaptive learning mechanisms to enable the system to continuously learn and improve its performance based on new data. This would enhance the system's ability to adapt to changing conditions and emerging challenges.

2. **Expanded Functionality:**

Extending the system's capabilities to include more complex tasks, such as detecting vehicle damage, identifying specific vehicle features, and recognizing driver behavior. This would broaden the system's applicability in various domains, including insurance, traffic safety, and autonomous driving.

3. **Robustness and Adversarial Attacks:**

Developing techniques to improve the system's robustness to adverse weather conditions, low-quality images, and potential adversarial attacks. This would enhance the system's reliability and security.

4. Edge Device Deployment:

Exploring the deployment of the system on edge devices to enable real-time processing and reduce latency. This would be particularly useful for applications that require immediate decision-making.

REFERENCES

[1] Davy, M.K., Banda, P.J. and Hamweendo, A., 2023. Automatic vehicle number plate recognition system. Phys Astron Int J, 7(1), pp.69-72.

[2] Burkpalli, V., Joshi, A., Warad, A.B. and Patil, A.,2022 Automatic Number Plate Recognition Using Tensorflow and Easyocr.

[3] Rafique, S., Gul, S., Jan, K. and Khan, G.M., 2023. Optimized real-time parking management framework using deep learning. Expert Systems with Applications, 220, p.119686.

[4] Jin, M., Sun, C. and Hu, Y., 2023. An intelligent traffic detection approach for vehicles on highway using pattern recognition and deep learning. Soft Computing, 27(8), pp.5041-5052.

[5] Sankaranarayanan, M., Mala, C. and Mathew, S., 2023. Efficient vehicle detection for traffic video-based intelligent transportation systems applications using recurrent architecture. Multimedia Tools and Applications, pp.1-19.

[6] Azfar, T., Li, J., Yu, H., Cheu, R.L., Lv, Y. and Ke, R., 2024. Deep Learning-Based Computer Vision Methods for Complex Traffic Environments Perception: A Review. Data Science for Transportation, 6(1), p.1.

[7] Rajyalakshmi, V. and Lakshmanna, K., 2024. Detection of car parking space by using Hybrid Deep DenseNet Optimization algorithm. International Journal of Network Management, 34(1), p.e2228.

[8] Mustafa, T. and Varol, A., 2020, June. Review of the internet of things for healthcare monitoring. In 2020 8th International Symposium on Digital Forensics and Security (ISDFS) (pp. 1-6). IEEE.

[9] Lyu, Yangxintong, Ionut Schiopu, Bruno Cornelis, and Adrian Munteanu. 2022. "Framework for Vehicle Make and Model Recognition—A New Large-Scale Dataset and an Efficient Two-Branch–Two-Stage Deep Learning Architecture" Sensors 22, no. 21: 8439. https://doi.org/10.3390/s22218439

[10] Saravanarajan, V.S., Chen, R.C., Dewi, C., Chen, L.S. and Ganesan, L., 2024. Car crash detection using ensemble deep learning. Multimedia Tools and Applications, 83(12), pp.36719-36737.

[11] Neupane, B.; Horanont, T.; Aryal, J. Real-Time Vehicle Classification and Tracking Using a Transfer Learning-Improved Deep Learning Network. Sensors 2022, 22, 3813.

[12] Mustafa, T. and Karabatak, M., 2023, May. Deep Learning Model for Automatic Number/License Plate Detection and Recognition System in Campus Gates. In 2023 11th International Symposium on Digital Forensics and Security (ISDFS) (pp. 1-5). IEEE.

[13] Mustafa, T. and Karabatak, M., 2023, May. Challenges in Automatic License Plate Recognition System Review. In 2023 11th International Symposium on Digital Forensics and Security (ISDFS) (pp. 1-6). IEEE.

[14] Kim, B., Natarajan, Y., Preethaa, K.S., Song, S., An, J. and Mohan, S., 2024. Real-time assessment of surface cracks in concrete structures using integrated deep neural networks with autonomous unmanned aerial vehicle. Engineering Applications of Artificial Intelligence, 129, p.107537.

[15] N. Vishwakarma, "Visualizing Model Insights: A Guide to Grad-CAM in Deep Learning," Analytics Vidhya, 11 1 2024. [Online]. Available: https://www.analyticsvidhya.com/blog/2023/12/grad-cam-in-deep-learning [Accessed 3 2 2024]

[16] Rathore, K.S., Battu, R.R., Kukreja, M., Sahni, N., Singh, A. and Sonia, S., 2024, January. A review on different techniques for vehicle number plates identification. In AIP Conference Proceedings (Vol. 2978, No. 1). AIP Publishing.

[17] Al-Hasan, T.M., Bonnefille, V. and Bensaali, F., 2024. Enhanced YOLOv8-Based System for Automatic Number Plate Recognition. Technologies, 12(9), p.164.

[18] Thakur, N., Bhattacharjee, E., Jain, R., Acharya, B. and Hu, Y.C., 2024. Deep learning-based parking occupancy detection framework using ResNet and VGG-16. Multimedia Tools and Applications, 83(1), pp.1941-1964.

