MIT Autonomous Vehicle Technology Study:
Large-Scale Deep Learning Based Analysis of Driver Behavior and Interaction with Automation

Lex Fridman*, Daniel E. Brown, Michael Glazer, William Angell, Spencer Dodd, Benedikt Jenik,
Jack Terwilliger, Julia Kindelsberger, Li Ding, Sean Seaman, Hillary Abraham, Alea Mehler,
Andrew Sipperley, Anthony Pettinato, Bobbie Seppelt, Linda Angell, Bruce Mehler, Bryan Reimer*

Abstract—Today, and possibly for a long time to come, the full driving task is too complex an activity to be fully formalized as a sensing-acting robotics system that can be explicitly solved through model-based and learning-based approaches in order to achieve full unconstrained vehicle autonomy. Localization, mapping, scene perception, vehicle control, trajectory optimization, and higher-level planning decisions associated with autonomous vehicle development remain full of open challenges. This is especially true for unconstrained, real-world operation where the margin of allowable error is extremely small and the number of edge-cases is extremely large. Until these problems are solved, human beings will remain an integral part of the driving task, monitoring the AI system as it performs anywhere from just over 0% to just under 100% of the driving. The governing objectives of the MIT Autonomous Vehicle Technology (MIT-AVT) study are to (1) undertake large-scale real-world driving data collection that includes high-definition video to fuel the development of deep learning based internal and external perception systems, (2) gain a holistic understanding of how human beings interact with vehicle automation technology by integrating video data with vehicle state data, driver characteristics, mental models, and self-reported experiences with technology, and (3) identify how technology and other factors related to automation adoption and use can be improved in ways that save lives. In pursuing these objectives, we have instrumented 21 Tesla Model S and Model X vehicles, 2 Volvo S90 vehicles, 2 Range Rover Evoque, and 2 Cadillac CT6 vehicles for both long-term (over a year per driver) and medium term (one month per driver) naturalistic driving data collection. Furthermore, we are continually developing new methods for analysis of the massive-scale dataset collected from the instrumented vehicle fleet. The recorded data streams include IMU, GPS, CAN messages, and high-definition video streams of the driver face, the driver cabin, the forward roadway, and the instrument cluster (on select vehicles). The study is on-going and growing. To date, we have 99 participants, 11,846 days of participation, 405,807 miles, and 5.5 billion video frames. This paper presents the design of the study, the data collection hardware, the processing of the data, and the computer vision algorithms currently being used to extract actionable knowledge from the data.

MIT Autonomous Vehicle Technology Study

Study months to-date: **30**
Participant days: **11,846**
Drivers: **99**
Vehicles: **29**
Miles driven: **405,807**
Video frames: **5.5 billion**

Study data collection is ongoing.
Statistics updated on: Jul 20, 2018.

Vehicle	Miles	Days in study
Tesla Model S	39,320 miles	583 days in study
Tesla Model S	33,177 miles	861 days in study
Tesla Model X	31,600 miles	748 days in study
Tesla Model S	25,491 miles	572 days in study
Range Rover Evoque	22,957 miles	598 days in study
Range Rover Evoque	22,644 miles	763 days in study
Tesla Model X	21,915 miles	499 days in study
Tesla Model S	20,433 miles	647 days in study
Volvo S90	19,231 miles	634 days in study
Tesla Model X	17,035 miles	701 days in study
Tesla Model S	15,735 miles	322 days in study
Volvo S90	15,570 miles	672 days in study
Tesla Model S	15,256 miles	714 days in study
Tesla Model S	14,398 miles	371 days in study
Tesla Model S	13,010 miles	463 days in study
Tesla Model S	12,353 miles	321 days in study
Tesla Model S	10,149 miles	146 days in study
Tesla Model X	9,556 miles	378 days in study
Tesla Model S	9,076 miles	183 days in study
Tesla Model X	8,587 miles	316 days in study
Tesla Model S	8,484 miles	325 days in study
Tesla Model S	6,718 miles	194 days in study
Tesla Model X	4,587 miles	233 days in study
Tesla Model X	4,441 miles	416 days in study
Tesla Model S	2,925 miles	133 days in study
Cadillac CT6	1,161 miles	53 days in study
Tesla Model S	(Offload Pending)	
Tesla Model S	(Offload Pending)	
Cadillac CT6	(Offload Pending)	

Fig. 1: Dataset statistics for the MIT-AVT study as a whole and for the individual vehicles in the study.

* Lex Fridman (fridman@mit.edu) and Bryan Reimer (reimer@mit.edu) are primary contacts. Linda Angell and Sean Seaman are affiliated with Touchstone Evaluations, Inc. All other authors are affiliated with Massachusetts Institute of Technology (MIT).

(a) Face Camera for Driver State.

(b) Driver Cabin Camera for Driver Body Position.

(c) Forward-Facing Camera for Driving Scene Perception.

(d) Instrument Cluster Camera for Vehicle State.

Fig. 2: Video frames from MIT-AVT cameras and visualization of computer vision tasks performed for each.

I. INTRODUCTION

The idea that human beings are poor drivers is well-documented in popular culture [1], [2]. While this idea is often over-dramatized, there is some truth to it in that we're at times distracted, drowsy, drunk, drugged, and irrational decision makers [3]. However, this does not mean it is easy to design and build a perception-control system that drives better than the average human driver. The 2007 DARPA Urban Challenge [4] was a landmark achievement in robotics, when 6 of the 11 autonomous vehicles in the finals successfully navigated an urban environment to reach the finish line, with the first place finisher traveling at an average speed of 15 mph. The success of this competition led many to declare the fully autonomous driving task a "solved problem", one with only a few remaining messy details to be resolved by automakers as part of delivering a commercial product. Today, over ten years later, the problems of localization, mapping, scene perception, vehicle control, trajectory optimization, and higher-level planning decisions associated with autonomous vehicle development remain full of open challenges that have yet to be fully solved by systems incorporated into a production platforms (e.g. offered for sale) for even a restricted operational space. The testing of prototype vehicles with a human supervisor responsible for taking control during periods where the AI system is "unsure" or unable to safely proceed remains the norm [5], [6].

The belief underlying the MIT Autonomous Vehicle Technology (MIT-AVT) study is that the DARPA Urban Challenge was only a first step down a long road toward developing autonomous vehicle systems. The Urban Challenge had no people participating in the scenario except the professional drivers controlling the other 30 cars on the road that day. The authors believe that the current real-world challenge is one that has the human being as an integral part of every aspect of the system. This challenge is made especially difficult due to the immense variability inherent to the driving task due to the following factors:

- The underlying uncertainty of human behavior as represented by every type of social interaction and conflict resolution between vehicles, pedestrians, and cyclists.
- The variability between driver styles, experience, and other characteristics that contribute to their understanding, trust, and use of automation.
- The complexities and edge-cases of the scene perception and understanding problem.
- The underactuated nature of the control problem [7] for every human-in-the-loop mechanical system in the car: from the driver interaction with the steering wheel to the tires contacting the road surface.
- The expected and unexpected limitation of and imperfections in the sensors.
- The reliance on software with all the challenges inherent to software-based systems: bugs, vulnerabilities, and the constantly changing feature set from minor and major version updates.
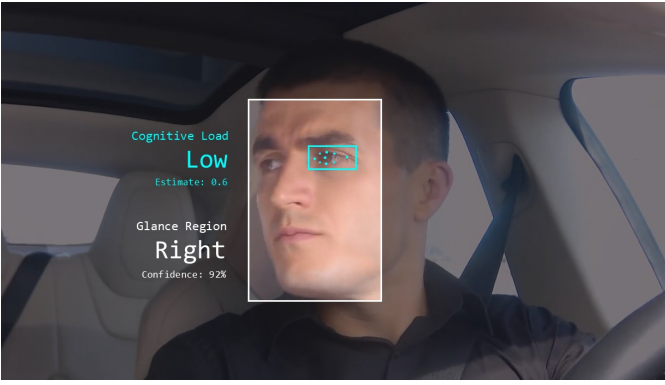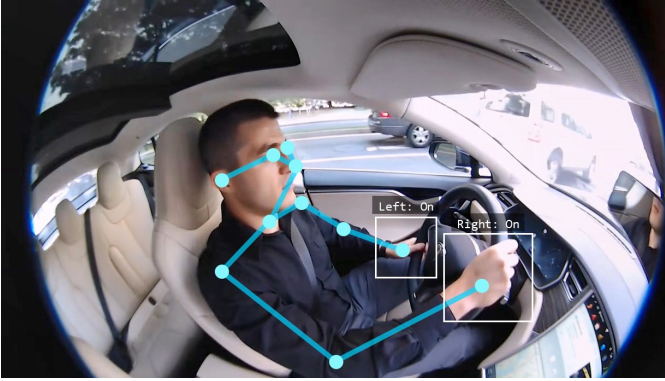
Fig. 3: Visualization of GPS points for trips in the MIT-AVT dataset local to the New England area. The full dataset contains trips that span over the entire continental United States.

- The need for a human driver to recognize, acknowledge, and be prepared to take control and adapt when system failure necessitates human control of the vehicle in order to resolve a potentially dangerous situation.
- The environmental conditions (i.e., weather, light conditions) that have a major impact on both the low-level perception and control tasks, as well as the high-level interaction dynamics among the people that take part in the interaction.
- Societal and individual tolerances to human and machine error.

