CLASSIFICATION OF MANEUVERS OF VEHICLES IN FRONT FOR DRIVER ASSISTANCE SYSTEMS

A Thesis Submitted to
the Graduate School of Engineering and Sciences of
İzmir Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of

DOCTOR OF PHILOSOPHY

in Computer Engineering

by
Yağız NALÇAKAN

July 2023
İZMİR

We approve the thesis of **Yağız NALÇAKAN**

Examining Committee Members:

Prof. Dr. Yalın BAŞTANLAR
Department of Computer Engineering, Izmir Institute of Technology

---

Assoc. Prof. Dr. Hacer YALIM KELEŞ
Department of Computer Engineering, Hacettepe University

Assoc. Prof. Dr. Mustafa ÖZUYSAL
Department of Computer Engineering, Izmir Institute of Technology

---

Assoc. Prof. Dr. Zerrin IŞIK
Department of Computer Engineering, Dokuz Eylül University

Asst. Prof. Dr. Nesli ERDOĞMUŞ
Department of Computer Engineering, Izmir Institute of Technology

20 July 2023

Prof. Dr. Yalın BAŞTANLAR
Supervisor, Department of Computer
Engineering
Izmir Institute of Technology

Prof. Dr. Cüneyt BAZLAMAÇCI
Head of the Department of
Computer Engineering

Prof. Dr. Mehtap EANES
Dean of the Graduate School of
Engineering and Sciences

ACKNOWLEDGMENTS

Firstly, I am indebted to my beloved wife Şeyda, who has supported me throughout my doctoral studies and even before, reminding me of my goals when I felt exhausted and disheartened, and to our playful dog Chomsky, who has kept us entertained with his antics during this process.

I have always thought that the process of forming our personalities and characters resembles the construction of a pyramid, climbing step by step and placing large stones on top of each other as we age. Therefore, I would like to express my gratitude to my mother and father, who have contributed immeasurably to who I am today, and to my grandfather and grandmother, who taught me to be humble and to understand human relationships during my primary and middle school years, and to my other grandfather and grandmother who taught me thoughtful thinking, to be a kind, generous, and detail-oriented person, and also equality, desire for freedom, and attention to detail during my high school years.

And when it comes to my doctoral education, I would like to thank my advisor Prof. Dr. Yalın Baştanlar, who enabled me to take perhaps the most important step in forming my personality and future and from whom I learn new things day by day.

Lastly, I am grateful to my professors, Assoc. Prof. Dr. Hacer Yalım Keleş and Assoc. Prof. Dr. Mustafa Özuysal, who have agreed to be part of my thesis monitoring committee. Their feedback and advice from the first days of my research to its conclusion have helped me achieve better results and have fostered my growth as a researcher and academic. Furthermore, my deep gratitude extends to Assoc. Prof. Dr. Zerrin Işık and Asst. Prof. Dr. Nesli Erdoğmuş, who agreed to be part of my thesis defense jury. I am thankful for their willingness to assist and their valuable contributions to my doctoral thesis.

This thesis work has been supported by the Scientific and Technological Research Council of Turkey (TUBITAK) 2244 Scholarship under Grant No: 2244-118C079. A significant part of the numerical calculations reported in this thesis were conducted at TUBITAK ULAKBIM, High Performance and Grid Computing Center (TRUBA resources).

ABSTRACT

CLASSIFICATION OF MANEUVERS OF VEHICLES IN FRONT FOR DRIVER ASSISTANCE SYSTEMS

Predicting vehicle maneuvers is a critical task for developing autonomous driving. These maneuvers have been identified as leading causes of fatal accidents, underscoring the need for robust and reliable detection systems. This thesis addresses this critical issue by developing and evaluating novel methodologies for classifying maneuvers, especially lane change and cut-in maneuvers in front of the vehicle. Two specific methods are proposed in this thesis work, and their effectiveness is evaluated on two datasets: the Prevention Lane Change Prediction dataset and the BDD-100K Cut-in/Lane-pass Classification Subset.

The first method is a model that utilizes features extracted from the bounding boxes of the target vehicle, feeding them into a single-layer LSTM network for cut-in/lane-pass classification. The second method involves training a 3-dimensional residual neural network in a self-supervised manner using contrastive video representation learning. For the self-supervised training phase, a novel scene representation is proposed to highlight vehicle motions. Afterward, the same model is fine-tuned using labeled video data. Lastly, an ensemble learning approach is introduced, which combines the predictive capabilities of the proposed LSTM-based and self-supervised contrastive video representation learning models, leveraging the strengths of both methods to enhance the overall maneuver classification performance.

The proposed methods made significant contributions to the field. The LSTM-based model achieved high classification accuracies compared to other studies in the literature. The self-supervised video representation learning model represents the first application of contrastive learning in maneuver classification. The ensemble learning approach has shown a significant improvement in the performance of the maneuver detection system.

ÖZET

SÜRÜCÜ DESTEK SİSTEMLERİ İÇİN ÖNDEKİ ARAÇLARIN MANEVRALARININ SINIFLANDIRILMASI

Araç manevralarının otonom sürüşün geliştirilmesi için kritik bir görevdir. Bu manevralar ölümcül kazaların önde gelen nedenlerindendir, bu yüksek başarılı ve güvenilir tespit sistemlerine duyulan ihtiyacın altını çizmektedir. Tez kapsamında araç içi kamera verilerini kullanarak aracın önünde meydana gelen şerit değiştirme ve ön kesme manevralarını sınıflandırmak için yeni metodolojiler önerilmiştir ve bunların etkinliği iki veri kümesi üzerinde değerlendirilmiştir: "Prevention" Şerit Değiştirme Tahmini veri seti ve "BDD-100K" Ön Kesme / Şerit Paralel Geçme Sınıflandırma veri seti.

İlk yöntem olan LSTM tabanlı yöntem, hedef aracın sınırlayıcı kutularından çıkarılan özellikleri kullanan ve bunları ön kesme/şerit paralel geçme sınıflandırması için tek katmanlı bir LSTM ağına besleyen bir modeldir. İkinci yöntem, araç manevralarını içeren video görüntüleri üzerinde "contrastive video representation learning (CVRL)" kullanarak 3-boyutlu bir artık sinir ağını kendi kendine denetimli bir şekilde eğitmeyi içerir. Bu yöntem için, araç hareketlerini vurgulamak üzere yeni bir sahne temsili önerilmiştir. Ardından, etiketli video verileri kullanılarak aynı modele ince ayar yapılır. Bu yöntem, şerit değişikliği algılama ve ön kesme manevrası algılama görevleri üzerinde değerlendirilmiştir. Son olarak, önerilen LSTM ve CVRL modellerinin tahmin yeteneklerini birleştiren ve şerit değişikliği manevra algılama sisteminin genel performansını artırmak için her iki yöntemin güçlü yönlerinden yararlanan bir topluluk öğrenme yaklaşımı tanıtılmıştır.

LSTM tabanlı model, basitliğine rağmen literatürdeki diğer çalışmalara kıyasla yüksek sınıflandırma doğrulukları elde etmiştir. Kendi kendine denetimli video temsili öğrenme modeli, manevra sınıflandırmasında "contrastive" öğrenmenin ilk uygulamasını temsil etmektedir. LSTM tabanlı ve CVRL modellerini entegre eden topluluk öğrenme yaklaşımı, manevra tespit sisteminin performansında önemli bir gelişme göstererek çoklu öğrenme algoritmalarından yararlanma potansiyelini ortaya koymuştur.

To Şeyda, Chomsky, and my family.

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

Figure **Page**

LIST OF TABLES

Table **Page**

LIST OF ABBREVIATIONS

3D ..... Three Dimensional

ACC ..... Accuracy

ADAS ..... Advanced Driver Assistance Systems

AD ..... Autonomous Driving

BDD ..... Berkeley Deep Drive

BEV ..... Bird's Eye View

CM ..... Confusion Matrix

CNN ..... Convolutional Neural Network

CV ..... Cross Validation

CVRL ..... Contrastive Video Representation Learning

DAS ..... Driver Assistance Systems

EV ..... Ego Vehicle

GPU ..... Graphical Processing Unit

GPS ..... Global Positioning System

GRU ..... Gated Recurrent Unit

HMM ..... Hidden Markov Model

IMU ..... Inertial Measurement Unit

LiDAR ..... Light Detection and Ranging

LSTM ..... Long Short-Term Memory

MLP ..... Multi Layer Perceptron

NCE ..... Noise-contrastive Estimation

OOBN ..... Object-oriented Bayesian Networks

RGB ..... Red Green Blue

ROI . . . . . Region of Interest

RNN . . . . . Recurrent Neural Network

RMSE . . . . . Root Mean Squared Error

SAE . . . . . The Society of Automotive Engineers

TV . . . . . Target Vehicle

TET . . . . . Temporal Elastic Transformation

VGMM . . . . . Variational Gaussian Mixture Model

CHAPTER 1

INTRODUCTION

Driver assistance systems (DAS) are a group of hardware and software technologies that assist drivers while driving, parking, and even not using the vehicle with statistical insights. They use sensors like cameras, radars, and LiDARs to detect nearby obstacles or driver errors and respond accordingly. As the automotive industry evolved, DAS started to be defined as Advanced Driver Assistance Systems (ADAS). ADAS are designed to improve safety and reduce the number of accidents caused by human error. They can also help drivers to be more aware of their surroundings and to drive more safely. In the last twenty years, ADAS features developed to enable various levels of autonomous driving, depending on the features installed in the car. Some of the most common ADAS features include:

- Adaptive cruise control: An automated system designed to regulate a vehicle's speed, maintaining a safe following distance from the vehicle positioned ahead.
- Automatic emergency braking: A system equipped with the capability to automatically engage the brakes if a pending collision with another vehicle or an unidentified object is detected.
- Blind spot monitoring: A system that utilizes sensors, such as radars, to identify vehicles within the driver's blind spots, subsequently alerting the driver with either visual or audible warnings.
- Collision avoidance system (a.k.a. pre-crash system): A system that employs radar and camera sensors to preemptively discern the environment's condition and respond appropriately. This could involve sounding an alarm, tightening the passengers' seat belts, closing the sunroof, or elevating either side of the vehicle.
- Lane departure warning: This system employs cameras to ascertain whether the vehicle is on the verge of drifting from its lane, alerting the driver with either visual or audible warnings in such instances.
- Park assist: A system that is equipped with the ability to park the vehicle automatically in both parallel and perpendicular parking spaces.

ADAS are becoming increasingly common in new cars. Many automakers are now offering ADAS as standard equipment on their vehicles. Governments have even

mandated some ADAS features on new vehicles in some countries. As ADAS technology develops, it will likely play an even more significant role in improving road safety and creating autonomous vehicles.

Recognizing the expanding role of ADAS in enhancing road safety and paving the way for autonomous vehicles, The Society of Automotive Engineers (SAE) has proposed six distinctive levels based on the extent of automation in ADAS (SAEJ3016 (2021)). Level 0 represents a stage where the driver is in complete control of the vehicle. ADAS merely provide informative cues for the driver to interpret and act upon. Examples of ADAS belonging to Level 0 include parking sensors, surround-view, traffic sign recognition, lane departure warning, night vision, blind spot information system, rear-cross traffic alert, and forward-collision warning. In both Levels 1 and 2, the majority of decision-making rests with the driver. The divergent feature is that Level 1 ADAS can seize control over a single function, whereas Level 2 systems can supervise multiple functions, thereby providing support to the driver. Features such as Adaptive cruise control, emergency brake assist, automatic emergency braking, lane keeping, and lane centering are examples of Level 1 ADAS features. On the other hand, highway assist, autonomous obstacle avoidance, and autonomous parking are considered Level 2. From Level 3 onwards, the degree of vehicle control incrementally escalates, topping in Level 5, where the vehicle demonstrates full autonomy. Some of these systems, however, are yet to be fully integrated into commercial vehicles. For instance, highway chauffeur (a Level 3 system) and automated valet parking (a Level 4 system) were not in widespread commercial use as of 2019 (Galvani (2019)). The levels can be intuitively conceptualized as: Level 0 - no automation; Level 1 - hands-on/shared control; Level 2 - hands-off; Level 3 - eyes-off; Level 4 - mind-off, and Level 5 - steering wheel optional. Figure 1.1. can be examined for a more detailed explanation of SAE Automation Levels.

As we comprehend the stages of automation defined by the Society of Automotive Engineers, we recognize the vital role of ADAS, not just in providing assistance but also in ensuring safety. However, as we progress from level 0 to level 5 automation, the complexity of vehicle safety mechanisms increases largely.

This complexity has started to diminish with the advent of technological advancements in radar, LiDAR, and camera-based sensing systems. The evolution of imaging technology has been remarkable, with cameras becoming more affordable, compact, and high-quality. Simultaneously, there has been a significant surge in computing power with the advent of platforms designed for parallel processing, such as multi-core processing and graphical processing units (GPUs). These advancements have paved the way for the real-time implementation of computer vision approaches for vehicle detection, tracking, and maneuver analysis.

The rapid progress in camera sensing and computational technologies has produced extensive research in vehicle detection using monocular vision, stereo vision, and sensor

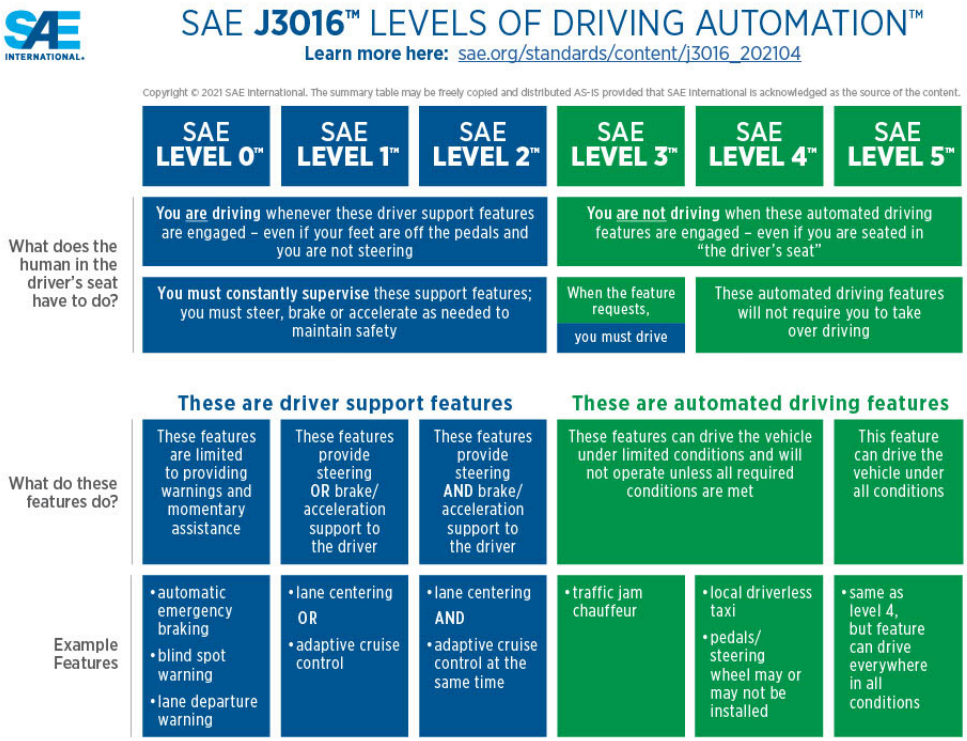

Figure 1.1. Detailed explanation of SAE Levels of Driving Automation (Source: SAEJ3016 (2021)).

fusion with vision within the autonomous vehicles community (Sivaraman and Trivedi (2013)). On-road vehicle tracking has also been a major focus of study (Barth and Franke (2010), Sivaraman and Trivedi (2011)), with many research efforts reporting the ability to reliably detect and track on-road vehicles in real time over extended periods.

These theoretical, practical, and algorithmic advancements have opened up new research opportunities that seek a higher level of semantic interpretation of on-road vehicle behavior. The integration of this spatiotemporal information from vehicle detection and tracking can be used to identify maneuvers and to learn, model, and classify on-road behavior.

As depicted in Figure 1.2., vision plays a crucial role in on-road interpretation. At the most basic level, various motion and appearance cues are used for on-road vehicle detection. The next level involves associating detected vehicles across frames to enable vehicle tracking, which measures the dynamics of the motion of detected vehicles. At the highest level, an aggregate of spatiotemporal features allows for the characterization of vehicle behavior, recognition of specific maneuvers, behavior classification, and long-term motion prediction. This emerging area of research includes predicting turning behavior (Barth and Franke (2010)), predicting lane changes (Kasper et al. (2012)), and modeling typical on-road behavior (Sivaraman et al. (2011)).

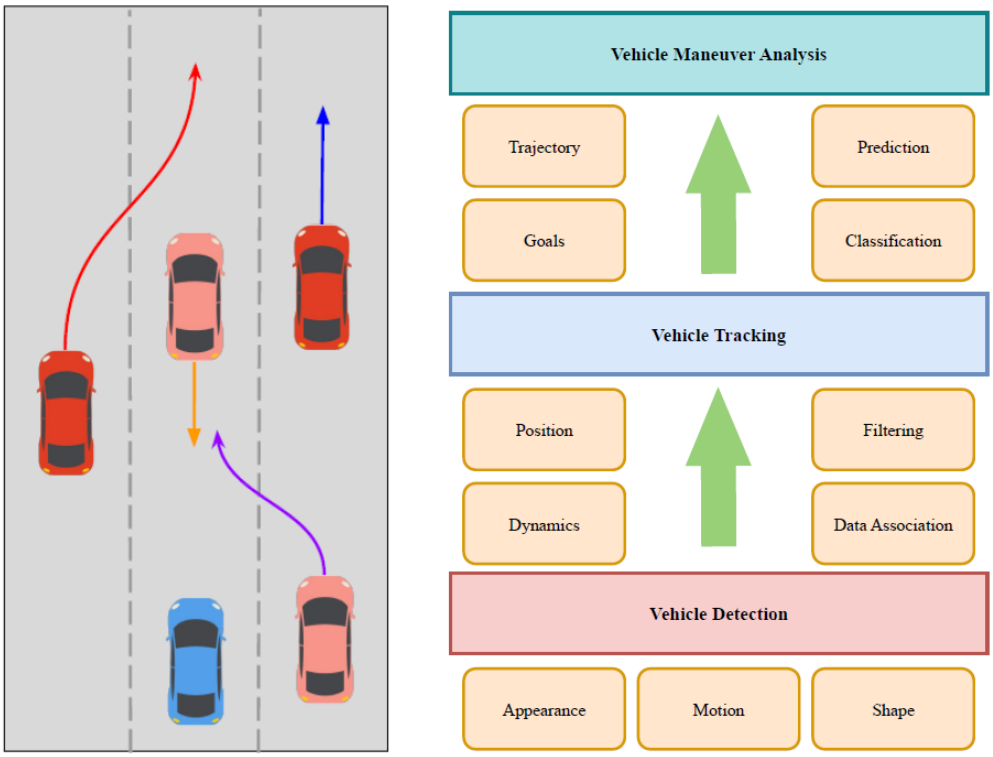

Figure 1.2. The figure illustrates the hierarchical levels of vision for semantic interpretation of the on-road environment. At the base level, features such as appearance, motion, and size are utilized for vehicle detection in images or videos. At the next level, techniques such as data association, temporal coherence, and filtering are employed for tracking vehicles. This allows for the reidentification of vehicles, measurement of dynamic parameters, and estimation of vehicle positions. At the final step of this hierarchy, a combination of spatiotemporal features is leveraged to learn, model, classify, and predict the maneuvers of other vehicles on the road. (Source: Sivaraman and Trivedi (2013)).

Having established the importance and role of vision in on-road interpretation, it's crucial to delve into specific challenges that arise in the context of ADAS features. One such challenge that stands out is the detection and analysis of lane change maneuvers, particularly the cut-in maneuvers, which have been identified as one of the foremost causes of fatal accidents according to a report published by the U.S. Department of Transportation National Highway Traffic Safety Administration in 2021 (III (2019)). The report indicates that in the USA during 2020, %6.8 of fatal crashes were caused by the failure to yield, while failure to change to a different lane caused %6.2. Therefore, research in detecting intended maneuvers of nearby vehicles is essential because it can reduce the number of fatal accidents caused by hazardous lane change maneuvers. By developing methods to detect these maneuvers, both drivers and autonomous vehicles can take evasive action to avoid collisions.