[19] Gamadita, Y.T., 2024, June. Real-Time Parking Occupancy Detection using Motion Detection and Object Detection on Edge Device. In 2024 International Conference on Smart Computing, IoT and Machine Learning (SIML) (pp. 309-315). IEEE.

[20] Vizcarra, C., Alhamed, S., Algosaibi, A., Alnaeem, M., Aldalbahi, A., Aljaafari, N., Sawalmeh, A., Nazzal, M., Khreishah, A., Alhumam, A. and Anan, M., 2024. Deep learning adversarial attacks and defenses on license plate recognition system. Cluster Computing, pp.1-18.

[21] Gayen, S., Maity, S., Singh, P.K., Geem, Z.W. and Sarkar, R., 2023. Two decades of vehicle make and model recognition–survey, challenges and future directions. Journal of King Saud University-Computer and Information Sciences, p.101885.

[22] Moussaoui, H., El Akkad, N. and Benslimane, M., 2024. License plate text recognition using deep learning, NLP, and image processing techniques. Statistics, Optimization & Information Computing, 12(3), pp.685-696.

[23] Hasan Yusuf, F. and A Mangoud, M., 2024. Real-Time Car Parking Detection with Deep Learning in Different Lighting Scenarios. International Journal of Computing and Digital Systems, 15(1), pp.1-9.

[24] Ranjan, S. and Senthamilarasu, S., 2020. Book, Applied Deep Learning and Computer Vision for Self-Driving Cars: Build autonomous vehicles using deep neural networks and behavior-cloning techniques. Packt Publishing Ltd CH2

[25] Prakash, M., Janarthanan, M. and Devi, D., 2023, May. Multiple Objects Identification for Autonomous Car using YOLO and CNN. In 2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 597-601). IEEE.

[26] Khan, M.H.M., Khan, C.A.S. and Oumeir, R.M., 2023. Vehicle recognition using convolution neural network. International Journal of Biometrics, 15(3-4), pp.344-358.

[27] Islam, S.U., Ferraioli, G., Pascazio, V., Vitale, S. and Amin, M., 2024. Performance Analysis of YOLOv3, YOLOv4 and MobileNet SSD for Real Time Object Detection. The Sciencetech, 5(2), pp.37-49.

[28] Wang, Y., Xu, S., Wang, P., Li, K., Song, Z., Zheng, Q., Li, Y. and He, Q., 2024. Lightweight vehicle detection based on improved YOLOv5s. Sensors, 24(4), p.1182.

[29] Sarkar, O., Sinha, S., Jena, A.K., Parida, A.K., Parida, N. and Parida, R.K., 2024, June. Automatic Number Plate Character Recognition using Paddle-OCR. In 2024 International Conference on Innovations and Challenges in Emerging Technologies (ICICET) (pp. 1-7). IEEE.

[30] Jagtap, V.H., Dhotre, R.V., Khandare, U.R., Khuspe, H.N. and Kokare, R., 2024. Automatic license plate recognition system. ITEGAM-JETIA, 10(48), pp.129-134.

[31] Zheng, L., He, X., Samali, B. and Yang, L.T., 2013. An algorithm for accuracy enhancement of license plate recognition. Journal of computer and system sciences, 79(2), pp.245-255.

[32] Gautam, A., Rana, D., Aggarwal, S., Bhosle, S. and Sharma, H., 2023. Deep learning approach to automatically recognise license number plates. Multimedia Tools and Applications, 82(20), pp.31487-31504.

[33] Soleimanzadeh Rasteh, F. and Motamed, S., 2024. Automatic License Plate Recognition (ALPR) Using Improved Convolutional Neural Network. Computational Intelligence in Electrical Engineering, 14(4).

[34] Li, J., Zhang, J. and Zhou, Y., 2024, April. License Plate Recognition System Based on Matlab. In 2024 IEEE 2nd International Conference on Control, Electronics and Computer Technology (ICCECT) (pp. 43-47). IEEE.

[35] Aruna, V.S., Ravi, S. and Suruthi, M., 2024, February. Detection and Recognition of License Plates Using Color Image Processing. In International Conference on Communications and Cyber Physical Engineering 2018 (pp. 133-140). Singapore: Springer Nature Singapore.

[36] Deb, K., Chae, H.U. and Jo, K.H., 2009. Vehicle license plate detection method based on sliding concentric windows and histogram. J. Comput., 4(8), pp.771-777.

[37] Zhang, H., Jia, W., He, X. and Wu, Q., 2006, August. Learning-based license plate detection using global and local features. In 18th International Conference on Pattern Recognition (ICPR'06) (Vol. 2, pp. 1102-1105). IEEE.

[38] Ramajo-Ballester, Á., Moreno, J.M.A. and de la Escalera Hueso, A., 2024. Dual license plate recognition and visual features encoding for vehicle identification. Robotics and Autonomous Systems, 172, p.104608.

[39] Li, H., Wang, P., You, M. and Shen, C., 2018. Reading car license plates using deep neural networks. Image and Vision Computing, 72, pp.14-23.

[40] Parasuraman, K., Nallaperumal, K., Purusothaman, S.B. and Amitha, I.C., Automatic License-Plate Location Based on Feature Salience for Video Images.