As human beings, we naturally take for granted how much intelligence, in the robotics sense of the word, is required to successfully attain enough situation awareness and understanding [8] to navigate through a world full of predictably irrational human beings moving about in cars, on bikes, and on foot. It may be decades before the majority of cars on the road are fully autonomous. During this time, the human is likely to remain the critical decision maker either as the driver or as the supervisor of the AI system doing the driving.

In this context, Human-Centered Artificial Intelligence (HCAI) is an area of computer science, robotics, and experience design that aims to achieve a deeper integration between human and artificial intelligence. It is likely that HCAI will play a critical role in the formation of technologies (algorithms, sensors, interfaces, and interaction paradigms) that support the driver's role in monitoring the AI system as it performs anywhere from just over 0% to just under 100% of the basic driving and higher order object and event detection tasks.

The MIT Autonomous Vehicle Technology (MIT-AVT) study seeks to collect and analyze large-scale naturalistic data of semi-autonomous driving in order to better characterize the state of current technology use, to extract insight on how automation-enabled technologies impact human-machine interaction across a range of environments, and to understand how we design shared autonomy systems that save lives as we transition from manual control to full autonomy in the coming decades. The effort is motivated by the need to better characterize and understand how drivers are engaging with advanced vehicle technology [9]. The goal is to propose, design, and build systems grounded in this understanding, so that shared autonomy between human and vehicle AI does not lead to a series of unintended consequences [10].

"Naturalistic driving" refers to driving that is not constrained by strict experimental design and a "naturalistic driving study" (NDS) is generally a type of study that systematically collects video, audio, vehicle telemetry, and other sensor data that captures various aspects of driving for long periods of time, ranging from multiple days to multiple months and even years. The term NDS is applied to studies in which data are acquired under conditions that closely align with the natural conditions under which drivers typically drive "in the wild." Often, a driver's own vehicle is instrumented (as unobtrusively as possible) and each driver is asked to continue using their vehicle as they ordinarily would. Data is collected throughout

periods of use. Further, use is unconstrained by any structured experimental design. The purpose is to provide a record of natural behavior that is as unaffected by the measurement process as possible. This contrasts with on-road experiments that are conducted in similarly instrumented vehicles, but in which experimenters are present in the vehicle, and ask drivers to carry out specific tasks at specific times on specific roads using specific technology systems in the vehicle.

The MIT-AVT study is a new generation of NDS that aims to discover insights and understanding of real-world interaction between human drivers and autonomous driving technology. Our goal is to derive insight from large-scale naturalistic data being collected through the project to aid in the design, development and delivery of new vehicle systems, inform insurance providers of the changing market for safety, and educate governments and other non-governmental stakeholders on how automation is being used in the wild.

This paper outlines the methodology and underlying principles governing the design and operation of the MIT-AVT study vehicle instrumentation, data collection, and the use of deep learning methods for automated analysis of human behavior. These guiding principles can be summarized as follows:

- **Autonomy at All Levels:** We seek to study and understand human behavior and interaction with every form of advanced vehicle technology that assists the driver through first sensing the external environment and the driver cabin, and then either controlling the vehicle or communicating with the driver based on the perceived state of the world. These technologies include everything from automated emergency braking systems that can take control in rare moments of imminent danger to semi-autonomous driving technology (e.g., Autopilot) that can help control the lateral and longitudinal movements of the vehicle continuously for long periods of driving on well-marked roadways (e.g., highways).
- **Beyond Epochs and Manual Annotation:** Successful large-scale naturalistic driving studies of the past in the United States [11], [12], [13], [14], [15] and in Europe [16] focused analysis on crash and near-crash epochs. Epochs were detected using traditional signal processing of vehicle kinematics. The extraction of driver state from video was done primarily with manual annotation. These approaches, by their nature, left the vast remainder of driving data unprocessed and un-analyzed. In contrast to this, the MIT-AVT study seeks to analyze the "long-tail" of shared-autonomy from both the human and machine perspectives. The "long-tail" is the part of data that is outside of short, easily-detectable epochs. It is, for example, the data capturing moment-to-moment allocation of glance over long stretches of driving (hundreds of hours in MIT-AVT) when the vehicle is driving itself. Analyzing the long-tail data requires processing billions of high-definition video frames with state-of-the-art computer vision algorithms multiple times as we learn both what to look for and how to interpret what we find. At the same time, despite the focus on deep learning based analysis

of large-scale data, the more traditional NDS analytic approaches remain valuable, including manual annotation, expert review of data, insight integration from technology suppliers, and contextualizing observed naturalistic behavior with driver characteristics, understanding, and perceptions of vehicle technology.
- **Multiple Study Duration:** We seek understanding human behavior in semi-autonomous systems both from the long-term perspective of over 1 year in subject-owned vehicles and from a medium-term perspective of 1 month in MIT-owned vehicles. The former provides insights into use of vehicle technology over time and the latter provides insights about initial interactions that involve learning the limitations and capabilities of each subsystem in a fashion more closely aligned with a driver's experience after purchasing a new vehicle equipped with a suite of technology that the driver may or may not be familiar with.
- **Multiple Analysis Modalities:** We use computer vision to extract knowledge from cameras that look at the driver face, driver body, and the external driving scene, but we also use GPS, IMU, and CAN bus data to add rich details about the context and frequency of technology use. This data is further complemented by detailed questionnaire and interview data that comprise driver history, exposure to various automated and non-automated technologies, mental model evaluation, perceptions of safety, trust, self-reported use, and enjoyment. With this interdisciplinary approach, the dataset allows for a holistic view of real-world advanced technology use, and identifies potential areas for design, policy, and educational improvements.

The key statistics about the MIT-AVT study as a whole and about the individual vehicles in the study are shown in Fig. 1. The key measures of the data with explanations of the measures are as follows:

- **Study months to-date:** 30
  (Number of months the study has been actively running with vehicles on the road.)
- **Participant days:** 11,846
  (Number of days of active data logger recording across all vehicles in the study.)
- **Drivers:** 99
  (Number of consented drivers across all vehicles in the study.)
- **Vehicles:** 29
  (Number of vehicles in the study.)
- **Miles driven:** 405,807
  (Number of miles driven.)
- **Video frames:** 5.5 billion
  (Number of video frames recorded and processed across all cameras and vehicles in the study.)

Latest dataset statistics can be obtained at http://hcai.mit.edu/avt (see §VI). The growth of the dataset over time is shown in Fig. 4. Data collection is actively on-going.
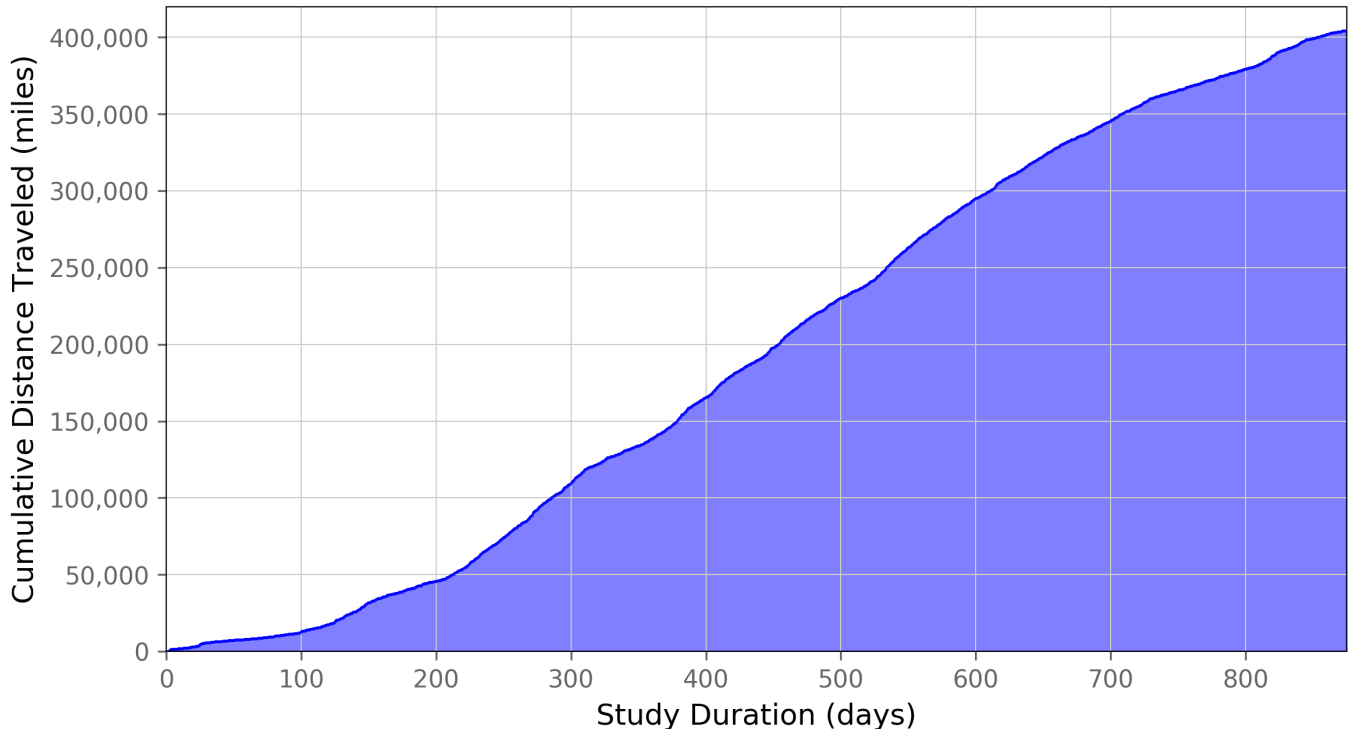
Fig. 4: Cumulative distance traveled by vehicles instrumented as part of the study. The plot shows miles traveled in the first 875 days. However, the study is ongoing and hundreds of miles worth of data is being collected every day.

