Review

Not peer-reviewed version

Practical Autonomous Driving: A Survey of Challenges and Opportunities

Shuvodeep De *, Debanjan Saha, Karam Sallam, Ali Mohamed, Ibrahim Radwan

Posted Date: 10 October 2023

doi: 10.20944/preprints202202.0123.v2

Keywords: cnn; ann; robotics; machine learning; ann; artificial intelligence

Article

Practical Autonomous Driving: A Survey of Challenges and Opportunities

Shuvodeep De [1,*], Debanjan Saha [2,], Karam M. Sallam [3,], Ali W. Mohamed [3,] and Ibrahim Radwan [4]

[1] Postdoctoral Researcher, University of Alabama and Oak Ridge National Lab; des1@ornl.gov
[2] College of Engineering, Northeastern University, Boston, MA, USA; saha.deb@northeastern.edu
[3] Faculty of Science and Technology, University of Canberra, Australia; karam.sallam@canberra.edu.au (K.M.S.); Ibrahim.Radwan@canberra.edu.au (I.R.)
[4] Operations Research Department, Faculty of Graduate Studies for Statistical Research, Cairo University, Giza 12613, Egypt

Abstract: Autonomous Driving (AD) has become a prominent research area in the field of Artificial Intelligence (AI) and Machine Learning (ML) in recent years. This opens the door wider for self-driving cars to surpass conventional vehicles in the current market share. Despite its apparent simplicity, AD is composed of complex and heterogeneous systems, which are in need of a high level of coordination and alignment to ensure both full automation and safety. Therefore, numerous research studies have been conducted over the last few years to facilitate such coordination and accelerate the capacity of these types of vehicles to be self-managed in complex situations. The paper summarises these approaches that led to building what is known nowadays as the autonomous driving pipeline. Moreover, although skepticism exists regarding the practicality of AD as a viable alternative to traditional vehicles, extensive research suggests the multiple benefits of relying on them for mobility. While challenges remain in implementing AD in the real world, including regulation and technical issues, substantial progress in recent years indicates a growing acceptance of AD in the near future. This paper further explores the advantages and opportunities of the conducted such systems in facilitating the practicality of Autonomous Driving.

Keywords: self-driving vehicle; autonomous driving; imitation learning; semantic segmentation; AD pipeline; computer vision; deep learning

1. Introduction

The self-driving car possesses the ability to operate without human intervention, relying on its own perception of the environment to navigate. Therefore, Autonomous Driving (AD) represents the direct advancement of the current progress in robotics and artificial intelligence. This indeed occurs by supporting self-driving cars with high-definition maps and protocols for computer vision, localisation, sensor fusion, prediction, planning, and controls. The core of self-driving technology lies in advanced concepts of artificial intelligence, machine learning, big data, and control systems. Currently, self-driving cars are a prominent subject of research and development worldwide, with continuous proposed techniques for innovations and technical advancements in the field. While more methods have explored AI-based architectures which are beneficial for AD, none of them have extensively addressed the practical challenges of implementing these techniques and their consequences. This paper endeavors to shed light on these issues by discussing the practicality of AD and self-driving vehicles.

In general, there are six distinct stages of vehicle automation. The first stage, known as "Zero Automation", relies solely on human decision-making and manual driving. Moving on to the second stage, also referred to as "Driver Support", intelligent features like parking sensors are introduced to assist the driver. The third stage, called "Restricted Automation", incorporates features like the "Lane-keeping System" and limited automatic cruise control, while still allowing the human driver to retain primary control of the vehicle. In the fourth stage, known as "Conditional Automation", human drivers can switch from manual driving to automatic driving for longer periods, such as highway driving, although occasional human intervention is still necessary. The fifth stage is "High-level Automation", where no human intervention is required, indicating that the vehicle operates autonomously. Lastly, the final stage is referred to as "Fully Autonomous," where neither a human interface nor human presence is required [1].

1.1. Advantages of the Autonomous Vehicle Compared to the Traditional Ones

Self-driving vehicles offer various opportunities and advantages compared to their traditional counterparts. We summarise them as follows:

- Every year approximately 1.25 million vehicle accidents occur globally, leading to a substantial loss of lives. In the United States, car accidents alone claim the lives of over $\approx 40,000$ individuals on an annual basis [2]. The main causes of such accidents are drowsiness, incapacitation, inattention, or intoxication [3]. According to advocates, most traffic accidents (more than 90%) are caused by human errors. Autonomous vehicles can address these issues by introducing a *zero-human error* approach, relying on computer-controlled systems to eliminate many of the mistakes that human drivers make on the road and improving safety for both passengers and pedestrians, while reducing loss of life and property damage.
- Enhanced Traffic Flow and Reliability: Due to limited human perception and reaction speeds, effectively utilising highway capacity becomes a challenge. Self-driving vehicles, being computer-controlled and interconnected, can enhance efficiency and alleviate congestion. This means that the existing roads will be able to accommodate more vehicles. With increased carrying capacity on current roadways, the need for constructing new roads or expanding existing ones to handle congestion will diminish. Consequently, the land designated for roads and parking can be repurposed for commercial and residential use. Additionally,the widespread adoption of self-driving cars in the transportation system can decrease traffic delays and accidents, leading to improved overall reliability [2].
- Environmental Advantages: With the advent of self-driving cars, the transportation system becomes safer and more dependable. Consequently, there is an opportunity to redesign vehicles, shifting away from heavy, tank-like models to lighter counterparts, which consume less fuel. Furthermore, self-driving concepts are also fostering the development of electric vehicles (EVs) and other alternative propulsion technologies. This collective reduction in fuel consumption promises to deliver significant environmental benefits that are highly valued, including a reduction in gas emissions and improved air quality.
- Mobility for Non-drivers: Self-driving cars provide an opportunity for non-drivers (*e.g.*, young, old, impaired, disabled, and people who do not possess a driving license) to have personal mobility [2].
- Reduced Driver Stress: A study involving 49 participants has demonstrated that the self-driving environment induces less stress compared to traditional driving [4]. This suggests that self-driving cars have the potential to improve overall well-being by reducing the workload and the associated stress of driving, as noted in a study by Parida et al. [2].
- The advent of self-driving cars is expected to significantly decrease the number of required parking spaces, particularly in the United States, by over 5.7 billion square meters, as reported by Parida et al. [2]. Several factors contribute to this improvement, including the elimination of the need for open-door space for self-parking vehicles, as there is no human driver who needs to exit the vehicle. As a result, vehicles can be parked more closely together, achieving an approximate 15% increase in parking density.
- Additional Time for Non-Driving Activities: Self-driving cars provide additional time for non-driving activities, enabling individuals to save up to approximately 50 minutes per day that would otherwise be spent on driving activities. This newfound time can be invested in work, relaxation, or entertainment.

- The concept of "mobility-on-demand" is gaining popularity in large cities, and self-driving vehicles are poised to support this feature. Through the deployment of self-driving taxis, private vehicle ride-sharing, buses, and trucks, high-demand routes can be efficiently served. Shared mobility has the potential to reduce vehicle ownership by up to 43% while increasing travel per vehicle by up to 75%.
- Real-Time Situational Awareness: Self-driving cars have the capability to utilise real-time traffic data, including travel time and incident reports. This enables them employing sophisticated navigation systems and efficient vehicle routing, resulting in improved performance and informed decision-making on the road, [2].
- Multidisciplinary Design Optimisation: Multidisciplinary design optimization (MDO) is a research field that explores the use of numerical optimization methods to design engineering systems based on multiple disciplines, such as structural analysis, aerodynamics, materials science, and control systems. MDO is widely used to design automobiles, aircraft, ships, space vehicles, electro-chemical systems, and more, with the goal of improving performance while minimizing weight and cost [5–13]. With the advancement of computational power, MDO frameworks can also be used for autonomous vehicles to improve their structural design, aerodynamics, powertrain optimization, sensor integration and placement, path planning, control system optimization, and energy management. There has been significant research into making vehicles lightweight without compromising their strength and safety, with composite materials being a popular choice for this purpose [14–16].
- The recent advancements in AI fields such as Computer Vision and Pattern Recognition, [17], open the door for various opportunities in achieving full automation in driving. This is evident by obtaining robust detection and tracking of ambient objects on roads [18,19].

1.2. The Intersection between Self-Driving Cars and Electric Vehicles (EVs)

Applying self-driving technology to electric vehicles (EVs) has its own set of pros and cons. While self-driving cars offer a safer and more reliable driving experience, they also prompt a transition from heavier vehicle designs, which prioritize safety, to lighter designs that are better suited for electric propulsion [2]. However, self-driving cars require a multitude of sensors and computers for processing, which consume significant amounts of energy. Since EVs already have limited battery range, self-driving EVs may only be capable of covering shorter distances. Nevertheless, experts believe that these challenges can be overcome through further technological advancements. By intelligently optimizing software and hardware adjustments, certain energy-consuming autonomous features can be improved, which can enhance the traveling range of self-driving EVs.

1.3. Why Self-Driving Cars Became Possible Due to the Development of Artificial Intelligence (AI)

AI serves as the deriving force behind self-driving cars, also known as autonomous vehicles.It aims to mimic human cognition and intelligence through machines, allowing them to perceive their surroundings, learn from experience, make informed decisions, and engage in logical reasoning [20].

The operation of a self-driving car entails complex tasks similar to those performed by human drivers, such as scene perception, detecting stationary and moving objects, ensuring safe navigation, adhering to traffic regulations, controlling speed, staying within lanes, changing lanes, maneuvering intersections, and avoiding accidents. Additionally, the presence of pedestrians and cyclists in congested urban environments poses further challenges for self-driving systems. Advanced concepts of AI, ML, big data processing, and control methodologies play significant roles in the development and implementation of self-driving technology [21].

Apart from the driving aspect, AI finds applications in various other domains related to the autonomous driving industry. For instance, in car manufacturing, the process can be seen as a puzzle-solving problem that requires precise fitting of numerous components. To achieve this, car manufacturers widely utilize robotic assistance and AI algorithms to ensure accurate and meticulous assembly of car parts. AI also contributes to car safety and driver protection by enabling features such as emergency braking, vehicle control, monitoring blind spots and cross-traffic, and synchronizing with traffic signs. Furthermore, AI aids in timely suggestions for predictive and prescriptive maintenance of cars by continuously monitoring their physical condition. Regulators and insurance companies benefit from the widespread use of AI as well. AI can collect data on a driver's behavior and risk profile, providing accurate information to insurance companies for assessing insurance costs. Additionally, AI assists drivers by adjusting seat positions, mirrors, air-conditioning, and even music preferences according to their individual choices.

1.4. Statistical Predictions about Expansion of Self-Driving Cars Industry in Near Future

The self-driving car industry, also known as autonomous driving (AD), has the potential to revolutionize the entire transportation system due to its numerous advantages over manual driving. In 2018, data indicated that the global market for self-driving cars was expected to grow from $5.6 billion to nearly $60 billion by 2030. Similarly, production levels of self-driving or robo-cars were projected to increase, reaching 800,000 units annually worldwide by 2030. Furthermore, a significant number of small businesses and startups (approximately 55%) were aware of the impending shift towards a fully autonomous transportation system within the next two decades. Consequently, they have been adapting and refining their business concepts to align with this transformative trend.

Table 1. Recent Survey Papers on Self-driving Cars

Authors	Year	Content
Yurtsever et al. [22]	2019	Autonomous vehicles, control, robotics, automation, intelligent vehicles, intelligent transportation systems
Liu et al. [23]	2021	Cooperative Autonomous Driving, IAAD, IGAD, IPAD
Huang et al. [24]	2020	deep learning, perception, mapping, localization, planning, control, prediction, simulation, V2X, safety, uncertainty, CNN, RNN, LSTM, GRU, GAN, simulation learning, reinforcement learning
Malik et al. [25]	2021	cooperative driving, collaboration, lane change, platooning, leader election
Contreras-Castillo et al. [26]	2019	Autonomous Car Stakeholders, Autonomy Models

Recently several survey papers on Autonomous Driving has been written. Table I gives examples of such reviews. The rest of this paper is organised as follows: a brief overview of various recent research and development in aid to artificial intelligence and control systems is discussed in Section 2. We discuss fundamentals of multi-task learning and meta learning in Section 3; a modular end-to-end pipeline of a self-driving car is discussed in Section 4; A practical case-study of comparison between two most popular Self-Driving Cars by Waymo (formerly Google Self-Driving Car Project) and Tesla Motors in Section 5; the challenges being faced and possible solutions have been reviewed in Section 6; at last, a brief synopsis of the paper is drawn.

