Towards Practical Driver Cognitive Load Detection Based on Visual Attention Information

by

Cheng Chen Liu

A thesis submitted in conformity with the requirements
for the degree of Master of Applied Science
Graduate Department of Electrical and Computer Engineering
University of Toronto

Abstract

Towards Practical Driver Cognitive Load Detection Based on Visual Attention Information

Cheng Chen Liu

Master of Applied Science

Graduate Department of Electrical and Computer Engineering

University of Toronto

2017

With growing popularities of intelligent in-vehicle technologies, monitoring of driver cognitive load is becoming more important for both safety and comfort concerns. This task is recognized to be challenging due to limited prior knowledge. This thesis explores the feasibility of classifying driver cognitive load levels based on visual attention features.

First, we contribute a dataset collected from 37 experienced drivers. The collection process is designed to elicit three levels of cognitive load effectively. The resulting dataset consists a total of eight measurements, gathered from visual, vehicular, physiological sensors and self-evaluations.

Next, we focus on the eye-tracking modality and propose four features for capturing variations in visual attention intensity and direction. Five machine learning algorithms are applied to the extracted features for subject-independent classification. Issues arose from the machine learning workflow, such as evaluation bias, are examined. The most promising algorithm (Random Forest) achieves 70.3% accuracy on classifying between high and low cognitive load.

Acknowledgements

First, I would like express my deep gratitude for the advise and guidance I received from Professor Konstantinos N. Plataniotis. I could not accomplish this thesis without his valuable insights and feedbacks. Furthermore, I truly appreciate the valuable time from my colleagues in Multimedia Lab, especially Haiyan Xu, spent on advising my presentations.

I sincerely appreciate the opportunity of working with Dengbo He, Professor Birson Donmez, Winnie Chen and Amir Aghaei during the data collection campaign. I would also like to thank Nicole Woosoon and Kathy Huynh for their help.

Finally, I would like to thank my friends for their encouragement and support. Last but not least, I could not be more grateful for the constant comfort and patience I received from my parents throughout my master study.

Contents

Glossary

AU Facial action unit.. 13, 15, 16, 19

BN Bayesian network. 17, 19

DBN Dynamic Bayesian network. 17

EOR Eyes-off-road. 16

faceLAB The commercial eye-tracker used in the experiment. 14

SVM Support vector machine. 15, 17, 19

List of Tables

List of Figures

Chapter 1

Introduction

Driver error is a leading cause of traffic accidents. Besides biological conditions like fatigue or drunkenness, danger can also arise if drivers were overloaded by various demands, such as talking with passengers, driving on an unfamiliar route, or dealing with chaotic circumstances (e.g. weather, traffic). In the past decade, sources of cognitive distraction in the driver cabin have been increasing dramatically by the rapid development of In-Vehicle Intelligent Systems (IVIS). These devices often incorporate functions such as auto speech recognition systems to help reduce the necessary visual or manual interactions; however, these would not alleviate the problem of cognitive overload. Researches and large-scale naturalistic driving studies have found that using hands-free systems still leads to degradation of driving performance [1]. These findings suggest that even with the driver's eyes on the road and hands on the wheel, attention directed towards the driving task could still be insufficient and cause problems.

The above suggested the necessity for assessing driver cognitive load in conventional vehicles, where the human still handle all the driving operations. However, this situation will likely be changed in the near-future as autonomous vehicles are developing rapidly. As the level of automation increases in smart vehicles, they require less human intervention in the operation aspects (e.g. braking or accelerating), which would open more opportunities for drivers to use IVIS. However, there might still exist difficult situations that require intervention from the human driver. Therefore, it is necessary to monitor driver's cognitive load in autonomous vehicles. This function not only helps avoid

accidents but also enhances the driving experience.

This study focuses on the challenging problem of estimating driver's cognitive load when there are no changes in visual or manual demands. This target corresponds to the case that drivers appear to be driving normally, but their minds were heavily occupied by other cognitive activities (such as processing incoming information, or daydreaming). Under this situation, drivers' awareness of the overall driving circumstances might degrade to the extent that harms their ability to make decisions critical to the driving task. To recreate this problem, experienced drivers are recruited to drive with different levels of cognitive load on a driving simulator, while comprehensive measurements from visual, physiological and vehicular sensors are collected. With these data, we explores the feasibility of estimating driver cognitive load with visual features capturing the subject's visual attention variations.

1.1 Background

In this section, two questions are addressed with related background information: 1) what is the target condition our system is trying to detect, and 2) why the video-based approach is chosen. For each part, a more general circumstance will be described first; then, it will be narrowed down to specifically define the problem space for this study.

1.1.1 Definition of Driver Cognitive Load

As both driving task and human mind are very complicated, there also exist various concepts for describing driver's internal cognitive states (e.g. distraction, inattention and arousal). In order to narrow and specify the "cognitive load" problem focused in this thesis, we address the differences and relationships between these concepts.

First, we start with the more prominent and frequently discussed driving problems, distraction and inattention. Their definitions could be unspecific and/or inconsistent across studies. Regan et al. reviewed and standardize these two definitions in order to provide a common ground for future dataset labeling and research discussions [2]. According to their definition, **driver inattention** is *"insufficient, or no attention, to*

activities critical for safe driving". Under this broad concept, sub-categories are identified based on the different causes of the inattention (e.g. sleepiness). Note that one of the sub-categories, i.e. "Driver Diverted Attention", accounts for the case in where inattention is caused by secondary non-driving task, which is synonymous to **cognitive distraction** in some other literatures.

On the other hand, mental **arousal** was also widely discussed as it was theorized to influence task performance. According to the famous Yerkes-Dodson law [3] illustrated in Figure 1.1, the left end of the curve corresponds to the danger of driving under low arousal situations (such as fatigue or low-vigilance); on the other side, when workload is too high and causes anxiety or stress, driving performance also degrades. When adopting the concept of arousal in driver monitoring studies, it was linked with mental workload [4].

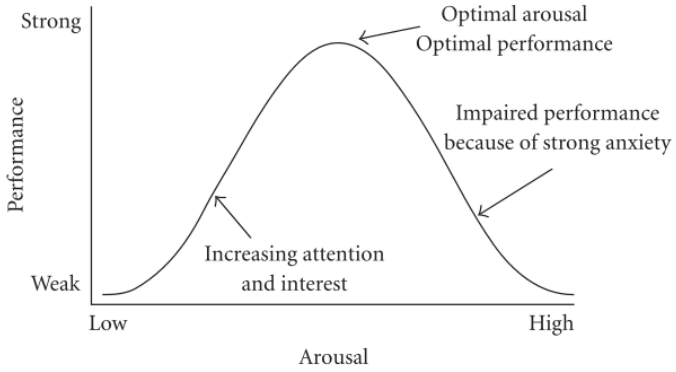


Figure 1.1: The Yerkes–Dodson law models an empirical relationship of task performance vs. mental arousal.

Finally, by definition, **cognitive load** refers to "the total amount of mental effort being used in the working memory" [5]. In field of driver monitoring, it is loosely used to describe the workload allocated for perceiving, processing and organizing information. When the driver is cognitively overloaded by secondary tasks (or other non-driving conditions), it could be problematic since it deprives resources allocated for maintaining driving safety (e.g. situational awareness and operational competence). On the other hand, immense cognitive load could also result in high arousal, and cause problems such as agitation or stress. Therefore, high cognitive load could lead to distraction and/or

high arousal.

Above paragraphs introduced three different aspects to describe driver's cognitive states. To compare and understand their correlations, Table 1.1 enumerates some example driving situations where each of the individual factor is active or not. Note that when there is no added cognitive load, there are still cases that the driver could become inattentive or agitated. On the other hand, even under the presence of an additional task load, there are still cases drivers may be competent and attentive to driving.

Table 1.1: Driving Cognitive States with Example Situations

High Cognitive Load	**Distraction**	**High Arousal**	**Possible Driver Situation**
0	0	0	Calm, comfortable driving.
0	0	1	Nervous due to bad weather (e.g. snow storm).
0	1	0	Operating air conditioning controls.
0	1	1	Crying baby in the backseats.
1	0	0	Daydreaming, or listening to radio.
1	0	1	Following GPS on unfamiliar routes stressfully.
1	1	0	Attempting to interact with a voice-command system.
1	1	1	Frustrated by an important phone conversation.

This study is targeting estimation of driver cognitive load, corresponding to the first column of Table 1.1. Depending on the difficulties of secondary tasks and driver's own capability, this might lead to either inattentive driving or mental stress, or even both of them; but these two conditions could also occur without added load, which would not be discussed in our research (the top half of Table 1.1). On the other hand, there is also advantage in ensuring data validity of the target driving problem was cognitive load. This is because the objectively controlled task conditions are more directly linked to cognitive load, but would need another layer of assumption to infer that cognitive distraction or anxiety were induced. A downside of focusing on cognitive load is that the driver might be still fully competent of driving with the added load; to address this, the data collection could be designed to carefully controlling other external conditions (e.g.

task difficulty) to induce the more threatening conditions.

1.1.2 Video-based Approach for Driver Monitoring

There exist many different approaches to analyze or estimate a variety of driving problems. Generally speaking, driver monitoring approaches could be categorized into the following four types [6, 7]:

- **Video-based measures**: observing voluntary or involuntary behavioral changes induced by different driver conditions.
  - Examples: blink, gaze direction, facial expressions, and body gestures.
  - Pros: fast, sensitive (comparing to vehicle-based measures) while also non-intrusive.
  - Cons: more complicated to acquire the input features from raw visual inputs, which also incures more computational expense.
- **Vehicle-based measures**: suboptimal driving states could result in degraded driving performance that could be captured by sensors on vehicle's CAN-bus.
  - Examples: lane deviation and speed control, steering wheel angle, operational mistakes, brake response time.
  - Pros: convenient to acquire, high user acceptance, mature and applied in commercial products [8].
  - Cons: for real-time applications, this type of measurement may not be sensitive enough since driver problems often do not manifest in degraded driving performance immediately.
- **Physiological measures**: directly obtain biological signals to observe driver's internal conditions.
  - Examples: Electrocardiogram (ECG), Electroencephalogram (EEG), skin conductance level, blood pressure, respiration rate.

  - Pros: high sensitivity and fast responses.
  - Cons: not practical as the biological sensors often needs to be attached on driver's body (i.e. relies on intrusive sensors), and signal might not be very robust when outside of laboratory [9].

- **Subjective measures**: evaluating driving states based on human evaluators observation or participants' self-reports.
  - Examples: NASA Task Load Index (NASA-TLX), Karolinska Sleepiness Scale (KSS)
  - Pros: direct, intuitive and gives insight of what was happening internal of the driver.
  - Cons: subjective and often not continuously measurable (if it was self-reports) or very labor-intensive (if it was expert-ratings).

For a practical real-time driver monitoring system, it is inconvenient and unrealistic to rely on subjective measures or apply intrusive physiological sensors on drivers. Therefore, we narrow down to external observations of driver's behaviors. Amongst the non-contact measures, vehicle-based measures alone might be insensitive to immediate changes of driver conditions, thus video-based signals will be focused in this study. An ideal video-based driver monitoring system should function similar to a co-pilot who continuously supervises how the driver is performing [10]. While the information available in this aspect is rich and useful, there are also more challenges associated with this approach.

As illustrated in Figure 1.2, a complete video-based driver monitoring pipeline is commonly constructed from several modules. The pipeline starts with an Image Acquisition system. For driver monitoring, use of hardware such as active lighting (e.g. Near-Infrared LED and Camera) is a typical way to help alleviate the problem of lighting variation [11, 12] and obtain crisp, high-quality visuals. The acquired video is fed into the next module for Detection and Tracking of features or landmarks. Then, another High-Level Feature Extraction is often performed to The extracted signals will then be used together for inference and classification, where fusion with information obtained

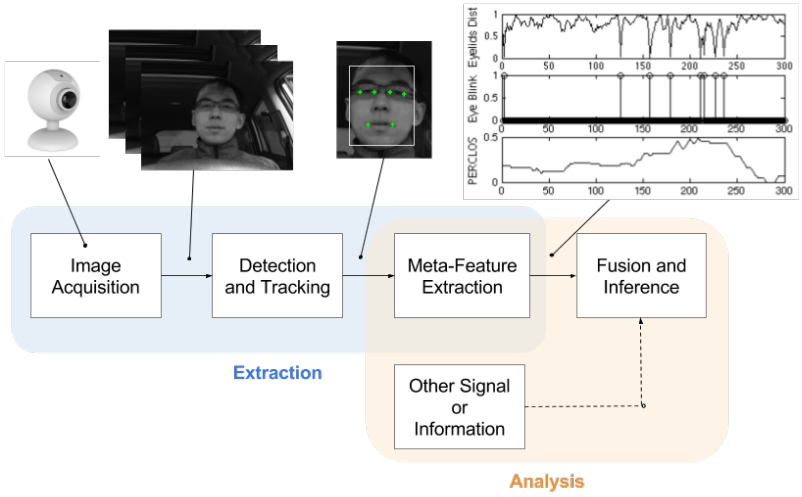


Figure 1.2: General pipeline of a complete video-based driver monitoring system.

from other sources (such as vehicle-based measures) might happen as well.

1.2 Thesis Scope and Challenges

Although there have been significant developments in detecting various driving problems [6, 13, 14], major gaps still exist for building a practical monitoring system targeting driver cognitive load. This thesis aims to develop an estimation model of driver cognitive load, which could be potentially employed by smart-vehicles to cope with driver cognitive overload. Specifically speaking, this study targets the condition when the driver is cognitively loaded, but not obviously driving carelessly (e.g. looking away from the road or taking hands off the wheel). To ensure the usefulness and user acceptance, the evidences for estimation must come from objective observations that could be acquired conveniently, such as remote cameras. Within this modality, a promising indicator emerged from previous human-factor studies was driver visual attention variations.

To achieve the final goal of this study, the research process could be broken into two main objectives. They are described in the ensuing subsections, with explanations of challenges associated with them.

1.2.1 Data Collection

The first objective is to gather data that can effectively represent drivers' responses under higher cognitive load. The main obstacle with this task is that the target problem (cognitive load) could not be easily determined objectively. Thus, modeling of this state via secondary tasks needs to be implemented. However, participants are highly likely to be influenced by other factors as well, such as the external driving environment or their own capability in information processing and multi-tasking. This posed additional difficulties on experimental design and execution. Previous studies could overlook some of these factors, and their datasets are seldom publicly available or suitable for the current study. Therefore, as our first task of conducting research in this area, we collect a solid dataset that not only facilitates the current study, but also potentially enables future studies beyond this thesis. The data collection experiment needs to be designed with many considerations in order to make the effects of experimental conditions (driving and cognitive tasks) consistent across participants. Collection of a comprehensive dataset also requires great commitment in terms of time and energy. The resulted dataset will be available for other researchers interested in studying driver cognitive load.

The research hypothesis for this phase is that with the proposed experimental design, participants' cognitive load is effectively altered in the collected dataset.

1.2.2 Driver Cognitive Load Estimation

The next objective is to explore whether using visual attention information to classify driver cognitive load would be feasible or not. The fundamental difficulty with developing a driver monitoring system targeting cognitive load is the lack of apparent, external observations that can be relied on for *predicting* the internal cognitive state changes (like drooping eyelids for fatigue, or deviated gaze direction for visual distractions). Several studies observed that drivers reduced their visual attention as their mind become heavily occupied, but their observations were obtained over a long period of time (e.g. one minute). Based on this prior knowledge, we propose features to describe visual attention characteristics such as frequent blinking or reduced peripheral/instrumental checks.

These features would be potentially feasible to capture with cameras, though they are extracted from eye-tracker outputs in this study.

We explore machine learning algorithms for building subject-independent estimation models based on these features. This approach were found to be advantageous because machine learning could automatically extract the discriminative power from input instances. We evaluate how well the proposed combination of features and modeling algorithms perform, and discuss issues associated with this process. The research hypothesis for this phase is that visual attention patterns are indeed promising indicators for estimating cognitive load levels.

1.3 Chapter Summary

In this chapter, we motivated for monitoring driver's cognitive workload. Various driving problems were listed and compared to specifically define the estimation target. Advantages and challenges for using a video-based monitoring approach and the potential features (visual attention information) were also provided. Then, the objectives and research questions were identified. The first stage of this research targets collection of representative data samples of high driver cognitive load. Then, the ensuing analysis performed based on this dataset aims to find promising features and modeling methods, which are the crucial components of a practical monitoring system.

1.3.1 Thesis Organization

This study focuses on classifying driver's cognitive load based on visual attention information. In this chapter, the background and research scope are identified. Chapter 2 is devoted to reviewing studies related to monitoring driver cognitive load with visual features. Chapter 3 describes the design and collection process of a dataset that effectively captures drivers' responses from incremental cognitive load. Chapter 4 focuses on developing an estimation model based on eye-tracking data from the dataset. It explains rationales for the methodologies, details of the implementation, and implications of the obtained results. Finally, Chapter 5 concludes contributions of this thesis, and provides

suggestions for future research.

Chapter 2

Related Work

Over the last decade, there have been some initial development towards detection of driver cognitive states based on non-intrusive signals (acquired passively from sensors like remote cameras or vehicle CAN-bus). Recently, subject-independent systems with real-time capability has been introduced, and promising detection accuracy was reported. The success of these systems could be attributed to two related areas: 1) findings of measurements sensitive to added cognitive loads during driving, and 2) the recent advancements in computer vision technologies and machine learning (ML) algorithms.

This chapter reviews the most relevant studies that contributed to the development of driver's cognitive load monitoring system. The scope is constrained to focus on studies under driving scenarios, because the added complexities from multitasking between driving and handling cognitive task differentiate this condition from other simpler setups (e.g. solely performing a designated task). On the other hand, although we focus on video-based signals as it is the interest of this research, the methodological and theoretical progressions might also apply to other researches that explore machine learning approaches to build estimation models for real-time applications. This chapter consists of three parts: 1) Section 2.1 introduces important statistical findings connecting increased driver cognitive load with variations of eye-related measures, 2) Section 2.2 reviews state-of-the-arts in video-based driver cognitive load estimation systems, and 3) Section 2.3 briefly visits recent advancements in hardware, computer vision and machine learning algorithms, which could have already been applied to the systems, or to be explored in

the future.

2.1 Sensitive Measurements of Driver Cognitive Load

A key challenge of the current research is identifying the changes of an internal state, i.e. driver cognitive load, using only external, objective measurements. Therefore, before reviewing state-of-the-arts solutions that can detect cognitive load based on observations, we first introduce prior findings that link cognitive load to certain measurements characterizing driver behaviors, especially the ones extracted from visual signals[15, 16, 17, 18]. From these studies, significant impact of cognitive load on driver's visual attention was consistently observed.

A popular concept in this area is that when the cognitive load increases, driver's visual attention would become narrowed (i.e. spend more time staring centrally ahead). This is often referred as gaze concentration or visual tunneling. This phenomenon was observed in a study that analyzed data from 119 subjects collected under naturalistic or simulated driving [15]. They found the auditory cognitive task had a statistical significant effect on two measures of gaze concentration: it mildly increased the Percentage of Road Center (PRC) and evidently reduced the standard deviation of the *combined* gaze direction (which is calculated using the Pythagorean theorem). Another naturalistic on-road experiment characterized the concentration of visual attention as reduced glances to non-road-center areas, which includes peripheral regions, odometers and mirrors [16]. As this finding suggested degraded safety awareness, it also revealed that hands-free systems might not reduce the harm of distraction. Further evidence of narrowed visual attention field was provided by a series of on-road experiments [17, 18]. The most pronounced measurement discovered here was the standard deviation of horizontal gaze dispersion. Since the above mentioned studies applied slightly different methods for measuring gaze concentration, a more recent analysis compared these methods based on a single set of data [19]. They suggested that 1) SD of horizontal gaze is more sensitive than the SD of combined gaze direction, and 2) a larger road center criteria (12° or 16°) for calculating PRC would lead to better sensitivity. Overall, although the specific measurement might

differ, gaze concentration was consistently found under increased cognitive load.

Another perspective of interpreting the impact of cognitive load on visual attention is that they compete for resources: when one of them is under higher demand, the other one would be suppressed. Blinks have been a popular index in psychophysiology studies as a measure for high cognitive load, information processing and initial stimuli perception [20]. In the driving domain, increased blinks was also found to be a sensitive measurement to high cognitive load [21]. However, blink measurements might not be very reliable for assessing drivers as the heavy visual demands associated with driving could inhibit blinks, and thus mixed results were found [22]. On the other hand, impairment of visual attention due to increased cognitive load was also linked with elongated gaze fixation [23], which could be interpreted as requiring longer time for understanding the visual inputs. But gaze fixation length is highly overlapping with gaze dispersion (e.g. SD of gaze position), and the latter is more commonly exploited. To summarize, it is intuitively reasonable to expect a decrease of visual attention intensity with added cognitive load; however, measurement of visual attention intensity could be sensitive to various uninterested factors as well.

2.1.1 Relationship to Cognitive Load Estimation

Although conclusions in this field served as the foundation for many proposed detection systems (as will be reviewed next in Section 2.2), they are completely different types of studies which could be easily confused. As illustrated by Figure 2.1, the fundamental difference is when trying to build a model relating the underlying condition and some observations, which side of the graph is considered as independent (known) or dependent (unknown). Studies in this section consider the cognitive load as the independent variable, and models are built to understand if certain measures would be impacted by changes of the independent variable or not. On the contrary, in a practical detection system, the input to the model would be observations (e.g. visual attention features), and the target output would be predictions of the unknown underlying condition; the model would be evaluated by the correctness of its predictions. In another word, the difference between these two analysis is whether the model is constructed for understanding

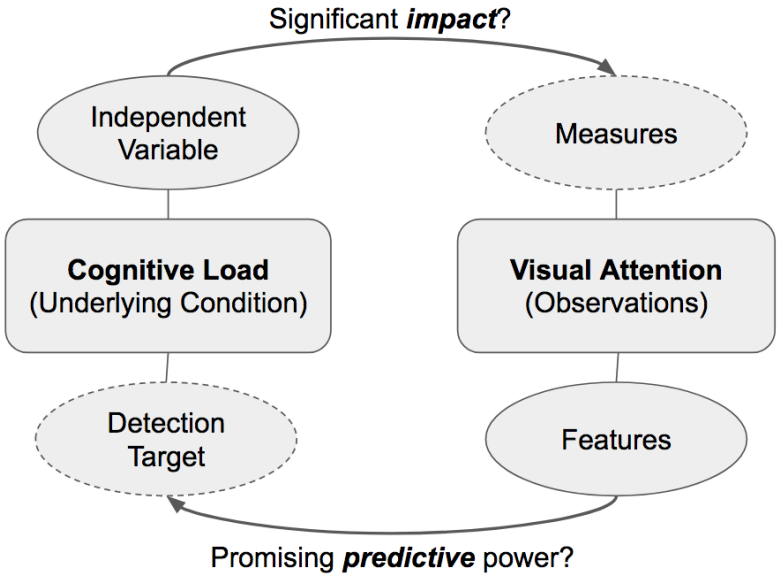


Figure 2.1: The difference between analyzing the impact of cognitive load on visual attention (Section 2.1), and using visual attention as features to predict cognitive load (Section 2.2).

or estimation/prediction.

There are a few circumstances about the statistical analysis presented above that might be beneficial to consider when trying to use their findings. First, the measures in these studies were extracted from longer time periods (the length of a trail or session), thus it could not be directly applied for driver monitoring systems that desire prompt feedbacks. Secondly, statistical models are built to test certain hypothesis (e.g. whether the difference between the measurements obtained under high or normal cognitive load are statistically significant or not), which results in models aimed for understanding rather than estimation. There are also limited freedom in terms of model construction techniques. Finally, these analysis could only investigate one measure at a time, and thus could not take advantage of all available signal modalities.

2.2 Detection of High Cognitive Load

As the knowledge of how visual attention changes due to high cognitive load accumulated, several studies proposed prediction models and reported promising results (81.1% test accuracy [24]). The most typical machine learning framework to this problem is depicted in Figure 2.2. Under this framework, inputs to the classification model are usually some summarization or characterization features extracted from a sliding window over the raw signals. Then, the classification model is trained using supervised learning algorithms. When applying this system for real-time evaluation, the same methods are applied to extract features from raw signals, and the pre-trained model predicts the driver states based on the these feature values.

Previous development of driver cognitive load detection systems often focused on two key questions: 1) which features to use, and 2) how to build the prediction model. Both of these would contribute to the overall system performance jointly, but in most cases they were treated and discussed separately. Thus, these two aspects are also reviewed in separate subsections bellow. In addition to these theoretical developments, practical but also crucial procedures (such as data collection and model evaluation) are also discussed.

2.2.1 Problem Definitions and Data Collection Methods

Development towards a driver cognitive state monitoring system could consist of a variety of focuses (as listed in Table 2.1). In terms of the detection target, although the current study is interested in high cognitive load, we also include studies focusing on cognitive distraction. As clarified in Subsection 1.1.1, these are two different concepts to describe driver's cognitive states. However, in this literature review, cognitive distraction will be considered as a case of high cognitive load. The reason for this was rooted in the data collection methodologies common to both kinds of studies: the targeted states were always elicited by adding secondary tasks onto driving scenarios.

