UNIVERSITY OF TURKU

Camera + LiDAR Sensor Fusion Methods for Semantic Segmentation in Autonomous Driving

A Literature Review

Department of Mechanical and Materials Engineering
Bachelor's thesis

Author:
Joonas Sallmén

Supervisors:
M.Sc. Carlos Roberto Cueto Zumaya
Prof. Wallace Moreira Bessa

16.5.2025
Turku

Bachelor's thesis

Subject: Mechanical Engineering
Author: Joonas Sallmén
Title: Camera + LiDAR Sensor Fusion Methods for Semantic Segmentation in Autonomous Driving
Supervisor(s): M.Sc. Carlos Roberto Cueto Zumaya, Prof. Wallace Moreira Bessa
Number of pages: 28 pages
Date: 16.5.2025

Autonomous vehicles require perception systems that are highly accurate, robust and capable of processing data in real-time to ensure reliable operation. To fulfill these requirements, autonomous systems can benefit from sensor fusion for semantic segmentation. The focus of this thesis is on the discussion about the fusion of a common sensor combination in autonomous vehicles; cameras and LiDAR. This thesis starts by giving an overview of semantic segmentation and sensor fusion in autonomous vehicles. Then follows explanations on fusion approaches (early, mid, late and asymmetric fusion) and common architecture types that serve as a basis for most of the fusion methods. The thesis ends with the literature review focused on deep learning-based fusion methods for camera and LiDAR, followed by a discussion on limitations of approaches and future directions.

Key words: semantic segmentation, sensor fusion, deep learning, autonomous driving, camera, LiDAR

Table of contents

1 Introduction

Semantic segmentation is a task in computer vision, where an image is divided into segments or areas of interest by pixel-wise labeling [1]. It has multiple different use cases ranging from satellite image analysis, medical industry and the automotive industry, where computer vision is getting increasingly vital with the rise of autonomy in traffic.

Autonomous vehicles rely on sensor systems, through which they can acquire accurate, robust and real-time information about their surroundings [2]. However, individual sensors have their strengths and weaknesses. Cameras provide RGB data but lack depth information and fail in low light conditions. LiDAR, on the other hand, contains the depth information needed, but point clouds can get sparse, and they are missing the RGB information that a camera has [3]. This is where the importance of sensor fusion in intelligent perception becomes evident. Sensor fusion is the act of fusing complementary data from multiple sources together for improved perception [4]. Currently, the field is developing swiftly, and new methods used for sensor fusion are coming out at a steady pace. Especially with the rise of methods based on the vision transformer (ViT), a new and updated literature review on the topic is needed.

The goal of this literature review is to review recent sensor fusion methods that could be used in autonomous driving systems that are equipped with cameras and LiDAR. This composition was chosen because the two sensors are widely used in autonomous vehicles [3]. The focus is on gathering new deep learning -based methods and grouping them by fusion approach (early, mid, late and asymmetric fusion). It was considered important to include approaches that use recent methods like self-supervised learning and ViTs.

This thesis is structured in the following way: Section 2 gives an overview of the fundamental concepts that the thesis is relies on; Section 3 explains the main architecture types used in the fusion methods and contains the literature review that is organized by fusion approach; Section 4 discusses the strengths and limitations of existing approaches, highlights trends in the field, speculates future directions and ends with a conclusion of the thesis.

2 Fundamentals and Background

2.1 Semantic Segmentation

Semantic segmentation is a core topic in the field of computer vision and serves as the basis for many complex visual tasks [5]. Compared to traditional image classification where a single label is assigned to an image or object detection which identifies objects within bounding boxes, in semantic segmentation each pixel in an image is labeled. It is used to get a detailed understanding of visual scenes in a wide range of application areas, such as autonomous driving, medical imaging and satellite image analysis. In autonomous driving, semantic segmentation is used to identify surfaces and objects from the surrounding environment. Ensuring safe navigation is crucial when autonomous vehicles operate alongside other road users and pedestrians [5]. Figure 1 shows an example of a segmented 2D image.

