AUTONOMOUS DRIVING USING AI-ML

Siddhi Manglam[1] Sumedha Poonia[2]
Author(s) Contact Details
Student of VIT Bhopal University

1. Abstract

The realm of autonomous driving is rapidly growing and revolutionizing the transportation sector. In this technology, artificial intelligence (AI) and machine learning (ML) play a vital role by enabling self-driving cars to independently navigate and make informed choices. Progress in AI and ML has significantly contributed to the advancement of autonomous driving systems, leading to notable improvements in perception, decision-making, and vehicle control. [1]

This paper is a review of research in autonomous driving using AI ML.

Examined recent developments in AI and ML that have been used to improve autonomous driving in this study. We go over the difficulties involved in creating autonomous driving systems, including the requirement for trustworthy control systems and robust perception and decision-making algorithms. We also examine some of the popular methods for creating autonomous vehicle systems, such as images, videos, lidar and radar sensor data, GPS and IMU data, and other relevant sensor data . collected from vehicles driving in various environments and conditions. [1]

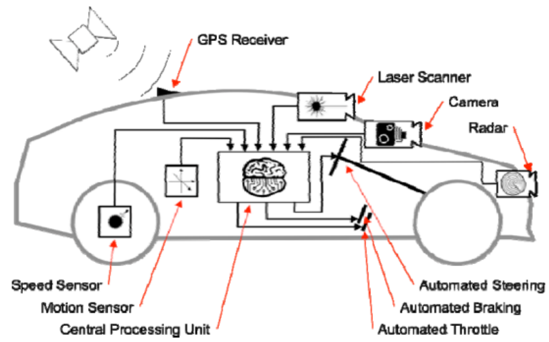

Figure 1: Basic layout of driver assistance system.
[1]

Figure 1 illustrates sensors responsible for detecting the environment and the vehicle's position, accompanied by computer algorithms that interpret the sensor data. These algorithms identify obstacles, categorize situations, plan a suitable route, and trigger actuators. As a result, the system is able to steer, brake, and control the car, or at the very least, warn the driver and give them important information. [1] The following definitions describe the various levels of system integration:

Warning and Information - In some circumstances, a passive system aids the driver in controlling the vehicle.
Examples include lane departure warning, park distance information, and navigation systems.[1].

Assisted Driving involves automating specific driving tasks for particular use cases. For instance, examples include Adaptive Cruise Control, Heading Control, and Lane Change Assistance.

Automated Driving takes automation a step further, encompassing all driving tasks for a specific use case. Examples of this are

Automated highway systems and Automated Parking.

Autonomous Driving represents the highest level of automation, where all driving tasks are automated for all use cases. A prominent example is the deployment of fully Autonomous vehicles. [1].

Keywords : Autonomous, artificial intelligence, machine learning, navigation, GPS, IMU.

2. Introduction

Autonomous driving is a complex task that requires advanced artificial intelligence (AI) and machine learning (ML) models. Here are some components of AI and ML models used for autonomous driving:

2.1 Computer Vision: Computer vision algorithms play a crucial role in identifying various objects like other vehicles, pedestrians, and road signs. These algorithms commonly rely on deep learning methods, such as convolutional neural networks (CNNs), which help extract relevant features from the images captured by cameras.[2]

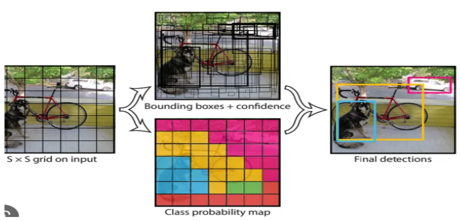

Figure 2 : Computer vision application in autonomous driving vehicle
[4].

Figure 2 illustrates Computer Vision's primary objective, which centers on emulating the intricate workings of the human visual system. It aims to empower machines or computers with the capability to recognize and interpret various objects in videos and images, akin to human perception.

As of now, there are two key models used in vision based autonomous systems, namely Mediated perception and Behaviour Reflex approach. A third significant one was introduced by Princeton university researchers in 2015, namely Direct Perception approach. While mediated perception methods analyze the entire scene to reach a driving decision, behavior reflex methods directly associate an input image with a driving action using a regressor. The novel approach, known as Direct Perception, strives to map an input image to a concise set of crucial perception indicators that directly correspond to the traffic state's suitability for driving. It is proposed to fall in between the aforementioned major models, and provide a balanced level of required abstraction. To illustrate this, researchers trained a deep Convolutional Neural Network using 12 hours of human driving recordings from a video game. Initial evaluations indicate that the model performs effectively across a wide range of virtual environments, simulating a car's behavior convincingly.
[18].

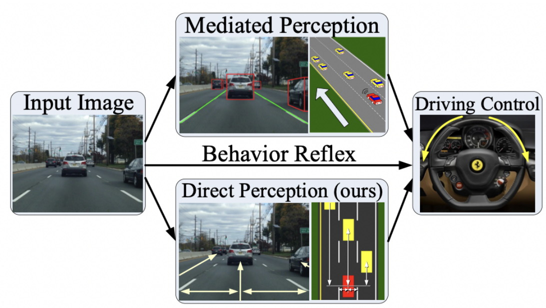

Figure 3 : Three paradigms of autonomous driving
[18]

Fig. 3 represents the three paradigms of autonomous driving as illustrated in the above paragraph. Within mediated

perception approaches, there are various components dedicated to the identification of pertinent objects such as traffic lights, signs, vehicles, and pedestrians. All of this information is then fed to an AI based engine which will then make a decision. However, this introduces additional intricacy and expense to an already challenging endeavor of accurately parsing objects in a scene, as only a small fraction of the detected objects hold significance for decision-making.
On the other hand, behavior reflex approaches establish a straightforward connection between sensory input and driving actions. To acquire knowledge, the system captures images and corresponding steering angles while a human drives along the road. This model, though, struggles to deal with traffic and other complications for a variety of reasons. Diverse drivers might arrive at entirely distinct decisions in identical situations, thus complicating regressor training. The low-level decision making of behavior reflex introduces an abstraction level that might fail to fully grasp the actual situation. Lastly, the model must tackle the challenge of identifying the relevant components within the entire image for decision making.
The Direct Perception approach's proposed system focuses on acquiring a mapping from an image to meaningful affordance indicators of the situation, encompassing essential relative angles and distances. It utilizes the ConvNet framework to learn image features for estimating these affordances. In contrast to conventional mediated perception methods, the Direct Perception approach boasts a simpler structure, yielding concise and task-specific affordances. The training set was built using recordings and screenshots of 12 hours of play of a car racing video game, namely TORCS by a human driver and using KITTI dataset.
To achieve the mapping from affordance to action, with the objective of minimizing the deviation between the car's current position and the lane's center line, the steering control is computed by taking into account the car's position and pose:

$$steerCmd = C * (angle - dist\_center/road\_width)$$

In this context, *dist_center* represents the distance from the car to the center of the lane, and *C* denotes a coefficient that adapts to different driving conditions. The angle ε lies within the range of $[-\pi, \pi]$.

At each step, the system computes desired speed as follows:

$$v(t)=v_{max}(1-exp(-c/v_{max}dist(t)-d))$$

dist(t) signifies the distance to the preceding car, while v_{max} denotes the maximum permissible speed. Additionally, *c* and *d* are coefficients that need to be calibrated.

This approach shows promising results in estimating affordances for driving actions instead of analyzing complete scenes or directly mapping the entire image to driving commands without discerning relevant indicators.
[18]

Moving on to the next persisting issue in computer vision, we look at the accuracy in detecting objects and taking intelligent decisions on local roads, unmarked lanes and places with unclear vision such as parking lots and unpaved roads. CNNs are predominantly utilized for pattern recognition by manually breaking down tasks into specific components like road and lane marking detection, semantic abstraction, path planning, and control. These components are often chosen for their ease of human interpretation, but such an approach does not inherently ensure optimal system performance.In a recent project conducted in 2016, a CNN was trained to directly correlate raw pixels obtained from a single front-facing camera with corresponding steering commands. In this system, the internal representation of essential processing steps is automatically learned without explicit training, using only the human steering angle as the training signal. Consequently, the systems learn to solve the problem with minimal steps, enabling the use of smaller networks. This project, DAVE 2, is built on DARPA autonomous vehicle (DAVE), with the help of Torch 7 machine learning library. [19]

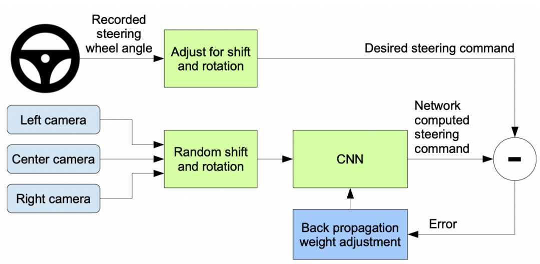

Figure 4: Training the neural network [19]

As depicted in Figure 4, a CNN processes images and computes a suggested steering command. This suggested command is then compared to the desired command for that specific image, and the CNN's weights are adjusted to minimize the difference between the output and the desired output. The weight adjustment is achieved using backpropagation.
The data for the same was collected on a diverse set of roads from around north-western USA, totalling 72 hours. The network's weights were trained by minimizing the mean squared error between the steering command generated by the network and the command provided by the human driver. The network architecture consists of a normalization layer, followed by 5 convolutional layers and 3 fully connected layers, resulting in a total of 9 layers. To process the input image, it was split into YUV planes before being passed through the network.
In simulation tests, the autonomy of the system can be sought.
Actual intervention would take nearly 6 seconds, including the time taken by the driver to retake control, re-center it and restart the self-steering mode.
This demonstrates CNN's capability to learn the complete task of lane and road following without requiring manual decomposition. It can derive meaningful road features from a relatively sparse training signal, although further efforts are necessary to enhance the model's accuracy and robustness.[19]