To emphasis variations in cognitive states, it is common to employ some kind of auditory task such that other non-cognitive demands (e.g. visual or manual) would not be varied by the task itself. The task could be more realistic ones, such as conversations

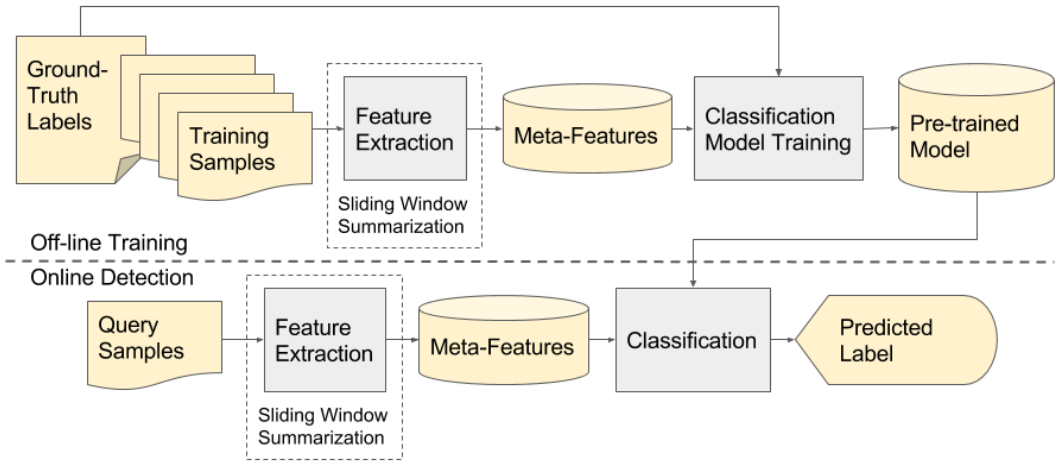


Figure 2.2: A typical machine learning framework for assessing driver's cognitive states. This approach involves training a prediction model using labeled data first, and then apply it for real-time detection.

[25, 26] or listening to an auditory stock ticker to capture the trend of target stocks [24]. On the other hand, for better control and simplicity, it is also beneficial to use surrogate (meaning artificial replacement) tasks, for example the n-back task [27], target sound counting task [28] or arithmetic task [26]. All of these tasks involved stimuli perception, short-term memory maintenance and information processing. Therefore, it is safe to assume that engaging in these tasks increases the subject's cognitive load. However, the same might not be claimed for cognitive distraction, since none of the task conditions explicitly divert attention from the primary driving task: the participant might still be able to drive with adequate attention while multitasking to handle the added cognitive tasks. For this reason, even for studies targeting cognitive distractions, considering the classification targets as high cognitive load should hold equivalent (if not more) validity.

This led to another key problem of data collection in these studies, which is obtaining the ground-truth labels. Accurate and unbiased labeling is an important prerequisite for applying supervised machine learning algorithms, which is the most common approach for constructing an estimation model in this field. Unlike many other problems, the ground-

Table 2.1: Overview of Studies Towards Assessing Driver's Cognitive States

Research Focus	**Features**	**Modeling Method**	**Findings**
Automatic information extraction with machine learning methods for driver workload estimation [31].	Gaze fixation duration, pupil diameter, lane position deviation	Decision tree with boosting	Best average correct detection rate of 81% when using all features and longest time window (30 sec). Eye-based features were more predictive than driving-based, and pupil diameter features (SD) were the most important eye feature.
Real-time driver cognitive distraction detection based on eye movement patterns [24].	Eye movement patterns, driving measures.	Logistic regression, support vector machine (SVM)s.	Across all models and participants, average detection accuracy is 81.1% (best 96.08% when using 40-sec window with 95% overlap). All input features contributed to the classification.
Comparison of modeling methodologies and parameters [24, 32].	Same as above.	Logistic regression, SVM, SBN and DBN.	DBN was found to be the best method as it achieved best sensitivity and accuracy, while SVM also achieved comparable accuracy. Summarizing time window may not have significant effect on the model performance.
Predicting cognitive distraction levels with high granularity [25].	Facial Action Unit (AU)s, auditory responses, vehicle measures.	Linear regression with regularization.	Average correlation score for cognitive distraction is 0.645, around 20 out of 348 features were selected. List of most frequently selected features.
Classifying between high or low cognitive distraction [25].	Same as above.	kMeans, SVM, Random Under Sampling Boosting and more	List of most frequently selected features (using forward feature selection analysis), showing some overlapping with the regression model. Lip-tightener IQR was often selected.
Using semi-supervised learning techniques to take advantage of unlabeled data [28].	Head pose, gaze and eye measures (includes saccade, blink, PERCLOS), driving measures	Semi-supervised or supervised version of SVM and ELM	Semi-supervised methods outperformed their supervised counterparts as unlabeled data can improve the model performance.

truth label of driver cognitive load could be vague, uncertain and unknown to researchers or other evaluators. In general, there exited three bases of obtaining ground-truth labels:

- **Task Conditions**: The simplest and most common labeling method was categorizing data from driving periods with or without secondary tasks into two distinct classes. Notably, this was also the method adopted by the studies on finding sensitive measures to driver cognitive load [15, 16, 17]. Variation of this method might consider longer or peripheral periods (meaning before or after the task engagement) rather than only strictly the periods of secondary task engagement. Different ways to define the "distracted" period was discussed in [24]: the "DRIVE" definition (the whole drive, including samples in between tasked engagements) was found to lead to best model performance, and thus was applied in the subsequent study using the same dataset [29] (though this logistic might introduce bias). Although altering the external conditions was widely accepted as a way of modeling cognitive states in research studies, it is generally agreed that the true cognitive state induced by the same condition could vary between people and change over time in reality. This is a major disadvantage of this labeling strategy.

- **Self-evaluation**: To address the above-mentioned problem of labeling based on the objective task conditions, an alternative way of labeling ground truth with the participants' self-evaluation was applied in [30]. After obtaining subjects' ratings at the end of each driver, a further step was taken to transform the scalar values into binary classes. However, because self-evaluation could only be collected at the end of each trail and participants could rate the same difficulty/workload differently, this information might still be inaccurate or biased.

- **External-evaluation**: The third way of labeling ground truth was based on evaluations from external evaluators. This was used to obtain continuous-valued ratings for visual and cognitive distractions in [25] (which was also transformed into binary classes later). In this study, each 10-second video segment was rated by three external evaluators. The correlation of results from all participated evaluators (a total of 30) were calculated to demonstrate the reliability and consistency of the

ratings. This process needed a lot of time and energy, and thus limited the amount of data that could be handled this way [25].

Finally, as part of a discussion on building dataset for studying a problem with (supervised) machine learning and data mining techniques, it might be worth noting the quantity and quality of the datasets. The raw data corpus were often obtained from 8 [26] to 34 [28] participants, with consistent or varying road conditions (e.g. highway, rural or urban) and secondary task conditions. The focus periods would also be cropped out from the raw recordings by different standards, depending on the specific definition of the targeted state adopted by each study (as discussed in the previous paragraph). All these factors differ across studies, resulting in various sizes and attributes in the final dataset. In general, previous data collection campaigns usually resulted in around 80 to 500 minutes of raw recordings that could be further processed and analyzed.

From here, the number of samples for each class would be determined by the research problem formulation as well as the employed feature extraction procedures. For example, constructing subject-independent models could use all available data (e.g. 4800 seconds in Li et al. [25]), while subject-dependent models could only take samples from the same subject (around 500 seconds for each model Liu et al. [28]). On the other hand, the feature extraction and outliers elimination process could also result in small number of samples. In the above mentioned case, with the same sliding window length for feature extraction (10 seconds), Li et al. uses non-overlapping window and extracted 480 instances from 4800 seconds of recording [25]; on the other hand, Liu et al. applied 95% overlapping, and thus obtained an average of 1053 instances from around 500 seconds of recording [28]. It is generally agreed that more data is always beneficial for training machine learning models as well as obtaining better generalization evaluation of the model. Thus, the number of instances that was extracted and used for machine learning should be considered when understanding a study.

To summarize, this subsection reviewed the data collection process for developing driver cognitive state monitoring systems, and justified the similarities of studies on high cognitive load and cognitive distraction detection. Specifically speaking, the methodologies and resulted dataset of the different data collection processes were reviewed. As

compared in Table 2.1, further discussions and comparisons on feature extraction or modeling methodologies will be presented in the next two subsections.

2.2.2 Feature Extraction and Selection

In the initial research, it was recognized that there exists no established index that could be used to assess internal cognitive load quickly [31] (similar to the "PERCLOS" index [33] for evaluating fatigue problems). As changes in visual attention have been reliably found with added cognitive load, these measures were frequently adopted as candidate features for cognitive load detection systems. To date, the selection of the most effective predictive features remains an open question. Table 2.2 provides an list of features that have been employed for detecting driver's cognitive load.

In general, selecting features to be exploited by the detection system could be classified into two approaches [32]:

- The top-down approach focuses on a smaller set of hand-picked features, which were selected based on prior knowledge of their sensitivities to higher cognitive loads.
- The bottom-up approach considers a large pool of features, and employ feature selection methods to select the ones that are more informative.

Hand-picked Features and Comparison Analysis

In most studies, due to the nature and limitation of the ensuing classification algorithm, a smaller number (e.g. 10) of features must be selected based on prior knowledge. The candidate features were usually measurements that were found sensitive to added cognitive load (as reviewed in Section 2.1); but instead of summarize over a whole trial, they are now calculated over a much smaller window size such that the system could be more practical and suitable for real-time applications. After picking the features, one common method to evaluate the importance of an individual feature or a group of features was examining the effect of adding/removing it from the classification inputs.

Eye movement patterns were exploited for real-time detection of driver cognitive in [24]. These patterns were generated by categorizing segments into fixations, saccades and

Table 2.2: Summary of Features Employed for Detecting Driver Cognitive Load

Features		Studies						
Gaze/Eye States	**Summarization Function**	Zhang et al. [31]	Liang et al. [24]	Liang et al. [29]	Miyaji et al. [26]	Son et al. [27]	Li et al. [25]	Liu et al. [28]
Pupil diameter	Mean	✓			✓			
	SD	✓						
Blink	Frequency			✓			✓	✓
	Duration							✓
Eye Closure	PERCLOS							✓
Gaze entries to a region	Count, duration	✓					✓	
Gaze position (X, Y)	Mean		✓	✓				
	SD		✓	✓		✓		
Gaze rotation (yaw, pitch)	Mean, SD							✓
	Combined SD				✓			
Fixation	Duration		✓	✓				
Pursuit	Duration, distance, direction		✓	✓				
Head rotation	Mean						✓	✓
	Combined SD				✓			
Facial AUs	Various statistics						✓	

smooth pursuits based on raw eye data's dispersion and velocity. Then, combined with several vehicle measures, the signals were sliced into instances of certain time windows to form inputs for SVM models with radial basis function (RBF) kernel to perform binary classification. They performed posthoc comparisons using the Turkey-Kramer method to verify that including mean gaze fixation position and vehicle-based measures both contributed to the model performance. However, the effectiveness of each input features or feature selection were not discussed. In their later work [29], instead of eye movement and fixation features, blink frequency was found to be most important for detecting cognitive distraction based on analysis of mutual information.

Liu et al. considered four feature categories [28], all obtained from eye-tracker results. Saccade, blinks and PERCLOS were included in the "Gaze Temporal" feature group, while the other three feature groups are "Head Position", "Head Rotation", and

"Gaze Rotation". Individual features were ranked based on correlation coefficient, class separability with Linear Discriminative Analysis (LDA) [34], and feedforward feature selection frequency using a LDA classifier. Looking at the average results of each feature group, all three analysis suggested the Gaze Temporal group to be the most important one, followed by Head Rotation. In the end, they acknowledged that all features could contribute to distraction detection for different individuals, and thus were all kept for the final modeling and analysis.

Following a similar methodology, later works explored other gaze features for driver's cognitive load detection. The combined gaze rotation angles and head rotation angle were considered as the baseline and achieved promising classification performance, but including features for pupil diameter and heart rate inter-beat intervals [26] showed further enhancement. The location of gaze intersection was used instead of gaze rotation angle in [27]. In this study, driving performance features are considered as the additional information. The best model performance was achieved by using the combined gaze intersection and SD of lane position, rather than all available features. These two studies only compared feature combinations based on average model performance scores.

Data Mining for Feature Selection

The bottom-up approach is considered advantageous as the responsibility of feature extraction is transfered to the machine learning algorithms. This type of solution must work with classification algorithms that could innately cope with large number of features and automatically decide which ones were more useful in the modeling.

This approach was first explored in one of the earliest studies on using machine learning to assess drivers' workload [31]. This work used decision tree models and all possible inputs were considered. They collected gaze position, pupil diameter, and various vehicle-based measures. This study posed no assumption on which sets of signals might bear more information. From past experience, they believed that frequency of feature selected for building the decision tree could reflect its predictive power. This analysis revealed the SD of pupil diameter to be the most important feature, which was generally agreed to be a good index of cognitive workload [21].

Most of above-mentioned studies acknowledged that every feature could contribute to the detection (although the degree could vary), and identifying redundant features was not really necessary since there were not huge number of possible candidates. However, in [25], a much larger variety of multimodal signals were measured for predicting visual and cognitive distractions separately or jointly. The authors obtained 20 facial AUs, coarse gaze and head movement using the CERT software [35]. Blinking was calculated based on the eye closure AU. This study also took one gaze-based measure (i.e. Eyes-off-Road, EOR) into account, however it was only an approximation estimated using drivers' head position/rotation. Vehicle measures, road camera videos and a microphone array were also used as inputs. All of the raw signals underwent further computation to obtain high level features (e.g. mean, SD, max/min, interquartile range), resulting in a final input vector of 348 dimensions (43 low-level feature times 8 high-level summarizer, plus 4 gaze features). In order to build more efficient model, regularization was applied when doing linear regression, and eliminated down to about 20 features. Individual features were compared based on their frequencies of being selected for constructing models. Overall, the Lip Tightener (AU23) interquartile range (IQR) appeared to be useful for detecting all kinds of distractions, while EOR duration and Outer Brow Raiser (AU2) Max were consistently selected for models targeting cognitive distraction.

Summary and Remarks

To summarize, this subsection reviewed feature extraction for cognitive load detection. Visual attention features such as mean gaze rotation or blink frequency were more commonly employed than other visually observable features like facial expression or body gestures. It could be noted that the specific measurements used for characterizing gaze information could be different (e.g. fixation or saccade, fixation location, or EOR), but the targeted information to extract seems to be highly correlated or overlapping. Blinking, on the other hand, were mostly evaluated by frequency and/or duration. Comparing to the eye-related features, facial expression features received less attention, possibly due to the lack of previous findings to support its correlation with cognitive loads and difficulties in obtaining these measurements. In addition to visual features, other supple-

mentary information (such as driving performance measures) was also often taken into consideration, and was found to help classification performance.

When considering more than one input feature, most studies simply concatenated all features into one vector and feed into the classification algorithm; feature-level fusion or dimension reduction techniques (such as Principal Component Analysis or LDA) was not employed. This might be attributed to the fact that most studies only considered a smaller group of hand-picked features that could be easily handled by the classification algorithms. Nonetheless, to verify the usefulness of including all the features or feature groups, comparison of the overall model performance when including/excluding certain features was commonly conducted. Most studies' findings encourage taking abundant number of features into consideration, then rely on some machine learning mechanisms to construct a predictive model on top of them.

Another important factor that is common to all feature extraction methods is the length of the time window. Only looking at one frame of the input signals and not considering the temporal information would be inefficient for analyzing time series inputs. All of the above studies summarized certain statistic functionals (e.g. mean, SD) over a certain time window to form input instances for ensuing the machine learning algorithms. In Liang et al.'s earlier study, the effect of time window were found to be significant when modeling with SVMs (longer time window would improve the performance) [24], which was consistent with the conclusion in [31]. However, in the later studies using the same dataset featuring DBN as the classification model, the results suggested the effect of time window was not significant [29, 32]. Recent studies often adopted 10-second time window [25, 28].

Considering from the practicality aspects, since the estimation would be performed based on features extracted from a *past* time segment, one could argue that there is no limitation in terms of how long in the past the system could look into (as long as the computational power and storage allows), which implies that it might be unnecessary to limit the window size. On the other hand, most studies assumed driver's cognitive state over the summarization time window to be constant, which could be the case under controlled driving studies - but highly unlikely in reality when the time window becomes

long (e.g. 30 seconds [27]). Therefore, keeping the time window small should actually be considered necessary for making the data samples properer representation of the reality, rather than improving the temporal resolution of the system. Note that the overlapping ratio of time windows should be more responsible for the system's temporal resolution, since it dictates how often the system generates an estimation results. For example, a system that uses 10-second time window with 90% overlapping would generates an estimation result every 1 second; on the contrary, using 5-second time window with no overlapping would lead to obtaining responses every 5 seconds. For future studies, it might be appealing to consider developing systems that are capable of handling varying cognitive states over the observation period, and thus the system could be allowed to take advantage of much longer periods.

2.2.3 Model Construction and Selection

Building a prediction model for detecting high cognitive load during driving is challenging. It was pointed out in an early study that incomplete domain knowledge, inability to simultaneously analyze various signal inputs, and inappropriate assumption (e.g. unimode Gaussian distribution) limited the power of traditional statistical analysis [31]. Therefore, various machine learning algorithms were explored to automatically extract the inherent structure from the training data, by which the dependencies on domain knowledge could be alleviated. The trained model would be tested on other unseen data (as depicted in Figure 2.2 above) to evaluate how well the model could generalize.

In this subsection, the machine learning algorithms previously applied for detecting driver's cognitive load are reviewed. On the other hand, it is also interesting to note the diverseness of evaluation procedures adopted in different studies, and thus are also discussed later in this subsection.

Machine Learning Algorithms

Originally proposed by Vapnik [36], **Support Vector Machine** (SVM) gained enormous popularity after performing well on many classification problems and the convenience of

LIBSVM toolbox [37]. This supervised binary classification method seeks for a decision boundary that best separates the sample points from two classes. Although the decision boundary is linear, the data samples are usually already transformed using non-linear *kernels* like the radial basis function (RBF) before the boundary is sought. By applying different types of kernels and altering the parameters, it is possible to transform complex, non-linear data samples to a space that could be better separated by a linear hyperplane. SVM's advantages in handling more complicated classification problems (i.e. detecting cognitive distraction) were recognized in [24], as the performance of SVM with RBF kernel is compared with a traditional linear classification algorithm, logistic regression. It should be noted that the difference of kernel hyper-parameters might have a dominant effect on the performance of the final model, thus it is considered necessary to search through a variety of hyper-parameters until the most suitable one is found [38].

One extension of the original SVM is the semi-supervised methods for constructing the models, such as the Laplacian SVM explored in a study for developing driver cognitive distraction detection system [28]. Semi-supervised learning (SSL) relaxes the requirement for the labeled training data as it could make use of the unlabeled data. This was especially attractive in studies focusing on driver's cognitive states, in which the ground truth labels are difficult to obtain or could be unreliable. In [28], the supervised and semi-supervised approach of two base classification algorithms were compared: SVM and Extreme Learning Machine (ELM). Laplacian SVM and Semi-supervised ELM both outperformed their supervised counterparts, and the improvement was statistically significant. Furthermore, the performance improved with more unlabeled data, and the size of labeled data has less effect on performance with certain amount of unlabeled data. This verified that SSL methods require less labeled training data comparing to supervised ones.

SVM is an non-probabilistic classifier; on the contrary, **Bayesian Network** (BN) is a type of graphical model designed for representing conditional dependencies between random variables. In Static Bayesian Networks (SBNs), the root of the graph is the hypothesis node, which could represent the underlying condition (e.g. driver's cognitive states). The next layer of this directed hierarchical graph is the hidden nodes, which are

intermediate abstraction that cannot be measured directly but could be important for the belief inference process. The hidden nodes then connect to the observation nodes, which are the measurable features (e.g. blink frequency). An extension of BNs, the Dynamite Bayesian Networks (DBNs), is capable of capturing the temporal relationship by connecting multiple SBNs. Each SBN represents the state at one time step, and the state of the nodes at the later time step is dependent on the ones at the earlier time step.BNs are also capable of handling complex, non-linear classification problems. Since BNs explicitly extract cause-effect links between variables, the trained model is considered interpretable and worth studying to understand human behaviors [39]. In [29], SBNs, and DBNs were applied to detect driver cognitive distraction and their performances were compared. In a later work, they added SVMs for comparison as well [32]. They built groups of models for each individual subjects with varying summarization window sizes, and the comparison is done by analyzing if certain performance scores were significantly correlated to the choice of modeling algorithms or other factors. Overall, DBNs and SVMs achieved the better accuracy than static BNs. Difference in modeling techniques did not significantly impact detection bias or false alarm rate.

Decision tree is a flowchart-like graphical model (a binary tree) in which each node tests one of the input attributes. When going from the root to a leaf, a sequence of binary splits would be made and a final output decision would be reached. As a very early proof of concept, Zhang et al. trained decision tree models to automatically extract knowledge instead of manually designing rules [31]. This is realized by various training algorithms, which all essentially aim to learn the most informative attribute for each node and the values that would make the best splits. In [31], they adopted the C4.5 decision tree learning algorithm [40]. Other algorithms for training decision tree models include Classification and Regression Tree (a.k.a. CART [41]), which differs slightly from C4.5 as it supports numerical target variables.

On top of the algorithms for training individual decision trees, a commonly applied technique is building more than one simple decision tree models using *ensemble* methods such as bootstrap aggregation ("bagging") or boosting. The fundamental idea behind ensemble methods is to construct a more reliable strong classifier with multiple weak

classifiers and a consensus policy. In [31], boosting is applied, which is a iterative learning process where the mis-classified training samples would receive higher weights when training for the next weak classifier; the model's final outcome is a majority voting based on outcomes from all trained weak classifiers. In a later work also focusing on cognitive load [26], a more sophisticated boosting method, called **AdaBoost** (Adaptive Boosting), was featured. It showed superior performance comparing to SVM.

All above studies considered binary classification problems, which could be attributed to the difficulties in obtaining finer granularity in the ground truth label to allow approaches like regression to take advantage of the richer information. In [25], Li et al. collected subjective evaluation from a group of human evaluators to obtain perceived visual and cognitive distraction levels, which was a continuous value on a scale from 0 to 1. This allowed them to build regression models, such as linear regression model with elastic net regularization (i.e. LASSO or ridge optimization) to directly predict for the distraction level. Moreover, they also performed unsupervised k-means clustering to identify four different distraction modes that could be used to characterize visual and cognitive distraction jointly (e.g. low-visual-medium-cognitive distractions, high-visual-high-cognitive distractions). Distraction mode recognition for the four distraction classes was performed using various classification algorithms (e.g. KNN or SVM) and achieved F-scores of 0.4 to 0.5, which are still significantly higher than chances (i.e. 0.25). They also attempted to detect specific distraction modes via binary classification. It is interested to note that in all of their analysis, they reported the most frequently selected features as they adopted machine learning algorithms that could innately select features from a large feature pool of 348 candidates. Their clustering analysis suggested correlation between various secondary tasks with distraction modes, but the relationship is not absolutely corresponding. This work presented a diverse set of analysis conducted using various machine learning techniques, which widened the possible approaches for estimating driver distraction; however, a large gap still exists in applying the findings from this study into practical applications.

Summary and Remarks

Conducting an unbiased performance evaluation consists of choosing the right evaluation metrics and partitioning testing data subsets properly; the former should be chosen per application requirement, while the later should aim to imitate valid generalization situations. In the literatures, how the driver cognitive monitoring systems were evaluated varies in both of these perspectives. This may be due to the nature of how the research problem was framed: as most studies were still exploratory and preliminary works, the proposed solutions should not be taken as working systems and thus could not be subjected to the same, standardized evaluation procedures. Nonetheless, lack of a consistent evaluation method makes the comparison between similar studies difficult. To this end, we summarize the evaluation methods applied in evaluating driver cognitive state detection in Table 2.3.

The most common metric used in previous studies on driver cognitive state detection is the test accuracy (ACC), which is the fraction of correctly predicted test samples of any classes. Though intuitive and popular, this metric is often criticized to be naive and prone to biases. However, limitations and disadvantages could be said for any other binary classification metrics such as sensitivity/recall, precision and F-score [42]. Since there exists no perfect performance metric, the choice of evaluation metric should be subjected to the specific application scenario. In the case of detecting high driver cognitive load, the costs for different types of errors (such as misses or false alarms) are unlikely to be equivalent. Also, the occurrence frequencies of the classes (normal or high cognitive load cases) should be highly unbalanced in reality, which also influences how the system performance should be evaluated. However, to our knowledge, these topics received very limited attention, and most studies in this field applied ACC or F-score as the main performance metric for model selection. Nonetheless, when correctly applied, these metrics could still be quite reliable as a basis of discussing model selection.

One could also find a variety of other metrics (e.g. sensitivity, bias) applied for comparing models constructed using different machine learning algorithms; however the practical significance of these values were not obvious nor explicitly discusses in the case

of driver cognitive monitoring. On top of listing the (averaged) scores in a table and comparing plotted results graphically, several studies also conducted statistical tests to conclude if impacts on a metric from certain factors was significant or not (e.g. [24]). These analysis led to suggestions for choosing modeling methods or features; however, the core question of how well the detection algorithm performs in a practical sense was less touched. An agreement on how to score (or determine) the successfulness of a driver cognitive load detection system is yet to be reached.

Another important aspect of obtaining proper model evaluation is reducing the bias embedded in the dataset. Because there is limited labeled data available for research studies, cross validation (CV) techniques were often adopted to repeat the modeling and evaluation process multiple times with the training/testing sets partitioned differently in each iteration. By evaluating the overall performance across all the CV iterations, the possibility of obtaining biased evaluation results by chance is increased. Although this general notion was well-received, the specific implementation could vary greatly in most aforementioned studies (as summarized in Table 2.3). There are different CV methodologies, such as leave-one-out CV (LOOCV) or k-fold CV; these guide how the training/testing sets systematically changes from iteration to iteration such that the whole dataset could be covered evenly. When the CV method does not exhaustively cover all possible permutation scenarios, the whole CV process could be repeated to further increase the robustness and fairness of the evaluation.

