Deep Learning Sensor Fusion for Autonomous Vehicles Perception and Localization: A Review

Jamil Fayyad, Mohammad A Jaradat, Dominique Gruyer, Homayoun Najjaran

sensors

MDPI

Review

Deep Learning Sensor Fusion for Autonomous Vehicle Perception and Localization: A Review

Jamil Fayyad [1], Mohammad A. Jaradat [2,3], Dominique Gruyer [4] and Homayoun Najjaran [1,*]

1 School of Engineering, University of British Columbia, Kelowna, BC V1V 1V7, Canada; jfayyad@alumni.ubc.ca
2 Department of Mechanical Engineering, American University of Sharjah, Sharjah, UAE; mjaradat@aus.edu
3 Department of Mechanical Engineering, Jordan University of Science & Technology, Irbid 22110, Jordan
4 PICS-L, COSYS, University Gustave Eiffel, IFSTTAR, 25 allée des Marronniers, 78000 Versailles, France; dominique.gruyer@univ-eiffel.fr
* Correspondence: homayoun.najjaran@ubc.ca

Received: 16 June 2020; Accepted: 24 July 2020; Published: 29 July 2020

Abstract: Autonomous vehicles (AV) are expected to improve, reshape, and revolutionize the future of ground transportation. It is anticipated that ordinary vehicles will one day be replaced with smart vehicles that are able to make decisions and perform driving tasks on their own. In order to achieve this objective, self-driving vehicles are equipped with sensors that are used to sense and perceive both their surroundings and the faraway environment, using further advances in communication technologies, such as 5G. In the meantime, local perception, as with human beings, will continue to be an effective means for controlling the vehicle at short range. In the other hand, extended perception allows for anticipation of distant events and produces smarter behavior to guide the vehicle to its destination while respecting a set of criteria (safety, energy management, traffic optimization, comfort). In spite of the remarkable advancements of sensor technologies in terms of their effectiveness and applicability for AV systems in recent years, sensors can still fail because of noise, ambient conditions, or manufacturing defects, among other factors; hence, it is not advisable to rely on a single sensor for any of the autonomous driving tasks. The practical solution is to incorporate multiple competitive and complementary sensors that work synergistically to overcome their individual shortcomings. This article provides a comprehensive review of the state-of-the-art methods utilized to improve the performance of AV systems in short-range or local vehicle environments. Specifically, it focuses on recent studies that use deep learning sensor fusion algorithms for perception, localization, and mapping. The article concludes by highlighting some of the current trends and possible future research directions.

Keywords: autonomous vehicles; self-driving cars; deep learning; sensor fusion; perception; localization and mapping

1. Introduction

Autonomous vehicles (AVs) have made impressive technological progress in recent years; these noticeable advancements have brought the concept of self-driving cars into reality. According to a report published by the U.S. Department of Transportation, 94% of vehicle crashes occur due to driver behavior [1]. For this reason, AVs are projected to lower the risk of drastic accidents and increase road safety. Additionally, it is anticipated that AVs will assist in reducing carbon emission levels, and hence protect the environment [2]. Moreover, self-driving cars are expected to smoothen traffic flow, increase productivity, and have enormous economic impacts.

According to the Society of Automobile Engineers (SAE) [3], there are six different levels of automated vehicles, starting from level 0 where the driver is in full control of the vehicle, and ending with level 5 where the vehicle is in full control of all driving aspects. These levels are portrayed in Figure 1. Currently, it can be confidently stated that levels 2 and 3 are being adopted in some of the commercial cars, such as GM's Cruise [4], Tesla's Autopilot [5], and BMW [6]. Several autonomous features are already being performed in these cars, such as adaptive cruise control, automatic braking, and lane-keeping aid systems.

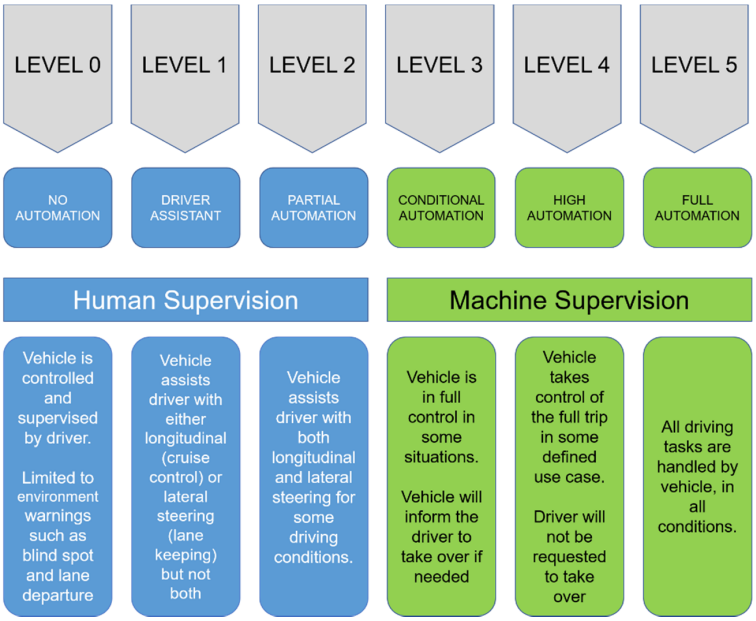

Figure 1. The six levels of autonomous vehicles as described by the Society of Automobile Engineers (SAE) [3], their definitions, and the features in each level.

Although different AV systems may differ slightly from one to another, they all need to present a solution for the autonomous navigation problem, which is generally divided into four main elements: perception, localization and mapping, path planning, and control. In perception, the vehicle utilizes a group of onboard sensors to detect, understand, and interpret the surrounding environment, including static and dynamic obstacles, such as other moving vehicles, pedestrians, road signs, traffic signals, and road curbs. Localization and mapping tasks attempt to locate the vehicle globally with respect to world coordinates. Additionally, they are responsible for building a map of the vehicle's surroundings and continuously tracking the vehicle's location with respect to that map. Subsequently, path planning exploits the output of the previous two tasks in order to adopt the optimal and safest feasible route for the AV to reach its destination, while considering all other possible obstacles on the road [7]. Lastly, based on the selected path, the control element outputs the necessary values of acceleration, torque, and steering angle for the vehicle to follow that selected path [8]. Additionally, multiple studies consider adding connected vehicle technologies [9,10], such as vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) technologies, where essential information is shared to create an enhanced cooperative driving environment, as shown in Figure 2. This extended and improved cooperative perception allows vehicles to predict the behavior of the key environmental components (obstacles, roads, ego-vehicles, environment, driver behavior) efficiently and to anticipate any possible hazardous events.

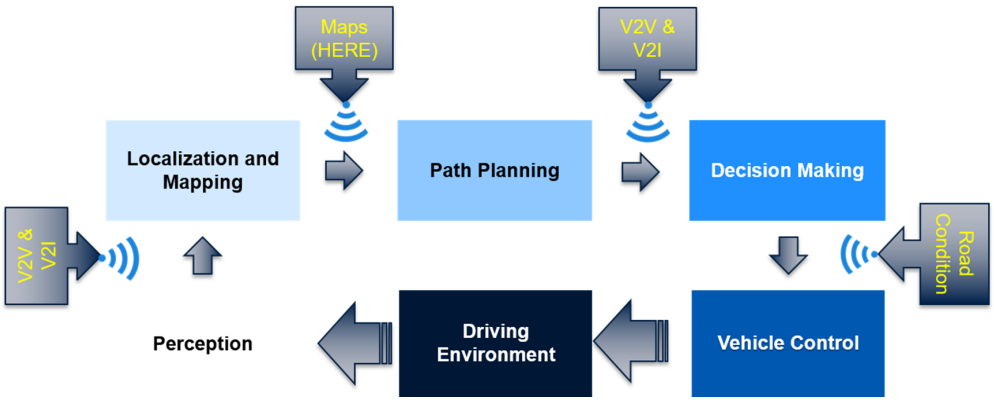

Figure 2. Full autonomous navigation system. Sensor technology and sensor fusion overview. V2V, vehicle-to-vehicle; V2I, vehicle-to-infrastructure.

One of the major considerations in any AV system is the selection of the proper group of sensors and their optimal configuration, which will be utilized to mimic the human ability to sense and create a reliable picture of the surroundings. It is always important to take into consideration the advantages, disadvantages, and limitations of this group of sensors as a whole, i.e., a logical and smart sensor. In many cases, the overall performance of the system is substantially improved when multiple sensors operating on different wavebands are placed to collaborate and produce a fused output. Consequently, sensor fusion is a vital process that is required in all AV systems to overcome the limitations of individual sensors, and hence improve the efficiency of the overall AV system.

Presently, there is an enormous amount of effort invested in improving the performance, reliability, robustness, and accuracy of self-driving vehicles modules, not to mention the cybersecurity and safety operating issues that can also be critically important under real driving conditions. While keeping in mind that vehicles are present in an environment that is highly complex, fast, and dynamic, the applied algorithms should be crafted in a special way that balances accuracy and fast real-time processing. With the emergence of new powerful computational technologies, such as graphics processing units (GPUs) and the availability of a large amount of data (so-called "big data"), a subset of artificial intelligent (AI) and machine learning known as deep learning has gained huge popularity in several applications related to object detection, object identification, road situation recognition, and more generally robotics issues [11]. Deep learning algorithms have been utilized in different aspects of AV systems, such as perception, mapping, and decision making. These algorithms have proven their ability to solve many of these difficulties, including computational loads faced by traditional algorithms while maintaining decent accuracy and fast processing speed.

This review paper will focus on two components of the AV systems: perception, and localization and mapping. The main aim is to provide a comprehensive review of the most useful deep learning algorithms in the field of sensor fusion for AV systems. The paper is organized as follows. Section 2 provides an overview of the advantages of recent sensor combinations and their applications in AVs, as well as different sensor fusion algorithms utilized in the field. Section 3 describes the task of environmental perception and provides an overview of the latest deep learning detection algorithms, along with an analysis of their performance. Section 4 discusses approaches for localization and mapping, and compares different sensors used for that task. Additionally, it evaluates various applications of deep learning algorithms and compares them with the traditional algorithms. Section 5 provides future research recommendations that might bridge the gap in research topics related to AV systems.

2. Sensor Technology and Sensor Fusion Overview

Sensors are generally categorized into two classes based on their operational principle. Proprioceptive sensors are the first category in which the sensor operates by capturing the dynamical

state and the internal measurements of the dynamic system. Global positioning system (GPS), encoders, gyroscopes, gyrometers, and accelerometers are examples of this category. The second class covers exteroceptive sensors, with which the external variables of the surrounding system are sensed. This applies to cameras (Complementary Metal-Oxide-Semiconductor (CMOS), Infrared, fisheye, and cyclops), radio detection and ranging (radar), and light detection and ranging (LiDAR). In addition to this categorization, sensors can be either passive or active. Passive sensors produce outputs by sensing the surrounding energy using cameras and GPS, while active sensors transmit energy and measure the reflected energy to produce outputs (LiDAR and Radar) [12].

In autonomous vehicles, different combinations of active and passive sensors are employed to perform the two main tasks of perception. (1) Environmental Perception: RGB cameras, thermal cameras, LiDAR, and radar are used for on-road vehicle detection and tracking, pedestrian detection, tracking, road surface detection, road lane detection, and road sign detection. (2) Localization: global navigation satellite systems (GNSS), inertial measurement units (IMU), inertial navigation systems (INS), odometers, cameras, and LiDAR are used to obtain the relative and absolute positions of the vehicle.

In general, it is difficult to generate data from a single independent source and use it in complex applications, such as AVs. The reasons are either due to sensor shortages, the nature of the sensed environment, or both. Sensors suffer from several inadequacies and limitations, which can degrade their performance. Some sources of performance degradation are due to drifting errors, where a small offset can lead to a huge error when readings are accumulated over time, as in IMU [13,14]. Additionally, errors can be due to low sensor resolution, surface irregularities, or wheel slipping, as in-wheel odometers. Finally, they can be due to uncertainty in readings. Some high-accuracy sensors exist that could overcome some of these limitations, such as differential global positioning systems (DGPS), real-time kinematic (RTK) positioning systems, and fiber optics IMU; however, they are often unavailable or impractical for use in AV systems due to their operating limits (occultation and multireflection effect) and their high cost.

Besides the sensors' own imperfections, the sensed environment conditions have an enormous effect on the sensors' outputs. Sensor noise, for example, disturbs camera images through sunlight intensity and illumination. Similarly, low light at nighttime degrades the outputs of color cameras. Moreover, GPS sensors are affected by outages in certain areas, such as tunnels and forests.

AV researchers use different combinations of sensors and fuse their readings at different levels in order to compensate for the limitations of the individual sensors. Vision cameras are essential sensors that generate a detailed environmental view of the AV surroundings. They are inexpensive sensors for a given level of performance (e.g., resolution, accuracy) compared to active ranging sensors, and can provide dense pixel information for the surrounding scene at relatively low cost. However, normal vision-based systems fail to provide the depth information needed to model the 3D environment. One alternative is to use a stereovision system that consists of multiple cameras with different locations. Nevertheless, these systems are also extremely sensitive to external environmental conditions, such as light intensity (low light and direct sunlight) [15]; and severe weather situations such as fog [16,17], snow, and rain. Fusing a vision-based system with LiDAR, for instance, creates a complementary output that provides the depth information while being robust to external weather conditions [18–20].

The use of infrared and thermal imaging is another active field that researchers often investigate for environment perception applications, especially in unfavorable light conditions and night vision. These systems are often used for applications such as pedestrian detection and tracking [21–23] due to their ability to detect humans regardless of the light intensity. In the literature, thermal cameras have been fused with either RGB-D [24] or LiDAR sensors [25,26] to add depth, and hence improve the system performance; however, this advantage can be dramatically compromised in extreme weather conditions, such as high temperatures.

Localization and mapping normally use a combination of different sensors, such as GPS, IMU, LiDAR, and cameras, to obtain accurate and reliable results. Despite the availability of highly accurate and reliable GPS sensors, it is common that GPS signals typically face blockages or outages in certain environmental conditions. Hence, to compensate for losses of GPS signal, the localization system is likely to be coupled with other sensors, such as IMUs [27].

Additionally, high-accuracy sensing devices are usually very expensive, making them unsuitable for use in applications other than accurate ground truth readings for evaluation and validation of the quality and the performance of an algorithm. Reducing the cost of sensing technologies while maintaining an efficient output is one of the priorities in AV systems, hence combining low-cost IMU data and GPS signals can yield continuous and accurate state estimations of vehicles [28]. Moreover, cameras and LiDAR [29,30] are used in a configuration that will allow extraction of specific environment primitives (road markings and static interest points) for use in either map building through simultaneous localization and mapping algorithms (SLAM) [31] or by matching them with a pre-existing high-definition (HD) map [32] and then obtaining accurate positions for both the ego vehicle and surrounding objects.

Table 1 provides a comprehensive list of various combinations, fusions, and association methods of the most common sensors used in self-driving vehicles. The table also describes the limitations of the sensors if they are to be used individually. Additionally, it lists the advantages of fusing a suitable set of sensors to achieve the desired output.

Table 1. Summary of AV applications, limitations of sensors, and advantages of sensor fusion.