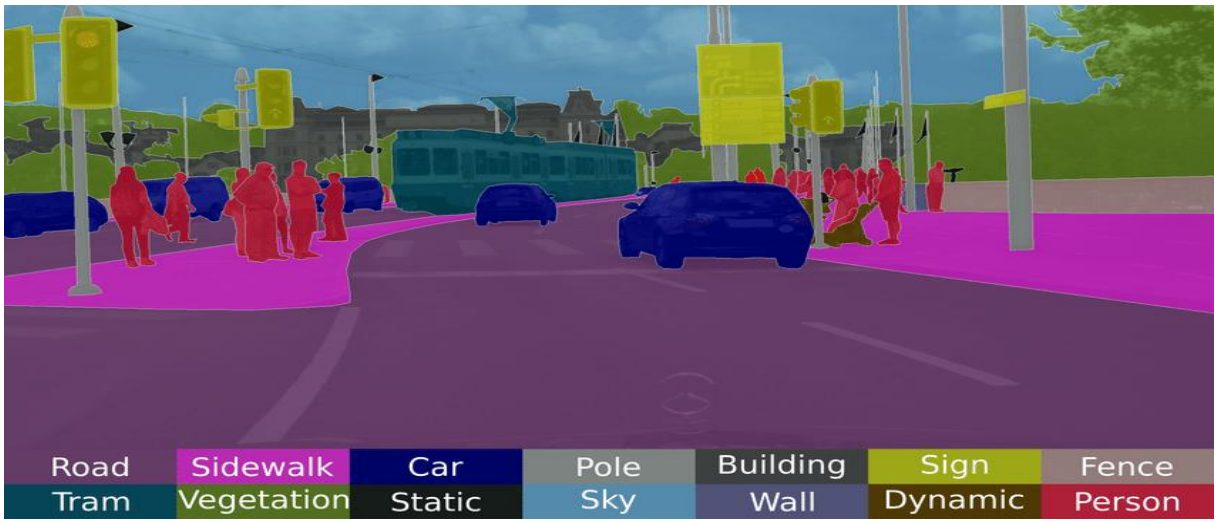


Figure 1: Segmented image [6], licensed under CC BY 4.0

Earlier approaches to semantic segmentation depended on machine learning methods such as Support Vector Machines (SVM) and Conditional Random Fields (CRF) which required extensive feature engineering and pre-processing [5]. SVMs work by representing pixels as feature vectors and separating these vectors into classes by using a hyperplane. For example, this hyperplane could separate vector representations of pixels that represent the road and pixels that do not. CRFs are used to create probabilistic models for finding connections between pixels after they have been individually labeled by another method like an SVM, therefore leading to a better segmentation result. While CRFs can be used as a post-processing layer, on their own

these traditional methods tend to be inefficient for complex problems such as segmenting multiple objects in an image [5].

Due to recent developments in deep learning, convolutional neural networks (CNN) took over and have become the most common method for semantic segmentation, offering significant improvements in segmentation accuracy while simultaneously decreasing the need for human involvement [7]. There are various deep learning architectures used for semantic segmentation, but many of them are designed around the working principle of the CNN. A CNN consists of a convolutional layer that filters input data (pixels) and detects patterns, an activation layer that learns relevant features automatically, a pooling layer reduces the size of the image while keeping all the most important information and a fully connected layer makes the decision on the final segmentation result [7]. New methods like the ViT have shown very promising results against CNN-based methods and are a key focus in ongoing research.

2.2 Sensor Fusion in Autonomous Driving

Sensor fusion is the process of combining data from multiple sensors. It is used to reduce uncertainty in the obtained information by leveraging strengths of different sensors [3]. In this thesis, we are focusing on the fusion of cameras and Light Detection and Ranging sensors (LiDAR) because they are widely used in autonomous vehicles [3]. Figure 2 demonstrates the information we can acquire from a 2D camera and LiDAR.

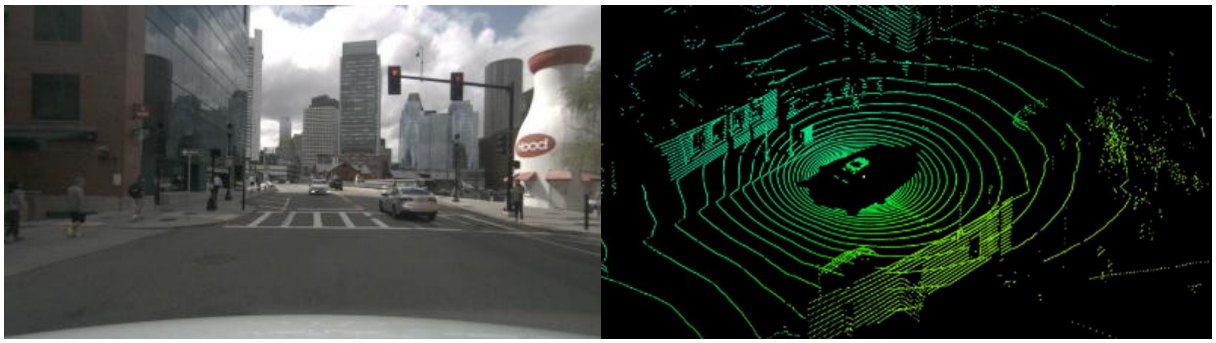


Figure 2: Camera image (left) and LiDAR point cloud (right), modified from [4] © IEEE (2024)