On the other hand, within each CV iteration, how the instances are assigned into training and testing sets could also differ. For example, in many cases, the assignment is done by randomly drawing instances into two sets; however, this would only be valid if all the instances are independent and identically distributed (i.i.d.) [43]. But this i.i.d. assumption would be violated in the case of time series data or grouped data (when collected from different subject), which is especially likely when the instances are summarization features extracted from neighboring, overlapping time windows (such as in [24, 30]). The problem of partitioning training/testing sets with time series data has received attention and what should be considered as "unseen" data was discussed in the general sense [43]. Some of these suggestions were adopted in more recent studies [25, 28],

where the partitioning is performed based on grouping of time-trunks or subjects. In a few other studies, this bias was also mitigated as there was no overlap between neighboring time windows [27]. However, there are several studies that applied random draw or did not specify their methodologies for partitioning dataset (e.g. [31].

Overall, as reflected in Table 2.3, there exist considerate diverseness in methods of performing CV, partitioning dataset, and scoring performance. These all posed obstacle in comparing and understanding the findings from different studies. Also, in some of the studies, it was possible that the bias from highly auto-correlated time series data contributed to present an overly optimistic result.

Model Construction and Selection Summary

To summarize, lack of established domain knowledge and complexity of the problem demanded the use of automatic information extraction process, which calls for the use of rapidly developing machine learning algorithms. Studies investigated if there exists more suitable algorithms, and validated that non-linearity, multi-modality and temporal correlation should be taken into consideration when constructing models for the complex task of assessing driver's cognitive states [32]. When a large number of models were created, statistical analysis might be applied to find correlations between the modeling technique and resulted performance. All of the studies in this field adopted existing machine learning algorithms and toolboxes to handle the model training and testing process; nowadays, there exist comprehensive machine learning toolbox in MATLAB or Python that included most of the methods applied in the aforementioned studies.

2.3 Related Advancements

2.3.1 Machine Vision Technologies

Unlike reading in signals from physiological or vehicular sensors, visual inputs need an extra fundamental step for extracting useful information (e.g. the location of facial fiducial points) from the images. Driver monitoring has always been a very important

Table 2.3: Summary of Model Selection and Evaluation Schemes

Study	**CV and Dataset Partition**	**Modeling Method**	**Performance Metrics and Results**
Zhang et al. 2004 [31].	Not specified.	Subject-independent binary classification using C4.5 decision tree (with boosting). 30-sec time window.	ACC 81%
Liang et al. 2007 [24]	Randomly select 200 instances for training and use the rest for validation.*	Subject-dependent binary classification using SVMs, consider all features and apply longest time window.	Average ACC 83.15%
Liang et al. 2008 [32]	Randomly select instances into training and testing sets with ratio 2:1.*	Subject-dependent binary classification with SVM, SBN, DBN.	Average ACC 86.4% with DBN.
Miyaji et al. 2009 [26]	2-fold CV, seems to be partitioned with time-based trunks (not specified).	Subject-independent binary classification with AdaBoost, SVM.	ACC 91.6% with AdaBoost
Son et al. 2012 [27]	2-fold CV with alternating train/test assignment between neighboring samples. (samples are non-overlapping).	Subject-independent binary classification with radial basis function neural network based on gaze X+Y and lane position	ACC 85.0%
Li et al. 2015 [25]	20-fold cross validation, with subject-based partitioning (leave-one-subject-out). Each fold has 24 samples.	Subject-independent binary classification of high/low cognitive distraction level	Precision, Recall, F-score of the overall performance is around 0.7
Li et al. 2015 [25]	Same as above.	Recognition of distraction type (e.g. Low-Visual-Medium-Cognitive vs. other cases).	Best model achieved F-score of 0.684.
Liu et al. 2016 [28]	4-fold CV, partitioned based on time-grouping, repeated 3 times.	Subject-dependent binary classification using semi-supervised SVM (LapSVM) and ELM.	ACC 97.2%. Semi-supervised methods improved G-mean by 0.0245.
Zhang et al. 2017 [30]	5-fold CV with random sample partitioning*, repeated 10 times.	Binary classification of SVM, kNN, Decision Tree	ACC 84.43% with 1NN and feature-level fusion

Note: * denotes that the feature extraction process in these studies generated instances from overlapping time windows.

application for face and/or eye tracking. Throughout the years, there have been methods developed specially for the driver monitoring task or for other general purposes. This section reviews how these methods evolved, and what are some new advancements that could potentially benefit the development of driver monitoring. Here we are only consider non-contact systems, thus head-mounted eye-trackers are not considered.

Hardware Approaches

Although many recent face alignment algorithms work in a coarse-to-fine manner, the earliest real-time driver monitoring systems actually detects a detailed feature, pupil, directly. The special optical property of human eyes, corneal reflection (often referred as "bright/dark pupil effect"), is exploited for direct pupil detection [11, 44]. This method requires two sets of non-collimated IR lights and an IR camera to be placed in a specific structure, such that when the coaxial set of lights are lit the pupils are bright and when the offset lights are lit the pupils are dark (as illustrated in Fig. 2.3). Two sets of lights are lit alternatively, and the exact location of eyes can be determined by subtracting the alternating "bright-pupil" frame and the "dark-pupil" frame. This is computationally cheap and thus enabled the first non-intrusive real-time driver monitoring systems. However, the sophisticated hardware setup imposes limitation for use of these methods in real driving situations. Commercial system such as DD850 Driver Fatigue Monitoring developed by Attention Technology, Inc. also uses structured lighting system to achieve similar effect.

On the other hand, active NIR lighting is also employed in a lot of video-based monitoring systems for ensuring the image quality in low-light environment (e.g. night time or in simulator rooms). Several commercial remote eye-trackers that were employed in driving studies exploits this characteristic to ensure high quality video inputs. For example, the faceLab eye-tracker includes NIR LED's and two CCD cameras with NIR filters, which outputs crisp images with good contrast for tracking feature patterns around the face using stereo imaging techniques. While it works well for driving simulator experiments, its performance degrades for on-road studies as NIR spectra in natural daylight creates noise. Researchers had to avoid direct sun light when using faceLab eye-

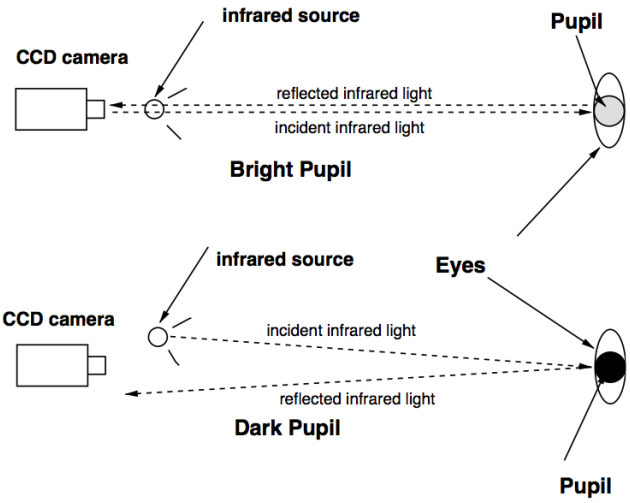


Figure 2.3: Illustration of the structured image acquisition system that features the bright/dark pupil effect from [11].

tracker for on-road experiments [28]. In an analysis regarding post-processing remote eye-tracker data from naturalistic driving studies [45], the authors reported problems such as reflections by glasses, strong sunshine and nighttime conditions can all reduce data quality. To mitigate tracking loss and error, they looked at quality and reliability of data, and applied quantile penalized regression for smoothing and interpolation. On the other hand, Cyganek et al. proposed an eye recognition system that uses color camera for daytime and NIR camera for nighttime. They have showed that the algorithm that were originally developed for color camera videos could be adapted easily to videos in NIR spectra [12]. Similarly, visible and IR-sensitive gray-scale camera was also applied jointly for video acquisition in a study estimating driver head pose [10].

Advancements of Face Alignment "in-the-Wild"

In the recent years, there has been significant development in the field of image understanding and computer vision, thanks to the improvements in algorithms, datasets and computational powers. The goal of modern face tracking or alignment algorithms also became much more ambitious, working towards achieving robust face alignment (or facial landmark localization) under completely "in-the-wild" conditions, meaning that no restrictions on the viewing angle, lighting condition, facial attributes and expressions are

placed. We briefly review this profound field to suggest possible applications of these advancements to the field of vision-based driver monitoring, which often involves accurately locating the driver's face and eyes from visuals acquired from a monocular, non-frontal color camera.

Face alignment methods could be categorized into three main approaches: Deformable Models, Discriminative Regressors, and Deep Learning. Although most methods strongly differ in terms of problem formulation and optimization process, they all attempted to resolve two fundamental questions: 1) how to model the local textures (e.g. "does this patch look like an eye corner or not"), 2) how to model the spatial constraints between located points.

The Deformable Model approach answers these two questions explicitly by building probabilistic models. For "holistic" models such as Active Appearance Model, a joint probabilistic model is created for both local texture and global shape constraints; on top of that, similarity transformations for mapping the target into a canonical space are often estimated to support generalization. It is computational expensive to fit the complicated model, which makes this method less suitable for real-time processing. On the other hand, Constrained Local Models (CLM) evaluates texture characteristics at the estimated fiducial points, and then use these responses to update the location of the fiducial points with a fitting strategy. A classic fitting strategy is the Regularized Landmark Mean-Shift (RLMS) [46], which many later works built and improved upon. Other fitting strategies such as "Consensus of Exemplars" [47] also showed promising performance. Another key component of CLM is building local texture models (also known as "patch experts"). For example, one such model could evaluate how likely the image patch is from an eye corner. Works focusing on this part proposed texture models like Local Neural Fields (LNF) [48] or Discriminative Response Map (DRMF) [49]. Due to the computational complexity of the texture models and fitting strategies, Deformable Model solutions usually works at a couple frames per second.

Comparing to Deformable Models, Discriminative Regressors are more implicit in modeling local textures and spatial constraints. This approach still involves iteratively updating the texture response and shape model, but instead of building patch experts,

features that could achieve the optimal shape update are selected from a large feature pool, such as various Local Binary Features (LBF) in [50]. This process exploits discriminative regression techniques (e.g. boosting) to alleviate the computation burden of direct optimization. The trained model could perform relatively fast during testing because the feature descriptor were usually simple and cheap to compute (e.g. pixel-differences). A major variation to this line of work was the Supervised Descent Method [51], which allowed using more complicated feature descriptor (e.g. scale-invariant feature transform [52]) and still achieve real-time performance speed. The core idea of this method was learning the descent direction from training examples when directly optimizing the objective function; this replaces the costly computation of Jacobian and Hessian and thus allowed great improvement in computation speed. All these above methods rely heavily on the initial guess of the landmark location because they essentially work on refining the locations.

Deep Artificial Neural Network (ANN) had been applied to a lot of computer vision problems and out-performed many state-of-the-arts methods. This was attributed to ANN's ability of automatically extracting features, which replaces the traditional hand-crafted feature descriptors. Different from the two above-mentioned face landmark detection approaches, this type of methods usually works in a coarse-to-fine manner: a higher-level representation of a certain patch size will be estimated at each layer, and as the network feeds forward, the size of the patch size refines, finally reaching a specific location [53]. The main benefit of this strategy is that it relies less on the initialization. Different network architectures were proposed for this task, including Auto-Encoder [53] and Tasks-Constrained Deep Convolutional Network (TCDCN) [54].

The above provided a brief overview for the vast landscape of facial landmark alignment methods. Detailed review papers such as [55] categorized and compared different approaches in a systematic way with details and technical analysis. Furthermore, there also exist several benchmarking datasets/competitions (e.g. 300 Faces in-the-Wild benchmark [56]) to enable direct comparison of methods' performance, and also act as good venue to present new advancements. Because of the largely improved robustness and accuracy, experimenting these methods on the problem of driver monitoring is desirable.

For example, an implementation of the LBF method [50] by the DLib library has been applied to a large-scale naturalistic driving dataset in [57].

Behavioral Understanding

After the machine obtained basic vision of where are driver's face/eyes, the next step is often estimating observable behaviors like head turning, gaze pursuing/fixation or blinking. These general results could be applied for fatigue and distraction detection. Therefore, like the previous section, we cover innovations developed towards various driving problems in this section, as well as related advancements in the general field of head pose or gaze estimation.

Murphy et al. designed a fully-automated head pose tracking system for inferring driver's attention that negates the cumbersome calibration process that is common in many eye-tracker systems [10]. It applied the Viola-Jones face detector [58] for rough facial region detection as the first step of their algorithm. As this object recognition algorithm is built based on analysis of pixel regions and does not have the flexibility to handle pose variations, their system needs to run three detectors for frontal face and two profile faces at the same time. Then, the successfully detected region will be scaled and normalized for computing the localized gradient orientation (LGO), which is used for training three support vector regressors for head pitch, yaw and roll. Then, the head pose can be estimated using this model given the LGO value of an input face region. Tracking was performed on with a module that estimates the 3-D motion of the head using particle filter for 3-D model tracking.

Fully automated initialization was achieved in the head pose and eye state estimation system proposed by Mbouna et al. [59]. They also relied on the Viola-Jones algorithm for face and eye region detection. Then, the head pose parameters were obtained via mapping the 2-D facial feature onto pre-rendered 3-D face model with certain Euler angles, and then solve for the rotation matrix using the POSIT algorithm. They also included a tracking module, but it was applied at the feature level using the optical flow method. In terms of evaluation and robustness, [10] obtained the ground truth of head movements using a marker-based motion capture system from 14 on-road drives, but [59]

only presented their experimental results of sub-modules (pupil detection and head pose estimation) using non-driving databases.

Gaze fixation is featured in a system for evaluation of driver distractions induced by IVIS [60]. Multiple types of distractions were considered and the experiment scenario included a variety of in-car activities like following GPS, using IVIS, and talking on phone. The main contribution of this study is a gaze estimation algorithm that requires no subject-specific calibration and could perform in a variety of simulated environments, including low lighting conditions. Instead of tracking specific gaze intersections or angles, their objective is classification of discrete fixation areas such as Front (road center), Left and Right Signals (road sides and nearby objects), Lateral Rear Mirrors and other IVIS devices (e.g. GPS). With the performance of their gaze fixation classification system verified against manually labeled ground truth, they proceed to propose using Percent Road Center (PRC) to measure distraction, which includes measures of time spent looking at road center as well as visual elements proximal to the road. While the time spent looking at Front is expected to decrease during when the secondary tasks is visually demanding, it can remain same or even increase when the distraction comes from mainly auditory or cognitive tasks. On the other hand, the time used looking at the Signals and Mirrors clearly reduced; but these contributes to very small percentage of the whole statistic. Though the proposed gaze fixation classification is online (30fps), they did not propose a real-time distraction classification method.

General Tools and Softwares

A lot of driving studies employ off-the-shelf tools for face and facial landmark tracking, and focuses on contributing to the extracting higher-level features and constructing predictive models. These tools could be commercial eye-trackers or even some other software packages, for example the Computer Expression Recognition Toolbox (CERT) [35]. These tools could also provide higher-level information like head pose or Facial AU estimation. It is noteworthy that algorithms in this field is developing very fast, today's computer vision is capable for fully-automated tracking of aligned face templates in a less constrained setup (e.g. the OpenFace [61] or IntraFace [62]).

However, when the studies employed existing off-the-shelf vision systems, it does pose question on how accurate the extracted features were. In many cases, the acquisition systems might not be so robust and some of the tasks such as AU estimation were quite challenging on its own. This imposed some uncertainty on the result and conclusion obtained from these studies.

2.3.2 Other Machine Learning Algorithms

Deep learning and artificial neural networks (ANN) are gaining enormous development in recent years because of improved algorithms, open resources, and dramatically increased computational power. The recurrent neural networks (RNN) with long short-term memory (LSTM) cells was explored for online driver manual/visual distraction detection, with low-level signals (driving and head tracking data) as well as statistical functional (e.g. means, peaks) as inputs [63]. Similarly, LSTM-RNN was employed for its strength in spatio-temporal reasoning in a driver activity anticipation system, which was trained explicitly to anticipate driving maneuvers a few seconds before they happen [64]. The optimized information sensory-fusion layer were highlighted, with the prior knowledge that expressive architecture to combine sensory inputs outperforms simple concatenation. Both work compared and showed that LSTM-RNN based machine learning model outperformed SVM. However, complex neural networks would require huge amount of data for training, otherwise overfitting would be a major concern.

2.4 Chapter Summary

This chapter surveyed three aspects of studies contributing to monitoring driver cognitive load, corresponding to the background knowledge, state-of-the-arts and potential future developments of this field.

First, visual attention measurements are found to be consistently impacted by driver cognitive loads. However, in these statistical analysis, the measurements were usually observed over a longer period of time. Also, the statistical models were constructed for analysis rather than estimation. Therefore, although the findings suggest direction for

monitoring driver cognitive load, the measurements and modeling methodologies might not be suitable for an practical estimation system.

Section 2.2 reviewed previously proposed solutions on estimating driver cognitive load with video-based inputs. This section started by a comparison of the problem definition and data collection setups employed in these studies. Then, two main components, feature extraction and estimation model training, are surveyed and discussed extensively. Features characterizing visual attention were found to be more popular than facial expression units, but the specific feature descriptor for a similar symptom (e.g. gaze concentration) could differ. Machine learning was adopted for training estimation models, and a variety of algorithms have been explored. These studies often the effectiveness of different feature groups, modeling algorithms and hyper-parameter settings based on empirical evaluation results.

Finally, a brief overview of related advancements in the field of computer vision and machine learning is provided in Section 2.3. The purpose of this is to explore potential development directions based on the most recent achievements could be applied to substitute components in a video-based monitoring system.

Chapter 3

Data Collection for Studying Driver Cognitive Load

A necessary step in research is gathering representative data samples for the targeted problem, which is particularly challenging for studies focusing on high cognitive load on drivers. Previous datasets used in studies of similar problems were usually collected for testing specific research hypothesis (e.g. [16]), and thus were seldom sharable or suitable for other studies. As part of an NSERC project titled "Enhancing Driver Interaction with Digital Media through Cognitive Monitoring" (abbreviate as "eDREAM"), we collected a dataset to enable research on applying advanced sensor and vision technologies to assess cognitive loads of drivers. A comprehensive set of sensory signals were collected, including participant-facing videos and remote eye-tracking results. Therefore, this dataset will be suitable for evaluating the proposed video-based driver monitoring system targeting high cognitive load problems in this study.

This chapter explains the experiment logistic, describes the procedures, and provides technical details of the resulted data. As an overview, the EDD collected various sensory and visual signals when participants experienced three different levels of cognitive load during driving. Each load level was presented in a separate drive: the lowest level was simply driving with no added secondary task, while the medium or high levels were imposed using modified n-back tasks with load factors of 1 or 2, respectively. Since the experiments were conducted in a simulator environment, control of most external

conditions could be achieved with preprogrammed scenarios. Furthermore, other design considerations such as workload from the driving task, and subjects' individual differences were also accommodated in attempt to isolate responses induced by increased cognitive load. In addition to the visual signals that will be focused in this study, EDD also collected vehicle-based measurements, subjective reports and various physiological signals including EEG, ECG, GSR and Respiration (RESP). These sensors will be introduced briefly in this chapter for they affected the experiment design.

Author of this thesis is one of the two main contributors completing the EDD collection [1], who both contributed equally in terms of experimental design, implementation, data collection as well as organization. Along with advices from experts [2] for problems and challenges encountered during experimental design, the final design presented in this chapter is a collective result after extensive pilot testing, literature research and experimentations.

3.1 Data Collection Methodology

This section describes the process for collecting EDD. Following conventions of describing driving experiments, it starts with basic experiment elements such as participant description and data collection devices. Then, the driving scenario and secondary task will be introduced. These required more efforts and considerations to design, especially in this experiment as it focuses on driver's cognitive states; logistics, assumptions and implementation details will be explained as well. Finally, this section will be concluded by a description of the whole experiment procedure.

A conceptual overview of the eDREAM experiment is given in Figure 3.1. This diagram points out variables and factors that could be presented during the actual data collection process, and thus needed to be considered in designing and implementing

[1] The other contributor of EDD is Dengbo He, currently a PhD candidate at the Human Factors & Applied Statistics (HFast) Lab.

[2] Professor Birsen Donmez, Doctor Winnie Chen from the HFast Lab and Doctor Amirhossein Shokouh Aghaei from the Multimedia Lab were consulted during the design and collection of EDD.

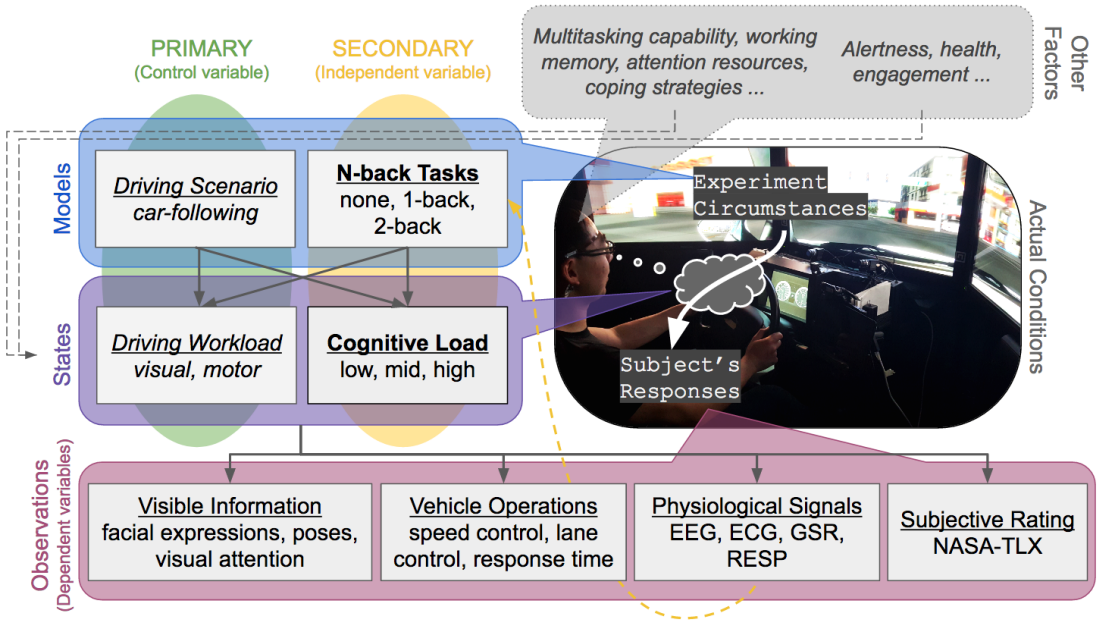


Figure 3.1: Conceptual overview of the eDREAM data collection experiment. The block with a photo of the experiment represents the actual conditions presented in reality, while blocks around it describe how they were abstracted in the design and analysis space. The bold font emphasis the elements related to the targeted research problem, while italic font marks uninterested but related factors also presented during the experiments. The causal elements are linked with gray arrows. Note that the primary driving task and secondary n-back tasks affect all types of states concurrently, and the observations measured the overall participant responses (instead of the cognitive load alone). This indicates the importance of controlling other factors such that the effect of the independent variable could be isolated.

the experiment. To further explain, the targeted cognitive load states were modeled using the n-back tasks, which will be introduced in subsection 3.1.3. In the context of driving, the added cognitive load must be presented concurrently with the primary driving task and the participants' responses could be influenced by both tasks. Therefore, ideally, the driving task should be controlled such that the impact of independent variable (i.e. cognitive load) on dependent variables (i.e. measurement of subject's responses via sensors or cameras) could be isolated. Implementation of these experiment circumstances will be explained further in subsection 3.1.4. There also exist other factors such as the individual differences in abilities and functioning states that could contribute to the participant's states. These variations were minimized by means of selecting a specific participant group and including extensive preparing procedures (e.g. trainings); details

will be noted in corresponding subsections.

3.1.1 Participants

A total of 37 healthy, gender-balanced participants were recruited through campus and online posts to participate in this driving simulator study. They all satisfied the following three requirements, which were placed to minimize participants' individual variation and ensure the usability of data collection instrument.

- Drive at least several times per month and hold a full driver's license (G license or equivalent) for at least 3 years;
- Must be under 35 years old;
- Drive without glasses (contact lenses are allowed).

The first requirement ensured the participants are practiced drivers, who should have no problem adapting to the driving simulator and also handling the secondary tasks while driving. The second requirement controlled the age group to minimize the variation in available mental resources, such as working memory. Previous driving studies have found age to be an significant factor for cognitive task performance, and the youngest age group showed best performance on a secondary task very similar to the one applied in this experiment [65]. The third requirement was to ensure the function of the eye-tracker system.

Participants were compensated at CAD$12 per hour for a 3-hours experiment session, and were told that they could receive a bonus of up to CAD$14 bonus amount based on their task performance as an incentive for engaging in the secondary task. They were informed that the All participants were paid for three hours and full bonus amount (i.e. CAD$50) regardless of their actual time spent or task performance.

3.1.2 Apparatus

This section describes the driving simulator, the devices used for collecting visual or physiological measures from participants, and the subjective questionnaires.

Driving Simulator

The study was conducted on a NADS miniSim™ driving simulator (Figure 3.2). This fixed-based simulator has three 42-inch screens, creating a 130° horizontal and 24° vertical field at a 48-inch viewing distance. The center screen displays the left and center parts of the windshield; the right screen displays the rest of the windshield, the rear-view mirror, and the right-side window and mirror; while the left screen displays the left-side window and mirror (see also in Figure 3.3).

