# DISTRACTED DRIVING DETECTION: VIDEO ANALYTICS PIPELINE WITH ADAPTIVE DRIVER FEATURE EXTRACTION

BY

YUZHONG DENG

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Computer Science
in the Graduate College of the
University of Illinois Urbana-Champaign, 2022

Urbana, Illinois

Adviser:

Professor Klara Nahrstedt

## ABSTRACT

Distracted driving has long been regarded as the main culprit of road accidents. Every two hours in the United States, a person's life is taken away by car crashes because of a distracted driver. Given the severity of distracted driving, there has been an increasing academic interest in finding a suitable method to detect and prevent distracted driving behaviors in real-time before tragedy hits. In this thesis, we provide a thorough review of the existing literature on distracted driving detection. Among the existing methods, there are two popular approaches: (1) use of specialized sensors to collect and monitor relevant driver information that may indicate a distracted driver; and (2) use of a smart camera to capture driver images and an on-camera video analytics pipeline to detect distracting behaviors in these driver images.

Inspired by these two approaches, we present a novel neural network architecture that can be used as a video analytics pipeline to detect distracting driving behaviors in real-time on the smart camera. Our model architecture incorporates a primary CNN neural network with a selected set of driver features (e.g., driver head pose, eye movement, and hand position). A novel contrastive training mechanism is introduced for the model to learn a compact representation of these external driver features. We have found that the learned driver embedding could then be used in downstream tasks such as distracted driving classification and is shown to be more effective compared to existing neural network methods. Most notably, our model is shown to be more robust against unseen drivers and distracted driving behaviors during test time, which makes our model a suitable candidate for real world application.

In summary, our thesis contributes to the current knowledge of distracted driving detection by providing an in-depth evaluation of the existing solutions. We also contribute to the existing study by identifying areas of improvements for the existing video-based analytics pipelines and presenting a new model architecture that could serve as an incremental improvement to the existing pipelines.

## ACKNOWLEDGMENTS

This thesis has been long in coming, and it would not have come to fruition but for the unceasing support and guidance I have received from my advisor Professor Klara Nahrstedt. I am deeply grateful and honored to have her as my thesis advisor. Her deep knowledge and research interests across different areas of edge computing have never ceased to inspire me. I also want to thank Professor Klara for her mentoring lessons and the occasional life tips that I picked up along the way.

I also want to express my appreciation to the MONET members for maintaining and promoting a great avenue to exchange ideas and share insights. I have personally learned a lot through our group's weekly meetings.

Finally, I could not have completed this work without the love and support from my partner and best friend, Joanna Zhang, who has always been there with me through the ups and downs of my master's study.

## TABLE OF CONTENTS

# CHAPTER 1: INTRODUCTION

Distracted driving is often considered as one of the main causes of road accidents. Studies have shown that distracted driving causes 3,000 deaths and 900,000 car accidents every year, [1] [2]. Given the severity of distracted driving, policy makers have been actively trying to mitigate this problem by raising public awareness and legislating laws. Commercial solutions have also been introduced to help prevent distracted driving, most notable of which is the advanced driver assistance systems (ADAS) that offer driver safety guidance such as lane departure assistance and forward collision warning. Despite these collective effort, distracted driving related deaths and incidents have remained high (see **Table 1.1**). A recent study even found that these ADAS systems can have unintended consequences of encouraging distracted driving behaviors, as drivers in over reliance on these ADAS systems are more likely to allow themselves to be distracted while driving [3].

In academics, currently there are two popular approaches to this distracted driving detection problem. The first approach attempted to detect distracted driving by identifying and monitoring key features that are correlated with a distracted driver. These key features are usually collected via some types of specialized sensors. For instance, hand gestures are often thought to contain important spatial information about the driver's current activities [4] [5]. If a driver's hand is away from the steering wheel, then it is likely that the driver is engaging in a non-driving-related task. Other commonly used features include head pose [6], eye movement [7] [8], and even feet movement [9]. Another popular approach involves a neural network to analyze driver images to detect any distracted behaviors. Unlike the sensor-based approach, the primary source of input for this approach comes from an interior camera or a mounted smart phone capturing driver actions during a trip.

Inspired by the sensor-based methods that studied the relationship between various driver features (e.g., hand pose) and a distracted driver (e.g., looking outside the window), we are interested in finding ways to incorporate different driver features into the existing neural network framework to robustly detect driver distraction. In this thesis, we present a novel neural network architecture that allows us to incorporate hand-crafted driver features as part of the model training process. Our model is intended to be included in a video analytics pipeline that can be run on a smart camera on the vehicle for real-time distracted driving detection. A novel contrastive training mechanism is also introduced to ensure that the model learns a comprehensive embedding representation from these selected driver features. We evaluate the performance of our proposed model against a neural network baseline model

Table 1.1: Distracted Driving Incidents and Deaths in Year 2010-2019

| | Statistics of Distracted Driving by Year | |
|---|---|---|
| **Year** | **Number of Deaths** | **Number of Accidents** |
| 2019 | 2,895 | 986,000 |
| 2018 | 2,645 | 938,000 |
| 2017 | 3,003 | 912,000 |
| 2016 | 3,197 | 905,000 |
| 2015 | 3,242 | 885,000 |
| 2014 | 2,972 | 967,000 |
| 2013 | 2,910 | 904,000 |
| 2012 | 3,098 | 908,000 |
| 2011 | 3,047 | 826,000 |
| 2010 | 2,993 | 900,000 |

*Note:* Reported numbers are taken from the NHTSA 2013 [1] and 2018 [2] research notes.

on a distracted driving detection dataset, and find that with similar training time, our model consistently outperforms the baseline model by a large margin. Our model is also shown to be more robust against unseen driver subjects and distracted driving behaviors during test time.

The rest of this thesis is structured as follows. **Chapter 2** describes the practical difference between driver inattention and driver distraction, and provides an in-depth review of the existing solutions. **Chapter 3** specifies the problem of interest within the study of distracted driving that this thesis aims to address, and considers the necessary requirements. Based on the insights drawn from evaluating the existing approaches, **Chapter 4** presents a new model architecture that would satisfy all problem requirements. **Chapter 5** discusses the model performance compared to the existing solutions. Finally, we conclude our study in **Chapter 6** with a summary of our findings and reflections on the lessons learned.

# CHAPTER 2: LITERATURE REVIEW

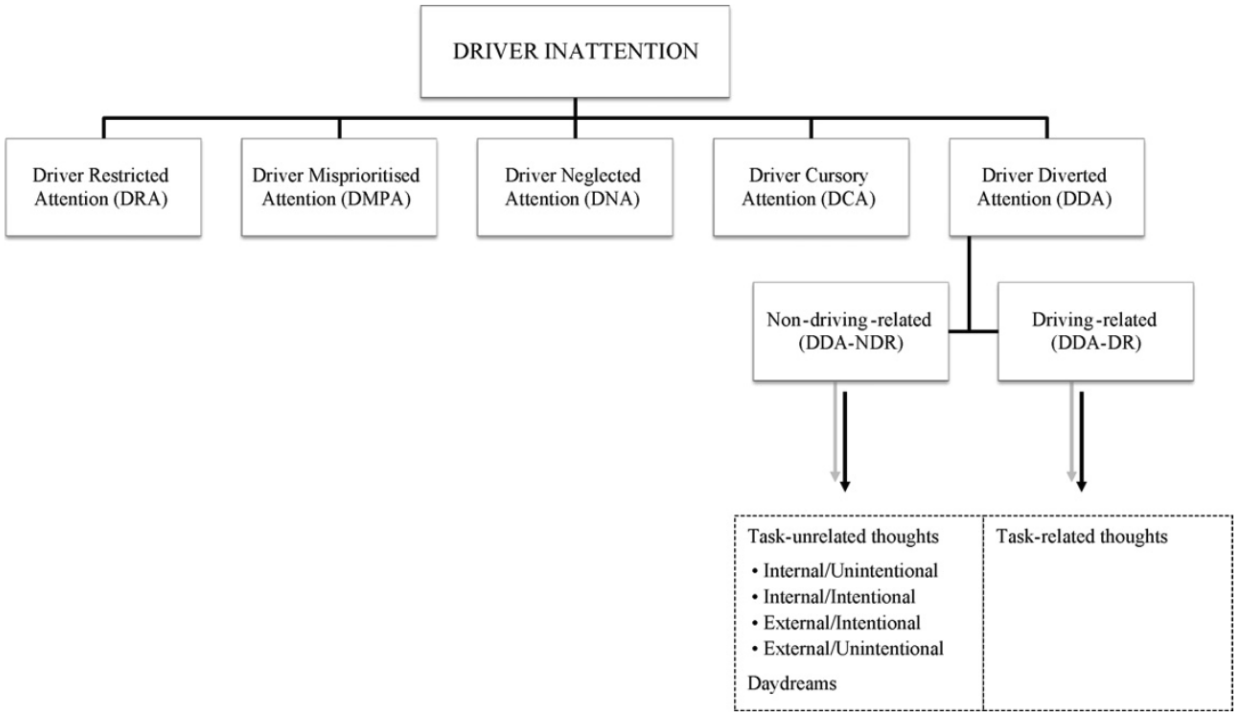
## 2.1 DRIVER INATTENTION AND DISTRACTION

Driving is a cognitively and physically complex task [10]. It requires drivers to have integrated control over the vehicle, with hands on the wheel, feet on the padel, and eyes on the road. Consider a common subtask of driving: a driver stops on the red light and wants to make a right turn. The driver needs to remember to use the turn signal to indicate his intention. Then, this driver would need to (1) look for vehicles on the perpendicular lines coming from the driver's left side, (2) check the blind spot on the right side of his vehicle for any passing cyclists or motorists, and (3) yield for any crossing pedestrians, all of which needs to be completed within a short period of time while controlling of the vehicle to actually make the turn. Making a right turn at an intersection is just one of the countless complex tasks that drivers have to perform during their trip. It is therefore not hard to imagine such high and continuous demands on attention can easily cause driver fatigue or loss of focus, with an elevated risk of being involved in a car accident.

The National Highway Traffic Safety Administration (NHTSA) [2] has identified driver inattention and driver distraction as the two main contributors of road accidents. Other government agencies such as the Federal Motor Carrier Safety Administration (FMCSA) [11] and the Centers for Disease Control and Prevention (CDC) [12], as well as research studies in academics [13] [14], also pointed to increasing evidence that driver inattention and distraction are major contributing factors in car and truck crashes. One of these academic studies conducted an in-depth study of the vehicle crash data occurring in Australia from 2000 through 2011 [13], and found that driver inattention and driver distraction account for three out of five samples examined in this study (i.e., 60 percent of the samples show evidence of driver inattention or distraction prior to the crash).

Moreover, the study also found that the most common types of distraction are voluntary non-driving related distraction, which accounts for about ten percent of the crashes. These voluntary distracting behaviors include interacting with passengers (most common), using vehicle systems (second most common), and mobile usage (third most common). A separate study looking at the types of non-driving activities that drivers would perform across over 2000 participants came up with a similar finding that the most frequent distracting activities are phone usage, interaction with passengers, and interaction with car systems. Furthermore, the researchers in this study noted that drivers in different age and ethnic groups are equally

Figure 2.1: Taxonomy of Driver Inattention




*Note:* Taxonomy and definitions of driver inattention are from Regan et. al. [15].

likely to be distracted while driving.