Perception systems must meet the following criteria; high accuracy, high robustness and rapid real-time processing. There is little room for uncertainty, and sensor fusion is a crucial technology to overcome this problem [3]. Cameras provide a 2D image with color and texture information, but they are vulnerable to low light conditions and lack depth information. Sensors like LiDAR, RADAR and Ultrasonic sensors are good complementary sensors with cameras

since they provide the information about depth and surface reflectivity that cameras do not capture [3]. LiDAR uses pulses of laser light to sense the distance between the sensor and surfaces around it. Based on this information gathered, it creates a 3D point cloud (3DPC) where each point has its position in the world coordinate system and an intensity value that represents the reflectivity of the surface where the laser made contact [3]. LiDAR provides accurate measurements and works in low-light conditions where normal cameras struggle. However, LiDAR data is increasingly sparse depending on the distance to the points measured.

2.3 Sensor Fusion Approaches

Sensor fusion approaches are typically divided into three categories based on the point in time when the information is fused. These three categories are early fusion, mid-fusion and late fusion [3], [4]. We are going to discuss each one of these more in depth and finally go over asymmetric fusion between modalities. Figure 3 contains the fusion approaches.

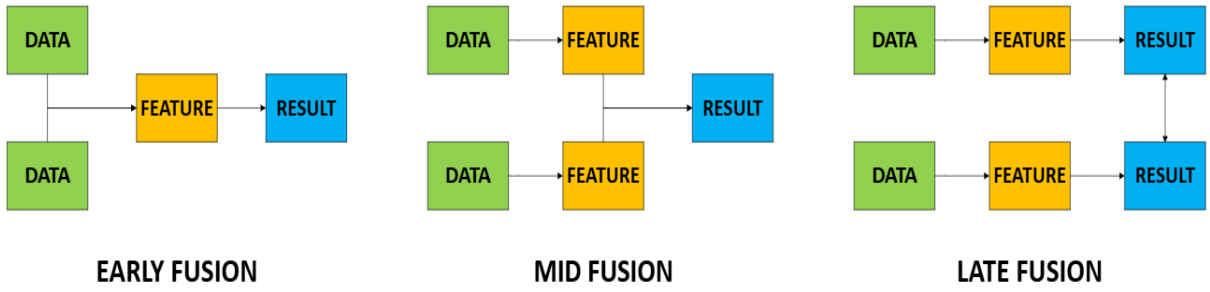


Figure 3: Traditional Fusion Approaches

2.3.1 Early Fusion

Early fusion combines sensor data before any features have been extracted. It can also be called data-data fusion because all modalities are fused at the data level [3]. This is achieved by making all inputs follow the same coordinate system, merging them into a unified tensor, and extracting features from it [4], [8]. For camera images and 3DPC this could mean turning the point cloud into a 2D representation of the points before merging or vice versa. Two common ways to achieve this are spherical projection and perspective projection. In spherical projection, LiDAR points are projected to a spherical map, where each point contains values for horizontal angle, vertical angle, depth and reflectivity, thus allowing direct point to pixel correspondences. In perspective projection, LiDAR points are projected to the camera image, but the downside of this method is that usually point clouds are sparse and require interpolation to get a corresponding point to every pixel [9]. These issues and growth of computational costs when

data gets more complex, are the main challenges with early fusion. Fusing information at an early stage can be advantageous since the fusion happens before pre-processing, causing only minimal information losses [3].