[41] Al-Shemarry, M.S., Li, Y. and Abdulla, S., 2023. Detecting Distorted Vehicle Licence Plates Using Novel Preprocessing Methods With Hybrid Feature Descriptors. IEEE Intelligent Transportation Systems Magazine, 15(2), pp.6-25.

[42] Akbar, G., Iqbal, M.M., Ramzan, S., Majeed, S. and Farooq, M., 2024. License Plate Identification using Machine Learning Techniques. Journal of Computing & Biomedical Informatics.

[43] Ayaz, M., Shah, S.K., Javed, M., Assam, M., Khan, W. and Najeeb, F., 2022. Automatic Vehicle Number Plate Recognition Approach Using Color Detection Technique. International Journal of Innovations in Science & Technology, 3(5), pp.166-176.

[44] Mufti, N. and Shah, S.A.A., 2021. Automatic number plate Recognition: A detailed survey of relevant algorithms. Sensors, 21(9), p.3028.

[45] Ruchita V. Patel, Asst.Prof. Saket J. Swarndeep "Literature Survey on Vehicle Recognition for Augmented Reality Using Android Smart Phones" IRJET 2020.

[46] Sweta Singh "Automatic Car License Plate Detection System: A Review" Materials Science and Engineering 2021.

[47] Ren and S. Lan, "Vehicle make and model recognition based on convolutional neural networks," in 7th IEEE Int. Conf. on Software Engineering and Service Science (ICSESS), 692 –695 (2016). https://doi.org/10.1109/ICSESS.2016.7883162.

[48] J. Prokaj and G. Medioni, "3-D model-based vehicle recognition," in Workshop on Applications of Computer Vision (WACV), 1 –7 (2009). https://doi.org/10.1109/WACV.2009.5403032.

[49] A. J. Siddiqui, A. Mammeri and A. Boukerche, "Real-time vehicle make and model recognition based on a bag of surf features," IEEE Trans. Intell. Transp. Syst., 17 3205 –3219 (2016). https://doi.org/10.1109/TITS.2016.2545640

[50] V. S. Petrovic and T. F. Cootes, "Analysis of features for rigid structure vehicle type recognition," in Proc. of the British Machine Vision Conf., 587 –596 (2004). https://doi.org/10.5244/C.18.61

[51] R. G. J. Wijnhoven and P. H. N. de With, "Unsupervised sub-categorization for object detection: finding cars from a driving vehicle," in IEEE Int. Conf. on Computer Vision Workshops (ICCV Workshops), 2077 –2083 (2011). https://doi.org/10.1109/ICCVW.2011.6130504

[52] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR'05), 886 –893 (2005). https://doi.org/10.1109/CVPR.2005.177

[53] Y. Zhou et al., "DAVE: a unified framework for fast vehicle detection and annotation," Lect. Notes Comput. Sci., 9906 278 –293 (2016). https://doi.org/10.1007/978-3-319-46475-6_18

[54] Zwemer, M.H., Brouwers, G.M., Wijnhoven, R.G. and de With, P.H., 2018. Semiautomatic training and evaluation of a learning-based vehicle make and model recognition system. Journal of Electronic Imaging, 27(5), pp.051225-051225.

[55] J. Wei, "AlexNet: The architecture that challenged CNNs," Towards Data Science. Accessed Nov.15, 2020. [Online].Available: https://towardsdatascience.com/alexnet-the-architecture-that-challenged-cnnse406d5297951

[56] Luo, R. Shen, J. Hu, J. Deng, L. Hu, and Q. Guan, “A deep convolution neuralnetwork model for vehicle recognition and face recognition,” Procedia Computer Science, vol. 107, pp. 715–720, Dec. 2017.

[57] H. Lee, I. Ullah, W. Wan, Y. Gao, and Z. Fang, “Real-time vehicle make and model recognition with the residual SqueezeNet Architecture,” Sensors, vol. 19, no. 5, p. 982, Feb. 26, 2019.

[58] LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W. and Jackel, L.D., 1989. Backpropagation applied to handwritten zip code recognition. Neural computation, 1(4), pp.541-551.

[59] A. Krizhevsky, I. Sutskever and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Advances in Neural Information Processing Systems, (2012)

[60] O. Russakovsky et al., “Imagenet large scale visual recognition challenge,” Int. J. Comput. Vision, 115 (3), 211 –252 (2015). https://doi.org/10.1007/s11263-015-0816-y IJCVEQ 0920-5691

[61] Y. Ren and S. Lan, “Vehicle make and model recognition based on convolutional neural networks,” in 7th IEEE Int. Conf. on Software Engineering and Service Science (ICSESS), 692 –695 (2016). https://doi.org/10.1109/ICSESS.2016.7883162

[62] L. Yang et al., “A large-scale car dataset for fine-grained categorization and verification,” in IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 3973 –3981 (2015). https://doi.org/10.1109/CVPR.2015.7299023

[63] P. Sermanet et al., “Overfeat: integrated recognition, localization and detection using convolutional networks,” in Int. Conf. on Learning Representations (ICLR), (2014).