2. Research and Development in AI and Control Strategies in the Field of Self-Driving Cars

Experiments on self-driving cars have been conducted since the 1920s, with notable milestones marking the progress of autonomous vehicle technology. In 1925, a demonstration called "American Wonder" showcased self-driving cars controlled by radio signals on the streets of New York City, establishing radio-controlled cars as a popular trend for many years.

In the 1980s, a significant development took place when Mercedes-Benz introduced a robotic van in Munich, Germany. This van utilized vision-based guidance, marking a shift towards the use of visual perception for autonomous driving. Subsequently, researchers and developers focused on integrating technologies such as LIDAR, GPS, and computer vision into a vision-based guidance system, comparing various methodologies to advance research in autonomous systems.

The concept of using trained neural networks for self-driving cars emerged in the early 1990s, pioneered by Dean Pomerleau, a Carnegie Mellon researcher. This approach involved capturing raw road images for neural network training, enabling the steering wheel to be controlled based on road conditions. The utilization of neural networks proved to be a significant breakthrough in self-driving cars, demonstrating high efficiency compared to other methods. Training methods like convolutional neural networks (R-CNN) became available, with neural networks relying on live video streaming and image data for training. However, it should be noted that neural networks are limited to two-dimensional (2D) images for training, which is a notable drawback.

To address the limitations of neural networks, the development of Light Detection and Ranging (LIDAR) technology has emerged. LIDAR collects real-time data in a 360-degree range around the car, providing a comprehensive view of the surroundings. Combining LIDAR technology with neural networks has been employed to enhance efficiency in self-driving cars. Additionally, GPS systems have been incorporated to improve navigation. As advancements progressed, odometry sensors and ultrasonic sensors were added to cars for motion detection and distance measurement, respectively. These technological advancements contribute to the ongoing development of self-driving cars. A central computer was also embedded in cars, to which all interfaced devices and the car's controlling systems were connected and controlled by [27].

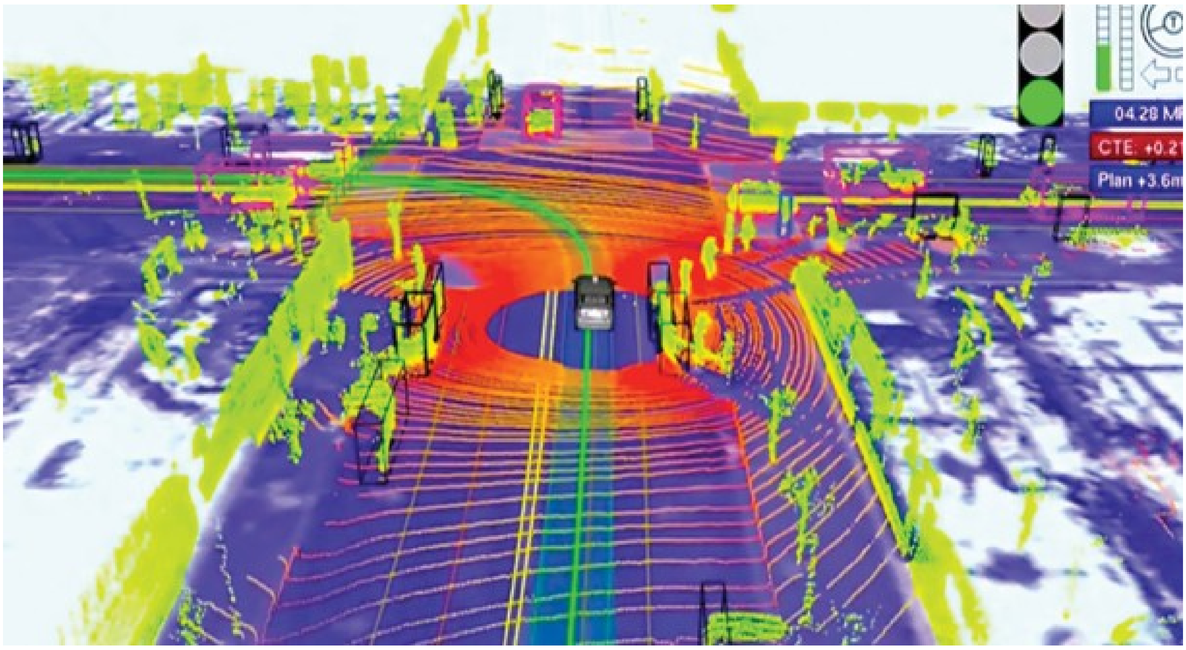

Figure 1. Application of LIDAR in path planning [28].

The self-driving car industry and autonomous driving (AD) have witnessed significant technological progress in recent years. Researchers and developers are actively engaged in research and development activities aimed at introducing innovative advancements and enhancements to self-driving cars. Cutting-edge concepts of artificial intelligence (AI) and efficient control strategies are being utilized to realize the desired features and incentives in autonomous vehicles.

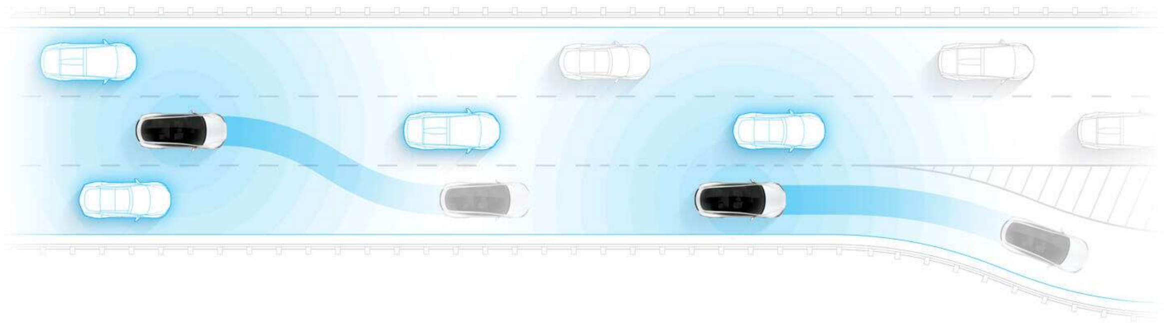

Figure 2. Predicting Lane Change by Tesla Autopilot (Source: https://www.tesla.com/autopilot).

In a publication by Li et al. (2019) [29], a new control policy called "Autonomous Driving Via Principled Simulations (ADAPS)" is proposed for self-driving cars. This control policy is claimed to be robust due to its inclusion of various scenarios, including rare ones like accidents, in the training data. ADAPS utilizes two simulation platforms to generate and analyze accidental scenarios. Through this process, a labeled training dataset and a hierarchical control policy are automatically generated. The hierarchical control policy, which incorporates memory, proves to be effective in avoiding collisions, as demonstrated through conducted experiments. Additionally, the online learning mechanism employed by ADAPS is deemed more efficient than other state-of-the-art methods like DAGGER, requiring fewer iterations in the learning process.

In another research work by Tan (2018) [30], the use of "Reinforcement Learning" and "Image Translation" for self-driving cars is recommended. While reinforcement learning does not necessitate an abundance of labeled data like supervised learning, it can only be trained in a virtual environment. The real challenge with reinforcement learning lies in bridging the gap between real and virtual environments, considering the unpredictable nature of real-world situations, such as accidents. The research presents an innovative platform that employs an "Image Semantic Segmentation Network" to enable adaptability to the real environment. Semantic images, which focus on essential content for training and decision-making while discarding irrelevant information, are used as input to reinforcement learning agents to address the disparity between real and virtual environments. However, the quality of the segmentation technique limits the outcomes of this approach. The model is trained in a virtual environment and then transferred to the real environment, minimizing the risks and significant losses associated with experimental failures.

The emergence of U-Net, a revolutionary semantic segmentation architecture developed by Ronneberger et al. [31], marked a significant breakthrough. This design combines the principles of a traditional Convolutional Network and is divided into two sections: the contracting path and the expanding path. In the contracting path, each convolutional block, along with activation and max pooling, has a stride of 2, resulting in down-sampling and doubling the number of feature channels. The expanding path then up-samples the convolutional blocks, reducing the feature channels by half. These blocks are merged with their counterparts from the contracting path, followed by 3x3 convolutions and ReLU activation. Finally, a 1x1 convolution is used to convert the 64-component feature into the desired number of classes.

Imitation learning has shown promise for self-driving cars, but it is not straightforward to train a model to achieve arbitrary goals. Goal-directed planning-based algorithms are typically employed to address this challenge, utilizing reward functions and dynamics models to accomplish diverse goals. However, specifying the reward function that elicits the desired behavior is difficult. To address this issue, researchers have proposed "Imitative Models" [32]. These probabilistic and predictive models can achieve specified goals by planning expert-like trajectories that are explainable. They can learn from expert demonstrations without requiring online data collection. The "Deep Imitative Model" has been evaluated for various goal objectives, such as energy-based, unconstrained, and constrained goal sets, and has demonstrated successful behavior alignment with the specified goal. The researchers also claimed that their model outperforms six state-of-the-art imitation learning approaches and a planning-based approach in a dynamic self-driving simulation task. Additionally, the model exhibits reasonable robustness when dealing with loosely specified objectives.

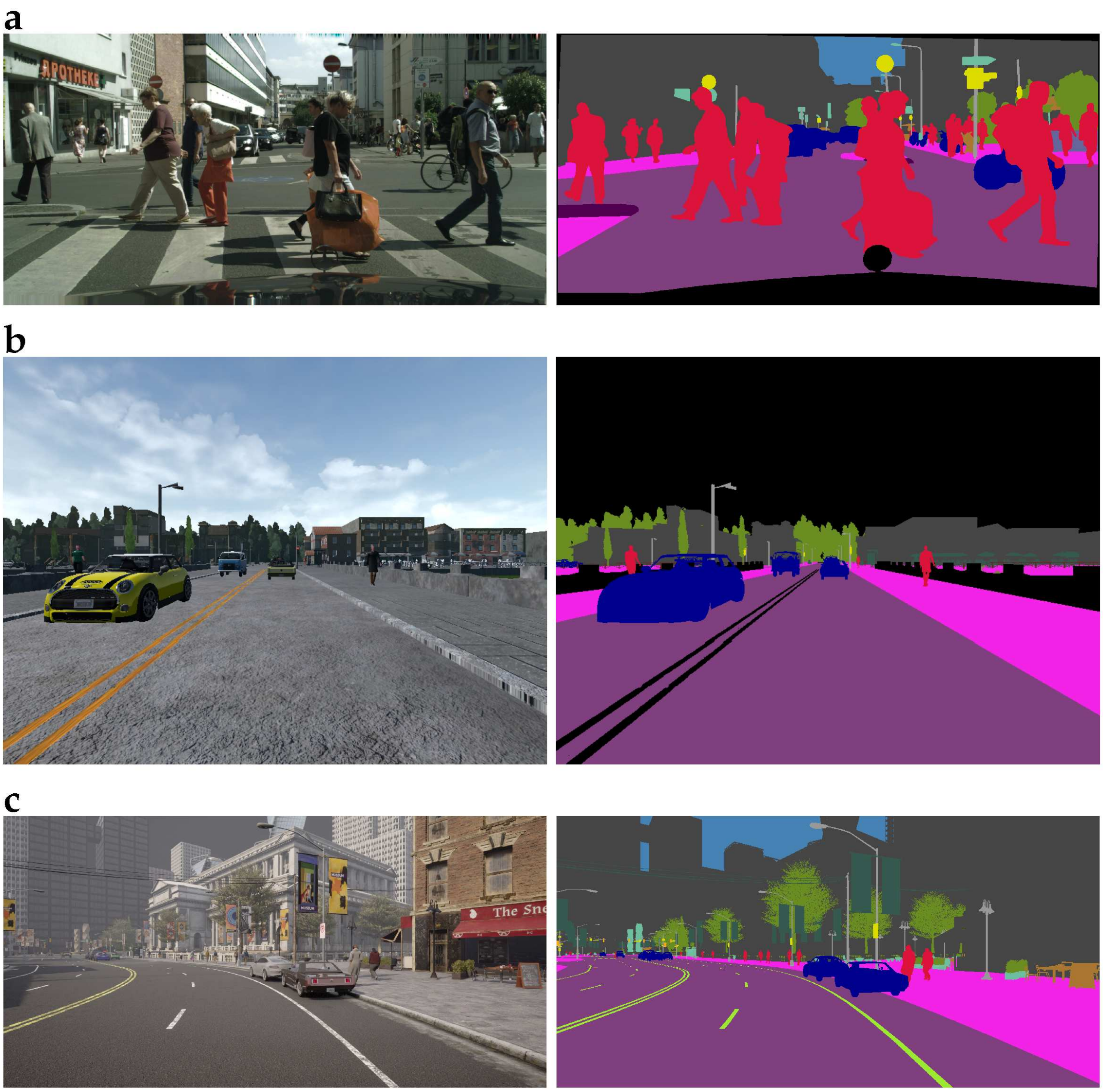

Figure 3. Semantic Segmentation [33].