Although the term driver inattention is often used interchangeably with driver distraction in these studies, there is a subtle difference between inattention and distraction. In [14], driver distraction is defined as occurring when "...a driver has chosen to engage in a secondary task that is not necessary to perform the primary driving task." Driver inattention, on the other hand, is a superset of driver distraction and also includes other forms of behaviors, as specified in a driver inattention taxonomy defined in [15]. More specifically, driver distraction is thought to be synonymous with the Driver Diverted Attention (DDA) subcategory in driver inattention (for more details, see **Figure 2.1**). In this taxonomy, driver distraction is distinguished from other forms of driver inattention. For example, drivers might be unable to attend to activities critical for safe driving due to uncontrollable factors such as sudden illness (heart attack) or environmental stimulus (change blindness). Or drivers could be inattentive to primary driving activities while conducting a secondary task critical for safe driving (e.g., check bind spots for vehicles before changing lanes). In both of these scenarios, the driver is clearly inattentive to the main driving task, but they are not considered as distracted driving. See **Table 2.1** for more examples of other forms of inattentive driving not considered as distracted driving.

Table 2.1: Definition of each Category of Inattentive Driving

| Category | Cause of Inattention | Example | Is Distracted Driving? |
|---|---|---|---|
| Driver Restricted Attention (DRA) | "[S]omething that physically prevents (due to biological factors) the driver..." | A driver missed that the traffic lights have already turned red as he fell into moments of micro sleeps. | No |
| Driver Neglected Attention (DNA) | Driver neglecting to perform the activities that are critical for safe driving. | A driver forgot to check for blind spots when making a right turn at the intersection and therefore did not see the cyclist on the bike path. | No |
| Driver Cursory Attention (DCA) | Driver giving insufficient attention to the activities that are critical for safe driving. | A driver who is in a rush and does not fully check for approaching vehicles from the rear side when merging to the main lane and therefore caused collision. | No |
| Driver Diverted Attention (DDA) | Driver diverting his attention to something else from the activities that are critical for safe driving. | A driver turns his head to talk to the passengers in the backseat and does not notice the vehicle in the front has made a harsh stop, thereby causing a forward collision. | Yes |

*Note:* Taxonomy and definitions of driver inattention are from Regan et. al. [15].

Stepping aside from the semantic and theoretical differences between driver distraction and inattention [16] [17], the key characteristics that can distinguish driver distraction from other forms of driver inattention are (1) *intent*: whether the driver is willingly engaging in the secondary task, and (2) *necessity*: whether the secondary task that diverts driver attention from the road is critical for safe driving. For distracted driving, drivers are often thought to be willingly engaging in secondary activities that are unrelated or non essential to the primary task of safe driving.

## 2.2 DIFFERENT TYPES OF DISTRACTED DRIVING BEHAVIORS

Distracted driving can be further divided into three different types: visual distraction, manual distraction, and cognitive distraction [12], [16], [18]. Visual distractions involve drivers losing their visual focus from the road, such as viewing the navigation map on their phones or looking at a billboard advertisement. Manual distractions are activities that physically prevent drivers from controlling the vehicle (steering wheels and pedals). In most cases, manual distractions refers to drivers taking their hands off the wheel for other

activities such as eating and drinking. Lastly, cognitive distractions refer to activities that divert drivers' cognitive attention. Common sources of cognitive distraction include talking to passengers, listening to audio/podcasts, or even self engaging in task-unrelated thoughts such as daydreaming or worrying about family/work problems.

More often than not, drivers would engage in one or more secondary activities simultaneously that would distract drivers' ability to drive on multiple fronts. For example, a driver typing a new destination in the navigation app on his phone while driving would not only distract his mind (cognitive distraction), but also keep his hands off the wheel (manual distraction for typing the address) and eyes off the road (visual distraction for staring at the screen). Therefore, these non-driving activities have been identified as a serious safety concern in road safety. Recent studies have found that despite that driving itself is already a challenging task, it is not infrequent for drivers to voluntarily engage in other distracting tasks [19]. A 2016 study found that distracted driving accounted for about 68 percent of the crash events analyzed in the study [20]. Once again, use of electronic devices was identified as one of the most common sources of distraction that causes personal injuries and property damages in these crash accidents.

## 2.3 EARLY DISTRACTED DRIVING DETECTION METHODS

Given its detrimental impact on safety of the distracted drivers and others sharing the road, distracted driving has received significant academic interest. Research has been conducted to identify the underlying causes of distracted driving. Surveys and controlled studies have been conducted to investigate the relationship between distracted driving behaviors and other factors. Preliminary systems were proposed to help mitigate the frequency of distracted driving behaviors [21], [22]. These distraction detection systems were developed in the early days (late twentieth century) in which on-device video analytics systems were not yet mature and readily available. In addition, smart cameras during this time period were not only expensive but also hard to acquire in large quantities.

Therefore, these distracted driving systems generally shied away from video analytics and focused on alternative data sources that could be collected via sensors or other devices. For example, an optical transducer sensor is installed in the vehicle to monitor the steering wheel movements that would alert the drivers when the movement rate drops below a predefined threshold [21]. Several commercially available systems back then can monitor driver head nod angle via a velocity sensor. These systems would sound an alarm when the sensor detects

a significant droop of the driver's head, which may suggest the driver is asleep or in moments of micro sleep [21]. Interestingly, driver body activity, eye gaze, and head movement (not just in the vertical direction) were also cited in this 1985 paper as possible indicators of driver distraction. However, the paper mentioned these information are hard to collect and process via the then current sensor technology and analytics platform [21].

## 2.4 CURRENT SENSOR-BASED FEATURE EXTRACTION METHODS

Owing to the rapid improvement of sensor technologies and material science, more powerful sensors and devices have been manufactured in smaller sizes over the past two decades. Researchers now are able to collect, track, and monitor different metrics and related data from the drivers either in a simulated environment or during test driving. Therefore, we have witnessed a plethora of novel distracted driving detection mechanisms that were developed over the past two decades. Two central questions asked and solved in these system papers are (1) what type of data these sensors should collect and (2) how to use these data to identify and correct distracted driving behaviors before any safety hazard is realized.

**Detect manual distractions by monitoring vehicle interactions:** Because driving is a task that requires drivers' integrated body movements to control the vehicle in response to road conditions, there has been an active line of research on monitoring drivers' interaction with the steering wheel and petals, or lack thereof, to determine whether the drivers are in a distracted state. For instance, a 1999 study installed steering wheel position sensors on a special purpose driving simulator that can collect a time-series history of steering angle data for the test participants. Combined with the vehicle speed data and other information such as obstacle avoidance rate, the researchers use these time-series data to calculate the steering error during the course of a simulated trip, and found that the steering error rate has a positive correlation with the likelihood that the test participant is distracted [23]. Other vehicle features such as vehicle speed, Inertial Measurement Unit (IMU) data, and location GPS points might not contain as much information about driver distraction as steering error data. However, these feature values contain useful insights regarding driving style and vehicle behavior, and they are often used as complimentary data in conjunction with other types of sensor data to determine distracted driving.

**Detect manual distractions by monitoring driver body movements:** In a separate study [9], a foot gesture analysis is conducted to examine the possibility of using driver's foot movement data to predict the driving state (distracted or focused) and driving style

(aggressive or defensive) of that driver. A foot-looking camera is installed at the bottom of the driver seat to observe drivers' foot movements and their interactions with the pedals. Computer vision techniques are used to estimate the optical flow of driver feet of each frame. This information, together with the vehicle GPS and speed data, are used to model and predict foot gestures in naturalistic driving scenarios with near real-time speed (at 10 fps).

Hand gesture analyses are also used to understand distracted driving behaviors that involve drivers taking their hands off the steering wheel. Both [4] and [5] proposed a hand detection framework that is applicable to be used in the driving settings to detect driver hands and classify hand action. Both studies rely on the same VIVA hand database [24], which contains bounding boxes of driver hands and passenger hands under different illumination and from different viewpoints.

**Detect visual distractions by monitoring driver head and eye movements:** Analyses on drivers' head, eye, and body movement that were dismissed as "difficult to monitor [with] complex interactions" in [21] were now made possible by the advancement of various motion tracking devices. [6] proposed a novel computationally cheap method to estimate driver head pose in real time. Instead of relying on deep learning methods, conventional computer vision algorithms were used to detect three facial keypoints (center of two eyes and nose tip), which were then used to estimate driver head pose angle.

Similarly, driver eye movements are thought to contain crucial information regarding driver attention, especially whether drivers are affected by visual distractions. Driver gaze estimation has gained popularity as the sensor technology can now detect a person's presence and follow the person's eye movement in real-time [8], [7], and [25] all used similar eye tracking devices that can convert eye movements into a data stream that contains information such as gaze vector. Empowered by these eye tracking devices, they were able to analyze and evaluate the relationship between driver gaze behaviors and cognitive load of the driving task.

**Detect cognitive distractions by monitoring driver fatigue:** In addition to manual and visual distraction, drivers could be cognitively distracted by fatigue, emotion, or task-unrelated thoughts. There are existing research papers studying the effect of driver fatigue on road safety. Numerous frameworks have also been proposed to detect driver fatigue. [26] considered a video-based analytic framework to detect certain driver activities that could

be associated with driver fatigue. Such activities include excessive eye rubbing, blinking, and yawning. On the other hand, [26] attempted to detect driver fatigue through a voice-based detection system that analyzes the tonality, cadence, and pronunciation of the drivers' speech. Other methods try to collect drivers' biological data such as eyelid closure and heart rate via electroencephalography (EEG) [27] or electrocardiogram (ECG) [28] in order to measure and detect driver fatigue.

A summary of the existing feature extraction methods can be found in **Table 2.2**, where the intrusion level is estimated based on the descriptions of each feature collection process in a survey paper [29]. As can be see in the table, although driver features (e.g., head pose, eye movement) are shown to contain more information about driver distraction than vehicle features (e.g., speed, GPS data), driver features are harder to collect in actual driving environments. For example, all eye-tracking devices used in the paper reviewed above either required them to be worn by test subjects as an attachment to eyeglasses [7] [8], or are only tested under a controlled lab environment [25]. Moreover, these head-mounted eye tracking devices are often felt as intrusive because they require physical contact with the users.

## 2.5 CURRENT VIDEO-BASED ANALYTIC METHODS

Another type of sensor used for driver distraction detection is interior cameras. Under this setting, a smart camera, or a smartphone, is mounted inside the vehicle cabin facing the driver. The driver-facing camera would capture the driver's movements for the full duration of a trip. The captured video would then be analyzed to determine whether a distraction event has occurred. By far, interior cameras include the most comprehensive view of a driver's driving state at any given point in time (one can look at a single frame in isolation and determine whether the driver is distracted in that frame). Additionally, collecting data using these interior cameras are generally viewed as less intrusive than using wearable devices to collect driver features.