As the process advances, there arises a necessity to address street classification, vehicle detection, and road segmentation simultaneously in a single forward pass. A proposed solution to this challenge involves a unified architecture where the encoder is shared among the three tasks. This approach aims to achieve joint classification, detection, and semantic segmentation, streamlining the process and enhancing efficiency.
The MultiNet approach is achieved by integrating all three tasks into a unified encoder-decoder architecture, where the encoder is a deep CNN generating rich features shared among all tasks.
Task-specific decoders then utilize these features to produce real-time outputs for each task. This straightforward approach

allows for end-to-end training and yields impressive performance on the KITTI dataset, surpassing the state-of-the-art in road segmentation, all while taking only 42.28 ms to execute all tasks. Additionally, compression techniques can be employed to reduce computational bottlenecks and lower energy consumption, further enhancing the system's efficiency.
[20]

To enhance accuracy, a deep CNN network has been suggested for land cover mapping in remote sensing images, particularly emphasizing urban areas. In remote sensing tasks, smaller objects are typically given less priority to achieve higher overall accuracy. Various deep architectures are utilized, such as patch-based and pixel-to-pixel approaches, or a combination of both, where each pixel in a set of aerial images is classified individually.[21]
In this model, the two approaches are integrated by combining the softmax probabilities from different methods. This combination is achieved by training one-versus-all linear Support Vector Machines (SVMs) on the combined softmax probabilities. The ISPRS dataset was used for training, and the combined model yielded the best accuracy. It performed exceptionally well, particularly in accurately identifying small objects, while maintaining high overall accuracy. The model achieved an overall classification accuracy of 87%. Notably, for the small object class "car," it obtained an F1 score of 80.6%, surpassing the state-of-the-art performance for this dataset.
[21]
Another possible high value approach could be that of uncertainty maps. [22]

2.2 LiDAR: Light Detection and Ranging (Lidar) sensors use laser beams to measure distances to objects around the vehicle . (Li, Y., & Ibanez-Guzman, J. ,2020)[5].
These sensors generate a 3D point cloud of the environment, which can be processed using machine learning techniques such as clustering algorithms to detect objects.

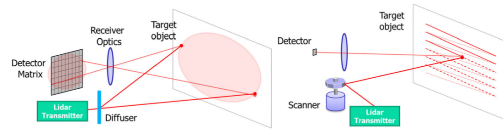

Figure 5: Structure of lidar system
[5].

In Figure 5, the emphasis is on a compact area, making the sensors ideal for long-distance and high-resolution imaging. The scanning lidar sensor directs a collimated beam towards the target, illuminating only a small portion of the object at a time. This approach allows for detailed and precise imaging of the target area.

By firing laser beams and then determining the reflected information they contain, LiDAR sensors can gather finely detailed 3D structures of a scene. Each pixel of the range picture comprises the reflective intensity, azimuth, and range in the spherical coordinate system. LiDAR sensors generate range images as their raw data format, which represents the distances to objects in the environment. To create a more structured representation, these range images can be transformed into point clouds by converting the spherical coordinates (azimuth, elevation, and range) into Cartesian coordinates (x, y, and z). The resulting point cloud is denoted as Ipoint RN3, where N refers to the number of points in the scene, and each point is represented by three coordinates (x, y, and z). Point clouds are

widely used in various applications such as 3D reconstruction, object detection, and mapping.
Range pictures and point clouds acquired directly by LiDAR sensors contain accurate 3D data. In contrast to cameras, LiDAR sensors excel at precise object localization in three-dimensional space and are less influenced by temporal and weather variations. However, their higher cost compared to webcams might restrict their usage in driving-related contexts.
[23].
The laser beams used for ground segmentation via a convolutional neural network are represented as channels in a multi-channel 2D signal, with the rotation angle on the horizontal axis and channels on the vertical axis. The authors initially predict road height and segmentation in a top-view 3D occupancy grid of LiDAR point cloud using deep learning. This allows them to perform road detection solely with LiDAR data in a top-view representation. To enhance object detection with geometric and semantic priors, different transformed representations are considered, aiming to improve accuracy and efficiency. However, the authors argue that a top-view representation is preferable to a front-view representation as it simplifies path planning and vehicle control implementation.
[24].

2.3 Radar: Radar sensors use radio waves to detect objects around the vehicle. These sensors can measure the speed and direction of objects, which can be used to predict their future trajectory[3].
Decision-making: Once objects have been detected, decision-making algorithms are used to determine the appropriate action for the vehicle to take.These algorithms can learn from experience and develop over time by utilizing strategies for reinforcement learning. Figure 6 autonomous driving (represented by arrows that point outward) and the difficulties presented by the surroundings (shown by arrows that point inside)[1].

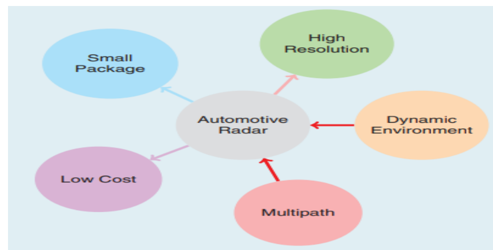

Figure 6 : Automotive radar specifications for ADAS
[3].

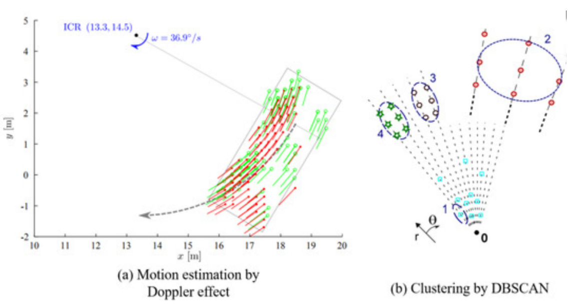

Figure 7: The effect of dynamic modeling based on radar data.[25]

The first technique calculates the velocity profile of an extended object using the Doppler data from two 77 GHz automotive radar sensors. RANSAC filters the outliers that do not belong to the same object. Then, a dynamic object that has been extended is captured in its complete 2D-motion state (yaw rate, longitudinal and lateral speed). The instantaneous center of rotation (ICR) associated parameters (,x0,y0) are calculated by velocity profile analysis. The target's size (w,h) and movement are then deduced. Due to its short processing cycle of 50 ms, this algorithm functions well in real time. It is also incredibly durable because the algorithm can withstand systematic signal changes and white noise. When the item, however, cannot be clearly derived from the data of a single scan, such methods tend to fail.

The second technique uses DBSCAN (Density Based Spatial Clustering of Applications with Noise) to estimate the extended information of targets, such as dimension and direction, and to cluster the original point clouds. DBSCAN is a popular clustering technique for handling the MMW radar's original point cloud. Both the hierarchical clustering approach and the partitioned clustering method cannot be used to handle radar data due to the radar point cloud's extreme sparsity, extensive clutter, and nonuniform density. DBSCAN, on the other hand, is adaptable to handle the issues mentioned above. The clustering issues brought on by poor angular resolution are resolved by the grid-based DBSCAN method using r grid modeling. Doppler velocity is applied to aid strengthening the clustering effect and on this premise, an adaptive clustering method for tracking is introduced to further enhance algorithm realizability. The methods listed above are suitable for high-resolution radar data.[25]

2.4 Localization: In order to navigate the surroundings, autonomous vehicles require a precise understanding of their location. To determine the position and orientation of the vehicle, localization algorithms make use of sensors like GPS, LiDAR, and cameras. [7]

Any additional perception or planification duties require the ability to localize a vehicle, whether in a global or local frame. Some classical approaches to localization are as discussed.

One is the Global Navigation Satellite System (GNSS), which however, despite the inherent accuracy cannot perform fully in dense urban areas as it does in open roads because tall buildings can mask satellites. These satellite signals continue to be a problem since they are impacted by unpredictable atmospheric conditions. The infrastructure may prevent direct signal reception and instead provide multipath interference or non-line-of-sight reception, both of which have terrible effects on the localisation that is being provided.

Another approach is to take advantage of road infrastructure to guide a vehicle in lane. Such lane marking detection is being integrated in commercialized cars with Advanced Driver-Assistance Systems (ADAS). This approach, however, is only sufficient in environments where the roadways are easily identifiable such as highways. It constraints the lateral position of the vehicle and is not suitable for more urbane and complex environments. [26]

After examining various options for localization in dense urban areas using 3D lidar data combined with GNSS and inertial data using a delivery robot, assuming that a point cloud map of the operation area is provided, 3D scan matching emerges as the best method when combined with IMU data inside of an unscented Kahman filter and GNSS data.[29]

2.5 Mapping: Lane markings and speed restrictions are among the additional details about the surroundings that are provided by high-definition maps. This data can be gleaned from sensor data using machine

learning techniques, which will update the map in real-time.
In order to detect the environment, make judgments, and direct the movement of the car, autonomous driving requires a combination of AI and ML models. [7]

Localization and mapping are closely related to one another. The Simultaneous Localization and Mapping (SLAM) framework aims to offer the best answer to problems involving localization and mapping. When considering autonomous driving as a whole, the map is of utmost importance because it offers the initial level of awareness required to arrive at the right conclusions.

The framework is general enough to allow the use of any sensor or estimation technique that suits the prerequisite of estimating both the localization and map at the same time. The maturity of such methods for autonomous driving differs vastly depending on the way data is handled (centralized or decentralized manner) [26].