Detecting lane change maneuvers can be achieved through various methods. These methods use sensors such as camera, radar, and LiDAR to understand the environment

Figure 1.3. Most used sensor technologies for autonomous driving applications.

(Figure 1.3.). We can summarize the use of these sensors as follows and the comparison of the sensors according to their sensing performance is given in Table 1.1.:

- Camera: Cameras can be used for tasks such as identifying objects in the environment, getting depth-based information, and tracking their movement based on previous frames.
- Radar: These sensors use radio waves to detect nearby objects. They can be used to measure the distance, speed, and direction of objects.
- LiDAR: LiDAR stands for "Light Detection and Ranging" so that it uses lasers to create a 3D map of the surrounding environment. Utilizing 3D maps is very effective in identifying and tracking objects.

Once the surrounding vehicles have been detected by the use of any one or more of these methods, different cues can be used to recognize the future maneuver of each vehicle, such as the speed of the vehicle, the distance between the vehicle and the other vehicles in the lane, vehicle blinker in on or off, the presence of any obstacles in the road, etc. Each of those cues can be used as input by different techniques like rule-based, probabilistic reasoning, and deep learning methods.

Table 1.1. Comparison of sensors on environment sensing (Source: Song and Li (2022)).

Criteria	**Camera**	**Radar**	**LiDAR**
Short range detection (0-1m)	Medium	Medium	Medium
Mid range detection (1-30m)	Good	Good	Good
Long range detection (>30m)	Poor	Medium	Good
Angular accuracy	Good	Medium	Good
Velocity accuracy	Poor	Good	Poor
Distance accuracy	Poor	Poor	Good
Operation in adverse weather	Poor	Good	Poor
Operation at night	Poor	Good	Good
Delay	Good	Good	Good

Rule-based systems can be used to define a set of rules that can be used to identify lane change maneuvers. These rules can be based on expert knowledge or on data that has been collected from real-world driving situations, as previously stated. Probabilistic reasoning techniques can be used to calculate the probability that a lane change is likely to be happened by considering the environmental cues.

After deep learning has become popular due to its effectiveness in numerous computer vision and natural language processing tasks (Tekir and Bastanlar (2020)), naturally researchers proposed deep neural network models to identify lane change maneuvers. These models are trained on a large dataset of videos or sensor data that includes examples of both lane-changing and non-lane-changing maneuvers.

The development of methods to detect lane change maneuvers is an active area of research. Several different approaches are being explored, and a combination of approaches will likely be needed to achieve the desired level of accuracy and reliability. There are many reasons why it is hard to develop methods to detect hazardous lane change maneuvers. One of the reasons that the road environment is full of different objects and conditions that can make it difficult to track the movement of vehicles. For example, other vehicles, pedestrians, cyclists, and buildings can all block the view of the road, making it difficult to see where other vehicles are and what they are doing. The weather can also make it difficult to see, such as rain, snow, and fog. Secondly, vehicles can change lanes in a variety of different ways. There is no single way that vehicles change lanes. Some vehicles may signal their intent to change lanes, while others may not. Some vehicles may change lanes quickly, while others may change lanes slowly. Additionally, vehicles may change lanes for a variety of reasons, such as to avoid an obstacle or to get to their destination faster. Lastly, the accuracy of the methods needs to be high. Even a tiny error detecting a lane change maneuver could lead to a collision. This is why it is important to develop highly accurate methods.

1.1 Contributions of the Thesis

The contribution of this thesis work can be summarized below:

- Due to the absence of a benchmark dataset for the classification of potentially dangerous cut-in maneuvers in traffic, two distinct cut-in/lane-pass classification datasets have been created from the publicly available Berkeley Deep Drive (BDD) dataset (Yu et al. (2020)) and Prevention dataset (Izquierdo et al. (2019b)).
- A new scene representation has been presented to be used in the classification of vehicle maneuvers, where the vehicle(s) and ego lane information are highlighted,

which is obtained from videos collected from the front camera of the vehicle.

- A new method utilizing a coordinate-based LSTM network has been introduced for the classification of cut-in/lane-pass maneuvers. This identical approach is also applied to the detection of lane change maneuvers (Nalcakan and Bastanlar (2022)).
- A self-supervised contrastive video representation learning approach has been introduced and evaluated on the cut-in/lane-pass dataset and the lane change classification dataset. This represents the first application of contrastive learning in the task of maneuver classification (Nalcakan and Bastanlar (2023)).
- Ensemble learning techniques are applied to two deep learning approaches proposed in this thesis work, and the performance evaluation has been done on the lane change prediction task.

1.2 Organization of the Thesis

The thesis is organized into seven distinct chapters. Chapter 2 lays the foundation with a problem background and literature review on trajectory-based and vision-based maneuver classification. Chapter 3 provides information on datasets used in the thesis, the BDD-100K Dataset and the Prevention Dataset, detailing their acquisition and labeling process. In Chapter 4, a monocular vision-based prediction model of cut-in maneuvers utilizing Long Short-Term Memory (LSTM) networks is explained. This chapter breaks down the model's methodology, the vehicle detection and tracking process, the feature extraction and network architecture, and experimental results. Chapter 5 introduces a novel approach to detecting lane change and cut-in maneuvers with self-supervised contrastive video representation learning (CVRL), explaining the methodology, the concepts of contrastive learning, scene and maneuver representation, video encoder, and experimental results, along with an ablation study. In Chapter 6, ensemble learning is applied for lane change maneuver detection, weaving together the methods proposed in Chapters 4 and 5. Finally, the conclusion summarizes the insights gathered during this thesis work and explores potential future directions.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

Lane change maneuvers (Figure 2.1.) represent an essential aspect of vehicle dynamics and safe driving. They involve a vehicle moving from its current lane to an adjacent one on multi-lane roadways. This maneuver is typically performed when a driver intends to overtake another vehicle, prepare for a turn at an intersection, or adjust their position due to road conditions. Executing a lane change maneuver requires a clear understanding of the surrounding traffic scenario, including nearby vehicles' speed, distance, and trajectory. Therefore, it is a complex task that involves several stages: the decision to change lanes, checking for a safe gap in the target lane, steering to move into the target lane, and finally stabilizing in the new lane. Detecting and predicting lane change maneuvers is crucial for driver assistance systems, as they need to react accordingly to ensure safe and smooth driving.

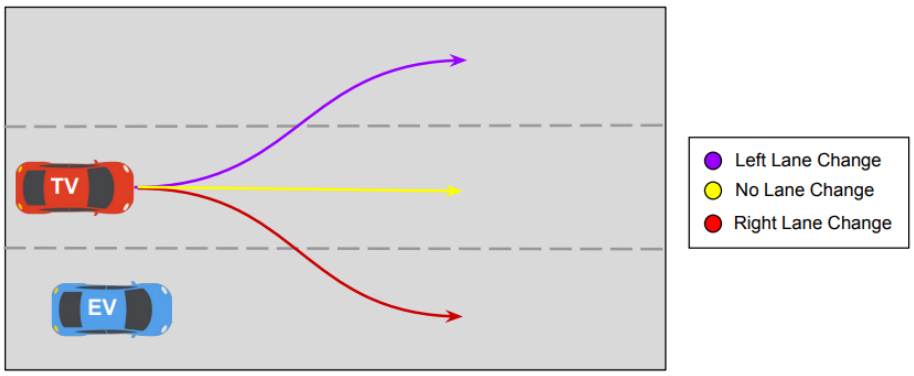

Figure 2.1. Lane change maneuvers representation. (EV: ego vehicle, TV: target vehicle.)

Moreover, a cut-in maneuver (Figure 2.2.) is a specific type of lane change maneuver that occurs when a vehicle moves into the lane ahead of the ego vehicle. This maneuver is particularly challenging for the DAS because it often happens suddenly, leaving little time for the ego vehicle driver to react. The complexity originates from the need to rapidly adjust speed or change lanes to avoid a potential collision. Various conditions, such as traffic congestion, road construction, or erratic driving behavior, can provoke cut-in maneuvers. Due to the sudden nature and the potential risks associated with cut-in maneuvers, accurately detecting and predicting them and responding accordingly is crucial

for the safety of both autonomous and human-operated vehicles. As such, the ability to detect and predict cut-in maneuvers in real-time forms a significant part of the research in ADAS.

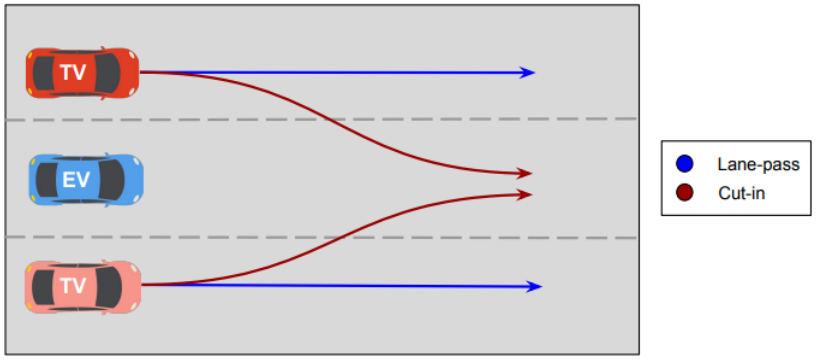

Figure 2.2. Lane-pass and cut-in maneuvers representation. (EV: ego vehicle, TV: target vehicle.)

There are several approaches to tackle those challenges, from traditional physics-based methods to modern machine learning techniques. Physics-based methods often utilize physical models of the driving environment and heuristics derived from human driving behavior. However, their performance is limited in complex and dynamic traffic scenarios due to their inability to adapt to unexpected situations. On the other hand, machine learning methods, especially deep learning, have shown great promise due to their ability to learn complex patterns with the help of large amounts of data.

Trajectory-based classification models represent a significant methodological approach within this realm. These models focus on the path, speed, and direction of vehicles, filtering a series of complex movements into simpler, more easily interpretable data. By using techniques like pattern recognition, these models aim to predict future actions based on past behavior. While these models have shown usefulness in certain scenarios, they struggle in more complex or unpredictable traffic situations. The accuracy of trajectory-based models relies heavily on the consistency of driving behavior, which can be influenced by various factors such as road conditions, driver distraction, or sudden events. Therefore, there is a need for more advanced and adaptable trajectory-based models that can handle such complexities and uncertainties.

Vision-based classification models offer an alternative approach that takes advantage of the information available in visual data. Using computer vision and deep learning techniques, these models can analyze images or videos to detect and classify maneuvers. They can capture complex visual details such as the orientation of the vehicle's wheels,

the vehicle's head direction, or the vehicle's position relative to the lane markings, which can provide valuable clues about the vehicle's intended maneuvers. Moreover, vision-based models can detect and respond to unexpected events or changes in the environment that are not easily captured by trajectory-based models. However, these models also face challenges, such as the need for large amounts of labeled data, the difficulty of handling diverse lighting and weather conditions, and the computational complexity of image analysis. Therefore, improving the robustness and efficiency of vision-based classification models is an important research direction.

Consequently, the review of literature on maneuver classification research was organized into two categories according to their respective methodologies: trajectory-based and vision-based.

2.1 Trajectory-based Maneuver Classification

While traditional physics-based methods have proven satisfactory for simple environments and short-term prediction tasks, their performance weakens in the face of more complex scenarios. An increased focus has been recently directed toward deep learning-based trajectory prediction methodologies, mainly due to their broader applicability. These modern techniques hold the capacity to not only account for physics-related and road-related factors - areas inherently catered to by conventional methods - but also incorporate interaction-related aspects. This inherent flexibility makes them notably suited to handle more complex prediction environments. To illustrate these advanced methods, an overview is provided in Figure 2.3..

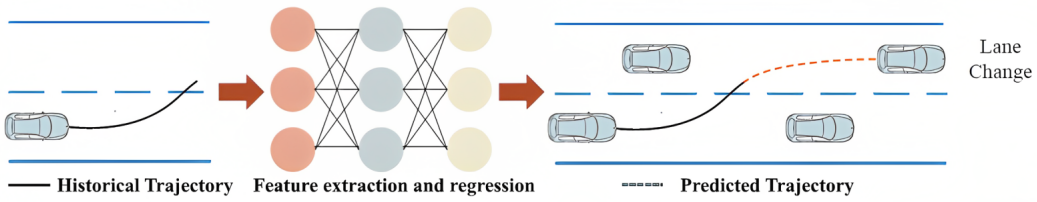

Figure 2.3. Overview of deep learning-based trajectory-based maneuver classification methods (Source: Huang et al. (2022)).

RNNs diverge from traditional machine learning-based methodologies and CNNs in their ability to handle temporal information effectively (Graves (2013), Sutskever et al. (2014)). RNNs retain information from preceding time steps and collaborate these data

with the input to generate the output. One practical challenge with RNNs is the propensity for gradients to diminish or explode when the number of time steps is substantial. The introduction of gated RNNs, such as Long Short-Term Memory Network (LSTM) and Gated Recurrent Unit (GRU), has been instrumental in mitigating this issue.

Several studies have employed Long Short-Term Memory (LSTM) as a sequence classifier to predict vehicle maneuvers, as demonstrated in the works of Zyner et al. (2018), Phillips et al. (2017) and Zyner et al. (2017). The process involves feature extraction performed by LSTM cells, with the last cell's hidden states channeled into the output layer, aiding in the prediction of maneuvers. Altché and de La Fortelle (2017) utilized a single-layer LSTM to anticipate the target vehicle's trajectory. An LSTM encoder architecture used for maneuver-based trajectory prediction was proposed by Ding and Shen (2019), which encodes the states of the target vehicle and combines the predicted maneuver with map information. They employed nonlinear optimization methods to refine the initial future trajectory, incorporating interaction-related aspects, traffic regulations, and map data. Zyner et al. (2019) employed a weighted Gaussian Mixture Model (GMM) to predict multi-modal trajectories, using an encoder-decoder three-layer LSTM to obtain the model parameters. This was followed by trajectory clustering based on the mode with the highest probability. Deo et al. (2018) recommended the use of the hidden Markov model (HMM) and the variational Gaussian mixture model (VGMM) for maneuver trajectory classification. Initially, they collected data on various maneuvers such as lane-pass, overtake, cut-in, and drift-into-ego-lane from highway cameras, radar, and LiDAR data. Subsequently, they categorized all trajectories into maneuvers using HMM and VGMM. Again, Deo and Trivedi (2018) utilized an LSTM encoder to extract temporal information of the surrounding vehicles, which was then used as an input to a social pooling layer (Alahi et al. (2016)) to form a 'social tensor'. The social pooling layer in this study captures interaction-related factors between vehicles following spatial rasterization. The social tensor is then passed to a set of CNNs to discern the spatial correlation of vehicles. Finally, six LSTM decoders are employed to generate distributions for six specific maneuvers, comprising three lateral (left lane change, right lane change, and keep lane) and two longitudinal (brake, normal speed) maneuvers.

Wang et al. (2021) used publicly available NGSIM dataset (US101 (2007)) as input to an LSTM following a fully connected layer. They converted the target vehicle's (x, y) coordinates, velocity, and acceleration to a feature vector accompanied by the availability in the right lane and left lane to classify future maneuvers. Similarly, Chen et al. (2022) used the same data but contrary to previous approaches, they extracted features from raw trajectory data with a multi-layer perceptron (MLP) than those features fed into a convolutional social LSTM (CS-LSTM) (Deo and Trivedi (2018)) to classify maneuver. In a different study, Scheel et al. (2019) fed the trajectories of the right lane change, left lane change, and follow maneuvers into an attention-based LSTM network, reporting

individual maneuver prediction accuracies of 0.784, 0.962, and 0.679 for left lane change, follow, and right lane change, respectively. In a different work, an encoder-decoder LSTM architecture was adopted by Park et al. (2018), which encoded historical trajectory features and determined the most probable future trajectories through the beam search algorithm. A dual LSTM network model for predicting the target vehicle's trajectory was suggested by Dai et al. (2019), with one network dedicated to modeling the trajectories of surrounding vehicles and the other focusing on the interaction among these vehicles. Xin et al. (2018) utilized an LSTM to predict the target vehicle's lane and employed a separate LSTM to predict the trajectory based on the vehicle's state and the predicted lane. An LSTM-based framework that integrates intention and trajectory predictions was proposed by Zhang et al. (2021). The vehicle's intention at intersections is predicted by one LSTM model, while the trajectory is predicted by another LSTM-based model utilizing previous trajectories. Zhang et al. (2020) utilized Temporal Convolutional Network (TCN) to predict the lane-change maneuver and trajectories. They used steering wheel angle α to represent lane-changing behavior and x,y coordinates of the vehicles as the vehicle's trajectory history.

There are studies in the literature that propose methods focusing solely on classifying the cut-in maneuver among trajectory-based maneuver classification studies. For instance, in a work of Chen et al. (2019), a unique control strategy for trajectory tracking in cut-in scenarios is proposed, employing an RNN with LSTM cells for driver behavior prediction and a Model Predictive Control (MPC) approach for tracking the reference trajectory, considering different cut-in behaviors. Another study Jeong and Yi (2020) introduces an interactive motion predictor utilizing a Bidirectional-LSTM module to distinguish the intentions of cut-in vehicles. The system, trained and validated on a robust set of human-driven vehicle trajectories, comprises three key components: maneuver recognition, trajectory prediction, and an interaction module, collectively working to predict future trajectories of surrounding vehicles based on potential maneuvers and collision risks. In recent work, Yoon et al. (2021) proposes a probabilistic trajectory prediction method for cut-in vehicles, which employs Gaussian Process Regression (GPR) to obtain a probability distribution of behavioral parameters from real-world vehicle trajectories. This predictive approach is integrated into the motion planning and control of autonomous vehicles, and evaluations show that it improves prediction accuracy and ride quality in multi-vehicle cut-in scenarios, whilst maintaining safety. The comprehensive review of mentioned studies is given in Table 2.1..

Table 2.1. Comparison of the trajectory-based maneuver classification studies.

Reference	Method	Features	Classification
Altché and de La Fortelle (2017)	LSTM	lat_{pos}, $long_{pos}$, lat_v, $long_v$	Lane
Deo and Trivedi (2018)	CS-LSTM	*trajectory history*	KL, LLC, RLC, B
Deo et al. (2018)	HMM+VGMM	*trajectory history*	LP, C, O, $Drift_F$, $Drift_B$
Xin et al. (2018)	Dual-LSTM	lat_{pos}, $long_{pos}$, lat_v, $long_v$	Lane
Park et al. (2018)	encoder-decoder LSTM	ego_V, ego_ψ, rel_{coord}, rel_V	Lane(Occ. Grid)
Ding and Shen (2019)	encoder-decoder RNN	*trajectory history*	Lane
Zyner et al. (2019)	RNN	*trajectory history*	LT, RT, S
Scheel et al. (2019)	attention LSTM	lat_{pos}, $long_{pos}$, lat_v, $long_v$, lane, d_{right}, d_{left}	LLC, RLC, F
Chen et al. (2019)	LSTM	*trajectory history*	LC, RC, S
Dai et al. (2019)	dual LSTM	*trajectory history*	Lane
Zhang et al. (2020)	TCN	lat_{pos}, $long_{pos}$, $steer_\alpha$	NLC, LLC, RLC
Laimona et al. (2020)	LSTM	lat_{pos}, $long_{pos}$	NLC, LLC, RLC
Wang et al. (2021)	LSTM	F_{fuzzy}	LK, LLC, RLC
Yoon et al. (2021)	GPR+EKF	lat_{offset}, $head_{offset}$, $long_v$, lat_{pos}, $long_{pos}$	LK, LC, RC
Zhang et al. (2021)	encoder-decoder LSTM	*trajectory history*	S, LLC, RLC
Chen et al. (2022)	MLP + CS-LSTM	*trajectory history*	Lane

Features: (lat_{pos}: lateral position, $long_{pos}$: longitudinal position, lat_v: lateral velocity, $long_v$: longitudinal velocity, ego_V: ego vehicle velocity, ego_ψ: ego vehicle heading angle, rel_{coord}: relative coordinates, rel_V: relative velocity, $steer_\alpha$: steering wheel angle, F_{fuzzy}: fuzzy logic features, lat_{offset}: lateral offset, $head_{offset}$: heading offset),
Classifications: (LP: lane-pass, C: cut-in, O: overtake, $Drift_F$: drift into the front, $Drift_B$: drift into the behind, KL: keep-lane, LLC: left lane change, RLC: right lane change, B: braking, S: straight, F: follow, RT: right turn, LT: left turn, LC: left cut-in, RC: right cut-in, NLC: no lane change)

2.2 Vision-based Maneuver Classification

The surge in deep learning applications, particularly in vision, has led recent research in vision-based maneuver classification to predominantly employ convolutional neural networks (CNNs) for capturing the scene's visual information. The typical methodology involves using a CNN as a feature extractor, where video frames serve as input, coupled with an RNN or LSTM for classification tasks. In a study by Izquierdo et al. (2019a) investigates two deep learning methodologies for predicting vehicle lane changes. The first technique in the study adopts a multi-channel representation of temporal data by mapping the scene appearance, target vehicle motion history, and surrounding vehicles' motion histories to the red, blue, and green channels respectively, which is then passed as input to a CNN model. The second approach blends CNN and LSTM to encapsulate temporal characteristics, with both methodologies aiming to embed local and global contexts with temporal insights to forecast lane change intentions. Another study (Fernández-Llorca et al. (2020)), created a method that initially crops Regions of Interest (ROIs) from the original frames and exploits two modes of input video, namely the high frame rate video and its optical flows. This approach facilitated a comparison between two-stream CNNs and spatio-temporal multiplier networks. A subsequent study by the same team (Biparva et al. (2021)) broadened this comparison by incorporating a slow-fast network which is an approach that utilizes videos of both high and low frame rates. This addition reportedly improved performance, indicating a slight edge over the previously evaluated alternatives. Differently, Laimona et al. (2020) focused on comparing the performance of RNN and LSTM on the same lane change prediction dataset, using only the center coordinates of the target vehicle that are extracted from target vehicle' segmentation mask as features. The authors observed that the LSTM was able to achieve commendable results in the classification of lane change maneuvers, despite the feature set being relatively small. Their findings underlined the efficacy of LSTM networks in such applications.