**Non-Neural-Network Approaches** In analyzing these driver-facing video footage, computer vision techniques are used to extract features that might be indicative of a distracted driver. For example, [30] extract drowsiness-related features such as eye closure and mouth yawning with metrics that are computationally efficient to obtain (e.g., eye aspect ratio (EAR) and mouth aspect ratio (MAR) metrics). Random Sample Consensus (RANSAC), a popular computer vision approach for estimating the fundamental matrix in stereo view, is used to determine driver's head pose with facial keypoints. With these inferred features, the

Table 2.2: Various Features Used for Distracted Driving Detection

| Feature | Test Environment | Sensor | Level of Intrusion |
|---|---|---|---|
| Head Pose | Simulated and Real-time | An absolute orientation sensor or a depth camera | Moderate |
| Eye Movement | Simulated | Eye-tracking devices attached to the driver's eyeglasses | High |
| Foot Gesture | Real-time | A foot-looking camera | Moderate |
| Hand Gesture | Simulated | A driver-looking camera | Moderate |
| Biological Metrics (e.g., Heart Rate) | Simulated | Electroencephalography (EEG) Electrocardiogram (ECG) | High |
| Vehicle Features (e.g., Steering Angle, Speed, Location Data) | Real-time | Embedded system sensors with the vehicle | Low |

*Note:*
1. Test environment is either simulated (conducted in a lab environment) or real-time (conducted in actual driving environment).
2. Intrusion level is one of Low, Moderate, or High, and is defined as the degree of intrusiveness that drivers feel about the sensor data collection process.

model was able to detect driver drowsiness based on heuristic measure of the EAR and MAR metrics, as well as driver looking at the right or left based on head pose information. Other non-deep-learning approaches were used to detect the presence of drivers' face and eyes in a video, such as the use of Haar Classifier Object detection in [26]. Because the methods used to collect driver features (e.g., head pose, eye location) in these papers are compute and memory efficient, the proposed models are able to run on an edge device with real-time detection speed.

**Neural-Network Approaches:** Deep learning frameworks are also used frequently in analyzing these driver-facing videos. To facilitate the research of using machine learning framework to detect distracted driving, numerous vision based driver monitoring datasets

have been introduced over the past decade. Two most popular datasets are (1) StateFarm distracted driver detection dataset [31] and (2) AUC Distracted Driver dataset [32]. StateFarm dataset is the first publicly available dataset for driver distraction classification that contains ten class labels (one for safety driving behavior and nine for distracting behaviors). AUC Distracted Driver dataset is an extension of the StateFarm dataset that contains additional training images with the same set of class labels. This AUC dataset is available only upon special request.

Another vision-based dataset for distracted driving is the Driver Anomaly Detection (DAD) dataset proposed in [33], which is intended to include a much broader set of distracting activities than the fixed 10 class labels provided in the StateFarm and AUC datasets. In addition to the common distracting behaviors such as phone usage and talking with passengers, this DAD dataset also includes other less common distracting activities such as rubbing eyes, taking on/off glasses, and head dropping. There are other hand-focused datasets for hand action recognition that can be used to detect driver hand position. Among them, the only dataset that is specific to the driving scenario is VIVA-hands [24], a multimodal dynamic hand gesture dataset specifically designed for studying natural human activities in real-world driving settings. VIVA-hands dataset contains about 900 videos by eight subjects with 19 different hand gesture labels.

Most papers that use a neural network view the challenge of detecting distracted driving behaviors as a traditional image classification problem [34] [35] [36]. Under this setting, the model is given an image of a driver, and the model is trying to make a binary prediction (whether or not the driver is distracted) or to select from one of the known, finite class labels (what kind of distracting activities). Because these machine learning models are usually run directly on the vehicle to detect distracted driving in real time, many studies choose a lightweight neural network architecture (such as MobileNet [37], ResNet 50 [38], Tiny Yolo [39]) that is built for embedded vision applications.

# CHAPTER 3: PROBLEM REQUIREMENTS AND DESCRIPTION

## 3.1 REQUIREMENTS FOR REAL WORLD APPLICATIONS

To answer the question of how to detect distracted driving behavior, one needs to answer a number of related questions: (1) what data do we need to collect, (2) what types of sensor do we need, and (3) how do we process the collected data and use them to detect distracted driving. The existing approaches that we evaluate in **Chapter 2** have their own answer to these three questions. For example, [7] and [8] collected driver eye movement data via wearable eye-tracking devices and used these data to predict driver distraction. Similar approaches have been made to use specialized sensors to obtain driver head pose and orientation information for distraction detection, [4],[5]. While these studies have shown promising results with hand-crafted features using specific sensors, others have elected to use machine learning approaches to automatically detect distraction behaviors given a stream of video images collected via a driver-facing interior camera. [35], [36].

However, there are unique requirements that these systems need to satisfy in order for them to be suitable in everyday use. To start, all models of any type need to be able to run directly on the vehicle, either on an edge device (such as a dashcam) inside the vehicle or embedded as part of the larger vehicle information system. Furthermore, these models need to be capable of making detections in real-time; otherwise just-in-time alarming would not be feasible to signal the drivers to correct their behaviors. In **Table 3.1**, we list the strengths and weaknesses of the three main types of distracted driving detection models that we considered: (1) sensor-based methods that explore a number of different driver and vehicle features, (2) video-based methods that use a neural network for detection, and (3) video-based methods that use alternative computer vision techniques. We will discuss each of these three methods in more details below.

**Requirements for sensor-based models:** For models that rely on specialized devices to collect data, the specialized devices that these models use must also be able to run on the vehicle. Moreover, the process with which these specialized devices use to collect data should not be felt as intrusive from the point of view of the drivers. If either of these requirements is not met, it would be difficult for such systems to be widely adopted in everyday use. For example, use of electroencephalography (EEG) to monitor driver brain activity and heart rate is not only excessively intrusive to drivers but also unlikely to be conducted in a moving vehicle. Another study that used an embedded camera on eyeglasses to capture driver eye

Table 3.1: Pros and Cons of Various Distracted Driving Models Considered

| Method Type | Benefits | Drawbacks |
|---|---|---|
| Sensor-based Methods | • Include a rich set of driver features (head pose, body movement, heart rate, eye blink)<br>• Include a rich set of vehicle features (IMU data, GPS location points) | • Required sensors might not be able to install on a vehicle<br>• Data collection process might be intrusive for drivers |
| Video-based Neural Network Methods | • Expressive model that allows for individual finetuning<br>• Only require a smart camera to collect data | • Computationally expensive with large memory footprint |
| Video-based Non Neural Network Methods | • Computationally efficient with light memory footprint<br>• Only require a smart camera to collect data | • Unable to support model individualization |

movement and gaze direction is also unlikely to be used in the real world [8].

Other practical considerations to consider include cost and reliability. A good example of this was described in a 2013 NHTSA report: measuring eye pupil response via eye-tracking devices. Although a driver's pupil response was shown to be correlated with his cognitive attention on the road, it is cost ineffective and therefore unlikely for a production vehicle to include any eye-tracking devices that has the required precision to capture this information. By the same token, other sensor-based systems that require state-of-the-art sensor technology to run their detection algorithm are unlikely to see mass adaptation until the required sensors can be purchased at an affordable price and in large quantities.

**Requirements for video-based models:** For video-based models with smart cameras as its only data source, remote viability of the sensors and intrusiveness of the data collection methods would be less of a concern. This is because recording drivers' driving behaviors via an in-cabin smart camera can be done reliably and is regarded as less intrusive compared

to other data collection methods that require physical contact. However, these video-based methods usually involve a neural network that demands high compute and memory resources that are at odds with what an edge device can offer. As such, the compute and memory requirements of these systems need to be counterbalanced with the hardware limitations without compromising the systems' ability to make inferences in real-time. Although the current state-of-the-art smart cameras are equipped with ample memory and powerful compute resources (most have dual- or quad-core processors, and some may even include AI hardware accelerators to facilitate CNN/DNN tasks), it is still unrealistic to run and train a full-size neural network in these cameras without significant performance degradation. Not to mention most video-based analytic models for distracted driving use budget cameras whose computer power and storage space are far lower than the state-of-the-art smart cameras.

Other video-based models such as [26] and [30] that use computer vision heuristic techniques to detect driver features are less likely to face the compute and memory constraints than the neural network methods. However, these models are more susceptible to the idiosyncratic nature of distracted driving. [40] and [41] both found that bad driving styles and distracted driving behaviors can exhibit in various forms, such as incorrect lane turning, tailgating, and harsh stop, much akin to Leo Tolstoy's famous saying "All happy families are alike, but every unhappy family is unhappy in its own way."

In addition to the individualistic nature of drivers' own driving behaviors, the environment in which the drivers operate their vehicle also tend to be different. For example, the driver-view video footage of a truck driver driving at night on a highway would be substantially different from those capturing a person driving in a local grocery in the morning. The differences in light variation, in-cabin settings, and driver outfits pose a unique challenge on these non-neural-network models because they are inflexible and hard to fine tune at the individual level.

**Requirements for all models:** Finally, just as many studies have considered the intrusiveness of data collection process as something that would negatively affect the acceptance of a distracted driving system, how frequently the system gives out false alarms would also play an important role in convincing drivers to adapt to the system. For example, a system that constantly sends out false reminders to a focused driver would be perceived as ineffective and wrose, distractive. In this case, the primary source of distraction that diverts the driver's attention is the very alerting sound from the system whose original intention is to prevent the driver from being distracted. Worse off, after the system gives out too

many false alarms, drivers would stop paying attention to the system or turn the system off in thinking that it is not functional. The perceived effectiveness of a system in this case plays an important rule because it determines the likelihood that the driver would follow the instructions of the system. If the drivers decide to ignore the alerting of a system that might occasionally emit faulty alarms but otherwise functionally working, then the actual effectiveness of that system would be far lower than what it suggested on paper. Therefore, we think that in order for such a system to have practical use cases, it is more important to have fewer false positives than to have fewer false negatives.

## 3.2 PROBLEM ASSUMPTIONS AND DESCRIPTION

In this thesis, we are interested in the video-based neural network approach, because we think it is most likely to be adapted for real-world applications among the three approaches considered above. Specifically, we want to propose an alternative neural network architecture for distracted driving detection. Our neural network model is intended to be run on a smart camera as part of the video analytics pipeline for real-time distracted driving detection.

**Assumption on data input** The primary source of input for our model is a smart camera mounted in the vehicle that is positioned to point inward towards the driver, outward to the road, or both. For distracted driving detection, we assume the smart camera is always pointing at the driver, and therefore the analytics pipeline would receive the video streaming data from the driver-facing camera. For the reliability and cost issues discussed above, in this thesis we do not consider specialized sensors as a possible source of input. However, we want our model architecture to be flexible enough such that it is feasible to include sensor data or telematics data into our video analytics pipeline should we want to in the future.

**Assumption on compute and storage constraints** As smart camera technology continues to improve, most smart cameras have similar compute and memory resources with smartphones. The current state-of-the-art smart cameras have even more CPU power and memory bandwidth comparable to those of a personal laptop. Some smart cameras even have built-in GPU accelerators for computer vision tasks. Therefore, in this thesis, we decide not to impose a stringent constraint on the smart camera's compute and storage capacity. As long as the detection model can be run in real-time on recent models of a smartphone, we will consider it as acceptable for real-world application use. In fact, we have seen an increasing number of edge machine learning frameworks that are built to run a resource-strapped device being developed over the past few years [37], [38]. Moreover, more and more of these

edge-device frameworks (such as DeepIoT [42] and DeepThings [43]) were developed and optimized for a specific set of use cases with higher performance but fewer compute and memory requirements.