A. Naturalistic Driving Studies

The focus of the MIT-AVT study is to gather naturalistic driving data and to build on the work and lessons-learned of the earlier generation of NDS studies carried out over the first decade of the 21st century [11], [12], [13], [14], [15]. These previous studies aimed to understand human behavior right before and right after moments of crashes and near-crashes as marked by periods of sudden deceleration. The second Strategic Highway Research Program (SHRP2) is the best known and largest scale of these studies [14].

In contrast to SHRP-2 and other first-generation NDS efforts, the MIT-AVT study aims to be the standard for the next generation of NDS programs where the focus is on large-scale computer vision based analysis of human behavior. Manually annotating specific epochs of driving, as the prior studies have done, is no longer sufficient for understanding the complexities of human behavior in the context of autonomous vehicle technology (i.e., driver glance or body position over thousands of miles of Autopilot use). For example, one of many metrics that are important to understanding driver behavior is moment-by-moment detection of glance region [17], [18] (see §I-C). In order to accurately extract this metric from the 1.7 billion frames of face video without the use of computer vision would require an immense investment in manual annotation, assuming the availability of an efficient annotation tool that is specifically designed for the manual glance region annotation task and can leverage distributed, online, crowdsourcing of the annotation task. The development of such a tool is a technical challenge that may take several years of continuous research and development [19], which may eclipse the cost human annotation hours. If this was the only metric of interest, perhaps such a significant investment would be justifiable and feasible. However, glance region is only one of many metrics of interest, and in terms of manual annotation cost, is one of the least expensive. Another example is driving scene segmentation, which for 1.7 billion frames would require an incredible investment [20]. For this reason, automatic or semi-automatic extraction of information from raw video is of paramount importance and is at the core of the motivation, design, research, and operation of MIT-AVT.

The fundamental belief underlying our approach to NDS is that only by looking at the entirety of the data (with algorithms that reveal human behavior and situation characteristics) can we begin to learn which parts to "zoom in" on: which triggers and markers will lead to analysis that is representative of system performance and human behavior in the data [21], [22], [23], [24], [25]. Furthermore, each new insight extracted from the data may completely change our understanding of where in the data we should look. For this reason, we believe understanding how humans and autonomous vehicles interact requires a much larger temporal window than an epoch of a few seconds or even minutes around a particular event. It requires looking at the long-tail of naturalistic driving that has up until now been largely ignored. It requires looking

5

at entire trips and the strategies through which humans engage the automation: when, where, and for how long it is turned on, when and where it is turned off, when control is exchanged, and many other questions. Processing this huge volume of data necessitates an entirely different approach to data analysis. We accomplish the automated aspect of the knowledge extraction process by using deep learning based computer vision approaches for driver state detection, driver body pose estimation, driving scene segmentation, and vehicle state detection from the instrument cluster video as shown in Fig. 2 and discussed in §IV. The result of using deep learning based automated annotation is that MIT-AVT can analyze the long-tail of driving in the context of shared autonomy, which in turn, permits the integration of complex observed interactions with the human's perception of their experience. This innovative interdisciplinary approach to analysis of NDS datasets in their entirety offers a unique opportunity to evaluate situation understanding of human-computer interaction in the context of automated driving.

B. Datasets for Application of Deep Learning

Deep learning [26] can be defined in two ways: (1) a branch of machine learning that uses neural networks that have many layers or (2) a branch of machine learning that seeks to form hierarchies of data representation with minimum input from a human being on the actual composition of the hierarchy. The latter definition is one that reveals the key characteristic of deep learning that is important for our work, which is the ability of automated representation learning to use large-scale data to generalize robustly over real-world edge cases that arise in any in-the-wild application of machine learning: occlusion, lighting, perspective, scale, inter-class variation, intra-class variation, etc. [27].

In order to leverage the power of deep learning for extracting human behavior from raw video, large-scale annotated datasets are required. Deep neural networks trained on these datasets can then be used for their learned representation and then fine-tuned for the particular application in the driving context. ImageNet [28] is an image dataset based on WordNet [29] where 100,000 synonym sets (or "synsets") each define a unique meaningful concept. The goal for ImageNet is to have 1000 annotated images for each of the 100,000 synsets. Currently it has 21,841 synsets with images and a total of 14,197,122 images. This dataset is commonly used to train neural network for image classification and object detection tasks [30]. The best performing networks are highlighted as part of the annual ImageNet Large Scale Visual Recognition Competition (ILSVRC) [31]. In this work, the terms "machine learning," "deep learning," "neural networks," and "computer vision" are often used interchangeably. This is due to the fact that the current state-of-the-art for most automated knowledge extraction tasks are dominated by learning-based approaches that rely on one of many variants of deep neural network architectures. Examples of other popular datasets leveraged in the development of algorithms for large-scale analysis of driver behavior in our dataset include:

- **COCO** [32]: Microsoft Common Objects in Context (COCO) dataset is a large-scale dataset that addresses the object detection task in scene understanding under two perspectives: detecting non-iconic views of objects, and the precise 2D localization of objects. The first task usually refers to object localization, which uses bounding boxes to denote the presence of objects. The second task refers to instance segmentation, for which the precise masks of objects are also needed. The whole dataset features over 200,000 images labeled within 80 object categories. Successful methods [30], [33], [34] jointly model the two tasks together and simultaneously output bounding boxes and masks of objects.
- **KITTI** [35], [36]: KITTI driving dataset develops challenging benchmarks for stereo vision, optical flow, visual odometry / SLAM and 3D object detection, captured by driving around in both rural areas and highways of Karlsruhe (a mid-size city in Germany). In total, there are 6 hours of traffic scenarios recorded at 10-100 Hz using a variety of sensor modalities such as high-resolution color and grayscale stereo cameras, a Velodyne 3D laser scanner and a high-precision GPS/IMU inertial navigation system. In addition, [37] also propose ground truth for 3D scene flow estimation by collecting 400 highly dynamic scenes from the raw dataset and augmenting them with semi-dense scene flow ground truth.
- **Cityscapes** [38]: The Cityscapes dataset focuses on semantic understanding of urban street scenes. It offers a large, diverse set of stereo video sequences recorded in streets from 50 different cities with pixel-level and instance-level semantic labeling. There are 5,000 fully segmented images with pixel-level annotations and an additional 20,000 partially segmented images with coarse annotations. Its two benchmark challenges have led to the development of many successful approaches for semantic segmentation [39], [40] and instance segmentation [33], [41].
- **CamVid** [42]: Cambridge-driving Labeled Video Database (CamVid) is the first dataset with frame-wise semantic labels in videos captured from the perspective of a driving automobile. The dataset provides ground truth labels that associate each pixel with one of 32 semantic classes. Manually specified per-pixel semantic segmentation of over 700 images total enables research on topics such as pedestrian detection [43], and label propagation [44].

C. Automotive Applications of Deep Learning

Design of perception and control systems in the driving domain have benefited significantly from learning-based approaches that leverage large-scale data collection and annotation in order to construct models that generalize over the edge cases of real-world operation. Leveraging the release large-scale annotated driving datasets [35], [38], automotive deep learning research aims to address detection, estimation, prediction, labeling, generation, control, and planning tasks.

As shown in Fig. 2, specific tasks have been defined such as fine-grained face recognition, body pose estimation, semantic scene perception, and driving state prediction. Current efforts are briefly summarized as follows:

- **Fine-grained Face Recognition**: Beyond classic face recognition studies, fine-grained face recognition focuses on understanding human behavior toward face perception, such as facial expression recognition [45], [46], eye gaze detection [47], [48]. In the driving context, [49], [50] explore the predictive power of driver glances. [51], [52] use facial expression to detect emotional stress for driving safety and the driving experience.
- **Body Pose Estimation**: Work on human body pose expands the performance, capabilities, and experience of many real-world applications in robotics and action recognition. Successful approaches vary from using depth images [53], via deep neural networks [54], or with both convolutional networks and graphical models [55]. Specifically for driving, [56] use driver pose, which is represented by skeleton data including positions of wrist, elbow, and shoulder joints, to model human driving behavior. [57] cast visual analysis of eye state and head pose for driver alertness monitoring.
- **Semantic Scene Perception**: Understanding the scene from 2D images has long been a challenging task in computer vision, which often refers to semantic image segmentation. By taking advantage of large scale datasets like Places [58], Cityscapes [38], many approaches [39], [40] manage to get state-of-the-art results with powerful deep learning techniques. As a result, precise driving scene perception [59], [60] for self-driving cars is now actively studied in both academia and industry.
- **Driving State Prediction**: Vehicle state is usually considered as a direct illustration of human decision in driving, which is also the goal for autonomous driving. In terms of machine learning, it serves as the ground truth for various tasks from different perspectives such as driving behavior [56] and steering commands [59], [60].

Many aspects of driver assistance, driver experience, and vehicle performance are increasingly being automated with learning-based approaches as representative datasets for these tasks are released to the broad research community. The MIT-AVT study aims to be the source of many such datasets that help train neural network architectures that provide current and future robust solutions for many modular and integrated subtasks of semi-autonomous and fully-autonomous driving.

II. MIT-AVT STUDY STRUCTURE AND GOALS

The governing principle underlying the design of all hardware, low-level software, and higher-level data processing performed in the MIT-AVT study is: continual, relentless innovation, while maintaining backward compatibility. From the beginning, we chose to operate at the cutting-edge of data collection, processing, and analysis approaches. This meant trying a lot of different approaches and developing completely new ones: from sensor selection and hardware design described in §III to the robust time-critical recording system and the highly sophisticated data pipeline described in §IV. It's a philosophy that allowed us to scale quickly and find new solutions at every level of the system stack.