2.3.2 Mid-Fusion

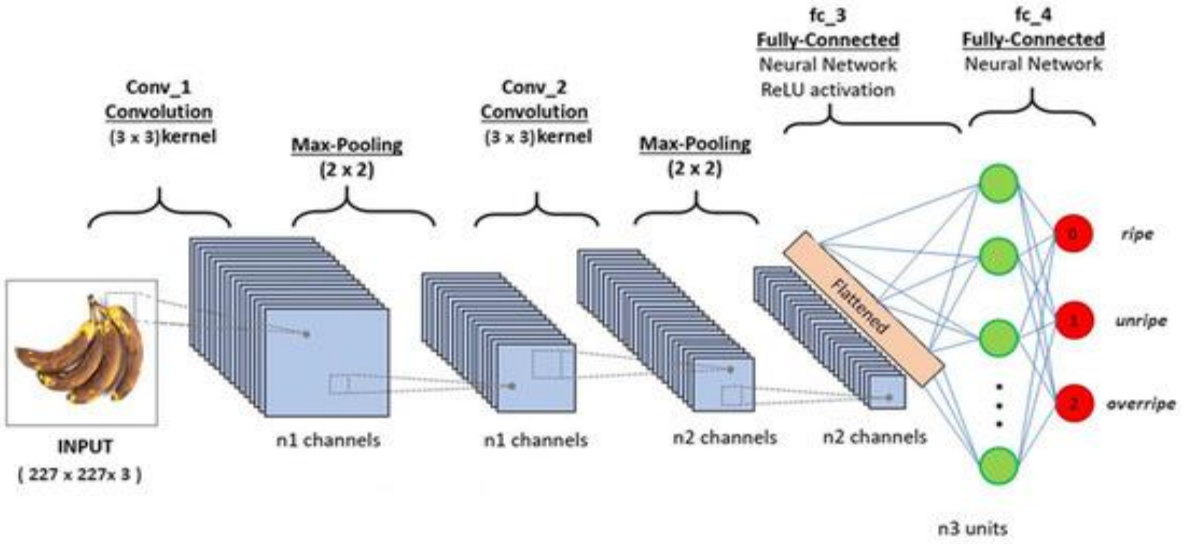
Mid-fusion happens after features have been extracted, which is why it is commonly called feature-feature fusion [3]. Data from each sensor is processed independently and after one or more intermediate layers features are extracted. After feature extraction, feature maps from each sensor are merged using concatenation (combining feature maps into a single longer list), element-wise addition (summing corresponding elements), or more advanced strategies [3], [4]. Mid-fusion is advantageous, especially when sensors capture different types of information. Extracting features before fusion allows better performance, since the data is used in its original form [3]. The time for fusion is trivial because it can happen anywhere in between the data layer and final predictions. However, determining the right time to fuse for best performance can be challenging [4].

2.3.3 Late Fusion

Late fusion combines sensor data after all modalities have been processed independently, and they have made their own final decision outputs. Outputs of each modality are then merged using weighted averaging, voting schemes or additional fusion networks to make the final prediction [3], [4]. Late fusion simplifies changing the sensor composition since the predictions are combined directly without requiring complex intermodality networks [4]. Moreover, compared to early fusion, late fusion is less sensitive to minor data misalignments since each modality is processed independently. However, challenges with late fusion include not benefiting from synergies between different sensors, which means that mid- and early fusion can make better predictions in some situations. Additionally, if different sensors make contradicting predictions, it can lead to an inconsistent or inaccurate final prediction after the fusion [3], [8].

2.3.4 Asymmetric Fusion

Asymmetric fusion refers to data being fused when modalities are in different stages of processing. In this case, one of the modalities acts as the primary source of information (e.g., LiDAR), while other sources (e.g., cameras) are there to support segmentation by providing context [3], [4]. Fusing data asymmetrically is beneficial since the fusion process can happen using lightweight methods, therefore reducing computational load [3]



.

Figure 4: Convolutional Neural Network [10], licensed under CC BY 4.0

3 Literature Review

3.1 Architecture Types

The methods we will discuss in the literature review are in most situations based on some form of a CNN or a ViT. It is typically a modification or combination of either one, but in the next three subsections, we will discuss these architecture types in detail.

3.1.1 Convolutional Neural Network (CNN)

CNN is a type of neural network that is widely used in tasks related to computer vision [5]. Figure 4 shows the structure of a CNN. A CNN consists of multiple layers, but the foundation of this kind of network is based on three types of layers:

The *Convolutional layer* is the first layer of a CNN. It takes a tensor representation of an image as input, typically consisting of three matrices, one for each RGB channel. The matrices consisting of pixel values is then filtered using a convolutional kernel, which is a basic function of a CNN. A convolutional kernel is commonly a 3x3 matrix that stores learnable weights used for pattern detection (e.g., edges and textures). As the kernel slides over an image, it calculates dot products between pixel values and kernel values inside each local patch, resulting in a feature map. Moreover, multiple kernels can be applied in the convolutional layer to generate multiple feature maps. By adding several convolutional layers, the network can progressively detect features of increasing complexity, from small edges to larger structures [7].