**What is considered in this thesis:** Although the video-based neural network method is the most favorable among the three approaches we considered, we think it still has substantial room for improvement. Most notably, existing neural network solutions fail to consider the hard-crafted driver features (head pose, eye blink, hand gesture) and vehicle features (speed, location, IMU data) explored in other papers. We believe the existing network is not sophisticated enough to learn the various driver features (e.g., hand position) and their correlation with drivers being distracted (e.g., hands not on the driving wheel). We think the model needs assistance with learning a comprehensive representation of these various driver features to make a more informed prediction on distracted driving .

Therefore, for the remainder of this thesis, we are interested in finding answers for the following question: How can we incorporate different driver features into the existing neural network framework to robustly detect driver distraction? Such a framework should be generic enough that one can easily include new features or remove irrelevant ones. At the same time, this framework needs to satisfy all the problem requirements specified in **Section 3.1** above. Namely, they are:

- the model can be run on a smart camera device in real-time;
- the model only requires video streaming data (collected from the smart camera) as its model input;
- the model allows for model individualization; and
- the model should be trained to minimize its false positives.

# CHAPTER 4: MODEL ARCHITECTURE

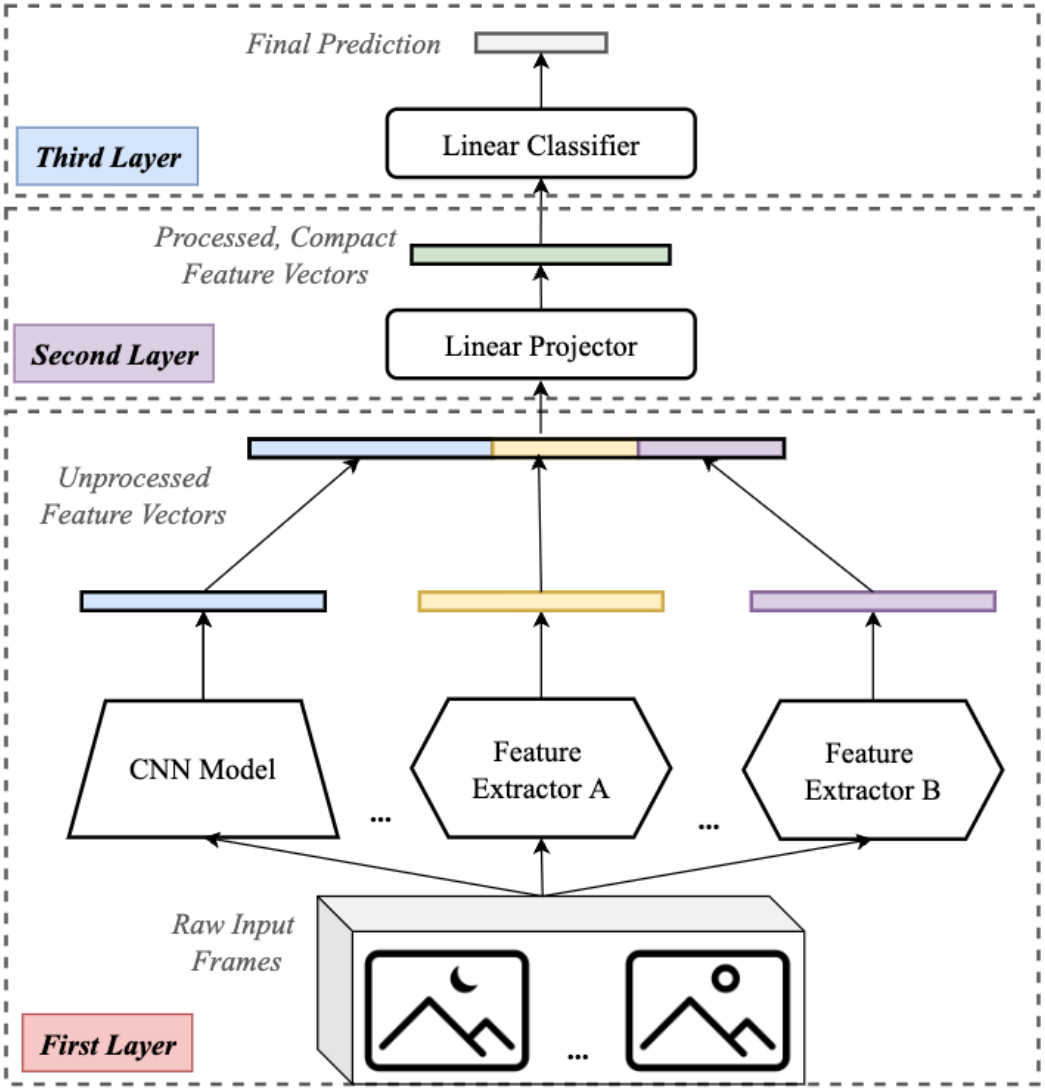
## 4.1 OVERVIEW

**Figure 4.1** depicts the model architecture for our own distracted driving detector. At a high level, our model architecture consists of a primary CNN neural network and a number of custom feature extractors that are aimed to capture additional driver and/or vehicle features not included in the main CNN model. Our model architecture is meant to be generic in that the primary CNN network can be replaced by any other alternatives, and that the current set of feature extractors can be easily expanded or removed. In this thesis, we chose ResNet-50 [38] as the primary CNN neural network. We also include three additional feature extractors. They are (1) HopeNet, a head pose estimation network [44], (2) AxGaze, an eye and gaze estimation network [45]), and finally (3) a hand position detector using YoloV4 Tiny [46]. Both HopeNet and AxGaze are the open-source implementation of [47], which first proposed a deep learning approach to extract head pose of a person without relying on facial keypoint estimation. As shown in the diagram, our model architecture consists of three layers:

**First Layer:** Residing in the first layer are the chosen CNN neural network and the feature extractors. They are responsible for converting the raw images into vector representation of the driver state. Although all feature extractors we included in this thesis are trainable, our model can also accept untrainable features as part of this layer. For example, we can also include vehicle speed and location data as part of a vehicle feature. We take the output of the last layer of the CNN model (a flattened vector of size 2048 by 1), and concatenate it with the output vectors from the feature extractors. The concatenated vector is then passed into the second layer in our model architecture.

**Second Layer:** We include a linear projector in this layer trained with a contrastive model training mechanism. The purpose of this linear projector is to learn a compact representation of driver behaviors from the CNN model output and the driver features. Because the feature extractors in the first layer may not be trainable, this linear projector allows us to train on these extractors indirectly in the same way as we could for our CNN models. The input of the projection layer varies depending on model selection in the first layer, but the output is a fixed size vector of length 512. A contrastive training algorithm is applied to ensure that we maximize the similarity of these fixed size vectors that have the same class label, and minimize the similarity for those with different labels.

Figure 4.1: Overview of Model Architecture




**Third Layer:** The structure of the final layer depends on the downstream tasks of this model. Because we use an image classification dataset with 10 class labels to evaluate our model, the last layer is a linear classifier that takes in the fixed-size vectors and outputs a prediction vector of size 10. **Table 4.1** includes a detailed description of each layer, as well as the input and output dimension of each of these models.

**Flexible model selection (§4.2)** Our architecture is meant to be generic and flexible in that the primary CNN network can be replaced by any other alternatives, and that the current set of feature extractors can be easily expanded or removed. This flexibility ensures that we can adaptively scale up or scale down the model capacity depending on the hardware limitation. Furthermore, as newer and more powerful on-device CNN frameworks

Table 4.1: Selected CNN Model and Feature Extractors

| Type | Selected Model | Trainable | Model Input (Input Dimension Bold Below) | Model Output (Output Dimension Bold Below) |
|---|---|---|---|---|
| **I. First Layer:** | | | | |
| Primary CNN Model | ResNet-50 | Yes | • RGB pixel values of the raw driver color images are in shape (256, 256, 3)<br>• Center crop to ***(224, 224, 3)*** as the expected input shape of the ResNet-50 model | • The last linear layer of the ResNet-50 model of shape ***(2048, 1)*** |
| Feature Extractor | HopeNet Lite (Head Pose Estimator) | No | • RGB pixel values of the raw driver color images are in shape (256, 256, 3)<br>• Top left crop to ***(192, 192, 3)*** as the expected input shape of the HopeNet model | • Euler angles for head pose estimation of shape ***(66, 3)*** in yaw, pitch, and roll axis<br>• Each axis vector is of shape (66,1), where 66 is a defined constant |
| | AxGazeNet (Gaze Estimator) | No | • Same as above | • Gaze estimation of shape ***(16, 2)***<br>• For each eye the model predicts a vector of shape (16, 1) in phi and theta angles, each of which is of length 8 |
| | Tiny YoloV4 (Hand Detector) | No | • RGB pixel values of the raw driver color images are in shape ***(256, 256, 3)*** and are directly passed into the Tiny YoloV4 model | • Hand bounding box location (4, 1), overlap with driver head (4, 1), and overlap with driving wheel (4, 1)<br>• Final output for both hands is therefore of shape ***(12, 2)*** |
| **II. Second Layer:** | | | | |
| Linear Projector | N/A | Yes | • Combine the last linear layer of the CNN model (2048, 1) with the head pose feature (66, 3), gaze feature (16, 2), and hand feature (12, 2)<br>• The concatenated, flattened vector is therefore ***(2302, 1)*** | • The projector output dimension is configurable<br>• In this thesis, we use a 512-length vector: ***(512, 1)*** |
| **III. Third Layer:** | | | | |
| Linear Classifier | N/A | Yes | • Output of the projection layer of shape ***(512, 1)*** | • Output of the linear classifier is of shape ***(10, 1)***, where 10 is the number of classes |

are developed in the future, they can be easily integrated into our proposed architecture with minimal changes.

**Contrastive model training (§4.3)** Because we concatenate the final layer of the CNN model with various different feature vectors before passing them into a linear classifier, we want the concatenated vector to be a fair representation for the driving state that a driver is

in. Inspired by the contrastive learning algorithms proposed in [33] and [48], we separately train a projection head to obtain a compact vector for the primary CNN model and the feature selectors. We can compare these compact vectors to ensure that similarity of any two compact vectors within the same label class is maximized, and that of any two compact vectors from different label classes is minimized.

**Consistent model prediction (§4.4)** Finally, to minimize the likelihood that the model gives out false positives (incorrectly alarting a focused driver) during real-time detection, we train a simple threshold decider to determine when to alert the driver given the model prediction. For now, this threshold decider only contains a single trainable parameter (only alerting the driver after the model has consistently detected a distracting behavior over the past few frames). Despite the simplicity of this decider, we have shown that in practice it significantly reduces the likelihood of false positives without substantially compromising on the model's accuracy.

## 4.2 DRIVER-RELATED FEATURE SELECTIONS

As we elaborate in **Chapter 2**, there is a wide variety of driver- and vehicle-related features containing critical information about drivers' distracted driving. Many of the driver-related features such as pupil contraction and head pose angle are usually collected via special-purpose sensors. However, with the rising popularity of video-based analytic pipelines, many have proposed alternative methods to extract these driver features by analyzing the driving-facing video footage, therefore bypassing the need of using a specialized sensor. Among these driver features now collectable via video analysis, we think driver head pose and gaze could be most useful for detecting distracted driving, because head pose and gaze features directly indicate whether the driver is visually distracted (i.e., not looking at the road). Other features that we consider in this section include driver's hand position, which is generally indicative of whether the driver is manually distracted (i.e., hands off the steering wheel).