While the aforementioned studies relied on the Prevention dataset (Izquierdo et al. (2019b)) for their method training and validation, some researchers have used different datasets for vision-based maneuver classification. For instance, Lee et al. (2017) proposed a methodology suitable for adaptive cruise control, leveraging front-facing radar and camera outputs to detect lane change maneuvers. Their strategy involved converting traffic scenes into simplified bird's eye view (SBEV) images, which were fed into a CNN for predicting lane keeping, right cut-in, and left cut-in intentions. Similarly, Yurtsever et al. (2019) presented a deep learning framework to classify potentially hazardous lane change behavior using in-car camera video footage. They employed a pre-trained Mask R-CNN model for segmenting vehicles in the scene and a CNN+LSTM model for categorizing the behaviors as either dangerous or safe. The work of Simoncini et al. (2022) proposed

a framework for unsafe maneuver classification. They used dashcam video data to train a ResNet50 model and GPS/IMU sensor data to extract position-based features. All features process through a novel spatio-temporal attention selector and fully connected layer for classification results. Peng et al. (2018) proposed a Driver Maneuvering Detection (DMD) system which considers right and left turn maneuvers and straight motion. The DMD system integrates a distance-based context representation, vehicle trajectory features, and VGG-19 network features from front-view video images. Additionally, it employs an LSTM-based neural network model to understand the temporal sequences in driving maneuvers. This system presents a unique blend of traditional and deep learning methods to enhance maneuver detection accuracy. The comprehensive review of mentioned studies is given in Table 2.2..

Table 2.2. Comparison of the vision-based maneuver classification studies.

Reference	Method	Features	Classification
Lee et al. (2017)	CNN	SBEV	LC
Peng et al. (2018)	VGG19+LSTM	V, L	TL, TR, S
Izquierdo et al. (2019a)	LeNet+LSTM	V	NLC, LLC, RLC
Yurtsever et al. (2019)	CNN+LSTM*	V	SF, R
Fernández-Llorca et al. (2020)	ST-CNN*	V, OF	NLC, LLC, RLC
Biparva et al. (2021)	Two Stream CNN*	V	NLC, LLC, RLC
Simoncini et al. (2022)	ResNet50 + MHA	V, L	SF, USF

Method: (MHA: multi-head attention, ST: spatio-temporal), **Features:** (V: video frames, OF: optical flow, L: location, SBEV: simplified bird's-eye view), **Classification:** (NLC: no lane change, LLC: left lane change, RLC: right lane change, LC: lane change, TL: turn left, TR: turn right, S: go straight, SF: safe, USF: unsafe, R: risky), *multiple CNN architectures evaluated.

Over the course of this thesis, three distinct yet interconnected approaches to maneuver classification were formulated and implemented.

- In the first approach, the focus was placed on the coordinate features of the target vehicle. Long Short-Term Memory (LSTM) networks were used for the classification task of their capability to handle sequential data. Three different LSTM network variants were experimented with to ensure a comprehensive evaluation.

- The second approach was based on extracting the rich spatial-temporal information embedded in video frames. These frames were given as input to a 3D variant of ResNet18, a deep Residual Neural Network that can process sequential data, which is videos in our case. A two-stage training process was employed with this network. The initial stage involved self-supervised contrastive learning, a technique

that captures intricate data patterns without supervision. The network was then fine-tuned with labeled data in the second stage, enhancing its maneuver classification capabilities.

- The third approach involved the application of ensemble learning techniques to combine the strengths of the best-performing methods from the first two approaches. The resulting model benefited from the individual strengths of each component method, creating a robust maneuver classification system.

Each of these methodologies, developed within the context of this thesis, offered a comprehensive and in-depth exploration of maneuver classification, contributing significantly to the literature in the field of advanced driving assistance systems.

CHAPTER 3

DATA SETS

While developing a machine learning-based solution for maneuver classification, we researchers need a wide collection of samples. The datasets used for maneuver classification fall into two categories. The first category includes nonpublic datasets, which have been collected by researchers or by companies, and remain inaccessible to the research community. On the other hand, the second category contains publicly shared datasets, offering more opportunities for comparative studies and the development of universally applicable solutions. For a comprehensive understanding of the open-source datasets leveraged for maneuver classification and the annotations they provide, refer to Table 3.1..

3.1 BDD-100K Cut-in/Lane-pass subset and Labeling

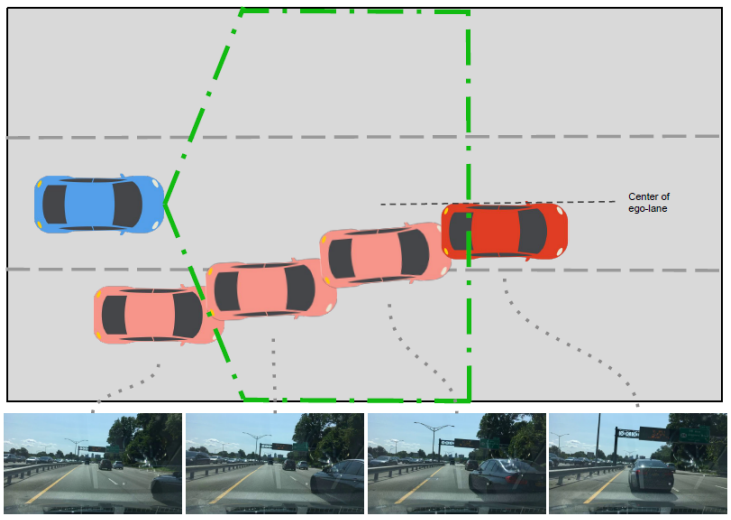

Figure 3.1. Start and end points of maneuvers that are labeled as cut-in.

A dedicated subset was crafted to tackle the unique challenges associated with cut-in and lane pass classification from the widely-used Berkeley Deep Drive Dataset (BDD-100K). This dataset has a collection of 100,000 driving videos, each labeled across ten diverse tasks ranging from road object detection to driveable area identification. The

Table 3.1. Comprehensive review of publicly available datasets for maneuver classification (Source: Fang et al. (2022)).

Dataset	# of Samples	Annotations	Labels	Task
BDD-100K Yu et al. (2020)	100000	L, 3DB, VT, V, W, SM, SD	10 types	TP
VIENA2 Aliakbarian et al. (2018)	15000	I, B	S, TL, TR, LC, C, NC	AP
INTERACTION Zhan et al. (2019)	40054	I, L, SM, B	RD, NC, LC, M	MC, TP
BLVD Xue et al. (2019)	654	I, L, T, B, 3DB	22 types	MC, FLP, TP
PREVENTION Izquierdo et al. (2019b)	11*	I, L, VT, B	LC	MC, FLP
LOKI Girase et al. (2021)	664	DES, I, L, 3DB, SM, W	14 types	TP, MC, FLP
DADA-2000 Fang et al. (2021)	2000	I, DA, B	C, LC, VO, NC	AP, MC, DAP

Annotations: (L: location, 3DB: 3D boxes, VT: vehicle type, V: velocity, T: trajectory, W: weather, B: behavior, DES: destination, SM: semantic map, SD: scene description), **Labels:** (C: cross, NC: non-cross, TL: turn left, TR: turn right, VO: vehicle overtake, LC: lane change, S: stop, M: merge, NC: near collision, RD: move along the roundabout), **Task:** (TP: trajectory prediction, AP: accident prediction, MC: maneuver classification, FLP: future location prediction, DAP: driver attention prediction), *given as driving records' number.

collected videos have varied driving conditions containing different times of the day and featuring data from several cities like New York, Berkeley, San Francisco, and Tel Aviv. From this extensive collection of videos, a subset of 875 sequences was extracted that contain cut-in and lane-pass maneuvers. These focused sequences were extracted from a collection of approximately 20,000 videos. The final distribution of this subset includes 405 cut-in samples and 470 lane-pass samples, offering a diverse range of examples for each maneuver type. The process of labeling these cut-in samples within our dataset is conducted by specific principles, as illustrated in Figure 3.1.. At the beginning of the sequence, the target vehicle is positioned in a separate lane, with no clear indicators of an upcoming cut-in. The critical lane change event (cut-in) is triggered when the target vehicle breaches the designated safety field, which is marked by green lines in the figure. The cut-in sequence is concluded as the vehicle fully enters the ego-lane, regardless of its alignment within the lane. For the lane-pass class, vehicles are labeled as they pass the ego-vehicle from either the right or left-hand side while they reside within the safety field. This approach to labeling provides a robust, well-annotated set of maneuvers ideal for the training and evaluation of the developed classification models.

3.2 Prevention Dataset and Cut-in/Lane-pass Subset Labeling

In 2019, Izquierdo et al. (2019b) from the University of Alacá published a benchmark dataset called "The PREVENTION dataset: a novel benchmark for PREdiction of VEhicles iNTentIONs" for lane change prediction task. The dataset has 356 hours of driving video which is recorded mostly on highways. They provide detections, trajectories, maneuver labels, and raw data. Continuous improvements are being made to the dataset according to Izquierdo et al. (2019b), but the current version offers only three labels for vehicle maneuvers: "left lane change", "right lane change", and "no lane change". The labeled maneuver distribution of the dataset is given in Table 3.2.. The addition of labels for left cut-in, right cut-in, cut-out, and hazardous classes are anticipated, but these labels are not yet included in the dataset. Formulation of their maneuver labeling approach is given in Figure 3.2..

Table 3.2. Labeled maneuvers distribution of lane change prediction dataset.

Label	**No. of samples**	**Average no. of frames**
No LC	3375	50.9
Left LC	218	96.8
Right LC	343	80.1

Two separate subsets were created from this dataset: a cut-in/lane-pass subset and a lane change detection subset. The number of samples utilized for lane change detection and cut-in detection tasks are detailed in Table 3.3.. The prepared subsets also have the same skewness since the original dataset's distribution of samples for the no lane change and lane change categories is imbalanced.

In the cut-in/lane-pass detection subset, sequences contain a minimum of 60 frames for compatibility with the BDD-100K cut-in/lane-pass dataset. The labeling of these sequences aligns with the process illustrated in Figure 3.1.. Meanwhile, for the lane change detection subset, sequences with at least 50 frames have been included, to maximize the quantity of samples. The lane change detection subset's label distribution is consistent with the studies that use the Prevention dataset as given in Figure 3.2..

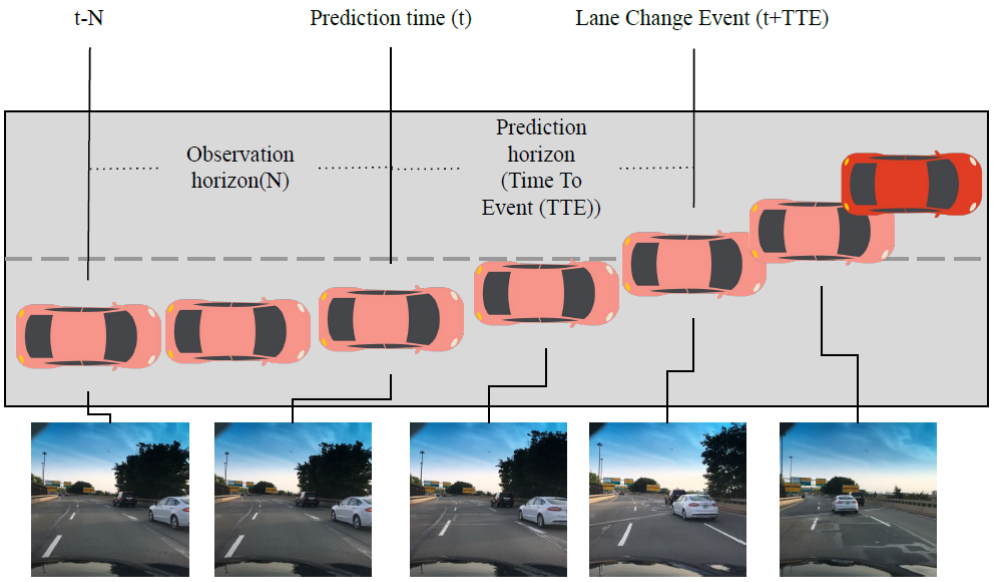

Figure 3.2. Start and end points of maneuvers that are labeled as a lane change.

Table 3.3. Sample distribution of both subsets on lane change detection and cut-in/lane-pass detection tasks created from the Prevention Dataset.

Task / Class	Lane Change Detection	Cut-in/Lane-pass Detection
No Lane Change	3375	-
Left Lane Change	218	-
Right Lane Change	343	-
Lane-pass	-	332
Cut-in	-	187

CHAPTER 4

Monocular Vision-based Prediction of Cut-in Maneuvers with LSTM Networks

4.1 Methodology

In this work, we developed a method aimed at detecting the cut-in maneuvers that could potentially pose a risk to the ego vehicle (Nalcakan and Bastanlar (2022)). Our strategy was rooted in computer vision, and we used videos collected from a single RGB camera installed in front of the vehicle. This setup allowed us to classify the maneuvers of the target vehicle based on the most recent video frames.

The pipeline is designed in two distinct stages: initially, two different CNN-based approaches are employed for vehicle detection and tracking. An LSTM-based strategy for the classification of maneuvers follows this. The focus is on maximizing computational efficiency, which leads to using a limited set of features for the classification stage rather than overloading the CNNs with RGB frames. Our research was centered on the vehicles ahead, and we utilized only a single forward-looking RGB camera installed within the vehicle. This approach offered a level of simplicity, distinguishing our work from other studies that used a combination of camera, radar, and LiDAR sensors.

The proposed methods have been evaluated using two different datasets: the BDD-100K and Prevention Cut-in/Lane-pass Classification subsets. These subsets were created as detailed in Chapters 3.1 and 3.2. The BDD-100K subset comprises 875 video clips, while the Prevention Dataset subset includes 519 video clips. Each clip contains vehicle maneuvers categorized as either cut-in or lane-pass and represents two seconds of action (30 frames per second). The number of frames used to represent this duration varies, including sequences of 15, 30, 45, or 60 frames. In addition to those datasets, the LSTM-3Class model was also evaluated using the Prevention Dataset for lane change detection. This dataset, which consists of 3936 video clips, provides a different set of data for comparison with the studies in the literature. Originally, the LSTM-3Class model operates with inputs of 15, 30, 45, and 60 frames. However, for this task, a maximum of 50 frames was included to align with previous works and retain more samples.

The methodology was structured in three steps. The first step involved using a CNN-based vehicle detection system, where we utilized YOLOv4 (Bochkovskiy et al.

(2020)) to recognize vehicles in each frame of the sequence. The second step involved tracking the detected vehicles using DeepSort (Wojke et al. (2017)). In the final step, we fed the features extracted from the detected and tracked bounding boxes of vehicles in front into an LSTM network for classification (Figure 4.1.).

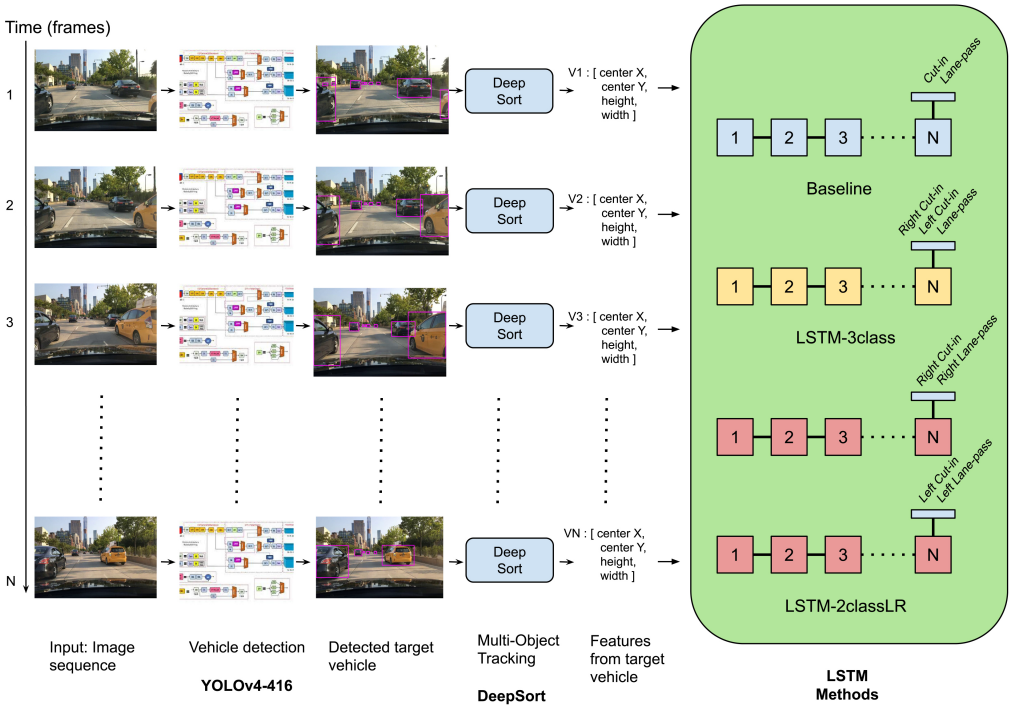

Figure 4.1. Pipeline of the proposed approach. Following the steps for extracting the bounding boxes of target vehicles (TVs), the baseline LSTM method uses feature vectors of TVs and classifies maneuvers as cut-in or lane-pass. LSTM-3class method classifies into 3: right cut-in, left cut-in, or lane-pass classes. As a third alternative, LSTM-2classLR has two separate LSTMs for left-hand side and right-hand side TVs. We conducted experiments with varying sequence lengths (15, 30, 45, and 60 frames) all representing two seconds of the video.

Our approach was designed to be computationally cost-effective compared to previous methods (Biparva et al. (2021); Fernández-Llorca et al. (2020); Izquierdo et al. (2019a); Yurtsever et al. (2019)) that fed CNNs with video frames and used complex architectures. We fed a limited number of features extracted from the input sequence into an LSTM, which allowed us to exceed the classification accuracies reported in other studies. Furthermore, our method was capable of producing a classification decision every two seconds.

4.1.1 Vehicle Detection & Tracking

Building upon the methodology we've outlined, the first crucial step in our research was the detection of vehicles. This was achieved using a Convolutional Neural Network (CNN), specifically the YOLOv4 model. This model was tasked with identifying vehicles in each frame of the video sequence under analysis. The decision to use a CNN-based approach for vehicle detection was strategic. CNNs have demonstrated exceptional performance in image recognition tasks, and their capacity to automatically learn spatial hierarchies of features made them an optimal choice for our process pipeline.