Study	AV Application	Fused Sensors	Limitations without Fusion	Fusion Advantages
[34–36]	Pedestrian Detection	Vision and LiDAR	Sensitive to illumination quality; Night vision difficulties by vision camera only Low resolution of LiDAR 3D scene reconstruction when used alone.	Ability to measure depth and range, with less computational power; Improvements in extreme weather conditions (fog and rain)
[37–42]	Pedestrian Detection	Vision and Infrared	Night vision difficulties with vision camera only; Thermal cameras lose fine details of objects due to their limited resolution.	Robustness to lighting effects and nighttime detection; Infrared camera provides distinct silhouettes of objects; Ability to operate in bad weather conditions.
[43–46]	Road Detection	Vision and LiDAR	Illumination and lighting conditions; High computational cost for vision depth measurements; Limited resolution and range measurements by LiDAR; Sparse and unorganized point cloud LiDAR data	Road scene geometry measurements (depth) while maintaining rich color information; Calibration of scattered LiDAR point cloud with the image
[47]	Road Detection	Vision and Polarization camera	Sensitive to lighting conditions; Lack of color information	Polarized images enhance scene understanding, especially with reflective surfaces.
[48–50]	Vehicle Detection Lane Detection	Vision and Radar	Low resolution of radar. Camera needs special lenses, arrangements, and heavy computation to measure distance.	Measure distance accurately; Performs well in bad weather conditions; Camera is well suited for lane detection applications

Table 1. *Cont.*

Study	AV Application	Fused Sensors	Limitations without Fusion	Fusion Advantages
[51]	Visual Odometry	2D Laser scanner and Vision	2D scanners can miss detection of objects in complex environments; 2D images are insufficient for capturing all the features of the 3D world.	Fusion of vision and 2D scanners can replace the need for 3D LiDAR, and hence reduce price and computation load.
[52,53]	SLAM	Vision and Inertial Measurement Unit	Illumination and lighting conditions by the camera; Camera suffers blur due to fast movement; Drifting error for IMU	Improved accuracy with less computational load; Robustness against vision noise, and corrective for IMU drifts.
[54]	Navigation	GPS and INS	GPS outage in denied and canyon areas; Drift in INS readings	Continuous navigation; Correction in INS readings
[32,55]	Ego Positioning	Map, vision, GPS, INS	GPS outage; INS drifts; HD map accuracy; Visibility of road markings	Accurate lateral positioning through road marking detection and HD map matching.

Both sensor fusion and information fusion can be defined as the process of managing and handling data and information coming from several types of sources in order to improve some specific criteria and data aspects for decision tasks. In our case, the process of fusion consists of combining the outputs of individual sensors or the outputs of specific algorithms (state vectors and uncertainty matrices) to produce a new combined outcome that is enhanced, extended, enriched, more reliable, more confident, and more robust than those generated by each of the individual sensors separately. The final goal is to use the redundancies, complementarities, and advantages of a set of data in order to obtain good enough perception data to make the best decision.

Figure 3 illustrates the five different levels of data processing for perception and decision applications. The first level represents the raw input data collected from various combinations of sensors. The second level portrays the process of filtering, spatial and temporal alignments, and uncertainty modeling. The outputs of the latter are observations, which will in turn pass to the third level, where feature extraction, object detection, clustering, data processing occur to generate representations of objects (e.g., sizes, shapes, colors). Layer four concludes the perception layer, where different objects can be identified by their behavior or specific properties and trajectories to build a proper representation of their interactions, which are inputs to higher-level processing, such as decision making at the fifth level. It is worth mentioning that an output is either used for one of the perception levels (tracking stage, information looping, or strong coupling) or used for the final decision layer [33].

The field of sensor fusion has been applied in multiple applications, ranging from military applications such as automated target recognition [56] to medical applications [57], remote sensing [58], and self-driving vehicles [59]. In autonomous vehicles, sensing of the surrounding environment is one of the crucial steps in building a successful and complete system (perception, decision, and action). Vehicles are usually equipped with different types of sensors that collaborate in order to initiate the right decisions. With that said, an enormous amount of research has been conducted over the past decade to adopt and improve sensor fusion methods for autonomous vehicles.

Searching the literature, it has been found that several categorization schemes of sensor fusion methods exist. In this section, the most used classes will be listed. The first category is based on the input type that is used in the fusion network, which includes data fusion (early fusion), where the fusion takes place at the raw data level. The second category is feature fusion (halfway fusion), where features are first extracted from sensor data and then these features are fused halfway through the network. The last category is decision fusion (late fusion), in which multiple classifiers are used to

generate decisions that are then combined to form a final decision. The architecture of the different levels is illustrated in Figure 4.

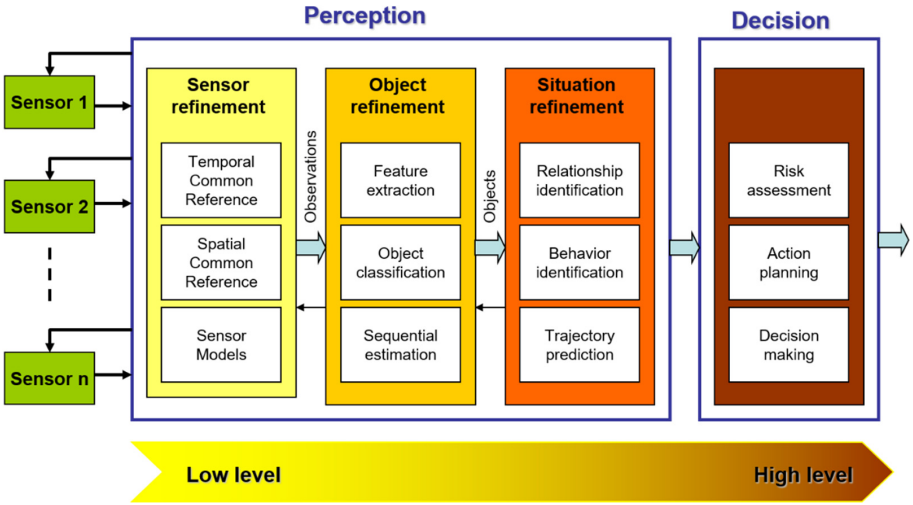

Figure 3. The different stages in the perception and the decision process.

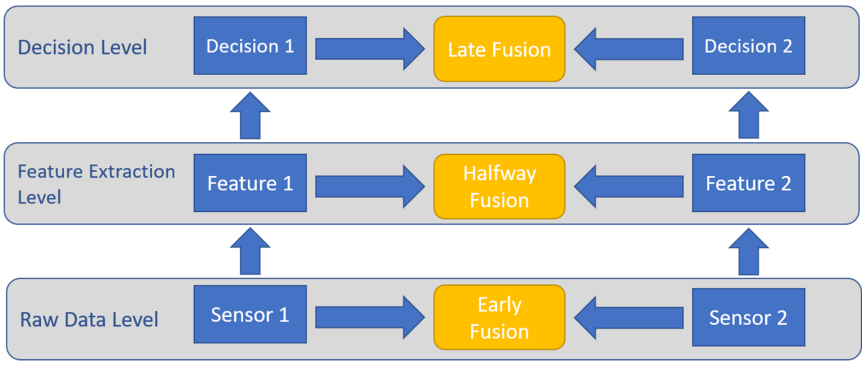

Figure 4. Sensor fusion architecture described in terms of the three different levels. Level one represents early fusion, level two represents halfway fusion, and level three represents late fusion.

A different categorization was explained by Dasarathy in [60], where he listed five detailed classification classes, including:

1. **Data in, data out**: The input to the fusion network is raw sensor data, while the output is processed (typically enhanced) raw data.
2. **Data in, feature out**: Raw data is integrated to produce a set of output features.
3. **Feature in, feature out**: Where the input and output of the fusion network are feature vectors. This class is commonly referred to as either feature fusion, symbolic fusion, or information fusion.
4. **Feature in, decision out**: As the name suggests, the input is a feature vector and the output is a decision. This class is more common in pattern recognition activities, where feature vectors are processed to be labeled.
5. **Decision in, decision out**: Where both inputs and outputs are decisions, usually referred to as decision fusion networks.

Based on the source of the fused data, sensor fusion can also be categorized as a multimodal fusion method [61], where the fused data are obtained from two or more different types of sensors. Fusing LiDAR point cloud and camera images is a good example of this type of fusion, where the two modalities complement the functionality of each other and provide an improved outcome. Another class

of this category is multitemporal fusion, where data are obtained from the same sensor but at different acquisition times. This type of fusion is common in satellite images used for monitoring changes on Earth. The third type of fusion is multifocus fusion [62], where images are obtained from different focal lengths. The last class is multispectral fusion [63], in which images are captured from different wavelength sensors, such as an RGB camera and a thermal camera. This type of fusion is found in applications similar to pedestrian detection and object recognition.

Additionally, based on sensor configuration, sensor fusion can be categorized into a complementary configuration, in which two independent sensors are used and their outputs are combined to complement each other. A perfect example of this type is the fusion of multiple ultrasonic sensors fixed on a robot bumper to expand the coverage area. Fusion can also be of a competitive configuration (also called redundant configuration), where multiple sensors are used to measure the same property and the outputs are used for correction purposes, as is the case in multiple IMU measurements. The last configuration is the cooperative configuration, where two or more sensors are used to provide an output that cannot be achieved by the individual sensors, such as integrating the outputs of two stereovision cameras in order to get a three-dimensional depth image.

Lastly, based on fusion architecture, sensor fusion can be categorized into centralized, decentralized, and hybrid fusion architectures [33]. In centralized fusion, all data from the different sensors are connected to a central processing unit. After all data are aligned to a common reference frame, the central unit receives the output as one source of information in order to fuse it. In decentralized fusion, data obtained from sensors are processed locally, then the obtained output is forwarded to a global processing unit for fusion. A hybrid architecture includes sets of data processed locally and forwarded to the global processor, where the remaining data will be processed and fused.

This review paper divides sensor fusion techniques and algorithms into classical algorithms and deep-learning-based algorithms. However, the scope of this study is to review the implementation of deep learning sensor fusion approaches in AV applications.

2.1. Traditional Sensor Fusion Approaches

There are several classical algorithms that utilize data fusion for the development of applications that require to modeling and propagation of data imperfections (inaccuracy, uncertainty). These algorithms apply methods and approaches that are based on theories of uncertainty, as illustrated in Figure 5 These methods include probabilistic methods, statistical methods, knowledge-based methods (fuzzy logic and possibility), interval analysis methods, and evidential reasoning methods. The variations of each category are listed in Figure 6 In this section, Table 2 briefly summarizes the common classical algorithms, along with their advantages and disadvantages. The readers interested in detailed discussions about traditional fusion algorithms are recommended to refer to [33,64–66].

Table 2. A comparison between traditional sensor fusion algorithms, their advantages, disadvantages, applications, and fusion level.

Algorithm	Characteristics	Advantages	Disadvantages	Applications Areas	Level of Fusion
Statistical Methods	Utilized to enhance data imputation using a statistical model to model the sensory information [64,67]	Can handle unknown correlations; Tolerant [68,69]	Limited to linear estimators; Computation complexity is high [65]	Estimation	Low [70]
Probabilistic Methods	Based on probability representation for the sensory information [64]	Uncertainty in the provided information is handled. handles nonlinear systems (particle filter, UKF, ...)	Requires prior knowledge of systems model and data	Estimation/ Classification	Low→Medium [70]

Table 2. *Cont.*

Algorithm	Characteristics	Advantages	Disadvantages	Applications Areas	Level of Fusion
Knowledge-based Theory Methods	Utilizes computational intelligence approaches inspired by human intelligence mechanisms. [71]	Handles Uncertainty and imprecision; Ability to handle complex nonlinear systems [72]	Depends on the expertise knowledge and extraction of knowledge	Classification/ Decision	Medium→High [70]
Evidence Reasoning Methods	Depends on the Dempster combination mechanism to implement the model [71]	Uncertainty degree is assigned to the provided information. Identification of conflicting situation. Modeling of complex assumption	High computation complexity. Require assumption of evidence.	Decision	High [70]
Interval Analysis theory	Shares the operating space in intervals [73]. Constraint satisfaction problem [74,75]	Guaranty integrity. Ability to handle complex nonlinear systems	Discretization of the operating space. High computation complexity.	Estimation	Low

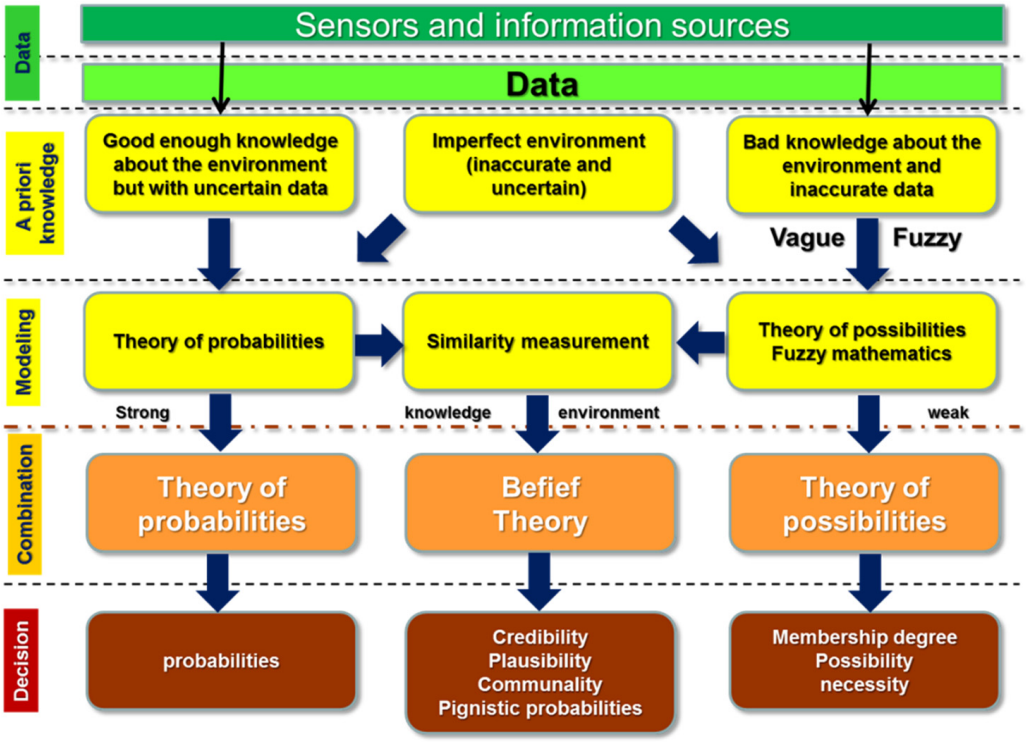

Figure 5. Theories of uncertainty for modeling and processing of "imperfect" data.

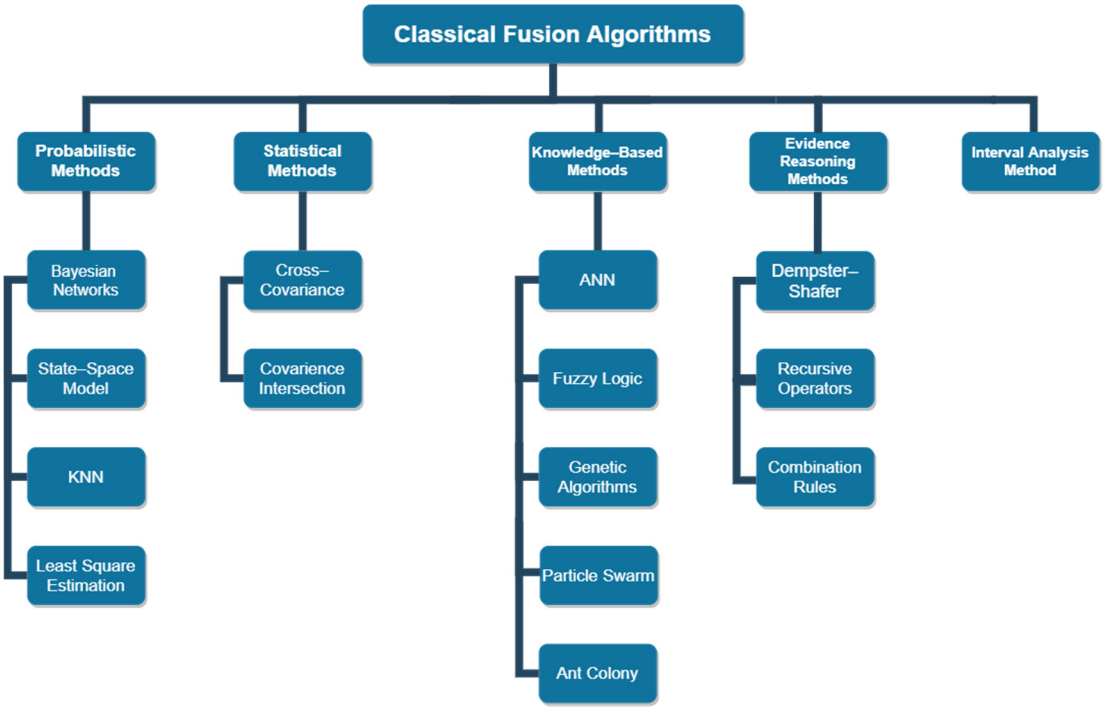

Figure 6. Classical approaches for sensor fusion algorithms.