Pooling layer is used for reducing dimensions in the image. It uses a filter to go over the image and either selecting the maximum value inside the filter window as the output value (max pooling) or calculating an average of the values inside the filter to produce an output value (average pooling).

Fully connected layer is the final layer of the network. It connects all the nodes from the final pooling layer to produce a final output, giving a classification result using an activation function like the softmax function.

Unlike traditional CNNs used for classification, semantic segmentation requires each pixel to be individually predicted instead of a single class prediction [5]. Instead of fully connected layers, segmentation networks use upsampling layers to restore spatial resolution and assign

class labels to each pixel [5]. Many architectures used for segmentation replace the fully connected layer with deconvolution layers, bilinear upsampling, and skip connections to further improve the segmentation output. Next, we are going to describe a few common architectures based on the CNN used for segmentation.

3.1.2 Common CNN-based Architectures

CNN-based encoder-decoder [5], [7] - This architecture consists of two parts: an encoder and a decoder. During the encoding phase, the network extracts features while simultaneously reducing spatial dimensions of the feature map. These objectives are achieved through multiple convolutional and pooling layers with the former handling feature extraction and the latter responsible for downsampling. The decoder upsamples the feature map back to its original dimensions and uses transposed convolutions, skip connections or upsampling layers to refine the result. Figure 5 presents an example of the encoder-decoder, SegNet [11].

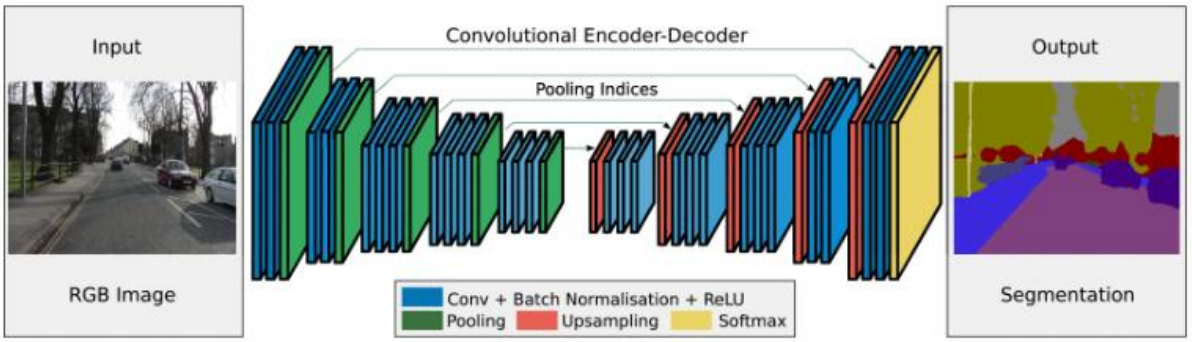


Figure 5: Example of the encoder-decoder architecture, SegNet [11], licensed under CC BY 4.0

Dilated convolution-based [7] - In this architecture the spatial resolution remains constant throughout the convolution process and feature extraction happens by changing the dilation rate of the convolution kernel. The dilation rate refers to the amount of empty space in between the pixels in the convolution kernel. Having a higher dilation rate allows the method to capture broader structures, while a lower dilation rate can be used for finer details. Figure 6 shows the effect of changing the dilation rate of the convolutional kernel.

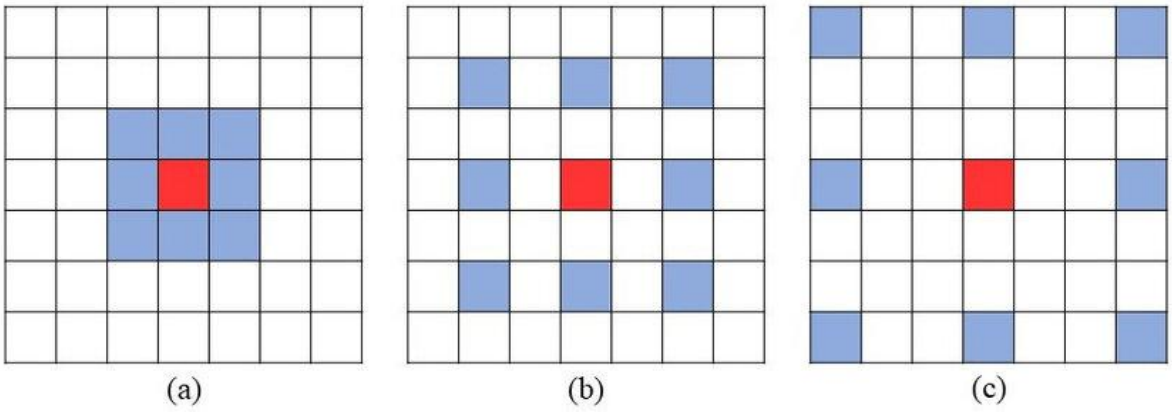


Figure 6: 3x3 convolutional kernel with dilation rates a=0, b=1 and c=2 [12], licensed under CC BY 4.0