Military, cartographic, search and rescue, and 3D scene capture are just a few of the uses for SLAM, which uses a robotic agent to map an unfamiliar environment while finding itself on a created map. [27]

SLAM algorithms often have front and back ends. While the back end uses dynamics and measurement models for inference over the processed data and generates compatible state estimates, the front end performs feature extraction, data association, and outlier identification on the raw sensor data. Backend algorithms often fall into one of two categories: batch optimization-based or Gaussian filtering-based. While optimisation methods iteratively estimate states as a solution to an optimisation problem with an objective built from an interior measurement unit (IMU), filtering methods iteratively refine the distribution of recent states under a Guassian prior.[28]

3. Dataset
The type of dataset required for autonomous driving using AI/ML would typically include large amounts of high-quality data such as images, videos, lidar and radar sensor data, GPS and IMU data, and other relevant sensor data collected from vehicles driving in various environments and conditions. [7]

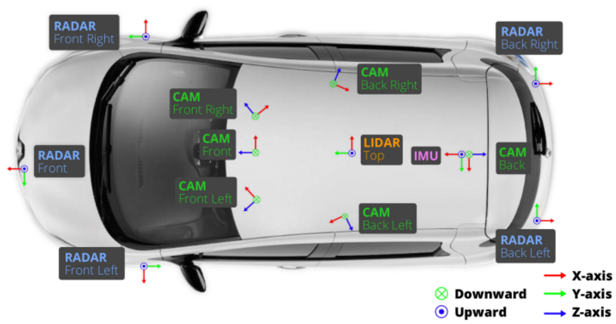

Figure 8 : Sensor setup for our data collection platform. [4]
Some of the key features of dataset that are present in Figure 8 are:

- **A wide variety of scenarios:**The dataset should include a variety of road and weather conditions, lighting conditions, and other scenarios that the autonomous vehicles may encounter in the real world. For example, heavy rain, fog, snow, strong winds, varying amounts of daylight, sunrise and sunset, urban environments with heavy traffic, rural areas with narrow or winding roads, construction zones, pedestrian crossings, cyclist interactions, complex intersections, highway driving, parking challenges, and road signs. The algorithms of the autonomous vehicles are thoroughly tested and made ready to manage the complexities and uncertainties of real-world driving circumstances thanks to the diverse range of scenarios. [8]

- **Annotation:**For training and developing autonomous driving systems or computer vision applications, the data should be properly annotated with precise and thorough labels indicating the position and orientation of vehicles, pedestrians, traffic signs, lane markings, and other pertinent objects. [8]

- **High resolution:** As they offer finer details and a more thorough understanding of the visual environment, high-resolution images and sensor data are crucial for training the AI/ML models to accurately detect and classify objects in the scene, allowing the algorithms to make more accurate and informed decisions. The rich textures, subtle color variations, and detailed object shapes captured by these high-resolution inputs enable the models to distinguish between related objects and precisely identify their properties. Additionally, the model's perception abilities are improved by detailed sensor data by supplying additional contextual information that helps with the precise localization and identification of objects in complicated settings, such as depth information or multispectral photography. AI/ML models can be taught to achieve better standards of accuracy and robustness with access to high-quality and complete data. [8]

- **Realistic simulation:** The performance of the AI/ML models can be trained and validated using realistic simulations of various driving scenarios. This enables engineers to test the algorithms in challenging environments without endangering the safety of actual drivers and pedestrians. These simulations offer a thorough testing environment for assessing the resilience and flexibility of the AI/ML models because they can precisely mimic various road conditions, weather patterns, traffic scenarios, and even unforeseen events. Developers can fine-tune and optimize their algorithms so they are ready to manage the complexities and uncertainties of real-world driving by exposing the models to a variety of simulated scenarios. Additionally, the collecting of enormous amounts of labeled data is made possible by simulated settings, which speeds up training and improves the models' capacity for generalization and make accurate predictions in real-time situations.[8]

- **Large sample size:** The dataset should be large enough to provide a sufficient amount of data for training and validation, as well as for testing the performance of the trained models in different scenarios.

There are various publicly available datasets that can be used for autonomous driving research, such as the KITTI dataset, the Waymo Open Dataset, the ApolloScape dataset, the Udacity Self-Driving Car dataset, and others. Additionally, companies working on autonomous driving technology may also collect their own proprietary datasets. [8].

A successful self-driving and widely applied vehicle must comply with three critical components- understanding the environment, understanding self-location and understanding semantics in view. Prevailing approaches for solving these tasks are mostly LIDAR dependent. [30]

Vision based approaches, which might have potentially very low costs, on the other hand, are quite challenging and under research. These require solving tasks like visual 3D scene reconstruction, self localization, semantic parsing, semantic instance understanding, object 3D instance understanding, etc.
[31][32] [33]
We discuss three popular datasets below.

KITTI Dataset
One of the first, this dataset was recorded while being driven through Karlsruhe, Germany in a VW station wagon. A number of sensor modalities, including high resolution color and grayscale stereo cameras, a Velodyne 3D laser scanner, and a very accurate GPS/IMU inertial navigation system, were used to record 6 hours of different traffic conditions at 10-100 Hz. [34]

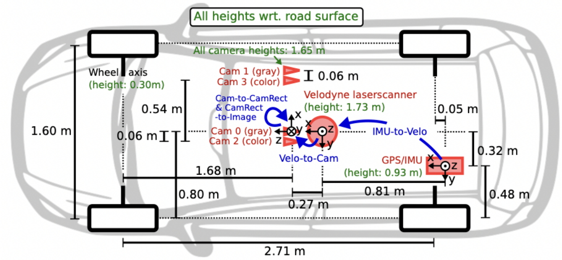

Figure 9 : Sensor setup.[34]

Figure 9 shows the sensors' measurements and mounting locations in relation to the car's body (in red). Green indicates heights above the ground, and blue indicates sensor changes.[34]

Each sensor stream's data format is saved as:
Images: color and grayscale with loss-less compression using 8-bit PNG files

a) OXTS (GPS/IMU): A text file containing 30 separate GPS/IMU data is kept for each frame.
b) Velodyne: Velodyne: scans are efficiently saved as floating point binaries that are simple to code-parse. The (x, y, z) coordinates of each point are recorded along with an additional reference value (r).

This dataset, which captures a variety of events, is calibrated, synchronized, and corrected for autonomous driving. [34]

Cityscapes Dataset
Datasets for urban scenes, like those in the KITTI Vision Benchmark Suite and CamVid, are typically smaller than datasets for more generic environments. In addition, they fall short of accurately representing the complexity and variation of real-world inner city traffic situations.
[35] [36]
Cityscapes features a broader selection of intricate inner city street sceneries that were photographed in 50 different German towns

and is specifically created for autonomous driving in urban environments. It delivers depth information through stereo vision in addition to instance level semantic tagging in annotations and evaluation metrics to help research on 3D scene understanding. This goes beyond semantic labeling at the pixel level. [36]

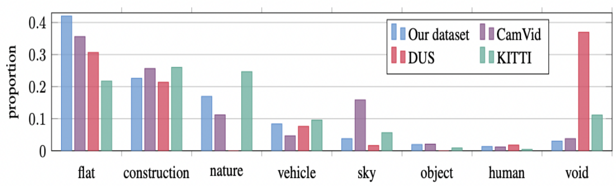

Figure 10 : Proportion of annotated pixels (y-axis) per category (x-axis) for Cityscapes, CamVid, DUS and KITTI. [36]

For systems that use huge amounts of weakly labeled data, 5,000 photos have high quality pixel level annotations and 20,000 additional images have coarse annotations. Modern techniques perform in a very different order relative to one another on Cityscapes than they do on other datasets. On KITTI and CamVid, it also reaches and even beats semantic labelling. [36]

ApolloScape Open Dataset
While Cityscapes only contains discrete semantic labeled frames without objectives like localization or 3D reconstruction, KITTI only includes 200 training images for semantic understanding or limited permutations of tasks. Apolloscape is a developing and unified dataset that improves upon earlier ones in terms of data size, label density, and task variety.
The properties of the same are as follows:

1) Dense semantics 3D point cloud for environment (20+ driving site)
2) Stereo driving videos (100+ hours)
3) High accuracy 6DoF camera pose (translation <= 50mm, rotation <= 0.015 degrees)
4) Videos at same site under different day times
5) Dense per-pixel per-frame semantic labeling (35 classes, 144k+ images)
6) Per-pixel lanemark labeling (35 classes, 160k+ images)
7) Semantic 2D instances segmentation (8 classes, 90k+ images)
8) 2D car key points and 3D car instance labeling (70K cars)

[33]

Figure 11: acquisition system. [33]
In figure 11, The acquisition system consists of two laser scanners, up to six cameras, and an integrated IMU/GNSS system, as seen. [33]

In addition to a high density 3D point cloud map, per-pixel, per-frame semantic image labeling, lane mark labeling, and semantic instance segmentation for a variety of videos, ApolloScape is a sizable, varied, and multi-task dataset for autonomous driving. Every frame is additionally geotagged using high-precision GPS and IMU sensors, with the possibility of attaching a panoramic camera system and Velodyne in the future to diversify and enhance the collection. [33]

4. Fully Self Driving Automation

Fully self-driving automation refers to a level of vehicle automation in which a vehicle can operate completely on its own without human intervention. This level of automation is often referred to as Level 5 autonomy and is the highest level of automation recognized by the Society of Automotive Engineers (SAE).

In a fully self-driving vehicle, there is no need for a driver to control the vehicle or even be present in the vehicle. The vehicle's on-board sensors, artificial intelligence algorithms, and computer systems work together to perceive the environment, make decisions, and control the vehicle's movements.

This level of automation has the potential to revolutionize transportation and make it safer, more efficient, and more convenient for people. [6]

However, there are still many technical and regulatory challenges that need to be addressed before fully self-driving vehicles become a common sight on our roads.

4.1 Unsupervised Laser Calibration

While calibrating single-beam sensors is frequently not too difficult, determining an accurate calibration for lasers that emit many simultaneous beams has proven to be a laborious and much more complex task. [6] [37]

$$J = \sum_{b_i=1}^{B} \sum_{b_j=b_i-N}^{b_i+N} \sum_{k} w_k \|\eta_k \cdot (p_k - m_k)\|^2$$

define an energy function on point clouds [37].

In the diagram above, B represents the total number of beams, N represents the number of neighboring beams we align each beam to, k iterates over the points seen by beam bj, pk represents the kth point projected according to the current transform, mk represents the point that is closest to pk seen by beam bi, k represents the surface normal at point mk, and wk is either 1 or 0.
[6] [37]

Traditionally, LIDAR sensors were equipped only with a single rotating beam, but with newer technological advances, endowment with various simultaneous rotating beams is increasingly common. [38]

This enables new applications in mapping [39], object detection and recognition [40], scene understanding [41] and SLAM [42].