A. Participation Considerations and Recruitment

As previously noted, the medium duration (one month long) NDS is conducted using MIT-owned vehicles, while the long duration (over 1 year) NDS is conducted in subject-owned vehicles. Participants are divided into primary and secondary drivers, all of whom, in order to take part in the study, must formally agree to the terms detailed in an informed consent form approved by an institutional review board (IRB). Primary drivers in the long NDS (usually the most frequent driver of the vehicle or the car owner) must be willing to provide permission to install the data acquisition equipment in the vehicle, warning labels on windows to advise non-consented passengers and drivers of the ongoing data collection, and coordinate with project staff for system maintenance and data retrieval. Recruitment is conducted through flyers, social networks, forums, online referrals, and word of mouth. Primary drivers are compensated for their time involvement in vehicle instrumentation, system maintenance appointments, data retrieval, and completing questionnaires.

To be accepted as a primary driver in an MIT-owned vehicle fleet requires that potential subjects' daily commutes include time on specific highways, a willingness to use a study vehicle for a period of approximately four weeks as the subject's primary commuting vehicle, signing an IRB approved informed consent form, passing a Criminal Offender Record Information (CORI) check and driving record review by MIT's Security and Emergency Management Office, participating in a training protocol that covers both basic and advanced vehicle features, and completing a series of questionnaires and interviews prior to and after their naturalistic driving experience. High-level overviews of the training protocol, questionnaire, and interview strategies can be found in §II-B and §II-C, respectively.

B. Training Conditions for One Month NDS

Participants in the medium duration (one month long) NDS are provided with introductions to the fleet vehicles in the form of an approximately 1.5 hour long training session. This session is intended to introduce drivers to the physical characteristics of the vehicle, and provide a sufficient understanding of vehicle features in order to support safe use of advanced technologies. Participants are provided with a study overview by a researcher and presented with manufacturer produced videos or information packets on one or more of the basic and advanced features available in the vehicle. After this initial introduction to systems outside of the vehicle, participants are seated in the vehicle and given a guided overview of the vehicle layout and settings (e.g. seat / mirror adjustments, touchscreen menu layout). Participant's phones are paired with the vehicle, and they are given the opportunity to practice several voice commands (e.g. placing a phone call, entering

a destination). Next, more detailed overviews are provided on the function, activation, and use of the following features:

- Adaptive Cruise Control (ACC)
- Pilot Assist (in the Volvo)
- Super Cruise (in the Cadillac)
- Forward Alert Warning / City Safety (in the Volvo)
- Automatic Emergency Braking
- Lane Departure Warning (LDW)
- Lane Keep Assist (LKA)
- Blind Spot Monitor

Following this stationary in-vehicle training, participants are provided with an on-road training drive on a multi-lane highway. This highway driving session lasts a minimum of 30 minutes to allow for practical exposure to the systems in real world setting. During the training drive participants are encouraged to utilize the researcher and ask questions when testing out the systems. Participants are encouraged to customize vehicle settings to their preferences and to develop sufficient familiarity to support the ability to choose to use or not use certain systems for the duration of their one month period of vehicle use.

C. Qualitative Approaches for One Month NDS

Self-report data collection methods are kept as unobtrusive to participation in the study as possible, while still capturing the richness of driver's experience with the vehicle and various systems, their thoughts on the technology after participating, and barriers toward their intentions to adopt or discard automation moving forward. Self-report data in the medium duration (one month long) NDS is captured using three questionnaire batteries and two semi-structured interviews. Self-report data is collected prior to and after the naturalistic portion of the experiment; at no point are participants asked to complete questionnaires or interviews while they are in possession of the vehicle.

The questionnaire batteries are deployed in three stages. The first occurs when a subject signs the consent form and completes the background check paperwork. The first questionnaire collects basic demographics and information on driving history, driving style, exposure to various advanced and established in-vehicle technologies, and general trust in technology. A second questionnaire is completed immediately following the training protocol outlined in §II-B, and captures participants' high level mental models, initial impressions, and reported trust in select vehicle technologies. The third and final questionnaire is completed at the end of the driver's one-month naturalistic driving period. This questionnaire assesses reported trust in select technologies, perceptions of safety, high- and detailed-level understanding of systems, and desire for having in their own future vehicles such systems as experienced during the NDS period and with hypothetical improvements. Many questions in the second and third questionnaires are identical, allowing analysis to explore how exposure to systems and experiential learning impact concepts such as trust and understanding of technologies.

A semi-structured interview is conducted in person between a research associate and the study participant at the end of the one-month naturalistic driving period, and lasts approximately 30-60 minutes. It consists of pre-defined questions focusing on initial reactions to the vehicle, experience during the training drive, how training affected their understanding of the technologies, and driver perceptions of the technologies.

D. Competitors Collaborate: Consortium Model

Naturalistic driving data and automated deep learning based interpretation of that data gives insights, suggestions, and well-grounded scenarios as to the path forward for safe and effective integration of artificial intelligence into modern and future vehicle systems. The raw data and the high-level understanding of human behavior and system performance in such autonomous vehicle technology is of interest to:

- Car companies (both established and newly formed)
- Automotive parts suppliers
- Insurance companies
- Technology companies
- Government agencies
- Academic and research organization

When the path forward is full of uncertainty, risks, potentially costly misaligned investments, and paradigm shifts, open innovation provides more value than closed competition. At this moment in time, autonomous vehicle technology is a space where competitors win by collaborating, sharing high-level insights and large-scale, real-world data.

High-level measures such as system use and system performance can be used to inform the design, development and validation of future vehicle systems. Basic driver behavior with and without technology use can fuel basic research on driver understanding, use characteristics, and decision models while aiding in the actuation of risk in the insurance market. Video recording inside and out of the vehicle can be used to develop perception, control, planning, driver sensing, and driver assistance systems. As such, the data collected in the MIT-AVT study can be leveraged for a range of quantitative and qualitative efforts. Members of the Advanced Vehicle Technology consortium [61] are collaborating to support the acquisition of data through the MIT-AVT study, development of new data processing approaches, and selected analysis. Full members of the consortia have rights to data access for proprietary or other internal use purposes. Several members of the effort are actively involved in independent research (with and without MIT involvement) using MIT-AVT study data.

III. Hardware: Data Logging and Real-Time Processing

The backbone of a successful naturalistic driving study is the hardware and low-level software that performs the data collection. In the MIT-AVT study, that role is served by a system named RIDER (Real-time Intelligent Driving Environment Recording system). RIDER was designed and continuously developed to satisfy the following goals and requirements:

1) **Timestamped Asynchronous Sensor Recording:** Record all sensors and data streams in a way that each sample of data (no matter its frequency or data source) is timestamped using a centralized, reliable time-keeper. In other words, data has to be timestamped in a way that allows perfect synchronization of multiple data streams in post-processing [62].

2) **High-Definition Video:** Capture and record 3 to 6 cameras at 720p (2.1 megapixels) resolution. The selection of camera positions, resolution, and compression was one of the most essential design decisions of the entire study. See §III-C for discussion of how this selection was made.

3) **CAN Bus:** Collect vehicle telemetry from the Controller Area Network (CAN) bus(es) of the vehicle [63]. Each vehicle has different ports and bus utilization policies, with little information made publicly available about the mapping of message ID's and the message content. Raw CAN messages must be recorded such that the essential information is contained within those messages even if at the time of collection those messages cannot be decoded.

4) **Remote Cellular Connectivity:** Low-bandwidth, infrequent communication of system status via a cellular connection in order to detect when RIDER system malfunction occurs.

5) **Discrete and Elegant Appearance:** Parts of the system that are visible from inside or outside the car should have a small form-factor and have visual design characteristics that do not detract from the overall appearance of the vehicle or have an impact on the overall driving experience.

6) **Camera Mounting is Robust but Removable:** Mounting must be consistent, reliable, and removable designed specifically for each vehicle's interior physical characteristics.

RIDER components include a real-time-clock, GPS, IMU, and the ability to record up to 6 cameras at 720p resolution, remote cellular connectivity. The developed system employs the use of common components tailored to suit its needs achieving a scalable ultra low cost, accurate, extendable and robust data recording platform.

To keep the electronics and stored data secure, RIDER is placed within in the trunk away from the elements and possible disturbances from passengers. Power and CAN data cables are run from the OBD-II or diagnostic port to the trunk into RIDER. USB cables for cameras are also run from each camera location into the trunk. All data and power cables are secured and hidden beneath interior trim panels.