Imitation learning trains deep networks based on driving demonstrations by human experts. These models can emulate expert behavior, such as avoiding obstacles and staying on straight roads. However, during testing, an end-to-end driving policy trained with imitation learning cannot be controlled or directed to make specific turns at intersections. To address this limitation, a technique called "Conditional Imitation Learning" [34] has been proposed. This model takes high-level commands as input during training, serving two important purposes. Firstly, these commands help resolve uncertainties in the mapping between perception and action, leading to better learning. Secondly, during testing, the controller is directed using the communication channel provided by these commands. The proposed approach has been evaluated in dynamic urban simulated environments as well as on a real robotic vehicle, showing improved performance compared to simple imitation learning in both scenarios.

Researchers have explored ways to further enhance the performance of imitation learning for robust driving in real-world conditions [35]. They found that cloning standard behaviors alone is insufficient to handle complex scenarios, even with a large number of examples. To address this challenge, they proposed synthesizing perturbations or interesting situations into the expert's driving behavior, such as going off the road or encountering collisions. Additionally, they introduced additional losses to exaggerate the imitation loss, discouraging failures and undesired events. These recommendations improved the robustness of the imitation learning model, referred to as the "ChauffeurNet model." The model demonstrated efficient handling of complex situations in simulations and ablation experiments.

Attention mechanisms, such as channel, spatial, and temporal attention, have gained popularity due to their resemblance to human visual systems. These mechanisms enable dynamic selection by adaptively weighting features based on their relevance. Wang et al. [36] introduced self-attention in computer vision, leading to significant advancements in video understanding and object detection. This breakthrough laid the groundwork for vision-transformers, and subsequent works [37–41] have demonstrated the remarkable potential of these attention mechanisms.

Direct perception is another innovative approach that combines the strengths of modular pipelines and imitation learning for autonomous driving. In modular pipelines, an elaborate model of the environment is created, while imitation learning directly maps images to control outputs. Direct perception techniques employ neural networks that learn from intermediate and low-dimensional representations. While direct perception approaches have been developed for highway scenarios, they have lacked features such as following traffic lights, obeying speed limitations, and navigating intersections. A recent research work introduced "Conditional Affordance Learning" [42], which applies the direct perception approach to complex urban environments. This method takes video input and maps it to intermediate representations, utilizing high-level directional commands for autonomous navigation. Researchers claimed a 68% improvement in goal-directed navigation compared to other reinforcement learning and conditional imitation learning approaches. Additionally, this approach successfully incorporates desirable features like smooth car-following, adherence to traffic lights, and compliance with speed limitations.

A comprehensive understanding of real traffic is crucial for self-driving cars. In the near future, combining various devices such as video recorders, cameras, and laser scanners will enable semantic comprehension of traffic. Currently, most approaches rely on large-scale videos for learning, as there is a lack of proper benchmarks for accurate laser scanner data. Researchers have introduced an innovative benchmark called "Driving Behaviour Net (DBNet)" [43] to address this gap. DBNet provides high-quality, large-scale point clouds obtained from Velodyne laser scanning and video recording with a dashboard camera. It also includes driving behavior information from a standard driver. The additional depth of information provided by the DBNet dataset proves beneficial for learning driving policies and improving prediction performance. Similarly, the "LIDAR-Video dataset" presented in the same work facilitates detailed understanding of real traffic behavior.

Another group of researchers has emphasized the importance of "Deep Object-Centric" models for self-driving cars [44]. Based on their findings in robotics tasks, representing objects explicitly in models leads to higher robustness in new scenarios and more intuitive visualizations. The researchers presented a classification of Object-Centric models that are advantageous for end-to-end learning and object instances. These proposed models were evaluated in a "Grand Theft Auto V simulator" for scenarios involving pedestrians, vehicles, and other objects. Performance metrics such as collision frequency, interventions, and distance driven were used to assess the models. The researchers claimed that their models outperformed object-agnostic methods, even when using a defective detector. Furthermore, evaluations in real environments with the "Berkeley DeepDrive Video dataset" demonstrated the models' effectiveness in low-data regimes.

End-to-end learning, a promising approach for self-driving cars, is being explored by a doctoral student [43]. It has the potential to achieve effective self-driving results in complex urban environments. Representation learning plays a crucial role in end-to-end learning, and auxiliary vision techniques, such as semantic segmentation, can facilitate it. While simple convolutional neural networks (CNNs) can handle reactive control tasks, they are not capable of complex reasoning. This doctoral thesis focuses on addressing scene-conditioned driving, which requires more than just reactive control. To tackle this, an algorithmic unified method is proposed, combining the advantages of imitation learning and reinforcement learning. This algorithm can learn from both the environment and demonstration data.

Transferring and scaling end-to-end driving policies from simulations to real-world environments is challenging. Real-world environments are more complex compared to the controlled and diverse environments provided by simulations for model training. Researchers have discovered a novel technique to scale simulated driving policies for real-world environments by using abstraction and modularity. This approach combines the benefits of end-to-end deep learning techniques with a modular architecture. The driving policy remains hidden from low-level vehicle dynamics and unprocessed perceptual input. The proposed methodology successfully transfers a driving policy trained on simulations to a 1/5-scale robotic truck without the need for fine-tuning. The robotic truck is evaluated in different circumstances on two continents, demonstrating the effectiveness of the approach.

The importance of surround-view cameras and route planners for end-to-end driving policy learning is emphasized in a study by [45]. Humans rely on side-view and rear-view mirrors, as well as their cognitive map, to understand their surroundings while driving. However, most self-driving research and development only utilize a front-facing camera without incorporating a route planner for driving policy learning. The researchers propose a more comprehensive setup that includes a route planner and a surround-view camera system with eight cameras, along with a CAN bus reader. This realistic sensor setup provides a 360° view of the vehicle's surroundings and a diverse driving dataset with varying weather conditions, illumination, and driving scenarios. The sensor setup also provides a proposed route to the destination and low-level driving activities, such as predicting speed and steering angle. The researchers empirically demonstrate that the setup with surround-view cameras and route planners improves the efficiency of autonomous driving, especially in urban driving environments with intersections.

Convolutional Neural Networks (CNNs) have been widely used in the field of self-driving cars. Previous end-to-end steering control methods often utilize CNNs for predicting steering angles based on image sequences. However, controlling the entire vehicle's operation solely through the prediction of steering angles is insufficient. To address this limitation, Yang et al. [46] propose a framework for multi-task learning that simultaneously predicts speed and steering angle for end-to-end learning. Accurately predicting speeds using visual input alone is not common practice. Their research introduces a multi-task and multi-modal network that incorporates sequences of images, recorded visualizations, and feedback speeds from previous instances as input. The proposed framework is evaluated on the "SAIC dataset" and the public Udacity dataset, demonstrating accurate predictions of speed and steering angle values.

While basic end-to-end visuomotor policies can be learned through behavior cloning, it is impossible to cover the entire range of driving behaviors. Researchers have conducted important exploratory work [34] and developed a benchmark to investigate the scalability and limitations of behavior cloning techniques. Their experiments show that behavior cloning techniques can perform well in arbitrary and unseen situations, including complicated longitudinal and lateral operations without explicit pre-programming. However, behavior cloning protocols have limitations such as dataset biasing and overfitting, model training instability, the presence of dynamic objects, and the lack of causality. Addressing these limitations requires further research and technical advancements to ensure practical driving in real-world environments.

It is ideal to evaluate self-driving policies and models in real-world environments with multiple vehicles. However, this approach is risky and impractical for the extensive amount of research being conducted in this field. Recognizing the need for a fair and appropriate evaluation method specifically designed for autonomous driving tasks, developers have realized the importance of "Vision-based Autonomous Steering Control (V-ASC) Models" [47]. They propose a suitable dataset for training models and develop a benchmark environment for evaluating different models, generating authentic quantitative and qualitative results. This platform allows the examination of the accuracy of steering value prediction for each frame and the evaluation of AD performance during continuous frame changes. The developers introduce a Software-In-the-Loop Simulator (S-ILS) that generates view-transformed image frames based on steering value changes, taking into account the vehicle and camera sensor models. They also present a baseline V-ASC model using custom-built features. A comparison between two state-of-the-art methods indicates that the end-to-end CNN technique is more efficient in achieving ground-truth (GT) tracking results, which strongly depend on human driving outcomes.

The need for an appropriate evaluation method is indispensable. Another group of researchers has investigated offline evaluation methods for AD models [34]. These methods involve collecting datasets with "Ground Truth Annotation" in advance for validation purposes. The research focuses on relating different offline and online evaluation metrics specific to AD models. The findings suggest that offline prediction error alone does not accurately indicate the driving quality of a model. Two models with the same prediction error can demonstrate significantly different driving performances. However, by carefully selecting validation datasets and appropriate metrics, the efficiency of offline evaluation methods can be greatly enhanced.

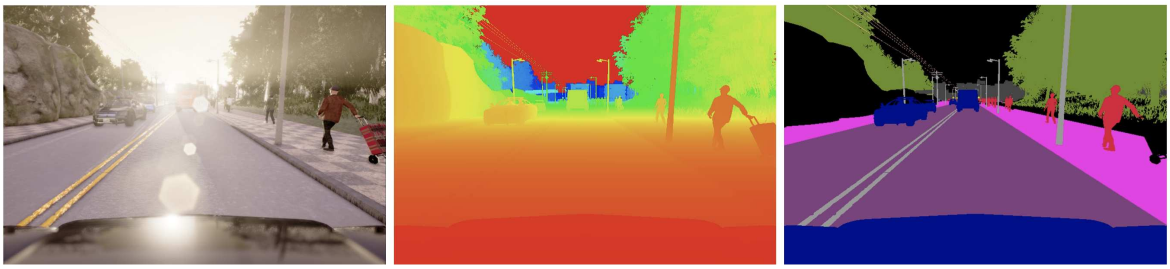

Figure 4. Sensing modalities by CARLA. From left to right: normal vision camera, ground-truth depth, ground-truth semantic segmentation [48]

Another novel approach, called "Controllable Imitative Reinforcement Learning (CIRL)," has been proposed for vision-based autonomous vehicles [49]. The researchers claimed that their driving agent, which relies solely on visual inputs, achieved a higher success rate in realistic simulations. Traditional reinforcement learning methods often struggle with the exploration of large and continuous action spaces in self-driving tasks, leading to inefficiencies. To address this issue, the CIRL technique restricts the action space by leveraging encoded experiences from the Deep Deterministic Policy Gradient (DDPG). The researchers also introduced adaptive strategies and steering-angle reward designs for various control signals such as "Turn Right," "Turn Left," "Follow," and "Go Straight." These proposals rely on shared representations and enable the model to handle diverse scenarios effectively. The researchers evaluated their methodology using the "CARLA Driving Benchmark" and demonstrated its superior performance compared to previously developed approaches in both goal-directed tasks and handling unseen scenarios.

Environments with a high presence of pedestrians, such as campus surroundings, pose challenges for self-driving policies. The development of robot or self-driving policies for such environments requires frequent intervention. Early detection and resolution of robotic decisions that could lead to failures are crucial. Recently, researchers proposed the "Learning from Intervention Dataset Aggregation (DAgger) algorithm" specifically designed for pedestrian-rich surroundings [50]. This algorithm incorporates an error backtrack function that can include expert interventions. It works in conjunction with a CNN and a hierarchically nested procedure for policy selection. The researchers claim that their algorithm outperforms standard imitation learning policies in pedestrian-rich environments. Furthermore, there is no need to explicitly model pedestrian behavior in the real world, as the control commands can be directly mapped to pixels.

doi:10.20944/preprints202202.0123.v2

"Learning by Cheating" is another promising approach for vision-based autonomous vehicles in urban areas [51]. The developers of this technique divided the learning process into two phases. In the first phase, a privileged agent is provided with additional information, including the locations of all traffic objects and the "Ground Truth Layout" of the surroundings. This privileged agent is allowed to exploit this privileged information and its own observations. In the second phase, a "Vision-based Sensorimotor" agent learns from the privileged agent without access to any privileged information or the ability to cheat. Although this two-step training algorithm may seem counterintuitive, the developers have experimentally demonstrated its benefits. They claim that their strategy achieves significantly better performance compared to other state-of-the-art methodologies on the CARLA benchmark and the NoCrash benchmark.