[64] C. Szegedy et al., “Going deeper with convolutions,” in IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), (2015). https://doi.org/10.1109/CVPR.2015.7298594.

[65] representations for fine-grained categorization. In 4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)

[66] Shah, D., Shah, V., Vora, D., Thaker, T. and Zobaliya, J., 2021. Car Image Classification and Recognition. International Journal for Research in Applied Science & Engineering Technology (IJRASET).

[67] Trivedi, J., Devi, M.S. and Dhara, D., 2021. Vehicle classification using the convolution neural network approach. Zeszyty Naukowe. Transport/Politechnika Śląska.

[68] Jahan, N., Islam, S. and Foysal, M.F.A., 2020, July. Real-time vehicle classification using CNN. In 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT) (pp. 1-6). IEEE.

[69] Patil, P., 2022. Applications of deep learning in traffic management: A review. International Journal of Business Intelligence and Big Data Analytics, 5(1), pp.16-23.

[70] Mufti, N. and Shah, S.A.A., 2021. Automatic number plate Recognition: A detailed survey of relevant algorithms. Sensors, 21(9), p.3028.

[71] Ruchita V. Patel, Asst.Prof. Saket J. Swarndeep “Literature Survey on Vehicle Recognition for Augmented Reality Using Android Smart Phones” IRJET 2020.

[72] Sweta Singh “Automatic Car License Plate Detection System: A Review” Materials Science and Engineering 2021.

[73] S. Saha, S. Basu, and M. Nasipuri, ‘‘Automatic localization and recognition of license plate characters for Indian vehicles,’’ Int. J. Comput. Sci. Emerg. Technol., vol. 2, pp. 520–533, Aug. 2011

[74] N. Abirami, ‘‘Accurate vehicle number plate recognition and real-time identification using Raspberry Pi,’’ Int. Res. J. Eng. Technol., vol. 5, no. 4, p. 7, 2018.

[75] S. Fakhar, M. Saad, A. Fauzan, R. Affendi, and M. Aidil, ‘‘Development of portable automatic number plate recognition (ANPR) system on Raspberry Pi,’’ Int. J. Electr. Comput. Eng. IJECE, vol. 9, no. 3, pp. 1805–1813, Jun. 2019, doi: 10.11591/ijece.v9i3.pp1765-1772.

[76] S. Tenzin, P. Dorji, B. Subba, and T. Tobgay, ‘‘Smart check-in check-out system for vehicles using automatic number plate recognition,’’ in Proc. 11th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT), Jul. 2020, pp. 1–6, doi: 10.1109/ICCCNT49239.2020.9225555.

[77] N. D. Keraf, P. Ehkan, V. H. Kelian, N. M. Noor, H. Darus, and E. Loke, ‘‘Automatic vehicle identification system using number plate recognition in POLIMAS,’’ IOP Conf. Ser., Mater. Sci. Eng., vol. 767, Mar. 2020, Art. no. 012056, doi: 10.1088/1757-899X/767/1/012056.

[78] J.-S. Chou and C.-H. Liu, ‘‘Automated sensing system for real-time recognition of trucks in river dredging areas using computer vision and convolutional deep learning,’’ Sensors, vol. 21, no. 2, p. 555, Jan. 2021, doi: 10.3390/s21020555.

[79] Y. Chen and W. Hu, ‘‘A video-based method with strong-robustness for vehicle detection and classification based on static appearance features and motion features,’’ *IEEE Access*, vol. 9, pp. 13083–13098, 2021, doi: 10.1109/ACCESS.2021.3051659

[80] N. D. V. Dalarmelina, M. A. Teixeira, and R. I. Meneguette, ‘‘A real-time automatic plate recognition system based on optical character recognition and wireless sensor networks for ITS,’’ Sensors, vol. 20, no. 1, p. 55, Dec. 2019, doi: 10.3390/s20010055.

[81] R. N. Babu, V. Sowmya, and K. P. Soman, ‘‘Indian car number plate recognition using deep learning,’’ in Proc. 2nd Int. Conf. Intell. Comput., Instrum. Control Technol. (ICICICT), Jul. 2019, pp. 1269–1272, doi: 10.1109/ICICICT46008.2019.8993238.

[82] S. Sugeng and E. Y. Syamsuddin, ‘‘Designing automatic number plate recognition (ANPR) systems based on K-NN machine learning on the Raspberry Pi embedded system,’’ Jurnal Teknik Elektro dan Vokasional, vol. 5, no. 1, Sep. 2019, Art. no. 1, doi: 10.24036/jtev.v5i1.1.106135.

[83] Adithya, T.G., Pavithra, G. and Manjunath, T.C., 2022. Automatic Number Plate Recognition Idea Development using AI-based ANNs. Journal of Communication Engineering and its Innovations, 8(1), pp.1-9.

[84] Sharma, G., 2018. Performance analysis of vehicle number plate recognition system using template matching techniques. Journal of Information Technology & Software Engineering, 8(2), pp.1-9.

[85] Xie, F., Zhang, M., Zhao, J., Yang, J., Liu, Y. and Yuan, X., 2018. A robust license plate detection and character recognition algorithm based on a combined feature extraction model and BPNN. Journal of Advanced Transportation, 2018.