2.2. Deep Learning Sensor Fusion Approach

Deep learning could be seen as an improvement of neural networks and is a subdivision of artificial intelligence and machine learning that aims to imitates the functionality of the human brain. The algorithms involve creating manifold networks that have multiple layers, allowing them to process raw data and extract certain patterns to perform complex and intelligent tasks. The core concept of deep learning is based on artificial neural networks (ANN), which can be traced back to 1943, when Walter Pitts and Warren McCulloch [76] took the first steps towards building a model that based on the working principle of a human's brain neural networks. While the basics of deep learning were founded long ago, its recent vast emergence is due to the development of powerful computing machines and the availability of the "big data" needed to train the models. Recently, deep learning is being extensively used in many different applications, such as in object detection [77], environment segmentation, semantic object identification, healthcare [78], self-driving vehicles [79,80], and finance [81], to name a few.

Several different algorithms exist that are listed under the category of deep learning. Each technique has its own unique properties, and hence is used for a certain application, where the goal is to achieve optimal performance. The frequently used deep learning methods can be listed as (1) convolutional neural networks (CNN), (2) recurrent neural networks (RNN), (3) deep belief networks (DBN), and (4) autoencoders (AE). Table 3 outlines an overview of these algorithms, along with their applications.

There is a noticeable increase in the amount of research associated with deep learning sensor fusion algorithms in autonomous driving. CNN and RNN are among the most commonly used algorithms in AVs. This paper, hence, aims to provide a detailed overview of the recent advancements in sensor fusion using deep learning approaches, while focusing on these two algorithms and their variations. Figure 7 depicts different variations of CNN and RNN that have been utilized in AV applications.

Table 3. A summary of different deep learning algorithms, their main properties, and applications.

DL Algorithm	Description	Applications
Convolutional Neural Network (CNN)	A feedforward network with convolution layers and pooling layers. CNN is very powerful in finding the relationship among image pixels.	Computer Vision [82–84]; Speech Recognition [85]
Recurrent Neural Network (RNN)	A class of feedback networks that uses previous output samples to predict new data sample. RNN deals with sequential data; both the input and output can be a sequence of data.	Image Caption [86]; Data Forecasting [87]; Natural Language Processing [88]
Deep Belief Net (DBN)	Multilayer generative energy-based model with a visible input layer and multiple hidden layers. DBN assigns probabilistic values to its model parameters.	Collaborative Filtering [89]; Handwritten Character Recognition [90]; acoustic modeling [91]
Autoencoders (AE)	A class of neural network that tends to learn the representation of data in an unsupervised manner. AE consists of an encoder and decoder, and it can be trained through minimizing the differences between the input and output.	Dimensionality Reduction [92]; Image Retrieval [93]; Data Denoising [94]

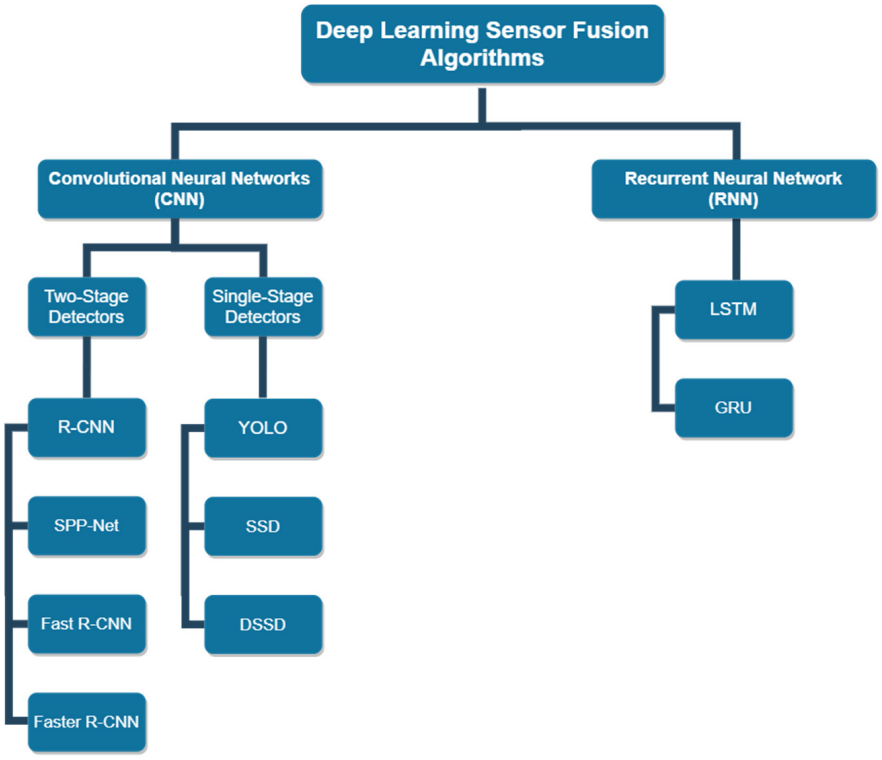

Figure 7. Common deep learning sensor fusion algorithms used in autonomous vehicle applications. R-CNN: Region-Based CNN; SPP-Net: Spatial Pyramid Pooling network; YOLO: You only look once; SSD: Single-Shot Multibox Detector; DSSD: Deconvolutional Single-Shot Multibox Detector; LSTM: Long-Short Term Memory; GRU: Gated Recurrent Unit.

In this review, the discussions around the power of deep learning methods are concentrated on:

1. Environmental perception, including vehicle detection, pedestrian detection, road surface detection, lane tracking, and road sign detection.
2. Localization and mapping.

3. Environmental Perception: Local Dynamic Perception Map

Environmental perception is the process by which AVs tend to sense, understand, and build a full awareness of the environment and the objects that surround it. This perception map is built from the information coming from the five main key components of the environment (obstacle, road, ego vehicle, environment, and driver). Vehicles are usually equipped with multiple sensors to assess the first two key components and detect a variety of objects in the environment, such as vehicles, pedestrians, signs, roads, and lanes. Generally speaking, RADAR, LiDAR, and vision-based systems are the most common sensors used in environmental perception. Therefore, it is very common to find literature discussing detection algorithms using convolutional neural networks (CNN), as they are extremely powerful with visual data.

Before CNN was introduced in 2012, multilayer perceptron (MLP) was heavily used in image recognition and classification. MLP is a feed-forward, fully connected neural network, which consists of an input layer, a hidden layer, and an output layer. With current advancements, it has been concluded that MLP has many limitations and is not a sufficient tool due to the following disadvantages. First, it has a growing number of parameters that need to be trained. Second, it loses spatial information and pixel arrangement of an image. Third, it is not translation-invariant. On the other hand, CNN is a subset of deep learning algorithms that uses convolution operation to process pixels in images. It has a different architecture compared to that of a MLP; the layers are organized into three dimensions of width, height, and depth. Additionally, the neurons of CNN are not fully connected to the layers.

A general CNN usually consists of the following layers, as shown in Figure 8:

- Input layer: This contains the data of the input image.
- Convolution layers: Convolution operation is performed in this layer to extract important features from the image.
- Pooling layers: Located between two convolution layers, which help in minimizing the computational cost by reducing some—but maintaining the most dominant—spatial information of the convoluted image.
- Fully connected layer: This serves as a classifier connecting all weights and neurons.
- Output layer: This stores the final output, which is the classification probability.

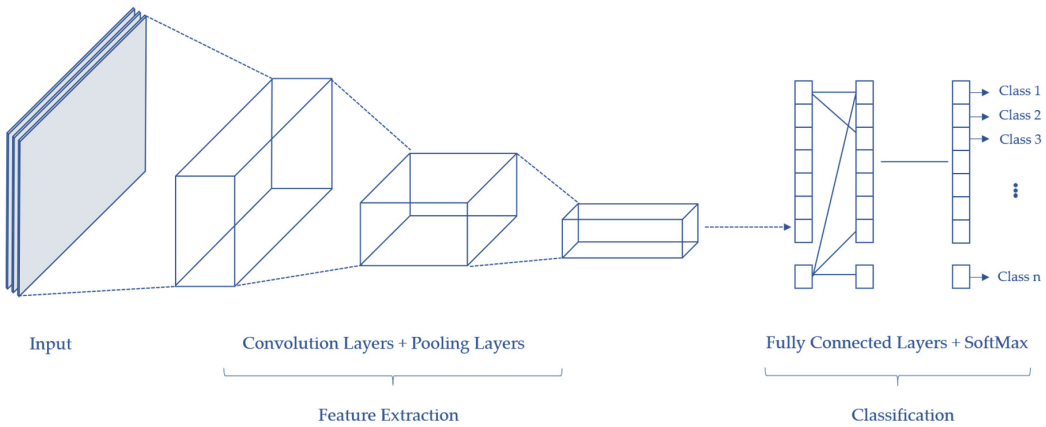

Figure 8. The different processing layers in a convolutional neural network (CNN) for object detection and identification.

CNN achieved its current popularity in 2012 [95] when Krizhevsky et al. proposed AlexNet and won the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC). Since that breakthrough, there has been increased research interest in CNN. Going down the timeline, advancements in CNN image detectors have gone through two parallel paths: (i) the two-stage detectors, which consist of region proposals first, then prediction; and (ii) the single-stage detectors, in which prediction is carried out directly without having an intermediate stage. Figure 9 presents the timeline of the most popular CNN detectors of both types.

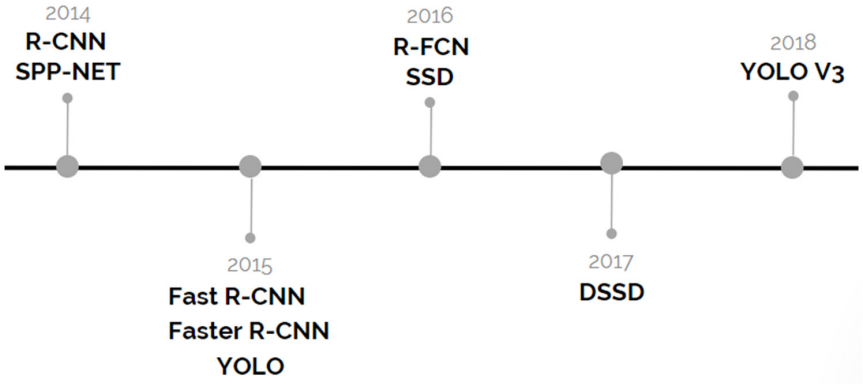

Figure 9. Timeline development of CNN-based detectors. R-FCN: Region-Based Fully Connected Convolution Network.

3.1. R-CNN

Region-based CNN (R-CNN) was the first two-stage detector introduced by Girshick et al. [96]. The purpose of this algorithm is to reduce the computation load and enhance the detection speed. This was achieved by creating 2000 regions in the image through a selective search algorithm instead of covering all regions of the image. Selected regions are processed by a CNN network for feature extraction, and later classified by a Support Vector Machine (SVM) classifier. The architecture of R-CNN is illustrated in Figure 10. Sensor fusion based on R-CNN is applied to different applications of AVs. Wanger et al. [39] studied the impact of fusing thermal images and visible images in pedestrian detection in both daytime and nighttime. In their research, the R-CNN algorithm was used with both early fusion (pixel-level) and late fusion architectures. Their results showed that pretrained late fusion architecture achieved better performance compared to state-of-the-art baseline algorithms. Similarly, in a different study, LiDAR depth features known as horizontal disparity, height above ground, and angle (HHA) features were fused with RGB images to detect pedestrians [34]. A R-CNN network was used and six different fusion architectures were generated based on the network layer at which the fusion takes place. While both studies showed improvements in the mean percentage error, the proposed approach still requires more work to reduce the processing time in order to be fully efficient in embedded real-time AV applications.

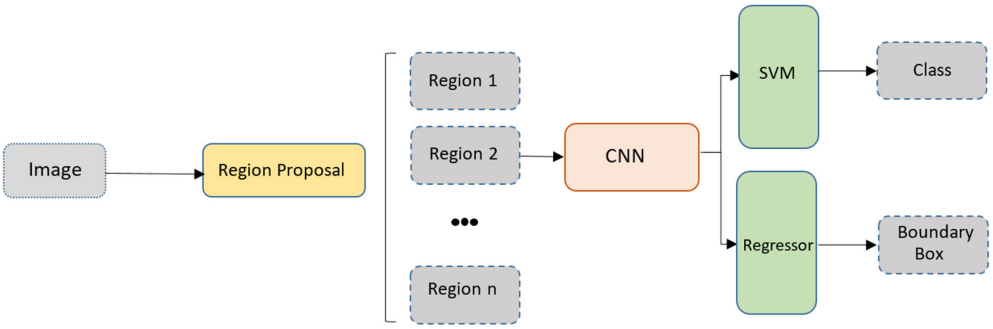

Figure 10. The architecture of the region-based CNN (R-CNN) algorithm.

3.2. SPP-Net

Despite the astonishing accuracy of R-CNN at the time, it was not optimal enough to be used in real-time autonomous driving application. The algorithm requires around 47 s for image detection, which is too long for real-time applications. In addition, R-CNN has to classify around 2000 proposed regions, which leads to massive training time. Multiple attempts have been made to overcome these drawbacks. One of these attempts was the introduction of spatial pyramid pooling (SPP-Net) [97]. This new technique has the advantage of eliminating the need for cropping, resizing, or changing the aspect ratio of the input image to a certain size by introducing multiple pooling layers with different scales. In addition to its ability to generate a fixed-length representation regardless of the input size, SPP-Net processes the full image at once instead of processing all 2000 regions generated by the region proposal, which leads to a noticeable improvement in the processing speed of the algorithm. The simplified architecture of the algorithm is illustrated in Figure 11, where regions and feature maps are passed into multiple pooling layers, before they are concatenated and fed into a Fully Connected (FC) Layer for classification and regression.

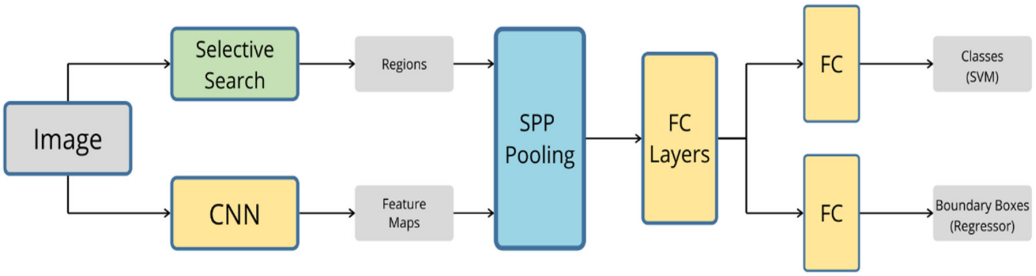

Figure 11. The architecture of spatial pyramid pooling (SPP-Net) algorithm.

3.3. Fast R-CNN

Fast R-CNN was also proposed by Girshick [98] to improve the speed of training and testing and to improve detection accuracy. In fast R-CNN, instead of processing the region proposals by the CNN network, the input image is processed and a convolutional feature map is produced. Regions of interest (ROI) are generated from the feature maps and fed to the fully connected layer, as shown in Figure 12. It is worth mentioning that fast R-CNN is nine times faster in training and 213 times faster in inferencing than R-CNN [98].

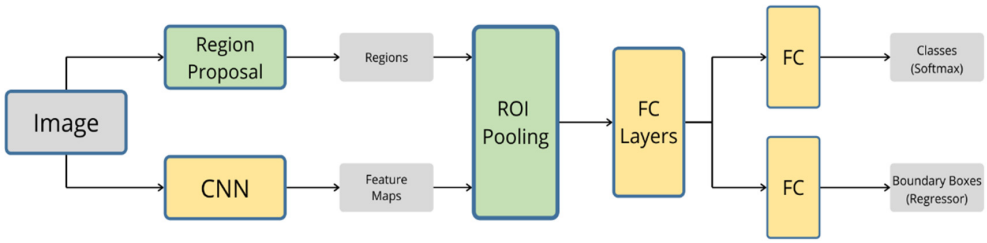

Figure 12. The architecture of fast R-CNN.