New calibration techniques are necessary in order to take advantage of this rise in beam count. Extrinsic calibration, intrinsic calibration for each beam, and remittance calibrations are a few of these techniques.[38]

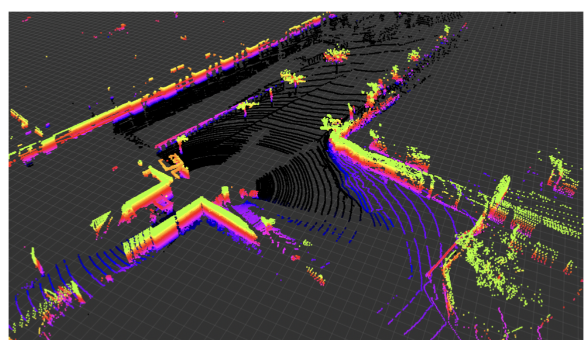

Figure 12 : Single 360 degree scan from 64 beam Velodyne LIDAR. Points are colored by height for visual clarity.
[38]

Figure 12 represents a supervised calibration method for the specified LIDAR that calls for a certain calibration target, a number of

hand measurements, and a conventional optimization phase.
[43]

Effective real-time algorithms for unsupervised calibration approaches are possible when combined with aggressive search techniques and sophisticated data trimming. Another expansion would be to jointly solve calibration, localization, and mapping problems when none are known with absolute certainty. [38]

4.2 Mapping and Localization
The localisation accuracy required for autonomous vehicle navigation in dynamic urban environments is not available with GPS-based inertial guidance systems. Although pre-driving the course was prohibited for the Urban Challenge, this is not the case in the real world, so we use Velodyne LIDAR, GPS, and IMU data to make a high-resolution infrared remittance ground map. Then, localization can be done using this ground map.
[2]

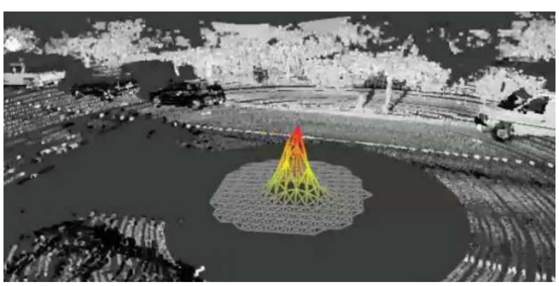

Figure 13 : A real time 2-dimensional histogram
[2]
Figure 13 Maps can be orthographic grid-based projections of LIDAR remittance returns onto the ground plane for algorithmic and computational simplicity.

As already mentioned, accurate localization in crowded metropolitan settings is quite difficult. Local motion estimate is integrated with real-time data matching against an intricately constructed map. To enable autonomous navigation and operation, several sensors can be applied in this situation.
Here, visual data provides more semantic and qualitative information
[44],[45],Despite the fact that LIDAR measurements are more precise and can describe objects from a geometric standpoint
[46]
In the past few years, SLAM algorithms have witnessed a rapid evolution. [47]
[48]

Here, data from a variety of sensors can be used to map the surroundings and instantly locate the car. This comprises wheel encoders, lidars, GNSS sensors, monocular or binocular cameras, ultrasonic sensors, or radars. It also includes inertial measurement units.
[49]. Detailed 3D maps in the form of point clouds can be generated from 3D lidars or with stereo vision. [29]

Following are some of the eminent localization methods, using different combinations of sensors:
using GNSS for localisation
The employment of a worldwide navigation satellite system is one of the oldest practices. Accuracy can be improved by using data from many satellite constellations, with the majority of data errors coming from multipath interference and atmospheric circumstances. With differential GNSS measurements, the effects of the environment and atmospheric conditions can be reduced on a greater scale.
[29]

IMU localization using GNSS

GNSS data can be easily combined with inertial data, such as accelerometer and accuracy data.
[29]

Lidar Odometry (LOAM)
In place of more traditional visual odometry methods, lidar odometry was used. From lidar data, geometrical features are derived and compared between successive scans. Since lidars can also estimate distances to distant objects with high accuracy, it can offer greater accuracy than visual-based odometry. [47]

NDT-based localization
It is an algorithm for registering 3D lidar point clouds in real time with existing high accuracy point cloud maps.

The points in lidar point clouds and point clouds on a point cloud map are not directly compared. The point cloud map is converted into a 3D normal distribution, where a variable *x* is the normal distribution. *x ~ (μ, δ)*

$$f(x) = \frac{1}{\delta\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\delta^2}}$$

μ depicts the mean of the variable distribution and δ^2, the variance.

Probability density function for multivariate normal distribution:

$$f(\vec{x}) = \frac{1}{(2\pi)^{\frac{D}{2}}\sqrt{|\sum|}} e^{-(\vec{x}-\vec{\mu})^T \sum^{-1}(\vec{x}-\vec{\mu})}$$

Where the covariance matrix is represented by, and x is the mean column.
The point cloud will now be divided into a 3D grid coordinate. Based on the grid's point distribution density, the probability distribution function (PDF) is computed for each cell. The next step is to determine the best transformation after each grid's PDF has been calculated.

The best transformation function is the transformation of the maximum likelihood function,

$$Likelihood : \theta = \prod_{k=1}^{n} f(T(\vec{p}, \vec{x_k}))$$

which is equivalent to the minimum negative logarithmic likelihood.

$$-\log\theta = -\sum_{k=1}^{n} \log f(T(\vec{p}, \vec{x_k}))$$

The next step is to use an optimisation procedure to reduce the negative logarithmic likelihood.
[29]

The NDT approach's stability when utilized alone is the main issue. Its downside is that it can be unstable on its own depending on the situation. As a result, when there is an abrupt change in location or orientation estimation, the NDT localization mechanism is reset using GNSS data.
[50]

NDT+IMU localization
This entails incorporating IMU data into the previously published NDT approach. This method improves precision while eliminating the previous method's instability. First, upon system startup or reset, an initial estimation of the vehicle location is obtained using GNSS data. Then, following initial movement, an estimation between lidar scans is provided via an unscented Kahman filter using IMU data as input. The NDT algorithm then uses the output of the

Kahman filter to compare scan results with a known map.
[29]

After conquering localization issues, the question of corrupted maps naturally comes into consideration, for it is highly impractical to assume that urban areas might not have map data that has been corrupted or outdated.The NDT algorithm can frequently become worthless due to noise. The discrepancy between the NDT localization and the GNSS and IMU positions can be used to determine when the vehicle is approaching these locations. In order to reconstruct the corrupted or missing data, lidar odometry and mapping is employed when a portion of the map cannot be matched with the most recent scans.
[29]

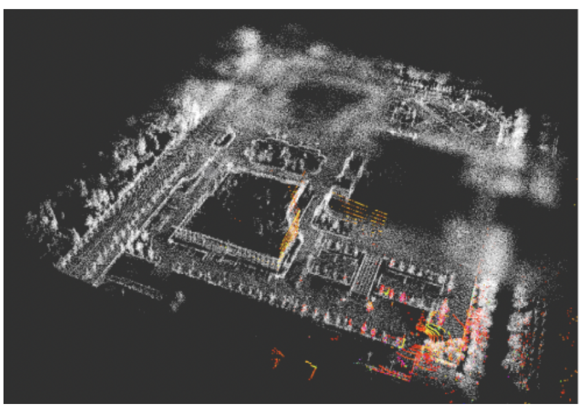

Figure 14 : Map with added noise to simulate corrupted data. [29]

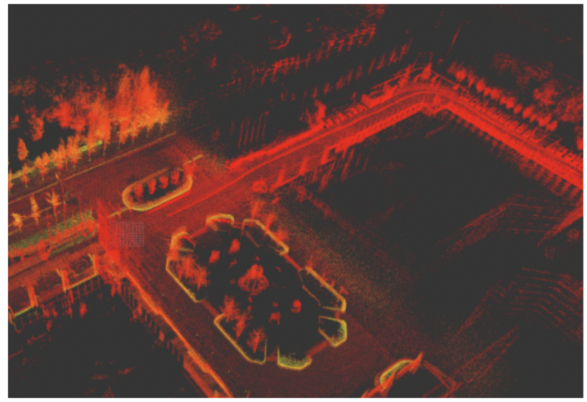

Figure 15 : Restored section of the previous map[29]

Here, a section of the map has been restored using GNSS, IMU and lidar odometry in real-time.
[29]

4.3 Object Recognition

For an autonomous vehicle to operate properly, a deeper comprehension of the environment is necessary.
For instance, knowing where pedestrians and bicycles are at junctions can help the car decide what comes first more intelligently. The object identification system in Figure 8 Junior can now identify automobiles, bicycles, and pedestrians. The main components of this system are summarized in the paragraphs that follow.
We are frequently able to categorize everything in every scan while retaining real-time operation by simply taking into account things that are close to the road and those that were previously identified as walkers, bikers, and autos.
[2]

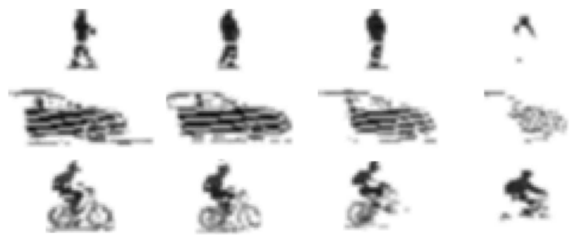

Figure 16 : Virtual orthographic camera intensity images.
[2]

Typically, there are two categories into which object detection can be divided: detection of individual instances and identification of general categories.
[51], [52]

A closed boundary, a precise pixel-by-pixel segmentation mask,[55] [56] or a bounding box can all be used to describe an object's location and size in space in a general way.
[53][54]

The main challenges to object recognition are accuracy, efficiency and scalability related challenges.
[57]