Computer Vision, particularly in the context of autonomous driving, has been a widely explored approach, thanks to the extensive research conducted in this field. Numerous object detection models have been proposed, often employing a combination of a backbone (encoder-decoder and transformer) such as VGG16, ResNet50, MobileNet [52,53], CSPDarkNet53, and a head for making predictions and bounding box estimations. The head architecture varies between one-stage detectors, such as YOLO, SSD [54], RetinaNet [55], CornerNet [56], and two-stage detectors, including different variants of the R-CNN family (R-CNN, R-FCN) [57]. In addition, some research like Feature Pyramid Network (FPN) [58], BiFPN [59], and NAS-FPN [60] have also included an additional *Neck* which basically collects information regarding the feature maps and performs regularization.

Data augmentation techniques, including photometric and geometric distortions, as well as object occlusion methods like CutOut [61], hide-and-seek [62], and grid mask [63], have shown significant improvements in image classification and object detection. Techniques such as DropOut [64], DropConnect [65] , and DropBlock [66], which replace randomly selected regions with zeros in feature maps, have been successful in dealing with the semantic bias and imbalanced datasets. Example mining techniques have been applied to two-stage detection models but not to the dense prediction layer in one-stage detection systems. To address this, Lin [67] introduced focal loss, which effectively handles the bias in the dense prediction layer.

Bounding box (BBox) estimation is crucial for image classification and object detection tasks. Traditional methods used various representations such as central coordinates, height and width, top-left and bottom-right corners, or anchor-based offset pairs to determine the size and position of the BBox. However, treating the object as an independent task sometimes compromised its integrity. To overcome this limitation, researchers introduced IoU loss [68], a scale-independent method that considers the ground truth when calculating the BBox area. Subsequently, GIoU [69] incorporated the object's form and orientation, DIoU [70] considered the central distance, and CIoU [70] took into account the object's overlap, center point distance, and aspect ratio. These advancements significantly improved the accuracy of BBox estimation.

Attention mechanisms, including Squeeze-and-Excitation (SE) [71] and Spatial Attention Module (SAM) [72], have been used for feature engineering. Spatial mechanisms like Spatial Pyramid Pooling (SPP) [73] have aggregated information from feature maps into a single-dimensional feature vector. SSPP was combined with max-pooling output of kernel size $k \times k, k \in [1, 5, 9, 13]$ and improved YOLOv3-608 AP$_{50}$ by 2.7% on the MS COCO dataset. Further, using RFB [74], it was improved to 5.7%. Other feature integration techniques, such as SFAM [75], ASFF [76], and BiFPN [77], have successfully integrated low-level physical features with high-level semantic features on a feature pyramid. PANet [78], proposed by Wang, performs per-pixel segmentation by leveraging few-shot non-parametric metric learning. YOLOv4, proposed by Bochkovskiy et al. [79], enhanced the YOLOv3 model by adding an SPP block, PANet as the path-aggregation neck, Mish-activation [80], DropBlock regularization, and employed Mosaic and Self-Adversarial Training (SAT) data aggregation. It also utilized a modified version of SAM, PAN, and Cross mini-Batch Normalization (CmBN), which selectively samples data within a batch, and employed CIoU loss for BBox estimation.

In summary, computer vision techniques, such as object detection architectures, data augmentation methods, attention mechanisms, and feature integration strategies, have greatly contributed to the advancement of autonomous driving. These techniques have addressed challenges related to exploration of action spaces, pedestrian-rich environments, and accurate object detection, enabling vision-based self-driving cars to achieve better performance and handle diverse scenarios effectively.

3. Multi-Task Learning and Meta Learning

AutoML has emerged as a popular field aimed at simplifying the use of machine learning techniques and reducing the reliance on experienced human experts. By automating time-consuming tasks such as *hyperparameter optimization* (HPO), using methods like *Bayesian optimization*, AutoML can enhance the efficiency of machine learning specialists. Google's *Neural Architecture Search* (NAS) [81] is a well-known AutoML deep learning approach. Another significant AutoML technology is Meta-learning [82], which involves teaching machine learning algorithms to learn from previous models through methods like *transfer learning* [30], *few-shot learning* [83], and even *zero-shot learning* [84].

In simple terms, a task $\mathcal{T}$ in the context of intelligent systems, such as robots, refers to accomplishing a specific objective by learning from repetitive actions. While single-task learning can be effective in environments where the learning objective remains consistent, it may not be applicable when the environment changes. For instance, in supervised learning, like classifying cat images, using a dataset of animal images $\mathcal{D}$, the task can be defined as follows:

$$\mathcal{D} = \{(\mathbf{x}, \mathbf{y})_k\}$$
$$\min_{\theta} L(\theta, \mathcal{D})$$

where, there are a lot of input-output (x, y) pairs. The objective is to minimize any some function function L which is a function of the dataset $\mathcal{D}$ and the parameters of the model θ. The loss function can vary a lot and is often subjected to the learning task. One common choice of such loss function can be a negative log likelihood loss in Equation (1).

$$\min_{\theta} L(\theta, \mathcal{D}) = -\mathbb{E}_{(x,y)\sim\mathcal{D}}[\log f_{\theta}(\mathbf{y}|\mathbf{x})] \tag{1}$$

which is the expectation of the datapoints in the dataset of the log probabilities of the target labels of the model. Thus, a task can be defined as a distribution of the input over the labels and the loss function as in Equation (2).

$$\mathcal{T}_i \triangleq \{p_i(\mathbf{x}), p_i(\mathbf{y}|\mathbf{x}), L_i\} \tag{2}$$

In the context of multi-task learning, there are scenarios where certain parameters remain consistent across all tasks. For instance, in multi-task classification, the loss function $\mathcal{L}_i$ is identical for all tasks. Similarly, in multi-label learning, both the loss function and the input distribution $(\mathcal{L}_i, p_i(\mathbf{x}))$ are the same across all tasks. An example of this is scene understanding, where the loss function can be approximated as shown in Equation (3). However, it's important to note that the loss functions may differ when the distribution changes (e.g., mixed discrete versus continuous labels) or when there is a preference for one task over another.

$$L_{tot} = w_{\text{depth}} L_{\text{depth}} + w_{\text{kpt}} L_{\text{kpt}} + w_{\text{normals}} L_{\text{normals}} \tag{3}$$

Here, it also uses a task descriptor $\mathbf{z}_i$ which is an index of the task or any meta-data related to the task. Thus, the objective changes to a summation over all the tasks as:

$$\min_{\theta} \sum_{i=1}^{T} L(\theta, \mathcal{D}) \tag{4}$$

also, the some parameters (θ) can be shared and optimized across all the tasks (θ^{sh}) and some parameters can be task-specific parameters (θ^{i}), which can in turn be shared and optimized among similar sub-tasks. Thus the objective function generalises to Equation (5). Choosing various task descriptor conditioning determines how these parameters are shared.

$$\min_{\theta^{sh},\theta^{1},\ldots,\theta^{T}} \sum_{i=1}^{T} L_i(\{\theta^{sh},\theta^{i}\},\mathcal{D}_i) \tag{5}$$

3.1. Conditioning Task Descriptor

Extensive research has been conducted on how to condition the task description with various techniques like:

3.1.1. Concatenation

In this method, the task descriptor is added to the neural network after passing it through a linear layer. This step is taken to adjust the dimensionality of the task vector to match the hidden layers.

3.1.2. Additive Conditioning

In this strategy, the network is divided into multiple heads, each consisting of task-specific layers. This allows the selection of specific segments of the network for different tasks.

3.1.3. Multi-head Architecture

In this strategy, the network is divided into multiple heads, each consisting of task-specific layers. This allows the selection of specific segments of the network for different tasks.

3.1.4. Multiplicative Conditioning

In this approach, the output of the task descriptor is projected onto the individual tasks within the input or hidden layers. Each task is trained separately, without sharing parameters across tasks. This provides precise control over features for specific tasks.

While more complex conditioning techniques for task descriptors exist, each of these methods is tailored to a specific optimization problem and requires domain knowledge for effective utilization.

3.2. Objective Optimization

Meta Learning follows standard neural network objective function optimization. A simple objective function minimization can be depicted in Algorithm below:

- Sample Mini Batch $\mathcal{B} \sim \{\mathcal{T}_i\}$
- Sample Mini Batch Datapoints for each task $\mathcal{D}_i^b \sim \mathcal{D}_i$
- Compute Loss on Mini Batch $\mathcal{L}(\theta,\mathcal{B}) = \sum_{\mathcal{T}_k \in \mathcal{B}} \mathcal{L}_k(\theta,\mathcal{D}_k^b)$
- Compute Gradient $\nabla_\theta \mathcal{L}$ via Backpropagation
- Optimize using Gradient information

3.3. Action Prediction

Initially, the perception module of the self-driving car detects the presence of obstacles in its surroundings. It utilizes obstacle data and basic information such as accelerations, headings, velocities, and positions as input. Upon processing this input, the perception module predicts the trajectories of all obstacles, along with their associated probabilities. Subsequently, the prediction module analyzes and forecasts the expected behavior of these obstacles. It is crucial for the prediction module to operate in real-time, with high precision and minimal latency. Additionally, the prediction module should possess the ability to learn and adapt to the varying behaviors exhibited by nearby obstacles. This section encompasses some cutting-edge approaches employed for action prediction in self-driving cars.

- Model-based Prediction
- Driven-based Prediction
- Lane sequence-based predictions
- Recurrent neural networks

Model-based Prediction: It comprises two models, one for describing the vehicle's movement (straight or curved) and another for determining turning positions.

Driven-based Prediction: It relies on machine learning and involves training a model on observations to generate real-time predictions.

Lane sequence-based predictions: This approach divides the original path into segments and uses sensors in self-driving cars to detect obstacles. Classical probabilistic graphical modeling methods, such as spatio-temporal graphs, factors graph, and dynamic Bayesian networks, are used to predict the state of detected obstacles and model temporal sequences.

Recurrent neural networks: This approach utilizes recurrent neural networks for prediction and takes advantage of time-series data. Object detection using a "Single Shot Detector (SSD)" module is performed for handling larger inputs. Recurrent neural networks can also detect videos. Trajectory planning and generation are carried out as a concluding step, where the action prediction module generates constraints to choose the most suitable trajectory for the self-driving car.

To achieve optimal path planning, the physical environment is digitally represented, enabling the search for the best and drivable lane and passage for the self-driving car. [85].

3.4. Depth and Flow Estimation

Precisely estimating optical flow, scene depth, and ego-motion poses significant challenges in computer vision, robotics, video analysis, simultaneous localization and mapping (SLAM), and self-driving cars. While humans can easily and quickly perceive ego-motion and scene direction, developing reconstruction models for real-world scenes is highly challenging. Obstacles such as light reflection, occlusion, and non-rigidity complicate the process [86].

Accurate depth estimation plays a critical role in various computer vision applications, including 3D reconstruction, object tracking, and self-driving assistance. Photometric-based methods for depth estimation can be categorized into two main approaches:

- Stereo Scenes
- Monocular Scenes [86]

Previous research in the field of depth estimation has attempted to replicate the binocular human vision system using Graphics Processing Units (GPUs) [87] for real-time processing of source scenes. These studies tried to achieve accurate and efficient depth estimation [88]; however, certain challenges remained, such as inadequate depth quality in occluded areas, differences in image scale, and variations in lighting conditions. To address these issues, a more effective approach involves analyzing sequential or unordered scenes and predicting motion or structure. One commonly used method for quickly generating scene structure is "Structure from Motion" (SfM), which leverages ego-motion. This approach is sensitive and capable of handling homogeneous regions [89]. Similarly, "Visual Odometry" techniques are widely adopted as learning methodologies in this context.

These methods involve training network structures using a portion of the ground truth dataset. Simultaneously, the same portion of the dataset is utilized for achieving odometry [**?**]. Typically, visual reconstruction encompasses tasks such as extracting feature points, matching scenes, and verifying geometry. Subsequently, visual reconstruction requires optimization through specialized procedures such as bundle adjustments and determining camera poses based on matching points. The main drawback of visual odometry techniques lies in their susceptibility to geometric constraints, including homogeneous regions, occlusions, and complex textures during feature point extraction [86].

Furthermore, many researchers are exploring the use of supervised learning techniques for scene reconstruction tasks, including camera pose estimation, feature matching, and camera ego-motion prediction. Convolutional neural networks (CNNs) are highly recommended for accurate extraction of feature points from input scenes [**?**]. Researchers have also been experimenting with recursive CNN models that make use of sequential scenes, claiming improved efficiency in feature extraction from individual scenes [90]. Additionally, incorporating probabilistic approaches like conditional random field (CRF) can further enhance the efficiency of motion estimation [91]. CNN architectures have been proven effective for both calibrated stereo scenes and monocular scenes in estimating depth.