3.4. Faster R-CNN

Even though inferencing time was decreased from 47 s in R-CNN to 2.3 s in fast R-CNN, the latter algorithm determines the regions and their bounding boxes using a selective search algorithm, which itself causes a considerable delay in the process. In 2015, Ren et al. proposed the region proposal network (RPN), which is a separate neural network used to predict the bounding boxes. This network is merged with R-CNN, which share the convolution features. The new algorithm, named faster R-CNN [99], has the architecture described in Figure 13. Faster R-CNN was the first place winner of

the ILSVRC competition. Faster R-CNN reports a testing time of 0.2 s, which makes it suitable for real-time applications.

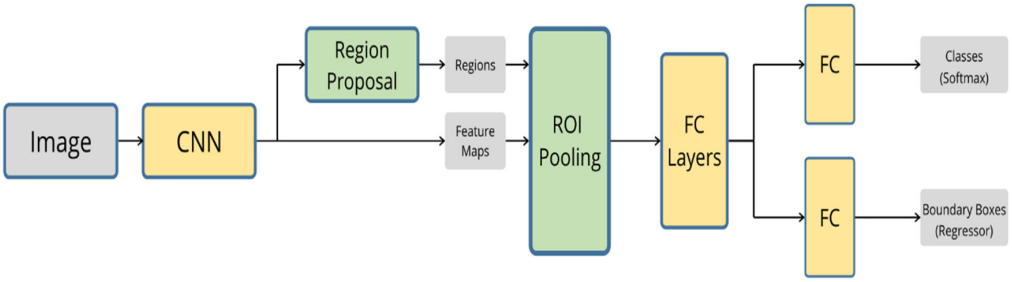

Figure 13. The architecture of faster R-CNN.

In comparison, faster R-CNN is noted as being the most used among other region-based CNN algorithms due to its accuracy and fast processing time. Liu et al. [37] used faster R-CNN in multispectral pedestrian detection, where thermal and color images are fused to provide the complementary information required for daytime and nighttime detection. Having complementary sensor data would undoubtedly enhance the detection results; however, choosing the correct fusion architecture would yield a better detection outcome. In [37], four fusion models named early fusion, halfway fusion, late fusion, and score fusion were designed and tested. It was found that halfway fusion achieved the best detection results compared to those of the baseline faster R-CNN method. By extending the work done in [37], two additional fusion architectures were added in [100], namely the input fusion and score fusion II. Additionally, an illumination-aware gating network that assigns different weights to the modalities based on the illumination condition was added. In a different approach, faster R-CNN was used in [101] to detect pedestrians at nighttime by fusing successive images from a monochrome image. It is claimed that successive frames can improve the detection results by increasing the information contained in the image, especially in dark conditions with low brightness and contrast.

Table 4 provides a quantitative comparison of the two-stage detector algorithms. It compares the time needed to train each of the networks with respect to the baseline algorithm (R-CNN). Additionally, the table lists the rate at which each algorithm needs to perform image recognition. The data shown in the table are from the experimental results reported in each corresponding study [96–99].

Table 4. Comparison of the training time and testing time of different region-based detection algorithms and improvements of each algorithm compared to R-CNN.

	R-CNN	SPP-Net	Fast R-CNN	Faster R-CNN
Training time (In hours)	84	25	9.5	NA
Speedup with respect to R-CNN	1×	3.4×	8.8×	NA
Testing rate (Seconds/Image)	47	2.3	0.3	0.2
Speedup with respect to R-CNN	1×	20×	146×	235×

3.5. YOLO

Single-stage detectors, on the other hand, consist of a one-step regression rather than a multistage classification process. One of the most popular algorithms is the "you only look once" detector (YOLO), founded in 2016 by Redmon et al. [102]. As seen in Figure 14, the input image is divided into a defined number of grids, then a single neural network is applied to predict bounding boxes and produce class probability for the boxes, which is all performed in one stage. Compared to the previous detectors mentioned above, YOLO is considered to have a very fast detection speed of 45 frames per second [102]; however, the use of YOLO is restricted due to its disadvantages of high localization error and low

detection accuracy when dealing with small objects. These limitations were addressed by proposing improved algorithms in YOLOv2, YOLO9000 [103], and YOLOv3 [104].

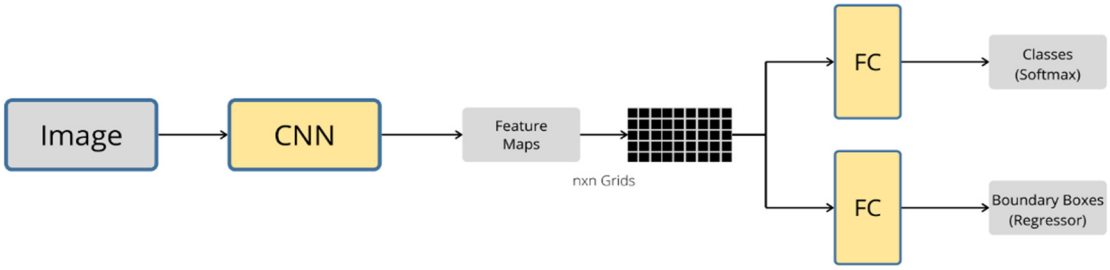

Figure 14. The architecture of "you only look once" (YOLO) algorithm.

In many research attempts, LiDAR and visible cameras were used together to obtain better detection results. In [105], for example, Asvadi et al. used depth maps (DMs), while reflectance maps (RMs) generated by 3D LiDAR were fused with RGB images to detect objects on the road. Three YOLO networks were used to process DM, RM, and RGB images separately and generate bounding boxes from each network. Features are extracted from the three modalities and a decision-level fusion is then applied to achieve vehicle detection. In a different study [106], LiDAR point cloud data were used to construct a map of the vehicle's view; then, these maps were used to generate regions of interest, which were then projected on the camera image. A YOLOv3 network was then utilized to perform real-time vehicle detection. In spite of the decent results that were presented in both papers, small objects such as pedestrians were not considered.

3.6. SSD

Throughout the literature, it has been noticed that the YOLO algorithm is mostly applied on large objects such as vehicles. In fact, according to [107], YOLO's accuracy is degraded when dealing with small and variant-scale objects. Moreover, YOLO applies spatial constraints on the bounding boxes, limiting the classification to a single class of objects [108]. Many noticeable efforts have been put forth to solve such restrictions. One of the starting points is the single-shot multibox detector (SSD) [109], which is the result of a recent study involving significant improvements and attempts to overcome the limitations of the previous state-of-the-art methods. SSD is designed to have bounding boxes with different sizes and aspect ratios. This property enables the algorithm to detect different objects with different sizes in the same image. SSD is reported to be faster and more accurate than YOLO. It matches the accuracy of faster R-CNN, but with a speed of 59 frames per second (more than 2500 times faster).

In [110], Kim et al. used the SSD algorithm for general object detection in the autonomous driving applications. LiDAR 3D point clouds were converted into 2D images, then these images were used along with RGB images as inputs for two separate SSD networks. Finally, gated fusion units (GFU) were used to assign selective weights to fuse both feature maps produced by the two SSD networks through a feature fusion level. The experimental results showed that the proposed GFU–SSD method outperformed the baseline SSD. The authors in [38] attempted to compare different fusion techniques with different CNN architectures while keeping SSD as the baseline detector. The fusion of thermal images and visible images was carried out with early and late fusion by using a SSD network and comparing it with other detectors, such as faster R-CNN and DefineNet. The results showed that the miss rate was reduced with the SSD detectors in both early and late fusion. Figure 15 illustrates the architecture of the SSD.

3.7. DSSD

Due to the fact that small objects yield a limited number of pixels and information, the detection of these objects becomes a burden. In most cases, the improvement of accuracy is traded with the speed of detection [111]. Some variations of the SSD networks have been implemented to improve the

accuracy with small objects while maintaining high detection speed. For example, the deconvolutional single-shot detector (DSSD) [112] uses ResNet101 instead of the original Visual Geometry Group (VGG) classifier and adds more context information into the existing SSD algorithm by augmenting it with deconvolutional layers. This provides feature maps with better resolution, which enhances the detection of small objects. For the pedestrian detection task, colored and thermal images were fused using halfway fusion through the DSSD network [40]. As pedestrians are small objects, the new algorithm was compared to previous studies that use different detectors. It has been shown that the overall accuracy is improved.

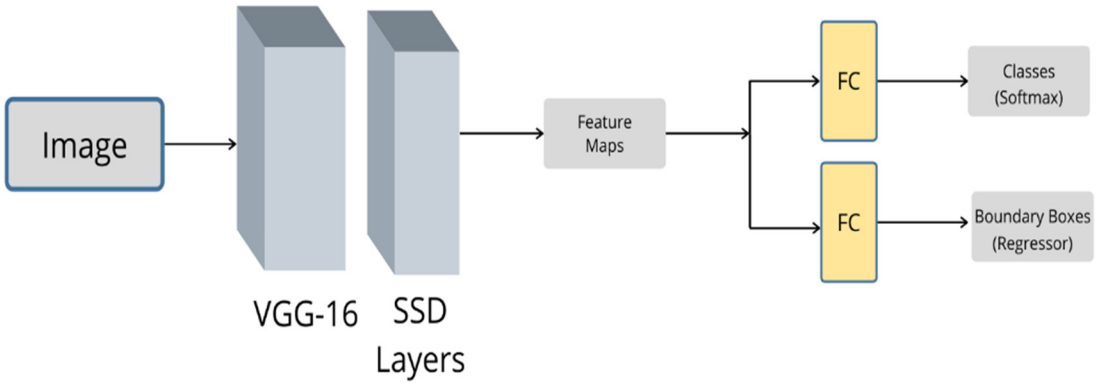

Figure 15. The architecture of the SSD algorithm. The CNN Network is VGG-16.

4. Ego-Localization and Mapping

An effective autonomous driving system requires the vehicle to determine its position and orientation accurately. Vehicles need to have accurate, reliable, and robust localization algorithms to assist in their maneuvering tasks, avoid surrounding obstacles, and perform the right driving actions. Moreover, the localization system needs to be robust to handle variant complex environments and severe weather conditions. Generally, localization is commonly performed using a variety of sensors, such as GNSS; and dead reckoning devices, such as IMU, vision sensors, and LiDAR (for visual odometry, primitive detection, and mapping, for example for SLAM algorithms). The fusion of two or more of these sensors is also a common practice to enhance the overall localization performance.

It is worth mentioning that some emerging studies [113–115] propose different driving algorithms that avoid the need for localization and mapping stages, and instead sense the environment and directly produce end-to-end driving decisions. This is known as the behavior reflex approach [113]. In contrast to the classical method above, known as the "mediated perception approach", this approach aims to reduce the computational load by eliminating unessential information, hence improving the speed of the process [113].

This section aims to analyze the localization techniques as part of the mediated perception approach, while focusing on the fusion of different sensors through deep learning algorithms. Table 5 provides a summary of the common ego-localization and mapping techniques.

Table 5. Comparison of localization and mapping techniques in terms of the accuracy, cost, computational load, source of external effects, and the storage size of data.

Method	Accuracy	Cost	Computational Load	External Effect	Data Size
GPS/IMU	Low	Medium	Low	Signal outage	Low
GPS/INS/LiDAR/Camera	High	Medium	Medium	Map accuracy	High
SLAM	High	Low	High	Illumination	High
Visual Odometry	Medium	Low	High	Illumination	High
Map-Based Matching	Very High	Medium	Very high	Map change	Very High

4.1. GNSS/IMU-Based Localization

GNSS is one category of the most commonly used sensors for localization in autonomous vehicles. They have the advantages of low production cost and ease of integration within the vehicle system. GNSS technology, however, has two main deficiencies that prevent them from being a reliable standalone source of information. The first disadvantage is their insufficient accuracy (in the range of 10 m). This range is unsatisfactory for AV applications, where accuracy in the centimeter range is required. The second disadvantage is signal blockage and multipaths, where GNSS signals can sometimes be interrupted in real driving environments. An adequate amount of research exists that focuses on solving these two shortages; some of it has already provided acceptable results, which are discussed below.

DGPS and RTK-GPS are used to enhance the accuracy of GNSS. Both DGPS and RTK rely on having a base GPS station that has a known position of high accuracy. In the case of DGPS, the base station uses its position for comparison with that calculated by GPS and sends the difference to the receivers in order to use it for corrections. RTK, on the other hand, uses the carrier wave to determine the number of cycles between the satellite and the receiver and performs corrections. Without the use of DGPS, it is possible to build a dynamic DGPS with a distributed GPS configuration and with the method proposed in [55]. Although results with decent accuracy were achieved with DGPS and RTK, discontinuity of GPS signals in urban environments and tunnels remains the main issue with these sensors.

GNSS needs to be integrated with other sensors that can compensate for the signal during any possible outage. IMU is a type of sensors that exploit built-in accelerometers and gyroscopes to measure both acceleration and velocity [116]. It then processes this information to estimate the state of the vehicle at a given time with respect to its original position. It is worth mentioning that IMUs suffer from drift error, which results from the accumulation of positioning error during the travel of the vehicle. Hence, an IMU need continuous correction to its estimated position. Despite this, the well-fused output of both GNSS and IMU achieves a state estimate of the vehicle and ensures a continuous localization process.

Various examples of GNSS/IMU sensor fusions exist in the literature. In [117–119], the Kalman filter was developed to integrate the outputs of both GPS and IMU. A Kalman filter (KF) consists of two main equations, named the prediction equation, which is based on the system knowledge (evolution matrix and command matrix) obtained from past measurements, and the update equation, which works on updating the knowledge from the current measurements, i.e., an update of the predicted estimation with the Kalman gain and the error between the predicted state vector and the new observation (GPS data). Generally, enormous improvements happen when fusing both sensors using the Kalman filter approach. Nonetheless, it is important to emphasize that the success of using Kalman-filter-based fusion relies on the perfect match between the state–space system model and the measurement model. Additionally, multiple assumptions should be taken into consideration, such as the linearity of the system and the presence of Gaussian distributed data.

Since system dynamics are not always represented with linear equations, an extension named the extended Kalman filter (EKF) was found to handle nonlinear systems. EFK works by linearizing the system's equations at each time step through the Taylor series and Jacobian matrix, then passing them through the ordinary KF. The main disadvantage of this approach lies in the process of approximation (linearization stage), as it introduces errors that will not be taken into consideration. The unscented Kalman filter (UKF) was next introduced to improve the performance of EKF. Instead of taking one point and approximating it to its linear state, UKF considers a group of weighted points, named sigma points, and uses them for approximation. UKF has achieved better performance in terms of accuracy compared to EKF [120]. Nevertheless, EKF and UKF are mono-model approaches. In order to improve these two well-known approaches, multiple models and multiple hypothesis methods have been developed. Included among these complex approaches are the particle filter approach, the interacting

multiple model (IMM) approach [121], and the optimized Kalman particle swarm (OKPS) approach, which is a merge of the particle filter and swarm method [122].

One of the key challenges in estimation methods is the need to have an accurate model of the system. In some cases, it is difficult to provide an accurate model, especially for complex systems that are highly dynamic. Additionally, most sensors are subject to inherent uncertainties, which usually cannot be incorporated in the system model, yielding an inaccurate model [123]. In this context, deep learning is extremely useful, as it allows for end-to-end learning, which eliminates the need for mathematical modeling of each sensor individually.

Based on the current literature, there are few studies on deep learning sensor fusion in localization. An early attempt to deploy artificial intelligence to fuse GNSS and INS is presented in [123], where an input-delayed neural network is utilized to model the errors of the INS based on current and previous data samples. The test results are compared to the conventional Kalman filter approach and they show several improvements in position estimation during GNSS signal outage.

RNN is a powerful tool that can be used with time-series data. It has the ability to save previous data samples through its memory feature. In [124], RNN was used to fuse both GNSS and INS sensors and produce continuous and reliable navigation. Through the recursive network and memory function, RNN uses past position and velocity data of INS to predict the errors in the current measurements. The proposed method showed a 60% improvement when compared to the conventional EKF method and 30% improvement compared to other neural network methods. Similarly, Kim et al. [125] integrated both GNSS and IMU data using long short-term memory (LSTM), a variance of the RNN algorithm. The purpose is to generate a model of the vehicle position estimation without the need to model each sensor analytically. The LSTM network was trained with GNSS absolute position and IMU data, and the predicted position was compared with the reference position obtained from a high-accuracy RTK GPS. While the results of the study aim to validate the use of LSTM as a fusion technique, the study needs to be further enhanced by testing it in real-life complex driving situations.