Region-based (two stages) and unified (one step) detection frameworks make up the two main types. Although one-stage detectors, such as YOLO [58][59] are typically faster than two-stage ones because of lightweight preprocessing, backbone networks, fewer candidate regions, and subnetwork fully convolutional classification, two-stage detectors produce higher detection accuracies due to more flexible structures. RCNN is one of the more popular two-stage detectors, along with [60][61][62][57]

Regardless of the number of steps, a few important decisions have come to define the design of detection frameworks:

- Fully convolutional pipeline
- Looking into supplementary data from other connected duties
- Sliding windows
- Combining data from various background layers

[57]

4.4 Trajectory Planning

As autonomous vehicles continue to evolve and adapt to real-world road conditions, they face various scenarios where they must account for the dynamics and behavior of other vehicles on the road. These possible scenarios include common driving actions including merging into traffic, passing cars while paying attention to oncoming traffic, changing lanes, and avoiding collisions with other cars. To effectively handle these situations, the concept of trajectory becomes crucial. To enable safe and effective maneuvering, trajectory planning explicitly accounts for both the planning and execution times.This approach, which is discussed in relation to self-driving cars, includes trajectory planning, allowing for the transfer of velocity and distance management to the planning level. By doing so, the autonomous vehicle can effectively navigate through complex and dynamic traffic scenarios while maintaining a high level of safety and operational efficiency.
[9]
Furthermore, the algorithm enables responsive avoidance of obstacles through the simultaneous implementation of steering and braking/acceleration.

In addition, the algorithm allows for responsive avoidance of obstacles by utilizing both steering and braking/acceleration. This proposed approach enables the breakdown of complex maneuvers into two sub-maneuvers, namely lane change and lane keeping, or a

combination of both. By primarily focusing on lane change maneuvers, trajectory planning becomes more generalized and simplified. The suggested method involves a two-step optimization-based process for planning stationary and dynamic trajectories when faced with a dynamic traffic environment. A simulation is performed to showcase the efficiency and efficacy of the proposed technique.
[87].

4.5 Traffic Light Detection

For autonomous driving in urban areas, traffic signal detection is essential. a passive camera-based pipeline for traffic light state detection that uses the geolocation of the vehicle and presupposes knowledge of the location of the traffic light. According to [9] Advanced driver assistance systems and driverless vehicles currently make extensive use of vision-based technologies. These systems don't require any additional sensors because they only rely on camera input photos. One of their primary duties is to recognise and locate all pertinent traffic lights within an image taken by a vehicle-mounted onboard camera in order to detect and comprehend traffic lights in a traffic environment.
A method for traffic light recognition that uses deep learning and adaptive thresholding for the localisation of traffic lights and region proposals. The number of available data samples is increased by using custom augmentation techniques. True and false positive rates derived from test data are used to demonstrate how well the designed system performs.
[88].

4.6 Generic sign detection

Modern autonomous vehicles still rely on humanly drawn road maps to tell them where all of the traffic signs are. A general laser-based sign detector that determines the location and orientation of every sign surrounding the autonomous vehicle is used to automate this operation. An autonomous vehicle may be able to distinguish between a 2-way stop and a 4-way stop with the aid of a direction-invariant classifier that can recognise stop signals from nonstop signs even when viewed from the side or back. [2]

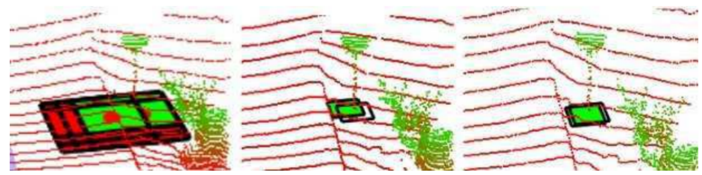

Figure 17 : Multiple detections for a single sign.
[2]
In figure 17, the classification results from several time frames are combined using a weighted voting approach to get a single classification score for each location.
Using this incoming data, a novel planning system generates thousands of candidate paths per second and selects the best one dynamically. The enhanced controller continuously chooses steering, brake, and throttle actuations that maximize comfort and reduce trajectory inaccuracy.

5. Literature Review

This review paper provides a comprehensive overview of the different AIML techniques that have been used in autonomous driving. The different stages of autonomous driving, including perception, decision-making, and control, and the different types of AIML algorithms that have been used in each stage. The paper also identifies the key challenges and future directions for AIML research in autonomous driving.
Overall, the above studies highlight the significant role that AIML plays in

autonomous driving. The papers discuss the different types of AIML techniques that have been used, including deep learning, reinforcement learning, and rule-based systems, and the different applications of AIML in autonomous driving, including perception, prediction, and control. They also identify the key challenges and future directions in each area, providing valuable insights for researchers and practitioners working in this field [1]

6. Model Creation

There are multiple steps involved in developing an AI/ML model for autonomous driving, which can be summed up as follows:

6.1 Data collection- Data collection is the initial stage in building the dataset that will be used to train the AI/ML model. Data can be collected from a variety of sources, such as sensors, cameras, and GPS systems.

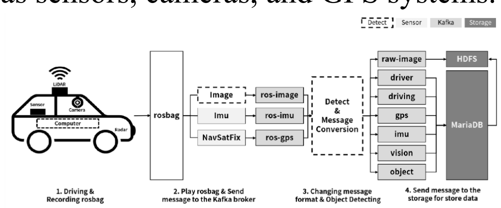

Figure 18 : Autonomous vehicle data collection process.[7]

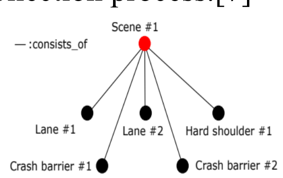

Figure 18 : First step of scene creation containing existential information
[8]

The created infrastructure scenes in figure 18 only lists the elements present in each scenario. It is necessary to arrange the instances in order to obtain a good scene description. In our approach, logical thinking begins with this phase. Elements can be positioned to the right, left, front, and back of one another.

6.2 Data Preprocessing- Data preprocessing is necessary to get rid of noise, mistakes, and useless information once the data has been gathered. Figure 20 Data normalization, data transformation, and cleanup are also components of this process.

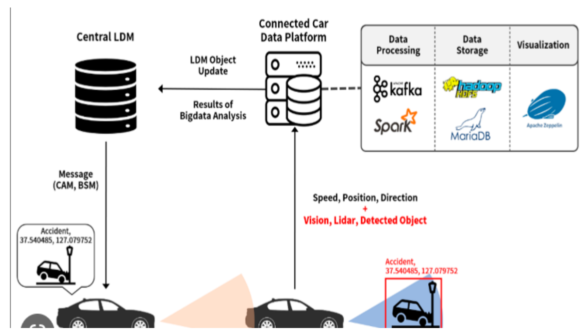

Figure 20: Sensor big data processing system for autonomous vehicles.
[7]

6.3 Feature Extraction

The following phase involves extracting pertinent features from the preprocessed data. The relevant variables that will be used to train the model are identified in this step.

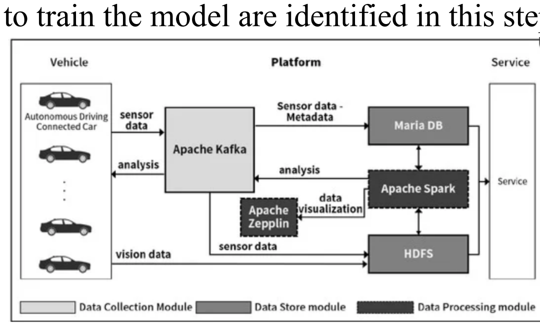

Figure 21 : Overview of the proposed C-ITS environment.
[9]

As shown in Figure 21, the vehicle system receives information from the equipment the vehicle has installed as well as driving information, and then uses real-time processing to drive the car on its own. In order to establish the foundation for cooperative driving, it also connects with nearby vehicles and the system architecture in the vehicle.

6.4 Model Selection/Visualization- Choose an appropriate machine learning method that can learn from the retrieved information and create predictions based on the issue statement.

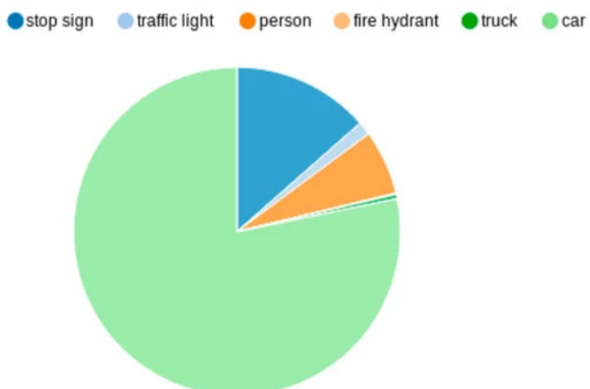

Figure 22 : Zeppelin-based data visualization. [7]

Based on the temporary view of the object table in Figure 22, users can grasp visually the different categories of items observed while driving.

6.5 Model training: To improve the performance of the chosen model, train it with the preprocessed data and the extracted features. To prevent overfitting, this stage entails fine-tuning the hyperparameters and doing cross-validation. [7]

6.6 Model Evaluation: Evaluates the performance of the model using a variety of parameters, including accuracy, precision, recall, and F1-score. This phase helps determine how effectively the model resolves the problem. [7]

6.7 Model Deployment: Use the trained model to forecast and regulate the autonomous vehicle's behavior in the real-world environment

It is important to keep in mind that creating an effective AI/ML model for autonomous driving is an iterative process that necessitates continuous assessment and improvement to improve performance and ensure the safety of the environment and the passengers.

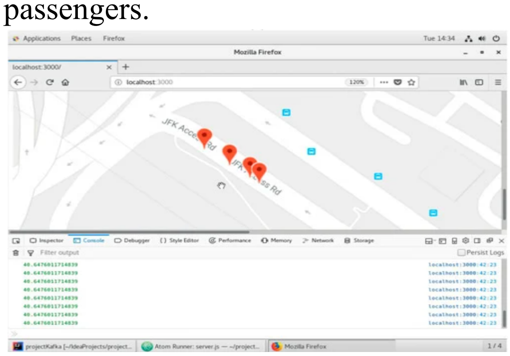

Figure 23 : web-based visualization of the messages delivered to the central LDM [10].