Moreover, depth and optical flow estimation problems have also been explored using unsupervised learning techniques. These approaches offer a solution to the geometric challenges posed by the camera, which were previously considered difficult to address through learning techniques. Since sequential scenes exhibit spatial smoothness, the loss function can be modeled by extending the concept of scene reconstruction (Reference 53). Similarly, the photometric discrepancy technique, along with other depth estimation approaches, can be employed to predict ego-motion from a monocular scene [92]. A Kalman filter is recommended for estimating depth and camera pose from monocular video sequences, as it enhances the smoothness of the camera's ego motion within the learning framework [93]. Additionally, the deployment of a multi-stream architecture of CNNs can help avoid aliasing problems [86].

3.5. Behavior Prediction of Nearby Objects

Having the ability to accurately predict the behaviors of nearby vehicles, cyclists, and pedestrians is extremely valuable for self-driving cars to navigate safely. In terms of safety, it is important for self-driving cars to maintain a safe distance from all other objects in their surroundings. At the same time, they must prioritize minimal fuel consumption, time efficiency, and adherence to traffic laws. However, this task of ensuring safe and efficient navigation becomes even more difficult in densely populated urban areas where there are more vehicles driven by humans, as well as pedestrians and cyclists. In order to drive cautiously and avoid any collisions, self-driving cars need to calculate the expected paths of all surrounding objects [94].

Due to the wide range of driving behaviors exhibited by human drivers, including aggressiveness and unpredictability, extensive research is being conducted to assess this significant factor. Self-driving cars are programmed to prioritize hyper-cautious driving in all situations, which can potentially frustrate human drivers nearby and result in minor accidents [95]. Conversely, human drivers are known to act recklessly in certain scenarios, such as sudden lane changes, aggressive maneuvers, and imprudent overtaking [94]. Mavrogiannis and colleagues have developed an innovative algorithm that utilizes learning techniques to take into account the behaviors of human drivers when predicting actions and navigating autonomous vehicles (AVs). They have categorized traffic participants into conservative individuals and those with varying degrees of aggressiveness.

To achieve this goal, a versatile simulator with rich behavior capabilities has been implemented. This simulator can generate various longitudinal and lateral behaviors and offers a range of driving styles and traffic densities for different scenarios. The prediction of high-level actions for a self-driving car when faced with aggressive drivers nearby is performed using a 'Markov Decision' process. Additionally, the interaction between different traffic participants is modeled using a 'Graph Convolutional Network (GCN)', which trains a behavioral policy [94]. Another group of researchers has recognized the significance of considering uncertainties associated with behavioral predictions. They have highlighted the need for self-driving cars to take into account not only the behavioral predictions of nearby traffic participants but also the corresponding uncertainties. To address this, they have proposed a new framework called 'uncertain-aware integrated prediction and planning (UAPP)'.

This framework gathers real-time information about other traffic participants and generates behavioral predictions along with their associated uncertainties [96].

The presence of many pedestrians in an area poses challenges for self-driving cars in predicting their behavior and ensuring safe navigation. Accurately predicting the future paths of nearby pedestrians over extended time periods becomes crucial. In order to address this issue, Jayaraman et al. introduced a hybrid model that estimates long-term trajectories (over 5 seconds) of pedestrians, enabling smoother interaction with self-driving cars. This model combines the dynamics of constant velocity with the pedestrians' tendency to accept gaps in traffic, as described in their analysis [97].

A research study has examined the behavior of cyclists, who are important road users in urban areas, and proposed an innovative framework for predicting their intentions. The researchers focused on four specific hand signals commonly used by cyclists and developed a data generation pipeline to overcome the challenge of insufficient data in this area. The pipeline is used to generate synthetic point cloud scans representing various cyclists in urban environments. Ultimately, the framework takes a set of these point cloud scans as input and produces predictions for the cyclists' intentions, assigning the highest probability [98].

3.6. One Shot Learning

One-shot learning is a key method in computer vision that enables the categorization of objects depicted in an image. Unlike other machine learning and deep learning techniques, which rely on large amounts of labeled data or numerous samples for effective training, one-shot learning aims to learn from just a single sample or image during the training phase. It then applies this learned knowledge to categorize objects during the testing phase. This approach takes inspiration from the human brain's ability to quickly learn thousands of object categories by observing only a few samples or examples from each category.

In the context of one-shot learning, when a new object category is presented to the algorithm with just a single example (without being previously included in the training dataset), the algorithm recognizes it as a new category. It then utilizes this single sample to learn and subsequently becomes capable of re-identifying instances of this object class in the future. One notable application of one-shot learning is face recognition, which is widely used in smart devices and security systems for re-identification purposes. For example, the face recognition system employed at passport security checks takes the passport photo as a single training sample and then determines whether the person holding the passport matches the image in the passport photo [99].

Researchers have developed many algorithms capable of performing one-shot learning. Some of them are:

- 'Probabilistic models' using 'Bayesian learning'
- 'Generative models' deploying 'probability density functions'
- Images transformation
- Memory augmented neural networks
- Meta learning
- Metric learning exploiting 'convolutional neural networks (CNN)' [99]

The realm of self-driving cars and robotics is making use of one-shot learning algorithms in various ways. Typically, deep neural networks are employed to visually understand driving scenes in fully autonomous driving systems. However, these networks necessitate large annotated databases for training. Fortunately, a recent research study has introduced a method called one-shot learning, which eliminates the need for manual annotation during the training of perception modules. This approach involves the development of a generative framework known as "Generative One-Shot Learning (GOL)." GOL takes in one-shot samples or generic patterns, along with some regularization samples, to guide the generative process [100].

In a similar vein, Hadad et al. have proposed the use of one-shot learning techniques for driver identification in shared mobility services. They have implemented a fully functional system using a camera module, Python language, and a Raspberry Pi. The application of one-shot learning for face recognition has yielded satisfactory results [101]. Additionally, one-shot learning can also be leveraged for surveillance anomaly detection without requiring precise temporal annotations. Researchers have developed an innovative framework that utilizes a three-dimensional CNN Siamese network to identify surveillance abnormalities. The discriminative properties of the 3D CNN enable the detection of similarities between two sequences of surveillance anomalies [102].

3.7. Few Shot Learning

Few-shot learning is a subset of machine learning that focuses on learning from a small number of labeled examples for each classification present in a database. It falls within the realm of supervised learning techniques. Few-shot learning finds its main applications in object detection and recognition, image classification, as well as sentiment classification. One of the key advantages of few-shot learning techniques is their ability to operate without the need for large databases, which is typically required by most machine learning approaches. Leveraging prior knowledge, few-shot learning can generalize to new tasks with limited examples and supervised information. However, a significant challenge in few-shot learning lies in the empirical unreliability of risk minimizers. Different methods in few-shot learning address this issue in various ways, thereby aiding in the categorization of these methods.

- Some methods use data to enhance supervised experience by using previously acquired knowledge
- Some methods use model to minimize the dimensions of hypothesis space by deploying previously acquired knowledge
- Others use previous knowledge to facilitate algorithm which helps in searching of optimal hypothesis present in given space [103]

Many researchers in the field of autonomous driving and robotics have embraced few-shot learning techniques to address specific challenges. With the escalating road traffic and diverse traffic scenarios, ensuring safety for self-driving cars has become increasingly crucial. To tackle this, a group of researchers has proposed equipping self-driving cars with multiple low-light cameras equipped with fish-eye lenses to enhance the assessment of nearby traffic conditions. Furthermore, a combination of LIDAR and infrared cameras captures the front view. The few-shot learning algorithm is utilized to identify nearby vehicles and pedestrians in the front view. Additionally, the researchers have developed a comprehensive embedded visual assistance system that integrates all these devices into a miniature setup [104].

Similarly, Majee et al. have encountered the "few-shot object detection (FSOD)" problem in real-world, class-imbalanced scenarios. They specifically selected the "India Driving Dataset (IDD)" due to its inclusion of less frequently occurring road agents. The researchers conducted experiments with both metric-learning and meta-learning approaches for FSOD and found that metric-learning yielded superior outcomes compared to the meta-learning method [105].

Furthermore, researchers have conducted experiments using a new approach called "Surrogate-gradient Online Error-triggered Learning (SOEL)" to achieve few-shot learning. This method utilizes neuromorphic processors and combines advanced techniques such as deep learning, transfer learning, and computational neuroscience. During the research, Spiking Neural Networks (SNNs) were partially trained and made adaptable online to unfamiliar data classes within a specific domain. The SOEL system automatically updates itself after encountering an error, enabling fast learning. Gesture recognition tasks have been successfully performed using the SOEL system, demonstrating its ability to rapidly learn new classes of gesture data using a few-shot learning paradigm. The developed system also has potential applications in predicting the behavior of pedestrians and cyclists in self-driving cars [106].

In the field of autonomous driving, point clouds play a crucial role, and their applicability continues to expand. While deep neural networks with labeled point clouds have been used for various supervised learning tasks, the annotation of point clouds remains a burdensome process that should be minimized. To address this issue, some researchers proposed an innovative solution involving the use of a cover-tree to hierarchically partition the point cloud, which is then encoded through two self-supervised learning pre-training tasks. Few-shot learning has been employed for the pre-training of downstream self-supervised learning networks, reducing the reliance on labeled data during the annotation process [107].

4. Modular Pipeline

This section presents important modules of self-driving car's pipeline along with their brief descriptions. The picture depicted in Figure 5 shows the pipeline.

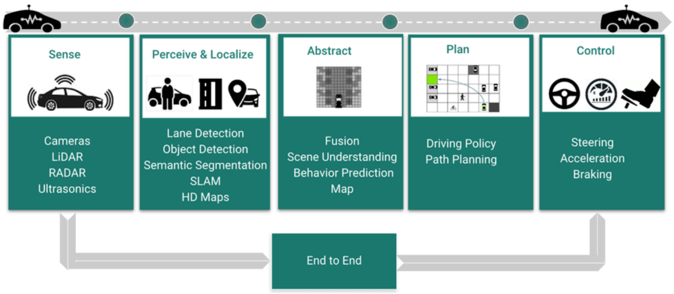

Figure 5. Self-Driving Car Pipeline [108].

4.1. Sensor Fusion

This section includes information on the essential requirement for self-driving vehicles to have precise perception of their surroundings. This allows them to make intelligent driving decisions in real-time by sensing the environment. To achieve this, a diverse range of sensors are extensively utilized in self-driving cars to provide both location and perception information. The section covers the two primary sensor categories and their different types.

- Exteroceptive Sensors: They mainly sense the external environment and measure distances to different traffic objects. The following technologies can be used as an exteroceptive sensor in self-driving cars.
  - LiDAR (Light Detection and Ranging): LiDAR can measure distances to different objects remotely by using energy-emitting sensors. It sends a pulse of laser and then senses "Time Of Fight (TOF)", by which pulse comes back.
  - Radar: Radar can sense distances to different objects along with their angle and velocity, by using electromagnetic radiation or radio waves.
  - Camera: A camera builds up a digital image by using passive light sensors. It can detect static as well as dynamic objects in the surroundings.
  - Ultrasonic: An Ultrasonic sensor also calculates distances to neighboring objects by using sound waves.
- Proprioceptive Sensors: They calculate different system values of the vehicle itself, such as the position of the wheels, the angles of the joints, and the speed of the motor.
  - GPS (Global Positioning System): GPS provides geolocation as well as time information all over the world. It is a radio-navigation system based on satellites.
  - IMU (Inertial Measurement Unit): The IMU calculates an object's force, magnetic field, and angular rate.
  - Encoders: It is an electro-mechanical instrument which takes an angular or linear shaft's position as its input and generates a corresponding digital or analogue signal as its output.

Each sensor possesses its own set of strengths and weaknesses. For instance, radar excels at accurately measuring range without being impacted by environmental lighting conditions, but it falls short in providing detailed information about an object's appearance. On the other hand, cameras excel at capturing appearance details but can be influenced by environmental lighting factors. To enhance the accuracy and reliability of detection, a logical solution is to combine the data from different sensors through sensor fusion. Numerous techniques are currently being developed for the purpose of sensor fusion. Some of them are:

- Vision - LiDAR/Radar: It is used for modelling of surroundings, vehicle localization as well as object detection.
- Vision – LiDAR: It can track dynamic objects by deploying LiDAR technology and a stereo camera.
- GPS-IMU: This system is developed for absolute localization by employing GPS, IMU and DR (Dead Reckoning).
- RSSI-IMU: This algorithm is suitable for indoor localization, featuring RSSI (Received Signal Strength Indicator), WLAN (Wireless Local Area Network) and IMU [109].