Figure 3.2: The NADS miniSim Driving Simulator displays mid-fidelity rendering of the simulated driving environment on three screens, and the speedometer on another smaller display behind the steering wheel. Other light sources are blocked or turned off to create a more immersing driving environment.

miniSim records a comprehensive set of driving data at 60 Hz, from external measurements (e.g. vehicle speed, lane deviation, car following distance) to vehicle operations (e.g. steering wheel angle, brake pedal force). In addition to the ease of acquiring accurate vehicle-based measures, another huge advantage of using a driving simulator is the freedom of designing specific traffic and task scenarios. All environmental events and cognitive task-loads were integrated into simulator "scenarios" that could be repeated for each participants, guaranteeing the consistency between experiment sessions. This will be elaborated in later sections on scenario design.

Eye-tracker

Eye-tracking information was obtained using faceLAB 5.0, a remote eye-tracker system developed by Seeing Machines.

- Hardware Setup: faceLAB uses an active Near-Infrared (NIR) LED and two NIR cameras, which could acquire high-quality imageries that would be invariant to change of illuminations. The pair of cameras are mounted at the center above dashboard, obtaining an ideal frontal view of the driver (see Figure 3.2); if not so, the eye-tracker might not provide as accurate results.

- Core Mechanisms: Although the exact details are not provided (as faceLAB is a commercial product), from the official user manual [**?**], the following four aspects were found to be the most important technical mechanisms that facilitated its good eye-tracking capability.

Figure 3.3: Camera and eye-tracker placements in the driving simulator. The webcams (left and upper) are framed in red, while GoPro camera (right) is framed in green. Eye-tracker cameras (centre) are also marked (in yellow). The grey area at lower middle of the image represents the approximate location of participant's head.

- Glint tracking: In order to obtain precise estimation of gaze direction (i.e. using "Precision" mode of faceLAB), the IR source must be placed at the center of two cameras, and this whole system should be located straight in front of the subject. This setup should create a "glint" within each of the subject's iris (shown in Figure 3.5), which are tracked for accurately estimating eye ball rotation. Since this NIR setup could be interfered heavily by natural NIR from sunlight, faceLAB is ideally used indoor; when used outdoor, strong sunlight must be avoided (as in [28]). Upon changing location or orientation of the cameras, its camera model would need to be re-calibrated using a checkerboard target. This process is usually referred as camera calibration.

- Facial feature points: In addition to the iris glints, faceLAB also relies on tracking various facial feature points (such as eye and mouth corners). These are used for creating customized 3D head models, which were calibrated for each participants. This allows faceLAB to estimate head pose, which contributes to the gaze direction estimation.

- Predefined virtual "world": faceLAB allows users to define objects or planes (region of interests) in its "world coordinate", which are used for calculating gaze intersections. The virtual world setup is illustrated in Figure 3.4, with planes representing the TV screens and dashboard screen of the miniSim; all of these were placed according to actual measurement. The midpoint between two cameras is defined to be the origin of the world coordinate in faceLAB. Thus, they should always be placed at a consistent location to ensure the predefined faceLAB world model would remain correct and accurate.

- Dot-tracing calibration: Finally, the gaze intersection on the main plane (noted with a yellow rectangle in Figure 3.4) could be further calibrated using a dot-chasing process. During this process, subject would be asked to follow a dot that stops at center, corners and edges of the main plane. With this prior, this final calibration step could help to enhance and stabilize the gaze tracking performance. Furthermore,the tracked results would be displayed once the

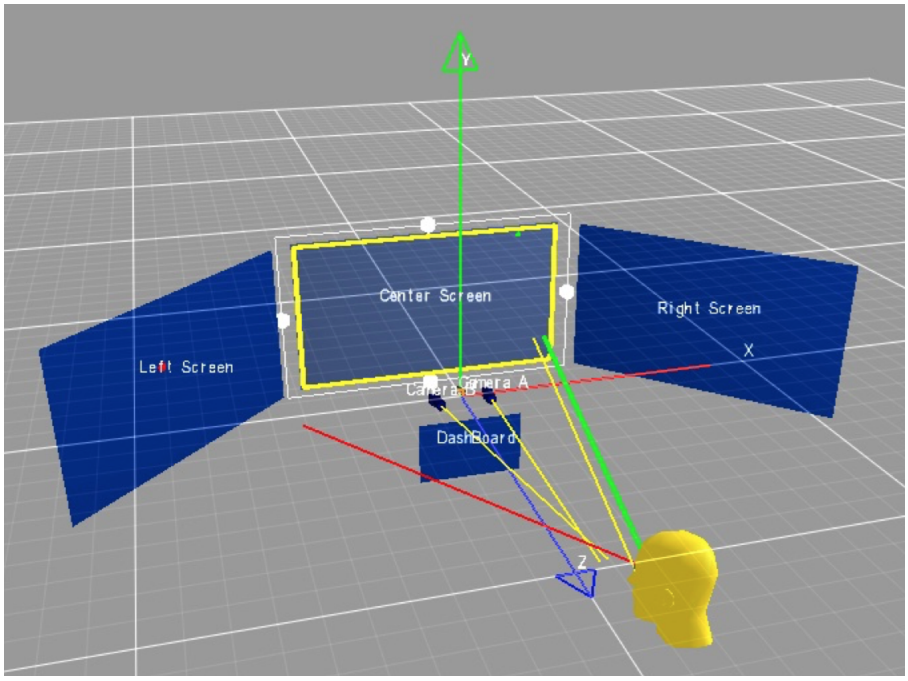


Figure 3.4: The virtual faceLAB world used in eDREAM dataset collection. We defined the three screens and the dashboard of the miniSim simulator. Origin of the world coordinate is the midpoint between two faceLAB cameras. The X, Y and Z axis extended out from the origin, denoted in red, green and blue arrows. The main plane on which the gaze intersection could be precisely calibrated is framed with yellow rectangle, which is essentially the center screen of the three miniSim screens in our setup. The yellow head model displays the current tracking, with the green ray representing the gaze direction and red ray representing the head rotation.

process is finished. Since the gaze intersections should be scattering around the stop locations, this could be used as a way to check the performance of the eye-tracker on individual participant.

- Signal Collection: In addition to all the abovementioned estimation results, faceLAB also computes several higher-level features (e.g. blinks or saccade detection, see detailed output description in section 3.2.2). All of these outputs are computed on the fly (rather than based on recorded videos), at a very high frequency of 60 Hz. When the system is running, the estimated results (e.g. gaze direction, eye-lid position) are overlaid on the visuals acquired from the cameras, allowing users to inspect the overall performance; an example screenshot is provided in Figure 3.5. However, possibly due to computational or storage limitations, these visuals could

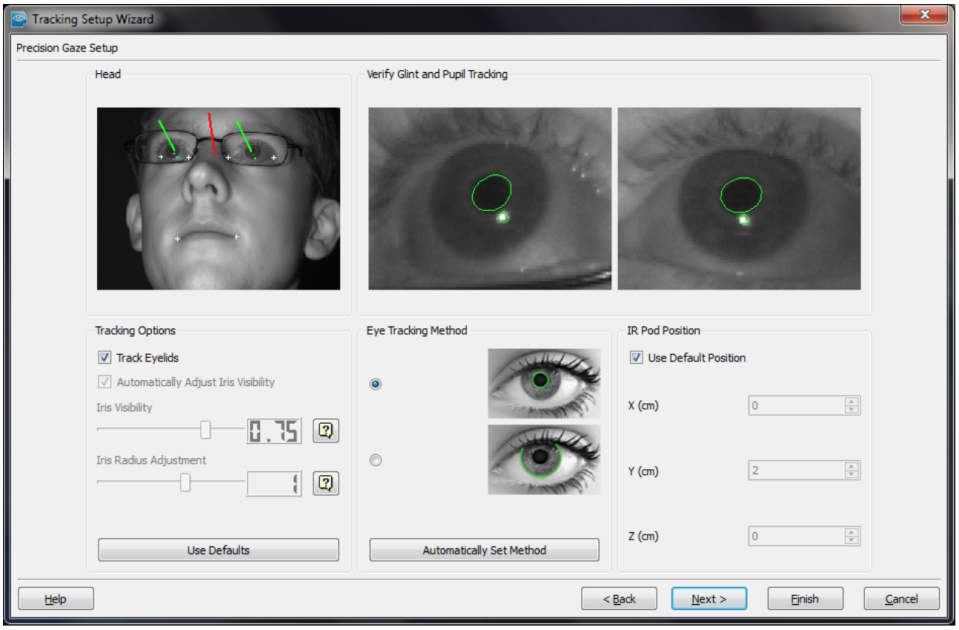


Figure 3.5: Example faceLab screen shot, displaying the visuals from the two cameras with overlaid tracking results. The head view shows the gaze direction in green and head direction in red. On the close up views of the eyes, the tracked pupils and glints are circled in green.

not be saved into files. Instead, the estimated values are logged in a binary format, which could be exported into text tables after the recordings are completed. These are recorded locally in the host computer of the software, while part of them are also forwarded to the miniSim computer and logged into miniSim recordings.

Color Cameras

A GoPro Hero4 camera is placed at front-right of the driver on a tripod. Also, there are two HD Logitech webcams (model numbers C920 and C930) fixed on the driving simulator, facing the driver from front-left and upper-front angles (see Figure 3.3 for an illustration). The placement of this set of cameras represents the possible camera positions in a real car, which would be most convenient to set up and interfere the least with the driver's sight field, but also provides a non-obstructed view of the driver face.

The GoPro camera records videos into a stand-alone SD card, while the Logitech webcams streams into D-Lab and were recorded along other physiological signals. Due to technical problems, signals from the Logitech webcams were often lost during the ex-

periments, possibly caused by bandwidth limitations of the D-Lab computer. Therefore, GoPro recordings should be considered as the primary video source.

Physiological Sensors

There are two systems employed to record various physiological signals. First, EEG data was collected using the Muse headband developed by Interaxon, a wireless non-intrusive headband consisting of 2 dry sensors located at Fp1 and Fp2 positions and two gel foam electrodes at TP9 and TP10 positions. The EEG headband was worn around the forehead (Fp1 and Fp2) with two electrodes attached behind the ears (TP9 and TP10). The associated software, MuseLab, was used to record EEG signals and estimate band powers.

On the other hand, ECG, GSR, and respiration signals were recorded by the D-Lab software developed by Ergoneers. These were collected using the traditional intrusive physiological sensors: solid gel foam electrodes were used for ECG and GSR to attach sensors directly onto participants' bodies.

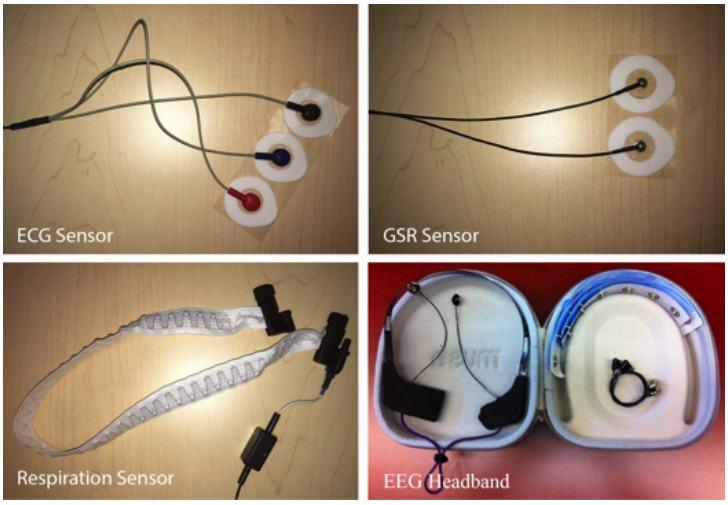


Figure 3.6: Physiological sensors used in collecting EDD.

Subjective Questionnaires

In order to collect participants' perceived cognitive workload level, they were asked to complete three subjective questionnaires after each drive that contained different task-

load: NASA-TLX [66], risk perception and Rating Scale Mental Effort (RSME).

NASA-TLX has been commonly applied to measure the perceived workload of operators, pilots, and drivers. It is a multi-dimensional scale that considers five aspects of the overall workload, which are all rated individually within a 100-points range (with 5-points steps). They are described to the participants as following:

- Mental Demand: How much mental and perceptual activity was required? Was the task easy or demanding, simple or complex?
- Physical Demand: How much physical activity was required? Was the task easy or demanding, slack or strenuous?
- Temporal Demand: How much time pressure did you feel due to the pace at which the tasks or task elements occurred? Was the pace slow or rapid?
- Overall Performance: How successful were you in performing the task? How satisfied were you with your performance?
- Frustration Level: How irritated, stressed, and annoyed versus content, relaxed, and complacent did you feel during the task?
- Effort: How hard did you have to work (mentally and physically) to accomplish your level of performance?

After the raw ratings are obtained, the second part of NASA-TLX asks participants to do pair-wise comparison between these sub-scales in terms of which one was more demanding. The number of times chosen would become the weight for each sub-scale when computing the final weighted NASA-TLX score.

The other two questionnaires only consist of one question in each. Risk perception asks the participant to rate how risky he or she found the previous drive was with 10 discrete levels. RSME asks the participants to indicate how much effort it took for them to complete the task on a continuous scale from 0 to 150.

3.1.3 Secondary Task for Controlling Cognitive Load

The EDD features three levels of cognitive load modeled by a modified n-back task, which is a auditory recall task with letter stimuli. The lowest cognitive load level was collected when participants were driving with no n-back task presented, while the medium and high cognitive load was collected from driving with 1-back tasks and 2-back tasks, respectively. These two tasks were described to participants as following:

- 1-back: participants were asked to count the number of times two identical letters appeared in pairs (e.g., "A, A"), and answer the count at the end;
- 2-back: participants were asked to count the number of times two identical letters appeared in pairs with one letter in between (e.g., "A, B, A"), and answer the count at the end.

The collection of EDD follows a paradigm that considers the presence and load-factor of the n-back task determines the cognitive load of the participants. This subsection explains what n-back tasks are, why this kind of task was chosen and how it was adapted for accommodating the driving task and collection of physiological signals.

The n-back Task

In general, "n-back" refers to a family of continuous performance tasks that tests the subject's working memory. The core idea is that when presented with a sequence of stimuli (e.g. numbers, letters, picture of objects) drawn randomly from a finite pool, subjects should try to identify if the current stimulus matches the one presented n steps ago.

For example, an audio 2-back task with letter stimuli could be an recording of someone reading the following sequence of letters at a constant pace:

C B H H C A ***C*** B F ***B***

And the subject is supposed to acknowledge that a 2-back pattern occurred when the letters marked in bold were read.

Since first introduced by Kirchner as a visuo-spatial task with four load factors (from 0-back to 3-back), this paradigm has been adapted and applied widely in the field of neuroscience [67]. The performance of n-back was used as a measurement of working memory[68], which is directly related to cognitive load. More recently, n-back was adapted in several driving studies as a mean of systematically increment the cognitive load, and was introduced as "an evolving international procedure for grading cognitive workload" [69].

The n-back tasks involves multiple cognitive processes that could be thought as abstraction of everyday tasks, such as perceiving and encoding the incoming stimuli, maintaining and updating working memory, and analyzing the materials (e.g. deciding if the current stimulus matches the one presented n steps ago, but not other ones). When the value of load factor n increases, participants would need to maintain more items, which makes the task more cognitively demanding. In this context, n-back task was considered to be a systematical way to impose higher cognitive task-load. The performance (i.e. correct rate) of this task has been used to support that higher load factor n would effectively increase the difficulty [17, 65].

Implementation and Adjustments

The modified n-back task used for collecting EDD followed similar methodology of the previous n-back task used for driving studies [69]. In each task, participants listened to a prerecorded series of 10 letters randomly drawn from a finite pool, and count the appearance of the n-back pattern. Audio recordings that each contained a group of three n-back tasks of the same load level were prepared beforehand (see illustration in Figure 3.7). At the very beginning of each recording, a brief introduction was provided to notify the participants which type of n-back it is; also, at the end of each task group, another notification was provided to let the participant know that the task had ended. These recordings were presented in the Critical Periods of driving scenarios, during which the driving environment were kept to be consistent across different experiment trials (see subsection 3.1.4).

Instead of visual stimuli, the audio format was chosen to account for cognitive de-

mands induced by the modern IVISs that minimizes visual or manual distractions [69]. This is also the case where driver inattention would be more challenging to detect, since it is not associated with obvious behavioral cues.

In order to minimize the facial muscle movements that interfere with the EEG signals, the n-back task used for EDD only requires participants to answer the total count of n-back pattern occurrences at the end of each series. This is different from the "delayed digit recall (n-back) task" described in [69], where participants continuously recall and answer the stimulus presented n steps ago, after every new stimulus. This modification might result in higher cognitive demands since participants also needed to keep track of the count, which is another item that needs to be cognitively maintained. To partially accommodate this added difficulty, letters instead of numbers were used as the stimuli to minimize the interference of remembering the stimuli and counting (e.g. avoid the confusion of the count "1" and the stimulus "1" that appeared).

Training of the n-back tasks would be given in a non-driving condition first. After an introductory explanation, example tasks that ensures participants understanding and capability of this task were always completed. Furthermore, participants also went through a "warm-up" drive during which they complete one group of 1-back and 2-back; this provided more familiarization and helped them to stabilize their coping strategies of multi-tasking during driving. This was added after an obvious suppression of vehicle speed was observed in earlier drives of pilot participants. More details of this drive will be provided near the end of Subsection 3.1.4.

Participant Engagement

Since participants in EDD only answered a count at the end of each modified n-back task, it was impossible to accurately evaluate their correctness at detecting each single n-back pattern. Participant's engagement could strongly influence the effectiveness of the n-back paradigm. There were four means to ensure a systematical increase of cognitive load was achieved on participants:

1. **Participant selection:** With proficient driving skill and better cognitive ability

of EDD's participant group, it is reasonable to expect them to divert cognitive resource to attend these secondary tasks (see 3.1.1).

2. **Cash incentive:** To encourage participants' engagement, they were told that a cash bonus ($14 CAD) would be rewarded based on their performance (see 3.1.1).

3. **Proper training:** Participants receive extensive training the tasks before the three formal drives should prepare the participants sufficiently, avoiding the cases of abandoning due to inability (see 3.1.4).

4. **Consistent presentation circumstances:** The presentation circumstances will be controlled by preprogrammed driving scenarios, which avoided drastically differences from the driving demand (see 3.1.4).

As a post-hoc validation, the effectiveness of the n-back model could be reflected by self-reports (i.e. NASA-TLX) (see 3.3.1).

3.1.4 Scenario Design

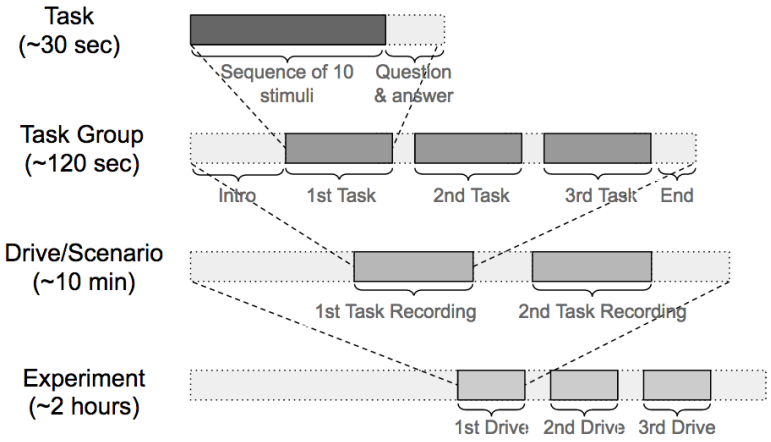


Figure 3.7: Overview of experiment scenario arrangements. Note that the Critical Period is where secondary tasks recordings could be presented, thus it corresponds to the "Task Group" or "Task Recordings" in this illustration.

The driving course, traffic conditions, and timings for presenting n-back tasks were all controlled by the preprogrammed miniSim scenarios. The same set of scenarios were

used across all participants to achieve consistency, but the ordering of three scenarios for each participants were counterbalanced in order to mitigate effects such as changes in proficiency level.

As an overview, in each driving scenario, the participant followed a Lead Vehicle (LV) at 40 Mph on a 4-lane urban route, which would not change lanes or make turns at intersections, but was programmed to brake abruptly at specific moments. In order to resemble a driving situation realistically, participants were instructed that the primary task was to drive the External Vehicle (EV) normally, with all the regular safety measures. The LV was introduced to participants as a friend's car that is leading them to an unfamiliar destination; thus, they were suggested to follow it with a reasonable gap.

Design Considerations: Control vs. Complexity

A key point of designing the driving scenarios was to minimize the variations introduced from factors other than cognitive load, such that data for the targeted research objective could be isolated. As shown in the illustrated overview of the eDREAM data collection process (Figure 3.1), this corresponds to achieving the control of the workload from the driving task, which should be a control variable of this experiment. Thus ideally, the driving scenario should be consistent, but also not overly demanding that participants could not spare attention to the added cognitive load. This was the reason behind using a car-following scenario, as participants did not need to worry about finding routes, changing lanes, or interacting with other vehicles. Therefore, it helped satisfying the objective of controlling the workload from the driving part and minimizing the differences between individual driving styles. In addition, the LV was designed to brake abruptly in order to collect the subjects' reaction time under different cognitive load levels.

Originally, the experiment used a single 40-minute scenario with different levels of cognitive loads induced during several segments of the route. However, pilot testing reflected that there was no need to pay attention to the overall driving situation, since this design eliminates possibilities of cars on the other lanes or at the intersections (see the "Deprecated" scenarios in Table 3.1). Participants only needed to maintain the speed and lane position of the EV, which was quite an unnatural driving situation and

could easily make participants drowsy. Furthermore, this also made it unlikely to obtain data of the targeted high cognitive load states, as this driving condition leaves plenty of mental resources available for secondary tasks. Therefore, creating a more realistic driving scenarios that impose reasonable workload themselves becomes another major design objective.

To address the problems encountered above, new scenarios were made with higher complexity. In the final implementation, each participant completed three 10-minute scenarios on a 4-lane urban route with multiple intersections. During the Critical Periods where the recordings of n-back tasks could be played (see Figure 3.7), variations introduced from factors other than cognitive load was minimized. There were two Critical Periods in each drive, residing in the later two straight segments of the route (as shown in Figure 3.8); the curved section in between acted as a break that resets participants cognitive states to avoid fatigue or stress (following the implementation in previous driving studies [69]). Outside of the Critical Periods, there could be higher variations, such as road curvatures, different LV behavior, and more interesting situations at the intersections (e.g. left-turning vehicles). Details of the final implementation will be introduced in the next subsections.

Table 3.1: Comparison of Driving Scenarios

		Road Conditions		Ambient Traffics		LV Behavior	
Scenario	Critical Periods	Curves	Red Lights	Crossing/ Merging	Traveling Beside	Intense Braking	Slow
Deprecated	Outside	●	⌀	⌀	⌀	○	⌀
	Inside	●	⌀	⌀	⌀	●	⌀
Training	-	●	○	●	●	●	⌀
Warm-up	-	●	●	●	●	●	●
Formal	Far away	●	●	●	●	●	●
	Nearby	●	○	○	○	○	○
	Inside	○	○	○	○	●	○

Notes about symbols: ● means the condition was actively presented; ○ means the condition was controlled or not presented; ⌀ means the condition was virtually not possible to happen (as perceived by participants).

The Driving Environment

A miniSim map with an urban 4-lane road was customized for this experiment (see Figure 3.8). It consists of three straight segments separated by two curved ones in between. Each straight segment lasted approximately 2 to 3 minutes (depending on the actual speed at which the participant was driving), while the curved segments were approximately 45 seconds long. This map was designed to allow the placement of two Critical Periods on the later two straight segments of the road. The curves introduced higher demand in vehicle control, which was necessary to avoid tiresome and boredom from an driving route with only straight roads. Static elements like buildings, billboards and cars parked on the side lane created some point of interests that might attract participants to observe, and made the driving scenario look more interesting.

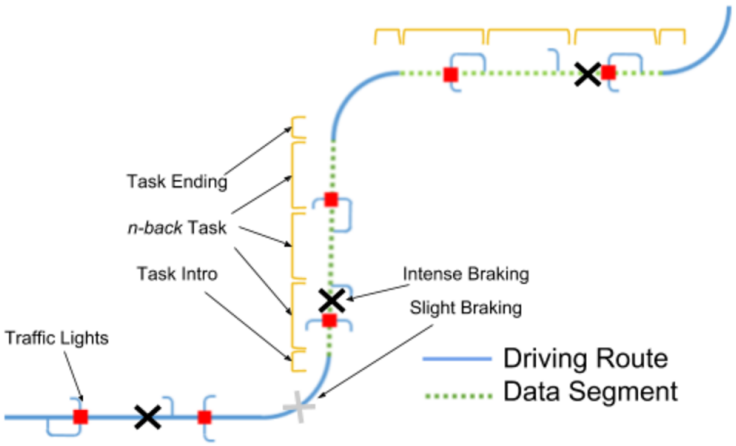


Figure 3.8: Illustration of a formal experiment scenario (map is not to scale). The driving route contained straight and curved segments. The green dotted portions of the map identified the data segments, within which all conditions are controlled such that changes of cognitive task-load could be isolated. The secondary task's audios (marked by yellow brackets on the side) lasted slightly longer than the data segment. Example placement of braking events were also marked in black crosses, note that this is changed in different drives. Traffic lights were always green during data segments.

By design, participants would not be interfered by other objects in the scenarios except the LV and some traffic lights. Nonetheless, light ambient traffics (e.g. vehicles driving in other lanes or waiting at intersections) and pedestrians were still added. These not only makes the scenarios more realistic, but also encouraged participants to pay

more attention to observe the peripheral environment, or they might completely negate common driving tasks such as mirror checking. There were some variations in placements of the vehicles and objects across the three scenarios to make them more realistic. But during the Critical Periods, these elements were all controlled to impose a similar, low demand on participants. This was achieved by only placing vehicles driving on the opposite direction, and ensuring the traffic lights to always be green within all Critical Periods. The arrangement of traffic elements is also demonstrated in Table 3.1).