Once vehicles were detected, we tracked them using DeepSort, a high-quality tracking algorithm. This step was crucial as it allowed us to maintain continuity in vehicle identification across multiple frames, which was essential for the subsequent maneuver classification step.

Our approach to processing detected vehicle information was designed with computational efficiency in mind. Instead of overloading the CNN with RGB frames, we extracted a small set of features from the detected and tracked bounding boxes of vehicles. This approach not only reduced computational load but also provided us with the essential information needed for maneuver classification.

However, it's important to note that our vehicle detection approach was not without its challenges. The accuracy of detection and tracking was dependent on various factors, including the quality of the video footage, the lighting conditions, and the movement of the vehicles.

4.1.2 Feature Extraction and Network Architecture

Following the vehicle detection stage, we moved on to the feature extraction and network architecture phase of our research. This stage was integral to our methodology and was designed to be both efficient and effective.

In our approach, we separated from previous studies by obtaining features directly from the bounding box of the target vehicle. This was a strategic decision, designed to reduce computational load and streamline the process. Instead of feeding a complex CNN with video frames, we extracted a small number of features from the detected and tracked bounding boxes of vehicles. This approach was computationally cheaper and provided us with the essential information needed for maneuver classification.

Our network architecture was designed around an LSTM network (Figure 4.2.).

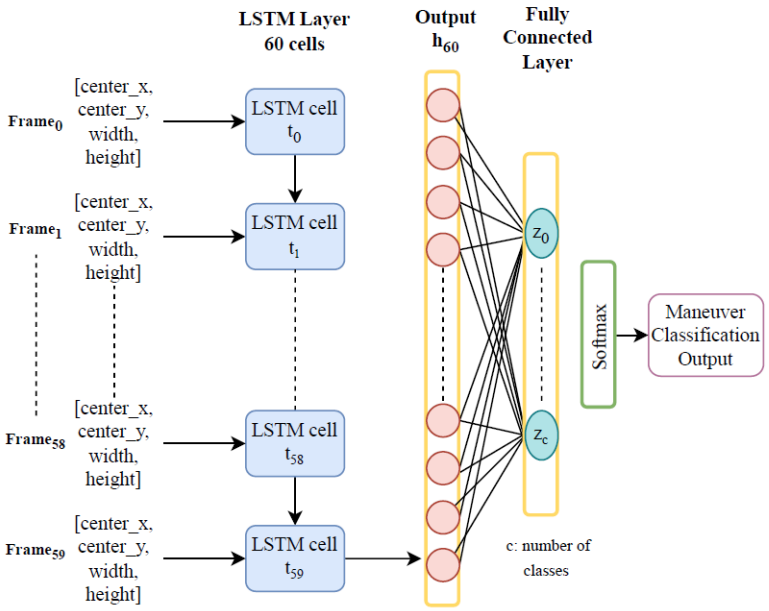

Figure 4.2. Utilized LSTM architecture.

The features we extracted from the bounding boxes of vehicles were fed into this LSTM network for classification. The LSTM network was designed to be simple yet effective. It consisted of a single layer and fewer parameters in the classification step, contributing to its computational efficiency. Despite its simplicity, the LSTM network was able to exceed the classification accuracies reported in the compared studies. Moreover, it was able to produce a classification decision every two seconds, demonstrating its effectiveness in real-time decision-making.

As hyperparameters of the LSTM, various hidden unit sizes, batch sizes, activation unit and optimizer types, and dropout rates are evaluated to find the best-performing LSTM architecture. Evaluated hyperparameters are given in Table 4.1..

Table 4.1. Evaluated LSTM hyperparameters

Hidden Units	**Batch Sizes**	**Optimizer**	**Activation**	**Dropout**
60	5	Adam	ReLU	0
128	10	RMSProp	Sigmoid	0.25
256	50	AdaDelta	Tanh	0.5
512	100			

4.2 Experimental Results

4.2.1 Classification Results on BDD-100K Cut-in/Lane-pass Classification Subset

We evaluated our approach with different methods where classification strategy varies. As a baseline method, a single layer LSTM model is trained with four features (center (x,y) coordinates, width and height of the target vehicle's bounding box) for a side-agnostic 2-class classification, i.e. each sample is a cut-in or a lane-pass. In LSTM-3class method, samples are classified as left-hand side cut-in, right-hand side cut-in and lane-pass. This second strategy is closer to several lane-change prediction studies in the literature, where maneuvers were classified as left lane-change, right lane-change and no lane-change. By examining the target vehicle's center coordinates, it is straightforward to extract if it is on the left or on the right of the ego vehicle. Thus, as a third method (LSTM-2classLR), we train two LSTM networks one responsible for left-hand side maneuvers and the other for the right-hand side, each performs a 2-class classification (cut-in/lane-pass). Test performances of these 3 methods are presented in Table 4.2..

Since most previous studies reported accuracy, precision, and recall, our evaluation is based on these metrics. For all models mentioned above, hyperparameters (Table 4.1.) were optimized by grid search and the model with the highest accuracy on the validation set was evaluated on the test set.

Our baseline method (best-performing model parameters are: 128 hidden units, 5 as batch size, 0.25 dropout), which classifies the sequences without separating if the maneuver is on the right or on the left, reached an accuracy of 0.9256 with 30-frame and 60-frame sequences and slightly lower accuracy for other sequence lengths. Taking into account the side of the cut-in maneuver (3 classes: right cut-in, left cut-in, lane-pass) caused a very slight increase in the performance, achieving 0.9324 accuracy. However, when we train two separate networks for the right-hand side and left-hand side (LSTM-2classLR (best-performing model parameters are: 512 hidden units, 10 as batch size, 0.25 dropout)), the classification accuracy increases to 0.9585. The best values were obtained with 30-frame sequences, but for almost all other lengths accuracies were increased compared to the baseline and LSTM-3class methods.

Table 4.2. Cut-in/Lane-pass classification results of different methods for varying sequence lengths on BDD-100K dataset

Method	Sequence Length	Accuracy	Precision (Cut-in)	Recall (Cut-in)	Precision (Lane-pass)	Recall (Lane-pass)
Baseline	15	0.8851	0.8551	0.8939	0.9114	0.8780
	30	**0.9256**	0.8841	**0.9531**	**0.9620**	0.9048
	45	0.9189	0.9275	0.9014	0.9114	0.9351
	60	0.9256	**0.9275**	0.9143	0.9241	**0.9359**
LSTM-3Class	15	**0.9324**	**0.8960**	0.9230	**0.9620**	**0.9383**
	30	0.9121	0.8667	0.9077	0.9494	0.9146
	45	0.9324	0.8769	**0.9245**	0.9494	0.9146
	60	0.9256	0.8829	0.9077	**0.9620**	0.9383
LSTM-2ClassLR	15	0.9311	0.9321	0.9021	0.9302	0.9524
	30	**0.9585**	**0.9494**	0.9500	0.9651	**0.9648**
	45	0.9452	0.9171	0.9483	0.9651	0.9434
	60	0.9519	0.9160	**0.9648**	**0.9767**	0.9439

4.2.2 Classification Results on Prevention Cut-in/Lane-pass Classification Subset

Each method tested on the BDD-100K dataset is also evaluated on the Prevention cut-in/lane-pass subset (cf. Section 4.2). Accuracy, precision, and recall results with different sequence lengths are shown in Table 4.3.. While all three models are highly successful, LSTM-2ClassLR (best-performing model parameters are: 512 hidden units, 10 as batch size, 0.25 dropout) is slightly better than the others, which is consistent with the results of the BDD-100K dataset. These results indicate that the success of the proposed approach is not specific to a dataset and works well on a benchmark dataset as well.

We observe occasional drops in precision and recall values of cut-in compared to those of the lane-pass class. This is due to the skewness in the dataset. Since the number of lane-pass samples is much higher, the model is inclined to prefer lane-pass more. In the lane-change study with this dataset, Biparva et al. (2021) as well, reported lane-change precision and recall are much lower than those of the no lane-change class.

Table 4.3. Cut-in/Lane-pass classification results of different methods for varying sequence lengths on Prevention dataset

Method	Sequence Length	Accuracy	Precision (Cut-in)	Recall (Cut-in)	Precision (Lane-pass)	Recall (Lane-pass)
Baseline	15	0.9718	0.9259	0.8333	0.9775	0.9909
	30	**0.9799**	**0.9630**	**0.8667**	**0.9820**	**0.9954**
	45	0.9638	0.8519	0.8214	0.9775	0.9819
	60	0.9759	0.8889	0.8276	0.9777	0.9865
LSTM-3Class	15	0.9638	0.6786	**0.9286**	**0.9865**	0.9733
	30	**0.9719**	**0.9524**	0.8958	0.9775	**0.9909**
	45	0.9558	0.7976	0.7976	0.9730	0.9730
	60	0.9598	0.6786	0.7500	0.9820	0.9732
LSTM-2ClassLR	15	0.9788	0.9285	**0.8903**	**0.9845**	0.9920
	30	0.9740	0.9286	0.8452	0.9791	0.9919
	45	0.9665	0.9286	0.7917	0.9710	0.9918
	60	**0.9789**	**0.9524**	0.8690	0.9818	**0.9946**

4.2.3 Lane Change Prediction Results on Prevention Dataset

To compare with the studies in the literature, we trained our LSTM-3Class model for lane change detection using the Prevention Dataset, which Biparva et al. (2021), Fernández-Llorca et al. (2020), and Izquierdo et al. (2019a) used in their studies. As we explained above, the LSTM-3Class model runs with 15, 30, 45, and 60-frame inputs. Since we allowed 50-frame sequences in the lane change detection subset (to keep more samples), we compare only the results of 15-frame, 30-frame, and 45-frame LSTM-3Class models with other studies. That comparison can be seen in Table 4.4..

Even though the focus of our study is cut-in prediction, we see that if the proposed LSTM-based approach (best-performing model parameters are: 512 hidden units, 100 as batch size, 0.25 dropout) is trained for lane change prediction, its performance exceeds the previously reported performances, which are 3-class lane-change prediction accuracies of 0.7440 in Izquierdo et al. (2019a), 0.9190 in Biparva et al. (2021), and 0.9194 in Fernández-Llorca et al. (2020).

Table 4.4. Comparison with the previous lane change maneuver classification studies.

	Method	**Accuracy**
Biparva et al. (2021)	Slow-Fast Networks	0.9190
Izquierdo et al. (2019a)	LeNet + LSTM	0.7440
Fernández-Llorca et al. (2020)	Spatiotemporal Multiplier Network	0.9194
Ours (15-frames)	LSTM-3Class	**0.9270**
Ours (30-frames)	LSTM-3Class	**0.9371**
Ours (45-frames)	LSTM-3Class	**0.9484**

4.2.4 Computational Efficiency

Execution times of our LSTM models for 30-frame input sequences can be seen in Table 4.5.. As shown, vehicle detection and tracking steps of our pipeline take much more time than the classification step, which is not more than 2 milliseconds. Since the vehicle detection step also exists in previous works before the classification step, our approach has the advantage of having just one layer and fewer parameters in the classification step.

Table 4.5. Execution time comparison of evaluated LSTM methods.

Method	**Vehicle Detection and Tracking (sec/seq)**	**Classification (msec/seq)**	**Total (sec/seq)**
Baseline	4.002	2.11	4.0041
LSTM-3class		1.29	4.0032
LSTM-2classLR		1.95	4.0039

All evaluations are done on a PC with Ubuntu 16.04, i7-7700K CPU, 16 GB RAM and an Nvidia GeForce GTX 1080 GPU.

Computation times are directly proportional to the number of frames. Thus, the total execution time is 2 seconds for 15-frame sequences and 8 seconds for 60-frame sequences. Please note that, in the proposed approach, we process two seconds of video regardless of the number of frames in the sequence (15, 30, 45, or 60). Vehicle detection and tracking modules can be executed as frames arrive. Thus, if we use 15-frame sequences, we are able to produce a classification result (cut-in/lane-pass) for the scene every two seconds. As can be seen in Tables 4.2. and 4.3., the results of 15-frame sequences are either the best or very close to the best results. From this point of view, we

can argue that the proposed approach can be considered for real-time implementations to detect cut-in maneuvers.

As mentioned before, we employ CNNs for vehicle detection, however different from the previous studies, we obtain features directly from the target vehicle bounding box and feed them to an LSTM. This is computationally cheaper than feeding a complex CNN with video frames to extract features (Biparva et al. (2021); Fernández-Llorca et al. (2020); Izquierdo et al. (2019a); Yurtsever et al. (2019)). Thus, the feature extraction and classification times are longer for the methods in the literature.

CHAPTER 5

Maneuver Detection with Self-supervised Contrastive Video Representation Learning

Supervised learning has revolutionized the landscape of machine learning, yet the insatiable demand for a large volume of labeled data remains a significant challenge. This limitation paves the way for the emergence of self-supervised learning, an effective method that leverages the bounty of unlabeled data to derive meaningful representations that can be subsequently used in fine-tuning a supervised training method with a smaller labeled dataset.

Self-supervised contrastive learning has recently gained significant traction, especially in computer vision, owing to its outstanding performance in several applications (Bastanlar and Orhan (2022); Le-Khac et al. (2020)). As an example of some of these applications, MoCo (He et al. (2020)) and SimCLR (Chen et al. (2020)) incorporate both negative (dissimilar) and positive (similar) examples in their strategies. However, BYOL (Grill et al. (2020)) and SimSiam (Chen and He (2021)) achieve comparable performance solely using positive examples, demonstrating different augmentations of the same instance. Although the task in these seminal works was image classification, soon after, many other tasks such as object detection and semantic segmentation benefited from self-supervised contrastive learning as well (Radford et al. (2021); Li et al. (2022); Orhan et al. (2022)).

In the context of video analysis, Qian et al. (2021) introduced Contrastive Video Representation Learning (CVRL), a self-supervised learning methodology aimed at extracting spatiotemporal visual features from unlabeled video footage. The technique utilizes a contrastive loss function to drive similar video clips in the embedding space closer together while distancing video clips originating from different videos. Building on this, Tao et al. (2020) and Han et al. (2020) expanded this framework with intra-negative and intra-positive sampling and concurrent network training, respectively, Lin et al. (2021) and Knights et al. (2021) incorporated meta-learning and temporal coherence loss to enhance video representation learning further.

Despite the impressive advancements achieved in human action classification using these methods, their application in vehicle maneuver classification, another task involving video analysis, still poses distinct challenges. The two tasks significantly differ in aspects like visual features, environmental variability, and the importance of temporal information. In contrast to human actions, vehicle maneuvers occur in a largely uniform environment

and lack distinct visual cues. Moreover, the temporal sequence of events is paramount in vehicle maneuver classification.

The presented work in this chapter extends the self-supervised learning approach by Qian et al. (2021), tailoring it to overcome the unique challenges in vehicle maneuver classification (Nalcakan and Bastanlar (2023)). Utilizing simplified video clips with suitably chosen augmentations, this thesis represents, to the best of the author's knowledge, the first application of self-supervised learning in predicting vehicle maneuvers.

5.1 Methodology

In this study, the potential of self-supervised learning was leveraged to enhance the detection of both lane change and cut-in maneuvers in front of the ego vehicle. This approach allowed for the use of a large dataset without the need for maneuver labeling. Self-supervised contrastive learning is utilized with the video representation learning methodology, which has been proposed by Qian et al. (2021) as the framework. The framework comprises two phases (Figure 5.1.). In the self-supervised training phase, the encoder network is pre-trained with unlabeled highway-recorded video clips with contrastive loss to learn vehicle maneuver representations and their interactions with the ego-lane. To enhance self-supervised learning, we converted the videos to high-level representations of the scene (simplified views), which is done by segmenting vehicles and ego-lane and subtracting background. The same high-level representation extraction is applied to the prepared cut-in/lane-pass datasets and lane change detection dataset and is used to fine-tune the encoder while training a classification head to classify the maneuver in the scene or maneuver of the target vehicle.

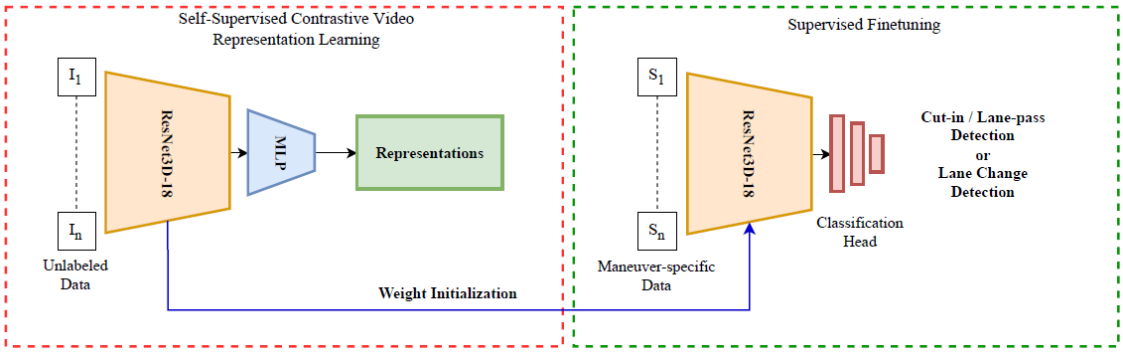

Figure 5.1. Overview of the proposed framework. After self-supervised learning with contrastive loss, the MLP head is discarded, a new classification head is added to the 3D ResNet backbone, and supervised retraining is conducted.

5.1.1 Contrastive Video Representation Learning

Contrastive video representation learning is a form of self-supervised learning. This approach is characterized by its ability to learn distinctive features from data by comparing similar (positive) and dissimilar (negative) examples. The goal is to make the representations of similar examples closer and those of dissimilar examples farther apart in the embedding space (Figure 5.2.).

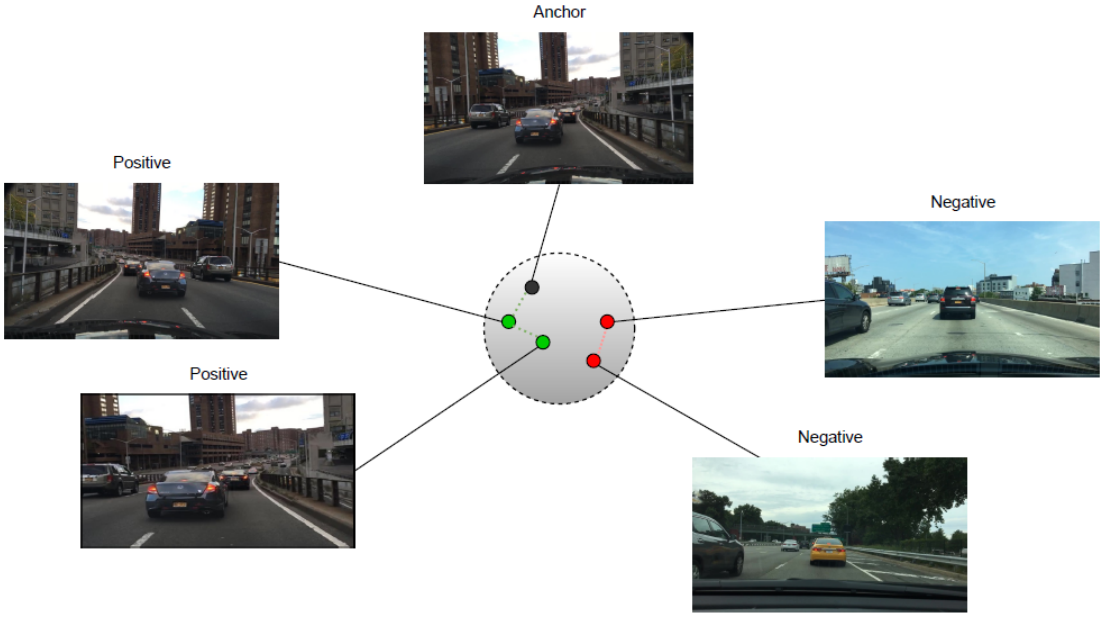

Figure 5.2. An example self-supervised contrastive learning scenario. The positive pairs are generated from the anchor frame(or video) with random center crop and horizontal flip, and negative ones are collected randomly from the dataset.