4.2. Localization

Accurate positioning of a vehicle on high-definition (HD) maps is essential for self-driving operations, and thus localization methods play a crucial role. HD maps provide crucial information in a 3D representation, encompassing details about road networks, lanes, intersections, and the positions of signboards, among other things. Solving the localization problem involves utilizing sensor data and HD maps as inputs. Various technologies are currently employed for achieving localization in self-driving cars.

- GNSS-RTk: GNSS (Global Navigation Satellite System) deploys 30 GPS satellites, which are being positioned in space at 20,000 km away from earth. RTK (Real-Time Kinematic) navigation system is also based on satellites and provides accurate position data.
- Inertial navigation: This system is also used for localization and uses motion sensors such as accelerometers, rotational sensors like gyroscopes, and a processing device for computations.
- LIDAR localization: LIDAR sensor provides 3D point clouds, containing information about surroundings. Localization is being performed by incessantly exposing and matching LIDAR data with HD maps. Algorithms used to test point clouds are "Iterative Closet Point (ICP)" and "Filter Algorithms (such as Kalman filter)".

4.3. Planning and Control

Planning serves as a fundamental step for decision-making and can be seen as a necessary requirement for vehicle routing. Routing involves creating a driving plan that encompasses the route from an initial point to a destination, utilizing input such as map data, the vehicle's location on the map, and the desired destination. Planning has two major steps:

- Path planning / generation: An appropriate path for vehicle is being planned. If a car needs to change the lane, it must be planned carefully without any accidental scenario.
- Speed planning / generation: It calculates the suitable speed of the vehicle. It also measures the speeds and distances of neighboring cars and utilizes this information in speed planning.

The control module is responsible for implementing the driving strategy of the vehicle, which includes controlling the steering wheel, brakes, and acceleration pedals [1].

The localization and perception modules play a crucial role by providing essential information to the planning and control modules, enabling their operation. In essence, the planning and control modules follow a protocol where they select one predefined behavior from a set of multiple behaviors based on the current situation at hand. This protocol is commonly practiced in the planning and control modules of self-driving cars.

- The Routing module gets destination data from the user and generates a suitable route accordingly by investigating road networks and maps.
- The behavioral planning module receives route information from the routing module and inspects applicable traffic rules and develops motion specifications.
- The motion planner receives both route information and motion specifications. It also exhibits localization and perception information. By utilising all provided information, it generates trajectories.
- Finally, the control system receives these developed trajectories and plans the car's motion. It also emends all execution errors in the planned movements in a reactive way.

4.4. Computer Vision

Computer vision is a specialized field of artificial intelligence that aims to extract valuable insights from visual inputs, such as camera outputs, videos, and digital images. These insights are then processed to make informed decisions and recommendations [110]. Computer vision concepts are also being utilized in the context of self-driving cars, particularly for localization and perception tasks. A novel approach known as "Visual Simultaneous Localization and Mapping (VSLAM)" is used to accurately determine the position of a vehicle in real-time. However, a limitation of the VSLAM technique is the accumulation of errors, where localization errors increase as the vehicle travels greater distances. To address this issue, the VSLAM method is combined with "Global Navigation Satellite System (GNSS)" localizations to enhance accuracy.

Furthermore, computer vision finds application in active perception. "Stereo Vision" can be employed to obtain detailed spatial information about various objects, while deep learning algorithms can provide semantic information about multiple items. By combining spatial and semantic data, it becomes possible to detect different objects such as pedestrians and other vehicles with greater accuracy [111].

5. Case Study: Waymo vs Tesla

Extensive research has enabled the development of actual self-driving vehicles in the real world. These vehicles have undergone extensive training in computer simulations and real-world conditions, often with passengers on board. Companies such as Waymo (formerly Google Self-Driving Car Project) and Tesla Motors have achieved remarkable progress by rigorously testing their vehicles and deploying a wide range of models that have traveled millions of miles without major accidents.

Pioneers in the field have explored diverse architectures and deep learning algorithms due to the vast combination of underlying framework parameters. Both approaches involve gathering substantial amounts of metadata about the vehicle's environment and surroundings, although the methods of processing and data collection may vary. Waymo, for example, utilizes a modular pipeline approach for autonomous vehicles, incorporating five onboard cameras, LiDAR, and HD maps. In contrast, Tesla Motors focuses exclusively on computer vision, relying on eight cameras. This approach increases the computational demands on the vision aspect but simplifies the interaction between different pipeline systems. The collected metadata, often referred to as "trigger-points," serves as input for solving meta-learning tasks. These triggers include moving objects, stationary objects, line markings, road markings, crosswalks, road signs, traffic signs, and various environmental factors, as shown in Figure 6. It is crucial to note that achieving high accuracy ($\sim 99.99\%$) in solving these tasks is essential for deploying such vehicles in real-world scenarios. Reliable vision systems require highly accurate models for operation. This is exemplified in Figure 7, where a Tesla car accurately recognizes environmental conditions, such as wet roads, in the real world and updates its neural network accordingly.

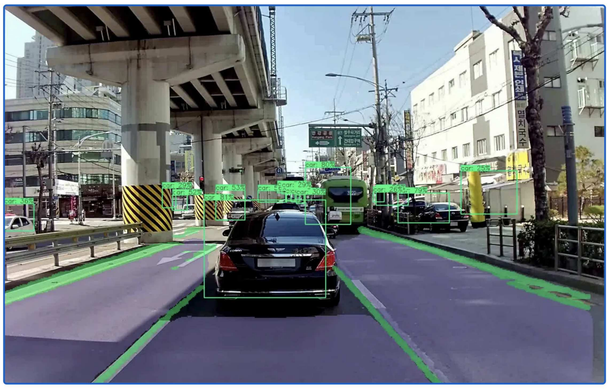

Figure 6. Various Attributes in Multi-Task Meta Learning Classification (Source:)

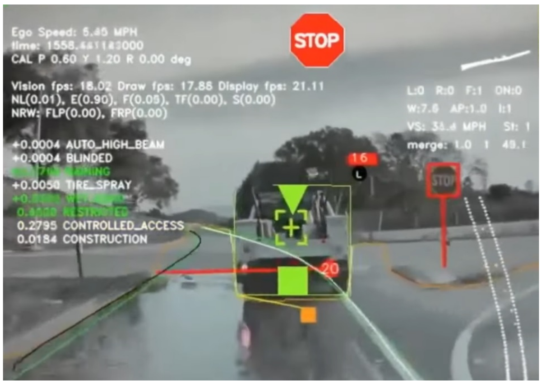

Figure 7. Real World Tesla Operation (Source:)

Tesla recently announced their plans to replace their previous technology stack, which included Radar, Vision, and Sensor Fusion, with solely Computer Vision. This approach is driven by the efficiency of Vision compared to their previous Sensor Fusion techniques. They argue that radar has evident disadvantages in various situations, particularly during emergency braking, where the radar signals exhibit a jagged response to sudden brakes. Additionally, when the vehicle passed under a bridge, the radar system struggled to determine if it was a stationary object or a stationary car. In such cases, Vision was relied upon for confirmation and successfully differentiated the object by reporting "negative velocity" for a few frames. Vision also offers vertical resolution, enabling the classification of stationary bridges or cars.

Another significant drawback of the Radar stack is its susceptibility to environmental triggers. As mentioned earlier, radar heavily relies on vision for classification. Therefore, a noisy vision layer can introduce delays in the radar stack's response, leading to a reduced reaction time to stationary objects. On the other hand, using Vision alone allows for early identification of such objects, resulting in longer and smoother braking responses. The utilization of Vision alone has demonstrated improved precision and recall compared to the previous technology stack.

In a Tesla vehicle, there are eight cameras that capture continuous sequences of images. These images are then processed using an image extractor, such as a standard Resnet model. The next step involves fusing all the images from the eight different vision sources together using a Transformer-like network. This fusion process enables the sharing of information across all the cameras and over time, utilizing Transformers (such as RNN, 3D CNN, or other hybrid structures).

Once the multi-camera fusion is completed, the information is passed through various components, including heads, trunks, and terminals, in a branching-like architecture employed by Tesla, as depicted in Figure 8. This architecture offers several advantages, particularly in situations where resources are limited, and employing individual neural networks for each independent task is not feasible. Feature sharing becomes crucial in such cases, as the amount of input data sampled from each input source may vary depending on the target feature. Additionally, this architecture allows for the decoupling of signals at the terminal level, independent of operations on other levels.

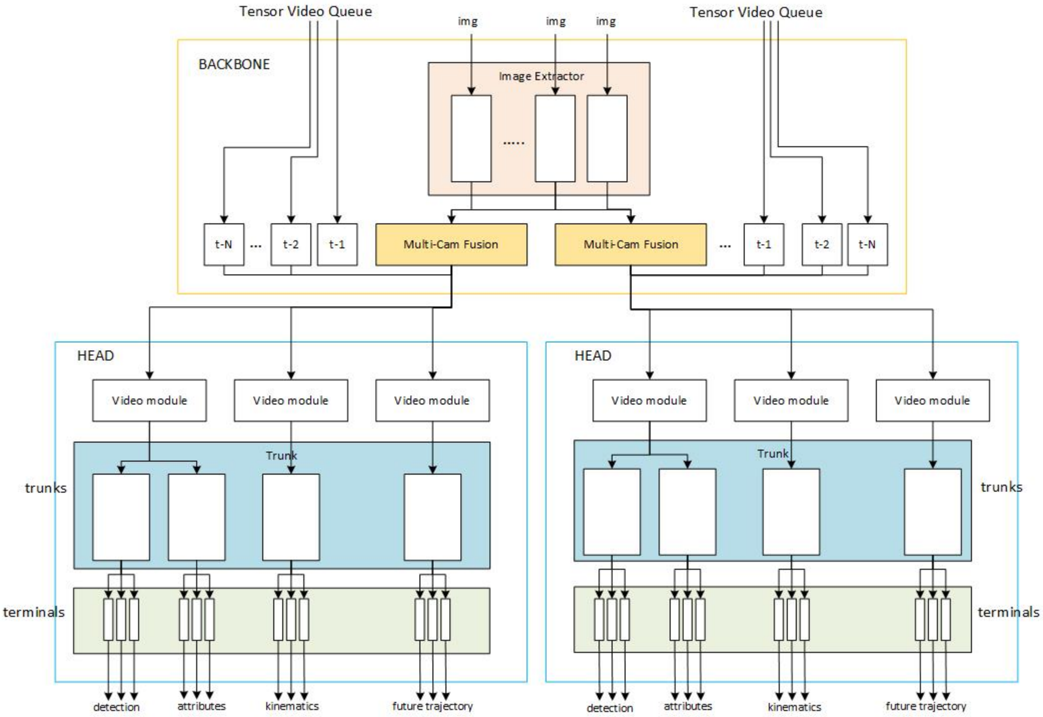

Figure 8. Tesla Neural Network Architecture (Source: Andrej Karpathy via CVPR'21).

6. Challenges

6.1. Current Challenges of Self-Driving Cars

The industry of self-driving cars is currently facing following challenges in the way to attain their potential advantages.

6.1.1. Ethical Issues

As the transition from manual driving to self-driving takes place, the infrastructure will be shared by both self-driving vehicles and human-driven vehicles. Additionally, pedestrians must also be taken into account. This necessitates careful consideration of numerous ethical issues by the programmers and designers of self-driving cars, leading to increased design complexity. Specifically, the ethical decision-making of self-driving cars in certain crash scenarios has faced significant criticism. As a result, software programmers for self-driving cars must prioritize crash avoidance protocols and devote more attention to this aspect [112].

6.1.2. Cybersecurity

Currently, the dominant focus of self-driving car research does not encompass the challenges linked to cybercrimes. Nonetheless, cybersecurity plays a crucial role in ensuring smooth operation and safety in this domain, considering the immense destructive potential of a single compromised vehicle. It is imperative to address cybersecurity concerns in both the functioning of self-driving cars and their communication networks [112].

6.1.3. Road Infrastructure and the Transition

The road infrastructure plays a crucial role in the transition towards Fully Automatic Driving (FAD), particularly during the initial phases when self-driving cars coexist with human drivers on the same roads. In this context, the main challenge arises from the unpredictable reactions of human drivers to random events [112]. Interactions between self-driving cars and human drivers can be particularly challenging. To address this issue, the concept of constructing autonomous-only traffic lanes has been proposed. These dedicated lanes would be exclusively reserved for self-driving cars until the complete transition to autonomous driving (AD) is achieved [113].