The Lead Vehicle's Behavior

The LV was programmed to brake intensively at particular moments of the scenarios, and the participants' reaction times were collected test participants' responsiveness to hazardous events under different cognitive load levels. Specifically, the LV were programmed to brake once during each Critical Period: either during the first task or the third task (always different within the same drive). Thus there were a total of 2 braking events for each cognitive load level. Note that this also reduces the amount of data samples where the influence cognitive load could be isolated, since the segments that contained braking events should be removed from this type of analysis. There were also several braking events deliberately placed randomly outside of the Critical Periods, such that participants were encouraged to pay attention to the LV throughout the drive.

To minimize the variation in braking events' demanding level, the LV was programmed to always automatically adjust itself to create a 2-sec time gap between the EV. This means that within a predetermined range (i.e. 30 Mph to 50 Mph), the LV would decelerate and try to shrink the gap if the participant slowed down, and vise versa. The actual time gaps achieved at the onset of LV's braking events could vary due to vehicle dynamics.

Preparation Drives

In addition to the three data-collection (or "formal") drives, participants needed to complete two preparation drives beforehand to ensure proper familiarization.

Placed at the beginning of the whole experiment, the "training" drive was the first

time participants were introduced to the driving simulator and the primary driving task of this experiment (i.e. following the LV at 40 Mph through an urban area). As a beginning, it was designed to be very simple such that participants could get accustomed to the rendered graphics and disappearance of the force of deceleration/acceleration. This was especially useful for people who were prone to motion sickness. Also during the training drive, guided by the investigators' explanations, participants experienced the LV mechanism of always trying to keep a "constant" time gap with them, therefore when their speed decreases the distance gap would seem to decrease, and vice versa. This helps to alleviate the fear of keeping a 2-second gap with the LV, which was closer than some people's preference and might be unsafe in a real-world driving situations. Participants were suggested to keep around the 40 Mph speed limit, under which condition the gap would feel more comfortable. In addition, the LV braking events would be introduced in the later half of the training drive. There were multiple braking events for them to practice the operation of the EV in miniSim. They were free to ask any questions during this drive; although investigators would not answer with exact implementation detail, they could guide them to keep a reasonable speed and respond for braking events, which usually alleviate the strangeness of driving on simulator. The training drive could last for as long as 15 minutes, but it could be ended earlier if the participants felt comfortable with all the driving conditions.

A "warm-up" drive would be completed after the training for n-back tasks and setup of all sensors (see Table 3.2 for the experiment procedure). Before introducing this drive, the pilot participants were observed to be more conservative during the first drive and maintains a lower average speed (no matter what cognitive load level it was). This was because even though they received proper training for the n-back task alone, multitasking it during driving requires more practice and familiarization such that they stabilized the coping strategy. The warm-up scenario contained a group of three 1-back tasks and a group of three 2-back tasks, also appeared during the Critical Periods as in the formal drives. This provided the needed practice and helped participants to feel less nervous or stressed during the actual Critical Periods (especially the first drive). On the other hand, there have been observations from the pilot testings that subjects might become

more attentive to the driving task during the n-back tasks (as compared to other no-task segments) as they anticipated abrupt braking events of the LV. To alleviate this difference, the warm-up drive contained more LV braking behaviors and diverse traffic situations to encourage participants to always pay attention to the driving environment (see Table 3.1 for comparisons). More specifically, frequent braking events were presented in a group of n-back tasks to minimize participants' anticipation of the systematic nature of braking events that were going to happen in the formal drives (i.e. once in every task group, or once in every straight segments of the road). To make it most effective, the warm-up drive was presented as the first out of four formal drives participants needed to complete. Finally, participants were also introduced to the NASA-TLX questionnaire, Risk Perception questionnaire and Rate Scale Mental Effort questionnaire at the end of the warm-up drive.

3.1.5 Experiment Flow

As last subsection about data collection methodology, the experiment flow is presented here to tie-in all the elements mentioned in the above subsections. The experiment steps and time allocations are summarized in Table 3.2.

Prior to the experiment, investigators must arrive approximately 20 minutes earlier to ensure everything were prepared accordingly: all computers and softwares must be started, sensors and cameras must be taken out from storage and placed at designated locations. Connectivity, battery level and relay program should also be checked. If scenario files that matches the current counterbalanced order were not produced, then new ones should be made by modifying previous files. Consent forms, cash and receipt were also prepared at this time.

Upon participant's arrival, the first thing was to verify eligibility and obtain consents. Then, the participant went through a training drive to get used to the driving on the simulator. They also practiced the primary driving task, as they following the lead vehicle at a 2-second time gap and experienced lead vehicle braking as it would happen in the later drives. If the participant did not experience severe motion sickness and wish to continue, they were then given written and oral instructions on the modified

n-back task and practiced it without driving. This ensured that they fully understood and were capable of doing the n-back task. Physiological sensors were then placed on the participants and the eye tracker was calibrated.

Next, participants were told they would complete four formal drives, while only the later three drives were formal data collection drives that contained Critical Periods. The first drive was designed to be an additional "warm-up" drive, where they could familiarize performing the n-back tasks while driving. The participants were told that this was an formal drive in order to minimize their anticipation of where and when lead vehicle braking events were to occur in the experimental drives, since the placements are drastically different between formal and warm-up drives.

Finally, participants went on to complete the three formal driving scenarios. The order of presenting cognitive load levels followed predetermined counterbalanced orders. Subjective questionnaires were were given during the 5-minute break after each drive. At the end of the experiment, participants were debriefed and received their payment.

3.2 Data Description

With the methodology described in the previous section, EDD have recorded various types of signals from a total of 37 participants. This section provides more details on the resulted dataset, and explains how it will be used to investigate research questions of the current study.

3.2.1 Data Labeling and Focus Periods

There are three levels of cognitive task load introduced in EDD: no-task (low), 1-back (medium), and 2-back (high). Amongst all the collected data, the most interesting periods are when the effect of cognitive load could be isolated. Several experimental conditions should be considered when finding these *focus periods*. These conditions are recorded as part of the driving simulator's log and later extracted as into "meta-data"

Table 3.2: Data Collection Experiment Steps

Step and Description	Time
Prepare the data collection system: Start all data recording systems, assemble physiological sensors, battery check	10:00
Arrange other necessities: Ensure scenarios, forms and cash were ready	10:00
Meeting the participant: Greeting, introducing the experiment, and signing consent form	10:00
Training drive: Introduce the driving simulator, driving conditions, and braking events	15:00
n-back training: Introduce the secondary cognitive task and allow practise	20:00
Break: Allow participants to use washroom before attaching sensors	5:00
Sensors setup: Calibrate the faceLAB eye-tracker and attach EEG, ECG, GSR, and RESP sensors	30:00
Warm-up drive: With a group of 1-back and a group of 2-back	10:00
Break: Complete the subjective questionnaires, refresh	5:00
First formal drive: Task condition following a predetermined counterbalanced order	10:00
Break: Complete the subjective questionnaires, refresh	5:00
Second formal drive: Task condition following a predetermined counterbalanced order	10:00
Break: Complete the subjective questionnaires, refresh	5:00
Third formal drive: Task condition following a predetermined counterbalanced order	10:00
Break: Complete the subjective questionnaires, refresh	5:00
Debriefing: Remove sensors, concluding remarks, payment and collect receipt	10:00
Contingency: Extra time for unexpected problems (e.g. sensor calibration) or delays (e.g. longer break time)	30:00
Cleaning up: Dissemble physiological sensors, turn-off all data recording systems, battery charging	10:00
Total time	**3:30:00**

files to facilitate easier query. An example meta-data recording is plotted in Figure 3.9.

By experimental design, two 1.5-minute Critical Periods were allocated in each drive, which are time slots for placing n-back recordings; road curves or other external incidents would not be presented during these times and thus impact of cognitive load could be isolated. In Figure 3.9, this corresponds to the dotted-blue signal. However, these Critical

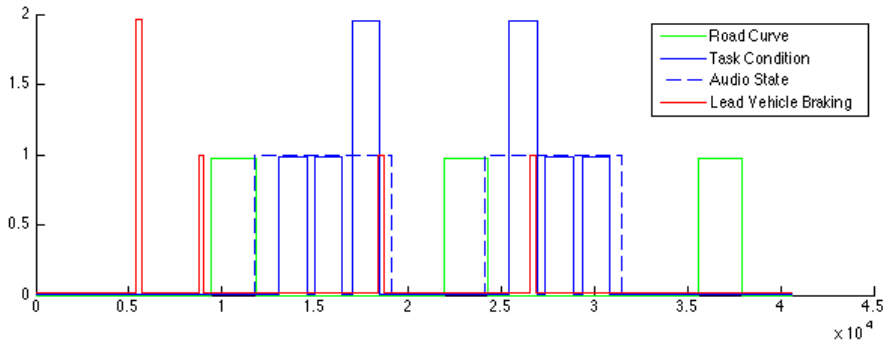


Figure 3.9: Experiment conditions recorded in meta-data files from a single drive.

Periods also contained the introduction and ending of each audio recording (also shown in Figure 3.7), as well as the gaps participants answered to the previous task and waited for the next task to start. For 1-back and 2-back scenarios, participants were likely to be more attentive to driving during these gaps, since they wanted to compensate the degraded driving performance during neighboring tasked periods. Furthermore, there were no added cognitive load demands during these times. Therefore, to be more precise, these portions should also be trimmed, meaning that *only the segments where the letter sequences were played* will be focused. This is marked as positive "Task Condition" in Figure 3.9. Each n-back task consists of 10 letters, presented at a rate of 2.25 seconds per letter, which makes each focus periods to be 25-second long. In the no-task scenarios, corresponding segments could be found and used for analysis.

In the end, because there are three tasks in each Critical Period and two Critical Periods per drive, there are 6 focus periods in each drive. Furthermore, analysis might consider excluding the braking events since they could pose significantly affect the participants, which could possibly override the effect of cognitive load. This could be easily achieved with the information available from the meta-data.

3.2.2 Visual Observations

There are two sources of visual information. The estimated results from the eye-tracker will be used for simulation studies, with the data gathered from the color cameras as complementary data. Their specific usages will be explained in detail when reporting

these phases. Below are some technical specifications of the raw data.

Eye Tracking Data

Eye tracking data was collected using faceLAB 5.0, which was described in details in 3.1.2. For each participant, the camera model, head model and screen model were always re-calibrated according to the sitting position of the participant to ensure the data quality. However, for some participants, the eye tracker still failed to work well. Three participants (P1, P08, P18) did not have valid eye-tracking results at all.

A single faceLAB recording consists of five output files that contains different types of information:

- The "time" file: experiment timing information (not used).
- The "head" file: head position, rotation and other related information.
- The "eye" file: eye closure, blinking, eyeball center and raw gaze data (rotation angle).
- The "face" file: facial feature points in 3D world coordinate.
- The "world" file: gaze intersection with items defined in the world (i.e. centre screen, right/left screens or dashboard).

faceLAB has an official "Output Data Reference Guide" that explains the data that are stored in each columns of each files. For most variables (e.g. blinking, gaze intersection), there are data for left eye, right eye and vergence, which in the cases of many participants could be different.

faceLAB was synced with the miniSim driving simulator, meaning that its frame number is forwarded to miniSim and stored in the DAQ files, in the field called ET_frame_num. However, it may not be very stable at the very beginning of the data recording (see an example in Figure **??**), possibly caused by miniSim's initialization process: the log may have started before it started receiving eye tracking information. Thus when syncing, it is advised to look at the data after it is stabilized. It would not affect the data segments where n-back tasks played.

Video Recordings

Three cameras were used in experiments: a GoPro camera that store video recording in its own SD card and two Logitech webcams (C920 and C930) that feeds video inputs into D-Lab recordings. They are approximately 27 degrees right-yaw, 12 degrees left-yaw and 12 degrees upper-pitch to the participant's face, respectively. Example frames from the camera are shown in Figure 3.10.

Figure 3.10: Video frames from three different cameras. From left to right, they are from GoPro, Logitech C930 and Logitech C920.

As mentioned above, there were some technical limitations that caused instability in D-Lab video recordings. Thus, the GoPro camera was added after the 6th participant to ensure there would be a reliable copy of video recording, which became the primary source of video data. GoPro camera outputs three files for each recording with the same name and different extensions: ".mp4", ".lrv" and ".thm". The latter two files are auxiliary files generated automatically by GoPro, which are essentially a low-resolution video file and a thumbnail image file. The main MP4 videos have the following format:

- Video encoding standard = h264 (High), yuvj420p
- Resolution = 720x480
- Frame-per-second (fps) = 29.97
- Audio encoding standards = AAC-LC, 48000 Hz

3.3 Effectiveness of the Experiment Design

The research hypothesis for the data collection phase is that the participants' cognitive load would be effectively changed with the above experiment methodology. In order to validate the resulted data collected using our experiment design, we examine two common measurements that are representative of cognitive load changes: subjective ratings from NASA-TLX and physiological responses. This section presents unprocessed raw data, which could result in non-ideal distributions (e.g. unexpected outliers) in some of the cases.

3.3.1 Perceived Workload and Mental Demand

NASA-TLX was collected immediately after each 10-minute drive. The final weighted score represents the perceived workload of the drive, which could also be used to suggest the participant's overall engagement. In addition, the "Mental Demand" sub-scale will also be considered as an indication of effectiveness of the cognitive secondary tasks. In order to show the overall distribution, box and whisker plots are produced for the whole participant group (see Figure 3.11). In addition, the differences between two incremental task-load conditions (i.e. 1-back vs. no-task, 2-back vs. no-task and 2-back vs. 1-back) are also computed for each participants and illustrated.

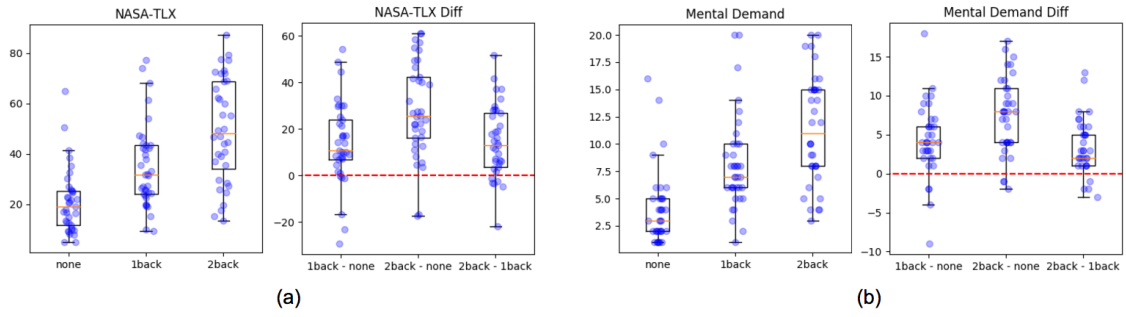


Figure 3.11: The distributions and differences due to incremental task-load levels of (a) weighted NASA-TLX scores and (b) Mental Demand. The boxes extend from the lower to upper quartile, with a line at the median. The whiskers extend until 1.5 times of the interquartile range. The actual sample points are overlaid on the plots.

There is an obvious increasing trend in both the distribution of the overall NASA-

TLX score and the Mental Demand score, suggesting that the incremental load was indeed effective. However, the range of ratings for the same load can deviate significantly across individuals: some of the participants gave a lower rating for the 2-back drive than other's no-task drive. This is very likely caused by individual differences in completing subjective ratings, but could also be a reflection of individual capability.

To remove this strong individual effect, the relative differences between the ratings are considered (second row of Figure 3.11). Overall speaking, the subjective rating differences between a higher and a lower task-load were positive. Most participants do found 1-back and 2-back was higher task-load than the no-task drive, except for a few outliers (the negative sample points). However, several participants considered the difference between 2-back and 1-back was not that significant, especially in the case of Mental Demand.

3.3.2 Physiological Responses

Subjective measurements could be biased or inaccurate, as participants might voluntarily give a higher rating for the more difficult tasks, and are only evaluated after the whole drive was completed. Physiological responses do not have these problems as they are objectively measured and obtained continuously as the task performs. Two of the popular measurement to indicate workload are the heart rate and skin conductance (i.e. sweat) level. They were also employed in previous driving studies and showed a consistent increasing trend under higher cognitive workload [70, 65]. EEG is another attractive option, however it is very complicated and its responses is less discussed in this field. Here, we simply look at the average heart rate and GSR across each participant's three task-load conditions. Similar to above, the differences are also presented. Participants that did not have valid signal recordings are excluded for analysis.

As shown in Figure 3.12, there is still an increasing trend in the distributions of these two measurements. However, there are more participants who did not get an expected positive difference when the task-load was higher, especially for GSR. This might be attributed to the unreliability of this signal, as the sensors are attached to participant's foot and might come loose without noticing. It was also unexplained why some participants' GSR values were low (around 2 μSiemens), while others were always

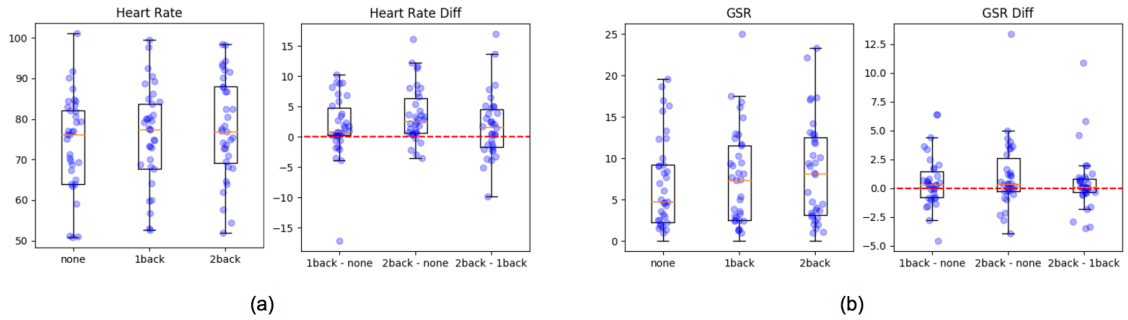


Figure 3.12: The distributions and differences due to incremental task-load levels of (a) heart rate and (b) GSR. The boxes extend from the lower to upper quartile, with a line at the median. The whiskers extend until 1.5 times of the interquartile range. The actual sample points are overlaid on the plots.

higher than 10 μSiemens.

On the other hand, heart rate were measured using ECG, whose quality could be visibly examined by looking for the obvious hear beat pattern during each experiment session. It was considered to be more reliable. While 2-back was successful in most cases at increasing the heart rate when compared to the no-task condition, other comparisons do not give as obvious conclusion.

3.4 Chapter Summary

This chapter introduced the evaluation dataset. Details of the data collection campaign are provided, which include technical details of the visual apparatus that were employed, and how the cognitive load conditions were achieved. Then, the collected data are described.

Distributions of the perceived workload and physiological responses are illustrated in box and whisker plots to show how effective were the n-back tasks and the experiment methodology. It was confirmed that the participants engagement was good, and they perceived higher workload and mental demand with more difficult n-back task. However their arousal level might not differ as much. Another observation is that the differences between each neighboring levels (i.e. 1-back vs. no-task, and 2-back vs. 1-back) are less significant.

Chapter 4

Estimating Cognitive Load with Visual Attention

Using the eye-tracking modality from the eDREAM dataset, we explore if visual attention patterns could be exploited to build driver cognitive load estimation system. The experiment presented in this chapter follows a similar logistic progression as previous literatures (as reviewed in Chapter 2). The detection of cognitive load level is framed as a classification problem, for which a model would be trained with machine learning approaches. Hand-crafted features to describe the direction and intensity of visual attention are proposed based on previous domain knowledge connecting eye-related measures with driver's cognitive load. Several commonly applied classification algorithms, including k-nearest-neighbor (KNN), support vector machine (SVM), decision tree with AdaBoost and Random Forest, are explored and compared for constructing the predictive classification model. Hyper-parameter selection is conducted with 5-fold cross validation (CV) when necessary. Then, the models are evaluated with another CV process to investigate the generalization performance.

With the encouraging results reported from previous studies, this research is developing towards a practical system. Therefore, the proposed solution should satisfy the potential requirements of functioning in real-time, and across different users. These requirements influences the modeling and evaluation methodologies. To evaluate the success of one such system, the metric and testing data should all be chosen with consid-

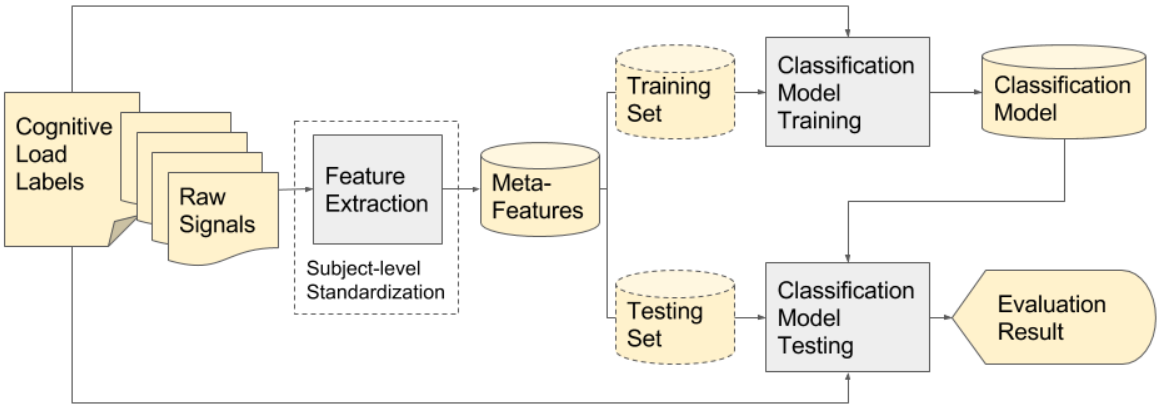


Figure 4.1: Overview of the simulation experiment. In this setup, the performance of the proposed system is evaluated based on the testing set that is held out from the training process. This is different to a practical machine learning framework depicted in Figure 2.2, where the performance is evaluated after the system is deployed and completely new data are given for classification.

erations specific to the domain; but as mentioned in Subsection 2.2.3, previous studies in this field are inconsistent in this aspect. In this experiment, special attention is directed towards discussing different evaluation procedures with three data splitting strategies. The result of the experiment is discussed for improving the system design as well as general recommendations in applying machine learning tools to model complicated, dependent data.

The key questions are the choices of input features and modeling techniques. Direct eye-tracker results will be used to provide the basis for extracting visual attention information, but the features are designed to be more suitable for practical implementation. This report will compare various machine learning approaches, and the evaluation process and associated problems are discussed.

4.1 Experiment Overview

An overview of this experiment is provided in Figure 4.1, which is a simulation of deploying a predictive model for estimating driver cognitive load in real-time using machine learning approaches (as depicted in Figure 2.2). The main difference from these two setups is that in simulation, testing set is only a subset held-out from the training set (which were collected together); but in real deployment, the model is tested with new, unseen data.

Inputs to our proposed system are raw, time-series signals $\mathbf{x}$ obtained from the faceLAB recordings included in the EDD dataset. It encompasses multiple channels that are useful for extracting eye closure and gaze direction information. The ground-truth cognitive load level, y, is retrieved based on the task conditions presented during the focus periods (as defined in Section 3.2.1) from where the raw signals were collected. Specifically speaking, in this experiment, the ground-truth labels are:

$$y = \begin{cases} \text{Low}, & \text{if subject is driving with no } n\text{-back tasks} \\ \text{Medium}, & \text{if subject is driving while performing 1-back} \\ \text{High}, & \text{if subject is driving while performing 2-back} \end{cases} \tag{4.1}$$

During the machine learning process, the target labels are denoted in the form of numerical labels 0, 1, 2 for low, medium and high cognitive load levels. In the ensuing analysis, the classifier modeling and evaluation are pursued with both 3-class and binary setups. In the binary case, only the low and high cognitive load cases are considered.

On the other hand, the raw signals $\mathbf{x}$ would be processed using the feature extraction method introduced in Section 4.2 to obtain four meta-features for describing subject's visual attention. They are called meta-features because they are obtained over several levels of information reduction, interpretation and summarization (over sliding time windows). Meta-features extracted from the same time window are concatenated into one feature vector to form a input instance ($\mathbf{X}$) for the classification model; it is coupled with the label y, marking the cognitive load level presented during the current time window.

With the input instances formed by extracted meta-features and ground-truth labels,

various supervised machine learning algorithms are applied to build the classification models using a subset of all available instances (training set). By comparing the performance of these models on classifying unseen data samples (testing set), recommendations could be made for choosing a better alternative for feature extraction or model development. The overall performance of the best setup would be taken as an indication of how promising the proposed system could function. In this experiment, we also investigate how the different strategies for partitioning the dataset into training and testing set during the CV process would influence these results.

4.1.1 Data Selection

As described in Chapter 3, EDD was collected with the goal of isolating the effect of different cognitive load levels on the participants, which was best ensured during the focus periods (see Section 3.2.1). In this experiment, eye-tracking data from the focus periods with no-task, 1-back task and 2-back task would be extracted and labeled with low, medium and high cognitive loads. In addition, a 50-second period at the beginning of each drive is also used to compute reference values for each individuals (see Subsection 4.2.2 for how it is used). The rest of dataset are not considered in this experiment.