[86] Yimyam, W. and Ketcham, M., 2017, March. The automated parking fee calculation using license plate recognition system. In 2017 International Conference on Digital Arts, Media and Technology (ICDAMT) (pp. 325-329). IEEE.

[87] Singh, J. and Bhushan, B., 2019, October. Real time Indian license plate detection using deep neural networks and optical character recognition using LSTM tesseract. In 2019 international conference on computing, communication, and intelligent systems (ICCCIS) (pp. 347-352). IEEE.

[88] Anekar, S., Yeginwar, S. and Sonune, H., 2022. Automated gate system using number plate recognition (NPR). In ICT Systems and Sustainability: Proceedings of ICT4SD 2021, Volume 1 (pp. 413-420). Singapore: Springer Nature Singapore.

[89] Hossain, S.N., Hassan, M.Z. and Masba, M.M.A., 2022. Automatic License Plate Recognition System for Bangladeshi Vehicles Using Deep Neural Network. In Proceedings of the International Conference on Big Data, IoT, and Machine Learning: BIM 2021 (pp. 91-102). Springer Singapore.

[90] Sharma, N., Haq, M.A., Dahiya, P.K., Marwah, B.R., Lalit, R., Mittal, N. and Keshta, I., 2023. Deep Learning and SVM-Based Approach for Indian Licence Plate Character Recognition. Cmc-Computers Materials & Continua, 74(1), pp.881-895.

[91] Salunke, D., Joshi, R., Patil, R. and Patinge, S., 2023. Automatic Vehicle Plate Recognition using Deep Learning.

[92] Ghulyani, S., Jain, D., Singh, P., Joshi, S. and Ahlawat, A., 2022, May. A Data Entry Optical Character Recognition Tool using Convolutional Neural Networks. In 2022 IEEE IAS Global Conference on Emerging Technologies (GlobConET) (pp. 721-728). IEEE.

[93] Sferle, R.M. and Moisi, E.V., 2019, June. Automatic number plate recognition for a smart service auto. In 2019 15th international conference on engineering of modern electric systems (EMES) (pp. 57-60). IEEE.

[94] Ammar, A., Koubaa, A., Boulila, W., Benjdira, B. and Alhabashi, Y., 2023. A multi-stage deep-learning-based vehicle and license plate recognition system with real-time edge inference. Sensors, 23(4), p.2120.

[95] Singh, A. and Agarwal, S., 2023. Cloud-Based License Plate Recognition for Smart City Using Deep Learning. In Cloud-based Intelligent Informative Engineering for Society 5.0 (pp. 141-156). Chapman and Hall/CRC.

[96] Jain, S., Patel, S., Mehta, A. and Verma, J.P., 2022, December. Number Plate Detection Using Drone Surveillance. In 2022 IEEE 9th Uttar Pradesh Section International Conference on Electrical, Electronics and Computer Engineering (UPCON) (pp. 1-6). IEEE.

[97] Ozbaran, Y. and Tasgin, S., 2019. Using Cameras of Automatic Number Plate Recognition System For Seat Belt Enforcement A Case Study of Sanliurfa (Turkey). Policing: An International Journal, 42(4), pp.688-700

[98] Kaur, P., Kumar, Y., Ahmed, S., Alhumam, A., Singla, R. and Ijaz, M.F., 2022. Automatic license plate recognition system for vehicles using a cnn. Computers, Materials & Continua, 71(1), pp.35-50.

[99] Usama, M., Anwar, H., Shahid, M.M., Anwar, A., Anwar, S. and Hlavacs, H., 2022. Vehicle and License Plate Recognition with Novel Dataset for Toll Collection. arXiv preprint arXiv:2202.05631.

[100] Truter, J.C., 2019. Evaluation of VANET standards and protocols for distributed licence plate detection and reporting (Doctoral dissertation).

[101] Tang, J., Wan, L., Schooling, J., Zhao, P., Chen, J. and Wei, S., 2022. Automatic number plate recognition (ANPR) in smart cities: A systematic review on technological advancements and application cases. Cities, 129, p.103833.

[102] Asaad, a., zaki, h.f.m. And faizabadi, a.r., 2022. Malaysian automatic license plate recognition using single-shot object detection model at low visibility and unconstrained environment. Perintis ejournal, 12(2), pp.81-94.

[103] Saha, S., 2019. A review on automatic license plate recognition system. arXiv preprint arXiv:1902.09385

[104] Yu, Y., Cui, Y., Zeng, J., He, C. and Wang, D., 2022. Identifying traffic clusters in urban networks based on graph theory using license plate recognition data. Physica A: Statistical Mechanics and its Applications, 591, p.126750.

[105] Tonge, S., Hemwani, B., Thawani, G., Katara, S. and Dhanwani, N., 2022. Automatic Number Plate Recognition. published by International Research Journal of Modernization in Engineering Technology and Science, 4

[106] Paulin, G. and Ivasic-Kos, M., 2023. Review and analysis of synthetic dataset generation methods and techniques for application in computer vision. Artificial Intelligence Review, pp.1-45.

[107] Najman, P. and Zemcík, P., 2022. Stereo camera pair calibration for traffic surveillance applications. Optical Engineering, 61(11), p.114103.