In a different context, Jiang et al. proposed the use of deep learning algorithms in [126] to model the INS signal noise in order to eliminate it, which improved the navigation system outcomes. In general, different statistical methods or artificial intelligent methods have been used to model the error signal, but all techniques have their own limitations [126]. To overcome those limitations, the RNN algorithm, along with a combination of LSTM and gated recurrent units (GRUs), was used for noise modeling. Due to the training accuracy of LSTM and the convergence efficiency of GRU, significant improvements were reported by the proposed hybrid algorithm.

4.2. Visual-Based Localization

Vision sensors are important elements in localization and mapping. Compared to other existing usable sensors, such as LiDAR and radar imaging, cameras are often chosen due to their low cost, availability, and ability to capture useful information (static and persistent primitives in the environment). Visual localization has been an active research area for autonomous vehicles. Visual-based localization includes (1) SLAM, (2) visual odometry (VO), and (3) map-matching-based localization. This section aims to review the contribution of deep learning algorithms in advancing each of the previous methods.

4.2.1. Simultaneous Localization and Mapping (SLAM)

SLAM is an algorithm that combines a set of sensors to build a map of the AV and its surroundings, while simultaneously keeping track of the vehicle's current position in reference to the built map. Although SLAM algorithms were initially applied in the field of mobile robots, researchers have put a noticeable effort into adjusting the algorithms to suit autonomous vehicle applications. This was done by taking into consideration different key challenges, such as the need for faster processing, the outdoor lighting conditions, and the dynamic road obstacles. It is important to point out that while SLAM mainly relies on vision-based sensors, other sensors such as GPS, LiDAR, and sonar have

also been used to implement SLAM algorithms. Surveys on recent SLAM methods have been done by [53,127]. Additionally, some new methods with performance evaluations are available on the KITTI website [128]. At this moment, the latest and best methods reported in [128], which do not use deep learning approaches, are presented in Table 6**.** Different algorithms based on different perception types are compared in Table 6 in terms of their accuracy (translation and rotation error) and the time required to run the algorithm (running time). The performance data shows that "traditional" algorithms generally do well in real-time SLAM implementations. Nonetheless, the continuous progress of SLAM algorithms is still an interesting and active research topic in the computer science and robotics community. It seems that in order to improve "traditional methods", it is relevant to share static and dynamic spaces. Such an approach has been proposed by [129]. From LiDAR data, this approach shares the dynamic space (detection and tracking of dynamic obstacles) and static space using a belief plot map concept. The interesting aspect of this method is its ability to model and account for large size objects with nonlinear and complex shapes.

Deep learning approaches have shown great improvements in image classification and detection; hence, there is good potential in applying these algorithms to enhance traditional SLAM algorithms. Although the deep learning applications in this field are still not mature, some studies propose replacing parts of classical SLAM blocks with deep learning modules to attain better accuracy, efficiency, reliability, and robustness. These studies include attempts to improve aspects of pose and depth estimations, loop closure detection, and feature descriptors of classical SLAM algorithms.

A very crucial aspect of a reliable SLAM system is its ability to perform well in dynamic environments. Most conventional SLAM algorithms were designed to operate in a static environment; hence, their overall performance, and in particular their accuracy, is dramatically compromised in real-world driving scenarios, where objects are often dynamic and the driving conditions are sometimes unpredictable. Traditionally, the behavior of dynamic objects in SLAM was estimated through filtering or tracking. However, most of these approaches require immense computational power and are impractical in real-time applications. For this purpose, Kaneko et al. [130] utilized deep-learning-based semantic segmentation to exclude feature points that exist in the sky and moving cars. These two categories are segmented and masked, and hence all feature points in the masked area are excluded. Similarly, Xiao et al. [131] used an SSD network as an object detection framework, whereby the output of the network is segmented into static objects and dynamic objects. The latter are considered as outliers and discarded. The proposed method reported higher accuracy compared to the baseline SLAM algorithm. A more generic solution was proposed in [132], where pixel-wise deep semantic segmentation was used to produce semantic labels. A tracking thread will generate feature points, out of which those belonging to moving objects will again be considered as outliers and excluded by an outlier rejection network.

One of the ongoing challenges in SLAM systems is the ability of the sensors to accurately measure the depth of the scene as captured by its vision sensors (stereo vision or optical flow). Although in some cases depth sensors and RGB-D are used, these sensors have shortcomings, such as their inadequate working range and their poor performance under direct sunlight. As an attempt to improve depth estimation, researchers in [82,83] trained a deep learning CNN network to estimate depth using a single monocular camera. Compared to classical monocular SLAM [133], CNN-based SLAM employs learning abilities to learn the absolute scale and eliminate the need for geometric assumptions to correct the scales of detected objects [134]. Another example of CNN-based depth estimation is presented in [135], where a real-time algorithm named DeepFusion is used to reconstruct dense maps by fusing the depth prediction of a stereo camera with the depth and gradient predictions of a CNN network in a probabilistic manner. For further improvement, Lee et al. [136] proposed the addition of a recurrent network to the existing CNN network to account for the spatiotemporal information in the image or video sequence for better depth estimation. In [137], Kuznietsov et al. proposed using a semisupervised deep learning method that can take the advantages of both supervised and

unsupervised methods. The proposed technique uses the sparse ground truth data for learning and utilizes CNN for depth prediction.

Table 6. SLAM algorithms based on non-deep-learning approaches, as reported on the KITTI website.

Date	Reference	Method	Translation	Rotation	Runtime	Sensor
2017	[138]	SOFT-SLAM 2	**0.65%**	**0.0014**	0.1 s	Stereo
2018	[139]	LG-SLAM	0.82%	0.0020	0.2 s	Stereo
2017	[140]	ORB-SLAM2	1.15%	0.0027	**0.06 s**	Stereo
2015	[141]	S-LSD-SLAM	1.20%	0.0033	0.07 s	Stereo
2018	[142]	IMLS-SLAM	0.69%	0.0018	1.25 s	LIDAR
2018	[143]	MC2SLAM	**0.69%**	**0.0016**	**0.1 s**	LIDAR
2018	[144]	CPFG-slam	0.87%	0.0025	0.03 s	LIDAR
2018	[145]	SuMa	1.39%	0.0034	0.1 s	LIDAR

Translation: relative translation error in percentage; rotation: relative rotation error in degrees per 100 m. Data in bold represents highest performance.

Another significant module that contributes much to the accuracy of the SLAM system is loop closure. This module checks the previously visited and mapped places and uses the results to reduce the error of the built map. Previously, classical approaches were used to perform detection and classification, such as bag-of-words (BoW), scale-invariant feature transform (SIFT), and speeded-up robust features (SURF) approaches. These approaches use appearance-based methods that are created through handcrafted features and have their own limitations. Deep learning can be highly utilized in the loop closure field, as it has already been proven to be very powerful in image recognition applications. Hou et al. [146] used a pretrained CNN-based descriptor to perform visual loop closure. Merrill et al. [147] also proposed an unsupervised deep autoencoder system for loop closure. The performance of the learning-based method was compared with several hand-crafted techniques under various lighting conditions. CNN has achieved an enhanced performance in the case of major light change and faster extraction speed as well.

Table 7 lists some of the recent deep learning SLAM algorithms. It is worth mentioning that in all of the listed studies, deep learning has been used to replace only a specific module, and the proposed algorithms have generally been built upon a traditional SLAM algorithm. Despite this, deep learning has improved the overall accuracy, and in some cases it has solved critical issues, such as operating in highly dynamic environments. With these continuous improvements, it is conceivable that in the near future there will be an end-to-end deep learning SLAM algorithm with superior accuracy and computational efficiency. Another point that can be observed from Table 7 is the diversity of the testing datasets in the previous studies. Unlike the traditional algorithms presented in Table 6, these algorithms are tested on different datasets; hence, a conclusive comparison of their performance based on the published results is not easy.

Table 7. Summary of recent deep-learning-based SLAM algorithms.

Year	Reference	Contribution of Deep Learning	Description	Architecture	Testing Datasets	Runtime
2018	[130]	Semantic Segmentation	Semantic segmentation produces a mask and the feature points on the mask are excluded.	DeepLab V2	CARLA	-
2019	[148]	Feature Descriptors	Replace handcrafted descriptors with learned feature descriptors.	TFeat	EuRoC/TUM	90 ms
2018	[132]	Semantic Segmentation	Semantic segmentation reduces the effect of dynamic objects and is used to build a dense map.	SegNet	TUM/Real Environment	76.5 ms

Table 7. *Cont.*

Year	Reference	Contribution of Deep Learning	Description	Architecture	Testing Datasets	Runtime
2019	[131]	Semantic Segmentation	SSD Network is used to detect dynamic objects. The selection tracking algorithm is used to eliminate dynamic objects and a missed detection compensation algorithm is used for improvements.	SSD	TUM/KITTI	45 ms
2018	[149]	Pose Estimation	End-to-end trained model that consist of a local pose estimation model, pose selection module, and graph optimization process.	FlowNet DTC	Viz-Doom simulated maze	-
2018	[147]	Loop Closure	Compact unsupervised loop closure algorithm that is based on convolutional autoencoders.	Autoencoders	KITTI	-
2019	[135]	Depth Estimation	Real time algorithm that is able to reconstruct dense depth maps from RGB images.	U-Net	ICL-NUIMTUM RGB	94 ms
2020	[136]	Depth Estimation	A recurrent CNN network that is used to process spatial and temporal information for map depth estimation.	Convolutional GRU (U-Net)	KITTI	80 ms

4.2.2. Visual Odometry (VO)

Visual odometry (VO) is defined as the process of obtaining the pose of a vehicle by tracking the change of its position from consecutive images over time. A general VO framework consists of camera calibration, image acquisition, feature detection, feature matching, feature tracking, and pose estimation. Traditionally, VO was performed through two main approaches: feature-based approaches, where features such as lines or corners are detected, and appearance-based approaches, in which pixel intensity values are considered instead [150]. Table 8 summarizes some of the recent traditional VO algorithms extracted from the KITTI website. The V-LOAM algorithm [151] is ranked first because of having the smallest reported translational and rotational errors. Despite the outstanding performance of the conventional VO, deep learning has been extensively studied to replace it, as it works as a generic feature extractor and improves the system by eliminating the need to design hard-coded features. Additionally, fine-tuned feature parameters are not required for deep learning; thus, the robustness and reliability of the systems are enhanced, which are otherwise sensitive to changes in the environment. Moreover, deep learning algorithms tend to learn to recover the absolute scales, and hence no prior information on the motion model or camera parameters are needed.

Many research articles attempt to evaluate the performance of deep learning algorithms in pose estimation by comparing their results with traditional feature-based algorithms, such as SURF and ORB. The results from [152] illustrate that a deep-learning-based algorithm performs better than conventional methods. An early study was conducted by the authors in [153], where two CNN networks were used in a supervised fashion with fully connected last layers, which regress the pose of the camera. Several experiments were reported using a combination of known and unknown testing environments. From the results obtained, it was shown that the network performs better with prior knowledge of the environment over unknown environments. However, the results of both cases tend to accumulate errors over time. As a result, it was recommended to add a recurrent network, which will help to alleviate the drift problem.

To demonstrate the advantage of adopting a recurrent neural network for VO, Wang et al. [154] implemented an end-to-end deep recurrent convolutional network that takes sequential RGB images and detects poses. Same authors then extended their study to take uncertainties into account [155]. A CNN network was utilized to extract important feature representations from the image and an RNN network in the form of stacked LSTM was used to process sequential data and model motion dynamics. The proposed method was tested for outdoor driving and the results were comparable to those produced through classical algorithms. However, the performance of the algorithm degraded when under certain conditions, including fast driving or driving in open areas, with fewer features leading to more outliers. One solution is to increase the size of the training dataset for the network to learn to reject the outliers. This requirement triggers a question—is it always possible to increase our testing dataset? The challenge rests in the process of labeling these data. This leads to exploring the field of self-supervised and unsupervised learning.

Table 8. Summary of recent VO approaches based on non-deep-learning approaches, as reported on the KITTI website.

Date	Reference	Method	Translation	Rotation	Runtime	Sensors
2015	[151]	V-LOAM	**0.54%**	**0.0013**	**0.1 s**	MC + LIDAR
2019	[143]	MC2SLAM	0.69%	0.0016	0.1 s	IMU + LIDAR
2018	[156]	LIMO2_GP	0.84%	0.0022	0.2 s	MC + LIDAR
2017	[157]	GDVO	0.86%	0.0031	0.09 s	SC
2018	[156]	LIMO	0.93%	0.0026	0.2 s	MC + LIDAR
2018	[156]	LiViOdo	1.22%	0.0042	0.5 s	MC + LIDAR
2019	[158]	SALO	1.37%	0.0051	0.6 s	LIDAR
2019	[159]	KLTVO	2.86%	0.0044	0.1 s	SC

Translation: relative translation error in percentage; rotation: relative rotation error in degrees per 100 m. Data in bold represents highest performance.

Unlike supervised learning, an unsupervised learning network does not rely on labeled data or ground truth data for training. Instead, it trains the model by minimizing the photometric error. In [160], Zhou et al. introduced two networks that are jointly trained from unlabeled video frames to predict the depth map and relative camera pose while using view synthesis (i.e., the ability to synthesize a target image by using the depth map and the pose of a nearby image). Li et al. [161] trained the CNN network using stereo images instead of using consecutive monocular images. This approach enabled the network to recover the absolute scale of the scene. In addition to depth and pose estimations, some studies found it essential to incorporate the uncertainty estimation, as the VO problem is considered as a state estimation problem. In [162], the network was trained in an unsupervised manner, but it was further modified to predict the depth and pose, also considering the uncertainty for VO optimization. A summary of the latest deep-learning-based VO studies is listed in Table 9.

Table 9. Recent VO algorithms based on deep learning approaches.

Year	Reference	Description	Architecture	Testing Datasets	Learning Model
2017	[155]	End-to-end algorithm for finding poses directly from RGB images using deep recurrent convolutional neural networks.	CNN-LSTM	KITTI	Supervised
2019	[163]	Encode-regress network that produces 6-Degree of Freedom (DoF) poses without the need of depth maps.	ERNet	KITTI	Semi-Supervised

Table 9. *Cont.*

Year	Reference	Description	Architecture	Testing Datasets	Learning Model
2016	[153]	Two parallel CNN networks are connected at the end by fully connected layers to generate the required pose.	AlexNet	KITTI	Supervised
2017	[160]	An end-to-end algorithm that uses single-view depth and multi-view pose for camera depth and motion estimation.	DispNet	KITTI	Unsupervised
2017	[152]	An approach that generates a 7-dimensional relative camera pose orientation and position vector.	AlexNet with SPP	DTU	Supervised
2018	[161]	Pose and dense depth map estimation with an absolute scale. This generates 6 DoF poses from unlabeled stereo images.	VGG-16 and Encoder-Decoder	KITTI	Unsupervised
2020	[162]	The algorithm uses deep networks for depth, pose, and uncertainty estimation of monocular odometry.	U-Net (DepthNet and PoseNet)	KITTI EuROC MAV	Unsupervised
2018	[164]	A global pose regression and relative pose estimation framework. The network takes two monocular frames and regresses the 6 DoF poses with inter-task correlation.	ResNet-50	Microsoft 7-Scenes Cambridge Landmarks	Supervised

4.3. Map-Matching-Based Localization

One of the well-known methods of AV localization and mapping is the use of prestored, offline maps, known as "a priori maps". In this method, a combination of sensors is used to capture the surrounding environment while predriving the area. The sensor outputs are stored to form a detailed map of the driven roads and areas, and later can be compared to current sensor outputs. This approach can achieve centimeter-level localization, as required for AV navigation applications. One of the key challenges in such methods is the need for frequent map updates to match the constantly changing urban and driving environments.

One of the vast emerging technologies in the field of maps is the building process of HD maps. HD maps provide very accurate lane-by-lane information, enabling vehicles to precisely localize themselves with respect to those maps. Several leading companies such as HERE rely on using the latest LiDAR technology to capture 3D point cloud data of different elements on the road, such as lane markings, road curvatures, road obstacles, and road signs. At the same time, they accommodate real-time changes by updating those maps [165] continuously.