Furthermore, some of the participants were excluded from this experiment due to problems experienced during the data collection process. They had one of the following problems (the excluded participants' identification numbers are noted in the brackets):

- Missing all or partial eye-tracking data because of setup failure (P1, P18)
- Self-reporting a lower workload through NASA-TLX as the level of cognitive task increased (P3, P31)
- Exhibiting obvious fatigue symptoms during the data collection experiment, such as drooping eye-lids and drifting away from the lane (P17)

Although removing noisy/unreliable periods or outliers could beneficial, this experiment is conducted without this additional preprocessing step. Without these exclusions,

the employed dataset represents a more realistic and practical condition, which in turn is also more challenges in developing the cognitive load monitoring system.

4.2 Meta-Features

As an early step of driver cognitive load monitoring system, feature extraction is performed to extract information embedded in the raw inputs. This experiment explores meta-features that are designed to capture two types of visual attention attributes: direction and intensity. The meta-features are listed and described in Table 4.1. The duration and count of gaze-off-center actions during a 10-second time window are estimated to capture the directional variation of visual attention; gaze-off-center actions could include incidents like checking the speedometer, mirrors, or any peripheral areas that exceeds a predefined center region. On the other hand, the duration and count of eye closure (EC) actions are included to indicate increase or decrease of visual attention intensity.

Table 4.1: Meta-Features for Summarizing Visual Behaviors in Time Windows

Notation	Description	Unit	Median[2]	IQR[2]
X_{GD_DUR}	Total duration of gaze-off-center: number of frames that the gaze direction (GD) is deviated from the reference direction by more than the threshold $thres_{GD}$.	# of frames[1]	70	(33, 121)
X_{GD_CNT}	Count of gaze-off-center times: number of times GD crossed $thres_{GD}$.	N/A	8	(4, 12)
X_{EC_DUR}	Total duration of eye closure (EC): number of frames that EC is greater than the threshold $thres_{EC}$.	# of frames[1]	26	(12, 48)
X_{EC_CNT}	Count of blinking times: number of times EC crossed $thres_{GD}$.	N/A	12	(6, 20)

Note[1]: The frequency of sample collection is 60 Hz.
Note[2]: The median and IQR (interquartile range) columns show values before doing any standardization. The time window is 10-second long. In the current implementation, $thres_{EC} = 0.50$, $thres_{GD} = 0.15$

These meta-features are proposed based on previous findings. As reviewed in Section 2.1, added cognitive load were consistently found to result in concentrated gaze and increased blinking. These findings motivated the current experiment as well as several previous studies (see Section 2.2.2) to classify cognitive load using features based on gaze direction (GD) and eye closure (EC) measurements. Two summarization methods, count and duration, are used to describe occurrences of actions checking the overall environment (indicated by gaze-off-center) or loss of visual attention (indicated by large eye closures). These two summarization methods were previously applied to describe similar events like blink or gaze entries to regions, but they were seldom applied jointly for the same event (e.g. it is more often to see use of blink frequency than together with blink duration).

The X_{EC_DUR} feature in this study is very similar to the PERCLOS metric, which is the percentage of time that the eye closure exceeds certain extend (e.g. 70% or 80%) in one minute [33]. The difference between these two measurements mainly lies in the time window and thresholding values used for calculation, which are a lot shorter (10-second) and lower (50%) for X_{EC_DUR}. PERCLOS was compared against blink metrics for assessing fatigue during driving, and was found to be more robust and coherent. The advantage could be attributed to use of fuzzier criteria, as it would allow more events than blinking (such as eye-lid "drooping") to contribute to the prediction. For this reason, X_{EC_DUR} instead of blink duration is employed in this experiment. This also motivates the use of gaze-off-center for assessing the visual attention diversion: X_{GD_DUR} could be viewed as computing the portion of time that the gaze direction is diverted from the predefined center.

In addition to the duration summarizations, the number of times the interested events occurred within the time window are also counted as the other two meta-features: X_{EC_CNT} and X_{GD_CNT}. These would contain additional information to reflect if the actions were taken rapidly or slowly (e.g. checking the speedometer once for several seconds, or checking it shortly for several times).

Though the background rationale supporting the feature choice is consistent for this study as well as previous studies, the specific meta-features could differ (previously employed features were summarized in Table 2.2). Choices of using one meta-feature rather

than another (e.g. SD of gaze rotation angle v.s. duration of gaze-off-center) were seldom explicitly discussed. This reflects the uncertainty and difficulty in translating an observed pattern into hand-crafted quantitative measures. Furthermore, in previous studies, the reason for employing slightly different meta-features for the same symptom could be due to the data collection setup, such as the specific driving environment. For example, when calculating for gaze intersection with items or gaze entries to regions, the specification of key items (e.g. mirrors or speedometer) or non-center regions could differ considerably between a single-screen simulator setup and a real driving cabin setup. This posed obstacles in employing certain features across setups.

The proposed meta-features are designed to be easy to estimate from any visual signals and setup, including tracking information from videos using more recent advancements (as reviewed in Section 2.3). They could be estimated in any data collection environment, without presuming locations of key objects (e.g. speedometer), which is a huge advantage for real-world application. Using features based on event occurrence rather than the specific characteristics of continuous-values signals also relaxes the demand in raw signal quality, thus could potentially increase the robustness and availability of the overall estimation.

4.2.1 Feature Extraction Method

In order to obtain the meta-features, the raw input signals undergo several steps of transformation as illustrated in Figure 4.2. The raw signals for the EC meta-features are the eye closure values estimated by faceLAB eye-tracker. It is a numerical value ranging from 0 to 1 indicating the fraction of eye closure:

$$x_{EC} = 1 - \frac{eyelid\_distance}{iris\_size}, \tag{4.2}$$

where $eyelid\_distance$ is the distance between top and bottom eyelids, and $iris\_size$ is the iris size, which is 12mm by default.

For the GD metafeatures, the raw signals are the gaze rotation estimated from face-

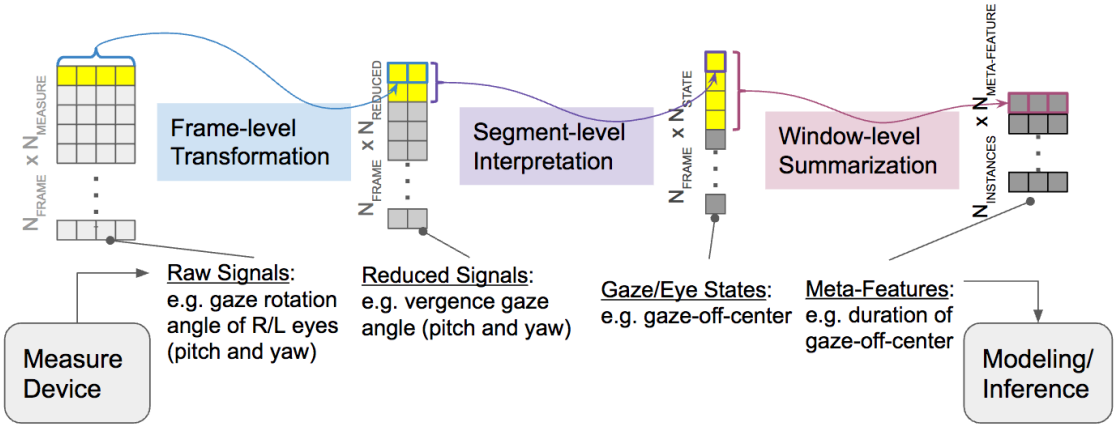


Figure 4.2: Feature extraction pipeline. The raw signals were collected at 60 Hz using the faceLAB eye-tracker employed during the data collection. It is preprocessed and interpreted at several different levels.

LAB. The values are the Euler angles measured in radians for rotation around the x-axis (pitch) and y-axis (yaw) of faceLAB's world coordinate (as illustrated in 3.4):

$$x_{GD} = [X_{GD,pitch}, X_{GD,yaw}] \tag{4.3}$$

$$= [arccos(\sqrt{u_x^2 + u_z^2}), arccos(\sqrt{u_y^2})], \tag{4.4}$$

where $\vec{u} = [u_x, u_y, u_z]$ is the unit vector pointing from pupil center to the object being looked at in world coordinate .

At each sampling frame, the raw measurements are produced for each individual eye. They are reduced into one value per measurement by taking the weighted average value. faceLAB also reports each eye's estimation confidence and gaze tracking quality level. These are used as the weights to reduce the signals measured from each eye into a single channel.

Next, occurrence of interested eye-states could be interpreted on the segment level. Two kinds of eye-states, large eye closure and gaze-off-center, are explored in the current experiment. To account for eye closure events, the measurement that indicates more than 50% of iris being covered would be binarized to 1; otherwise, it would be binarizaed

to 0. For gaze direction, the threshold value would be binarized to 1 if the deviation from the mean gaze direction in any direction exceeded the threshold, and would be binarized to 0 otherwise. Currently, both horizontal and vertical direction uses the same threshold, which is 0.15 radian (approximately 8.6 degrees). The reference gaze direction is usually not 0 radian because of the placement of eye-tracker cameras (as shown in Figure 4.3), and it is estimated on the reference data of each participant [1]. Figure 4.3 illustrates an example of the reduction and thresholding process. Looking at inferred visual action based on the behavioral signals means that some detailed information could be overlooked; however, the assumption is that most useful information should still be kept at this coarser level.

After getting the estimation of interested events, attributes could be summarized at the window level. Two kinds of attributes are considered: how long the interested events were presented, and how many times the even happened. These are referred as duration and count of a certain event. The duration could be obtained by summing the binarized signal across the current window, while count could be calculated by counting how many times the binarized signal changes values.

The window length and overlapping ratio need to be predetermined. Because behavioral events like eye closure or gaze-off-center usually occurs once every couple seconds, a time window that is more than a few seconds is required. As discussed in Subsection 2.2.2, although the system could look as far in the past as the computation/storage allows, it is still generally desirable to have the smaller time windows for the sake of data validity. Currently, a 10-second window is chosen. For each participant, the data segments with the same cognitive load level is concatenated into one, large segment in order to avoid losing too much data at the edges (i.e. the window rolls to the next available segment automatically). Figure 4.4 shows an example of the extracted feature values (in red) and the thresholded signals (in blue). After this, the extracted features are further down-sampled by every 60 frames, corresponding to generating one input instance every

[1] This information represents the prior knowledge of the subjects' behavioral characteristics. In real-life, this might be obtained per each driver either as available records or from an initialization period. Currently in this experiment, this is obtained as the statistics describing the whole trimmed data from the subject.

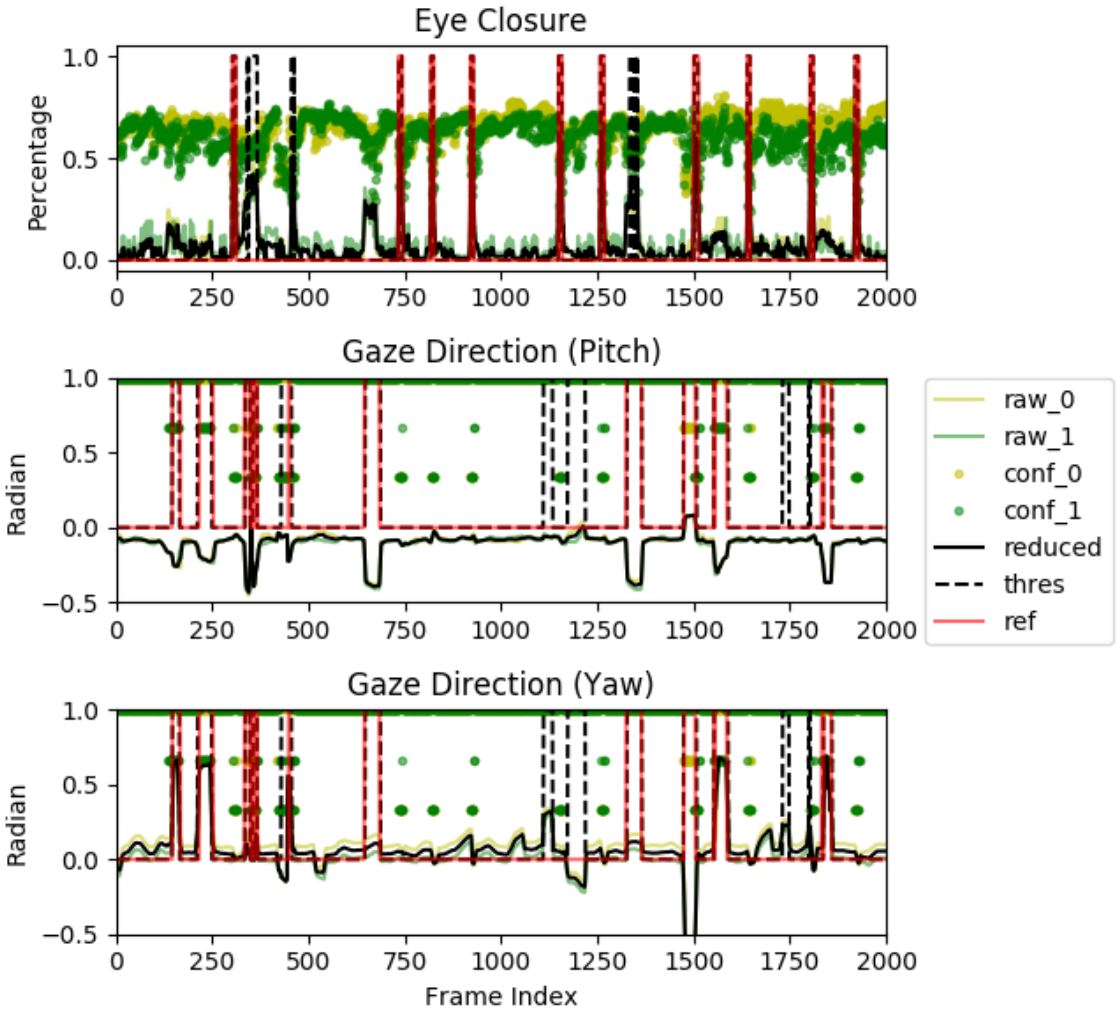


Figure 4.3: Examples of data reduction and thresholding results of a single focus period. Eye closure, horizontal gaze direction and vertical gaze direction are plotted from top to bottom. The raw signals from left and right eyes ("raw_0" and "raw_1") are plotted with light green and yellow lines, while the samples quality is plotted with scattered points of the same color. Each pair of raw signals are reduced into one single signal (in solid black lines), and the thresholded value is in black dotted lines (labeled "thres"). Note that for gaze direction, both pitch and yaw signals would be combined to obtain a single thresholded value, thus the "thres" signals in the second and third plot are identical. The "ref" signal in red is the eye-tracker generated detection of blinking for the first plot, and eyes-off-center (obtained based on the estimated gaze intersection item (see Subsection 3.1.2) for the second and third plot.

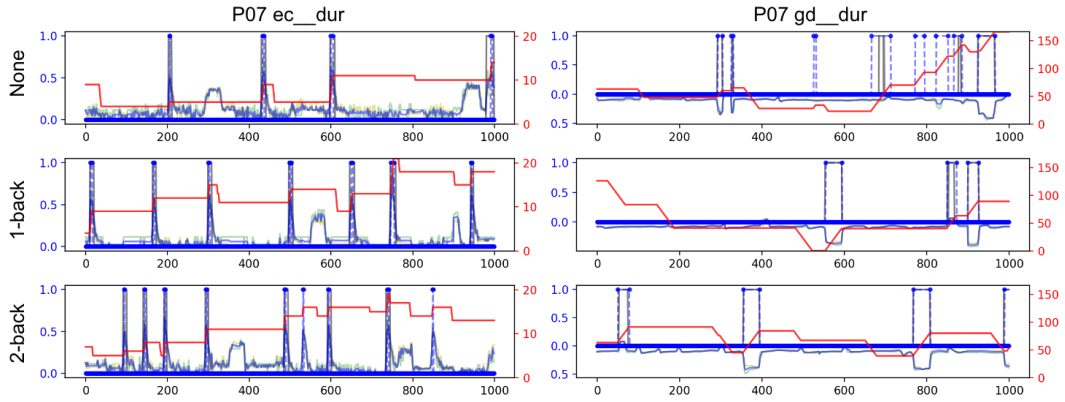


Figure 4.4: Example of X_{EC_DUR} and X_{GD_DUR} (in red) from Participant 07, calculated based on the thresholded signals (in blue). The raw signals (in light yellow, green and blue) are also plotted. Each row shows a segment of data from a different cognitive load: low, medium and high. The feature values' vertical axis are on the right.

second as the signals are 60 Hz. This setup is equivalent to an overlapping ratio of 90%.

4.2.2 Subject-level Standardization

After feature extraction, the resulted data could already be fed into the classification models. The box plots of the resulted data points is provided in the first row of Figure 4.5. This can reflect the inter-subject variation of each subjects' behavior characteristic. For example, the common blink rate could differ because of biological differences (such as drier eyes), and attentiveness to observing the external environment could vary between driving styles. This could pose trouble for subject-independent analysis.

In order to alleviate the problem of individual differences, each subjects' data are standardized according to their own statistical characteristics [1]. A common way of standardization is to remove mean and scale by SD. In the current experiment, it is actually performed by removing median and scale by inter quantile range (IQR), which makes the standardization more robust against outliers. The resulted data point distribution is shown in the second row of Figure 4.5. It could be read as the relative value of one's performance, which should be zero-median after the standardization.

From the boxes and whistles in Figure 4.5, we could identify a slight increasing trend for eye closure related features, and a decreasing trend for gaze direction related features.

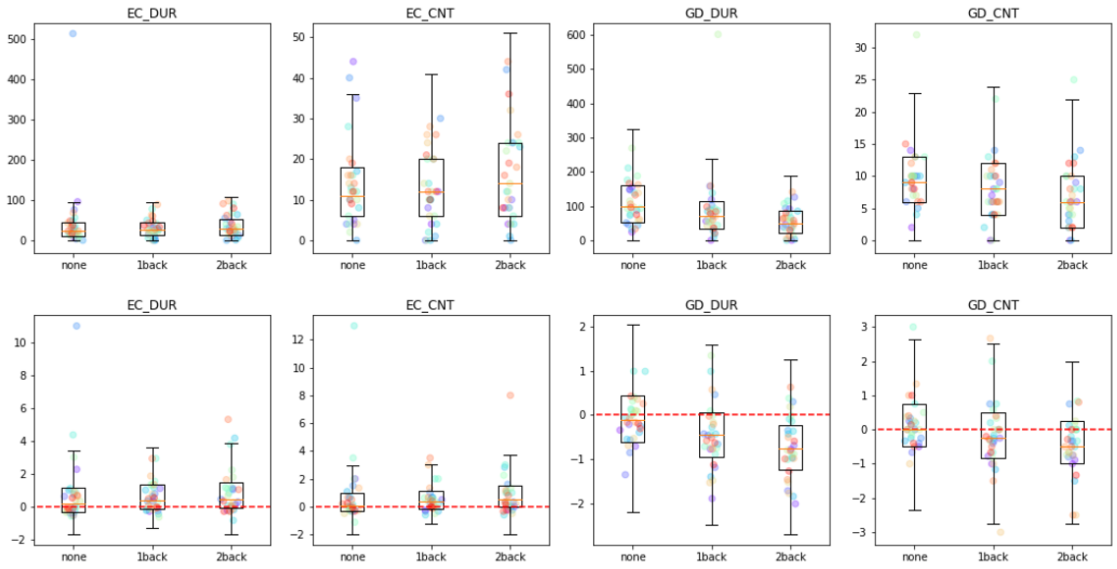


Figure 4.5: Box plots of the four extracted meta-features over the whole population under each cognitive load condition: eye closure duration, eye closure frequency, gaze duration, and gaze direction frequency. The first and second row are the results before and after subject-level standardization. Each box marks the range between first and third quartile, and the median is marked by an orange horizontal line within the box. The upper and lower whiskers reach 1.5 times of IQR beyond boundary of the box. The box plots are generated with all feature values across all subjects, but the overlaid scatter points show the median of each subjects' data to indicate the individual differences.

This matches the expectation that under higher cognitive loads, blinks would increase and peripheral/instrumental checks would decrease. Instead of the actual data points, the color-coded scattered points presents the median of each subject's data points with the purpose of exploring individual differences.

4.3 Classification Algorithms

Due to the lack of strong domain knowledge to build rule-based model, machine learning algorithms that could discover the underlying structure of data should be explored for detecting driver cognitive load [31]. As reviewed in Section 2.2.3, a variety of classification algorithms were already experimented previously for detecting driver's cognitive distractions. The principle rationale between different classification models could differ significantly. However, comparing to theoretical arguments, the superiority of the chosen

classification algorithm were more often verified empirically through evaluation results.

For this reason, this experiment also explores a few existing machine learning algorithms that were found promising for handling the current problem. Five classification algorithms are considered: k-nearest neighbor (KNN), Logistic Regression (LR), Support Vector Machines (SVM), AdaBoost, and Random Forest. The logistic behind these algorithms were provided in Section 2.2.3. This section focuses more on specifying details of the training methodologies and range of hyper-parameters.

The simulation is implemented as Python programs, and the Scikit-learn library is applied for all of the classification algorithms [71]. This library also contains useful functions for performing evaluations and cross validations. Other necessary tools used in this experiment includes the Numpy and Matplotlib libraries in Python, which handles the array data structure and visualization of results.

4.3.1 k-Nearest Neighbors

KNN classifies a given test instance with the labels of k closest training data points in the feature space. Therefore there is no training process for the KNN algorithm. The Minkowski metric is used for distance calculation, which could be thought as a generalized version of Euclidean distance to higher dimensions. There is an option of using the inverse of distance to weight the votes from the neighbors, which will result in the closer points to be more influential than the further points; if this is not enabled, all of the k nearest neighbors' votes will be considered uniformly.

The hyper-parameters and possible values considered for KNN are the following in this experiment:

- Number of neighbors (K): $1, 5, 10, 20, 50, 100$
- Voting weights: uniform, distance

KNN could inherently handle multi-class problem, since the testing point would just be classified as the class that scored the highest votes.

4.3.2 Logistic Regression

LR is a binary classification algorithm suitable for handling linear-separable classes. The result of LR could be interpreted as probability, which is the only candidate algorithm in this experiment that have this special property. In this experiment, LR is optimized using Newton conjugate gradient (Newton-CG) algorithm [72]. Similar to SVM, it has a parameter (C) that controls the regularization strength of the estimation.

The hyper-parameter and possible values considered for LR are the following in this experiment:

- Inverse of regularization strength (C): $2^{-5}, 2^{-4}, 2^{-3}, ..., 2^{4}, 2^{5}$

Multi-class logistic regression is implemented with the one-versus-rest ("OVR") strategy, which trains a classifier for each class while considering all the rest of classes as negatives. The final labeling is obtained by finding the class assignment that results in highest probability

4.3.3 Support Vector Machines

SVM with Gaussian radial basis function (RBF) kernel has been a popular binary classification algorithm [37]. The original SVM algorithm proposed by Vapnik seeks a linear hyper-plane that separates two classes of data with maximum margin [36]. Because it is not always to separate the training samples perfectly, a "soft-margin" is more often implemented by minimizing the hinge loss function, which assigns a penalization (with a hyper-parameter "C") for misclassified training samples. Another necessary trick for most SVM application is transforming the original samples via a non-linear kernel, which could prepare the original samples to become more linear-separable. We apply the popular RBF kernel in this experiment, which only has one hyper-parameter ("$gamma$") controlling the influence of each training sample. It is very important to search over a range of possible hyper-parameters to determine the optimum setup [38].

The hyper-parameter and possible values for SVM considered in this experiment are the following [24]:

- Influence of each training sample (*gamma*): $2^{-5}, 2^{-4}, 2^{-3}, ..., 2^{4}, 2^{5}$

- Cost of misclassifying samples (C): $2^{-5}, 2^{-4}, 2^{-3}, ..., 2^{4}, 2^{5}$

To apply SVM on multi-class problems, a group of models are built with the OVR strategy, which is the same strategy applied for LR. The final labeling is predicted based on finding which class assignment would grant the largest overall margin across models.

4.3.4 Ensemble of Decision Trees

Decision trees are another family of supervised machine learning algorithm that could handle classification tasks. In this experiment, the CART algorithm [41] is employed for the weak classifiers in the ensemble methods. The maximum depth of the base decision trees is 3 layers. The problem with directly applying decision trees is that it tends to overfit the training samples a lot, therefore it is usually applied as the base estimator (or weak classifier) in an ensemble method. There are two main strategies of building ensemble of decision trees, namely boosting or bootstrap aggregation ("bagging"). This experiment employed the AdaBoost and Random Forest algorithms for each of the ensemble strategies. Both of them could consider the number of base estimators as a hyper-parameter.

The hyper-parameter and possible values considered for AdaBoost are the following in this experiment:

- Number of estimators: $5, 10, 100, 500, 1000$

The hyper-parameter and possible values considered for Random Forest are the following in this experiment:

- Number of estimators: $5, 10, 50$

Both AdaBoost and Random Forest could innately handle muli-class classification, since the classes could be simply encoded as the leaf nodes.

4.4 Evaluation Procedure

As depicted in Figure 4.1 at the beginning of this chapter, simulation experiment with machine learning approaches often requires artificially splitting up all available data into non-overlapping subsets for training and testing. This is necessary for avoiding training a predictive model and evaluating its performance on the same data, as it would lead to over-optimistic result [43]. It is widely acknowledged that how the training and testing set are split apart could impact the result significantly, and this section is dedicated to describe the different methods employed in the current experiment.