A. Power Management System

The power systems for RIDER has many constraints: it demanded flexibility to transfer into different vehicles and draw minimal power when not in use as to not drain the primary vehicle battery. The power system consists of a main smart CAN monitoring section and a buck converter. When active and logging data, RIDER draws less than 8 watts of
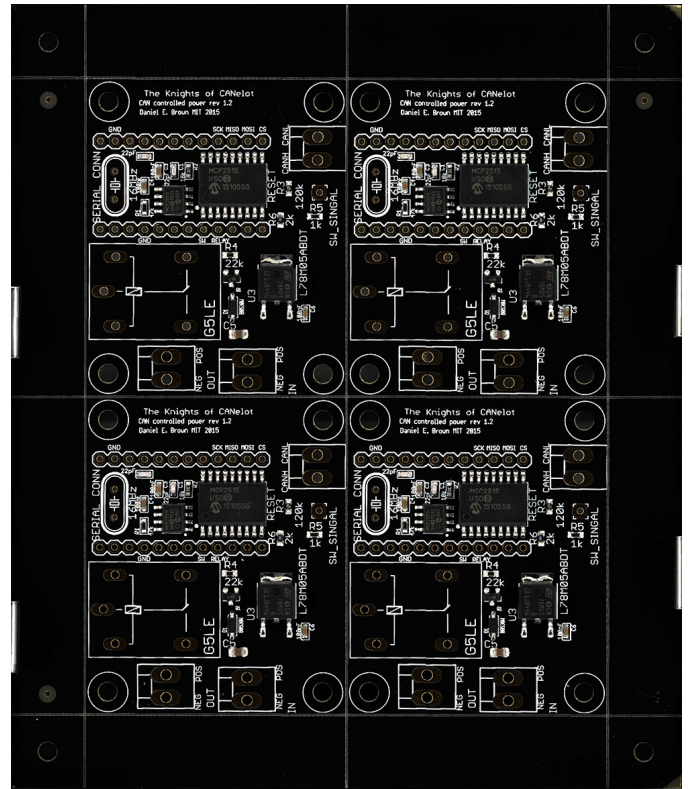

Fig. 5: Knights of CANelot, CAN controlled power board. Power board mid-assembly showing populated CAN controller, transceiver, and power regulation. Also shown, unpopulated positions for the power relay, microcontroller, oscillator and connectors.

power. When in standby, RIDER's quiescent current draw is less than 1/10th of a watt.

The Knights of CANelot is a CAN controlled power board that contains a microchip MCP2515 CAN controller and MCP2551 CAN transceiver, along with an Atmega328p microcontroller to monitor CAN bus traffic. By default when powered this microcontroller places itself into sleep and does not allow power to enter the system by way of a switching relay. When the CAN controller detects a specific predefined CAN message indicating the vehicle CANbus is active, the microcontroller is sent an interrupt by the CAN controller waking up the microcontroller from sleep and triggering the relay to power the primary buck converter. This begins the booting sequence to the rest of the system. When the vehicle shuts off and the CANbus within the car enters into a sleep state, a signal is sent via the Knights of CANelot microcontroller to gracefully stop all video and data recording, shutdown the compute system, disconnect main power then enter sleep mode once again.

B. Computing Platform and Sensors

A single board computer was chosen for this application for its wide variety of I/O options, small form factor and ease of
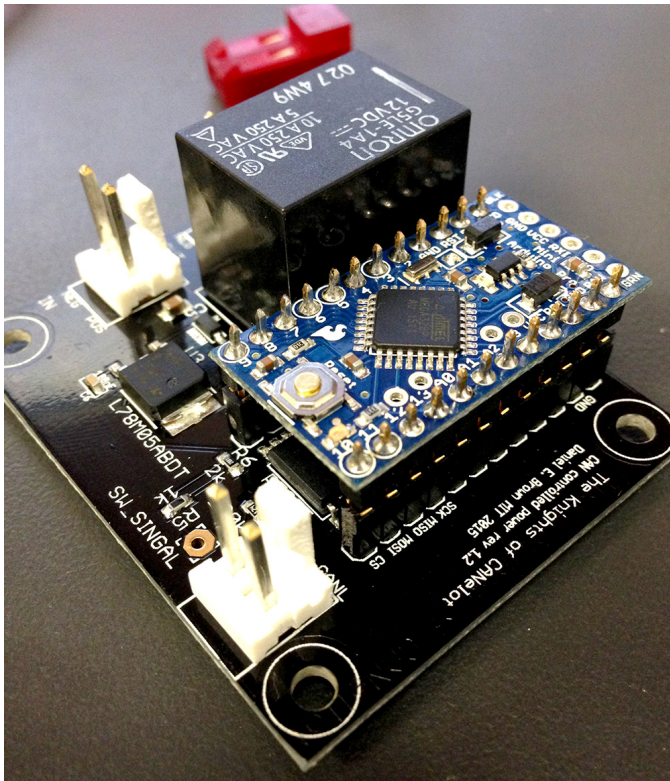
Fig. 6: Fully assembled Knights of CANelot board, showing populated microcontroller, power relay, CAN and power connections.
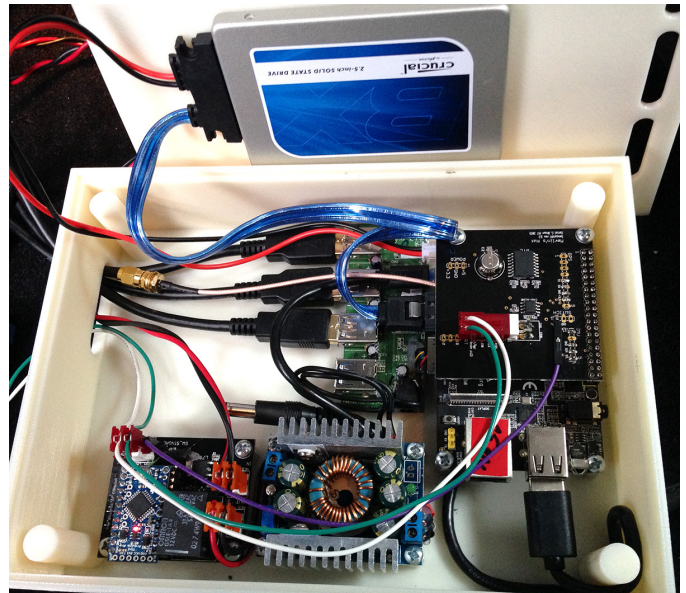

Fig. 7: Final prototype version of RIDER enclosed by 3D printed case. From top to bottom, clockwise, attached to the top of the case is external storage in the form of a 1 terabyte solid state hard drive. The USB cameras connect via a USB hub shown in the center. To the right of the USB hub, Banana Pi covered by the black SensorHAT with CAN transceiver, GPS, IMU, and real time clock. Bottom center, buck converter for stepping down vehicle battery voltage from 12-13.8 volts to 5 volts for all compute systems. Lower left, Knights of CANelot CAN controlled power board.

development. We chose to work with the Banana Pi Pro with the follow sensors and specifications:

- 1GHz ARM Cortex-A7 processor, 1GB of RAM
- Expandable GPIO ports for IMU/GPS/CAN
- Native onboard SATA
- Professionally manufactured daughter board for sensor integration
- ARM processor features onboard CAN controller for vehicle telemetry data collection
- Maxim Integrated DS3231 real-time clock for accurate timekeeping/time-stamping +/-2 ppm accuracy
- Texas Instruments SN65HVD230 CAN transceiver
- 9 degrees-of freedom inertial measurement unit (STMicro L3GD20H(gyro), LSM303D(accelerometer/compass))
- GlobalTop MTK3339 GPS unit, 6 channel, DGPS capability accurate within 5 meters
- Huawei E397Bu-501 4G LTE USB module
- USB 3.0 4-port hub, powered
- 1TB/2TB solid state hard drive

C. Cameras

Three or four Logitech C920 webcams record at a resolution of 1280x720 at 30 frames per second within the car. Two of these cameras have been modified to accept standard CS type lens mount for adaptability within the car for either face or body pose orientation. The third camera is the standard webcam that is mounted on the windshield for a forward road perspective. Occasionally a fourth camera is placed within the instrument cluster to capture information unavailable on the CANbus. These cameras also contain microphones for audio capture and recording. Custom mounts were designed for specialty placement within the vehicle.

Most single board computers like our Banana Pi lack the required computational ability to encode and compress more than one raw HD video stream. The Logitech C920 camera provides the ability to off-load compression from the compute platform and instead takes place directly on the camera. This configuration allows for possibility of up to 6 cameras in a single RIDER installation.

D. Ongoing Hardware Development and Innovation

RIDER is a strong and proven instrumentation platform with adequate data collection abilities for naturalistic driving research. During the research, development, and testing process we met some limitations of the system. While a single board computer is sufficient for most collection processes, limitations of minimal system memory could create issues when expanding the system. Similarly, a Dual-Core ARM processor is very capable when interfacing with sensors and writing data out to files, but performance can fluctuate if any preprocessing of the data is required onboard. From our work

we have proposed the following improvements to some of these common issues.

The largest enhancement for the entire RIDER system would be to upgrade the single board computing platform. Use of the Nvidia Jetson TX2 would provide more expandability both for I/O and processing. With greater processing and GPU bandwidth available, real-time systems could be implemented using both video and sensor data simultaneously for detection and driver warning systems, internal annotation of data and more. With greater I/O capability, upgraded sensors packages with higher data bandwidths can be implemented. Much like the Banana Pi Pro the Jetson TX2 has not one, but two fully supported CAN controllers to interface with a secondary CANbus system on the vehicle. Jetson TX2 has expandability not only for SATA but also PCIe and mSATA, allowing for even greater expansion of third party modules. The enhanced processing via CPU and GPU with 8 times the onboard RAM allows the potential for preprocessing and integration of real-time driver monitoring systems. The Jetson also has the major advantage of being supported for use in multiple configurations for in vehicle applications. Below are the specifications and added improvements of the Jetson TX2 over the Banana Pi.