While these were the steps associated with model creation, let us have a look at the tasks targeted by these models, which form a backbone of how they are formulated, developed, tested and finally deployed:

Figure 24 : Tasks involved in autonomous driving
[63]

1) **Perception:** It is perhaps the most crucial task, as a system needs to perceive its surroundings accurately for proper navigation.
A deep learning-based perception technology called Simultaneous Segmentation and Detection Network (SSADNet) uses LIDAR point cloud data. It can discern between obstructions and drivable areas. In this case, segmentation branch training is performed while detection branch training is applied. The segmentation result has a classification accuracy at the pixel level of 96.9%. So it can perform both dissection and recognition simultaneously.
[64]
Modeling object identification and distance estimation can be done using a Multi-Task Learning (MTL) method that depends on Convolutional Neural Networks (CNN).
[65]
Dynamic Bayesian Networks (DBN), jointly with interaction aware motion models, is used as a risk evaluation technique which combines network level crash estimation with a risk estimate for dense, urban localities.
[66]
For robustness of semantic segmentation models, multiple lidar sensors can be used to mechanize the process and enhance the system, removing the need to put in extra efforts for manual labeling of the data used for validation.
[67]
Two adaptive driving algorithms in urban environments are Sparse Space Convolutional Neural Network (SSCNN), which uses vision deep learning and Sensor-Weighted Integration Field (SWIF), which uses an algorithm for sensor integration.
[68]

2) **Motion Planning:** Two major categories of the methods for designing algorithms for motion planning are Sample-based technique, Artificial Potential Field (APF).
[69]
The APF builds the possible field of vehicle motion based on target pull and impediment repulsion, exploring the path which is shortest to the target. The computation cost is small, but it can barely account for forthcoming driving acts.
In the sampling based method, the issue of continuous planning is converted into a distinct-and-search one. The configuration space is discretized into a sequence of trial points. Random trees are very frequently used for this purpose.
[69]
An overall scheme of an automated driving system consists of a controller, environment mapping, perception and motion planning. The ordered control system comprises Global Path Planning (GPP), decision making, local path planning and control pyramids from high to low.
[63]

3) **Pedestrian Detection:** Pedestrian recognition techniques can be categorized into two groups based on

feature acquisition: Machine Learning method based on artificial features and Deep Learning method built on Convolutional Neural Network (CNN) features.
The framework of ML techniques comprises feature extraction and classifiers. Here, features mainly include Histogram of Oriented Gradients (HOG), Local Binary Pattern (LBP), Deformable Part Model (DPM) and Aggregate Channel Feature (ACF). Classifiers include Support Vector Machine (SVM), Decision Tree, Random Forests and Ada-Boost.
The framework of DL techniques comprises a deep CNN and classifier. DL techniques have improved performance since ML techniques, but their training involves hardware with high computing powers and huge datasets.
[70]

4) **Traffic Sign Detection:** This is based on video, tracking and recognition because of the absence of Vehicle-to-Infrastructure (V2I) communication. An empirical ROI finder primarily attempts to discover all likely traffic lights, with the CNN classifier then attempting to classify the true class of every ROI, indicating the results. For an algorithm using a camera, the image processing flow is classified into three steps: pre-processing, detection and recognition. Initially, the RGB color space is transformed to HSV as a part of pre-processing. In the detection step, the mystical color threshold technique is utilized for initial filtering, meanwhile, the former data is used to examine the scene to build candidate regions.
[63]

5) **Road-marking Detection:** The detection system for road-marking contains 3 networks: feature extractor (VGG-16) having feature fusion, network detector- a fast RCNN detector comprising ROI pooling layer, classification and regression network and thirdly, Region Proposal Network (RPN).
[71]
Recognizing road marks inside images has been difficult because of the blocking of road marks by vehicles, disparity in view angle, variations in light conditions, multifaceted backgrounds and reduced image qualities. They are spotted using techniques like YOLO (You Only Look Once)- an ingenious CNN for object detection in real-time and Single-Shot Detector (SSD) or an area proposal based method.
Region-based-Convolutional-Neural-Network (R-CNN) also is a becoming a widely accepted algorithm for the same
[63]

6) **Self localization:** Localization involves approximating its location and positioning in a map. Localization data from different sources is fused using the Kahman filter. Localization can be improved by modifying probabilistic laser localization- Monte Carlo Localization (MCL), upgrading the weights of the particles by integrating the Kahman filtered GNSS data. [72]

Given a route, the first mission of an autonomous vehicle is to recognize and localize itself in the neighboring environment. The most prevalent technique is the combination of GNSS and IMU. Efforts are ongoing to surge localization correctness with map matching approaches depending on LIDAR as GNSS does not permanently have submeter correctness. [73]

7) **Automated Parking:** Using directional entering line regression and categorization on a Deep Convolutional Neural Network (DCNN), parking spaces may be found. A feature extractor and a detection head make up the detector, the feature extractor of which is based on current object detection frameworks.
Such a detector contains 3 modules:
    - round-view image fusion
    - directional entering line detector
    - parking slot implication

   [74]
   It is important to have the ability for less speedy maneuvers in constrained parking situations. Computer vision algorithms can be used for this.
   [75]

8) **Motion Control Algorithms:** For effective pathway tracking control, a Fault Tolerant Model Predictive Control (FTMPC) method is used. An improved weighted combination technique is used to combine the motion state data that numerous GPS, vision, and LIDAR sensors provide to perception systems, depending on the output fault variance.
[76]
The algorithms to manage lane switch maneuvers are divided into three separate sections.:
    - System for deciding whether to start lane switching
    - Creating a path for the vehicle to follow
    - Replanning the path in real time during the entire process

   [77]
   Vehicle trajectory prediction is also a crucial aspect of autonomous driving. Generative Adversarial Networks (GANs) can sample trajectories by employing a vehicle trajectory generator and a discriminator, differentiating real and generated trajectories.
   [78]

9) **Vehicle Cybersecurity:** The possibility of cyber attacks increases with the propagation of embedded technologies in them. Communication systems allow hackers to pick on system susceptibilities. Hence, a cyber risk categorization model incorporating identified susceptibilities can be used. Bayesian Network (BN) probabilistic models, prefaced on variables and underlying associations resulting from the Common Vulnerability Scoring Scheme [79]
Data analysis methods depending on machine learning are also used to spot abnormal behaviors due to malware.
[80]

10) **Fault diagnosis:** Fault diagnosis is the foremost method to augment vehicle security. A fault tolerant control approach is intended at longitudinal vehicle dynamics of the

automated vehicle. This includes identifying possible failures of the speedo sensor and maintaining the vehicle in a safe state. A parting principle, made of a still output feedback controller is used.
[81]
Traditional methods for Fault Diagnosis of Cyber-Physical Systems (CPSs) are primarily engrossed on physical item or intangible data model exclusively. A deliberation of both areas concurrently is needed. Fault Detection and Isolation (FDI) systems might take care of that.
[82]

7. Challenges Involved
Along with the wide array of advantages of driverless cars, there is also a long list of serious challenges we must incur in the implementation of this technology. Some of the challenges are listed below:

Cost: The current manufacturing costs turn out to be pretty high. Car manufacturers have to spend huge amounts of money for the design of EV cars. Although it is predicted that costs can be cut down to half with the possible future use of EV cars going up, the current scenario, especially in light of one of the recent models Google invested in cost $80,000, it is safe to say that manufacturing cost is one of the biggest hurdles.
[12]

Infrastructure: According to a report, it might take another 10-15 years to develop a type of infrastructure which facilitates the use of EV cars. This is in spite of various big companies like BMW, Nissan and Audi committing to launch driverless cars.[12]

Data Volume
Each autonomous vehicle is expected to generate 3-6 TB of raw data per hour of operation, depending on the sensors involved. Data in such volumes cannot be stored or transmitted at large scale. This proves to be a major issue. To reduce data while minimizing information loss, three approaches can be adopted: data filtering to meet specific classes of queries, algorithms for balancing frame quality and data volume, and alternate formats like embeddings to store images.
[83]

Bias
Due to spatial and temporal patterns occurring in traffic, the data is bound to have spatial and temporal biases. Distribution of samples and observed phenomenon may not be independent and can skew results where data is used.
[83]

Replacing Conventional Cars:
Autonomous cars would require higher efficiency and lower costs to replace currently existing vehicles. Apart from that, different types of vehicles under the same platform might lead to unexpected results.
[12]

Security and Privacy concerns:
Autonomous vehicles pose a serious concern to security and privacy breach which include but are not limited to the user's private data getting hacked which would include specifics about the user's vehicles, location based on GPS and other personal information. The collection of recorded footage for the purpose of dataset creation is also a matter of debate for being a privacy
[12]

Most people expressed “overwhelming discomfort” at the idea of AV data being used to identify and track them [84], while most felt comfortable sharing anonymized data collected by their car.
[85], [86]

8. Ethical and Legal Issues
Autonomous driving using AI/ML presents a range of ethical and legal issues that need to be addressed. These issues include safety, privacy, liability, bias, security, and employment.
[13]

Safety is a significant concern as system failures or errors can lead to accidents and fatalities.
Privacy is also a concern as the collection of data from sensors and cameras raises questions about data protection.

Liability becomes complicated in accidents involving autonomous vehicles. Bias can occur in the decision-making processes of AI and machine learning algorithms.

Security is an issue as autonomous vehicles are vulnerable to cyber-attacks. [13]

Loss to manufacturers and distributors of conventional cars

Unemployment concerns arise from the possibility that human drivers may be replaced by autonomous vehicles, which poses issues concerning how to sustain the displaced workers.
For autonomous driving employing AI/ML to be used safely and ethically, several ethical and legal challenges must be addressed.
[14].