The InfoNCE loss (Oord et al. (2018)), a type of contrastive loss, plays a crucial role in this learning process. It is a function that quantifies the difference between the representations of positive and negative pairs. By minimizing the InfoNCE loss, the model is encouraged to generate similar representations for different augmentations of the same video clip and dissimilar representations for augmentations of different video clips. Furthermore, it allows positive pairs to be close and lets others to be distant on feature space by using the equation $\mathcal{L} = \frac{1}{N}\sum_{i=1}^{N}\mathcal{L}_i$ and $\mathcal{L}_i$:

$$\mathcal{L}_i = -\log \frac{\exp\left(\text{sim}\left(z_i, z_i'\right)/\tau\right)}{\sum_{k=1}^{2N} \mathbf{1}_{[k\neq i]} \exp\left(\text{sim}\left(z_i, z_k\right)/\tau\right)} \tag{5.1}$$

where z_i, z_i' denotes the encoded representations of the two augmented clips of the i^{th} video, N is the number of samples in the batch producing a total of $2N$ augmentations per batch, $\mathrm{sim}(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^\top \boldsymbol{v} / \|\boldsymbol{u}\|_2 \|\boldsymbol{v}\|_2$ is the inner product between two ℓ_2 normalized vectors, $\mathbf{1}_{[.]}$ is an indicator to exclude the self-similarity of video z_i, and $\tau > 0$ is a temperature parameter.

Therefore, augmentations have a significant impact on self-supervised learning. They introduce variability and help the model generalize better to unseen data. In the context of this study, different augmentations of the same video clip were used to create positive pairs, while augmentations of different video clips were used to create negative pairs. This strategy not only enriched the diversity of the training data but also enhanced the model's ability to learn a wide range of maneuver patterns and scenarios. Consequently, this led to an improvement in the detection capabilities of the model for both cut-in maneuvers and lane-change maneuvers. Figure 5.3. shows details of the proposed maneuver representation learning phase.

5.1.2 Scene Representation

The extraction of scene representation from videos is a crucial aspect of our methodology. Since the detection of maneuvers is directly related to vehicles and lane lines, we chose to feed the framework with a simpler representation of the scene, which includes the vehicles in front and ego-lane, rather than the original image sequence (Figure 5.4.). These representations capture the essential features of each scene, such as the positions and movements of surrounding vehicles and the ego-lane, while filtering out less relevant details. Vehicle representations were created by a state-of-the-art instance segmentation method, Detectron 2 (Wu et al. (2019)), and the ego-lane mask was extracted by YOLOPv2 (Han et al. (2022)) model. We also down-sampled the original video clips from 60 to 20 frames by taking one of every three frames. Since lane change prediction studies in the literature make decisions per vehicle, i.e., target-based, the simplified view we created for lane change prediction only includes the target vehicle mask and the ego-lane mask. In summary, we used scene-based simplified view data for all models in our cut-in/lane-pass detection methods and target-based simplified view data for all models in our lane change detection methods.

The advantage of using these scene representations over raw video frames is twofold. Firstly, they provide a simplified and more abstract view of the scene, which can help the model focus on the most important elements for maneuver detection. Secondly, they reduce the dimensionality of the input data, which can make the learning process more

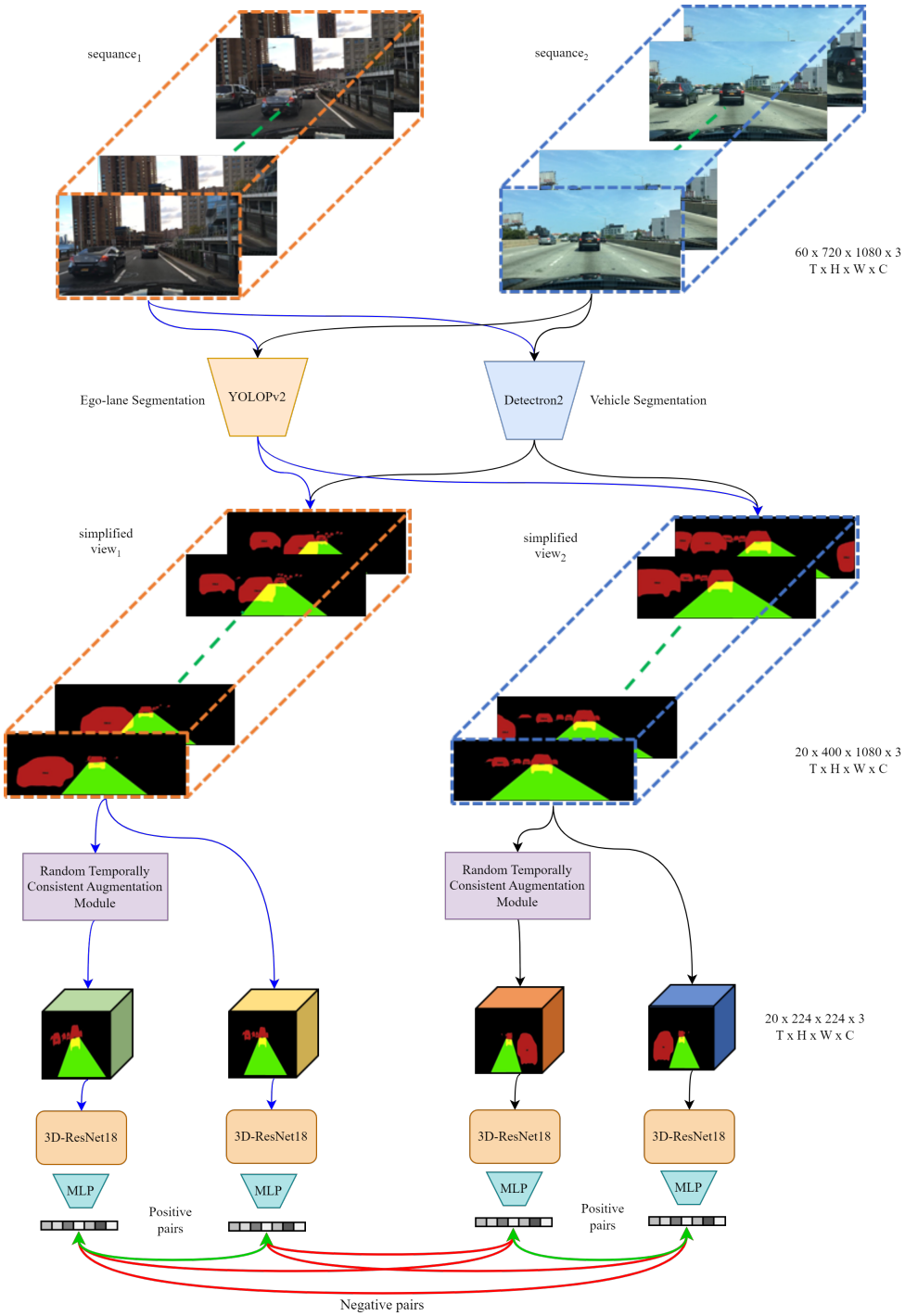

Figure 5.3. Contrastive Maneuver Representation Learning. First, we create simplified video clips by extracting the vehicle and ego-lane masks from raw videos. Then, different temporally consistent augmentations (e.g. crop, flip, shear, rotation) are applied to the simplified video clips randomly before giving them as input to our video encoder (3D-ResNet18). For self-supervised learning, extracted feature tensors of each sample in mini-batch are compared with InfoNCE loss such that representations of positive pairs (green arrows) are brought together, and representations of negative pairs (red arrows) are put far apart.

efficient and less prone to overfitting. The use of scene representations also has a significant impact on the creation of new augmentations for self-supervised learning. By working with these representations, we can generate augmentations that maintain the essential features of the original scenes while introducing variability in less important aspects. This can help the model learn to recognize a wide range of maneuver patterns and scenarios and generalize better to unseen data. Furthermore, the use of scene representations allows for the creation of more diverse and challenging negative pairs, which can further enhance the effectiveness of the contrastive learning approach.

Figure 5.4. Generation of scene-based simplified view with an example cut-in maneuver from BDD-100K Cut-in/Lane-pass Subset. Overlapping masks of vehicles and ego-lane are in different colors. The figure shows four frames of a single sequence, whereas the 3D network uses 20 of them for classification. The frame height is reduced from 600 to 400 pixels to remove the ego vehicle's hood and some sky.

5.1.3 Augmentations

To enable the self-supervised model to learn the spatial and temporal attributes of the scene, the augmentations we use should imitate different situations that may not be included in the labeled data set. At the same time, augmentations should not include cases that would not occur in real life. For that reason, we employed five different augmentations for video representation learning. Of these methods, *random rotation* and *random shear* were used to imitate the various differences in the road view of the in-vehicle camera, *horizontal flip* to simulate that the maneuver could take place on the opposite side, and *center crop* to simulate that the camera may have a narrower field of view. To ensure that representations are not affected by the random selection of augmentations, they are kept consistent temporally. In other words, the same augmentation is applied to all frames of a video clip in the same way. Application of the augmentations to scene-based simplified

view of cut-in sequences and lane-pass sequences are given in Figure 5.5..

In addition to the four augmentations mentioned above, considering that the speed of the maneuvering vehicle can change in time, we employed another augmentation which is called *temporal elastic transformation* (TET)(Stamoulakatos et al. (2021)). The method works in one of the two different paths. In the first possibility, it stretches the beginning and the end of the video shrinks the middle, and in the second, it does the opposite.

Algorithm 1 conveys the details of producing spatial and temporal augmentations in self-supervised contrastive learning.

Algorithm 1: Random Temporally Consistent Augmentation

Input: Video clip $V = \{I_1, I_2, \cdots, I_N\}$ with N frames `// N equal to 20 frames`

Crop: Randomly select a scale value **s** from $[0.6, 1.0]$ Define $newwidth = width * \mathbf{s}$ and $newheight = height * \mathbf{s}$

Resize: Resize the cropped region to size of 224×224

Rotate: Randomly select a rotation degree **d** from $[-5°, 5°]$

Flip: Select a flag $\mathbf{F}_f$ from $\{0, 1\}$ with 50% on 1

Shear: Randomly select a shear value **sh** for **x** and **y** coordinates from $[-0.2, 0.2]$

TET: Randomly select operation type from $\{-1, 1\}$

Select: Randomly select a number from $[1, 5]$

$number = Select()$

for $n \in \{1, 2, \cdots, N\}$ **do**

$I'_n = Flip(I_n)$ if $\mathbf{F}_f = 1$ and if $number = 1$

$I'_n = Rotate(I_n)$ by d degree if $number = 2$

$I'_n = Resize(Crop(I_n, newwidth, newhight))$ if $number = 3$

$I'_n = Shear(I_n)$ transform **x** and **y** with **sh** shear value if $number = 4$

$I'_n = TET(I_n)$ if $number = 5$

end

Output: Augmented video clip $V' = \{I'_1, I'_2, \cdots, I'_N\}$

According to Tao et al. (2020), all the augmentations mentioned above are defined as intra-positive. They propose a method called Inter-intra Contrastive training. This uses intra-negative samples (same video with broken temporal relationships between frames) next to inter-negative samples for negative sampling and optical flow or frame differences for positive sampling in contrastive learning (Figure 5.6.). They suggest that the use of intra-negative augmentations in self-supervised contrastive learning also contributes positively to success. Therefore, to evaluate the effect of intra-negative augmentations for our method, we tried two of their intra-negative augmentations (*temporal repeating* and *temporal adjacent shuffle*) together with intra-positive augmentations as an ablation study. Temporal repeating means a frame is randomly selected from the video, and the entire video is composed only from this frame. In this way, temporal and motion information is made non-existent. Temporal adjacent shuffle means the temporal order of the motion is broken by randomly shuffling the frames of the video. As a note, the horizontal flip that we used as a positive augmentation for the cut-in/lane-pass detection task was repurposed as an intra-negative augmentation for the lane change prediction task. This was due to the fact that the horizontal flip disrupted class-specific information (by confusing the right

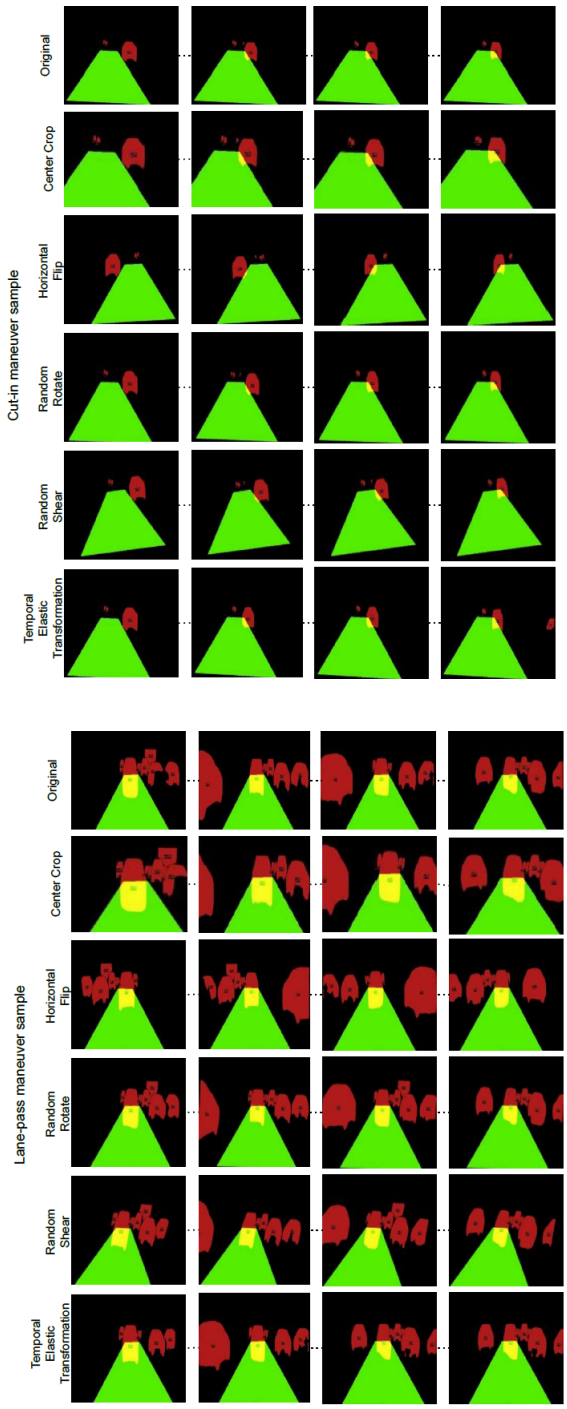

Figure 5.5. Example outputs of applied augmentations on cut-in and lane-pass maneuvers. Each row shows a different augmentation of the original sequence (top row). Only *temporal elastic transformation* (TET) augmentation is not included in the figure. Since it stretches/shrinks the video sequence in time, showing the effect with a few frames is not possible.

and left side information). As an ablation study, six negative samples were added to each batch in self-supervised training next to the other 10 video samples.

As an ablation study on another augmentation, since we will be using target-based simplified view data for the lane change prediction task, we proposed two target-based

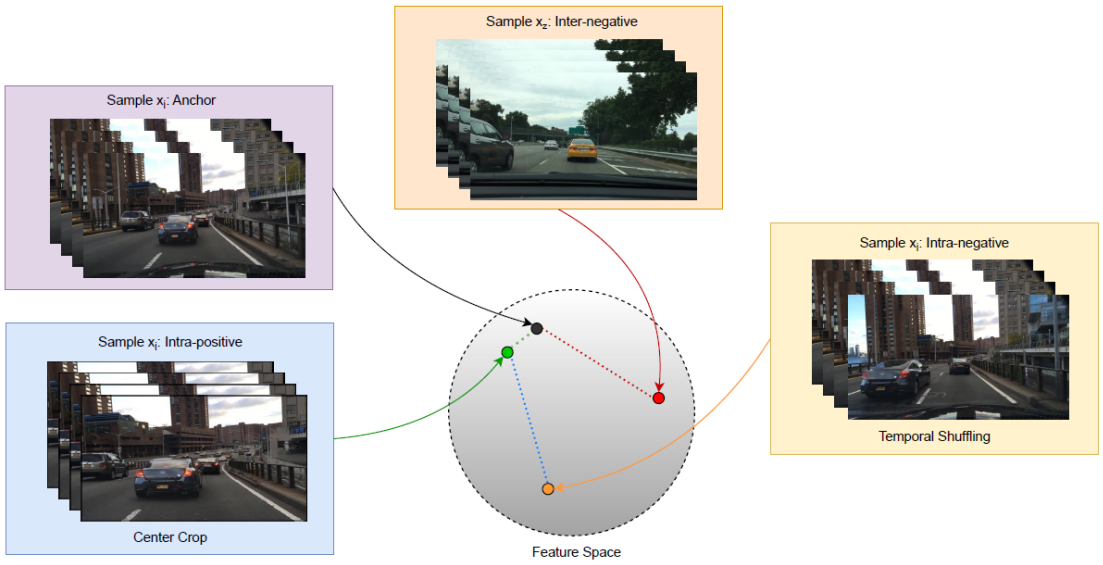

Figure 5.6. Feature space representation of intra-positive, intra-negative, and inter-negative samples. (Source: Tao et al. (2020))

augmentations (*random mask move* and *random mask scale*) where we apply various transformations to the target vehicle mask and the ego-lane mask without disturbing the information about the maneuver class. We used these target-based augmentations in addition to the augmentations we use in self-supervised contrastive learning.

5.1.4 Video Encoder

We processed video frames using ResNet3D-18 (Hara et al. (2018)) with 3D convolution kernels instead of the 2D convolution kernels as in original ResNet architectures to encode spatio-temporal features of scenes. We trained two different backbone architectures, one is single ResNet3D-18, and the other is ResNet3D-18 with a multi-layer projection (MLP) head on top, as suggested in Chen et al. (2020); Qian et al. (2021). During supervised retraining, we dropped the MLP head and added the classification head onto ResNet3D-18 (Figure 5.7.b).

For the supervised retraining phase, we evaluated three different classification heads (one linear layer, two nonlinear layers, and four nonlinear layers) on top of backbones. Evaluated architectures are depicted in Figure 5.7.a.

Self-supervised training of video encoder was performed with Adam optimizer (Kingma and Ba (2014)), 0.1 as initial learning rate, 32 as batch size, 500 as training epochs, and 0.1 for temperature τ. In the supervised re-training phase, the same optimizer

was used, but the batch size was reduced to 8 and the learning rate to 0.001. Different numbers of epochs (between 200 and 500) were evaluated.

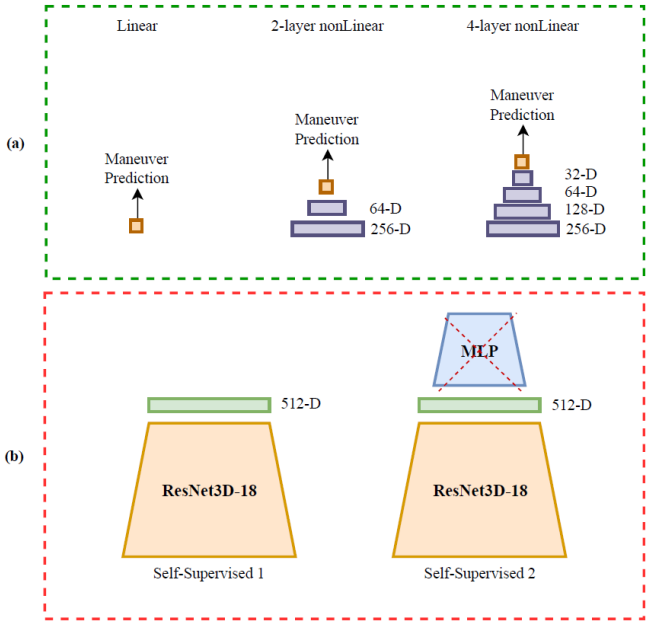

Figure 5.7. Evaluated network architecture alternatives for classification head (a) and self-supervised training (b).

5.2 Experimental Results

5.2.1 Results on BDD-100K Cut-in/Lane-pass Classification Subset

We compared the classification performances of four approaches by reporting the best fold and 5-fold cross-validation accuracies and F1 scores of each approach on the Cut-in/Lane-pass maneuver detection task. As supervised baselines, first we used a 2D-CNN(MobileNetV2 (Sandler et al. (2018))) to extract features from each frame of the clip and give those as input to an LSTM (Baseline 1), secondly, the same 3D-CNN architecture of our CVRL approach which is a ResNet3D-18 was evaluated as Baseline 2. Baselines 1 and 2 achieved 5-fold CV accuracy of 82.03% and 90.69% with 2-layer nonlinear classification head on the BDD cut-in/lane-pass subset (Table 5.1.).