4.4.1 Place in the Whole Machine Learning Workflow

Evaluations are performed for two purposes: model selection (for parameter tuning) and performance evaluation (for getting final results). As illustrated in Figure 4.6, model selection is conducted by trying different hyper-parameters and choosing the combination that led to best evaluation score; then, the selected model is tested with new, unseen data to indicate how well the trained model generalize under realistic conditions. Note that in the current setup (Figure 4.6), after the optimum hyper-parameters are chosen, the classification model will be refitted with all data considered "available" for model development, which include both the training and evaluation during model selection. In order to obtain robust and unbiased evaluation of the best trained model, two different subsets of data should be used for evaluation at each level: the subset used for model selection is referred as "*validation*" set while the one used for the final performance evaluation is referred as "*testing*" set. This is the nested CV setup recommended by [73]. Conversely, in a non-nested CV setup, one would simply take the score of the best performing model during the model selection process as the final performance score, which would lead to over-optimistic result since the testing data was not truly independent of the data used for developing the model [73].

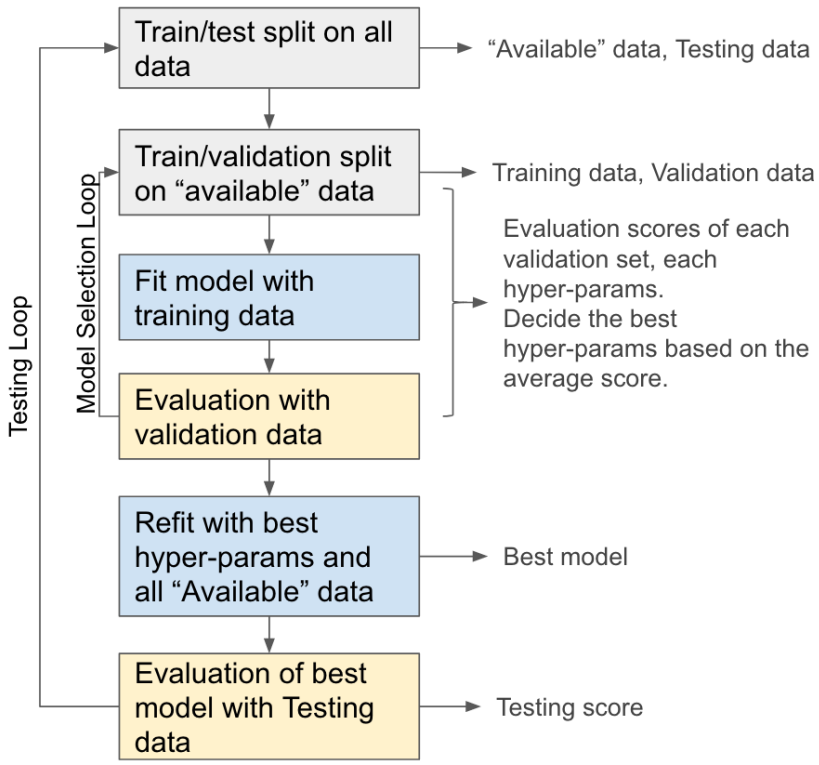


Figure 4.6: Illustration of a common workflow for developing and evaluating predictive models with machine learning approaches. Blue boxes indicate steps steps involving training, while yellow boxes denote evaluation steps.

4.4.2 Cross Validation

Because amount of data is limited in any traditional machine learning simulation, a technique called cross validation (CV) is widely applied to repeat the evaluation multiple times with different training/testing splitting. This would allow the final result to be more comprehensive and unbiased, since it lowered the possibility of obtaining unrealistic result by chance (e.g. when one particular training/testing split provide over-optimistic or over-pessimistic result). CV could be applied with a number of ways, amongst which the k-fold and leave-one-out methods are the most common ones. The current experiment employed 5-fold CV. With this method, the whole dataset is partitioned into 5 folds, and in each iteration one of the folds would be taken as the testing set while the rest would be used to train the model. The evaluation will be repeated 5 times, thus all of the samples will join in the testing set once (and only once), which ensured an even and exhaustive coverage. The average score across all iterations become the final evaluation result. This experiment applied 5-fold CV in both model selection and evaluation. A trick called stratification is applied to control the number of instances from each class is always balanced in each fold.

4.4.3 Dataset Partitioning Methods

Another part of the evaluation procedure implementation is determined how the dataset is split into training and testing sets in each CV iteration. Comparing to choosing a CV method amongst all the properly named candidates (e.g. k-Fold, leave-one-out), this part of the story is less attended and sometimes even neglected. Many studies in this field applied random drawing, while some more recent studies became aware of the innate dependency of the input instances to the classification algorithms (see the survey in Table 2.3). In this experiment, three data partitioning schemes (illustrated in Figure 4.7) are applied in order to investigate the difference they could make.

In the original k-fold CV, the data samples are required to be independent and identically distributed (i.i.d.) [43]. Although this property is seldom explicitly proved, it is often sensible to make this assumption based on the characteristics of datasets. For

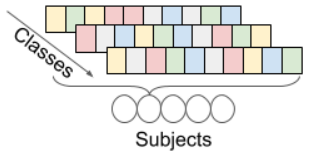
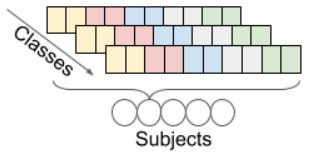
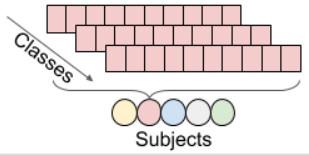
Grouping Method	Correspondent Scenario	Implementation	Illustration Color-code for folds 0 1 2 3 4
None	Data instances are i.i.d.	Drawing samples into training set or testing set randomly	
Time-based	Train a model with data from some subjects, and apply the model to predict data of unseen periods from the same subjects	Group the data from the same run into several blocks, and cross validate at the group level	
Subject-based	Train a model with data from some subjects, and apply the model to predict data from unseen subjects	Always put the data from a subject into one fold	

Figure 4.7: Illustration of data partitioning methods applied in CV iterations.

example, many of the image datasets should be valid for k-fold CV, since the samples are independently generated and their features are dominated by the underlying conditions (e.g. face or non-face images in a face detection dataset). When the same class instances are grouped together, shuffling could be applied to create an effect similar to randomly drawing samples from each class. Although this approach was adopted in several studies (e.g. [24]), it is concerned whether or not the i.i.d. assumption would hold in these cases. Two kinds of dependencies might exist in datasets studied for predicting driver cognitive load. First of all, the instances neighboring in time could be correlated. This is going to happen especially if an overlapping window is applied for extracting the meta-data. Secondly, the subject from which the instances were generated could also have significant influence.

A method to address the dependencies is to group the data instances based on its membership. In this experiment, the membership could be which drive or subject the data instances were generated from. Figure 4.7 provides illustrations and comparisons between data partitioning schemes with no grouping, time-based grouping and subject-based grouping. As the name suggested, time-based grouping divide the instances gener-

ated in the same time period (e.g. a single drive) into 5 consecutive folds. On the other hand, subject-based grouping always keeps data from a single subject in one fold. From a practical point of view, splitting data without any grouping does not correspond to any realistic situation, while the two grouped-partitioning schemes were inspired by application scenario. Time-based grouping could be thought as using previously collected data from certain user groups to pre-train a predictive model, and then applying the model onto new data generated from the same group of users. Subject-based grouping corresponds to a scenario where the model is developed using data from training participants, and then deployed to handle new, unseen users.

4.4.4 Evaluation Metrics

In the literatures, the most common metric used evaluating classifying of driver cognitive state is the accuracy (ACC) [31, 24, 26]. It is the fraction of correctly predicted test samples of any classes. Let $\hat{y}$ denote the predicted classes and y denote the true classes, accuracy is calculated as:

$$ACC(\hat{y}, y) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} [\hat{y}_i = y_i],$$

where square bracket "[.]" represents the indicator function.

A shortcoming of ACC is that it considers all classes to be equally important. Since this experiment is developing towards a driver monitoring system for improving safety and comfort, failing to detect the presence of high cognitive load leads to more dangerous cases. On the other hand, frequent false alarms could annoy users, but would not result in severe harm. Trade-off between these two kinds of failures is common to many detection problems. They are often evaluated using a pair of metrics in binary classification: recall and precision. Recall measures how many of the truly overloaded samples are correctly classified, while precision measures how many samples classified as higher cognitive load are indeed the case.

$$recall = \frac{\text{Number of True Positives}}{\text{Number of Condition Positives}} \tag{4.5}$$

$$precision = \frac{\text{Number of True Positives}}{\text{Number of Predicted Positives}} \tag{4.6}$$

However, training of machine learning model often requires a metric to give a single numerical quantity. For this, precision and recall could be combined by getting the harmonic mean of them:

$$F_1 = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} \tag{4.7}$$

$$= 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{4.8}$$

This measurement is known as the F-score. Traditionally, the weight of two averaged values are balanced, but it is also possible to generalize to a weighted version with:

$$F_\beta = \frac{1 + \beta^2}{\frac{1}{precision} + \frac{\beta^2}{recall}} \tag{4.9}$$

$$= (1 + \beta^2) \cdot \frac{precision \cdot recall}{\beta^2 \cdot precision + recall} \tag{4.10}$$

Here, β is the weight of recall, which should be a non-negative value. Because miss detecting high cognitive load situations could result in more severe consequences than false alarms, the β is set to 2 for the current study. In other words, the F_2-score is employed as an additional evaluation metrics for the cognitive load prediction module.

For multi-class classification, the F_2-score could be computed for each class and averaged across classes. Similar process could be done for precision and recall values. This is referred as the macro-averaging method in [74], which is a method that treats all classes equally. This is valid for the current experiment since the classes are balanced.

Table 4.2: Performance Evaluation for Binary Classification

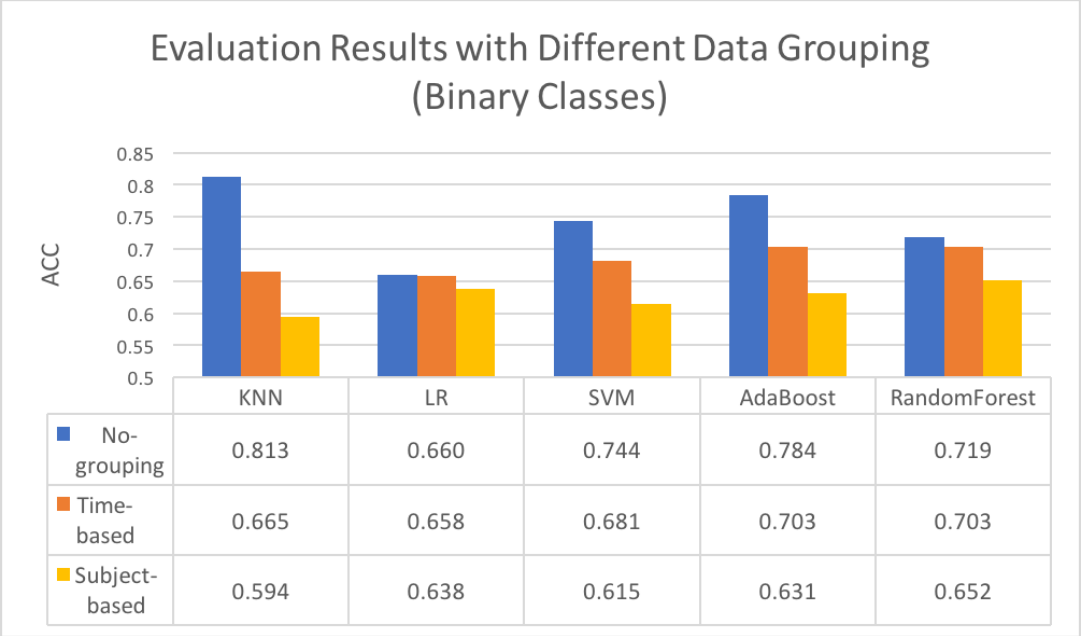
	No Grouping				Time-based Grouping				Subject-based Grouping			
	ACC	P	R	F2	ACC	P	R	F2	ACC	P	R	F2
KNN	**.813**	**.803**	**.829**	**.824**	.665	.661	.675	.671	.594	.593	.576	.575
LR	.66	.66	.661	.661	.658	.664	.662	.661	.637	.63	.653	.644
SVM	.743	.76	.711	.72	.68	.692	.652	.659	.614	.631	.552	.562
AdaBoost	.783	.789	.774	.777	.703	.708	**.688**	**.692**	.631	.65	.586	.591
RF	.718	.727	.699	.704	**.703**	**.714**	.681	.687	**.651**	**.666**	**.627**	**.629**
Guess	.498	.498	.502	.501	.488	.488	.492	.491	.503	.503	.507	.506

Note: ACC = accuracy, P = precision, R = recall, F2 = F2-score. RF = Random Forest.

4.5 Results on Binary Classes

This section presents the evaluation results of the proposed system (with various classification algorithm candidates) when applied to classify between high and low cognitive load levels (as defined in Equation 4.1). For the current section, the data of medium cognitive load is neglected (results of classification between all three classes will be presented and discussed in the next section.). The reason for only studying only two classes from the dataset is that this formulation is more comparable to previous studies (reviewed in Section 2.2), in which the classification was always between binary classes. Furthermore, this simpler setup makes it more straightforward for discussing the problems encountered when applying existing machine learning algorithm and pipeline, as many considerations necessary for multi-class cases would not arise yet.

To evaluate and compare the performance of each classification algorithms, Table 4.2 reports the test ACC, precision, recall and F2-score obtained with three evaluation procedures described in Section 4.4. The highest values under each grouping methods are noted in bold. Although these metrics are quite different, they lead to very consistent comparison results for the current experiment. Further, Figure 4.8 is created to graphically display the ACC obtained under different evaluation methods.



	KNN	LR	SVM	AdaBoost	RandomForest
No-grouping	0.813	0.660	0.744	0.784	0.719
Time-based	0.665	0.658	0.681	0.703	0.703
Subject-based	0.594	0.638	0.615	0.631	0.652

Figure 4.8: Classification accuracies with different CV grouping methods on binary-class data.

4.5.1 Differences Introduced by Evaluation Methods

As explained in Subsection 4.4.3, proper splitting of training/testing set might require some considerations when there exists prominent memberships amongst the dataset. Therefore, we performed evaluation using three data grouping methods (previously illustrated in Figure 4.7): no-grouping, time-based grouping, and subject-based grouping. As reflected in Figure 4.8, there is a significant drop in the classification performance when the grouping unit becomes larger. This result strongly speculates evaluation results obtained on time-series-like inputs without grouping could be over-optimistic.

The low performance scores obtained with subject-based grouping revealed that this is indeed the most challenging case, which corresponds to testing the system with data from new users that the model had no prior knowledge of. Although it would be ideal for a practical system to be capable of performing well under this condition, the differences between individuals' behavioral habits deemed this to be a very ambitious goal. Clearly, the drop of performance score when moving into this setup indicates that individual differences is still very prominent albeit the simple subject-level standardization included in the proposed system (described in Subsection 4.2.2). In general, it might not be

feasible to rely solely on the classification algorithms to handle the heterogeneousness within input data, though some processes (e.g. standardization) and algorithms might be able to help. A more promising direction would be investing in composing classification instances, such as considering more meta-features or fusing higher number of inputs with dimension reduction techniques (e.g. principal component analysis). Other alternative solutions could involve changing the system pipeline by adding procedures like clustering subjects by behavioral characteristics and then develop the prediction model separately for each types of subjects.

The time-based grouping resulted in intermediate performance scores, which was expected as it was considered to present an medium difficulty. It could be thought as representing an application scenario where the testing instances are new observations from the same group of subjects used for training; but it is not as harsh as subject-based grouping since the model could gain some knowledge of the users' individual characteristic during training. As the two grouped-CV represents more realistic application scenarios, results obtained under these setups should receive more recognition.

4.5.2 Algorithm Comparison

Judging from the scores, the simpler algorithms, KNN and LR, are clearly incompetent in handling classifying the current dataset under grouped-CV evaluations. However, their disadvantages are quite distinct. As shown in Table 4.2, KNN actually achieves the highest score (81.3% ACC) under no-grouping CV. The hyper-parameter selection for this condition is $K = 1$, meaning only one neighbor is sufficient for estimation. This was also reported in another study [30]. The surprisingly good evaluation score is most likely due to the high auto-correlation within the classification instances. As the instances were extracted from overlapping windows, the consecutive ones in time would have similar (or even identical) feature values. Thus, a testing instance could just be classified correctly since the votes were from its consecutive instances that got into the training set. To support this argument, the drop of performance score when grouped-CV was used is also the severest for KNN. The auto-correlation resulted from time-series data could be the reason all algorithms achieved the highest score when evaluated with no-grouping CV,

though the degree of this unfair benefit differs. As KNN is not a "smart" classification algorithm, it is very sensitive to this bias. Therefore, if KNN "outperformed" other more advanced classification methods on difficult problems, it might be worth investigating if the evaluation was done improperly and information were leaked between training and testing sets.

LR was already found inferior to SVM in another study for estimating driver cognitive load [24]. The disadvantage is attributed to LR's incapability of handling non-linear data. However, the drop of performance score under grouped-CV is least significant for LR. This could also because LR was a linear classifier, and this strict condition limited its flexibility and tendency of overfitting the training set. Thus, contradictory to KNN, LR took the least of unfair advantage from the correlation between training and testing data.

Now we focus the discussion on the other three more complicated algorithms, which achieved more promising results under grouped-CV evaluations. Both of the ensemble methods outperformed SVM in the current experiment: they achieved the best performance with 70.3% ACC under time-based grouping, while Random Forest scored the highest ACC (63.9%) under subject-based grouping. The overall differences between the performance scores for these two ensemble methods are quite small. However, AdaBoost might have a tendency of overfitting, reflected by the high scores obtained when evaluation is performed with no grouping. On the other hand, Random Forest's robustness against overfitting (as claimed in [75]) is demonstrated in our results, since the performance decline is smaller.

4.6 Results on Ternary Classes

The same machine learning pipeline is applied to classify instances with all three class labels (low, medium and high cognitive load). For the binary classification algorithms (LR and SVM), the multi-class problem is handled with one-versus-rest strategy. The evaluation metrics, test ACC, precision, recall and F2-score are reported in Table 4.3; again, these are reported using all of the three data partitioning methods described in Section 4.4. Figure 4.9 is created to graphically display the ACC obtained under different

Table 4.3: Performance Evaluation for 3-Class Classification

	No Grouping				Time-based Grouping				Subject-based Grouping			
	ACC	P	R	F2	ACC	P	R	F2	ACC	P	R	F2
KNN	**.678**	.678	.678	.678	.463	.463	.463	.463	.376	.376	.376	.375
LR	.432	.41	.432	.42	.423	.403	.423	.412	.408	.39	.408	.393
SVM	.57	.573	.57	.569	.48	.48	.48	.478	.395	.393	.395	.39
AdaBoost	.599	.598	.599	.598	**.496**	.493	.496	.494	.409	.406	.409	.406
RF	.519	.516	.519	.514	.474	.468	.474	.465	**.418**	.406	.418	.41
Guess	.341	.341	.341	.341	.337	.337	.337	.337	.334	.334	.334	.334

Note: ACC = accuracy, P = precision, R = recall, F2 = F2-score. RF = Random Forest.

evaluation methods.

For ternary classes, ACC achieved by random guess would be close to $\frac{1}{3}$. Thus, all of the classifiers achieved better-than-guess performance, though they are not able to correctly predict for cognitive load levels on large portion of testing data. The results confirmed that the proposed meta-features do carry useful predictive power for driver cognitive load estimation; however, identifying which one of the three cognitive load levels solely based on the visual attention meta-features might be inadequate.

The same declining tendency introduced by making the evaluation procedures more realistic remains very evident in the ternary class case. KNN again exhibits a strong sensitivity for dataset bias as it tops evaluation scores when the CV is conducted with no grouping. LR is still the most robust algorithm against this problem, followed by Random Forest; however, the latter could achieve much more promising performance with the strength of more complicated models. AdaBoost achieves best scores under time-based grouping with 49.6% ACC, while Random Forest is the best performing algorithm under subject-based grouping. Overall, the ensemble methods are still found to be more superior than SVM for classification between ternary cases.

4.6.1 Metrics for Multi-Class Problems

Precision, recall and F2-score differs substantially from ACC when there is a unique positive class, which exists naturally for binary class problems (see Table 4.2). Interestingly,

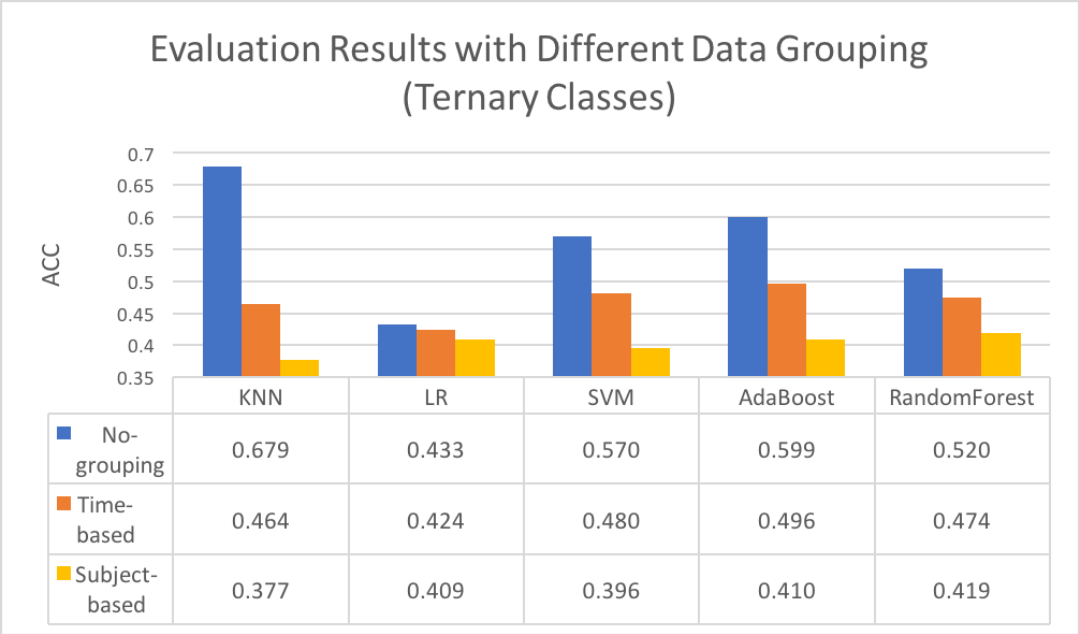


Figure 4.9: Classification accuracies with different CV grouping methods with ternary-class data.

as shown in Table 4.3, these metrics appear to report very similar values for a given algorithm and evaluation condition. Note that the reported values are rounded and thus the actual values are not as identical as it appears in this table. The similarity might be caused by the macro-averaging method applied for obtaining binary metrics (precision, recall and F_β-score) for multi-class classification, which was explained in 4.4.4. When all of the classes get a chance to be considered as the positive case, the specialness of the positive case is washed out. Similarly, the reasoning for assigning the β value in F_β-score also no longer holds after macro-average.

The repetition observed in the current results suggests that adapting methodologies suitable for binary classification into multi-class problems might need further research and considerations. On the other hand, this reveals some advantage of the popular ACC metric, which could be a satisfying metric when class balancing was considered and handled. This is also the reason why evaluations and discussions are primarily conducted with ACC in this chapter.

4.6.2 Difficulties with Ternary Classes

Difficulties of predicting the correct label after adding the medium cognitive load data could be intuitively understood. By introducing an intermediate level, it not only adds another erroneous candidate, but also requires the model to be more discriminative as the margin between each level would probably become smaller. For example, consider an high cognitive load instance, the current binary classification results (Section 4.5) already suggested that the chance of mis-classifying it as low cognitive load is not very slim. If we assume this instance was classified correctly in the binary case, there would still be a very high chance that the 3-class classifier might confuse it with the medium load case.

That been said, dealing with ordered classes (e.g. low, medium, high) should be treated differently than categorical classes (e.g. car, human, cat); however, the ordinary nature of target classes is not exploited in this experiment. Furthermore, current evaluation metrics would penalize mis-classifying an high-load instance into low-load or medium-load equally, which might not be the most helpful feedback for model development either.

An alternative way of utilizing dataset with multi-class labels is to translate several classes into positive or negative cases and then conduct binary classification, which is an approach taken by several recent studies [25, 30]. This would help avoiding issues and difficulties with multi-class classifications, but still maintaining a promising usefulness if the system could reliably distinguish between normal and abnormal conditions. Another benefit of forming binary classes based on higher-granularity labels is that the division could be better justified. This was discussed quite extensively in [25], where unsupervised clustering was performed to identify binary classification targets.

With the 3-class labels in this experiment, a possible alternative setup could be training classifiers to distinguish between the high-load cases with the rest, which would satisfy the need of detecting cognitive overload situations. Furthermore, with addition of another classifier separating the high-load, medium-load cases with the low-load case, the combined system could achieve better granularity desirable for more precise monitoring. However, the training and evaluating process for developing these two models might

involve less confusion and challenge.

4.7 Chapter Summary

This chapter proposed and evaluated a driver cognitive load estimation system developed with the eDREAM dataset. The system has two major components: the feature extraction module designed based on prior knowledge, and the estimation module trained with supervised machine learning algorithms.

The fundamental idea of our proposed system is to exploit the decrease and narrowing of drivers' visual attention when imposed with higher cognitive load, which was consistently observed in many previous human factor studies. However, the problem remains for finding effective feature descriptors to abstract these observations into predictive, quantitative measurements. X_{EC_DUR} and X_{EC_CNT} are designed to capture the proportion and frequency of cases where subjects' eyes are mostly closed, which is an indication for loss of visual attention. Parallel to these two meta-features, X_{GD_DUR} and X_{GD_CNT} are employed to describe gaze-off-center actions, which is theorized to decrease when the limited mental resources are occupied with high cognitive demands. These two tendencies are confirmed by plotting the data distribution using box-plots (Figure 4.5). In addition, in order to alleviate the problem of individual differences, data samples are standardized with reference medium and IQR computed from three 50-second data segments near the start of each drive.