9. Impacts
The adoption of autonomous vehicles at a large scale will lead to considerable long and short term impacts. Some of which are:

Environmentally, electric vehicles are friendlier than conventional cars and will lead to a more positive impact on our surroundings. Shortest path and lane restrictions will be considered which will inevitably lead to lesser fuel consumption [13].

Algorithm-based, naturally, considering shortest path, lane restrictions and other constraints, algorithms developed will be more optimal and might lead to spin off discoveries and inventions.
[14].

Cost: Although they cost a considerable amount right now, costs are expected to come down and with widespread usage it might actually be financially beneficial for customers to purchase EV cars.
[13].

Complete Replacement and Conversion: Widespread popularity might lead to complete replacement of conventional cars which will convert the auto landscape.
[13].

Strides in AI: Successful implementation of this technology will be a long step in AI and will encourage further strides in this field.
[14].

10. Conclusion
Autonomous and vision-guided vehicles will expand the capabilities of current auto and AI technology. It may totally alter the modes of transportation used today, giving

rise to a system that is more secure, intelligent, efficient, and environmentally friendly than the one we currently use. If used appropriately, this technology will be crucial to entering areas that would be extremely dangerous for manually driven vehicles in search operations, military transportation, and public transit.

The necessary infrastructure and architecture for self-guiding cars will take some time to develop. Manufacturers must provide comprehensive user safety, security, and privacy in addition to accounting for the benefits, and they must consider the numerous points of convention. [15]
That said, we can safely conclude that we are on the brink of a life altering revolution with the advent of autonomous vehicles, albeit all caveats are kept in mind and risks accounted for .
The concept of autonomous driving in smart cities is depicted in Figure 25. Autonomous vehicles help people in their daily lives by giving elderly and disabled persons dependable and safe transportation options, resolving parking issues, and reducing the frequency of accidents previously brought on by human mistake. [16]

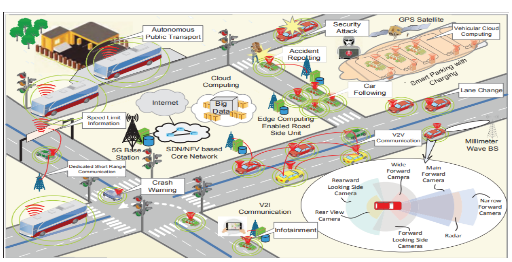

Figure 25 : Illustration of autonomous driving cars in smart cities
[16]

11. References

[1]
Yurtsever, E., Lambert, J., Carballo, A., & Takeda, K. (2020). A survey of autonomous driving: Common practices and emerging technologies. *IEEE access*, *8*, 58443-58469.

[2]
Beiker, S. A. (2012). Legal aspects of autonomous driving. *Santa Clara L. Rev.*, *52*, 1145.

[3]
Sun, S., Petropulu, A. P., & Poor, H. V. (2020). MIMO radar for advanced driver-assistance systems and autonomous driving: Advantages and challenges. *IEEE Signal Processing Magazine*, *37*(4), 98-117.

[4]
Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., ... & Beijbom, O. (2020). nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11621-11631).

[5]
Li, Y., & Ibanez-Guzman, J. (2020). Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems. *IEEE Signal Processing Magazine*, *37*(4), 50-61.

[6]
Nielsen, T. A. S., & Haustein, S. (2018). On sceptics and enthusiasts: What are the expectations towards self-driving cars?. *Transport policy*, *66*, 49-55.

[7]
Banks, V. A., & Stanton, N. A. (2019). Analysis of driver roles: Modelling the changing role of the driver in automated driving systems using EAST. *Theoretical issues in ergonomics science*, *20*(3), 284-300.

[8]

Kaur, P., Taghavi, S., Tian, Z., & Shi, W. (2021, April). A survey on simulators for testing self-driving cars. In *2021 Fourth International Conference on Connected and Autonomous Driving (MetroCAD)* (pp. 62-70). IEEE.

[9]
Nalic, D., Mihalj, T., Bäumler, M., Lehmann, M., Eichberger, A., & Bernsteiner, S. (2020, November). Scenario based testing of automated driving systems: A literature survey. In *FISITA web Congress* (Vol. 10).

[10]
Vaidya, B., & Mouftah, H. T. (2020). IoT applications and services for connected and autonomous electric vehicles. *Arabian Journal for Science and Engineering*, *45*(4), 2559-2569.

[11]
Navarro, J. (2019). A state of science on highly automated driving. *Theoretical Issues in Ergonomics Science*, *20*(3), 366-396.

[12]
Holstein, T., Dodig-Crnkovic, G., & Pelliccione, P. (2018). Ethical and social aspects of self-driving cars. *arXiv preprint arXiv:1802.04103*.

[13]
Borenstein, J., Herkert, J., & Miller, K. (2017). Self-driving cars: Ethical responsibilities of design engineers. *IEEE Technology and Society Magazine*, *36*(2), 67-75.

[14]
de Almeida Correia, G. H., & van Arem, B. (2016). Solving the User Optimum Privately Owned Automated Vehicles Assignment Problem (UO-POAVAP): A model to explore the impacts of self-driving vehicles on urban mobility. *Transportation Research Part B: Methodological*, *87*, 64-88.

[15]
Maurer, M., Gerdes, J. C., Lenz, B., & Winner, H. (2016). *Autonomous driving: technical, legal and social aspects*. Springer Nature.

[16]
Janai, J., Güney, F., Behl, A., & Geiger, A. (2020). Computer vision for autonomous vehicles: Problems, datasets and state of the art.

[17]
Foundations and Trends® in Computer Graphics and Vision, *12*(1–3), 1-308.
Sebe, N., 2005. *Machine learning in computer vision* (Vol. 29). Springer Science & Business Media.

[18]
Chen, C., Seff, A., Kornhauser, A. and Xiao, J., 2015. Deepdriving: Learning affordance for direct perception in autonomous driving. In *Proceedings of the IEEE international conference on computer vision* (pp. 2722-2730)

[19]
Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L.D., Monfort, M., Muller, U., Zhang, J. and Zhang, X., 2016. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*.

[20]
Teichmann, M., Weber, M., Zoellner, M., Cipolla, R. and Urtasun, R., 2018, June. Multinet: Real-time joint semantic reasoning for autonomous driving. In *2018 IEEE*

intelligent vehicles symposium (IV) (pp. 1013-1020). IEEE.

[21]
Kampffmeyer, M.C., Salberg, A., & Jenssen, R. (2016). Semantic Segmentation of Small Objects and Modeling of Uncertainty in Urban Remote Sensing Images Using Deep Convolutional Neural Networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 680-688.

[22]
Gal, Y. and Ghahramani, Z., 2016, June. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*(pp. 1050-1059). PMLR.

[23]
Mao, J., Shi, S., Wang, X. and Li, H., 2022. 3d object detection for autonomous driving: A review and new outlooks. *arXiv preprint arXiv:2206.09474*.

[24]
F. Yan, K. Wang, B. Zou, L. Tang, W. Li and C. Lv, "LiDAR-Based Multi-Task Road Perception Network for Autonomous Vehicles," in IEEE Access, vol. 8, pp. 86753-86764, 2020, doi: 10.1109/ACCESS.2020.2993578.

[25]
Zhou, T., Yang, M., Jiang, K., Wong, H. and Yang, D., 2020. MMW radar-based technologies in autonomous driving: A review. *Sensors*, *20*(24), p.7283.

[26]
Bresson, G., Alsayed, Z., Yu, L., & Glaser, S. (2017). Simultaneous Localization and Mapping: A Survey of Current Trends in Autonomous Driving. *IEEE Transactions on Intelligent Vehicles, 2*, 194-220.

[27]
Leonard, J.J., & Durrant-Whyte, H.F. (1991). Simultaneous map building and localization for an autonomous mobile robot. *Proceedings IROS '91:IEEE/RSJ International Workshop on Intelligent Robots and Systems '91*, 1442-1447 vol.3.

[28]
Saxena, A., Chiu, C.Y., Shrivastava, R., Menke, J. and Sastry, S., 2022. Simultaneous localization and mapping: Through the lens of nonlinear optimization. *IEEE Robotics and Automation Letters*, *7*(3), pp.7148-7155.

[29]
Li, Q., Queralta, J.P., Gia, T.N., Zou, Z. and Westerlund, T., 2020. Multi-sensor fusion for navigation and mapping in autonomous vehicles: Accurate localization in urban environments. *Unmanned Systems*, *8*(03), pp.229-237

[30]
Huang, X., Wang, P., Cheng, X., Zhou, D., Geng, Q. and Yang, R., 2019. The apolloscape open dataset for autonomous driving and its application. *IEEE transactions on pattern analysis and machine intelligence*, *42*(10), pp.2702-2719.

[31]
Kar, A., Häne, C. and Malik, J., 2017. Learning a multi-view stereo machine. *Advances in neural information processing systems*, *30*.

[32]
Schönberger, J.L., Pollefeys, M., Geiger, A. and Sattler, T., 2018. Semantic visual localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 6896-6906)

[33]
Song, X., Wang, P., Zhou, D., Zhu, R., Guan, C., Dai, Y., Su, H., Li, H. and Yang, R., 2019. Apollocar3d: A large 3d car instance understanding benchmark for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 5452-5462)

[34]
Geiger, A., Lenz, P., Stiller, C., & Urtasun, R. (2013). Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research, 32*, 1231 - 1237.

[35]
Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S. and Schiele, B., 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3213-3223)

[36]
Brostow, G.J., Fauqueur, J., & Cipolla, R. (2009). Semantic object classes in video: A high-definition ground truth database. *Pattern Recognit. Lett., 30*, 88-97.