Classification accuracy increased by ~2% and achieved 92.52% with self-supervised pre-trained ResNet3D-18 network and 4-layer nonlinear head.

Table 5.1. 5-fold cross-validation results of both supervised baselines and self-supervised approaches on BDD-100K Cut-in/Lane-pass subset.

Approach	**Backbone**	**Classification head type**	**5-fold CV acc** (%)	**5-fold CV F1 score**
Baseline 1	MobileNetV2+LSTM	Linear	78.35 ± 04.91	79.32 ± 04.90
		2-layer nonlin.	82.03 ± 02.30	82.53 ± 01.83
		4-layer nonlin.	79.72 ± 03.49	79.74 ± 03.52
Baseline 2	ResNet3D-18	Linear	85.19 ± 03.35	85.19 ± 03.48
		2-layer nonlin.	90.69 ± 02.19	90.59 ± 02.27
		4-layer nonlin.	88.70 ± 02.12	88.70 ± 02.27
Self-sup.1	ResNet3D-18	Linear	91.15 ± 02.45	91.18 ± 02.42
		2-layer nonlin.	92.37 ± 01.79	92.30 ± 01.81
		4-layer nonlin.	**92.52 ± 01.74**	**92.54 ± 01.78**
Self-sup.2	ResNet3D-18+MLP	Linear	90.99 ± 02.05	90.98 ± 02.11
		2-layer nonlin.	92.21 ± 01.47	92.18 ± 01.48
		4-layer nonlin.	92.21 ± 02.38	92.16 ± 02.37

5.2.2 Results on Prevention Cut-in/Lane-pass Classification Subset

The same experimental configuration as in 5.2.1 was used on the Prevention cut-in/lane-pass classification subset. Baselines 1 and 2 performed 5-fold CV accuracies of 86.20% and 89.40% respectively with the 4-layer nonlinear classification head (refer to Table 5.2.). Reflecting the performance in the BDD-100K subset, self-supervised training enhanced the performance of the baselines by approximately 2%. This was achieved using the Self-sup.2 model, which is the ResNet3D-18+MLP network with a linear classification head, reaching a 5-fold CV accuracy of 91.60%.

Table 5.2. 5-fold cross-validation results of both supervised baselines and self-supervised approaches on the Prevention Cut-in/Lane-pass subset.

Approach	**Backbone**	**Classification head type**	**5-fold CV acc** (%)	**5-fold CV F1 score**
Baseline 1	MobileNetV2+LSTM	Linear	82.80 ± 04.66	80.55 ± 05.29
		2-layer nonlin.	84.00 ± 02.92	82.09 ± 03.54
		4-layer nonlin.	86.20 ± 04.09	84.32 ± 04.95
Baseline 2	ResNet3D-18	Linear	88.60 ± 03.21	87.11 ± 03.70
		2-layer nonlin.	88.80 ± 02.17	87.44 ± 02.45
		4-layer nonlin.	89.40 ± 02.30	88.00 ± 02.70
Self-sup.1	ResNet3D-18	Linear	90.20 ± 01.64	88.77 ± 01.89
		2-layer nonlin.	88.60 ± 02.41	87.01 ± 02.66
		4-layer nonlin.	87.00 ± 02.92	85.03 ± 03.43
Self-sup.2	ResNet3D-18+MLP	Linear	**91.60 ± 01.95**	**90.41 ± 02.32**
		2-layer nonlin.	90.20 ± 02.39	88.79 ± 02.72
		4-layer nonlin.	90.00 ± 03.24	88.58 ± 03.74

5.2.3 Results on Prevention Lane Change Detection Dataset

The results presented in Table 5.3. demonstrate the effectiveness of the four approaches used in this study for the Lane Change Detection task. The table shows the 5-fold cross-validation results of both supervised baselines and self-supervised approaches on the Prevention Lane Change Prediction Dataset.

The most successful method among previous vision-based studies had reached a lane change classification accuracy of 91.94% with supervised learning (without cross-validation) on Prevention Dataset.

The self-supervised approach 1, which used a linear classification head type with the ResNet3D-18 backbone, achieved the highest 5-fold cross-validation accuracy of 92.13% and F1 score of 73.13%. This approach outperformed all other methods, including the supervised baselines and the self-supervised approach 2, representing an increase of 0.19 percent in accuracy over the highest reported result in the literature. However, studies in the literature that worked on the same dataset did not report cross-validation results which are crucial for assessing the robustness of a model. Since the results they reported are from their best models, if we compare these with the best fold results from our methods, we can say that our most successful model, the self-supervised approach 2, using a 4-layer nonlinear classification head with the ResNet3D-18+MLP backbone, achieved an accuracy of 93.06% and an F1 score of 75.90%. This represents a 1.12 percent increase in accuracy over the most successful model in the literature.

These results demonstrate the potential of the proposed methods for improving

Table 5.3. 5-fold cross-validation results of both supervised baselines and self-supervised approaches on the Prevention Lane Change Prediction Dataset.

Approach	**Backbone**	**Classification head type**	**5-fold CV acc** (%)	**5-fold CV F1 score**
Baseline 1	MobileNetV2+LSTM	Linear	88.28 ± 01.11	63.26 ± 03.37
		2-layer nonlin.	89.35 ± 01.89	66.66 ± 07.27
		4-layer nonlin.	88.51 ± 02.51	76.22 ± 06.85
Baseline 2	ResNet3D-18	Linear	90.15 ± 00.60	70.42 ± 02.45
		2-layer nonlin.	90.08 ± 00.60	68.08 ± 03.37
		4-layer nonlin.	89.62 ± 01.09	63.60 ± 05.52
Self-sup.1	ResNet3D-18	Linear	**92.13 ± 00.61**	**73.13 ± 01.40**
		2-layer nonlin.	91.74 ± 00.44	72.59 ± 01.64
		4-layer nonlin.	91.40 ± 00.26	70.25 ± 01.70
Self-sup.2	ResNet3D-18+MLP	Linear	91.21 ± 00.71	71.21 ± 01.76
		2-layer nonlin.	91.47 ± 00.63	71.87 ± 02.22
		4-layer nonlin.	91.93 ± 00.77	72.63 ± 02.38

lane change detection accuracy. Notably, the proposed methods have allowed us to exceed the classification accuracies reported in other studies that used the Prevention Dataset. This comparison with previous studies highlights the advancements made in this research, setting a new benchmark for lane change detection accuracy. A significant aspect of this work is the use of self-supervised contrastive learning for the first time in the context of lane change detection. This innovative approach has contributed to the superior performance of our methods, demonstrating the potential of self-supervised learning in this domain.

5.3 Ablation Study

In the ablation study section, we explore the performance and effectiveness of various components of our methodology. This section is divided into four ablation studies. Each of these studies was conducted to investigate the impact of different elements in our proposed self-supervised method on performance and to explore whether alternative techniques could have been proposed. This comprehensive examination allows us to understand the significance of each element in our approach and provides a foundation for potential enhancements or modifications in future research.

The first two studies, 'Effect of Simplified Scene Representation' and 'Effect of Spatial and Temporal Augmentations', are conducted on the BDD-100K Cut-in/Lane-pass Classification Subset. We employ our best-performing approach, Self-supervised 1, with the ResNet3D-18 backbone for these studies. The other two studies, 'Effect of Negative

Augmentations' and 'Effect of Target-based Augmentations', are carried out on the Prevention Lane Change Classification dataset. In these cases, we utilize both self-supervised methods, ResNet3D-18 and ResNet3D-18+MLP, to comprehensively understand their performance under different conditions.

5.3.1 Effect of Simplified Scene Representation

To measure the impact of our proposed simplified scene representation on performance, which is designed to facilitate the learning of vehicle movements and their relationship with the ego lane through the proposed augmentations in the self-supervised model, we created five different datasets from our main dataset, each with a different setting. These datasets are original RGB frames, RGB frames with vehicle masks, RGB frames with vehicle masks and ego lane mask, black background with vehicle masks, and the proposed simplified scene representation (black background with vehicle masks and ego lane mask). This variety of settings allows us to isolate and evaluate the contribution of each element in the scene representation.

As results in Table 5.4. indicate, the simplified version outperformed the original RGB input type in either masked or unmasked form. Furthermore, the application of vehicle and ego-lane masks appears to enhance the classification performance across both input types, supporting the effectiveness of our proposed simplified scene representation.

Table 5.4. 5-fold CV accuracies (%) with different data types to evaluate the impact of simplified scene representation. Video clips' original versions and simplified versions were evaluated with different highlighted information in the scene.

Input type	**Vehicle masks**	**Ego-lane mask**	**Linear layer**	**2-layer nonLinear**	**4-layer nonLinear**
Original	x	x	57.29	58.86	59.00
	✓	x	57.57	60.14	60.14
	✓	✓	70.29	69.43	60.29
Simplified	✓	x	67.00	69.86	67.43
	✓	✓	**91.15**	**92.37**	**92.21**

5.3.2 Effect of Spatial and Temporal Augmentations

In a subsequent ablation study, the impact of individual augmentations on the performance of the top-performing approach (Self-sup.1 as per Table 5.1.) was examined. As shown in Table 5.5., the most effective results were achieved when all augmentations were activated during training. Interestingly, while temporal augmentation alone did not prove sufficient, its combination with spatial augmentations led to a noticeable enhancement in performance. This suggests that the interplay between spatial and temporal augmentations plays a significant role in optimizing the model's performance.

Table 5.5. Ablation study results of different augmentation types applied in the self-supervised learning phase. Self-sup.1 approach's (ResNet3D-18) results are given since it is the best performer in Table 5.1..

Classification head type	**Augmentations**	**5-fold CV acc (%)**	**5-fold CV F1 score**
Linear	Spatial	90.53 ± 01.99	90.40 ± 02.02
2-layer nonlin.	Spatial	92.06 ± 01.92	92.03 ± 01.86
4-layer nonlin.	Spatial	92.21 ± 01.74	92.18 ± 01.80
Linear	Temporal	87.63 ± 01.00	87.58 ± 01.00
2-layer nonlin.	Temporal	81.22 ± 02.78	81.02 ± 02.80
4-layer nonlin.	Temporal	80.31 ± 02.05	80.01 ± 02.07
Linear	Spatial&Temporal	91.15 ± 02.45	91.18 ± 02.42
2-layer nonlin.	Spatial&Temporal	92.37 ± 01.79	92.30 ± 01.81
4-layer nonlin.	Spatial&Temporal	**92.52 ± 01.74**	**92.54 ± 01.78**

5.3.3 Effect of the Use of Intra-positive and Intra-negative Augmentations

In this ablation study, we examined the impact of negative augmentations on the performance of our self-supervised models. The experiments for this study were conducted on the Prevention Lane Change Classification dataset, utilizing both self-supervised methods, ResNet3D-18 and ResNet3D-18+MLP, to gain a comprehensive understanding of their performance under different conditions.

As detailed in Chapter 5.1.3, augmentations in self-supervised contrastive learning can be classified as intra-positive, inter-negative, and intra-negative. In this study, we evaluated the effect of two intra-negative augmentations, namely *temporal repeating* and *temporal adjacent shuffle*. Additionally, horizontal flip augmentation which was used as a positive augmentation for the cut-in/lane-pass detection task was repurposed as an intra-negative augmentation for the lane change prediction task due to its disruption of class-specific information (by confusing the right and left side information). For this ablation study, six randomly created negative samples were added to each batch in self-supervised training alongside the other ten video samples which some are intra-positive augmented samples and some are not augmented samples.

Table 5.6. presents the results of the experiments. For the ResNet3D-18 backbone, the 2-layer nonlinear classification head type achieved the highest 5-fold cross-validation accuracy of 91.90%, with a corresponding F1 score of 73.01%. For the ResNet3D-18+MLP backbone, the linear classification head type achieved the highest 5-fold cross-validation accuracy of 91.27%, with a corresponding F1 score of 70.76%. Although both models with intra-negative augmentations get close to the results of our original method using intra-positive augmentations, they were unable to surpass those results. Furthermore, upon examining the instances where the models made errors, we found that they were unable to correct the mistakes made by our original model configuration. Therefore, we did not consider conducting a self-supervised training by combining intra-positive and intra-negative augmentations.

Table 5.6. Experimental results of the use of intra-positive and intra-negative augmentation types applied in the self-supervised learning phase. Results can be compared with Table 5.3.

Backbone	**Classification head type**	**5-fold CV acc** (%)	**5-fold CV F1 score**
ResNet3D-18	Linear	91.56 ± 00.86	71.79 ± 03.41
	2-layer nonlin.	**91.90 ± 00.61**	**73.01 ± 02.12**
	4-layer nonlin.	91.57 ± 00.61	72.89 ± 02.29
ResNet3D-18+MLP	Linear	91.27 ± 00.46	70.76 ± 02.22
	2-layer nonlin.	90.61 ± 00.85	69.52 ± 02.14
	4-layer nonlin.	90.91 ± 00.69	70.38 ± 01.85

5.3.4 Effect of Target-based Augmentations

Since the lane change detection task involves maneuvers based on the target vehicle, as an ablation study we wanted to see if applying augmentations to the target vehicle mask and the ego lane mask could teach new representations of the maneuvers during self-supervised training.

As previously mentioned in Chapter 5.1.3, we proposed two target-based augmentations, "random mask move" and "random mask scale". These augmentations apply various transformations to the target vehicle mask and the ego-lane mask without disturbing the information about the maneuver class. For this study, only these target-based augmentations were used in self-supervised contrastive learning.

Looking at the 5-fold cross-validation results (Table 5.7.), the ResNet3D-18 model with a linear classification head type achieved an accuracy of 90.12% and an F1 score of 66.40%. On the other hand, the ResNet3D-18+MLP model with a linear classification head type achieved a slightly higher accuracy of 90.39% and an F1 score of 67.69%. Although the target-based augmentations contributed to the training of the model, they were not as successful in learning the representations of the maneuvers as the augmentations in the original version of the model. Moreover, when we look at the instances where the models made errors, as in the case of negative augmentations, we see that they did not improve the original model.

Table 5.7. Experimental results of target augmentation types applied in the self-supervised learning phase. Results can be compared with Table 5.3.

Backbone	**Classification head type**	**5-fold CV acc** (%)	**5-fold CV F1 score**
ResNet3D-18	Linear	90.12 ± 00.73	66.40 ± 02.78
	2-layer nonlin.	89.79 ± 00.56	65.88 ± 01.34
	4-layer nonlin.	89.22 ± 00.69	64.35 ± 03.76
ResNet3D-18+MLP	Linear	90.39 ± 00.84	67.69 ± 02.61
	2-layer nonlin.	**90.48 ± 00.74**	**68.24 ± 01.84**
	4-layer nonlin.	89.75 ± 01.03	66.59 ± 02.60

CHAPTER 6

Ensemble Learning for Maneuver Detection

Ensemble learning is a powerful technique that leverages multiple learning algorithms to obtain better predictive performance than the results that can be achieved from any single learning algorithm alone. Recently, ensemble methods have been incorporated into the realm of deep learning and have demonstrated impressive results across a variety of tasks (Ganaie et al. (2022)). Ensemble techniques typically involve training multiple neural networks by varying the initial parameters or using different architectures altogether. The individual models (referred to as "base learners") are then combined using various strategies to improve generalization ability and robustness over any single model.

There are several ways to create ensemble models in deep learning. One method is bootstrapped ensembles, where different subsets of the original data are used to train different models. Another approach is architectural ensembles, where models with different architectures are combined. This can be particularly effective, as models with different architectures often learn to represent different features and thus can complement each other (Zhang and Ma (2012)). Another approach is known as Snapshot Ensembles (Huang et al. (2017)), where instead of relying on random initialization to produce different models, an ensemble is created from the different local minima found during the training process of a single network. These strategies can be further combined to form more sophisticated ensembles.

In the context of vehicle maneuver detection, an integral component of Advanced Driver Assistance Systems, ensemble learning is beneficial and critical. ADAS are designated to enable vehicle safety and enhance the driving experience. Therefore, ensuring high performance, robustness, and accuracy of these systems is vital. Ensemble learning serves as a powerful mechanism toward this goal. By integrating the predictive capabilities of multiple models, the technique offers a strategic means to offset individual model weaknesses and improve the overall performance of the maneuver detection system.

Equally significant is the ability of ensemble learning to account for a wide variety of situations that a single model might fail to consider. Given the dynamic and unpredictable nature of road scenarios, it is vital for ADAS to be comprehensive in their detection and response mechanisms. Ensemble learning, with its capacity to combine diverse model perspectives, helps ensure the detection system is well-equipped to handle an extensive range of driving situations. The synthesis of multiple models' insights aids in building a more resilient and reliable maneuver detection system, thereby contributing substantially to vehicle safety and the effectiveness of ADAS.

The applicability of ensemble learning to our problem was explored by the methodologies explained in Chapters 4 and 5 of this thesis work. These methods were evaluated with two ensemble learning strategies, soft voting and weighted sum voting. They are designed to leverage the strengths of each method, offset their weaknesses, and optimize overall performance. Since the most studied task in the literature is lane change prediction, we decided to examine our methods' effectiveness with ensemble learning when training on the Prevention Lane Change Prediction dataset. An outline of the ensemble learning approach is presented in Figure 6.1..

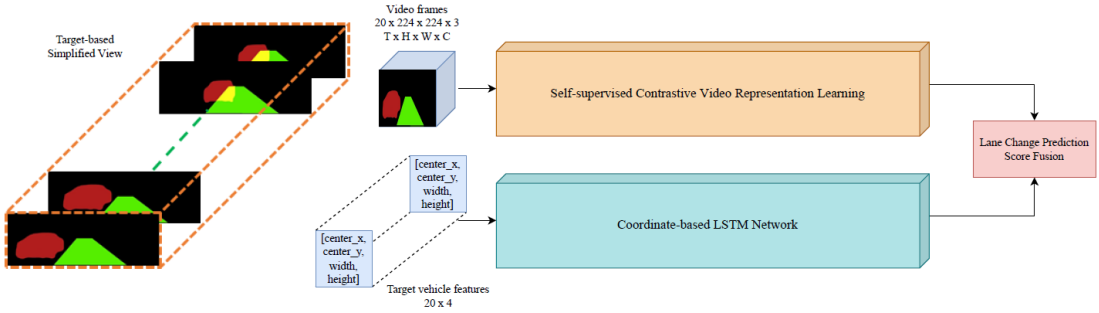

Figure 6.1. Overview of the ensemble learning approach. The target-based simplified view is used to extract features for each of the networks. Center coordinates, width, and height of the TV mask are given as input to the LSTM3class model from Chapter 4, and the simplified view clip is processed via the CVRL model from Chapter 5. As the ensemble learning method, two score-based fusion approaches are applied to the prediction probabilities of each model.

Soft Voting combines the probabilistic predictions from all models. Unlike 'hard' voting which only considers the most frequently predicted class, soft voting accounts for the degree of confidence (probability) associated with each prediction. This technique enhances predictive performance, particularly when the ensemble models yield well-calibrated probabilistic outputs. Thus, soft voting offers a nuanced mechanism that can leverage the predictive certainty of individual models to derive more accurate ensemble predictions. In our case, the prediction probability outputs of the coordinate-based LSTM network (LSTM3class) and self-supervised contrastive video representation learning model (CVRL) were summed separately for each class and the class with the maximum prediction probability was selected as the mutual prediction.

Weighted Sum Voting also known as weighted majority voting, is a method for making final predictions based on combining predictions from multiple models. The basic idea is that instead of giving each model in the ensemble an equal vote in the final decision (as in soft voting), each model is assigned a weight that reflects its performance

or importance. These weights can be set based on each model's accuracy or any other performance metric. This method was chosen to leverage the complementary strengths of the individual models. Each model was allocated a weight, representing its relative contribution to the final decision-making process. Weights were set to range from 0.1 to 0.9 for each model, requiring that the total sum of the weights equals 1, ensuring a balanced contribution from both models. This array of weight assignments enabled us to systematically explore the impact of the relative model influence on the overall ensemble performance.