1) Jetson TX2 - Capabilities:
- Quad-Core ARM Cortex-A57 @ 2GHz + Dual-Core NVIDIA Denver 2 @ 2GHz + NVIDIA Pascal GPU with 8GB of RAM
- Industrial cameras; GigE vision, USB3 Vision
- LIDAR
- Onboard video encoding
- Real-time
- 802.11a/b/g/n/ac WiFi
- Bluetooth 4.1
- USB3
- 10/100/1000 BASE-T Ethernet
- 12 lanes MIPI CSI 2.0, 2.5 Gb/sec per lane
- PCIe gen 2.0
- mSATA
- SDcard
- 2x CAN bus interface

Much like the Jetson TX2, Nvidia's Drive PX2 was specifically designed for the automotive environment to operate semi- and fully autonomous vehicles. Its well equipped for simple tasks such as data logging, up to advanced situations where interfacing with many cameras, high bandwidth sensors and the ability to control both longitudinal and lateral movement of the vehicle are required. While the Drive PX2 would be under utilized in a data-collection-only scheme, it is a great platform to begin developing systems for final integration into automobiles as a production system. Below are some of the Drive PX2 compute and sensor capabilities.

2) Drive PX2 - Capabilities:
- 2x Quad-Core ARM Cortex-A57 @ 2GHz + 2x Dual-Core NVIDIA Denver 2 @ 2GHz with 8GB of LPDDR4 RAM + 2x integrated Pascal + 2x dedicated Pascal MXM modules with 4GB of GDDR5 VRAM

- Two computers, connected over ethernet
- 8 TFLOPS, 24 DL TOPS (using int8)
- 6x CANbus interfaces
- 4x LINbus interfaces
- 2x FlexRay network interfaces
- Ability to control car with real time system
- 12x GSML camera (industrial cameras)
- 2x 1GbE + 1x 10GbE (ethernet)
- USB3

A customized Linux kernel was developed specifically for RIDER based on the hardware and system requirements listed above. The filesystem is stored on a replaceable micro SD card on-board the Banana Pi. The next section outlines and describes the software running within RIDER and the management of the data after collection.

IV. Software: Data Pipeline and Deep Learning Model Training

Building on the robust, reliable, and flexible hardware architecture of RIDER is a vast software framework that handles the recording of raw sensory data and takes that data through many steps across thousands of GPU-enabled compute cores to the extracted knowledge and insights about human behavior in the context of autonomous vehicle technologies. Fig. 8 shows the journey from raw timestamped sensor data to actionable knowledge. The high-level steps are (1) data cleaning and synchronization, (2) automated or semi-automated data annotation, context interpretation, and knowledge extraction, and (3) aggregate analysis and visualization.

This section will discuss the data pipeline (Fig. 8), which includes software implemented on RIDER boxes that enables data streaming and recording. In addition, the software that is used to offload and process the data on a central server will be discussed. The operational requirement of software operating on RIDER boxes are as follows:

1) Power on whenever the vehicle is turned on
2) Create a trip directory on an external solid state drive
3) Redirect all data streams into timestamped trip files
4) Log and transmit metadata to the lab in real time
5) Power down after the vehicle is turned off

A. Microcontroller

The microcontroller on the Knights of CANelot power management board runs a small C program that is responsible for powering the RIDER system in sync with the vehicle. By default, this microcontroller is in a sleep state, awaiting a specfic CAN message. By listening to the vehicle's CANbus, this program can recognize when CAN message for a specific signal begins, which signifies the car has turned on. If this signal is observed, the C program then connects the vehicle's power to the rest of the system, starting the data collection. When the specified message ends, meaning the car is off, the microcontroller sends a signal to the Banana Pi to close all files and shutdown gracefully. It then waits 60 seconds to finally disconnect power from the rest of the system and enters its original sleep state.
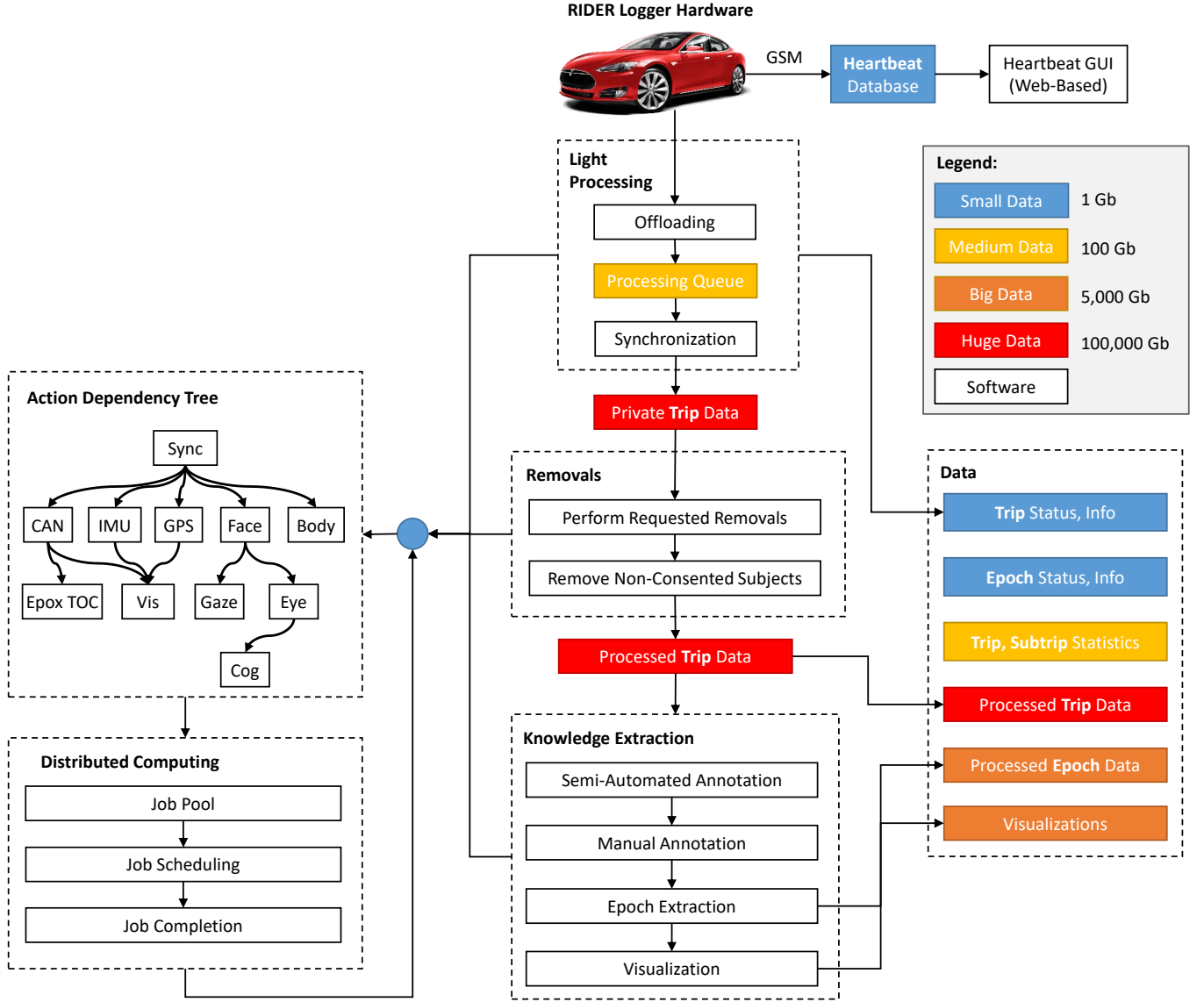
Fig. 8: The MIT-AVT data pipeline, showing the process of offloading, cleaning, synchronizing, and extracting knowledge from data. On the left is the dependency-constrained, asynchronous, distributed computing framework. In the middle is the sequence of high level procedures that perform several levels of knowledge extraction. On the right are broad categories of data produced by the pipeline, organized by size.

B. Single Board Computer

Our single board computer, the Banana Pi, contains a 32GB SD card that stores the RIDER filesystem, software and configuration files. The Banana Pi runs a modified Linux kernel using custom kernel modules and a tweaked Bannanian operating system with performance and security enhancements. Performance was improved by disabling unnecessary kernel modules and removing extraneous Linux services. Security enhancements included disabling all CAN transmission, thereby prohibiting malicious or unintentional transmission of actuating messages to a vehicle's systems. Additional security improvements included altering the network settings

to prevent any remote connection from logging in. Specific MIT machines were white listed to allow configuration files to be altered through a physical connection. The default system services were also altered to run a series of locally installed programs that manage data collection whenever the system boots.

C. Startup Scripts

The Banana Pi runs a series of data recording initialization bash startup scripts whenever the system boots. First, the on-board clock on the Pi is synchronized with a real-time clock that maintains high resolution timing information. Modules for device communication such as UART, I2C, SPI, UVC, and

CAN are then loaded to allow interaction with incoming data streams. A monitoring script is started that shuts down the system if a specified signal is received from the Knights of CANelot microcontroller, and an additional GSM monitoring script helps reconnect to the cellular network after losing connection. The last initialization steps are to start the python scripts Dacman and Lighthouse.

D. Dacman

Dacman represents the central data handler script that manages all data streams. It uses a configuration file called `trip_dacman.json` that contains unique device IDs for all cameras. In addition, it contains a unique RIDER ID associated with the RIDER box it is stored in. This configuration file also contains unique ID values for the subject, vehicle and study this driver is associated with. Once started, Dacman creates a trip directory on the external solid state drive named according to the date it was created using a unique naming convention: `rider-id_date_timestamp` (e.g. `20_20160726_1469546998634990`). This trip directory contains a copy of `trip_dacman.json`, any data related CSV files reflecting included subsystems, as well as a specifications file called `trip_specs.json` that contains microsecond timestamps denoting the beginning and end of every subsystem and the trip itself.