[37]
Levinson, J., Askeland, J., Becker, J., Dolson, J., Held, D., Kammel, S., Kolter, J.Z., Langer, D., Pink, O., Pratt, V.R., Sokolsky, M., Stanek, G., Stavens, D., Teichman, A., Werling, M., & Thrun, S. (2011). Towards fully autonomous driving: Systems and algorithms. *2011 IEEE Intelligent Vehicles Symposium (IV)*, 163-168.

[38]
LEVINSON, J. S., THRUN, S., KOLLER, D., & LEVOY, M. (2011). *Automatic laser calibration, mapping, and localization for autonomous vehicles*. http://purl.stanford.edu/zx701jr9713

[39]
Levinson, J., & Thrun, S. (2010). Robust vehicle localization in urban environments using probabilistic maps. *2010 IEEE International Conference on Robotics and Automation*, 4372-4378..

[40]
Douillard, B., Brooks, A., & Ramos, F.T. (2009). A 3D laser and vision based classifier. *2009 International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP)*, 295-300.

[41]
Steinhauser, D., Ruepp, O., & Burschka, D. (2008). Motion segmentation and scene classification from 3D LIDAR data. *2008 IEEE Intelligent Vehicles Symposium*, 398-403.

[42]
Segal, A.V., Hähnel, D., & Thrun, S. (2009). Generalized-ICP. *Robotics: Science and Systems*.

[43]
Muhammad, N., & Lacroix, S. (2010). Calibration of a rotating multi-beam lidar. *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 5648-5653.

[44]
Se, S., Lowe, D.G., & Little, J. (2005). Vision-based global localization and mapping for mobile robots. *IEEE Transactions on Robotics, 21*, 364-375.

[45]
Garcia-Fidalgo, E., & Ortiz, A. (2015). Vision-based topological mapping and

localization methods: A survey. *Robotics Auton. Syst., 64*, 1-20.

[46]
Premebida, C., Garrote, L., Asvadi, A., Ribeiro, A.P. and Nunes, U., 2016, November. High-resolution lidar-based depth mapping using bilateral filter. In *2016 IEEE 19th international conference on intelligent transportation systems (ITSC)* (pp. 2469-2474). IEEE

[47]
Zhang, J., & Singh, S. (2014). LOAM: Lidar Odometry and Mapping in Real-time. *Robotics: Science and Systems*.

[48]
Zhang, J., & Singh, S. (2015). Visual-lidar odometry and mapping: low-drift, robust, and fast. *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2174-2181.

[49]
Cho, H., Seo, Y.-W., Kumar, B. V., & Rajkumar, R. R. (2014). A multi-sensor fusion system for moving object detection and tracking in urban driving environments. In 2014 IEEE International Conference on Robotics and Automation (ICRA) (pp. 1836-1843). IEEE

[50]
Wen, W., Hsu, L.-T., & Zhang, G. (2018). Performance analysis of NDT-based graph SLAM for autonomous vehicle in diverse typical driving scenarios of Hong Kong. Sensors, 18(11), 3928

[51]
Grauman, K., & Leibe, B. (2011). Visual object recognition. Synthesis Lectures on Artificial Intelligence and Machine Learning, 5(2), 1-181

[52]
Zhang, X., Yang, Y., Han, Z., Wang, H., & Gao, C. (2013). Object class detection: A survey. ACM Computing Surveys, 46(1), 10:1-10:53

[53]
Everingham, M., Gool, L. V., Williams, C., Winn, J., & Zisserman, A. (2010). The Pascal Visual Object Classes (VOC) Challenge. International Journal of Computer Vision, 88(2), 303-338

[54]
Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A., & Li, F. (2015). ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision, 115(3), 211-252

[55]
Lin, T., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, L. (2014). Microsoft COCO: Common Objects in Context. In European Conference on Computer Vision (ECCV) (pp. 740-775)

[56]
Russell, B., Torralba, A., Murphy, K., & Freeman, W. (2008). LabelMe: A database and web-based tool for image annotation. International Journal of Computer Vision, 77(1-3), 157-173

[57]
Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X. and Pietikäinen, M., 2020. Deep learning for generic object detection: A survey. *International journal of computer vision*, *128*, pp.261-318

[58]
Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 779-788)

[59]
Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C., & Berg, A. (2016). SSD: Single shot multibox detector. In European Conference on Computer Vision (ECCV) (pp. 21-37)

[60]
Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In Advances in Neural Information Processing Systems (NIPS) (pp. 91-99)

[61]
Dai, J., Li, Y., He, K., & Sun, J. (2016). RFCN: Object detection via region-based fully convolutional networks. In Advances in Neural Information Processing Systems (NIPS) (pp. 379-387)

[62]
He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV)

[63]
Bachute, M.R., & Subhedar, J.M. (2021). Autonomous Driving Architectures: Insights of Machine Learning and Deep Learning Algorithms. *Machine Learning with Applications*.

[64]
Lee, Y., & Park, S. (2020). A Deep Learning-Based Perception Algorithm Using 3D LiDAR for Autonomous Driving: Simultaneous Segmentation and Detection Network (SSADNet). *Applied Sciences*.

[65]
Chen, Y., Zhao, D., Lv, L., & Zhang, Q. (2018). Multi-task learning for dangerous object detection in autonomous driving. Information Sciences, 432, 559–571.

[66]
Katrakazas, C., Quddus, M., & Chen, W. H. (2019). A new integrated collision risk assessment methodology for autonomous vehicles. Accident Analysis and Prevention, 127(September 2018), 61–79

[67]
Zhou, W., Berrio, J. S., Worrall, S., & Nebot, E. (2019). Automated evaluation of semantic segmentation robustness for autonomous driving. IEEE Transactions on Intelligent Transportation Systems, PP, 1–13

[68]
Oh, M., Cha, B., Bae, I., Choi, G., & Lim, Y. (2020). An urban auto driving algorithm based on a sensor-weighted integration field with deep learning. Electronics (Switzerland), 9(1)

[69]
Dang, D., Gao, F., & Hu, Q. (2020). Motion planning for autonomous vehicles considering longitudinal and lateral dynamics coupling. Applied Sciences (Switzerland), 10(9)

[70]
Li, Guiyuan, Zong, C., Liu, G., & Zhu, T. (2020). Application of convolutional neural network (CNN)-Adaboost algorithm in

pedestrian detection. Sensors and Materials, 32(6), 1997–2006

[71]
Hu, J., Abubakar, S., Liu, S., Dai, X., Yang, G., & Sha, H. (2019). Near-infrared road-marking detection based on a modified faster regional convolutional neural network. Journal of Sensors, (2019)

[72]
de Miguel, Miguel Ángel, G, F., A, J. M., de Miguel, Á, M., García, F., et al. (2020). Improved LiDAR probabilistic localization for autonomous vehicles using GNSS. Sensors -MDPI, 20(Mcl).

[73]
Li, T., Qian, Y., de La Fortelle, A., Chan, C. Y., & Wang, C. (2020). Lane-level localization system using surround-view cameras adaptive to different driving conditions. International Journal of Advanced Robotic Systems, 17(2), 1–13

[74]
Li, W., Cao, H., Liao, J., Xia, J., Cao, L., & Knoll, A. (2020). Parking slot detection on around-view images using DCNN. Frontiers in Neurorobotics, 14(July), 1–9

[75]
Heimberger, M., Horgan, J., Hughes, C., McDonald, J., & Yogamani, S. (2017). Computer vision in automated parking systems: Design, implementation and challenges. Image and Vision Computing, 68, 88–101

[76]
Geng, K., Chulin, N. A., & Wang, Z. (2020). "Fault-tolerant model predictive control algorithm for path tracking of an autonomous vehicle." Sensors (Switzerland), 20(15), 1–20.

[77]
Tang, J., Yu, R., Liu, S., & Gaudiot, J. L. (2020). A container based edge offloading framework for autonomous driving. IEEE Access, 8(Vm), 33713–33726

[78]
Ortega, J., Lengyel, H., & Szalay, Z. (2020). Overtaking maneuver scenario build- ing for autonomous vehicles with PreScan software. Transportation Engineering, 2(September), Article 100029

[79]
Sheehan, B., Murphy, F., Mullins, M., & Ryan, C. (2019). Connected and autonomous vehicles: A cyber-risk classification framework. Transportation Research Part A: Policy and Practice, 124(November 2018), 523–536

[80]
Park, S., & Choi, J. Y. (2020). "Malware detection in self-driving vehicles using machine learning algorithms." Journal of Advanced Transportation, 2020

[81]
Boukhari, M. R., Chaibet, A., Boukhnifer, M., & Glaser, S. (2018). Proprioceptive sensors' fault-tolerant control strategy for an autonomous vehicle. Sensors (Switzerland), 18(6), 1–24

[82]
Yin, S., Kaynak, O., & Reza Karimi, H. (2020). IEEE Access special section editorial: Data-driven monitoring, fault diagnosis and control of cyber–physical systems. IEEE Access, 8, 54110–54114

[83]
Kazhamiaka, F., Zaharia, M.A., & Bailis, P.D. (2021). Challenges and Opportunities

for Autonomous Vehicle Query Systems. *Conference on Innovative Data Systems Research*.

[84]
Bloom, C., Tan, J., Ramjohn, J., & Bauer, L. (2017). Self-driving cars and data collection: Privacy perceptions of networked autonomous vehicles. In Thirteenth Symposium on Usable Privacy and Security (SOUPS 2017) (pp. 357–375)

[85]
de l'Automobile, F. I. (2017). What Europeans think about connected cars

[86]
Otonomo & E. Research. (2019). What American drivers think about connected car data and privacy. Retrieved from https://info.otonomo.io/connected-car-survey-edison-lp

[87]
Zhang, S., Deng, W., Zhao, Q., Sun, H. and Litkouhi, B., 2013, October. Dynamic trajectory planning for vehicle autonomous driving. In *2013 IEEE International Conference on Systems, Man, and Cybernetics* (pp. 4161-4166). IEEE.

[88]
Vitas, D., Tomic, M. and Burul, M., 2020. Traffic light detection in autonomous driving systems. *IEEE Consumer Electronics Magazine*, *9*(4), pp.90-96.