Data set. Given that our lane change detection dataset involves making decisions per target vehicle, our simplified view includes only the target vehicle mask and the ego-lane mask (Figure 6.2.). The target vehicle mask is generated using a state-of-the-art instance segmentation method, Detectron 2 (Wu et al. (2019)), while the ego-lane mask is extracted using YOLOPv2 (Han et al. (2022)). In order to ensure that the models work with the same sequence lengths, we have included the videos with a length of at least 60 frames from the examples in the Prevention Lane Change Prediction dataset into our dataset. The first 60 frames of each video were selected and then reduced to 20 frames as shown in Figure 6.2. and the same dataset was used both in LSTM-3class and in the self-supervised contrastive video representation learning method. Since the average number of frames per video for the examples belonging to the "no lane change" class is below 60 (Table 3.2.), the result obtained by LSTM-3class is lower than Table 4.4.. Additionally, we removed samples from our data set in this experiment that either belonged to vehicles coming from the opposite lane or had a vehicle mask area value smaller than the threshold we set. The dataset was distributed as follows, 2734 instances of "no lane change" along with 340 "right lane change" and 217 "left lane change" instances.

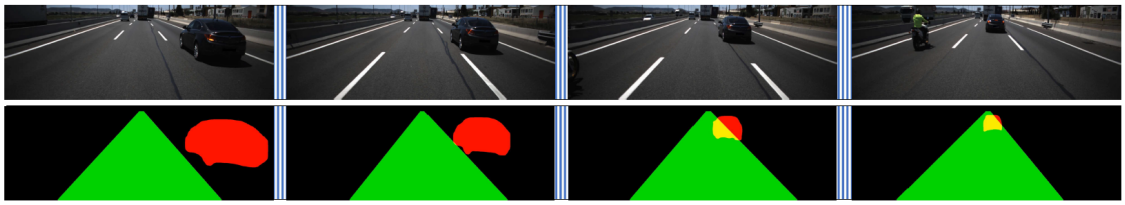

Figure 6.2. Generation of target-based scene representation with an example left lane change maneuver from Prevention Dataset. Overlapping masks of vehicles and ego-lane are in different colors. The figure shows four frames of a single sequence, whereas the LSTM Network and 3D network use 20 of them for classification. The frame height is reduced from 600 to 400 pixels to remove the ego vehicle's hood and some sky.

6.1 Experimental Results

Firstly, we conduct a comparative analysis of the standalone classification performances of three distinct approaches. This comparison is based on the highest fold and 5-fold cross-validation accuracies and the F1 scores on the lane change classification task (refer to Table 6.1.). This evaluation allows us to assess the effectiveness of each approach in isolation, providing a clear benchmark for their capabilities.

Table 6.1. 5-fold cross-validation results of image coordinate-based LSTM (LSTM3class) approach (Chapter 4) and self-supervised representation learning approach (Chapter 5) on the Prevention dataset.

Backbone	**Best fold acc (%)**	**5-fold CV acc (%)**	**Best fold F1-score**	**5-fold CV F1-score**
LSTM3class	91.83	89.90	81.60	79.68
R3D-18	92.89	92.13	75.38	73.13
R3D-18+MLP	93.06	91.93	75.90	72.63

The ensemble results of the LSTM3class model with two distinct self-supervised video representation learning models are provided in Figures 6.3. and 6.4.. Here, R3D signifies the standalone result of the self-supervised ResNet3D-18 model and R3D+MLP represents the standalone result of the self-supervised ResNet3D-18+MLP model, while LSTM3class indicates the standalone result of the image coordinate-based LSTM network. The weights of the models in the ensemble method are denoted as [W_1, W_2], where W_1 is the weight applied to LSTM3class, and W_2 is the weight applied to the self-supervised method. Notably, the soft voting and [0.5, 0.5] weighting are represented by a single bar as they are the same.

When we examine the previous studies on the Prevention dataset, we see that lane change classification accuracies can reach at most 90% (Biparva et al. (2021)). According to results in Table 6.1., our video-based approach with the ResNet3D-18 model managed to improve upon the success of methods in the literature by $\sim 2\%$ in terms of the accuracy metric. LSTM3class alone is not able to exceed the best accuracy in the literature, but it boosted the accuracies when used in an ensemble with ResNet3D-18 or ResNet3D-18+MLP. As seen in Figures 6.3. and 6.4., with the best [W_1, W_2] pairs, ensembling reached $\%94.05$ accuracy ($\sim 2\%$ improvement) for the model using ResNet3D-18 and $\%94.25$ accuracy ($\sim \%2.5$ improvement) for the model using ResNet3D-18+MLP. Even if equal W_1 and W_2 are used (soft voting), the improvement in accuracy is significant when

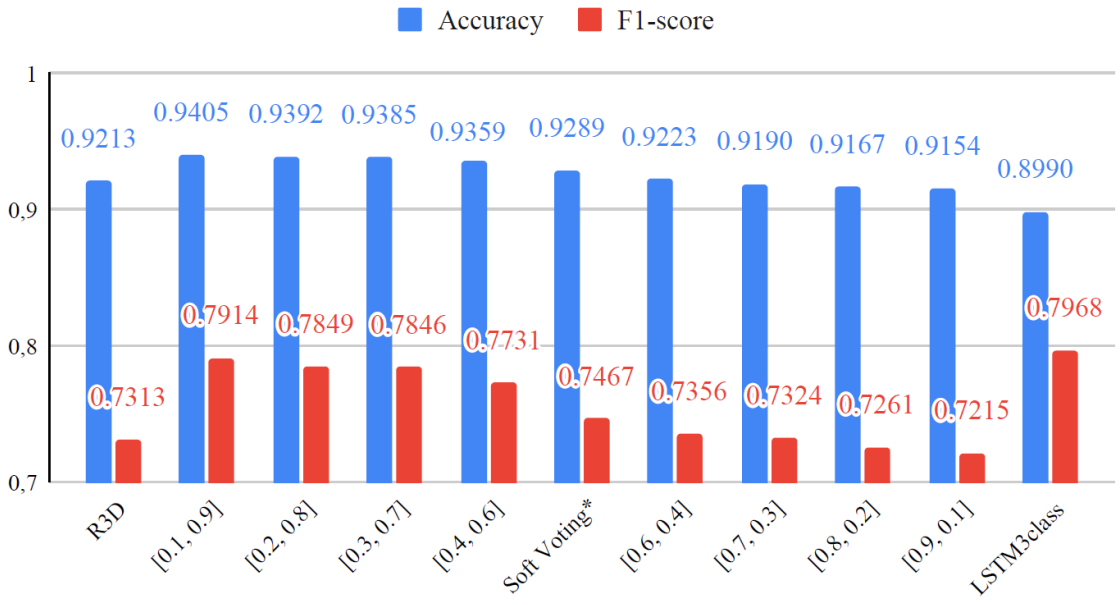

Figure 6.3. Ensemble learning results comparison of LSTM3class and ResNet3D-18(Self-sup.1) on Prevention Lane Change Prediction dataset. *Soft voting also means [0.5, 0.5] weighted average. 5-fold cross-validation was applied.

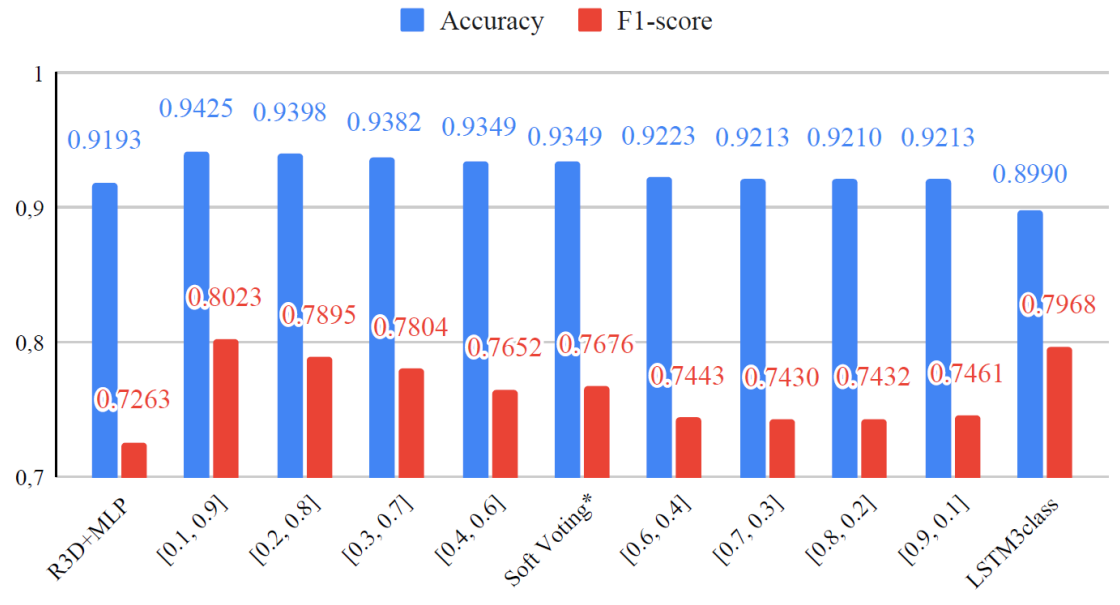

Figure 6.4. Ensemble learning results comparison of LSTM3class and ResNet3D-18+MLP(Self-sup.2) on Prevention Lane Change Prediction dataset. *Soft voting also means [0.5, 0.5] weighted average. 5-fold cross-validation was applied.

compared to standalone methods.

Moreover, in order to observe the contributions of each ensemble method to the class-based performances of the models in no lane change (NLC), left lane change (LLC),

and right lane change (RLC) categories, the confusion matrix results of each method from cross-validation have been provided in Table 6.2. and 6.3. This allows us to examine the strengths and weaknesses of the ensemble methods, by emphasizing how well each method performs in accurately classifying each lane change maneuver type.

Table 6.2. Class-based performances of applied ensemble learning techniques with LSTM3class and ResNet3D-18 (Self-sup.1) on Prevention Lane Change Prediction dataset (w: weight, *: f1-score).

True / Pred.	NLC	LLC	RLC	Precision
NLC	501	2	5	0.986
LLC	8	20	13	0.488
RLC	6	12	43	0.705
Recall	0.973	0.588	0.705	0.747*
Accuracy				0.929

(a) Soft Voting

True / Pred.	NLC	LLC	RLC	Precision
NLC	500	3	4	0.986
LLC	7	22	12	0.537
RLC	4	8	48	0.800
Recall	0.978	0.667	0.750	0.791*
Accuracy				0.940

(b) LSTM3class w: 0.1, R3D-18 w: 0.9

True / Pred.	NLC	LLC	RLC	Precision
NLC	500	3	4	0.986
LLC	8	20	13	0.488
RLC	4	8	48	0.800
Recall	0.977	0.645	0.738	0.785*
Accuracy				0.939

(c) LSTM3class w: 0.3, R3D-18 w: 0.7

True / Pred.	NLC	LLC	RLC	Precision
NLC	497	3	7	0.980
LLC	7	20	14	0.488
RLC	6	12	42	0.700
Recall	0.975	0.571	0.667	0.736*
Accuracy				0.922

(d) LSTM3class w: 0.6, R3D-18 w: 0.4

True / Pred.	NLC	LLC	RLC	Precision
NLC	495	4	9	0.974
LLC	7	21	14	0.500
RLC	6	12	42	0.700
Recall	0.974	0.568	0.646	0.732*
Accuracy				0.919

(e) LSTM3class w: 0.7, R3D-18 w: 0.3

True / Pred.	NLC	LLC	RLC	Precision
NLC	494	4	9	0.974
LLC	7	20	14	0.488
RLC	6	13	41	0.683
Recall	0.974	0.541	0.641	0.722*
Accuracy				0.915

(f) LSTM3class w: 0.9, R3D-18 w: 0.1

Due to the imbalanced nature of the Prevention lane change detection dataset, the LSTM3class method and self-supervised contrastive learning method struggled to distinguish left lane change and right lane change maneuvers from the video sequences in the no lane change class. In addition, despite experimenting with various augmentations

Table 6.3. Class-based performances of applied ensemble learning techniques with LSTM3class and ResNet3D-18+MLP (Self-sup.2) on Prevention Lane Change Prediction dataset (w: weight).

Pred \ True	NLC	LLC	RLC	Precision
NLC	500	2	5	0.986
LLC	8	19	14	0.463
RLC	4	8	48	0.800
Recall	0.977	0.655	0.716	0.768*
Accuracy				0.935

(a) Soft Voting

Pred \ True	NLC	LLC	RLC	Precision
NLC	500	3	4	0.986
LLC	7	23	11	0.561
RLC	4	8	48	0.800
Recall	0.979	0.677	0.762	0.802*
Accuracy				0.943

(b) LSTM3class w: 0.1, R3D-18+MLP w: 0.9

Pred \ True	NLC	LLC	RLC	Precision
NLC	500	3	4	0.986
LLC	7	21	13	0.512
RLC	3	9	48	0.800
Recall	0.980	0.636	0.739	0.780*
Accuracy				0.938

(c) LSTM3class w: 0.3, R3D-18+MLP w: 0.7

Pred \ True	NLC	LLC	RLC	Precision
NLC	493	5	10	0.971
LLC	7	19	15	0.463
RLC	3	10	47	0.783
Recall	0.980	0.559	0.653	0.744*
Accuracy				0.922

(d) LSTM3class w: 0.6, R3D-18+MLP w: 0.4

Pred \ True	NLC	LLC	RLC	Precision
NLC	492	5	10	0.970
LLC	7	20	15	0.476
RLC	4	10	47	0.770
Recall	0.978	0.571	0.653	0.743*
Accuracy				0.921

(e) LSTM3class w: 0.7, R3D-18+MLP w: 0.3

Pred \ True	NLC	LLC	RLC	Precision
NLC	492	6	10	0.969
LLC	7	20	15	0.476
RLC	4	9	47	0.783
Recall	0.978	0.571	0.653	0.746*
Accuracy				0.921

(f) LSTM3class w: 0.9, R3D-18+MLP w: 0.1

during self-supervised training in an attempt to increase the success of the self-supervised contrastive learning model in recognizing right and left lane change maneuvers, we were unable to enhance the precision and recall performances for these two classes.

However, almost all ensemble methods we implemented with LSTM3class not only led to an increase in the overall accuracy rate but also allowed us to correctly classify some right and left lane change examples that we were unable to classify accurately when using the self-supervised method alone. This resulted in an improvement in the precision and recall values for these classes. This demonstrates the potential of ensemble methods in addressing classification challenges posed by imbalanced datasets and improving the performance of individual models.

Given the computational efficiency inherent to the LSTM-3class method, we considered it to report the ensemble computation times. To this end, we loaded both models onto the same GPUs and measured the inference time per sequence.

As illustrated in Table 6.4., the per sequence classification time is effectively real-time. This is because the input sequences correspond to a two-second video clip, allowing us to make a classification decision within 0.667 seconds. Vehicle and ego lane segmentation take most of the time, but it can still be considered real-time since it is below one second. This quick decision-making capability is crucial in autonomous driving systems, where timely and accurate maneuver detection is essential.

Table 6.4. Execution time comparison of evaluated methods.

Method	**Vehicle and Ego lane Segmentation (sec/seq)**	**Classification (sec/seq)**	**Total (sec/seq)**
LSTM-3class + ResNet3D-18	0.659*	0.011	0.6670
LSTM-3class + ResNet3D-18-MLP		0.012	0.6671

All evaluations are done on a GPU server with two parallel Nvidia Tesla P100 16 GB GPUs.
*Computation time of vehicle and ego lane segmentation operation given as total computation times of Detectron2 and YOLOPv2.

CHAPTER 7

CONCLUSIONS

In this thesis, we have explored the use of deep learning for predicting two safety-critical behaviors in autonomous vehicles: lane change and cut-in maneuvers. Our approach involved the analysis of video data from onboard cameras. Progressing from more straightforward to more complex methods, we have proposed three distinct solutions to these challenges, offering a comprehensive study on improving safety in autonomous vehicles.

Three distinct datasets were utilized in the evaluation of our methods. We employed the widely acknowledged benchmark for lane change detection, the open-source Prevention Dataset. Given the lack of publicly available datasets for cut-in maneuver detection, we constructed two unique datasets from the Berkeley Deep Drive and the Prevention datasets. These datasets were used to evaluate our methods for detecting cut-in and lane-pass maneuvers.

Our first method was based on the idea that lateral and longitudinal movements of vehicles can be good clues when we examine the nature of maneuvers. In this method, we designed an LSTM-based framework to recognize cut-in maneuvers. For this framework, the bounding box value of the target vehicle was detected and tracked with state-of-the-art models and used in the feature extraction step. In the feature extraction step, the center coordinates, width, and height information that we extracted from the bounding box of the target vehicle were used as input in the different LSTM architectures we proposed, and the maneuver performed by the target vehicle was recognized with this feature sequence. The LSTM architectures we proposed have achieved successful results compared to those in the existing literature for the cut-in maneuver detection task. However, when we attempted to enhance the LSTM network, we encountered a limitation in our proposed method's ability to generalize across diverse maneuver detection datasets. For maneuver types where the model typically demonstrated high accuracy, it began to make erroneous decisions when the data was altered. This indicated a limitation in the model's adaptability to different data conditions.

Moreover, our most successful LSTM model made decisions based on the side of the vehicle where the maneuver occurred. While the existing approach had its advantages, a method that could process the entire dataset without any prerequisite knowledge might offer enhanced efficiency. This led us to propose a new approach that does not rely on prior knowledge of the maneuver's location, thereby aiming to improve the model's generalizability and performance across varying datasets. Hence, as a second approach in

this thesis work, we developed a self-supervised method to propose a more consistent and robust model for maneuver detection.

Leveraging self-supervised learning as our second method, we proposed a framework for detecting lane change and cut-in maneuvers in front of the ego vehicle. This framework is utilized by self-supervised contrastive video representation learning methodology. Transitioning from an LSTM network to a self-supervised method offered multiple advantages. The contrastive learning technique allowed the proposed model to focus on essential features while introducing variability in less critical aspects, helping the model to recognize a wide range of maneuver patterns and generalize better to unseen data. Furthermore, by employing different augmentations for video representation learning, imitating different situations that may not be included in the labeled data set, thus helping the model to learn the spatial and temporal attributes of the scene. A significant advantage of this self-supervised learning approach is that it allows for the use of a large dataset without the need for maneuver labeling, which can be time-consuming and expensive.

The self-supervised contrastive video representation learning framework we presented had two phases. In the self-supervised training phase, the encoder network was pre-trained with unlabeled highway-recorded video clips with contrastive loss to learn vehicle maneuver representations and their interactions with the ego-lane. Then self-supervised trained backbone network was fine-tuned with a labeled dataset with different classification head types to classify the maneuver in the scene or maneuver of the target vehicle.

For the dataset we used to train this method, we developed a technique called simplified scene representation to enhance self-supervised learning. In that technique, videos were converted to high-level representations of the scene (simplified views), which is done by segmenting vehicles and ego-lane and subtracting the background. The same high-level representation extraction was applied to the prepared cut-in/lane-pass and lane change detection datasets.

This model demonstrated better performance than other alternative methods, including the evaluated supervised baselines and the secondary self-supervised approach. In the context of the lane change detection task, the ResNet3D-18 model, when integrated with a linear classification head, achieved the highest 5-fold cross-validation accuracy, marked at 92.13%, along with an F1 score of 73.13%. Moreover, for the cut-in maneuver detection task, on our BDD-100K cut-in/lane-pass classification subset, the ResNet3D-18 model, pre-trained via self-supervision and accompanied by a four-layer non-linear classification head, yielded an accuracy of 92.52%. This illustrates an enhancement of approximately $\sim 2\%$ over the closest supervised baseline. Similarly, in our Prevention cut-in/lane-pass classification subset, models pre-trained through self-supervision have demonstrated a significant improvement in accuracy and overall performance, surpassing supervised baselines by approximately 2%. These findings underline the potential bene-

fits and applicability of self-supervised learning approaches in the domain of autonomous driving systems.

This introduced novel framework was used for the first time in the literature for the detection of vehicle maneuvers. Our findings underscore the effectiveness of self-supervised training methodologies in maneuver detection, outperforming traditional supervised training across multiple tasks. Furthermore, we propose a technique of scene simplification, which, in contrast to the utilization of raw video data, has been shown to significantly enhance the performance of models.