**Head Pose and Gaze:** Driver head pose and gaze direction are strongly correlated with driver distraction. In most cases, driver head pose alone is sufficient to determine whether a driver is visually distracted. If the driver's head angle is deviated significantly from the normal position when the driver is looking straight, then it is likely that the driver is distracted.

To estimate driver head pose, we use the HopeNet model described above [44], which takes in the RGB pixel values of a driver-facing image with an image resolution of $192 \times 192$. Thus, the shape of the expected input of the HopeNet model is $(192, 192, 3)$. Because the image resolution of the training images is $256 \times 256$, we crop the top left corner of the image to the expected shape before passing the image to the HopeNet model. For model training efficiency, we top-left crop the images because the driver is situated at the top left part of the image. If the driver's position can not be determined in advance, such as the case when the camera mounting angle could vary, we could still re-scale the image size by some downsampling techniques. The HopeNet model outputs the driver's estimated head pose in Euler angles (yaw, pitch, and roll). Because each axis angle is a fixed-length vector of shape $(66, 1)$, where 66 is a predefined constant, the output of the HopeNet model is of shape $(66, 3)$.

To estimate driver gaze direction, we use the AxGaze model [45], whose model input is the same as the HopeNet model. The AxGaze model outputs the driver's estimated gaze direction for both eyes in polar angles (phi and theta). Because each axis angle is a fixed-length vector of shape $(8, 1)$, where 8 is a predefined constant, the output of the AxGaze model is of shape $(16, 2)$.

In **Figure 4.2**, we plotted the annotated outputs of some driving images using HopeNet and AxGazeNet. The first set of images are showing a focused driver looking straight at the front of the vehicle. The corresponding head pose and gaze vectors annotated in the figure are also pointing forward in the same direction. On the other hand, shown in the third set of images is a distracted driver talking to a passenger. In this case, the corresponding head pose and gaze vectors are pointing to the direction of the passenger seat. However, there are some rare cases in which the driver's head is tilted away yet his eyes are still looking straight on the road, such as the case shown in the second set of images: the driver tilted the angle of his head as he scratched his head with his opposite hand. However, this driver's eyes remain fixated on the road while performing this action.

Admittedly, the answer of whether the driver in **Figure 4.2(c)** is distracted in this case is contextual and situational. A human annotator would need to look at what this driver is doing around the time period in which this image is captured before he could conclude whether this driver is distracted. However, we find that our base model (Resnet-50 without any head pose and gaze context) would mis-categorize it as safe driving. If we include the headpose information to the base model, it is able to flag this as a distracting behavior

Figure 4.2: Driver-facing Images with Annotated Headpose and Gaze Vectors

a. Focused driver

b. Focused driver (annotated)

c. Partially distracted

d. Partially distracted (annotated)

e. Distracted driver

f. Distracted driver (annotated)

with high confidence. However, the headpose-aware model tends to confuse it with other distracting behaviors that have a similar head angle tilt such as driver reaching from behind

or talking to fellow passengers. After including the gaze feature into the base model, we find that the model is more likely to predict the correct distracting behavior.

**Relative Hand Position:** Another set of driver features that is explored often in existing research is drivers' hand position. Specifically, [4] and [5] find that certain types of driver hand gestures are highly correlated with drivers being distracted. This makes intuitive sense since if the driver's hands are not on the wheel, it is likely that the driver is engaging in a non-driving-related activity that requires the use of the hand (manual distraction).

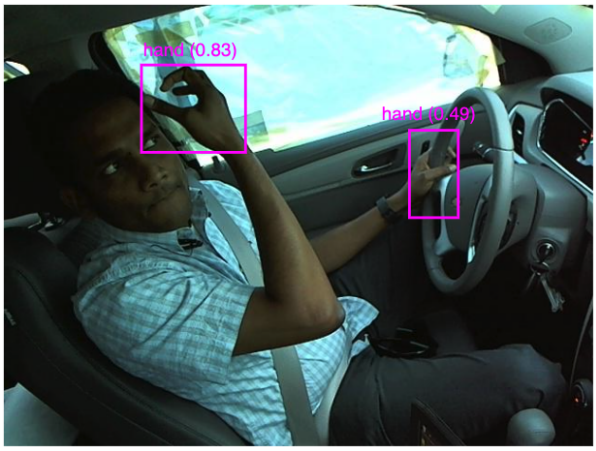
To validate this intuition, we adopt a simplified version of the driver hand feature where we only look at the position of a driver's hands relative to the position of the driver's head and the steering wheel. To obtain the location of the driver's hands in a given image, we use the Tiny Yolo neural network, a popular deep learning framework for real-time object detection [49]. Because the dataset we use for evaluation does not contain bounding box locations for drivers' hands, we use an open-source implementation of YoloV4 that is already pre-trained for hand detection [46]. Similarly, the Yolo detector takes in the RGB pixel values of a driver-facing image of resolution $256 \times 256$. Therefore, the shape of Yolo detector's expected input is $(256, 256, 3)$, the same as the original shape of the training images.

To estimate driver gaze direction, we choose the top 2 candidates with the highest confidence score from the list of bounding box locations predicted from the Yolo detector. See **Figure 4.3** for some sample outputs. In addition, we also calculate the area of intersection between the bounding box for the driver's hand and the bounding box for the driver's head (obtained from HopeNet above as part of the output). The area of intersection between hand and steering wheel is also included. For simplicity, we hard coded the value of the bounding box location for the steering wheel because the steering wheel position remains largely unchanged across the training images. If this assumption is not valid, one could detect the steering wheel location using the same Tiny YoloV4 detector. Each bounding box is of shape $(4, 1)$, since we need four numbers to express the location and size of the bounding box. Because we have 3 bounding box locations (including two area of overlap bounding boxes) for each hand, the output of this Yolo hand detector is of shape $(12, 2)$.

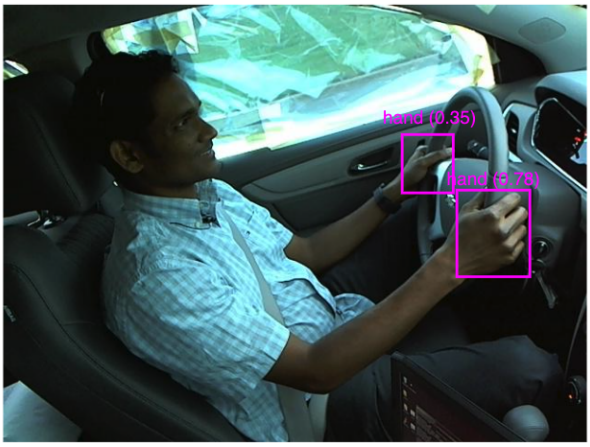
## 4.3 CONSISTENCY WITH CONTRASTIVE MODEL TRAINING

As explained in the sections above, we include a list of driver features (i.e., head pose, gaze, and hand) in our model. The feature extractors used to collect these features may not

Figure 4.3: Driver-facing Images with Annotated Headpose and Gaze Vectors




a. Distracted driver




b. Focused driver

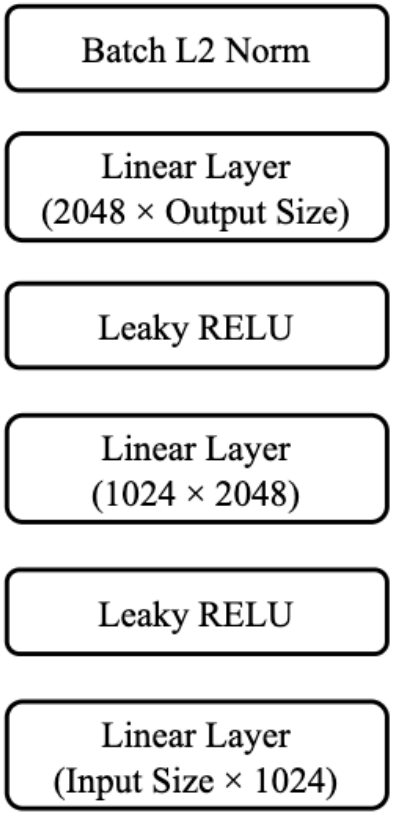
always be end-to-end trainable. Features that are obtained via non-neural-network methods are non-trainable by nature (such as vehicle speed data by an speedometer or driver heart rates by an Apple watch). Even features obtained from neural-network methods may not always be trainable. For example, we cannot train the Yolo hand detector because the dataset we use to train our model does not contain the bounding box locations for driver hands.

To address the issue that some or all driver features are not trainable, we train a projection layer that aims to learn a more comprehensive representation of the driver embedding from the incoming features. This approach is inspired by [33], which proposed a self-supervised learning method to differentiate safe driving behaviors from unsafe behaviors. The projection layer consists of three linear layers, separated by an activation layer (Leaky RELU). Detail architecture of the projection layer can be found in **Figure 4.4**.

In this thesis, we assume the neural-network-based feature extractors used in our model (i.e., HopeNet, AxGaze, and the Yolo hand detector) are not trainable. The goal of our contrastive training mechanism is to pretrain the projection layer such that it learns to differentiate the driver embedding of a safe driving image from that of a distracted driving image. Once the we pretrain the projection layer, we would then end-to-end train our entire model (final linear prediction layer, projection layer, and the primary CNN model) as a regular classification problem using a cross entropy loss on the 10 class labels.

To start, we first run the training images through our primary CNN model and the feature selectors and combine the outputs from these models to obtain a concatenated unprocessed