Multi-scale feature fusion [7] - There are two main strategies to multi-scale feature fusion: parallel multibranch networks and skip connections. In parallel multibranch networks, input features are processed in multiple different scales concurrently for detecting features of varying sizes. Outputs of each branch are then merged to create a comprehensive feature representation. In skip connections, early layers of the network are fused with deeper layers using a connection that skips over the layers in between. Figure 7 demonstrates the CLFusion method [13] utilizing both parallel networks and skip connections. The fusion module gradually combines features from two parallel encoders, while skip connections link corresponding layers between the encoder and decoder.

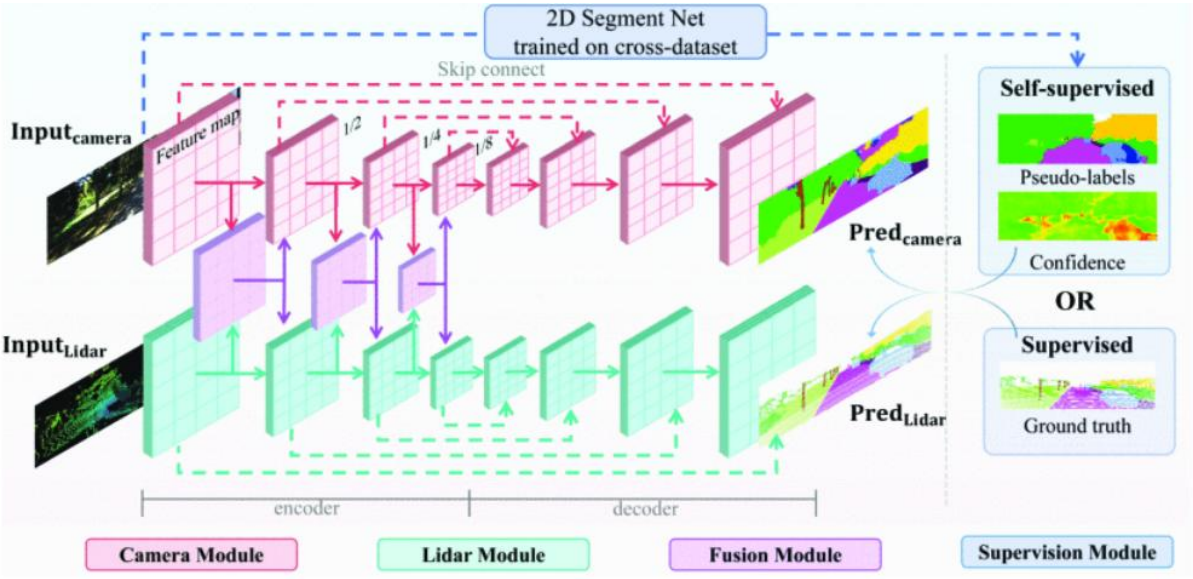


Figure 7: CLFusion 3D segmentation method [13] © (2024) IEEE

3.1.3 Vision Transformer (ViT)

The first large use case for transformers was natural language processing, where they proved to be a great success [14]. CNN-based image segmentation methods suffer from low resolution in the final output due to many pooling and convolution layers, which led to the development of the ViT [15]. Compared to CNN-based methods that learn local features, ViTs can map long-range dependencies, leading to more accurate segmentation results [15]. Transformers extract features via self-attention, whereas CNNs do so using convolutional kernels. Although ViTs have superior performance to CNNs, their downside is that they require large amounts of data to function properly [16]. A ViT has three stages: patch embedding, transformer encoder and classification. Figure 8 contains a visual representation of the architecture.

Patch embedding is the first stage in a ViT. During embedding, images are split into fixed size patches, typically 16x16 pixels. These patches are flattened into a one-dimensional vector form, and through linear projection they are mapped into fixed-size feature space that can be processed by the transformer encoder [16]. Before feeding patches into the encoder, their position in the image is memorized through positional embedding, which means we have a matrix containing all the positions of different patches [16].