Dacman calls a manager python script for every subsystem (e.g. `audio_manager.py` or `can_manager.py`), which makes the relevant system calls to record data. Throughout the course of the current vehicle trip, all data is written to CSV files with timestamping information included in each row. Dacman calls two other programs written in C in order to help generate these files: `cam2hd` for managing cameras and `dump_can` for creating CAN files. Audio or camera data is recorded to RAW and H264 formats respectively, with an accompanying CSV denoting the microsecond timestamp at which each frame was recorded. If any errors are encountered while Dacman is running, the system restarts up to two times in an attempt to resolve them, and shuts down if unable to resolve them.

E. Cam2HD

`Cam2hd` is a program written in C that opens and records all camera data. It relies on V4L (Video4Linux), which is an open source project containing a collection of camera drivers in Linux. V4L enables low level access to cameras connected to RIDER by setting the incoming image resolution to 720p and allows the writing of raw H264 frames.

F. DumpCAN

`Dump_can` is another program written in C that configures and receives data from the Allwinner A20 CAN controller. This program uses the can4linux module to produce a CSV containing all CAN data received from the connected CANbus. In addition, it offers low level manipulation of the CAN controller. This allows `dump_can` to set listen only mode on the can controller, which enables a heightened degree of

security. By removing the need to send acknowledgements when listening to messages on the CAN network, any possible interference with existing systems on the CAN bus is minimized.

G. Lighthouse

Lighthouse is a python script that sends information about each trip to Homebase. Information sent includes timing information for the trip, GPS data, power consumption, temperature and available external drive space. The interval between communications is specified in the dacman configuration file. All communications are sent in JSON format and are encrypted using public-key cryptography based on elliptic curve Curve25519 due to its speed. This means that each RIDER uses the public key of the server, as well a unique public/private key to encrypt and transmit data. Lighthouse is written in Python and depends on libzmq/libsodium.

H. Homebase

Homebase is a script that receives, decrypts and records all information received from Lighthouse and stores them in the RIDER database. This allows remote monitoring of drive space and system health. All RIDER key management is done here in order to decrypt messages from each unique box.

I. Heartbeat

Heartbeat is an engineer facing interface that displays RIDER system status information in order to validate successful operation or gain insights as to potential system malfunction. Heartbeat uses the information committed to the database from Homebase to keep track of various RIDER logs. This is useful for analyzing the current state of the vehicle fleet, and assists in determining which instrumented vehicles are in need of drive swaps (due to the hard drive running out of space) or system repairs. It is also useful for verifying that any repairs made were successful.

J. RIDER Database

A PostgreSQL database is used to store all incoming trip information, as well as to house information about all trips offloaded to a storage server. After additional processing, useful information about each trip can be added to the database. Queries can then be structured to obtain specific trips or times in which specific events or conditions occurred. The following tables are fundamental to the trip processing pipeline:

- **instrumentations**: dates and vehicle IDs for the installation of RIDER boxes
- **participations**: unique subject and study IDs are combined to identify primary and secondary drivers
- **riders**: rider IDs paired with notes and IP addresses
- **vehicles**: vehicle information is paired with vehicle IDs such as the make and model, the manufacture date, color, and availability of specific technologies
- **trips**: provides a unique ID for each centrally offloaded trip as well as the study, vehicle, subject and rider IDs. Also provides information about synchronization state,

available camera types and subsystem data. Metadata about the content of the trip itself is included, such as the presence of sun, gps frequency and the presence of certain technology uses or acceleration events.

- **epochs_*epoch-label***: tables for each epoch type are labeled and used to identify trips and video frame ranges for which they occur (e.g. autopilot use in Teslas would be in epochs_autopilot)
- **homebase_log**: contains streamed log information from the homebase script that keeps track of RIDER system health and state

K. Cleaning

After raw trip data is offloaded to a storage server, all trips must be inspected for any inconsistencies. Some trips may have inconsistencies that can be fixed, as in the case where timestamping information could be obtained from multiple files, or when a nonessential subsystem failed during a trip (e.g. IMU or audio). In unrecoverable cases, like the event where a camera was unplugged during a trip, that trip is removed from the dataset. Trips that have valid data files may also be removed from the dataset if that trip meets some set of filtering constraints, like when a vehicle is turned on, but does not move before turning off again.

L. Synchronization

After completing cleaning and filtration, valid trips undergo a series of synchronization steps. First, the timestamps of every frame gathered from every camera are aligned in a single video CSV file at 30 frames per second using the latest camera start timestamp and the earliest camera end timestamp. In low lighting conditions the cameras may drop to recording at 15 frames per second. In these cases, some frames may be repeated to achieve 30 frames per second in the synced video.

After all raw videos have been aligned, new synchronized video files can then be created at 30 frames per second. CAN data is then decoded by creating a CSV with all relevant CAN messages as columns and synced frame IDs as rows. CAN message values are then inserted frame-by-frame based on the closest timestamp to each decoded CAN message. A final synchronized visualization can then be generated that shows all video streams and CAN info in separate panels in the same video. The data is then ready to be processed by any algorithm running statistics, detection tasks, or manual annotation tasks.

V. TRIPS AND FILES

This section will define how trip data files may be stored in a trip directory. A trip directory represents a trip that a driver took with their vehicle from start to finish. These are the files that are offloaded from the external storage drive in a RIDER box onto a central server, where the data can be cleaned, synchronized, or processed in some other way.

A. Trip Configuration Files

Trip configuration files store specifications and information about available subsystems are included to manage the data logging process.

- **trip_dacman.json**: a configuration file containing subject and systems information used to record the trip
- **trip_diagnostics.log**: a text file containing diagnostics information recorded during the trip: includes external temperature, PMU temperature, HDD temperature, power usage and free disk space
- **trip_specs.json**: a json file containing start and end timestamps for all subsystems

B. Trip Data Files

Trip data files are the end point of all recording RIDER data streams. They include numerous CSV (comma separated values) files that provide timestamping information, as well as raw video files in H264 and audio files in RAW formats.

- *camera-directory*: a directory named by camera type (all contained files are also named by that camera type)
  - *camera-name***.h264**: a raw H264 file
  - *camera-name***.error**: contains camera-specific errors
  - *camera-name***.csv**: matches recorded frames with system timestamps for later synchronization
    * `frame,ts_micro`
- **data_can.csv**: contains CAN data
  - `ts_micro, arbitration_id, data_-length, packet_data`
- **data_gps.csv**: contains GPS data
  - `ts_micro, latitude, longitude, altitude, speed, track, climb`
- **data_imu.csv**: contains IMU data
  - `ts_micro, x_accel, y_accel, z_accel, roll, pitch, yaw`
- **audio.raw**: contains raw output from a specified camera
- **can.error, gps.error, imu.error, audio.error**: text-based error files for CAN, GPS, IMU and audio recordings

C. Cleaning Criteria

The following cases represent recoverable errors that a trip may contain, as well as their implemented solutions:

- **Invalid permissions**: UNIX permissions of the trip directory must allow group-only read/write access
- **Missing backup**: Raw essential files are backed up to allow a rollback to previous versions
- **Missing trip_specs.json**: The trip_specs.json file can sometimes be reconstructed using recorded timestamps
- **Missing or invalid ID**: Vehicle, camera or subject IDs may be corrected based on trip context
- **Invalid Nonessential Files**: If IMU or audio have failed, they can be removed and the trip can be preserved
- **Invalid last CSV line**: Interrupted subsystems may write incomplete lines to their data file, which can be removed

D. Filtering Criteria

The following cases represent unrecoverable errors or chosen criteria that result in the removal of a trip from the dataset:

- **Nonconsenting driver**: When the driver is not a consented participant in the study

- **Requested removal**: When the subject requests certain trips, dates or times be removed
- **Vehicle doesn't move**: When the kinematics of the vehicle indicate no change in speed
- **Trip data files < 15MB**: When the total size of a trip's files are less than 15MB (faster than duration checks)
- **Trip duration < 30 seconds**: When the shortest camera recording is less than 30 seconds in duration
- **Missing essential files**: When camera files, `trip_dacman.json` or `data_can.csv` are missing
- **Outside volunteer participation range**: Indicative of MIT staff driving the vehicle to be maintained or washed
- **Large essential subsystem error files**: When there are many errors for a camera or for CAN
- **Mismatches in subsystem timestamps**: When one subsystem ends at least one minute earlier than another

E. Synchronized Files

Synchronized files are created by synchronization scripts that run after cleaning and filtering has taken place. These scripts align video frames and CAN messages at a rate of 30 frames per second. They are created using the same trip naming convention in a separate, processed directory.

- **synced_video.csv**: every row contains a video frame ID and timestamp from every camera at 30 frames per second
- **synced_video_*camera-name*.mp4**: Synchronized with all other videos at 30 FPS using H264 encoding
- **synced_can.csv**: each row represents a synced video frame and the closest CAN values associated with that timestamp for every CAN message
- **synced_vis_panels.mp4**: an optional visualization video file that displays all synced videos in separate panels where CAN data may be also displayed

VI. CONCLUSION AND ON-GOING FUTURE WORK

The application of state-of-the-art embedded system programming, software engineering, data processing, distributed computing, computer vision and deep learning techniques to the collection and analysis of large-scale naturalistic driving data in the MIT-AVT study seeks to break new ground in offering insights into how human and autonomous vehicles interact in the rapidly changing transportation system. This work presents the methodology behind the MIT-AVT study which aims to define and inspire the next generation of naturalistic driving studies. The governing design principle of this study is that, in addition to prior successful NDS approaches, we leverage the power of computer vision and deep learning to automatically extract patterns of human interaction with various levels of autonomous vehicle technology. We both (1) use AI to analyze the entirety of the driving experience in large-scale data and (2) use human expertise and qualitative analysis to dive deep into the data to gain case-specific understanding. To date, the dataset includes 99 participants, 11,846 days of participation, 405,807 miles, and 5.5 billion video frames. Statistics about the size and scope of the MIT-AVT dataset are updated regularly on https://hcai.mit.edu/avt.