[108] Srinivasan, S., Prabha, D., Raffic, N.M., Babu, K.G., Thirumurugaveerakumar, S. and Sangeetha, K., 2022. Automated Vehicle Number Plate Recognition System, Using Convolution Long Short-Term Memory Technique. In Object Detection with Deep Learning Models (pp. 101-115). Chapman and Hall/CRC.

[109] Srividhya, S.R., Kavitha, C., Lai, W.C., Mani, V. and Khalaf, O.I., 2022. A Machine Learning Algorithm to Automate Vehicle Classification and License Plate Detection. Wireless Communications and Mobile Computing, 2022.

[110] The Difference Between Make and Model?, Hearst Autos Research, https://www.caranddriver.com/research/a31875496/difference-between-make-and-model/ accessed on September 1st , 2024

[111] The Make and Model of a Car?, https://www.finn.com/en-US/blog/car-buying/car-make-and-model accessed on September 1st , 2024.

[112] Kumawat, K., Jain, A. and Tiwari, N., 2024. Relevance of Automatic Number Plate Recognition Systems in Vehicle Theft Detection. Engineering Proceedings, 59(1), p.185.

[113] Shyaa, T.A. and Hashim, A.A., 2024. Superior Use of YOLOv8 to Enhance Car License Plates Detection Speed and Accuracy. Revue d'Intelligence Artificielle, 38(1).

[114] Mustafa, T. and Karabatak, M., 2024. Real time car model and plate detection system by using deep learning architectures. IEEE Access.

[115] Tippannavar, S.S. and Yashwanth, S.D., 2023. Real-time vehicle identification for improving the traffic management system-a review. Journal of Trends in Computer Science and Smart Technology, 5(3), pp.323-342.

[116] Hector Martinez, what is Deep Learning? on April 17, 2021. Revised and updated 4/10/2023. [Online]. Available: https://pyimagesearch.com/2021/04/17/what-is-deep-learning/ Accessed August.25th , 2024.

[117] Introduction to Deep Learning, Last Updated : 26 May, 2024, [Online]. Available: https://www.geeksforgeeks.org/introduction-deep-learning/ Accessed August.2nd , 2024.

[118] Jim Holdsworth, Mark Scapicchio, what is deep learning? Updated: 17 June 2024, [online]. Available: https://www.ibm.com/topics/deep-learning Accessed August.4th , 2024.

[119] Simplilearn, Unveiling the Power of Deep Learning: What You Must Know, last updated on Sep 17, 2024, [online]. Available: https://www.simplilearn.com/tutorials/deep-learning-tutorial/what-is-deep-learning, Accessed August.7th , 2024.

[120] Arne Wolfewicz, Deep Learning vs. Machine Learning – What's The Difference? February 15th , 2023, [online]. Available: https://levity.ai/blog/difference-machine-learning-deep-learning Accessed August.24, 2024.

[121] Miikkulainen, R., Liang, J., Meyerson, E., Rawal, A., Fink, D., Francon, O., Raju, B., Shahrzad, H., Navruzyan, A., Duffy, N. and Hodjat, B., 2024. Evolving deep neural networks. In Artificial intelligence in the age of neural networks and brain computing (pp. 269-287). Academic Press.

[122] Mahrishi, M., Hiran, K.K., Meena, G. and Sharma, P. eds., 2020. Machine learning and deep learning in real-time applications. IGI global.

[123] Möller, D.P., 2023. Machine learning and deep learning. In Guide to Cybersecurity in Digital Transformation: Trends, Methods, Technologies, Applications and Best Practices (pp. 347-384). Cham: Springer Nature Switzerland.

[124] Manakitsa, N., Maraslidis, G.S., Moysis, L. and Fragulis, G.F., 2024. A review of machine learning and deep learning for object detection, semantic segmentation, and human action recognition in machine and robotic vision. Technologies, 12(2), p.15.

[125] Advantages and Disadvantages of Deep Learning, Last Updated: 30 Jan, 2023, [online]. Available: https://www.geeksforgeeks.org/advantages-and-disadvantages-of-deep-learning/ Accessed August.05th , 2024.

[126] El-Nabi, S.A., El-Shafai, W., El-Rabaie, E.S.M., Ramadan, K.F., Abd El-Samie, F.E. and Mohsen, S., 2024. Machine learning and deep learning techniques for driver fatigue and drowsiness detection: a review. Multimedia Tools and Applications, 83(3), pp.9441-9477.

[127] Aditya Sharma, Introduction to the YOLO Family, April 4, 2022, [online]. Available: https://pyimagesearch.com/2022/04/04/introduction-to-the-yolo-family/ Accessed July.05th , 2024.

[128] Yuan, G., Chen, L., Cai, J., Yang, C. and Liu, J., 2024, May. A Comprehensive Overview of Object Detection Based on Deep Learning. In 2024 10th IEEE International Conference on Intelligent Data and Security (IDS) (pp. 80-85). IEEE.

[129] Kaur, J. and Singh, W., 2024. A systematic review of object detection from images using deep learning. Multimedia Tools and Applications, 83(4), pp.12253-12338.