Figure 4.4: Projection Layer Architecture



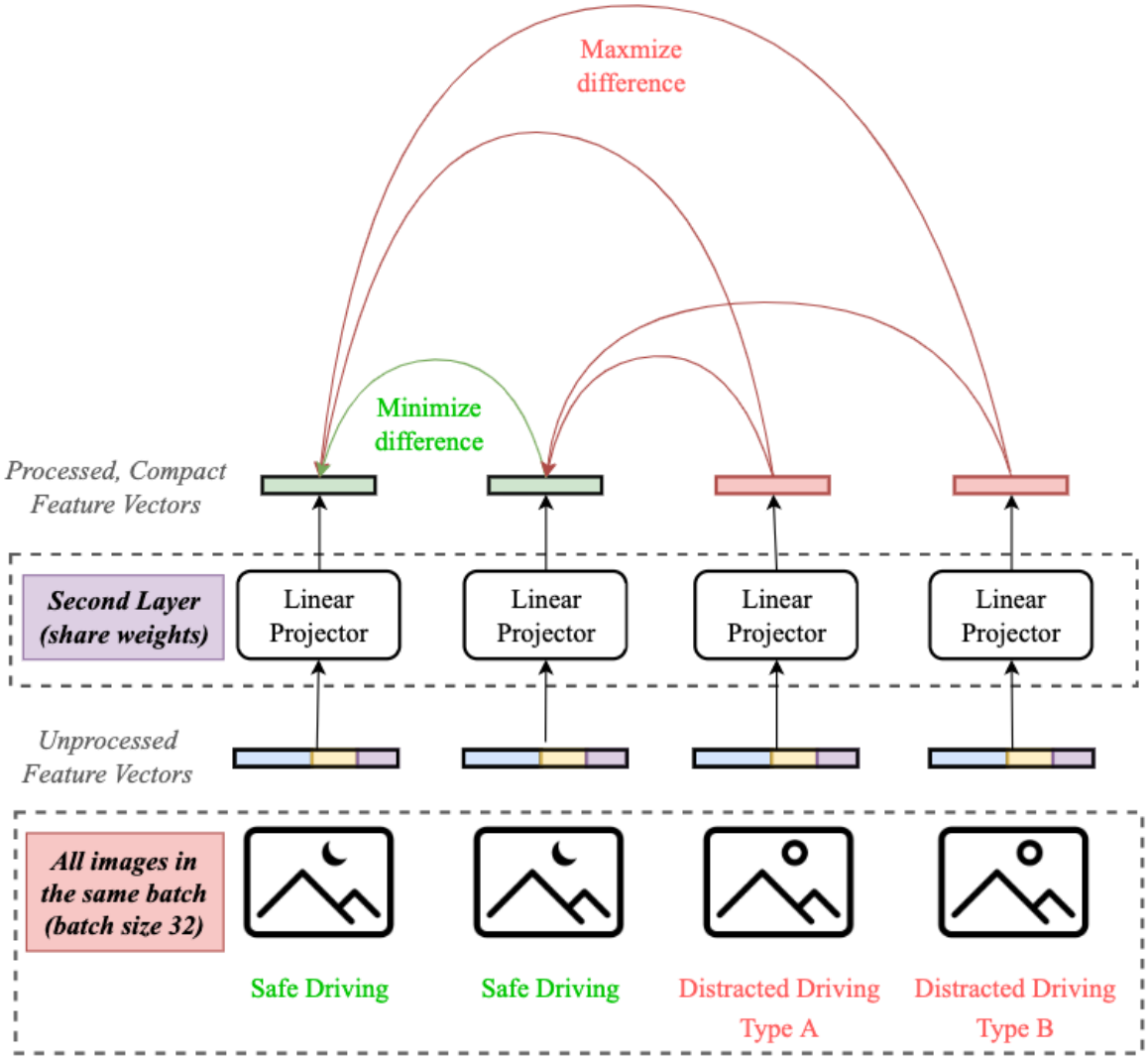

*Note:* Input size and output depend on the architecture of the detection model. In our case, the input size is 2302 and the output size is 502.

vector of shape $(2302, 1)$ for each training image. Next, these concatenated flattened vectors are then passed into the projection layer in batches where the batch size is 32. For each concatenated flattened vector, the output of the projection layer is a fixed-size driver embedding vector of shape $(512, 1)$. During contrastive training, we want to maximize the difference of the embedding vector of a safe driving image from the that of a distracted driving image within the same batch. At the same time, the difference between the embedding vectors of two safe driving images within the same batch should be minimized. An illustration of this pretraining process can be seen in **Figure 4.5**.

To allow for a faster pretraining, we consolidate the nine distracting behavior labels to two main categories: mobile phone usage (texting left/right, calling left/right) and other distracted behaviors (hair and makeup, talking to passengers, drinking, reaching behind, using radio). Therefore, within the same batch of size $N = 32 = K + L + M$, we have:

- $K$ images of safe driving: $(img^{safe})$

Figure 4.5: Overview of Contrastive Training Mechanism




- $L$ images of mobile usage distracting behaviors: $(img^{mobile})$
- $M$ images of other distracting behaviors: $(img^{others})$

Furthermore, let $\mathbf{F}$ be a trainable function of the projection layer's parameters $(w_{proj})$ that transforms an driver image to the fixed-size driver embedding vector. We therefore obtain $K$, $L$, and $M$ driver embedding vectors for safe driving $(v^{safe})$, mobile usage distracted driving behaviors $(v^{mobile})$, and other distracted driving behaviors $(v^{others})$, respectively.

- $K \times (K-1)$ positive pairs for safe driving: $({v_i}^{safe}, {v_j}^{safe})$
- $K \times L$ negative pairs for mobile usage distracting behaviors: $({v_i}^{safe}, v^{mobile})$
- $K \times M$ negative pairs for other distracting behaviors: $({v_i}^{safe}, v^{others})$

The goal of the contrastive training is then to optimize our function $\mathbf{F}(w_{proj})$ such that its produced embedding vectors for safe driving images are minimally different between each other yet maximally different from the function's output for distracted driving images. To achieve this goal, we optimize the following loss function:

$$\mathcal{L} = \sum_{i=1}^{K} \sum_{j \neq i}^{K} \frac{\mathcal{L}_{i,j}}{K \cdot (K-1)}, \text{ where} \tag{4.1}$$

$$\mathcal{L}_{i,j} = -\log \frac{\exp\left(v_i^{safe} \cdot v_j^{safe}\right)}{\exp\left(v_i^{safe} \cdot v_j^{safe}\right) + \delta_{mobile}(v_i^{safe}) + \delta_{others}(v_i^{safe})} \tag{4.2}$$

$$\delta_{mobile}(v_i^{safe}) = \sum_{l=1}^{L} \exp\left(v_i^{safe} \cdot v_l^{mobile}\right) \tag{4.3}$$

$$\delta_{others}(v_i^{safe}) = \sum_{m=1}^{M} \exp\left(v_i^{safe} \cdot v_m^{others}\right) \tag{4.4}$$

By optimizing this loss function, we train the projection layer to learn the driver representations that maximize the similarity of any pair of safe driving vectors. At the same time, the similarity between a safe driving vector and any distracted driving vector is minimized.

## 4.4 CONSISTENCY WITH REAL-TIME MODEL PREDICTION

As discussed in **Section 3.1** above, false positives of a distracted driving detection model are viewed much more unfavorably than the model's false negatives by drivers in daily use. To minimize the occurrence of false positives, an intuitive solution would be to alter the loss function during training to force the model to favor precision over recall.

However, doing so has one significant drawback: the model is now less likely to predict distracted driving unless highly confident, because the loss function favors predicting safe driving when unsure. Therefore, the system is stuck in a difficult state where it is hard to improve recall without hurting precision. This conundrum happens because we change the prediction logic of the system to satisfy the requirement in the alerting part of the system. We

think a better solution here is to train the model as is with regular loss function that favors recall and precision equally but separately train an alerting decider to determine whether to remind the driver when a distracting behavior is detected by the model. Doing so not only allows us to have a clear separation of concern between the detection and prediction logic, we are also able to obtain an objective view of the model detection performance that is comparable with other existing detection methods.

The current alerting decider logic is simple. After training, we would replay the train images in the correct order as they appear in the video. We evaluate the model's detection conviction level by looking at the ratio $Y/X$ of how many $Y$ frames of the last $X$ frames are classified as distracted driving by the model, where $X$ is set to 10. The alerting decider then asks the question of what is the minimum $Y$ value such that the false positive rate is below $Z$ percent. For each possible value of $Y \in \mathbb{Z}[0, 10]$, we calculate the false positive rate and overall alerting accuracy and find the best value for $Y$.

# CHAPTER 5: EVALUATION

## 5.1 ENVIRONMENT

Model training is done on an AWS EC2 instance (p3.2xlarge) with 8 cores and 1 Tesla V100 GPU [50]. Most model evaluations are done on an Apple Macbook Air with dual-core Intel Core i5 (1.6GHz) without any GPU accelerators. Model evaluations that require GPU accelerations are conducted on Google Colab, a cloud notebook environment with free access to a GPU accelerator. We use ResNet-50 as our primary CNN neural network, whose implementation is obtained from the Pytorch library. The Pytorch library offers an ImageNet pre-trained version of the ResNet-50 model, which is what we use in this evaluation section. Driver head pose and gaze features are obtained from open-source Github packages of HopeNet [44] and AxGazeNet [45]. Driver hand bounding box locations are obtained from the Yolo hand detector available at [46]. Our model implementation is done in PyTorch.

## 5.2 DATASET

We use the StateFarm dataset to evaluate our model's performance. The StateFarm dataset is an image-based distracted driving classification dataset. These images are taken from an interior camera pointing at the driver in a moving vehicle. Ideally, we would want to have a dataset that contains video clips such that we can simulate the real-life scenario where our model would receive a stream of frames from the camera, but the availability of such datasets is rare. For each subject, we can stitch these driver images back into a video clip to mimic the behavior of live streaming video. Although no information regarding the original video's FPS can be obtained, we experiment with different values and find that at 30 frames per second the synthetic video has the best quality.

The full StateFarm dataset has about a hundred thousand images collected from 26 different subjects. For a faster model training and evaluation timeline, we use a smaller subset of the StateFarm dataset (about 22,000 images from 11 subjects) to create our own train, validation, and test sets. Each image is labeled either as safe driving or as one of the nine distracted driving behaviors.

Table 5.1: Classification class labels and the train/test split for Scenarios 1A-1C

| Class Label | Total Number of Available Images | Number of Images in Scenario 1A | | Number of Images in Scenario 1B | | Number of Images in Scenario 1C | |
|---|---|---|---|---|---|---|---|
| | | Train/Val | Test | Train/Val | Test | Train/Val | Test |
| Safe Driving | 2,489 | 1,991 | 498 | 1,245 | 1,245 | 249 | 2,240 |
| Texting (Left) | 2,267 | 1,814 | 453 | 1,134 | 1,134 | 227 | 2,040 |
| Talking on the Phone (Left) | 2,317 | 1,854 | 463 | 1,159 | 1,159 | 232 | 2,085 |
| Texting (Right) | 2,346 | 1,877 | 469 | 1,173 | 1,173 | 235 | 2,111 |
| Talking on the Phone (Right) | 2,326 | 1,861 | 465 | 1,163 | 1,163 | 233 | 2,093 |
| Reaching Behind | 2,002 | 1,602 | 400 | 1,001 | 1,001 | 200 | 1,802 |
| Drinking | 2,325 | 1,860 | 465 | 1,163 | 1,163 | 233 | 2,093 |
| Talking to the Passenger | 2,129 | 1,703 | 426 | 1,065 | 1,065 | 213 | 1,916 |
| Using Radio | 2,316 | 1,853 | 463 | 1,158 | 1,158 | 232 | 2,084 |
| Hair and Makeup | 1,907 | 1,526 | 381 | 954 | 954 | 191 | 1,716 |
| **Total:** | **22,424** | **17,939** | **4,485** | **11,212** | **11,212** | **2,242** | **20,182** |

*Note:* Class labels and definitions are obtained from [31].

## 5.3 SCENARIOS

In this thesis, we consider three different testing scenarios, with a difficult level from easy to hard. For these three scenarios, our model is trying to predict the correct label class (from the 10 provided labels) given a driver image. More details of the class labels and how they are distributed in each scenario can be found in **Table 5.1**.

- *Scenario 1A*: 25 percent of the images are reserved as the test set, and all 11 subjects occur in both train and test set.
- *Scenario 1B*: half of the images are reserved as the test set, and only six subjects occur in both the train and test set.
- *Scenario 1C*: we train the model with only one subject's images, which account for less than 10 percent of total available images. The images of the remaining 10 subjects occur in the test set only.

These three scenarios aim to test how well our model performs on unseen subjects (i.e.., drivers) and whether our model learns a fair representation of each driving behavior that is robust across different subjects. In addition to subject idiosyncrasy, however, we also want to evaluate our model's performance for unseen driving behaviors. In the real-world settings, a set of possible driving behaviors that our model would see if potentially unbounded, and we want to make sure that our model is also robust against unseen class labels.