The disadvantage of the simplified view method we propose is that it can be difficult to represent some maneuvers with the vehicle mask and ego lane. If the maneuvers to be detected are not associated with the ego lane, other features may be needed to learn the representation of this maneuver by the model. Indeed, in some examples related to right and left lane changes, the self-supervised method failed, and we were able to solve part of this problem with the decision-level ensemble learning we proposed.

Although the self-supervised method can be applied to any dataset, it shows dependency on the camera's characteristics. In other words, if there is a difference in camera resolution among the data, the vehicle mask and ego-lane dimensions may vary, making it challenging to use the data collected with different camera resolutions together in the model training. However, when data collected with the same camera resolution was added to self-supervised training, we observed an increase in success.

When we examined the impact of the augmentations we applied to self-supervised contrastive learning, we observed that the horizontal flip, which can diversify side information in the cut-in maneuver detection task, and random shear and random rotate, which can mimic the changes in the ego lane during turns, had a positive impact on accuracy performance. However, while the horizontal flip was the most contributing augmentation for cut-in maneuver detection, it served as an intra-negative augmentation in the lane change detection task, where different classes (right lane change & left lane change) represent the side on which the maneuver occurred. We also determined that the temporal elastic transformation, which we used as temporal augmentation, improved performance with its ability to mimic changes in vehicle speed. Even though the target-based and intra-negative augmentations we tested as ablation studies contributed to the performance, they could not achieve as successful results as the version where intra-positive augmentations were used alone.

In the domain of autonomous vehicles, the application of ensemble learning methodologies is of paramount importance for bolstering the safety and decision-making efficacy of these systems. By leveraging the collective strengths of multiple deep learning models, it significantly improves the overall performance and accuracy of vehicle maneuver detection, a critical factor in ensuring safe and effective autonomous driving.

Therefore, as a final step in our study, we integrate the two successful method-

ologies that we proposed in this thesis using two different ensemble techniques. This approach is then evaluated for its effectiveness in the lane change maneuver detection task. We used soft voting and weighted sum voting as ensemble techniques. For the soft voting, each model's prediction probability per class is summed and the class with the maximum prediction probability was selected as the mutual prediction. For weighted sum voting, the weight parameters for each model were selected from 0.1 to 0.9, necessitating that the cumulative sum of the weights equates to 1. This condition ensured a balanced contribution from both models in the ensemble. Both ensemble methods achieved state-of-the-art performance for the lane change detection task by at least a 2% increase over the performance of the standalone models. Moreover, both ensemble methods improved the class-based classification performance of the models, achieving higher recall and precision metric values than the standalone models. The observed increase in performance can be attributed to the synergistic effect of the ensemble of the LSTM-based method and the self-supervised pre-trained method. Individually, both these methods encountered difficulties in distinguishing between right and left lane change maneuvers. This challenge can be attributed to the inherent complexities in differentiating these maneuvers based on the available data, mainly when the data is subject to variations in factors such as lighting conditions, vehicle speed, and distance from the ego vehicle.

However, when these two models were combined in an ensemble, they were able to compensate for each other's weaknesses to a certain degree. The LSTM-based method, with its ability to capture temporal dependencies in the data, complemented the self-supervised pre-trained method, which excels in extracting useful features from the raw data. This combination allowed the ensemble to leverage the strengths of both methods, leading to a more robust and accurate maneuver detection system.

To present the numerical results, with the best W_1 and W_2 pairs on weighted sum voting, the ensemble achieves an accuracy of 94.05% ($\sim 2\%$ improvement) for the model using ResNet3D-18 and an accuracy of 94.25% ($\sim 2.5\%$ improvement) for the model using ResNet3D-18+MLP. Even with equal W_1 and W_2 (soft voting), the improvement in accuracy is significant compared to the standalone methods.

As a potential future research, in order to extend the applicability and robustness of our approach in real-world traffic environments, we plan to investigate the efficacy of our proposed self-supervised video representation learning method for maneuvers occurring in traffic scenarios that differ from those considered in this study. Expanding the scope of our method to incorporate a broader range of maneuvers and diverse traffic elements - including motorbikes, bicycles, and pedestrians - could provide valuable insights and enhance our ability to detect maneuvers. This endeavor would involve developing a more comprehensive model capable of effectively analyzing and identifying maneuvers for various types of vehicles and other factors related to traffic. By addressing these aspects. We can expand the utility of our approach within different real-world traffic settings.

REFERENCES

Alahi, A., K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese (2016). Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 961–971.

Aliakbarian, M. S., F. S. Saleh, M. Salzmann, B. Fernando, L. Petersson, and L. Andersson (2018). Viena2: A driving anticipation dataset. In *Asian Conference on Computer Vision (ACCV)*.

Altché, F. and A. de La Fortelle (2017). An lstm network for highway trajectory prediction. In *International Conference on Intelligent Transportation Systems (ITSC)*.

Barth, A. and U. Franke (2010). Tracking oncoming and turning vehicles at intersections. In *13th International IEEE Conference on Intelligent Transportation Systems*, pp. 861–868. IEEE.

Bastanlar, Y. and S. Orhan (2022). Self-supervised contrastive representation learning in computer vision. In *Artificial Intelligence - Annual Volume 2022, IntechOpen.*

Biparva, M., D. Fernández-Llorca, R. Izquierdo-Gonzalo, and J. K. Tsotsos (2021). Video action recognition for lane-change classification and prediction of surrounding vehicles. Preprint at https://arxiv.org/abs/2101.05043.

Bochkovskiy, A., C.-Y. Wang, and H.-Y. M. Liao (2020). Yolov4: Optimal speed and accuracy of object detection. Preprint at https://arxiv.org/abs/2004.10934.

Chen, T., S. Kornblith, M. Norouzi, and G. Hinton (2020). A simple framework for contrastive learning of visual representations. In *The International Conference on Machine Learning (ICML)*.

Chen, X. and K. He (2021). Exploring simple siamese representation learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*.

Chen, X., H. Zhang, F. Zhao, Y. Hu, C. Tan, and J. Yang (2022). Intention-aware vehicle trajectory prediction based on spatial-temporal dynamic attention network for internet of vehicles. *IEEE Transactions on Intelligent Transportation Systems 23*(10), 19471–19483.

Chen, Y., C. Hu, and J. Wang (2019). Human-centered trajectory tracking control for autonomous vehicles with driver cut-in behavior prediction. *IEEE Transactions on Vehicular Technology 68*(9), 8461–8471.

Dai, S., L. Li, and Z. Li (2019). Modeling vehicle interactions via modified lstm models for trajectory prediction. *IEEE Access 7*, 38287–38296.

Deo, N., A. Rangesh, and M. M. Trivedi (2018). How would surround vehicles move? a unified framework for maneuver classification and motion prediction. In *IEEE Transactions on Intelligent Vehicles*, Volume 3, pp. 129–140.

Deo, N. and M. M. Trivedi (2018). Convolutional social pooling for vehicle trajectory prediction. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 1468–1476.

Ding, W. and S. Shen (2019). Online vehicle trajectory prediction using policy anticipation network and optimization-based context reasoning. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 9610–9616. IEEE.

Fang, J., F. Wang, P. Shen, Z. Zheng, J. Xue, and T.-s. Chua (2022). Behavioral intention prediction in driving scenes: A survey. Preprint at https://arxiv.org/abs/2211.00385.

Fang, J., D. Yan, J. Qiao, J. Xue, and H. Yu (2021). Dada: Driver attention prediction in driving accident scenarios. *IEEE Transactions on Intelligent Transportation Systems 23*(6), 4959–4971.

Fernández-Llorca, D., M. Biparva, R. Izquierdo-Gonzalo, and J. K. Tsotsos (2020). Two-stream networks for lane-change prediction of surrounding vehicles. In *International Conference on Intelligent Transportation Systems (ITSC)*.

Galvani, M. (2019). History and future of driver assistance. *IEEE Instrumentation & Measurement Magazine 22*(1), 11–16.

Ganaie, M. A., M. Hu, A. K. Malik, M. Tanveer, and P. N. Suganthan (2022). Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence 115*, 105151.

Girase, H., H. Gang, S. Malla, J. Li, A. Kanehara, K. Mangalam, and C. Choi (2021). Loki: Long term and key intentions for trajectory prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9803–9812.

Graves, A. (2013). Generating sequences with recurrent neural networks. Preprint at https://arxiv.org/abs/1308.0850.

Grill, J.-B., F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, B. Piot, K. Kavukcuoglu, R. Munos, and M. Valko (2020). Bootstrap your own latent: A new approach to self-supervised learning. *Advances in neural information processing systems 33*, 21271–21284.

Han, C., Q. Zhao, S. Zhang, Y. Chen, Z. Zhang, and J. Yuan (2022). Yolopv2: Better, faster, stronger for panoptic driving perception. Preprint at https://arxiv.org/abs/2208.11434.

Han, T., W. Xie, and A. Zisserman (2020). Self-supervised co-training for video representation learning. *Advances in Neural Information Processing Systems 33*, 5679–5690.

Hara, K., H. Kataoka, and Y. Satoh (2018). Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet? In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*.

He, K., H. Fan, Y. Wu, S. Xie, and R. Girshick (2020). Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*.

Huang, G., Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft, and K. Q. Weinberger (2017). Snapshot ensembles: Train 1, get m for free. Preprint at https://arxiv.org/abs/1704.00109.

Huang, Y., J. Du, Z. Yang, Z. Zhou, L. Zhang, and H. Chen (2022). A survey on trajectory-prediction methods for autonomous driving. *IEEE Transactions on Intelligent Vehicles 7*(3), 652–674.

III (2019). Report: Facts + statistics: Highway safety. `https://www.iii.org/fact-statistic/facts-statistics-highway-safety`. Accessed: 2023-05-19.

Izquierdo, R., A. Quintanar, I. Parra, D. Fernández-Llorca, and M. Sotelo (2019a). Experimental validation of lane-change intention prediction methodologies based on cnn and lstm. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pp. 3657–3662. IEEE.

Izquierdo, R., A. Quintanar, I. Parra, D. Fernández-Llorca, and M. Sotelo (2019b). The prevention dataset: A novel benchmark for prediction of vehicles intentions. In *International Conference on Intelligent Transportation Systems (ITSC)*.

Jeong, Y. and K. Yi (2020). Bidirectional long short-term memory-based interactive motion prediction of cut-in vehicles in urban environments. *IEEE Access 8*, 106183–106197.

Kasper, D., G. Weidl, T. Dang, G. Breuel, A. Tamke, A. Wedel, and W. Rosenstiel (2012). Object-oriented bayesian networks for detection of lane change maneuvers. In *IEEE Intelligent Transportation Systems Magazine*, Volume 4, pp. 19–31.

Kingma, D. P. and J. Ba (2014). Adam: A method for stochastic optimization. Preprint at https://arxiv.org/abs/1412.6980.

Knights, J., B. Harwood, D. Ward, A. Vanderkop, O. Mackenzie-Ross, and P. Moghadam (2021). Temporally coherent embeddings for self-supervised video representation learning. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 8914–8921. IEEE.

Laimona, O., M. A. Manzour, O. M. Shehata, and E. I. Morgan (2020). Implementation and evaluation of an enhanced intention prediction algorithm for lane-changing scenarios on highway roads. In *2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES)*, pp. 128–133.

Le-Khac, P. H., G. Healy, and A. F. Smeaton (2020). Contrastive representation learning: A framework and review. *IEEE Access 8*, 193907–193934.

Lee, D., Y. P. Kwon, S. McMains, and J. K. Hedrick (2017). Convolution neural network-based lane change intention prediction of surrounding vehicles for acc. In *International Conference on Intelligent Transportation Systems (ITSC)*.

Li, B., K. Q. Weinberger, S. Belongie, V. Koltun, and R. R (2022). Language-driven semantic segmentation. In *International Conference on Learning Representations (ICLR)*.

Lin, Y., X. Guo, and Y. Lu (2021). Self-supervised video representation learning with meta-contrastive network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8239–8249.

Nalcakan, Y. and Y. Bastanlar (2022). Monocular vision-based prediction of cut-in maneuvers with lstm networks. Preprint at https://arxiv.org/abs/2203.10707.

Nalcakan, Y. and Y. Bastanlar (2023). Cut-in maneuver detection with self-supervised contrastive video representation learning. *Signal, Image and Video Processing 17*, 29151–2923.

Oord, A. v. d., Y. Li, and O. Vinyals (2018). Representation learning with contrastive predictive coding. Preprint at https://arxiv.org/abs/1807.03748.

Orhan, S., J. Guerrero, and Y. Bastanlar (2022). Semantic pose verification for outdoor visual localization with self- supervised contrastive learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Park, S. H., B. Kim, C. M. Kang, C. C. Chung, and J. W. Choi (2018). Sequence-to-sequence prediction of vehicle trajectory via lstm encoder-decoder architecture. In *2018 IEEE intelligent vehicles symposium (IV)*, pp. 1672–1678. IEEE.

Peng, X., R. Liu, Y. L. Murphey, S. Stent, and Y. Li (2018). Driving maneuver detection via sequence learning from vehicle signals and video images. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pp. 1265–1270. IEEE.

Phillips, D. J., T. A. Wheeler, and M. J. Kochenderfer (2017). Generalizable intention prediction of human drivers at intersections. In *2017 IEEE intelligent vehicles symposium (IV)*, pp. 1665–1670. IEEE.

Qian, R., T. Meng, B. Gong, M.-H. Yang, H. Wang, S. Belongie, and Y. Cui (2021). Spatiotemporal contrastive video representation learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*.

Radford, A., J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, and S. Agarwal (2021). Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*.

SAEJ3016 (2021). Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles j3016_202104. `https://www.sae.org/standards/content/j3016_202104/`. Accessed: 2023-05-19.

Sandler, M., A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 4510–4520.

Scheel, O., N. S. Nagaraja, L. Schwarz, N. Navab, and F. Tombari (2019). Attention-based lane change prediction. In *ICRA*.

Simoncini, M., D. C. de Andrade, L. Taccari, S. Salti, L. Kubin, F. Schoen, and F. Sambo (2022). Unsafe maneuver classification from dashcam video and gps/imu sensors using spatio-temporal attention selector. *IEEE Transactions on Intelligent Transportation Systems 23*(9), 15605–15615.

Sivaraman, S., B. Morris, and M. Trivedi (2011). Learning multi-lane trajectories using vehicle-based vision. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pp. 2070–2076. IEEE.

Sivaraman, S. and M. M. Trivedi (2011). Combining monocular and stereo-vision for real-time vehicle ranging and tracking on multilane highways. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 1249–1254. IEEE.

Sivaraman, S. and M. M. Trivedi (2013). Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis. *IEEE transactions on intelligent transportation systems 14*(4), 1773–1795.

Song, R. and B. Li (2022). Surrounding vehicles' lane change maneuver prediction and detection for intelligent vehicles: A comprehensive review. *Trans. Intell. Transport. Sys. 23*(7), 6046–6062.

Stamoulakatos, A., J. Cardona, C. Michie, I. Andonovic, P. Lazaridis, X. Bellekens, R. Atkinson, M. M. Hossain, and C. Tachtatzis (2021). A comparison of the performance of 2d and 3d convolutional neural networks for subsea survey video classification. In *OCEANS 2021: San Diego–Porto*, pp. 1–10. IEEE.

Sutskever, I., O. Vinyals, and Q. V. Le (2014). Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pp. 3104–3112. MIT Press.

Tao, L., X. Wang, and T. Yamasaki (2020). Self-supervised video representation learning using inter-intra contrastive framework. In *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 2193–2201.

Tekir, S. and Y. Bastanlar (2020). Deep learning: Exemplar studies in natural language processing and computer vision. In *Data Mining Methods, Applications and Systems, IntechOpen*.

US101 (2007). U.S. Federal Highway Administration - US Highway 101 dataset. `https://ops.fhwa.dot.gov/trafficanalysistools/ngsim.htm`. Accessed: 2021-06-30.

Wang, W., T. Qie, C. Yang, W. Liu, C. Xiang, and K. Huang (2021). An intelligent lane-changing behavior prediction and decision-making strategy for an autonomous vehicle. *IEEE Transactions on Industrial Electronics 69*(3), 2927–2937.

Wojke, N., A. Bewley, and D. Paulus (2017). Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, pp. 3645–3649.

Wu, Y., A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick (2019). Detectron2. `https://github.com/facebookresearch/detectron2`.

Xin, L., P. Wang, C.-Y. Chan, J. Chen, S. E. Li, and B. Cheng (2018). Intention-aware long horizon trajectory prediction of surrounding vehicles using dual lstm networks. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1441–1446. IEEE.

Xue, J., J. Fang, T. Li, B. Zhang, P. Zhang, Z. Ye, and J. Dou (2019). Blvd: Building a large-scale 5d semantics benchmark for autonomous driving. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 6685–6691. IEEE.

Yoon, Y., C. Kim, J. Lee, and K. Yi (2021). Interaction-aware probabilistic trajectory prediction of cut-in vehicles using gaussian process for proactive control of autonomous vehicles. *IEEE Access 9*, 63440–63455.

Yu, F., H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell (2020). Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*.

Yurtsever, E., Y. Liu, J. Lambert, C. Miyajima, E. Takeuchi, K. Takeda, and J. H. Hansen (2019). Risky action recognition in lane change video clips using deep spatiotemporal networks with segmentation mask transfer. In *International Conference on Intelligent Transportation Systems (ITSC)*.

Zhan, W., L. Sun, D. Wang, H. Shi, A. Clausse, M. Naumann, J. Kummerle, H. Konigshof, C. Stiller, A. de La Fortelle, et al. (2019). Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps. Preprint at https://arxiv.org/abs/1910.03088.

Zhang, C. and Y. Ma (2012). *Ensemble machine learning: methods and applications*. Springer.

Zhang, T., W. Song, M. Fu, Y. Yang, and M. Wang (2021). Vehicle motion prediction at intersections based on the turning intention and prior trajectories model. *IEEE/CAA Journal of Automatica Sinica 8*(10), 1657–1666.

Zhang, Y., Y. Zou, J. Tang, and J. Liang (2020). A lane-changing prediction method based on temporal convolution network. Preprint at https://arxiv.org/abs/2101.05043.

Zyner, A., S. Worrall, and E. Nebot (2018). A recurrent neural network solution for predicting driver intention at unsignalized intersections. *IEEE Robotics and Automation Letters 3*(3), 1759–1764.

Zyner, A., S. Worrall, and E. Nebot (2019). Naturalistic driver intention and path prediction using recurrent neural networks. *IEEE transactions on intelligent transportation systems 21*(4), 1584–1594.

Zyner, A., S. Worrall, J. Ward, and E. Nebot (2017). Long short term memory for driver intent prediction. In *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1484–1489. IEEE.

VITA

Yağız NALÇAKAN

Education

- **M.Sc.:** 2017-2019, Istanbul University-Cerrahpasa, Faculty of Engineering, Computer Engineering
  Thesis Title: "Diagnosis of Alzheimer's Disease with Deep Learning"
  Advisor: Asst. Prof. Dr. Tolga Ensari
- **B.Sc.:** 2011-2015, Trakya University, Faculty of Engineering, Computer Engineering

Academic Experience

- 2020-2023, İzmir Institute of Technology, Research Assistant
- 2022, Seoul National University, Visiting Research Fellow
- 2017-2019, İstanbul Altinbas University, Research Assistant

Publications

- **Nalcakan, Y.**, & Bastanlar, Y. (2023). Cut-in maneuver detection with self-supervised contrastive video representation learning. Signal, Image and Video Processing, 1-9.
- **Nalcakan, Y.**, & Bastanlar, Y. (2022). Monocular Vision-based Prediction of Cut-in Maneuvers with LSTM Networks. International Conference on Science, Engineering Management and Information Technology, held March 4, 2022.
- Yılmaz, R., **Nalçakan, Y.**, & Haktanır, E. (2022). A novel feature to predict buggy changes in a software system. International Conference on Intelligent and Fuzzy Systems, held August 24-26, 2021.
- Aşıroğlu, B., Mete, B. R., Yıldız, E., **Nalçakan, Y.**, Sezen, A., Dağtekin, M., & Ensari, T. (2019, April). Automatic HTML code generation from mock-up images using machine learning techniques. In 2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT) (pp. 1-4). IEEE.
- **Nalçakan, Y.**, & Ensari, T. (2019). Decision of neural networks hyperparameters with a population-based algorithm. 4th International Conference In Machine Learning, Optimization, and Data Science, September 13-16, 2018.