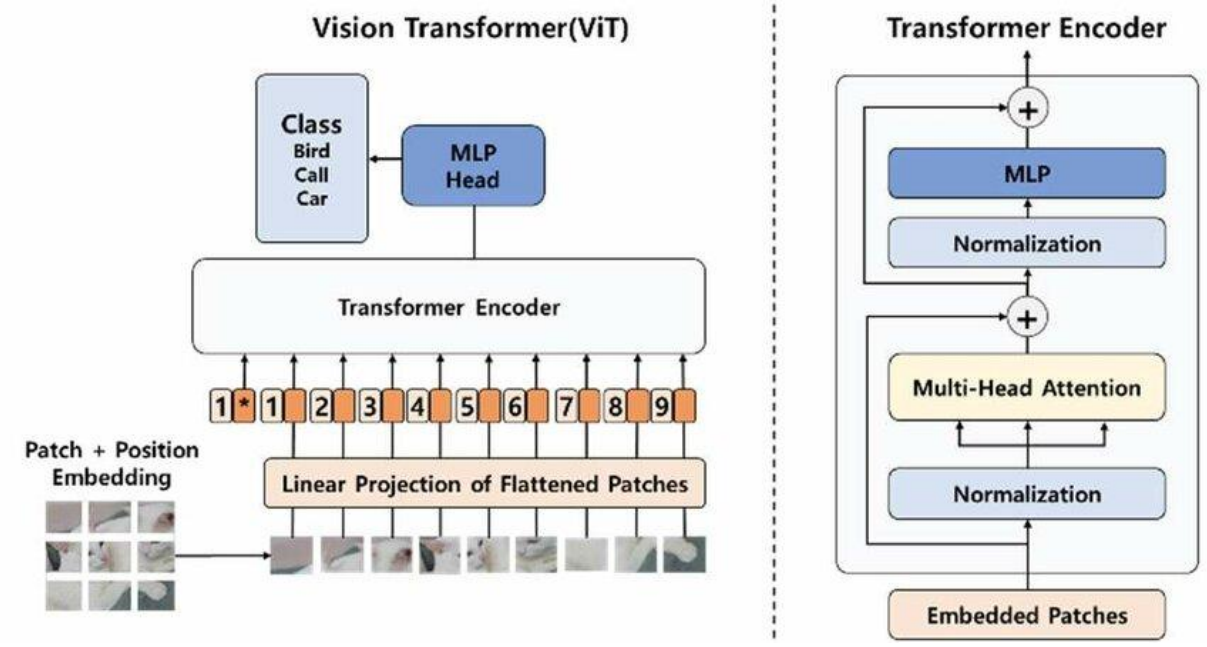


Figure 8: Vision Transformer Architecture [17], licensed under CC BY-NC-ND 4.0

The *Transformer encoder* receives the patches and processes them through multiple blocks containing three layers:

- *Multi-Head Self-Attention (MHSA)* captures long-range dependencies between patches. It takes the patch embedding and creates 3 separate vectors based on it: query, key and value. They help the model to decide which patches should focus on each other. Because it is multi-headed, each input can be split into multiple heads, each creating their own query, key and value vectors [14].

- *Multilayer Perceptron (MLP)* expands the MHSA output dimensionality, applies a non-linear activation function and projects the image back to its original size. This process helps keep the network effective and preserve information [14].

- *Normalization layer* stabilizes inputs and improves convergence.

Classification happens in the final stage of a typical ViT [16]. Similarly to the CNN structure, this last section needs to be replaced with a decoder that handles the upsampling for segmentation instead of a single class output.

3.2 Review of Recent Sensor Fusion Methods for Semantic Segmentation

This section extends on the concepts introduced in Section 2.3 by categorizing and reviewing sensor fusion methods for semantic segmentation in autonomous driving. Methods are grouped by fusion approach (early, mid, late and asymmetric) and each subsection focuses on how LiDAR and camera data are integrated within the methods. It is followed by a discussion, where strengths and limitations of early, mid, late and asymmetric fusion are discussed. A summary of the methods grouped by fusion approach is introduced in Table 1. This section follows a similar structure to Section 2.3, starting with early fusion and finalizing it with asymmetric fusion methods.