6.1.4. Regulatory Needs

The government and regulatory bodies have a crucial role to play in facilitating the transition towards autonomous driving (AD). Their actions are vital in creating an environment conducive to the successful realization of AD [113]. One way the government can contribute is by offering incentives to attract automotive Original Equipment Manufacturers (OEMs) and start-ups to invest in AD. Additionally, the government should avoid imposing regulations that discourage research and development, as well as innovation in the field of AD. Collaborations between legislative bodies, partnerships, and government agencies can also expedite progress and advancements in this research domain [2].

6.1.5. Hardware Requirements and Resource Allocation

Autonomous vehicles rely on multiple sources of input data, and the volume of data to be processed increases exponentially within a short period of time. Consequently, specialized on-premise hardware is necessary to accurately solve these diverse classification tasks. As the complexity of the system grows, the number of individual classification tasks also increases. Assigning separate neural networks to address each task can lead to potential bottlenecks. To address this, companies often implement resource sharing among different tasks. Furthermore, the amount of data sampled from each task can vary significantly. This is managed by adjusting batch intervals, allowing for less frequent data sampling for certain tasks and higher-frequency sampling for others, depending on their specific requirements.

6.1.6. Haywire Environment

Many real-world scenarios are unpredictable and require on-the-spot processing. Autonomous driving (AD) environments typically rely on well-defined markers as input for decision-making. However, this poses a significant challenge in countries where traffic regulations are not clearly defined. For example, in areas where multiple vehicles and pedestrians cross the road simultaneously, or where traffic lights and road markers are not properly established. In such cases, it becomes extremely difficult for a self-driving vehicle to adapt to the situation due to the unpredictable nature of traffic movement patterns. To address this, it is crucial to enforce strict legal traffic guidelines and establish dedicated road infrastructure services that can provide clear guidance in such conditions.

6.2. User Acceptance and Public Opinion and How It Can Be Improved Further

The complete transition to an autonomous driving (AD) environment relies heavily on public acceptance of the shift from manual driving to self-driving. Public opinion plays a significant role, and numerous studies and online surveys have been conducted to assess user acceptance. One crowdsourced online survey, involving 8,862 participants, revealed that 39% of respondents were in favor of AD, while 23% expressed a negative attitude towards it [114].

Similarly, another online survey with 5,000 participants from 109 countries indicated that 69% of respondents believed that the AD industry would capture a 50% market share by 2050.

However, participants raised concerns regarding software misuse and hacking, safety issues, and legal considerations as major obstacles to AD technology [115].

To increase the percentage of user acceptance, it is crucial to address the challenges faced by the self-driving car industry. Further research, development, innovation, and technical advancements are essential in this field. Implementing more reliable and cautious self-driving policies and models can attract more proponents and address the reservations of those who are skeptical. As mentioned previously, ethical issues, cybersecurity, and robust crash avoidance protocols are key factors influencing user acceptance.

References

1. Yoganandhan, A.; Subhash, S.; Jothi, J.H.; Mohanavel, V. Fundamentals and development of self-driving cars. *Materials today: proceedings* **2020**, *33*, 3303–3310.
2. Parida, S.; Franz, M.; Abanteriba, S.; Mallavarapu, S. Autonomous driving cars: future prospects, obstacles, user acceptance and public opinion. International Conference on Applied Human Factors and Ergonomics. Springer, 2018, pp. 318–328.
3. Campbell, M.; Egerstedt, M.; How, J.; Murray, R. Autonomous driving in urban environments: approaches, lessons and challenges. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* **2010**, *368*, 4649 – 4672.
4. Jamson, H.; Merat, N.; Carsten, O.; Lai, F. Fully-automated driving: The road to future vehicles. Proceedings of the Sixth International Driving Symposium on Human Factors in Driver Assessment, Training and Vehicle Design. Citeseer, 2011, pp. 2–9.
5. De, S.; Singh, K.; Seo, J.; Kapania, R.K.; Ostergaard, E.; Angelini, N.; Aguero, R. Structural Design and Optimization of Commercial Vehicles Chassis under Multiple Load Cases and Constraints. AIAA Scitech 2019 Forum, 2019, p. 0705.
6. Jrad, M.; De, S.; Kapania, R.K. Global-Local Aeroelastic Optimization of Internal Structure of Transport Aircraft Wing. 18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2017, p. 4321.
7. Robinson, J.H.; Doyle, S.; Ogawa, G.; Baker, M.; De, S.; Jrad, M.; Kapania, R.K. Aeroelastic Optimization of Wing Structure Using Curvilinear Spars and Ribs (SpaRibs). 17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, 2016, p. 3994.
8. De, S.; Singh, K.; Seo, J.; Kapania, R.K.; Ostergaard, E.; Angelini, N.; Aguero, R. Lightweight Chassis Design of Hybrid Trucks Considering Multiple Road Conditions and Constraints. *World Electric Vehicle Journal* **2021**, *12*, 3.
9. De, S.; Jrad, M.; Locatelli, D.; Kapania, R.K.; Baker, M. SpaRibs geometry parameterization for wings with multiple sections using single design space. 58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 2017, p. 0570.
10. De, S.; Singh, K.; Seo, J.; Kapania, R.; Aguero, R.; Ostergaard, E.; Angelini, N. Unconventional Truck Chassis Design with Multi-Functional Cross Members. Technical report, SAE Technical Paper, 2019.
11. De, S. Structural Modeling and Optimization of Aircraft Wings having Curvilinear Spars and Ribs (SpaRibs). PhD thesis, Virginia Tech, 2017.
12. De, S.; Singh, K.; Alanbay, B.; Kapania, R.K.; Aguero, R. Structural Optimization of Truck Front-Frame Under Multiple Load Cases. ASME International Mechanical Engineering Congress and Exposition. American Society of Mechanical Engineers, 2018, Vol. 52187, p. V013T05A039.
13. De, S.; Kapania, R.K. Algorithms For 2d Mesh Decomposition In Distributed Design Optimization. *arXiv preprint arXiv:2002.00525* **2020**.
14. Devarajan, B. Free Vibration analysis of Curvilinearly Stiffened Composite plates with an arbitrarily shaped cutout using Isogeometric Analysis. *arXiv preprint arXiv:2104.12856* **2021**.
15. Devarajan, B.; Kapania, R.K. Thermal buckling of curvilinearly stiffened laminated composite plates with cutouts using isogeometric analysis. *Composite Structures* **2020**, *238*, 111881.
16. Devarajan, B.; Kapania, R.K. Analyzing thermal buckling in curvilinearly stiffened composite plates with arbitrary shaped cutouts using isogeometric level set method. *Aerospace Science and Technology* **2022**, p. 107350.

17. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Liong, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. nuscenes: A multimodal dataset for autonomous driving. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 11621–11631.
18. Wang, B.; Zhu, M.; Lu, Y.; Wang, J.; Gao, W.; Wei, H. Real-time 3D object detection from point cloud through foreground segmentation. *IEEE Access* **2021**, *9*, 84886–84898.
19. Li, J.; Dai, H.; Han, H.; Ding, Y. MSeg3D: Multi-modal 3D Semantic Segmentation for Autonomous Driving. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 21694–21704.
20. Ponnusamy, B. The role of artificial intelligence in future technology. *International Journal of Innovative Research in Advanced Engineering* **2018**, *5*, 146–148.
21. Stone, P.; Brooks, R.; Brynjolfsson, E.; Calo, R.; Etzioni, O.; Hager, G.; Hirschberg, J.; Kalyanakrishnan, S.; Kamar, E.; Kraus, S.; others. Artificial intelligence and life in 2030: the one hundred year study on artificial intelligence **2016**.
22. Yurtsever, E.; Lambert, J.; Carballo, A.; Takeda, K. A Survey of Autonomous Driving: Common Practices and Emerging Technologies. *CoRR* **2019**, *abs/1906.05113*,
23. Liu, S.; Yu, B.; Tang, J.; Zhu, Q. Towards Fully Intelligent Transportation through Infrastructure-Vehicle Cooperative Autonomous Driving: Challenges and Opportunities. *CoRR* **2021**, *abs/2103.02176*,
24. Huang, Y.; Chen, Y. Autonomous Driving with Deep Learning: A Survey of State-of-Art Technologies. *CoRR* **2020**, *abs/2006.06091*,
25. Malik, S.; Khan, M.A.; El-Sayed, H. Collaborative autonomous driving—A survey of solution approaches and future challenges. *Sensors* **2021**, *21*, 3783.
26. Contreras-Castillo, J.; Zeadally, S.; Guerrero-Ibánez, J. Autonomous cars: challenges and opportunities. *IT Professional* **2019**, *21*, 6–13.
27. Shreyas, V.; Bharadwaj, S.N.; Srinidhi, S.; Ankith, K.; Rajendra, A. Self-driving cars: An overview of various autonomous driving systems. *Advances in Data and Information Sciences* **2020**, pp. 361–371.
28. Ondruš, J.; Kolla, E.; Vertal', P.; Šarić, Ž. How do autonomous cars work? *Transportation Research Procedia* **2020**, *44*, 226–233.
29. Li, W.; Wolinski, D.; Lin, M.C. ADAPS: Autonomous driving via principled simulations. 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019, pp. 7625–7631.
30. Tan, C.; Sun, F.; Kong, T.; Zhang, W.; Yang, C.; Liu, C. A Survey on Deep Transfer Learning. *ArXiv* **2018**, *abs/1808.01974*.
31. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. MICCAI, 2015.
32. Rhinehart, N.; McAllister, R.; Levine, S. Deep imitative models for flexible inference, planning, and control. *arXiv preprint arXiv:1810.06544* **2018**.
33. Ivanovs, M.; Ozols, K.; Dobrajs, A.; Kadikis, R. Improving Semantic Segmentation of Urban Scenes for Self-Driving Cars with Synthetic Images. *Sensors* **2022**, *22*. doi:10.3390/s22062252.
34. Codevilla, F.; Müller, M.; López, A.; Koltun, V.; Dosovitskiy, A. End-to-end driving via conditional imitation learning. 2018 IEEE international conference on robotics and automation (ICRA). IEEE, 2018, pp. 4693–4700.
35. Bansal, M.; Krizhevsky, A.; Ogale, A. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *arXiv preprint arXiv:1812.03079* **2018**.
36. Wang, X.; Girshick, R.B.; Gupta, A.K.; He, K. Non-local Neural Networks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* **2018**, pp. 7794–7803.
37. Guo, M.H.; Cai, J.; Liu, Z.N.; Mu, T.J.; Martin, R.R.; Hu, S. PCT: Point Cloud Transformer. *Comput. Vis. Media* **2021**, *7*, 187–199.
38. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; Houlsby, N. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *ArXiv* **2021**, *abs/2010.11929*.
39. Yuan, L.; Chen, Y.; Wang, T.; Yu, W.; Shi, Y.; Tay, F.E.H.; Feng, J.; Yan, S. Tokens-to-Token ViT: Training Vision Transformers from Scratch on ImageNet. *ArXiv* **2021**, *abs/2101.11986*.
40. Wang, W.; Xie, E.; Li, X.; Fan, D.P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; Shao, L. Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions. *ArXiv* **2021**, *abs/2102.12122*.