[130] Mukhopadhyay, A. and Biswas, P., 2024. Advancements in Deep-Learning-Based Object Detection in Challenging Environments. Wireless World Research and Trends Magazine, pp.1-6.

[131] Santiary, P.A.W., Swardika, I.K., Dewi, D.A.I.C. and Sugirianta, I.B.K., 2023. Intra-class deep learning object detection on embedded computer system. Int J Artif Intell ISSN, 2252(8938), p.8938.

[132] Ghosh, K., Bellinger, C., Corizzo, R., Branco, P., Krawczyk, B. and Japkowicz, N., 2024. The class imbalance problem in deep learning. Machine Learning, 113(7), pp.4845-4901.

[133] Viola–Jones object detection framework, 2004, [online]. Available: https://en.wikipedia.org/wiki/Viola%E2%80%93Jones_object_detection_framework , Accessed July.05th , 2024.

[134] Tan, M., Pang, R. and Le, Q.V., 2020. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 10781-10790).

[135] Girshick, R., Donahue, J., Darrell, T. and Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).

[136] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y. and Berg, A.C., 2016. Ssd: Single shot multibox detector. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14 (pp. 21-37). Springer International Publishing.

[137] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A., 2016. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788).

[138] Redmon, J. and Farhadi, A., 2017. YOLO9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7263-7271).

[139] Redmon, J., 2018. YOLOv3: an incremental improvement/Joseph Redmon, Ali Farhadi-University of.

[140] Bochkovskiy, A., Wang, C.Y. and Liao, H.Y.M., 2020. Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934.

[141] Glenn Jocher, YOLOv5, 2020, [online]. Available: https://github.com/ultralytics/yolov5, Accessed September, 2nd , 2024.

[142] Long, X., Deng, K., Wang, G., Zhang, Y., Dang, Q., Gao, Y., Shen, H., Ren, J., Han, S., Ding, E. and Wen, S., 2020. PP-YOLO: An effective and efficient implementation of object detector. arXiv preprint arXiv:2007.12099.

[143] PaddleDetection,TingquanGao, A High-Efficient Development Toolkit for Object Detection based on PaddlePaddle, https://github.com/PaddlePaddle/PaddleDetection/blob/release/2.7/README_en.md [online]. Available. Accessed September, 2nd, 2024.

[144] Huang, X., Wang, X., Lv, W., Bai, X., Long, X., Deng, K., Dang, Q., Han, S., Liu, Q., Hu, X. and Yu, D., 2021. PP-YOLOv2: A practical object detector. arXiv preprint arXiv:2104.10419.

[145] Ge, Z., 2021. Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430.

[146] Krauss, P., 2024. Artificial Intelligence and Brain Research: Neural Networks, Deep Learning and the Future of Cognition. Springer Nature.

[147] Chen, Y., Wang, S., Lin, L., Cui, Z. and Zong, Y., 2024. Computer Vision and Deep Learning Transforming Image Recognition and Beyond. International Journal of Computer Science and Information Technology, 2(1), pp.45-51.

[148] Quantum News, Yann LeCun. The French AI Pioneer Behind the Convolutional Neural Network, 2024, [online]. Available https://quantumzeitgeist.com/yann-lecun-the-french-ai-pioneer-behind-the-convolutional-neural-network/ Accessed August, 2nd, 2024.

[149] Zhao, X., Wang, L., Zhang, Y., Han, X., Deveci, M. and Parmar, M., 2024. A review of convolutional neural networks in computer vision. Artificial Intelligence Review, 57(4), p.99.

[150] Kaul, D., Raju, H. and Tripathy, B.K., 2022. Deep learning in healthcare. Deep Learning in Data Analytics: Recent Techniques, Practices and Applications, pp.97-115.

[151] Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M.A., Al-Amidie, M. and Farhan, L., 2021. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. Journal of big Data, 8, pp.1-74.

[152] Avijeet Biswal, Convolutional Neural Network Tutorial, last updated on Sep 17, 2024, [online]. Available:https://www.simplilearn.com/tutorials/deep-learning-tutorial/convolutional-neural-network Accessed July, 2nd, 2024.

[153] Zhao, L. and Zhang, Z., 2024. A improved pooling method for convolutional neural networks. Scientific Reports, 14(1), p.1589.

[154] Paolini, E., Valcarenghi, L., Maggiani, L. and Andriolli, N., 2024. Real-Time Network Packet Classification Exploiting Computer Vision Architectures. IEEE Open Journal of the Communications Society.

[155] Cong, S. and Zhou, Y., 2023. A review of convolutional neural network architectures and their optimizations. Artificial Intelligence Review, 56(3), pp.1905-1969.

[156] Li, M., Zhu, Z., Xu, R., Feng, Y. and Xiao, L., 2024. Research on image classification and semantic segmentation model based on convolutional neural network. Journal of Computing and Electronic Information Management, 12(3), pp.94-100.

[157] Stieger, J.R., Engel, S.A., Suma, D. and He, B., 2021. Benefits of deep learning classification of continuous noninvasive brain–computer interface control. Journal of neural engineering, 18(4), p.046082.