In the last decade, localization based on map matching has received significant consideration. In the literature, several different methods and algorithms exist to achieve accurate vehicle localization. Traditional registration methods such as iterative closest point (ICP), Monte Carlo localization, and normal distribution transform (NDT) have reported satisfactory results. Nevertheless, these algorithms are highly dependent on manual calibration, postprocessing fine-tuning, and handcrafted features for matching.

Employing deep learning algorithms for localization is still an open research topic, even though these algorithms have proven their effectiveness in performing detection, classification, and learning semantics. Some recent studies demonstrate promising results, such as the one presented in [166], which aims to localize vehicle position using LiDAR measurements and prestored point cloud map. The proposed method consists of (i) keypoint extraction and feature descriptors, (ii) a 3D CNN network that takes the cost volume and regularizes it in order to find the matching cost between each captured key point and its equivalent location on the map, and (iii) an RNN module that is used to learn historical relations between sequential frames and perform temporal smoothness for smoother trajectory predictions.

In an attempt to improve the accuracy of localization, in [167] Vaquero et al. suggested improving the quality of the prebuilt map first. They proposed segmentation of the dynamic moving objects in the map, such as other vehicles and pedestrians, in order to obtain a map that is valid for use for a longer period. For this, the LiDAR front view and birds eye viewpoint cloud are processed by dual deep CNN networks to perform segmentation for both views and then filter out all the movable objects.

5. Conclusions and Future Research Recommendations

The field of autonomous vehicles and self-driving cars is vast, as it involves a great variety of subjects ranging from electronics, sensors, and hardware to control and decision-making algorithms, as well as all the social and economic aspects. For this reason, the research opportunities in this field are endless and have growing potential for future expansion. Prospective AV research areas related to technical aspects can cover more advanced sensor technologies, algorithm enhancement, data collection and storage, communication security, and overall performance improvements. In addition, research domains can be extended to cover nontechnical topics, such as the level of societal acceptance of autonomous driving, environmental effects, changes to urban design, and economic benefits.

In this study, we surveyed and critiqued work on perception, localization, and mapping tasks of autonomous vehicles, particularly those empowered by deep learning algorithms that can take advantage of data-driven knowledge discovery rather than physics-based models. As related to the scope of the study, this section aims to summarize the potential research areas that will possibly improve and enrich the field of autonomous vehicles. The recommendations will focus on both environmental perception, localization and mapping, and how to further utilize deep learning algorithms to improve the performance of sensor fusion networks.

5.1. Harsh Weather Conditions

One of the remaining challenges of self-driving cars is their compromised maneuverability and performance in bad weather conditions, such as rain, snow, dust storms, or fog, which can compromise vision and range measurements (degradation of the visibility distance). In such conditions, the performance of most current active and passive sensors is significantly compromised, which in turn leads to erroneous and even misleading outputs. The consequence of a partial or complete sensor failure can be catastrophic for autonomous vehicles and their surroundings. A possible measure to alleviate this problem is to evaluate the risk of failure early in the process based on learned experiences and historical data using deep learning algorithms and to allow the driver to interrupt or completely disengage the autonomous system. Approaching such an issue could go through two main paths. The first path would be to utilize already existing sensors that have complementary outputs and enhance the fusion algorithm through deep learning approaches [168]. The second path would be to invest in the sensor hardware technology, as seen in short-wave gated camera and short-wave infrared LiDAR approaches [169]. Both paths have room for further development and enhancement.

5.2. Landmark Map-Matching

Improvement of localization and mapping is an ongoing research topic in the field of AV systems. It is vital to achieving a sub-decimeter accuracy level to avoid collisions and navigate a vehicle safely. One of the recently emerging techniques is to improve localization by detecting repetitive and distinct landmarks, such as light poles, traffic signs, or road markings, and compare their perceived location with an a priori offline map. Most of the previous work relies on traditional fusion algorithms with inefficient detection algorithms [170–172]. Replacing those methods with deep learning algorithms will accelerate learning if such landmarks and their possible variations, without the need to define them explicitly. The generalization ability of deep learning methods will enhance the reliability of the landmark matching, as its efficiency has already been proven in many related fields, such as object recognition and detection. An important example is the emergence of 3D computer vision and 3D image understanding, which refer to the analysis and recognition of the objects using volumetric images and point clouds [11,173–175]. Benefiting from both visual and geometrical information, 3D or shape-based computer vision methods can be significantly more useful than 2D or image-based methods in landmark recognition and matching. The superiority of 3D computer vision methods is because volumetric images contain more information and features of the objects and are less affected by camouflage, disguise, lighting conditions, image quality, and noise. However, three-dimensional analyses of volumetric images are also more complex, and hence more prone to error, if not treated properly. Thus, the implementation of the 3D computer vision paradigm in real-world settings imposes additional challenges that need to be addressed before it becomes a practical and reliable solution for AV applications.

5.3. Deep Learning Algorithms for Localization

Undoubtedly, it can be concluded that deep learning algorithms, in particular CNN, have been heavily applied to perform environment perception. CNN is able to learn features automatically and is very powerful in image-related tasks; hence, it is the ultimate choice for perception, where the majority of the efforts include image recognition and classification. In contrast, applying deep learning to localization has not drawn the same attention or reached the same level of maturity. Thus, there is great potential to apply RNN algorithms to tackle the sequential localization data and improve it further.

Deep learning has been used to replace certain modules of the traditional SLAM algorithms, and so far has improved the performance of localization and mapping to a certain extent. In the future, learning algorithms may offer an end-to-end deep learning SLAM system that can avoid feature modeling and data association, and consequently reduce errors and uncertainties associated with unmodeled dynamics and imperfect modelling. Moreover, similar to the VO end-to-end systems, the SLAM algorithms developed in this fashion will maintain a unified benchmark, making it possible to compare the performance of different approaches.

5.4. Issues to Solve: Cybersecurity, Reliability, and Repeatability

While deep learning approaches have dramatically improved different AV perception and localization modules, it is important to stress that these approaches require large datasets that are generated over an extended period of time. The outcomes of these approaches depend on the quality and comprehensiveness of the training datasets, and hence the results could vary in terms of relevance and reliability. Merging classical model-based methods and deep learning approaches can improve the robustness and reliability of the existing methods.

One important concern regarding the certification and homologation of the perception and localization of deep learning-based approaches is to guarantee the maintenance of their high level of efficiency. On the other hand, some recent experiments have revealed the sensitivity of the data-driven approaches to small disturbances and interferences in the sensor data. In [176], for instance, the author proposed adversarial physical conditions, which compromised object recognition and subsequently

misled the whole system. AdvHat introduced in [177] is an interesting adversarial attack method that attacks face ID systems. This method can easily breach the best public face ID model. The same approach may be used to attack road perception functions and cause huge damage to the AV system. Additionally, [178,179] introduced an overview that illustrates how deep learning methods can be deceived and breached. Nevertheless, other researchers have demonstrated the robustness of deep learning algorithms against computation failures [180].

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Singh, S. *Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey. Traffic Safety Facts Crash Stats. Report No. DOT HS 812 115*; National Center for Statistics and Analysis: Washington, DC, USA, 2015.
2. Olia, A.; Abdelgawad, H.; Abdulhai, B.; Razavi, S.N. Assessing the Potential Impacts of Connected Vehicles: Mobility, Environmental, and Safety Perspectives. *J. Intell. Transp. Syst.* **2016**, *20*, 229–243. [CrossRef]
3. Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles (J3016 Ground Vehicle Standard)—SAE Mobilus. Available online: https://saemobilus.sae.org/content/j3016_201806 (accessed on 23 October 2019).
4. Learn More About General Motors' Approach to Safely Putting Self-Driving Cars on the Roads in 2019. Available online: https://www.gm.com/our-stories/self-driving-cars.html (accessed on 23 October 2019).
5. Autopilot. Available online: https://www.tesla.com/autopilot (accessed on 23 October 2019).
6. BMW Group, Intel and Mobileye Team Up to Bring Fully Autonomous Driving to Streets by 2021. Available online: https://newsroom.intel.com/news-releases/intel-bmw-group-mobileye-autonomous-driving/ (accessed on 23 October 2019).
7. Katrakazas, C.; Quddus, M.; Chen, W.-H.; Deka, L. Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. *Transp. Res. Part C Emerg. Technol.* **2015**, *60*, 416–442. [CrossRef]
8. Pendleton, S.; Andersen, H.; Du, X.; Shen, X.; Meghjani, M.; Eng, Y.; Rus, D.; Ang, M. Perception, Planning, Control, and Coordination for Autonomous Vehicles. *Machines* **2017**, *5*, 6. [CrossRef]
9. Kaviani, S.; O'Brien, M.; Van Brummelen, J.; Najjaran, H.; Michelson, D. INS/GPS localization for reliable cooperative driving. In Proceedings of the 2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), Vancouver, BC, Canada, 15–18 May 2016; pp. 1–4.
10. Kato, S.; Tsugawa, S.; Tokuda, K.; Matsui, T.; Fujii, H. Vehicle control algorithms for cooperative driving with automated vehicles and intervehicle communications. *IEEE Trans. Intell. Transp. Syst.* **2002**, *3*, 155–161. [CrossRef]
11. Chen, X.; Chen, Y.; Najjaran, H. 3D object classification with point convolution network. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 783–788.
12. Siegwart, R.; Nourbakhsh, I.R.; Scaramuzza, D. *Introduction to Autonomous Mobile Robots*; MIT Press: Cambridge, MA, USA, 2011; ISBN 978-0-262-01535-6.
13. Pirník, R.; Hruboš, M.; Nemec, D.; Mravec, T.; Božek, P. Integration of Inertial Sensor Data into Control of the Mobile Platform. In *Proceedings of the 2015 Federated Conference on Software Development and Object Technologies*; Janech, J., Kostolny, J., Gratkowski, T., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 271–282.
14. Božek, P.; Bezák, P.; Nikitin, Y.; Fedorko, G.; Fabian, M. Increasing the production system productivity using inertial navigation. *Manuf. Technol.* **2015**, *15*, 274–278. [CrossRef]
15. Aubert, D.; Brémond, R.; Cord, A.; Dumont, E.; Gruyer, D.; Hautière, N.; Nicolle, P.; Tarel, J.P.; Boucher, V.; Charbonnier, P.; et al. Digital imaging for assessing and improving highway visibility. In Proceedings of the Transport Research Arena 2014 (TRA 2014), Paris, France, 14–17 April 2014; pp. 14–17.
16. Cord, A.; Gimonet, N. Detecting Unfocused Raindrops: In-Vehicle Multipurpose Cameras. *IEEE Robot. Autom. Mag.* **2014**, *21*, 49–56. [CrossRef]

17. Cord, A.; Aubert, D. Process and Device for Detection of Drops in a Digital Image and Computer Program for Executing This Method. U.S. Patent US9058643B2, 16 June 2015.
18. Hu, X.; Rodríguez, F.S.A.; Gepperth, A. A multi-modal system for road detection and segmentation. In Proceedings of the 2014 IEEE Intelligent Vehicles Symposium Proceedings, Dearborn, MI, USA, 8–11 June 2014; pp. 1365–1370.
19. Xiao, L.; Wang, R.; Dai, B.; Fang, Y.; Liu, D.; Wu, T. Hybrid conditional random field based camera-LIDAR fusion for road detection. *Inf. Sci.* **2018**, *432*, 543–558. [CrossRef]
20. Shinzato, P.Y.; Wolf, D.F.; Stiller, C. Road terrain detection: Avoiding common obstacle detection assumptions using sensor fusion. In Proceedings of the 2014 IEEE Intelligent Vehicles Symposium Proceedings, Dearborn, MI, USA, 8–11 June 2014; IEEE: Dearborn, MI, USA, 2014; pp. 687–692.
21. Choi, E.J.; Park, D.J. Human detection using image fusion of thermal and visible image with new joint bilateral filter. In Proceedings of the 5th International Conference on Computer Sciences and Convergence Information Technology, Seoul, Korea, 30 November–2 December 2010; pp. 882–885.
22. Torresan, H.; Turgeon, B.; Ibarra-Castanedo, C.; Hebert, P.; Maldague, X.P. Advanced surveillance systems: Combining video and thermal imagery for pedestrian detection. Presented at the SPIE, Orlando, FL, USA, 13–15 April 2004; SPIE: Bellingham, WA, USA, 2004. [CrossRef]
23. Mees, O.; Eitel, A.; Burgard, W. Choosing smartly: Adaptive multimodal fusion for object detection in changing environments. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; IEEE: Daejeon, Korea, 2016; pp. 151–156.
24. Vandersteegen, M.; Van Beeck, K.; Goedemé, T. Real-Time Multispectral Pedestrian Detection with a Single-Pass Deep Neural Network. In *Image Analysis and Recognition*; Campilho, A., Karray, F., ter Haar Romeny, B., Eds.; Springer International Publishing: Cham, Switzerland, 2018; Volume 10882, pp. 419–426. ISBN 978-3-319-92999-6.
25. Fritsche, P.; Zeise, B.; Hemme, P.; Wagner, B. Fusion of radar, LiDAR and thermal information for hazard detection in low visibility environments. In Proceedings of the 2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR), Shanghai, China, 11–13 October 2017; pp. 96–101.
26. Chen, X.; Ma, H.; Wan, J.; Li, B.; Xia, T. Multi-view 3D Object Detection Network for Autonomous Driving. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: Honolulu, HI, USA, 2017; pp. 6526–6534.
27. Wang, S.; Deng, Z.; Yin, G. An Accurate GPS-IMU/DR Data Fusion Method for Driverless Car Based on a Set of Predictive Models and Grid Constraints. *Sensors* **2016**, *16*, 280. [CrossRef]
28. Saadeddin, K.; Abdel-Hafez, M.F.; Jaradat, M.A.; Jarrah, M.A. Performance enhancement of low-cost, high-accuracy, state estimation for vehicle collision prevention system using ANFIS. *Mech. Syst. Signal Process.* **2013**, *41*, 239–253. [CrossRef]
29. Moutarde, F.; Bresson, G.; Li, Y.; Joly, C. Vehicle absolute ego-localization from vision, using only pre-existing geo-referenced panoramas. In Proceedings of the Reliability and Statistics in Transportation and Communications, Riga, Latvia, 16–19 October 2019.
30. Bresson, G.; Yu, L.; Joly, C.; Moutarde, F. Urban Localization with Street Views using a Convolutional Neural Network for End-to-End Camera Pose Regression. In Proceedings of the 2019 IEEE Intelligent Vehicles Symposium (IV), Paris, France, 9–12 June 2019; pp. 1199–1204.
31. Bresson, G.; Rahal, M.-C.; Gruyer, D.; Revilloud, M.; Alsayed, Z. A cooperative fusion architecture for robust localization: Application to autonomous driving. In Proceedings of the 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), Rio de Janeiro, Brazil, 1–4 November 2016; pp. 859–866.
32. Gruyer, D.; Belaroussi, R.; Revilloud, M. Accurate lateral positioning from map data and road marking detection. *Expert Syst. Appl.* **2016**, *43*, 1–8. [CrossRef]
33. Gruyer, D.; Magnier, V.; Hamdi, K.; Claussmann, L.; Orfila, O.; Rakotonirainy, A. Perception, information processing and modeling: Critical stages for autonomous driving applications. *Annu. Rev. Control* **2017**, *44*, 323–341. [CrossRef]
34. Schlosser, J.; Chow, C.K.; Kira, Z. Fusing LIDAR and images for pedestrian detection using convolutional neural networks. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; IEEE: Stockholm, Sweden, 2016; pp. 2198–2205.