41. Wu, H.; Xiao, B.; Codella, N.C.F.; Liu, M.; Dai, X.; Yuan, L.; Zhang, L. CvT: Introducing Convolutions to Vision Transformers. *ArXiv* **2021**, *abs/2103.15808*.
42. Sauer, A.; Savinov, N.; Geiger, A. Conditional affordance learning for driving in urban environments. Conference on Robot Learning. PMLR, 2018, pp. 237–252.
43. Chen, Y.; Wang, J.; Li, J.; Lu, C.; Luo, Z.; Xue, H.; Wang, C. Lidar-video driving dataset: Learning driving policies effectively. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5870–5878.
44. Wang, D.; Devin, C.; Cai, Q.Z.; Yu, F.; Darrell, T. Deep object-centric policies for autonomous driving. 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019, pp. 8853–8859.
45. Hecker, S.; Dai, D.; Van Gool, L. End-to-end learning of driving models with surround-view cameras and route planners. Proceedings of the european conference on computer vision (eccv), 2018, pp. 435–453.
46. Yang, Z.; Zhang, Y.; Yu, J.; Cai, J.; Luo, J. End-to-end multi-modal multi-task vehicle control for self-driving cars with visual perceptions. 2018 24th International Conference on Pattern Recognition (ICPR). IEEE, 2018, pp. 2289–2294.
47. Kwon, S.; Park, J.; Jung, H.; Jung, J.; Choi, M.K.; Tayibnapis, I.R.; Lee, J.H.; Won, W.J.; Youn, S.H.; Kim, K.H.; others. Framework for Evaluating Vision-based Autonomous Steering Control Model. 2018 21st International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2018, pp. 1310–1316.
48. Dosovitskiy, A.; Ros, G.; Codevilla, F.; Lopez, A.; Koltun, V. CARLA: An Open Urban Driving Simulator. Proceedings of the 1st Annual Conference on Robot Learning, 2017, pp. 1–16.
49. Liang, X.; Wang, T.; Yang, L.; Xing, E. Cirl: Controllable imitative reinforcement learning for vision-based self-driving. Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 584–599.
50. Bi, J.; Xiao, T.; Sun, Q.; Xu, C. Navigation by imitation in a pedestrian-rich environment. *arXiv preprint arXiv:1811.00506* **2018**.
51. Chen, D.; Zhou, B.; Koltun, V.; Krähenbühl, P. Learning by Cheating. *ArXiv* **2019**, *abs/1912.12294*.
52. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *ArXiv* **2017**, *abs/1704.04861*.
53. Sandler, M.; Howard, A.G.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* **2018**, pp. 4510–4520.
54. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.E.; Fu, C.Y.; Berg, A. SSD: Single Shot MultiBox Detector. ECCV, 2016.
55. Lin, T.Y.; Goyal, P.; Girshick, R.B.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2020**, *42*, 318–327.
56. Law, H.; Deng, J. CornerNet: Detecting Objects as Paired Keypoints. *ArXiv* **2018**, *abs/1808.01244*.
57. Hariharan, B.; Arbeláez, P.; Girshick, R.B.; Malik, J. Hypercolumns for object segmentation and fine-grained localization. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* **2015**, pp. 447–456.
58. Lin, T.Y.; Dollár, P.; Girshick, R.B.; He, K.; Hariharan, B.; Belongie, S.J. Feature Pyramid Networks for Object Detection. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* **2017**, pp. 936–944.
59. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* **2020**, pp. 10778–10787.
60. Ghiasi, G.; Lin, T.Y.; Pang, R.; Le, Q.V. NAS-FPN: Learning Scalable Feature Pyramid Architecture for Object Detection. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* **2019**, pp. 7029–7038.
61. Devries, T.; Taylor, G.W. Improved Regularization of Convolutional Neural Networks with Cutout. *ArXiv* **2017**, *abs/1708.04552*.
62. Singh, K.K.; Yu, H.; Sarmasi, A.; Pradeep, G.; Lee, Y.J. Hide-and-Seek: A Data Augmentation Technique for Weakly-Supervised Localization and Beyond. *ArXiv* **2018**, *abs/1811.02545*.
63. Chen, P.; Liu, S.; Zhao, H.; Jia, J. GridMask Data Augmentation. *ArXiv* **2020**, *abs/2001.04086*.
64. Srivastava, N.; Hinton, G.E.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
65. Wan, L.; Zeiler, M.D.; Zhang, S.; LeCun, Y.; Fergus, R. Regularization of Neural Networks using DropConnect. ICML, 2013.
66. Ghiasi, G.; Lin, T.Y.; Le, Q.V. DropBlock: A regularization method for convolutional networks. NeurIPS, 2018.

67. Lin, T.Y.; Goyal, P.; Girshick, R.B.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2020**, *42*, 318–327.
68. Yu, J.; Jiang, Y.; Wang, Z.; Cao, Z.; Huang, T.S. UnitBox: An Advanced Object Detection Network. *Proceedings of the 24th ACM international conference on Multimedia* **2016**.
69. Rezatofighi, S.H.; Tsoi, N.; Gwak, J.; Sadeghian, A.; Reid, I.; Savarese, S. Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* **2019**, pp. 658–666.
70. Zheng, Z.; Wang, P.; Liu, W.; Li, J.; Ye, R.; Ren, D. Distance-IoU Loss: Faster and Better Learning for Bounding Box Regression. *ArXiv* **2020**, *abs/1911.08287*.
71. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-Excitation Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2020**, *42*, 2011–2023.
72. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. ECCV, 2018.
73. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2015**, *37*, 1904–1916.
74. Liu, S.; Huang, D.; Wang, Y. Receptive Field Block Net for Accurate and Fast Object Detection. *ArXiv* **2018**, *abs/1711.07767*.
75. Zhao, Q.; Sheng, T.; Wang, Y.; Tang, Z.; Chen, Y.; Cai, L.; Ling, H. M2Det: A Single-Shot Object Detector based on Multi-Level Feature Pyramid Network. AAAI, 2019.
76. Liu, S.; Huang, D.; Wang, Y. Learning Spatial Fusion for Single-Shot Object Detection. *ArXiv* **2019**, *abs/1911.09516*.
77. Tan, M.; Pang, R.; Le, Q.V. EfficientDet: Scalable and Efficient Object Detection. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* **2020**, pp. 10778–10787.
78. Wang, K.; Liew, J.H.; Zou, Y.; Zhou, D.; Feng, J. PANet: Few-Shot Image Semantic Segmentation with Prototype Alignment, 2020, arXiv:cs.CV/1908.06391.
79. Bochkovskiy, A.; Wang, C.Y.; Liao, H. YOLOv4: Optimal Speed and Accuracy of Object Detection. *ArXiv* **2020**, *abs/2004.10934*.
80. Misra, D. Mish: A Self Regularized Non-Monotonic Activation Function. BMVC, 2020.
81. Elsken, T.; Metzen, J.H.; Hutter, F. Neural Architecture Search: A Survey. *ArXiv* **2019**, *abs/1808.05377*.
82. Vanschoren, J. Meta-Learning: A Survey. *ArXiv* **2018**, *abs/1810.03548*.
83. Wang, Y.; Yao, Q.; Kwok, J.; Ni, L. Generalizing from a Few Examples: A Survey on Few-Shot Learning. *arXiv: Learning* **2019**.
84. Wang, W.; Zheng, V.; Yu, H.; Miao, C. A Survey of Zero-Shot Learning. *ACM Transactions on Intelligent Systems and Technology (TIST)* **2019**, *10*, 1 – 37.
85. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* **2014**, *15*, 1929–1958.
86. Mun, J.H.; Jeon, M.; Lee, B.G. Unsupervised learning for depth, ego-motion, and optical flow estimation using coupled consistency conditions. *Sensors* **2019**, *19*, 2459.
87. Hirschmuller, H. Accurate and efficient stereo processing by semi-global matching and mutual information. 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05). IEEE, 2005, Vol. 2, pp. 807–814.
88. Espinosa Morales, A.M.; Moure Lopez, J.C. Embedded Real-time stereo estimation via semi-global matching on the GPU. *Procedia Computer Science, 2016, vol. 80, p. 143-153* **2016**.
89. Newcombe, R.A.; Lovegrove, S.J.; Davison, A.J. DTAM: Dense tracking and mapping in real-time. 2011 international conference on computer vision. IEEE, 2011, pp. 2320–2327.
90. Wang, S.; Clark, R.; Wen, H.; Trigoni, N. Deepvo: Towards end-to-end visual odometry with deep recurrent convolutional neural networks. 2017 IEEE international conference on robotics and automation (ICRA). IEEE, 2017, pp. 2043–2050.
91. Wang, P.; Shen, X.; Lin, Z.; Cohen, S.; Price, B.; Yuille, A.L. Towards unified depth and semantic prediction from a single image. Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 2800–2809.
92. Vijayanarasimhan, S.; Ricco, S.; Schmid, C.; Sukthankar, R.; Fragkiadaki, K. Sfm-net: Learning of structure and motion from video. *arXiv preprint arXiv:1704.07804* **2017**.

93. Wang, Y.; Xu, Y.F. Unsupervised learning of accurate camera pose and depth from video sequences with Kalman filter. *Ieee Access* **2019**, *7*, 32796–32804.
94. Mavrogiannis, A.; Chandra, R.; Manocha, D. B-gap: Behavior-guided action prediction for autonomous navigation. *arXiv preprint arXiv:2011.03748* **2020**.
95. Tesla, D. miles of full self driving, tesla challenge 2, autopilot,”, 25.
96. Wang, L.; Sun, L.; Tomizuka, M.; Zhan, W. Socially-compatible behavior design of autonomous vehicles with verification on real human data. *IEEE Robotics and Automation Letters* **2021**, *6*, 3421–3428.
97. Jayaraman, S.K.; Tilbury, D.M.; Yang, X.J.; Pradhan, A.K.; Robert, L.P. Analysis and prediction of pedestrian crosswalk behavior during automated vehicle interactions. 2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020, pp. 6426–6432.
98. Saleh, K.; Abobakr, A.; Nahavandi, D.; Iskander, J.; Attia, M.; Hossny, M.; Nahavandi, S. Cyclist intent prediction using 3d lidar sensors for fully automated vehicles. 2019 IEEE Intelligent Transportation Systems Conference (ITSC). IEEE, 2019, pp. 2020–2026.
99. O’Mahony, N.; Campbell, S.; Carvalho, A.; Krpalkova, L.; Hernandez, G.V.; Harapanahalli, S.; Riordan, D.; Walsh, J. One-shot learning for custom identification tasks; a review. *Procedia Manufacturing* **2019**, *38*, 186–193.
100. Grigorescu, S.M. Generative One-Shot Learning (GOL): A semi-parametric approach to one-shot learning in autonomous vision. 2018 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2018, pp. 7127–7134.
101. Haddad, M.; Sanders, D.; Langner, M.C.; Tewkesbury, G. One shot learning approach to identify drivers. IntelliSys 2021. Springer, 2021, pp. 622–629.
102. Ullah, A.; Muhammad, K.; Haydarov, K.; Haq, I.U.; Lee, M.; Baik, S.W. One-shot learning for surveillance anomaly recognition using siamese 3d cnn. 2020 International Joint Conference on Neural Networks (IJCNN). IEEE, 2020, pp. 1–8.
103. Wang, Y.; Yao, Q.; Kwok, J.; Ni, L. Generalizing from a few examples: A survey on few-shot learning. arXiv 2019. *arXiv preprint arXiv:1904.05046* **1904**.
104. Liu, S.; Tang, Y.; Tian, Y.; Su, H. Visual driving assistance system based on few-shot learning. *Multimedia Systems* **2021**, pp. 1–11.
105. Majee, A.; Agrawal, K.; Subramanian, A. Few-shot learning for road object detection. AAAI Workshop on Meta-Learning and MetaDL Challenge. PMLR, 2021, pp. 115–126.
106. Stewart, K.; Orchard, G.; Shrestha, S.B.; Neftci, E. Online few-shot gesture learning on a neuromorphic processor. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems* **2020**, *10*, 512–521.
107. Sharma, C.; Kaul, M. Self-supervised few-shot learning on point clouds. *Advances in Neural Information Processing Systems* **2020**, *33*, 7212–7221.
108. Ranga, A.; Giruzzi, F.; Bhanushali, J.; Wirbel, E.; Pé rez, P.; Vu, T.H.; Perotton, X. VRUNet: Multi-Task Learning Model for Intent Prediction of Vulnerable Road Users. *Electronic Imaging* **2020**, *32*, 109–1–109–10. doi:10.2352/issn.2470-1173.2020.16.avm-109.
109. Campbell, S.; O’Mahony, N.; Krpalcova, L.; Riordan, D.; Walsh, J.; Murphy, A.; Ryan, C. Sensor technology in autonomous vehicles: A review. 2018 29th Irish Signals and Systems Conference (ISSC). IEEE, 2018, pp. 1–4.
110. Janai, J.; Güney, F.; Behl, A.; Geiger, A.; others. Computer vision for autonomous vehicles: Problems, datasets and state of the art. *Foundations and Trends® in Computer Graphics and Vision* **2020**, *12*, 1–308.
111. Liu, S. *Engineering autonomous vehicles and robots: the DragonFly modular-based approach*; John Wiley & Sons, 2020.
112. Bagloee, S.A.; Tavana, M.; Asadi, M.; Oliver, T. Autonomous vehicles: challenges, opportunities, and future implications for transportation policies. *Journal of modern transportation* **2016**, *24*, 284–303.
113. Rupp, J.D.; King, A.G. Autonomous driving-a practical roadmap. Technical report, SAE Technical Paper, 2010.

114. Bazilinskyy, P.; Kyriakidis, M.; de Winter, J. An international crowdsourcing study into people's statements on fully automated driving. *Procedia Manufacturing* **2015**, *3*, 2534–2542.
115. Kyriakidis, M.; Happee, R.; de Winter, J.C. Public opinion on automated driving: Results of an international questionnaire among 5000 respondents. *Transportation research part F: traffic psychology and behaviour* **2015**, *32*, 127–140.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.