[158] Dhillon, A. and Verma, G.K., 2020. Convolutional neural network: a review of models, methodologies and applications to object detection. Progress in Artificial Intelligence, 9(2), pp.85-112.

[159] Tareen, S.A.K. and Tareen, F.K., 2023. Convolutional neural networks for beginners. Available at SSRN 4566310.

[160] Javed Mehedi Shamrat, F.M., Chakraborty, S., Afrin, S., Moharram, M.S., Amina, M. and Roy, T., 2022, July. A model based on convolutional neural network (CNN) for vehicle classification. In Congress on Intelligent Systems: Proceedings of CIS 2021, Volume 1 (pp. 519-530). Singapore: Springer Nature Singapore

[161] Javed Mehedi Shamrat, F.M., Chakraborty, S., Afrin, S., Moharram, M.S., Amina, M. and Roy, T., 2022, July. A model based on convolutional neural network (CNN) for vehicle classification. In Congress on Intelligent Systems: Proceedings of CIS 2021, Volume 1 (pp. 519-530). Singapore: Springer Nature Singapore.

[162] Muhib, R.B., Ahmad, I.S. and Boufama, B., 2023, October. Deep Learning-Based Vehicle Classification. In Proceedings of the Future Technologies Conference (pp. 244-258). Cham: Springer Nature Switzerland.

[163] Qin, Y., Tang, Q., Xin, J., Yang, C., Zhang, Z. and Yang, X., 2023. A Rapid Identification Technique of Moving Loads Based on MobileNetV2 and Transfer Learning. Buildings, 13(2), p.572.

[164] Nekkanti, L.K.K. and Rao, V., 2023, June. A Yolo-Based Deep Learning Approach for Vehicle Class Classification. In International Conference on Multi-disciplinary Trends in Artificial Intelligence (pp. 554-568). Cham: Springer Nature Switzerland.

[165] Han, X., Wang, Y., Zhai, B., You, Q. and Yang, H., 2024. COCO is" ALL"You Need for Visual Instruction Fine-tuning. arXiv preprint arXiv:2401.08968.

[166] Peng, Y., He, L., Hu, D., Liu, Y., Yang, L. and Shang, S., 2024. Decoupling Deep Learning for Enhanced Image Recognition Interpretability. ACM Transactions on Multimedia Computing, Communications and Applications.

CURRICULUM VITAE

Twana Saeed MUSTAFA

[redacted]

ACADEMIC ACTIVITIES

Paper:

1. Mustafa, T. and Karabatak, M., 2024. Real time car model and plate detection system by using deep learning architectures. IEEE Access.
2. Shwany, Z.A.A., Ismail, S.K., Khoshnaw, K.H.K., Mustafa, T.S., Karabatak, M. and Othman, N.A., 2024, April. A Real-Time Hand Sign Language Recognition System for Threatening Situations Using Deep Learning. In 2024 12th International Symposium on Digital Forensics and Security (ISDFS) (pp. 1-6). IEEE.

3. Taha, H.M., Mohammed, M.A., Jghef, Y.S., Sulaiman, S.K. and Mustafa, T.S., 2024. The integration of blockchain technology in 5G-enabled IoT. In ITM Web of Conferences (Vol. 64, p. 01001). EDP Sciences.
4. Mustafa, T. and Karabatak, M., 2023, May. Deep Learning Model for Automatic Number/License Plate Detection and Recognition System in Campus Gates. In 2023 11th International Symposium on Digital Forensics and Security (ISDFS) (pp. 1-5). IEEE.
5. Mustafa, T. and Karabatak, M., 2023, May. Challenges in Automatic License Plate Recognition System Review. In 2023 11th International Symposium on Digital Forensics and Security (ISDFS) (pp. 1-6). IEEE.
6. Mustafa, T. and Karabatak, M., 2023, May. Feature selection for phishing website by using naive bayes classifier. In 2023 11th International Symposium on Digital Forensics and Security (ISDFS) (pp. 1-4). IEEE.
7. Mustafa, T., Shwany, Z.A., Saeed, S.H., Khoshnaw, K., Ismail, S.K. and Rashid, R.F., 2022. Geolocation-Based Mobile Application. International Journal of Software Innovation (IJSI), 10(1), pp.1-12.
8. Khoshnaw, K.H.K., Shwany, Z.A.A., Mustafa, T. and Ismail, S.K., 2022. Mobile recommender system based on smart city graph. Indones. J. Electr. Eng. Comput. Sci, 25(3), pp.1771-1776.
9. Karabatak, M., Mustafa, T. and Hamaali, C., 2020, June. Remote monitoring real time air pollution-IoT (cloud based). In 2020 8th International Symposium on Digital Forensics and Security (ISDFS) (pp. 1-6). IEEE.
10. Mustafa, T. and Varol, A., 2020, June. Review of the internet of things for healthcare monitoring. In 2020 8th International Symposium on Digital Forensics and Security (ISDFS) (pp. 1-6). IEEE.
11. Mousa, A., Karabatak, M. and Mustafa, T., 2020, June. Database security threats and challenges. In 2020 8th International Symposium on Digital Forensics and Security (ISDFS) (pp. 1-5). IEEE.