35. Melotti, G.; Premebida, C.; Gonçalves, N.M.D.S.; Nunes, U.J.; Faria, D.R. Multimodal CNN Pedestrian Classification: A Study on Combining LIDAR and Camera Data. In Proceedings of the 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, 4–7 November 2018; IEEE: Maui, HI, USA, 2018; pp. 3138–3143.
36. Labayrade, R.; Gruyer, D.; Royere, C.; Perrollaz, M.; Aubert, D. Obstacle Detection Based on Fusion between Stereovision and 2D Laser Scanner. In *Mobile Robots: Perception & Navigation*; Kolski, S., Ed.; Pro Literatur Verlag: Augsburg, Germany, 2007.
37. Liu, J.; Zhang, S.; Wang, S.; Metaxas, D. Multispectral Deep Neural Networks for Pedestrian Detection. *arXiv* **2016**, arXiv:1611.02644.
38. Hou, Y.-L.; Song, Y.; Hao, X.; Shen, Y.; Qian, M.; Chen, H. Multispectral pedestrian detection based on deep convolutional neural networks. *Infrared Phys. Technol.* **2018**, *94*, 69–77. [CrossRef]
39. Wagner, J.; Fischer, V.; Herman, M.; Behnke, S. Multispectral Pedestrian Detection using Deep Fusion Convolutional Neural Networks. In Proceedings of the ESANN, Bruges, Belgium, 27–29 April 2016.
40. Lee, Y.; Bui, T.D.; Shin, J. Pedestrian Detection based on Deep Fusion Network using Feature Correlation. In Proceedings of the 2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Honolulu, HI, USA, 12–15 November 2018; IEEE: Honolulu, HI, USA, 2018; pp. 694–699.
41. Zheng, Y.; Izzat, I.H.; Ziaee, S. GFD-SSD: Gated Fusion Double SSD for Multispectral Pedestrian Detection. *arXiv* **2019**, arXiv:1903.06999.
42. Shopovska, I.; Jovanov, L.; Philips, W. Deep Visible and Thermal Image Fusion for Enhanced Pedestrian Visibility. *Sensors* **2019**, *19*, 3727. [CrossRef]
43. Gu, S.; Lu, T.; Zhang, Y.; Alvarez, J.M.; Yang, J.; Kong, H. 3-D LiDAR + Monocular Camera: An Inverse-Depth-Induced Fusion Framework for Urban Road Detection. *IEEE Trans. Intell. Veh.* **2018**, *3*, 351–360. [CrossRef]
44. Yang, F.; Yang, J.; Jin, Z.; Wang, H. A Fusion Model for Road Detection based on Deep Learning and Fully Connected CRF. In Proceedings of the 2018 13th Annual Conference on System of Systems Engineering (SoSE), Paris, France, 19–22 June 2018; IEEE: Paris, France, 2018; pp. 29–36.
45. Lv, X.; Liu, Z.; Xin, J.; Zheng, N. A Novel Approach for Detecting Road Based on Two-Stream Fusion Fully Convolutional Network. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; IEEE: Changshu, China, 2018; pp. 1464–1469.
46. Caltagirone, L.; Bellone, M.; Svensson, L.; Wahde, M. LIDAR-Camera Fusion for Road Detection Using Fully Convolutional Neural Networks. *Robot. Auton. Syst.* **2019**, *111*, 125–131. [CrossRef]
47. Zhang, Y.; Morel, O.; Blanchon, M.; Seulin, R.; Rastgoo, M.; Sidibé, D. Exploration of Deep Learning-based Multimodal Fusion for Semantic Road Scene Segmentation. In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*; SCITEPRESS—Science and Technology Publications: Prague, Czech Republic, 2019; pp. 336–343.
48. Kato, T.; Ninomiya, Y.; Masaki, I. An obstacle detection method by fusion of radar and motion stereo. *IEEE Trans. Intell. Transp. Syst.* **2002**, *3*, 182–188. [CrossRef]
49. Bertozzi, M.; Bombini, L.; Cerri, P.; Medici, P.; Antonello, P.C.; Miglietta, M. Obstacle detection and classification fusing radar and vision. In Proceedings of the 2008 IEEE Intelligent Vehicles Symposium, Eindhoven, The Netherlands, 4–6 June 2008; pp. 608–613.
50. Du, X.; Ang, M.H.; Rus, D. Car detection for autonomous vehicle: LIDAR and vision fusion approach through deep learning framework. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; IEEE: Vancouver, BC, Canada, 2017; pp. 749–754.
51. Valente, M.; Joly, C.; de La Fortelle, A. Deep Sensor Fusion for Real-Time Odometry Estimation. *arXiv* **2019**, arXiv:1908.00524.
52. Alatise, M.B.; Hancke, G.P. Pose Estimation of a Mobile Robot Based on Fusion of IMU Data and Vision Data Using an Extended Kalman Filter. *Sensors* **2017**, *17*, 2164. [CrossRef] [PubMed]
53. Bresson, G.; Alsayed, Z.; Yu, L.; Glaser, S. Simultaneous Localization and Mapping: A Survey of Current Trends in Autonomous Driving. *IEEE Trans. Intell. Veh.* **2017**, *2*, 194–220. [CrossRef]
54. Jaradat, M.A.K.; Abdel-Hafez, M.F. Non-Linear Autoregressive Delay-Dependent INS/GPS Navigation System Using Neural Networks. *IEEE Sens. J.* **2017**, *17*, 1105–1115. [CrossRef]

55. Rohani, M.; Gingras, D.; Gruyer, D. A Novel Approach for Improved Vehicular Positioning Using Cooperative Map Matching and Dynamic Base Station DGPS Concept. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 230–239. [CrossRef]
56. Hall, D.L.; Llinas, J. An Introduction to Multisensor Data Fusion. *Proc. IEEE* **1997**, *85*, 18. [CrossRef]
57. Bhateja, V.; Patel, H.; Krishn, A.; Sahu, A.; Lay-Ekuakille, A. Multimodal Medical Image Sensor Fusion Framework Using Cascade of Wavelet and Contourlet Transform Domains. *IEEE Sens. J.* **2015**, *15*, 6783–6790. [CrossRef]
58. Liu, X.; Liu, Q.; Wang, Y. Remote Sensing Image Fusion Based on Two-stream Fusion Network. *Inf. Fusion* **2019**. [CrossRef]
59. Smaili, C.; Najjar, M.E.E.; Charpillet, F. Multi-sensor Fusion Method Using Dynamic Bayesian Network for Precise Vehicle Localization and Road Matching. In Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2007), Patras, Greece, 29–31 October 2007; Volume 1, pp. 146–151.
60. Dasarathy, B.V. Sensor fusion potential exploitation-innovative architectures and illustrative applications. *Proc. IEEE* **1997**, *85*, 24–38. [CrossRef]
61. Feng, D.; Haase-Schuetz, C.; Rosenbaum, L.; Hertlein, H.; Glaeser, C.; Timm, F.; Wiesbeck, W.; Dietmayer, K. Deep Multi-modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges. *IEEE Trans. Intell. Transp. Syst.* **2019**. [CrossRef]
62. Malviya, A.; Bhirud, S.G. Wavelet based multi-focus image fusion. In Proceedings of the 2009 International Conference on Methods and Models in Computer Science (ICM2CS), Delhi, India, 14–15 December 2009; pp. 1–6.
63. Guan, D.; Cao, Y.; Yang, J.; Cao, Y.; Yang, M.Y. Fusion of multispectral data through illumination-aware deep neural networks for pedestrian detection. *Inf. Fusion* **2019**, *50*, 148–157. [CrossRef]
64. Castanedo, F. A Review of Data Fusion Techniques. *Sci. World J.* **2013**, *2013*, 1–19. [CrossRef]
65. Pires, I.; Garcia, N.; Pombo, N.; Flórez-Revuelta, F. From Data Acquisition to Data Fusion: A Comprehensive Review and a Roadmap for the Identification of Activities of Daily Living Using Mobile Devices. *Sensors* **2016**, *16*, 184. [CrossRef]
66. Van Brummelen, J.; O'Brien, M.; Gruyer, D.; Najjaran, H. Autonomous vehicle perception: The technology of today and tomorrow. *Transp. Res. Part C Emerg. Technol.* **2018**, *89*, 384–406. [CrossRef]
67. Santoso, F.; Garratt, M.A.; Anavatti, S.G. Visual–Inertial Navigation Systems for Aerial Robotics: Sensor Fusion and Technology. *IEEE Trans. Autom. Sci. Eng.* **2017**, *14*, 260–275. [CrossRef]
68. Jaradat, M.A.K.; Abdel-Hafez, M.F. Enhanced, Delay Dependent, Intelligent Fusion for INS/GPS Navigation System. *IEEE Sens. J.* **2014**, *14*, 1545–1554. [CrossRef]
69. Alkhawaja, F.; Jaradat, M.; Romdhane, L. Techniques of Indoor Positioning Systems (IPS): A Survey. In Proceedings of the 2019 Advances in Science and Engineering Technology International Conferences (ASET), Dubai, UAE, 26 March–10 April 2019; pp. 1–8.
70. Luo, R.C.; Chang, C.-C. Multisensor Fusion and Integration: A Review on Approaches and Its Applications in Mechatronics. *IEEE Trans. Ind. Inform.* **2012**, *8*, 49–60. [CrossRef]
71. Khaleghi, B.; Khamis, A.; Karray, F.O.; Razavi, S.N. Multisensor data fusion: A review of the state-of-the-art. *Inf. Fusion* **2013**, *14*, 28–44. [CrossRef]
72. Nagla, K.S.; Uddin, M.; Singh, D. Multisensor Data Fusion and Integration for Mobile Robots: A Review. *IAES Int. J. Robot. Autom. IJRA* **2014**, *3*, 131–138. [CrossRef]
73. Vincke, B.; Lambert, A.; Gruyera, D.; Elouardi, A.; Seignez, E. Static and dynamic fusion for outdoor vehicle localization. In Proceedings of the 2010 11th International Conference on Control Automation Robotics Vision, Singapore, 7–10 December 2010; pp. 437–442.
74. Kueviakoe, K.; Wang, Z.; Lambert, A.; Frenoux, E.; Tarroux, P. Localization of a Vehicle: A Dynamic Interval Constraint Satisfaction Problem-Based Approach. Available online: https://www.hindawi.com/journals/js/2018/3769058/ (accessed on 11 May 2020).
75. Wang, Z.; Lambert, A. A Reliable and Low Cost Vehicle Localization Approach Using Interval Analysis. In Proceedings of the 2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech), Athens, Greece, 12–15 August 2018; pp. 480–487.

76. McCulloch, W.S.; Pitts, W. A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* **1943**, *5*, 115–133. [CrossRef]
77. Ouyang, W.; Wang, X.; Zeng, X.; Qiu, S.; Luo, P.; Tian, Y.; Li, H.; Yang, S.; Wang, Z.; Loy, C.-C.; et al. DeepID-Net: Deformable Deep Convolutional Neural Networks for Object Detection. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015.
78. Gulshan, V.; Peng, L.; Coram, M.; Stumpe, M.C.; Wu, D.; Narayanaswamy, A.; Venugopalan, S.; Widner, K.; Madams, T.; Cuadros, J.; et al. Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. *JAMA* **2016**, *316*, 2402–2410. [CrossRef]
79. Chen, X.; Kundu, K.; Zhang, Z.; Ma, H.; Fidler, S.; Urtasun, R. Monocular 3D Object Detection for Autonomous Driving. In Proceedings of the Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016. [CrossRef]
80. Yan, S.; Teng, Y.; Smith, J.S.; Zhang, B. Driver behavior recognition based on deep convolutional neural networks. In Proceedings of the 2016 12th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD), Changsha, China, 13–15 August 2016; pp. 636–641.
81. Zhao, Y.; Li, J.; Yu, L. A deep learning ensemble approach for crude oil price forecasting. *Energy Econ.* **2017**, *66*, 9–16. [CrossRef]
82. Matsugu, M.; Mori, K.; Mitari, Y.; Kaneda, Y. Subject independent facial expression recognition with robust face detection using a convolutional neural network. *Neural Netw.* **2003**, *16*, 555–559. [CrossRef]
83. Gao, H.; Cheng, B.; Wang, J.; Li, K.; Zhao, J.; Li, D. Object Classification Using CNN-Based Fusion of Vision and LIDAR in Autonomous Vehicle Environment. *IEEE Trans. Ind. Inform.* **2018**, *14*, 4224–4231. [CrossRef]
84. Melotti, G.; Asvadi, A.; Premebida, C. CNN-LIDAR pedestrian classification: Combining range and reflectance data. In Proceedings of the 2018 IEEE International Conference on Vehicular Electronics and Safety (ICVES), Madrid, Spain, 12–14 September 2018; IEEE: Madrid, Spain, 2018; pp. 1–6.
85. Xiong, W.; Wu, L.; Alleva, F.; Droppo, J.; Huang, X.; Stolcke, A. The Microsoft 2017 Conversational Speech Recognition System. In Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada, 15–20 April 2018; pp. 5934–5938.
86. Mao, J.; Xu, W.; Yang, Y.; Wang, J.; Huang, Z.; Yuille, A. Deep Captioning with Multimodal Recurrent Neural Networks (m-RNN). *arXiv* **2014**, arXiv:1412.6632.
87. Shi, H.; Xu, M.; Li, R. Deep Learning for Household Load Forecasting—A Novel Pooling Deep RNN. *IEEE Trans. Smart Grid* **2018**, *9*, 5271–5280. [CrossRef]
88. Conneau, A.; Schwenk, H.; Barrault, L.; Lecun, Y. Very Deep Convolutional Networks for Text Classification. *arXiv* **2016**, arXiv:1606.01781.
89. Hongliang, C.; Xiaona, Q. The Video Recommendation System Based on DBN. In Proceedings of the 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, Liverpool, UK, 26–28 October 2015; pp. 1016–1021. [CrossRef]
90. Sazal, M.M.R.; Biswas, S.K.; Amin, M.F.; Murase, K. Bangla handwritten character recognition using deep belief network. In Proceedings of the 2013 International Conference on Electrical Information and Communication Technology (EICT), Khulna, Bangladesh, 13–15 February 2014; pp. 1–5.
91. Mohamed, A.; Dahl, G.; Hinton, G. Deep belief networks for phone recognition. In Proceedings of the NIPS Workshop on Deep Learning for Speech Recognition and Related Applications; MIT Press: Whister, BC, Canada, 2009; Volume 1, p. 39.
92. Hinton, G.E. Reducing the Dimensionality of Data with Neural Networks. *Science* **2006**, *313*, 504–507. [CrossRef]
93. Krizhevsky, A.; Hinton, G.E. Using very deep autoencoders for content-based image retrieval. In Proceedings of the ESANN, Bruges, Belgium, 27–29 April 2011.
94. Lu, X.; Tsao, Y.; Matsuda, S.; Hori, C. Speech enhancement based on deep denoising autoencoder. In Proceedings of the Annual Conference of International Speech Communication Association; INTERSPEECH, Lyon, France, 25–29 August 2013; pp. 436–440.
95. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems—Volume 1*; Curran Associates Inc.: Red Hook, NY, USA, 2012; pp. 1097–1105.

96. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *arXiv* **2013**, arXiv:1311.2524.
97. He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In Proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence; IEEE: Piscataway, NJ, USA, 2015; pp. 1904–1916. [CrossRef]
98. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
99. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *arXiv* **2015**, arXiv:150601497. [CrossRef]
100. Li, C.; Song, D.; Tong, R.; Tang, M. Illumination-aware faster R-CNN for robust multispectral pedestrian detection. *Pattern Recognit.* **2019**, *85*, 161–171. [CrossRef]
101. Kim, J.H.; Batchuluun, G.; Park, K.R. Pedestrian detection based on faster R-CNN in nighttime by fusing deep convolutional features of successive images. *Expert Syst. Appl.* **2018**, *114*, 15–33. [CrossRef]
102. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. *arXiv* **2015**, arXiv:150602640.
103. Redmon, J.; Farhadi, A. YOLO9000: Better, Faster, Stronger. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6517–6525.
104. Redmon, J.; Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv* **2018**, arXiv:180402767.
105. Asvadi, A.; Garrote, L.; Premebida, C.; Peixoto, P.J.; Nunes, U. Multimodal vehicle detection: Fusing 3D-LIDAR and color camera data. *Pattern Recognit. Lett.* **2018**, *115*, 20–29. [CrossRef]
106. Wang, H.; Lou, X.; Cai, Y.; Li, Y.; Chen, L. Real-Time Vehicle Detection Algorithm Based on Vision and Lidar Point Cloud Fusion. Available online: https://www.hindawi.com/journals/js/2019/8473980/ (accessed on 18 August 2019).
107. Dou, J.; Fang, J.; Li, T.; Xue, J. Boosting CNN-Based Pedestrian Detection via 3D LiDAR Fusion in Autonomous Driving. In *Proceedings of the Image and Graphics*; Zhao, Y., Kong, X., Taubman, D., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 3–13.
108. Han, J.; Liao, Y.; Zhang, J.; Wang, S.; Li, S. Target Fusion Detection of LiDAR and Camera Based on the Improved YOLO Algorithm. *Mathematics* **2018**, *6*, 213. [CrossRef]
109. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. *arXiv* **2015**, arXiv:1512.02325.
110. Kim, J.; Choi, J.; Kim, Y.; Koh, J.; Chung, C.C.; Choi, J.W. Robust Camera Lidar Sensor Fusion Via Deep Gated Information Fusion Network. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; IEEE: Changshu, China, 2018; pp. 1620–1625.
111. Li, Z.; Zhou, F. FSSD: Feature Fusion Single Shot Multibox Detector. *arXiv* **2017**, arXiv:171200960.
112. Fu, C.-Y.; Liu, W.; Ranga, A.; Tyagi, A.; Berg, A.C. DSSD: Deconvolutional Single Shot Detector. *arXiv* **2017**, arXiv:abs/1701.06659.
113. Chen, C.; Seff, A.; Kornhauser, A.; Xiao, J. DeepDriving: Learning Affordance for Direct Perception in Autonomous Driving. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 2722–2730.
114. Kim, Y.-H.; Jang, J.-I.; Yun, S. End-to-end deep learning for autonomous navigation of mobile robot. In Proceedings of the 2018 IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 12–14 January 2018; pp. 1–6.
115. Pfeiffer, M.; Schaeuble, M.; Nieto, J.; Siegwart, R.; Cadena, C. From perception to decision: A data-driven approach to end-to-end motion planning for autonomous ground robots. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 1527–1533.
116. Qazizada, M.E.; Pivarčiová, E. Mobile Robot Controlling Possibilities of Inertial Navigation System. *Procedia Eng.* **2016**, *149*, 404–413. [CrossRef]
117. Caron, F.; Duflos, E.; Pomorski, D.; Vanheeghe, P. GPS/IMU data fusion using multisensor Kalman filtering: Introduction of contextual aspects. *Inf. Fusion* **2006**, *7*, 221–230. [CrossRef]
118. Qi, H.; Moore, J.B. Direct Kalman filtering approach for GPS/INS integration. *IEEE Trans. Aerosp. Electron. Syst.* **2002**, *38*, 687–693. [CrossRef]
119. Wang, G.; Han, Y.; Chen, J.; Wang, S.; Zhang, Z.; Du, N.; Zheng, Y. A GNSS/INS Integrated Navigation Algorithm Based on Kalman Filter. *IFAC-Pap.* **2018**, *51*, 232–237. [CrossRef]

120. Wan, E.A.; Merwe, R.V.D. The unscented Kalman filter for nonlinear estimation. In Proceedings of the IEEE 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium (Cat. No.00EX373), Lake Louise, AB, Canada, 4 October 2000; pp. 153–158.
121. Ndjeng Ndjeng, A.; Gruyer, D.; Glaser, S.; Lambert, A. Low cost IMU–Odometer–GPS ego localization for unusual maneuvers. *Inf. Fusion* **2011**, *12*, 264–274. [CrossRef]
122. Bacha, A.R.A.; Gruyer, D.; Lambert, A. OKPS: A Reactive/Cooperative Multi-Sensors Data Fusion Approach Designed for Robust Vehicle Localization. *Positioning* **2015**, *7*, 1–20. [CrossRef]
123. Noureldin, A.; El-Shafie, A.; Bayoumi, M. GPS/INS integration utilizing dynamic neural networks for vehicular navigation. *Inf. Fusion* **2011**, *12*, 48–57. [CrossRef]
124. Dai, H.; Bian, H.; Wang, R.; Ma, H. An INS/GNSS integrated navigation in GNSS denied environment using recurrent neural network. *Def. Technol.* **2019**. [CrossRef]
125. Kim, H.-U.; Bae, T.-S. Deep Learning-Based GNSS Network-Based Real-Time Kinematic Improvement for Autonomous Ground Vehicle Navigation. *J. Sens.* **2019**. [CrossRef]
126. Jiang, C.; Chen, Y.; Chen, S.; Bo, Y.; Li, W.; Tian, W.; Guo, J. A Mixed Deep Recurrent Neural Network for MEMS Gyroscope Noise Suppressing. *Electronics* **2019**, *8*, 181. [CrossRef]
127. Singandhupe, A.; La, H.M. A Review of SLAM Techniques and Security in Autonomous Driving. In Proceedings of the 2019 Third IEEE International Conference on Robotic Computing (IRC), Naples, Italy, 25–27 February 2019; pp. 602–607.
128. The KITTI Vision Benchmark Suite. Available online: http://www.cvlibs.net/datasets/kitti/eval_odometry.php (accessed on 12 May 2020).
129. Magnier, V. *Multi-Sensor Data Fusion for the Estimation of the Navigable Space for the Autonomous Vehicle*; University Paris Saclay and Renault: Versailles, France, 2018.
130. Kaneko, M.; Iwami, K.; Ogawa, T.; Yamasaki, T.; Aizawa, K. Mask-SLAM: Robust Feature-Based Monocular SLAM by Masking Using Semantic Segmentation. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 18–22 June 2018; pp. 371–3718.
131. Xiao, L.; Wang, J.; Qiu, X.; Rong, Z.; Zou, X. Dynamic-SLAM: Semantic monocular visual localization and mapping based on deep learning in dynamic environment. *Robot. Auton. Syst.* **2019**, *117*, 1–16. [CrossRef]
132. Yu, C.; Liu, Z.; Liu, X.; Xie, F.; Yang, Y.; Wei, Q.; Fei, Q. DS-SLAM: A Semantic Visual SLAM towards Dynamic Environments. *2018 IEEERSJ Int. Conf. Intell. Robots Syst. IROS* **2018**, 1168–1174. [CrossRef]
133. Farrokhsiar, M.; Najjaran, H. A Velocity-Based Rao-Blackwellized Particle Filter Approach to Monocular vSLAM. *J. Intell. Learn. Syst. Appl.* **2011**, *3*, 113–121. [CrossRef]
134. Tateno, K.; Tombari, F.; Laina, I.; Navab, N. CNN-SLAM: Real-Time Dense Monocular SLAM with Learned Depth Prediction. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6565–6574.
135. Laidlow, T.; Czarnowski, J.; Leutenegger, S. DeepFusion: Real-Time Dense 3D Reconstruction for Monocular SLAM using Single-View Depth and Gradient Predictions. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; pp. 4068–4074.
136. Lee, S.J.; Choi, H.; Hwang, S.S. Real-time Depth Estimation Using Recurrent CNN with Sparse Depth Cues for SLAM System. *Int. J. Control Autom. Syst.* **2020**, *18*, 206–216. [CrossRef]
137. Kuznietsov, Y.; Stuckler, J.; Leibe, B. Semi-Supervised Deep Learning for Monocular Depth Map Prediction. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2215–2223.
138. Cvišić, I.; Ćesić, J.; Marković, I.; Petrović, I. SOFT-SLAM: Computationally efficient stereo visual simultaneous localization and mapping for autonomous unmanned aerial vehicles. *J. Field Robot.* **2018**, *35*, 578–595. [CrossRef]
139. Lenac, K.; Ćesić, J.; Marković, I.; Petrović, I. Exactly sparse delayed state filter on Lie groups for long-term pose graph SLAM. *Int. J. Robot. Res.* **2018**, *37*, 585–610. [CrossRef]
140. Mur-Artal, R.; Tardós, J.D. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [CrossRef]
141. Engel, J.; Stückler, J.; Cremers, D. Large-scale direct SLAM with stereo cameras. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 1935–1942.

142. Deschaud, J.-E. IMLS-SLAM: Scan-to-Model Matching Based on 3D Data. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 2480–2485.
143. Neuhaus, F.; Koß, T.; Kohnen, R.; Paulus, D. MC2SLAM: Real-Time Inertial Lidar Odometry Using Two-Scan Motion Compensation. In *Proceedings of the Pattern Recognition*; Brox, T., Bruhn, A., Fritz, M., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 60–72.
144. Ji, K.; Chen, H.; Di, H.; Gong, J.; Xiong, G.; Qi, J.; Yi, T. CPFG-SLAM:a Robust Simultaneous Localization and Mapping based on LIDAR in Off-Road Environment. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; pp. 650–655.
145. Behley, J.; Stachniss, C. Efficient Surfel-Based SLAM using 3D Laser Range Data in Urban Environments. In *Robotics: Science and System XIV*; Carnegie Mellon University: Pittsburgh, PA, USA, 2018.
146. Hou, Y.; Zhang, H.; Zhou, S. Convolutional neural network-based image representation for visual loop closure detection. In Proceedings of the 2015 IEEE International Conference on Information and Automation, Lijiang, China, 8–10 August 2015; pp. 2238–2245.
147. Merrill, N.; Huang, G. Lightweight Unsupervised Deep Loop Closure. *arXiv* **2018**, arXiv:180507703.
148. Kang, R.; Shi, J.; Li, X.; Liu, Y.; Liu, X. DF-SLAM: A Deep-Learning Enhanced Visual SLAM System based on Deep Local Features. *arXiv* **2019**, arXiv:190107223.
149. Parisotto, E.; Chaplot, D.S.; Zhang, J.; Salakhutdinov, R. Global Pose Estimation with an Attention-based Recurrent Network. *arXiv* **2018**, arXiv:180206857.
150. Yousif, K.; Bab-Hadiashar, A.; Hoseinnezhad, R. An Overview to Visual Odometry and Visual SLAM: Applications to Mobile Robotics. *Intell. Ind. Syst.* **2015**, *1*, 289–311. [CrossRef]
151. Zhang, J.; Singh, S. Visual-lidar odometry and mapping: Low-drift, robust, and fast. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 2174–2181.
152. Melekhov, I.; Ylioinas, J.; Kannala, J.; Rahtu, E. Relative Camera Pose Estimation Using Convolutional Neural Networks. *arXiv* **2017**, arXiv:170201381.
153. Mohanty, V.; Agrawal, S.; Datta, S.; Ghosh, A.; Sharma, V.D.; Chakravarty, D. DeepVO: A Deep Learning approach for Monocular Visual Odometry. *arXiv* **2016**, arXiv:161106069.
154. Wang, S.; Clark, R.; Wen, H.; Trigoni, N. DeepVO: Towards End-to-End Visual Odometry with Deep Recurrent Convolutional Neural Networks. *IEEE Int. Conf. Robot. Autom. ICRA* **2017**, 2043–2050. [CrossRef]
155. Wang, S.; Clark, R.; Wen, H.; Trigoni, N. End-to-end, sequence-to-sequence probabilistic visual odometry through deep neural networks. *Int. J. Robot. Res.* **2018**, *37*, 513–542. [CrossRef]
156. Graeter, J.; Wilczynski, A.; Lauer, M. LIMO: Lidar-Monocular Visual Odometry. *arXiv* **2018**, arXiv:180707524.
157. Zhu, J. Image Gradient-based Joint Direct Visual Odometry for Stereo Camera. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17, Melbourne, Australia, 19–25 August 2017; pp. 4558–4564.
158. Kovalenko, D.; Korobkin, M.; Minin, A. Sensor Aware Lidar Odometry. *arXiv* **2020**, arXiv:190709167.
159. Dias, N.; Laureano, G. Accurate Stereo Visual Odometry Based on Keypoint Selection. In Proceedings of the 2019 Latin American Robotics Symposium (LARS), 2019 Brazilian Symposium on Robotics (SBR) and 2019 Workshop on Robotics in Education (WRE), Rio Grande, Brazil, 23–25 October 2019; pp. 74–79.
160. Zhou, T.; Brown, M.; Snavely, N.; Lowe, D.G. Unsupervised Learning of Depth and Ego-Motion from Video. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6612–6619.
161. Li, R.; Wang, S.; Long, Z.; Gu, D. UnDeepVO: Monocular Visual Odometry through Unsupervised Deep Learning. *arXiv* **2018**, arXiv:170906841.
162. Yang, N.; Stumberg, L.V.; Wang, R.; Cremers, D. D3VO: Deep Depth, Deep Pose and Deep Uncertainty for Monocular Visual Odometry. *arXiv* **2020**, arXiv:2003.01060.
163. Chen, D. Semi-Supervised Deep Learning Framework for Monocular Visual Odometry. 2019.
164. Valada, A.; Radwan, N.; Burgard, W. Deep Auxiliary Learning for Visual Localization and Odometry. *arXiv* **2018**, arXiv:180303642.
165. Kent, L. HERE Introduces HD Maps for Highly Automated Vehicle Testing. Available online: https://360.here.com/2015/07/20/here-introduces-hd-maps-for-highly-automated-vehicle-testing/ (accessed on 15 October 2019).

166. Lu, W.; Zhou, Y.; Wan, G.; Hou, S.; Song, S. L3-Net: Towards Learning Based LiDAR Localization for Autonomous Driving. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–21 June 2019; pp. 6382–6391.
167. Vaquero, V.; Fischer, K.; Moreno-Noguer, F.; Sanfeliu, A.; Milz, S. Improving Map Re-localization with Deep "Movable" Objects Segmentation on 3D LiDAR Point Clouds. *arXiv* **2019**, arXiv:191003336.
168. Bijelic, M.; Mannan, F.; Gruber, T.; Ritter, W.; Dietmayer, K.; Heide, F. Seeing Through Fog Without Seeing Fog: Deep Sensor Fusion in the Absence of Labeled Training Data. *arXiv* **2019**, arXiv:190208913.
169. Ritter, W.; Bijelic, M.; Gruber, T.; Kutila, M.; Holzhüter, H. DENSE: Environment Perception in Bad Weather—First Results. In *Proceedings of the Electronic Components and Systems for Automotive Applications*; Langheim, J., Ed.; Springer International Publishing: Cham, Switzerland, 2019; pp. 143–159.
170. Sefati, M.; Daum, M.; Sondermann, B.; Kreisköther, K.D.; Kampker, A. Improving vehicle localization using semantic and pole-like landmarks. In Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV), Los Angeles, CA, USA, 11–14 June 2017; pp. 13–19.
171. Fang, J.; Wang, Z.; Zhang, H.; Zong, W. Self-localization of Intelligent Vehicles Based on Environmental Contours. In Proceedings of the 2018 3rd International Conference on Advanced Robotics and Mechatronics (ICARM), Singapore, 18–20 July 2018; pp. 624–629.
172. DOrazio, L.; Conci, N.; Stoffella, F. Exploitation of road signalling for localization refinement of autonomous vehicles. In Proceedings of the 2018 International Conference of Electrical and Electronic Technologies for Automotive, Milan, Italy, 9–11 July 2018; pp. 1–6.
173. Maturana, D.; Scherer, S. VoxNet: A 3D Convolutional Neural Network for real-time object recognition. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 922–928.
174. Chen, X.; Chen, Y.; Gupta, K.; Zhou, J.; Najjaran, H. SliceNet: A proficient model for real-time 3D shape-based recognition. *Neurocomputing* **2018**, *316*, 144–155. [CrossRef]
175. Charles, R.Q.; Su, H.; Kaichun, M.; Guibas, L.J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 77–85.
176. Eykholt, K.; Evtimov, I.; Fernandes, E.; Li, B.; Rahmati, A.; Xiao, C.; Prakash, A.; Kohno, T.; Song, D. Robust Physical-World Attacks on Deep Learning Models. *arXiv* **2018**, arXiv:170708945.
177. Komkov, S.; Petiushko, A. AdvHat: Real-world adversarial attack on ArcFace Face ID system. *arXiv* **2019**, arXiv:190808705.
178. Nguyen, A.; Yosinski, J.; Clune, J. Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images. *arXiv* **2015**, arXiv:14121897.
179. Heaven, D. Why deep-learning AIs are so easy to fool. *Nature* **2019**, *574*, 163–166. [CrossRef]
180. Vialatte, J.-C.; Leduc-Primeau, F. A Study of Deep Learning Robustness against Computation Failures. *arXiv* **2017**, arXiv:170405396.