The introduced meta-features are concatenated into input instances for classification models. Five existing machine learning algorithms are explored: KNN, LR, SVM, AdaBoost and Random Forest. Since these algorithms were already implemented in software toolboxes, they could be applied quite conveniently. However, some issues specific to the application condition might be overlooked, such as dependencies (or memberships) that exist in the features generated from time-series data. Both binary classification and ternary classification results (Figure 4.8 and 4.9) prove that with time-correlated inputs, simply evaluating model performance with no-grouping 5-fold CV would lead to over-optimistic scores. This is further supported by seeing that 1NN outperforms all the other

"smarter" algorithms, as this is most likely achieved by exploiting the correlation between training and testing instances.

On the other hand, time-based and subject-based grouping constraints the correlated instances to fall together into either training or testing set in each CV iterations. They could correspond to realistic situations for evaluating the model performance after deployment. Thus, results obtained with grouped-CV are more valid and robust. For both binary and ternary classifications, all classifications achieved better than guess performance with grouped-CV, demonstrating the predictive power within the proposed features. However, even with the more promising algorithms (AdaBoost and Random Forest), there is still large room for improvement ($< 70\%$ for binary cases, and $< 50\%$ for ternary cases). This reflects that relying solely on the four meta-feature might not be sufficient to handle the challenging problem of predicting driver cognitive load.

Chapter 5

Conclusions

As the level of vehicle automation increases nowadays, the necessity for human intervention will likely remain for high-level, strategic decisions as opposed to low-level operations. Thus, this research is developing towards a practical driver monitoring system to detect high cognitive loads.

The exact definition of driver's cognitive load and its importance in ensuring a safe and comfortable driving experience were introduced in Chapter 1, where the reasoning for adopting visual inputs for performing estimation was also explained. Then, Chapter 2 provided an extensive review of research studies contributing towards a video-based driver monitoring system for detecting high cognitive load. Gaze concentration and increased blink rate were found to be the reliable patterns associated with increased cognitive load during driving, which led to the choice of meta-features in our proposed system. Chapter 3 reported The data collection setup, process, and results with emphasis on how the experiment was designed to isolate cognitive load variations within participants. The resulted dataset was employed for developing the proposed driver cognitive load monitoring system, as presented in Chapter 4. Two key components of the proposed system (feature extraction and training prediction model) were described separately, with implementation details. The overall system performance is evaluated with three different grouping setups concerning time and subject correlations within the dataset. The ensuing discussion based on the obtained results investigated application of machine learning methods and compared the classification algorithms.

5.1 Conclusions and Summary of Contributions

Estimation of a person's internal cognitive load based on external observations is already a very complicated task, and objectives such as practicality, subject-independence and real-time capability add further difficulties. Another level of complication is introduced when the subject is multitasking instead of concentrating solely on handling the cognitive task. The problem of estimating cognitive load during driving falls under this challenging case. This thesis completes and investigates necessary foundational tasks towards this final goal.

In the first stage of this research, the objective was to gather data that could represent the targeted problem effectively. The uncertainty of eliciting high cognitive load posed major difficulties on this task. When collecting the eDREAM dataset, we employed the widely-used n-back task to model driver cognitive states, where the load factor ("n") controls the number of items participants needed to organize mentally. The task was applied to model three differing cognitive load levels: no-task, 1-back task and 2-back task were presented in separate drives, corresponding to three levels of cognitive load (low, medium and high). Since the data collection experiment was conducted on a driving simulator, the other environmental and driving conditions could be cautiously controlled to ensure the collected responses were only impacted by the change of cognitive load. However, an issue with the well-controlled experiment environment (as compared to naturalistic setup) was that it could lead to unwanted conditions such as boredom, sleepiness or stress. This was discovered and addressed with pilot testing before the formal data collection campaign started.

In addition to the eye-tracking data exploited in this thesis, this dataset also encompasses seven other measurements, including signals from physiological, vehicle and visual sensors, as well as subjective evaluations of perceived workload. With the self-evaluation and physiological measures (ECG and GSR) that could directly measure person workload level, their values could be an indication of the effectiveness of our experimental design on increasing participants' cognitive load. To our knowledge, this was also the first dataset that encompasses all these eight modalities together, which could enable

many new approaches (e.g. data fusion, or relating different modalities) for analyzing driver cognitive load.

Though high number of parallel signals is very advantageous and valuable, collecting them jointly complicated the process considerably, and raised many practical difficulties (such as synchronization). This data collection process consumed a significant amount of time and energy to recruit participants, complete the initial collection, and clean the dataset. The resulted eDREAM dataset is organized and available through the dataset's website [76]. This dataset opens up many research opportunities that could advance the state-of-arts of driver cognitive load monitoring.

This thesis explored the feasibility of predicting driver cognitive load based on the eye-tracking data, which is one of the modality from this dataset. Transformation, interpretation, and summarization (over sliding-windows) were performed to refine the raw data into more informative meta-features. Two types of incidents are focused: 1) extensive eye closures (e.g. blinks) that indicate loss of attention to the visual aspects, and 2) deviation of gaze from the central region that suggests adequate attention is still available for maintaining awareness of the driving environment. Proportion and frequency of these two events within a 10-second time window were computed, providing us four meta-features for capturing intensity and directional variations of subject's visual attention. The thresholds to detect the occurrence of two focused incidents could be determined using reference data collected near the start of each drive, alleviating the individual differences that hinder the meta-features' usefulness. Although similar background knowledge (loss of visual attention with higher cognitive load) was exploited in many previous systems, the proposed meta-features in this thesis are designed to be easy to extract under varying physical collection setups. Notably, they could be measured using computer vision algorithms with videos captured by ordinary, non-frontal cameras, which is easy to collect and more practical for real-world application than traditional eye-trackers.

Next, we explored five machine learning algorithms (KNN, LR, SVM, AdaBoost and Random Forest). Though available machine learning toolboxes aided greatly for applying the algorithms, the adaptation of the general machine learning workflow (Figure 4.6) to

the specific research problem and data characteristics required in-depth considerations. We examined and showed that the evaluation results were impacted substantially by the methods of data grouping applied during CV. Without grouping, k-Fold CV's prerequisite (the i.i.d. assumption) would be violated with data like the meta-features proposed in this research [43]. This situation is common to many fields, especially the ones that analyze sequential data collected from different subject. For example, Leave-One-Trial-Out CV was recommended for functional MRI studies due to the correlation between frames from the same trial [77]. However, in the current domain, there lacked discussions on this important matter and diverse CV partitioning methods were applied. Our analysis suggested splitting correlated data randomly into training/testing sets would lead to over-optimistic conclusions, and therefore time-based grouping and subject-based grouping that carried more practical significance are preferred in evaluating future estimation models.

On the other hand, our experiment also revealed that using binary classification metrics (e.g. recall and precision) with macro-averaging to obtain evaluation scores for multi-class problems might need further adjustment. These metrics were observed to result in almost identical values, possibly because all classes received a chance to be considered as the "positive" case. The accuracy metric, on the other hand, does not have this problem and is easy to interpret. With all the above considerations, the final evaluation results proved the usefulness of our proposed visual attention meta-features for estimating driver cognitive load. Amongst the five classification algorithms we examined, the ensemble methods (AdaBoost and Random Forest) are found to perform better under the proposed classification pipeline.

Overall, this thesis answers some primary questions towards developing a practical driver monitoring system targeting cognitive load estimations. With contributions of a comprehensive dataset and discussions on proper evaluation formulation, we provide a common foundation for future research to explore and compare a lot more features and algorithms.

5.2 Future Works

5.2.1 Improvements of the Proposed System

Several technical aspects should be explored for enhancing the proposed solution for monitoring driver cognitive load:

- **Adding Features.** As mentioned when discussing classification results (Section 4.5), classifiers could not perform well if the input instances do not carry enough discriminative power, or the information might be masked by other conditions like individual differences. Since only four meta-features are exploited, the enclosed information could be limited and insufficient. More features should be explored (possibly with better data fusion strategies) to improve the accuracy, robustness, and availability of the driver monitoring system. These features could be generated from other signal modalities available from the eDREAM dataset, such as vehicle measurements or EEG signals.

- **Enhancing Preprocessing.** Problems such as noises or outliers were handled very crudely or even overlooked in this experiment. More sophisticated preprocessing with signal processing and statistical techniques should be explored. For example, when the raw signal is very unstable with a lot of rapid jumps, smoothing might be applied to remove the noises. Cleaned data would be especially advantageous for supervised machine learning since confusing instances could heavily impede the training process. Refinement like this should be conducted with a good understanding of the signal collection conditions and apparatus to avoid removing useful information or introducing biases.

- **Exploiting Time Variations.** This thesis considered a simple machine learning pipeline that treats each input instances isolated away from the other ones. The temporal variations of the raw measurements were extracted with summarization functions over sliding-windows, which is a basic, primal approach. There exist alternative strategies (such as time-series analysis) that are capable of directly analyze the data before this summarization step, and might provide better performance in

our problem. Algorithms like Recursive Neural Network or Dynamic Bayesian Network models the temporal relationships and were found to perform well in driver monitoring domains [64, 29].

- **Model Tuning.** Most machine learning algorithms have hyper-parameters that could be tuned to achieve better results. Currently, this model selection process is performed by exhaustively trying all possible options on the parameter grid, which is very inefficient and has a considerable risk of missing the optimum settings. More intelligent strategies such as Bayesian optimization [78] could be applied to search for the ideal setting with better efficiency.

5.2.2 General Future Work

This thesis also suggests for exploring some more general grounds:

- **Identification of Risks.** Evaluation of a system's performance should be associated with practical risks. In the current problem, missing high cognitive load could put a driver into dangerous conditions, but the opposite condition might not result in severe consequences. On the other hand, in reality, most of the driving time should be evaluated as normal (meaning the classes would be unbalanced), which requires the detection of high cognitive load cases to have good precision. Further investigation should be performed to formulate the trade-off, which would determine how the successfulness of a proposed solution should be evaluated. Alternatively, algorithms outputting probabilistic results could be employed since the threshold for generating a positive case could be adapted based on the desired application.

- **Targeting State Variations.** The data collection process focused on obtaining genuine, spontaneous responses from real participants. However, a fundamental problem common to many similar datasets (featuring human responses) is that the labeling of data is usually quite inflexible. For example, a whole trail of data would be labeled together according to the presented condition, while in reality, a person's

mental state could vary within a second. The labels obtained this way could still be considered as valid approximations of the overall situation. However, on the other hand, this methodology prevents studies to detect the onset and variation of the targeted problems (e.g. when the subject transformed from low cognitive load into high cognitive load). Intuitively, these conditions might be advantageous and easier to detect for practical applications. When collecting the eDREAM dataset, we observed participants might change into a more nervous facial expression or body gesture when the n-back tasks started; however, this was less obvious to spot without comparison to their natural, relaxed states. Unfortunately, the addition of cognitive load happened quite sparingly with the current experimental design (only two instances per subject per condition), obscuring application of data mining approaches. Therefore, generating more samples of the onset cases could be considered as an objective in future data collection.

- **Application on Naturalistic Data.** Solutions developed with in-laboratory dataset could be extended and tested on more general scenarios (e.g. [79]). The most common modalities available in large-scale naturalistic datasets are collected with non-contact sensors like camera or vehicle CAN-bus. Therefore, focusing analysis of these more practical modalities could open up the opportunity of employing a massive amount of data. For example, computer vision algorithms that are becoming more powerful and robust nowadays could aid adapting extraction of the proposed meta-feature with raw visual inputs captured from low-quality, non-frontal cameras. Then, the performance of the system could be benefit from mining larger amount of realistic data.

Bibliography

[1] Y. Ishigami and R. M. Klein, "Is a hands-free phone safer than a handheld phone?" *Journal of Safety Research*, vol. 40, no. 2, pp. 157–164, 2009.

[2] M. A. Regan, C. Hallett, and C. P. Gordon, "Driver distraction and driver inattention: Definition, relationship and taxonomy," *Accident Analysis & Prevention*, vol. 43, no. 5, pp. 1771–1781, 2011.

[3] R. M. Yerkes and J. D. Dodson, "The relation of strength of stimulus to rapidity of habit-formation," *Journal of Comparative Neurology and Psychology*, vol. 18, no. 5, pp. 459–482, 1908.

[4] J. F. Coughlin, B. Reimer, and B. Mehler, "Monitoring, managing, and motivating driver safety and well-being." *IEEE Pervasive Computing*, vol. 10, no. 3, 2011.

[5] J. Sweller, "Cognitive load during problem solving: Effects on learning," *Cognitive science*, vol. 12, no. 2, pp. 257–285, 1988.

[6] Y. Dong, Z. Hu, K. Uchimura, and N. Murayama, "Driver inattention monitoring system for intelligent vehicles: A review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 596–614, 2011.

[7] A. S. Aghaei, B. Donmez, C. C. Liu, D. He, G. Liu, K. N. Plataniotis, H.-Y. W. Chen, and Z. Sojoudi, "Smart driver monitoring: When signal processing meets human factors: In the driver's seat," *IEEE Signal Processing Magazine*, vol. 33, no. 6, pp. 35–48, 2016.

[8] A. Čolić, O. Marques, and B. Furht, *Driver Drowsiness Detection: Systems and Solutions*. Springer, 2014.

[9] D. Novak, "Engineering issues in physiological computing," in *Advances in Physiological Computing*. Springer, 2014, pp. 17–38.

[10] E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation and augmented reality tracking: An integrated system and evaluation for monitoring driver awareness," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 2, pp. 300–311, 2010.

[11] Q. Ji, Z. Zhu, and P. Lan, "Real-time nonintrusive monitoring and prediction of driver fatigue," *IEEE Transactions on Vehicular Technology*, vol. 53, no. 4, pp. 1052–1068, 2004.

[12] B. Cyganek and S. Gruszczyński, "Hybrid computer vision system for drivers' eye recognition and fatigue monitoring," *Neurocomputing*, vol. 126, pp. 78–94, 2014.

[13] M.-H. Sigari, M.-R. Pourshahabi, M. Soryani, and M. Fathy, "A review on driver face monitoring systems for fatigue and distraction detection," *International Journal of Advanced Science and Technology*, 2014.

[14] H.-B. Kang, "Various approaches for driver and driving behavior monitoring: A review," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2013, pp. 616–623.

[15] T. W. Victor, J. L. Harbluk, and J. A. Engström, "Sensitivity of eye-movement measures to in-vehicle task difficulty," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 8, no. 2, pp. 167–190, 2005.

[16] J. L. Harbluk, Y. I. Noy, P. L. Trbovich, and M. Eizenman, "An on-road assessment of cognitive distraction: Impacts on drivers' visual behavior and braking performance," *Accident Analysis & Prevention*, vol. 39, no. 2, pp. 372–379, 2007.

[17] B. Reimer, "Impact of cognitive task complexity on drivers' visual tunneling," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2138, pp. 13–19, 2009.

[18] B. Reimer, B. Mehler, Y. Wang, and J. F. Coughlin, "A field study on the impact of variations in short-term memory demands on drivers' visual attention and driving performance across three age groups," *Human Factors*, vol. 54, no. 3, pp. 454–468, 2012.

[19] Y. Wang, B. Reimer, J. Dobres, and B. Mehler, "The sensitivity of different methodologies for characterizing drivers' gaze concentration under increased cognitive demand," *Transportation Research Part F: Traffic Psychology and Behaviour*, vol. 26, pp. 227–237, 2014.

[20] G. J. Siegle, N. Ichikawa, and S. Steinhauer, "Blink before and after you think: blinks occur prior to and following cognitive load indexed by pupillary responses," *Psychophysiology*, vol. 45, no. 5, pp. 679–687, 2008.

[21] M. Á. Recarte, E. Pérez, Á. Conchillo, and L. M. Nunes, "Mental workload and visual impairment: Differences between pupil, blink, and subjective rating," *The Spanish Journal of Psychology*, vol. 11, no. 02, pp. 374–385, 2008.

[22] G. Marquart, C. Cabrall, and J. de Winter, "Review of eye-related measures of drivers' mental workload," *Procedia Manufacturing*, vol. 3, pp. 2854–2861, 2015.

[23] M. A. Recarte and L. M. Nunes, "Effects of verbal and spatial-imagery tasks on eye fixations while driving." *Journal of Experimental Psychology: Applied*, vol. 6, no. 1, p. 31, 2000.

[24] Y. Liang, M. L. Reyes, and J. D. Lee, "Real-time detection of driver cognitive distraction using support vector machines," *IEEE Transactions on Intelligent Transportation Systems*, vol. 8, no. 2, pp. 340–350, 2007.

[25] N. Li and C. Busso, "Predicting perceived visual and cognitive distractions of drivers with multimodal features," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 1, pp. 51–65, 2015.

[26] M. Miyaji, H. Kawanaka, and K. Oguri, "Driver's cognitive distraction detection using physiological features by the adaboost," in *Proceedings of 12th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2009, pp. 1–6.

[27] J. Son, M. Park, and H. Oh, "Detecting cognitive workload using driving performance and eye movement in a driving simulator," in *Proceedings of the 11th International Symposium on Advanced Vehicle Control*, 2012.

[28] T. Liu, Y. Yang, G.-B. Huang, Y. K. Yeo, and Z. Lin, "Driver distraction detection using semi-supervised machine learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 4, pp. 1108–1120, 2016.

[29] Y. Liang, J. Lee, and M. Reyes, "Nonintrusive detection of driver cognitive distraction in real time using bayesian networks," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2018, pp. 1–8, 2007.

[30] L. Zhang, J. Wade, D. Bian, J. Fan, A. Swanson, A. Weitlauf, Z. Warren, and N. Sarkar, "Cognitive load measurement in a virtual reality-based driving system for autism intervention," *IEEE Transactions on Affective Computing*, vol. 8, no. 2, pp. 176–189, 2017.

[31] Y. Zhang, Y. Owechko, and J. Zhang, "Driver cognitive workload estimation: A data-driven perspective," in *Proceedings of The 7th International IEEE Conference on Intelligent Transportation Systems*. IEEE, 2004, pp. 642–647.

[32] Y. Liang and J. D. Lee, "Driver cognitive distraction detection using eye movements," in *Passive Eye Monitoring*. Springer, 2008, pp. 285–300.

[33] D. F. Dinges and R. Grace, "Perclos: A valid psychophysiological measure of alertness as assessed by psychomotor vigilance," US Department of Transportation,

Federal Highway Administration, Tech. Rep. FHWA-MCRT-98-006, 1998. [Online]. Available: https://ntl.bts.gov/lib/10000/10100/10114/tb98-006.pdf

[34] R. O. Duda, P. E. Hart, D. G. Stork *et al.*, *Pattern classification*. Wiley New York, 1973, vol. 2.

[35] G. Littlewort, J. Whitehill, T. Wu, I. Fasel, M. Frank, J. Movellan, and M. Bartlett, "The computer expression recognition toolbox (cert)," in *IEEE International Conference on Automatic Face & Gesture Recognition and Workshops*. IEEE, 2011, pp. 298–305.

[36] V. Vapnik, *The nature of statistical learning theory*. Springer cience & business media, 2013.

[37] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[38] C.-W. Hsu, C.-C. Chang, C.-J. Lin *et al.*, "A practical guide to support vector classification," 2003.

[39] T. Kumagai and M. Akamatsu, "Prediction of human driving behavior using dynamic bayesian networks," *IEICE Transactions on Information and Systems*, vol. 89, no. 2, pp. 857–860, 2006.

[40] J. R. Quinlan, *C4. 5: programs for machine learning*. Elsevier, 2014.

[41] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen, *Classification and regression trees*. CRC press, 1984.

[42] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," 2011.

[43] S. Arlot, A. Celisse *et al.*, "A survey of cross-validation procedures for model selection," *Statistics Surveys*, vol. 4, pp. 40–79, 2010.

[44] L. M. Bergasa, J. Nuevo, M. A. Sotelo, R. Barea, and M. E. Lopez, "Real-time system for monitoring driver vigilance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 63–77, 2006.

[45] C. Ahlstrom, T. Victor, C. Wege, and E. Steinmetz, "Processing of eye/head-tracking data in large-scale naturalistic driving data sets," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 553–564, 2012.

[46] J. M. Saragih, S. Lucey, and J. F. Cohn, "Deformable model fitting by regularized landmark mean-shift," *International Journal of Computer Vision*, vol. 91, no. 2, pp. 200–215, 2011.

[47] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar, "Localizing parts of faces using a consensus of exemplars," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2930–2940, 2013.

[48] T. Baltrusaitis, P. Robinson, and L.-P. Morency, "Constrained local neural fields for robust facial landmark detection in the wild," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2013, pp. 354–361.

[49] A. Asthana, S. Zafeiriou, S. Cheng, and M. Pantic, "Robust discriminative response map fitting with constrained local models," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3444–3451.

[50] S. Ren, X. Cao, Y. Wei, and J. Sun, "Face alignment at 3000 fps via regressing local binary features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1685–1692.

[51] X. Xiong and F. De la Torre, "Supervised descent method and its applications to face alignment," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 532–539.

[52] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2. Ieee, 1999, pp. 1150–1157.

[53] J. Zhang, S. Shan, M. Kan, and X. Chen, "Coarse-to-fine auto-encoder networks (cfan) for real-time face alignment," in *European Conference on Computer Vision*. Springer, 2014, pp. 1–16.

[54] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Learning deep representation for face alignment with auxiliary attributes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 5, pp. 918–930, 2016.

[55] N. Wang, X. Gao, D. Tao, H. Yang, and X. Li, "Facial feature point detection: A comprehensive survey," *Neurocomputing*, 2017.

[56] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: The first facial landmark localization challenge," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2013, pp. 397–403.

[57] J. Paone, D. Bolme, R. Ferrell, D. Aykac, and T. Karnowski, "Baseline face detection, head pose estimation, and coarse direction detection for facial data in the shrp2 naturalistic driving study," in *Intelligent Vehicles Symposium (IV)*. IEEE, 2015, pp. 174–179.

[58] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE, 2001, pp. I–511.

[59] R. O. Mbouna, S. G. Kong, and M.-G. Chun, "Visual analysis of eye state and head pose for driver alertness monitoring," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1462–1469, 2013.

[60] P. Jimenez, L. M. Bergasa, J. Nuevo, N. Hernandez, and I. G. Daza, "Gaze fixation system for the evaluation of driver distractions induced by ivis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 3, pp. 1167–1178, 2012.

[61] T. Baltru, P. Robinson, L.-P. Morency *et al.*, "Openface: an open source facial behavior analysis toolkit," in *IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2016, pp. 1–10.

[62] F. De la Torre, W.-S. Chu, X. Xiong, F. Vicente, X. Ding, and J. Cohn, "Intraface," in *11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition*, vol. 1. IEEE, 2015, pp. 1–8.

[63] M. Wollmer, C. Blaschke, T. Schindl, B. Schuller, B. Farber, S. Mayer, and B. Trefflich, "Online driver distraction detection using long short-term memory," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 574–582, 2011.

[64] A. Jain, A. Singh, H. S. Koppula, S. Soh, and A. Saxena, "Recurrent neural networks for driver activity anticipation via sensory-fusion architecture," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 3118–3125.

[65] B. Mehler, B. Reimer, and J. F. Coughlin, "Sensitivity of physiological measures for detecting systematic variations in cognitive demand from a working memory task an on-road study across three age groups," *Human Factors: The Journal of the Human Factors and Ergonomics Society*, vol. 54, no. 3, pp. 396–412, 2012.

[66] S. G. Hart and L. E. Staveland, "Development of nasa-tlx (task load index): Results of empirical and theoretical research," *Advances in Psychology*, vol. 52, pp. 139–183, 1988.

[67] C. H. Chatham, S. A. Herd, A. M. Brant, T. E. Hazy, A. Miyake, R. O'Reilly, and N. P. Friedman, "From an executive network to executive control: a computational model of the n-back task," *Journal of Cognitive Neuroscience*, vol. 23, no. 11, pp. 3598–3619, 2011.

[68] S. M. Jaeggi, M. Buschkuehl, W. J. Perrig, and B. Meier, "The concurrent validity of the n-back task as a working memory measure," *Memory*, vol. 18, no. 4, pp. 394–412, 2010.

[69] B. Mehler, B. Reimer, and J. Dusek, "Mit agelab delayed digit recall task (n-back)," Massachusetts Institute of Technology, Cambridge, MA, Tech.

Rep. 2011-3B, 2011. [Online]. Available: http://agelab.mit.edu/system/files/Mehler_et_al_n-back-white-paper_2011_B.pdf

[70] B. Mehler, B. Reimer, J. Coughlin, and J. Dusek, "Impact of incremental increases in cognitive workload on physiological arousal and performance in young adult drivers," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2138, pp. 6–12, 2009.

[71] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[72] T. P. Minka, "A comparison of numerical optimizers for logistic regression," *Unpublished Draft*, 2003.

[73] G. C. Cawley and N. L. Talbot, "On over-fitting in model selection and subsequent selection bias in performance evaluation," *Journal of Machine Learning Research*, vol. 11, no. Jul, pp. 2079–2107, 2010.

[74] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.

[75] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[76] (2017) edream dataset. [Online]. Available: http://www.dsp.utoronto.ca/projects/eDREAM/

[77] M. Esterman, B. J. Tamber-Rosenau, Y.-C. Chiu, and S. Yantis, "Avoiding non-independence in fmri data analysis: leave one subject out," *Neuroimage*, vol. 50, no. 2, pp. 572–576, 2010.

[78] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Advances in Neural Information Processing Systems*, 2012, pp. 2951–2959.

[79] K. L. Campbell, "The shrp 2 naturalistic driving study: Addressing driver performance and behavior in traffic safety," *Transportation Research News*, no. 282, 2012.