REFERENCES

[1] A. Davies, "Oh look, more evidence humans shouldn't be driving," May 2015. [Online]. Available: https://www.wired.com/2015/05/oh-look-evidence-humans-shouldnt-driving/

[2] T. Vanderbilt and B. Brenner, "Traffic: Why we drive the way we do (and what it says about us) , alfred a. knopf, new york, 2008; 978-0-307-26478-7," 2009.

[3] W. H. Organization, *Global status report on road safety 2015*. World Health Organization, 2015.

[4] M. Buehler, K. Iagnemma, and S. Singh, *The DARPA urban challenge: autonomous vehicles in city traffic*. springer, 2009, vol. 56.

[5] V. V. Dixit, S. Chand, and D. J. Nair, "Autonomous vehicles: disengagements, accidents and reaction times," *PLoS one*, vol. 11, no. 12, p. e0168054, 2016.

[6] F. M. Favarò, N. Nader, S. O. Eurich, M. Tripp, and N. Varadaraju, "Examining accident reports involving autonomous vehicles in california," *PLoS one*, vol. 12, no. 9, p. e0184952, 2017.

[7] R. Tedrake, "Underactuated robotics: Algorithms for walking, running, swimming, flying, and manipulation (course notes for mit 6.832)," 2016.

[8] M. R. Endsley and E. O. Kiris, "The out-of-the-loop performance problem and level of control in automation," *Human factors*, vol. 37, no. 2, pp. 381–394, 1995.

[9] B. Reimer, "Driver assistance systems and the transition to automated vehicles: A path to increase older adult safety and mobility?" *Public Policy & Aging Report*, vol. 24, no. 1, pp. 27–31, 2014.

[10] K. Barry, "Too much safety could make drivers less safe," July 2011. [Online]. Available: https://www.wired.com/2011/07/active-safety-systems-could-create-passive-drivers/

[11] V. L. Neale, T. A. Dingus, S. G. Klauer, J. Sudweeks, and M. Goodman, "An overview of the 100-car naturalistic study and findings," *National Highway Traffic Safety Administration, Paper*, no. 05-0400, 2005.

[12] T. A. Dingus, S. G. Klauer, V. L. Neale, A. Petersen, S. E. Lee, J. Sudweeks, M. Perez, J. Hankey, D. Ramsey, S. Gupta *et al.*, "The 100-car naturalistic driving study, phase ii-results of the 100-car field experiment," Tech. Rep., 2006.

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

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

[15] T. Victor, M. Dozza, J. Bärgman, C.-N. Boda, J. Engström, C. Flannagan, J. D. Lee, and G. Markkula, "Analysis of naturalistic driving study data: Safer glances, driver inattention, and crash risk," Tech. Rep., 2015.

[16] M. Benmimoun, F. Fahrenkrog, A. Zlocki, and L. Eckstein, "Incident detection based on vehicle can-data within the large scale field operational test (eurofot)," in *22nd Enhanced Safety of Vehicles Conference (ESV 2011), Washington, DC/USA*, 2011.

[17] L. Fridman, P. Langhans, J. Lee, and B. Reimer, "Driver gaze region estimation without use of eye movement," *IEEE Intelligent Systems*, vol. 31, no. 3, pp. 49–56, 2016.

[18] L. Fridman, J. Lee, B. Reimer, and T. Victor, "Owl and lizard: patterns of head pose and eye pose in driver gaze classification," *IET Computer Vision*, vol. 10, no. 4, pp. 308–313, 2016.

[19] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, "Labelme: a database and web-based tool for image annotation," *International journal of computer vision*, vol. 77, no. 1, pp. 157–173, 2008.

[20] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3213–3223.

[21] R. R. Knipling, "Naturalistic driving events: No harm, no foul, no validity," in *Driving Assessment 2015: International Symposium on Human Factors in Driver Assessment, Training, and Vehicle Design*. Public Policy Center, University of Iowa Iowa City, 2015, pp. 196–202.

[22] ——, "Crash heterogeneity: implications for naturalistic driving studies and for understanding crash risks," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2663, pp. 117–125, 2017.

[23] L. Fridman, B. Jenik, and B. Reimer, "Arguing machines: Perception-control system redundancy and edge case discovery in real-world autonomous driving," *arXiv preprint arXiv:1710.04459*, 2017.

[24] V. Shankar, P. Jovanis, J. Aguero-Valverde, and F. Gross, "Analysis of naturalistic driving data: prospective view on methodological paradigms," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2061, pp. 1–8, 2008.

[25] N. Kalra and S. M. Paddock, "Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability?" *Transportation Research Part A: Policy and Practice*, vol. 94, pp. 182–193, 2016.

[26] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[27] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[28] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 248–255.

[29] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller, "Introduction to wordnet: An on-line lexical database," *International journal of lexicography*, vol. 3, no. 4, pp. 235–244, 1990.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[31] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[32] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[33] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask r-cnn," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[34] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[35] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *International Journal of Robotics Research (IJRR)*, 2013.

[36] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[37] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[38] M. Cordts, M. Omran, S. Ramos, T. Scharwächter, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset," 2015.

[39] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[40] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell, "Understanding convolution for semantic segmentation," *arXiv preprint arXiv:1702.08502*, 2017.

[41] S. Liu, J. Jia, S. Fidler, and R. Urtasun, "Sgn: Sequential grouping networks for instance segmentation," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[42] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognition Letters*, vol. 30, no. 2, pp. 88–97, 2009.

[43] Y. Tian, P. Luo, X. Wang, and X. Tang, "Pedestrian detection aided by deep learning semantic tasks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 5079–5087.

[44] V. Badrinarayanan, F. Galasso, and R. Cipolla, "Label propagation in video sequences," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. IEEE, 2010, pp. 3265–3272.

[45] P. Liu, S. Han, Z. Meng, and Y. Tong, "Facial expression recognition via a boosted deep belief network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1805–1812.

[46] Z. Yu and C. Zhang, "Image based static facial expression recognition with multiple deep network learning," in *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*. ACM, 2015, pp. 435–442.

[47] E. A. Hoffman and J. V. Haxby, "Distinct representations of eye gaze and identity in the distributed human neural system for face perception," *Nature neuroscience*, vol. 3, no. 1, pp. 80–84, 2000.

[48] J. Wiśniewska, M. Rezaei, and R. Klette, "Robust eye gaze estimation," in *International Conference on Computer Vision and Graphics*. Springer, 2014, pp. 636–644.

[49] L. Fridman, H. Toyoda, S. Seaman, B. Seppelt, L. Angell, J. Lee, B. Mehler, and B. Reimer, "What can be predicted from six seconds of driver glances?" in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 2017, pp. 2805–2813.

[50] F. Vicente, Z. Huang, X. Xiong, F. De la Torre, W. Zhang, and D. Levi, "Driver gaze tracking and eyes off the road detection system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 2014–2027, 2015.

[51] H. Gao, A. Yüce, and J.-P. Thiran, "Detecting emotional stress from facial expressions for driving safety," in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 5961–5965.

[52] I. Abdic, L. Fridman, D. McDuff, E. Marchi, B. Reimer, and B. Schuller, "Driver frustration detection from audio and video in the wild," in *KI 2016: Advances in Artificial Intelligence: 39th Annual German Conference on AI, Klagenfurt, Austria, September 26-30, 2016, Proceedings*, vol. 9904. Springer, 2016, p. 237.

[53] J. Shotton, T. Sharp, A. Kipman, A. Fitzgibbon, M. Finocchio, A. Blake, M. Cook, and R. Moore, "Real-time human pose recognition in parts from single depth images," *Communications of the ACM*, vol. 56, no. 1, pp. 116–124, 2013.

[54] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1653–1660.

[55] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler, "Joint training of a convolutional network and a graphical model for human pose estimation," in *Advances in neural information processing systems*, 2014, pp. 1799–1807.

[56] D. Sadigh, K. Driggs-Campbell, A. Puggelli, W. Li, V. Shia, R. Bajcsy, A. L. Sangiovanni-Vincentelli, S. S. Sastry, and S. A. Seshia, "Data-driven probabilistic modeling and verification of human driver behavior," *Formal Verification and Modeling in Human-Machine Systems*, 2014.

[57] 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.

[58] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Advances in neural information processing systems*, 2014, pp. 487–495.

[59] H. Xu, Y. Gao, F. Yu, and T. Darrell, "End-to-end learning of driving models from large-scale video datasets," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[60] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[61] "Advanced vehicle technology consortium (avt)," 2016. [Online]. Available: http://agelab.mit.edu/avt

[62] L. Fridman, D. E. Brown, W. Angell, I. Abdić, B. Reimer, and H. Y. Noh, "Automated synchronization of driving data using vibration and steering events," *Pattern Recognition Letters*, vol. 75, pp. 9–15, 2016.

[63] R. Li, C. Liu, and F. Luo, "A design for automotive can bus monitoring system," in *Vehicle Power and Propulsion Conference, 2008. VPPC'08. IEEE*. IEEE, 2008, pp. 1–5.