Table 5.2: Binary Classification and the train/test split for Scenarios 2A-2C

| Class Label | Total Number of Available Images | Number of Images in Scenario 2A | | Number of Images in Scenario 2B | | Number of Images in Scenario 2C | |
|---|---|---|---|---|---|---|---|
| | | Train/Val | Test | Train/Val | Test | Train/Val | Test |
| Safe Driving | 2,489 | 1,989 | 500 | 1,489 | 1,000 | 1,489 | 1,000 |
| Texting (Left) | 2,267 | 1,767 | 500 | 1,267 | 1,000 | 0 | 1,000 |
| Talking on the Phone (Left) | 2,317 | 1,817 | 500 | 0 | 1,000 | 0 | 1,000 |
| Texting (Right) | 2,346 | 1,846 | 500 | 0 | 1,000 | 0 | 1,000 |
| Talking on the Phone (Right) | 2,326 | 1,826 | 500 | 1,326 | 1,000 | 1,326 | 1,000 |
| Reaching Behind | 2,002 | 1,502 | 500 | 1,002 | 1,000 | 0 | 1,000 |
| Drinking | 2,325 | 1,825 | 500 | 1,325 | 1,000 | 0 | 1,000 |
| Talking to the Passenger | 2,129 | 1,629 | 500 | 1,129 | 1,000 | 1,129 | 1,000 |
| Using Radio | 2,316 | 0 | 1,000 | 0 | 1,000 | 0 | 1,000 |
| Hair and Makeup | 1,907 | 0 | 1,000 | 0 | 1,000 | 0 | 1,000 |
| **Total:** | **22,424** | **14,201** | **6,000** | **7,538** | **10,000** | **3,944** | **10,000** |

*Note:* Class labels and definitions are obtained from [31].

Therefore, we propose three additional scenarios. For these three new scenarios, our model is trying to make a binary prediction of whether the current frame contains distracted driving behaviors. More details of the class labels and how they are distributed in each scenario can be found in **Table 5.2**.

- *Scenario 2A*: Two type of distracted driving classes are reserved for the test set: Using radio and Touching hair
- *Scenario 2B*: Four type of distracted driving classes are reserved for the test set: Same as above, but additionally include Texting (Left) and Talking on the Phone (Right)
- *Scenario 2C*: Seven type of distracted driving classes are reserved for the test set: Same as above, but additionally include Texting (Right), Reaching from Behind and Drinking

For both scenarios above, the performance metrics we are interested in measuring are the model prediction accuracy as well as inference speed.

## 5.4 OVERALL PERFORMANCE

**Baseline model:** To evaluate the performance of our presented model, we compare it with a baseline CNN neural network without any driver feature extractions. For a fair comparison,

Table 5.3: Model Performance for For Test Sets with Unseen Drivers

| Model | Accuracy on *Scenario 1* Test Sets | | | Inference Speed |
|---|---|---|---|---|
| | Easy (1A) | Moderate (1B) | Hard (1C) | |
| *Baseline:* ResNet-50 | 86.5% | 85.9% | 83.8% | 110-115 fps |
| + *Add* Head Pose and Eye Features | 87.2% | 85.2% | 84.0% | 85-90 fps |
| + *Add* Hand Position Features | **88.3%** | 87.0% | **84.4%** | 55-60 fps |
| + *Add* Contrastive Training Schedules | 86.5% | **87.4%** | 84.2% | 55-60 fps |

*Note:* A GPU accelerator is used to run the Yolo hand detector [46] to extract the bounding box locations for driver's hands.

we decide to use ResNet-50 as our baseline model. The baseline model is pre-trained on ImageNet and we train the baseline model end-to-end on the train images. Standard data augmentation is conducted during training, including a random flip of 0.1 probability, a random rotation with 0.5 probability, and a mean normalization using ImageNet statistics. It is worth noting that for both the baseline model and our proposed model, the training length is limited to 10 epochs for faster model iteration and evaluation cycle. Additionally, minimal hyperparameter optimization is performed for the baseline model and our proposed model.

**Model performance with unseen drivers in test:** For *Scenarios 1A-1C*, the model tries to classify a given driver frame into one of the 10 class labels (one for safe driving and the other nine for distracted driving). Therefore, the output of the final linear prediction layer for both our proposed model and the baseline model is a 1-d vector of size 10. The accuracy of our model compared with the baseline can be found in **Table 5.3**. We find that adding feature selections to our primary CNN neural model is able to improve our model test performance, although the absolute increase in each of the three cases is not significant. Interestingly, pretraining the projection layer of our model with contrastive training mechanism does not always help on final prediction accuracy. For the easy and hard cases (*Scenario* 1A and 1C), including the contrastive training mechanism actually lowers our

model's accuracy slightly. These results initially come as a surprise to us, but in retrospect they make intuitive sense. Contrastive training mechanism is best suited for semi-supervised learning, and the presentation the model learned from Contrastive training mechanism may deviate significantly from the representation needed for multi-class classification, as noted in [51].

Another interesting observation we discover is that we do not see a significant performance degradation going from the easy case to the hard case even for the baseline model. The test to train ratio increases dramatically from 0.25:1 (0.25 test images per train image) in the easy case to 9:1 (9 test images per train image) in the hard case. Despite a sheer decrease in the training set volume, we only witnessed a less than 3 percent decrease on our model's accuracy and that of the baseline model. This observation seems to suggest the train and test images exhibit a low degree of variation within each label class such that even a vanilla neural network model can differentiate these different class labels with little training required.

Finally, we also evaluate our model against the *baseline* model on the model's inference speed, as shown in the last column in **Table 5.3**. All our model components are able to run inference at or above 80 frames per second except for the driver hand feature extractor, which uses a pre-trained Tiny YoloV4 network. In order to bring our model's inference speed close to the real-time inference threshold of 60 frames per second, we have to run the driver hand feature extractor via a GPU accelerator. Since the only information we need from the driver hand feature extractor is the bounding box locations of the driver's two hands, we can rely on other computationally efficient methods to obtain the bounding box locations. In fact, the HopeNet [44] contains a head detection algorithm without keypoint estimations that could be applied to detect driver hands.

**Model performance with unseen behaviors in test** Next, we look at the model's performance for *Scenarios 2A-2C*, which contains unseen distracted driving behaviors in the test set. Under this scenario, the model tries to label a given driver frame into safety driving or distracted driving. Therefore, the size of the final linear prediction layer is adjusted to 1 for a binary prediction. Because we combine all distracted driving labels into the a generic label, the number of training images for distracted driving outnumbered that of safe driving images. To ensure class balance, we upsample the safe driving class and downsample the distracted driving class accordingly to ensure the number of training images in each class is consistent across all three cases in *Scenario 2*.

Table 5.4: Model Performance for For Test Sets with Unseen Distracted Driving Behaviors

| Model | Accuracy on *Scenario 2* Test Sets | | | Inference Speed |
|---|---|---|---|---|
| | Easy (2A) | Moderate (2B) | Hard (2C) | |
| *Baseline:* ResNet-50 | 91.0% | 87.2% | 80.8% | 110-115 fps |
| + *Add* Head Pose and Eye Features | 92.4% | 90.2% | 85.1% | 85-90 fps |
| + *Add* Hand Position Features | 93.8% | 91.9% | 87.1% | 55-60 fps |
| + *Add* Contrastive Training Schedules | **94.5%** | **93.4%** | **90.6%** | 55-60 fps |

*Note:* A GPU accelerator is used to run the Yolo hand detector [46] to extract the bounding box locations for driver's hands.

In **Table 5.4**, we show the accuracy of our model compared with the baseline model. The baseline model is able to achieve an ever higher level of accuracy for the easy case at 91 percent. In fact, most of the errors that the baseline model makes come from the two unseen distracted driving classes, which only have an accuracy of 82.5 percent, while the model accuracy for the seen classes is close to 95 percent. As we increase the difficulty level by removing more distracted driving classes from the train set, however, the performance of our baseline model starts to deteriorate. The baseline model performance drops below 90 percent to 87.2 percent if two more distracted driving class labels are removed from the train set (Scenario 2B), and drops even lower to 80.8 percent if a total of seven class labels are removed from the train set (Scenario 2C). It is worth pointing out that the accuracy of the baseline model on seen class labels has remained consistently high even in the moderate and hard case at around 93 percent.

On the other hand, our proposed model is more robust against unseen distracted driving behaviors in the test set. Our best model is able to achieve an accuracy over 90 percent for the easy, moderate, and hard cases, with less than 5 percent absolute decrease from the ease case to the hard case. As expected, including a contrastive training mechanism seems to help with the model performance and makes it more robust against unseen behavior classes.

## 5.5 COMPARISON WITH EXISTING METHODS

Unlike other computer vision action recognition problems, there is not a golden video-based dataset for distracted driving detection. The two widely used distracted driving datasets [31], [32] are both image-based and for distracted driving classification. Of the video-based neural network methods that we reviewed in **Chapter 2**, only [35] and [36] use the StateFarm dataset as part of their model evaluation. On its face, both models are able to achieve an accuracy of over 90 percent on their own train/test test split. However, these two models seem to be trained on all distracted driving labels and all driver subjects, as there is no mention in the paper that they are doing otherwise. Because of the low variation exhibited within each class label, it is not surprising that these models are also able to achieve high accuracy on this dataset.

Additionally, [36] used a larger neural network with longer training duration (100 epochs versus 10). This paper used VGG-16 as its neural network with over 138 million parameters, about six times the size of our primary CNN network ResNet-50, which has only 23 million parameters. Similarly, [35] also adopted a larger neural model, and the model requires longer time to process each image (over 100ms), which means that model at most can process incoming driver frames at 10 frames per second, far below the real-time requirement. Other neural network approaches such as [33] and [4] evaluated their models with custom datasets that cannot be compared directly.

# CHAPTER 6: CONCLUSIONS

## 6.1 SUMMARY

In this thesis, we consider the problem of real-time distracted driving detection and conduct an in-depth review of the existing solutions. In particular, we look at the sensor-based approaches that use specialized sensors to collect driver-related features such as driver head pose and eye movement and attempt to predict distracted driving using these collected features. The other popular approach evaluated in this thesis is an on-camera video analytics pipeline that runs a neural network model to detect distracting behaviors in a series of driver images.

Inspired by the various driver features used in the sensor-based approach, we propose a novel neural network architecture for real-time distracted driving detection that allows us to incorporate different hand-crafted driver features as part of the model training process. We also introduce a novel contrastive training mechanism to learn a compact driver embedding from the primary CNN model output and the selected driver features. The learned driver embedding could then be used in downstream tasks such as distracted driving classification. In the experiment section, we show that with a similar amount of training time, our model is more effective at learning how to detect distracted driving compared to the existing vanilla neural network methods. Specifically, our model is shown to be more robust against unseen distracted driving behaviors during test time.

Moreover, our proposed model is designed to be flexible in accepting any number of driver and vehicle features, while retaining the ability to train the whole pipeline end-to-end with a CNN neural network. This thesis contributes to the current knowledge on distracted driving detection by identifying the gaps in the existing video-based analytics pipelines, and also by proposing a new architecture that incorporates the hand-crafted driver and vehicle features that are ignored in the existing pipelines. The initial results presented in this thesis show the potential of unifying the hand-crafted features from the sensor-based approach with the training and decision process of the existing video-analytics pipelines. We believe the preliminary success of our work has shed light on a new area of research for future work on distracted driving detection.

## 6.2 LESSONS LEARNED:

The biggest challenge that we have to overcome in this thesis is that the existing driver distraction detection datasets are not challenging enough for neural network methods. As shown in Chapter 5, our baseline model that use a ResNet-50 without extensive hyperparameter search is able to achieve over 85 percent accuracy on the dataset. Even though the StateFarm dataset consists of 26 subjects of over 100,000 images, there is a low variation in light sources, vehicle cabin setup, and driver behaviors.

Furthermore, the two most widely used datasets, StateFarm dataset [31] and AUC dataset [32] are both image-based and not video-based. Therefore, these two datasets can not be used directly to evaluate existing video-based analytic pipelines. Although the images in the StateFarm dataset can be stitched back heuristically to be played back as short video clips, these synthesized video clips are not suitable for evaluating video analytics pipelines. This is because as a classification dataset, the StateFarm dataset maintains the class balance across its 10 class labels, only one of which is for safe driving. This means that in the synthesized video clips, about 90 percent of time the driver is engaging in some kind of a distracted driving behavior.

Another challenge we face is to determine the set of driver and vehicle features to include in our model. Given the limitation of our dataset, we are unable to collect any vehicle related features. Because we can only extract driver features from the driver images without any external sensors, we found that most reliable methods involve using a neural network to detect object of interests, and we need to manage our resources carefully with running multiple neural networks on the same device. In the future, we would like to consider other compute-efficient feature extraction methods such as the one using computer vision techniques to detect facial keypoints.

## REFERENCES


[1] "NHTSA summary of statistical findings - distracted driving 2013," 2013. [Online]. Available: https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812132

[2] "NHTSA summary of statistical findings - distracted driving 2018," 2018. [Online]. Available: https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812926

[3] "AAA: Distracted driving increases with adas technology," 2019. [Online]. Available: https://aashtojournal.org/2019/12/20/aaa-distracted-driving-increases-with-adas-technology/

[4] T. H. N. Le, K. G. Quach, C. Zhu, C. N. Duong, K. Luu, and M. Savvides, "Robust hand detection and classification in vehicles and in the wild," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 1203–1210.

[5] T. Zhou, P. J. Pillai, and V. G. Yalla, "Hierarchical context-aware hand detection algorithm for naturalistic driving," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016, pp. 1291–1297.

[6] K. Díaz-Chito, A. Hernàndez, and A. López, "A reduced feature set for driver head pose estimation," *Applied Soft Computing*, vol. 45, pp. 98–107, 04 2016.

[7] H. Koma, T. Harada, A. Yoshizawa, and H. Iwasaki, "Considering eye movement type when applying random forest to detect cognitive distraction," in *2016 IEEE 15th International Conference on Cognitive Informatics and Cognitive Computing (ICCI*CC)*, 2016, pp. 377–382.

[8] A. Palazzi, D. Abati, S. Calderara, F. Solera, and R. Cucchiara, "Predicting the driver's focus of attention: the dr(eye)ve project," *CoRR*, vol. abs/1705.03854, 2017. [Online]. Available: http://arxiv.org/abs/1705.03854

[9] C. Tran, A. Doshi, and M. M. Trivedi, "Modeling and prediction of driver behavior by foot gesture analysis," *Computer Vision and Image Understanding*, vol. 116, no. 3, pp. 435–445, 2012, special issue on Semantic Understanding of Human Behaviors in Image Sequences. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1077314211002086

[10] J. Groeger, *Understanding Driving: Applying Cognitive Psychology to a Complex Everyday Task*, 01 2000.

[11] "FMCSA driver distraction in commercial vehicle operations," 2009. [Online]. Available: https://www.fmcsa.dot.gov/safety/research-and-analysis/driver-distraction-commercial-vehicle-operations

[12] "The CDC website for distracted driving," 2022. [Online]. Available: https://www.cdc.gov/transportationsafety/distracted_driving/index.html/

[13] V. Beanland, M. Fitzharris, K. L. Young, and M. G. Lenné, "Driver inattention and driver distraction in serious casualty crashes: Data from the australian national crash in-depth study," *Accident Analysis and Prevention*, vol. 54, pp. 99–107, 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S000145751300047X

[14] S. G. Klauer, T. A. Dingus, V. L. Neale, J. Sudweeks, and D. J. Ramsey, "The impact of driver inattention on near-crash/crash risk: An analysis using the 100-car naturalistic driving study data," 2006.

[15] M. A. Regan, C. Hallett, and C. P. Gordon, "Driver distraction and driver inattention: Definition, relationship and taxonomy," *Accident Analysis and Prevention*, vol. 43, no. 5, pp. 1771–1781, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0001457511000893

[16] M. Pettitt, G. E. Burnett, and A. Stevens, "Defining driver distraction," in *Defining Driver Distraction*, 2005.

[17] C. Wickens and J. McCarley, *Applied Attention Theory*. CRC Press, 2007. [Online]. Available: https://books.google.com/books?id=dIagIraXHPUC

[18] Y. Qi, R. Vennu, and R. Pokhrel, "Distracted driving: A literature review literature review on distracted driving in illinois illinois center for transportation," 04 2020.

[19] M. Née, B. Contrand, L. Orriols, C. Gil-Jardiné, C. Galéra, and E. Lagarde, "Road safety and distraction, results from a responsibility case-control study among a sample of road users interviewed at the emergency room," *Accident Analysis and Prevention*, vol. 122, pp. 19–24, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0001457518307103

[20] T. A. Dingus, F. Guo, S. Lee, J. F. Antin, M. Perez, M. Buchanan-King, and J. Hankey, "Driver crash risk factors and prevalence evaluation using naturalistic driving data," *Proceedings of the National Academy of Sciences*, vol. 113, no. 10, pp. 2636–2641, 2016.

[21] S. E. Donald, B. H., B. Madnick, and R. Walter, "Driver inattention and highway safety," *United States. National Highway Traffic Safety Administration*, 1985.

[22] H. Bishop, B. Madnick, R. Walter, and E. D. Sussman, "Potential for driver attention monitoring system development," *United States. National Highway Traffic Safety Administration*, 1985.

[23] O. Nakayama, T. Futami, T. Nakamura, and E. R. Boer, "Development of a steering entropy method for evaluating driver workload," *SAE Transactions*, vol. 108, pp. 1686–1695, 1999. [Online]. Available: http://www.jstor.org/stable/44668044

[24] S. Martin, K. Yuen, and M. M. Trivedi, "Vision for intelligent vehicles and applications (viva): Face detection and head pose challenge," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, 2016, pp. 1010–1014.

[25] T. Čegovnik, K. Stojmenova, G. Jakus, and J. Sodnik, "An analysis of the suitability of a low-cost eye tracker for assessing the cognitive load of drivers," *Applied Ergonomics*, vol. 68, pp. 1–11, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0003687017302326

[26] R. Manoharan and S. Chandrakala, "Android opencv based effective driver fatigue and distraction monitoring system," in *2015 International Conference on Computing and Communications Technologies (ICCCT)*, 2015, pp. 262–266.

[27] A. Tsuchida, M. S. Bhuiyan, and K. Oguri, "Estimation of drowsiness level based on eyelid closure and heart rate variability," *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 2543–2546, 2009.

[28] V. Häkkinen, K. Hirvonen, J. Hasan, M. Kataja, A. Värri, P. Loula, and H. Eskola, "The effect of small differences in electrode position on eog signals: application to vigilance studies," *Electroencephalography and Clinical Neurophysiology*, vol. 86, no. 4, pp. 294–300, 1993. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0013469493901118

[29] G. Sikander and S. Anwar, "Driver fatigue detection systems: A review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2339–2352, 2019.

[30] A. U. Nambi, S. Bannur, I. Mehta, H. Kalra, A. Virmani, V. N. Padmanabhan, R. Bhandari, and B. Raman, "Hams: Driver and driving monitoring using a smartphone," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '18. New York, NY, USA: Association for Computing Machinery, 2018. [Online]. Available: https://doi.org/10.1145/3241539.3267723 p. 840–842.

[31] "State farm distracted driver detection challenge," 2016. [Online]. Available: https://www.kaggle.com/c/state-farm-distracted-driver-detection

[32] Y. Abouelnaga, H. M. Eraqi, and M. N. Moustafa, "Real-time distracted driver posture classification," *32nd Conference on Neural Information Processing Systems (NIPS 2018)*, 2017. [Online]. Available: https://arxiv.org/abs/1706.09498

[33] O. Köpüklü, J. Zheng, H. Xu, and G. Rigoll, "Driver anomaly detection: A dataset and contrastive learning approach," 2020. [Online]. Available: https://arxiv.org/abs/2009.14660

[34] A. Berg, M. Oskarsson, and M. O'Connor, "Deep ordinal regression with label diversity," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, jan 2021. [Online]. Available: https://doi.org/10.11092Ficpr48806.2021.9412608

[35] B. Alotaibi and M. Alotaibi, "Distracted driver classification using deep learning," *Signal Image and Video Processing*, vol. 14, 04 2020.

[36] M. Z. Khalid A. AlShalfan, "Detecting driver distraction using deep-learning approach," *Computers, Materials and Continua*, vol. 68, no. 1, pp. 689–704, 2021. [Online]. Available: http://www.techscience.com/cmc/v68n1/41832

[37] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *CoRR*, vol. abs/1704.04861, 2017. [Online]. Available: http://arxiv.org/abs/1704.04861

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[39] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," 2018.

[40] A. Dasgupta, D. Rahman, and A. Routray, "A smartphone-based drowsiness detection and warning system for automotive drivers," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 11, pp. 4045–4054, 2019.

[41] M. M. Bejani and M. Ghatee, "A context aware system for driving style evaluation by an ensemble learning on smartphone sensors data," *Transportation Research Part C: Emerging Technologies*, vol. 89, pp. 303–320, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0968090X18301931

[42] S. Yao, Y. Zhao, A. Zhang, L. Su, and T. Abdelzaher, "Deepiot: Compressing deep neural network structures for sensing systems with a compressor-critic framework," 2017. [Online]. Available: https://arxiv.org/abs/1706.01215

[43] Z. Zhao, K. M. Barijough, and A. Gerstlauer, "Deepthings: Distributed adaptive deep learning inference on resource-constrained iot edge clusters," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 11, pp. 2348–2359, 2018.

[44] "Github: Hopenet - head pose estimation network," 2019. [Online]. Available: https://github.com/axinc-ai/ailia-models/tree/master/face_recognition/hopenet

[45] "Github: Axgaze - gaze estimation network," 2019. [Online]. Available: https://github.com/axinc-ai/ailia-models/tree/master/face_recognition/ax_gaze_estimation

[46] "Github: Yolo hand detection," 2021. [Online]. Available: https://github.com/cansik/yolo-hand-detection

[47] N. Ruiz, E. Chong, and J. M. Rehg, "Fine-grained head pose estimation without keypoints," *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017. [Online]. Available: https://arxiv.org/abs/1710.00925

[48] W. Im, S. Hong, S.-E. Yoon, and H. S. Yang, "Scale-varying triplet ranking with classification loss for facial age estimation," in *Computer Vision – ACCV 2018*, C. Jawahar, H. Li, G. Mori, and K. Schindler, Eds. Cham: Springer International Publishing, 2019, pp. 247–259.

[49] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 2016.

[50] "Amazon EC2 instance types," 2022. [Online]. Available: https://aws.amazon.com/ec2/instance-types/

[51] "Lil'blog: Contrastive representation learning," 2021. [Online]. Available: https://lilianweng.github.io/posts/2021-05-31-contrastive/