3.2.1 Methods for Early Fusion

The *Object-based Inverse Projection Algorithm (OIPA*) [18] is a method that can handle fusion of camera images with either 3D or 2D point cloud, and it uses algorithmic approach to achieve this without deep learning-based methods. It consists of two different parts: calibration and segmentation. During the calibration part, this method uses algorithms like Line Segment Detection (LSD) [19] to extract lines and ellipse-shaped features (road signs) from the 2D image. From the 2D LiDAR point cloud projection, the location of edges of the detected features are estimated with geometric calculations. As a final step, a projection matrix is created to encode the alignment of camera and LiDAR. In the segmentation part, objects are first detected with bounding boxes from 2D image using the object detection method YOLO-v10 [20]. This detection result is then inversely projected back into the LiDAR point cloud to corresponding 3D regions. Bounding boxes are used to focus on segments of the cloud where objects are located, and points outside the focused regions can be removed. This allows object-focused point cloud segmentation.

XYZDIRGB [21] is a fusion method that is built on top of the SqueezeSeg [22], a CNN-based method used for point cloud segmentation. In this method, the LiDAR point cloud is converted into a polar grid map, which is a tensor representation of the cloud. In the tensor each row corresponds to a vertical LiDAR layer, each column is a horizontal angular step over a 360-degree field of view and the third dimension contains depth and reflectance values. The third-dimension values are concatenated to RGB values from the camera image during the fusion process. After concatenation this tensor is fed to SqueezeSeg which performs the feature extraction and segmentation.

Table 1: Sensor Fusion Methods

	Method Name	Year	Architecture	Output Domain	Distinctive Feature
Early Fusion	OIPA [18]	*2025*	Non-DL	2D/3D	Inverse projection of bounding boxes
	UNRLF [23]	*2023*	CNN	2D	Used for road segmentation
	XYZDIRGB [21]	*2019*	CNN	3D	Uses polar grid mapping
Mid-Fusion	CLFusion [13]	*2024*	CNN + ViT	3D	Self-supervised training
	MFSA-Net [24]	*2024*	CNN + Attention	3D	Dual-distance attention feature aggregation
	CLFT [25]	*2024*	ViT	2D	First open-source transformer-based method for camera + LiDAR
	CMX [26]	*2023*	ViT	2D	Cross-modal feature rectification and can be used with various sensor configurations
	PMF [27]	*2021*	CNN	3D	Perspective projection
	FuseSeg [28]	*2019*	CNN	3D	Feature warping fusion
	XYZDI+DIRGB [21]	*2019*	CNN	3D	Uses polar grid mapping
Late Fusion	PCR6+RF [29]	*2024*	CNN	2D	Designed for multimodality
	EDF [29]	*2024*	CNN	2D	Shannon entropy for decision-making
	SSCLF [30]	*2021*	CNN	3D	Semi-supervised learning
Asymmetric Fusion	PFN [31]	*2024*	CNN	3D	First fusion method for 3D panoptic segmentation
	LIF-Seg [32]	*2024*	CNN	3D	Coarse, offset and refinement structure
	CMDFusion [33]	*2024*	CNN	3D	2D to 3D and 3D to 2D (bidirectional) fusion scheme for 3D feature enhancement
	MPFN [34]	*2023*	CNN + Attention	3D	Weakly supervised training for pixel-wise labels

U-Net-based RGB and LiDAR Fusion (UNRLF) [23] is a method used for road segmentation. Three variations of the same CNN-based method were created, and early fusion was found to be the most effective. Depth information from LiDAR is concatenated with RGB values of image pixels, and this 4-channel input is processed through the U-Net [35].

3.2.2 Methods for Mid-Fusion

XYZDI+DIRGB [21] is a mid-fusion version of the method XYZDIRGB discussed earlier in this review, and they were introduced in the same paper. In this method, features are extracted from both sensors in their own encoders, the resulting feature maps are concatenated in the third dimension of the tensor and segmentation happens in a shared decoder. This method was considered to be a worse option to the early fusion method, since it is computationally more complex and offers only a slightly improved accuracy.

Perception-aware Multi-sensor Fusion (PMF) [27] is a fusion method, where a 3DPC is projected into a camera coordinate system and features are extracted from modalities individually using a two-stream network. These streams are connected through residual-based fusion networks that are used for fusing features from both modalities together.

Camera-LiDAR Fusion Transformer (CLFT) [25] is a ViT-based fusion method designed for semantic segmentation applications in autonomous driving. It works by processing camera and LiDAR inputs separately in the ViT encoder stage, and features are fused using a cross-fusion strategy. This means that they are not fused at a single point, and the fusion is an ongoing process during the whole decoder stage.

Cross-Modal Fusion for RGB-X (CMX) [26] is a transformer-based fusion method, where RGB-X refers to the fact that this method can be used for fusing camera RGB data to various other modalities besides LiDAR. It works through a similar two-stream network used in PMF [27]. The difference in the CMX comes from the ability of features to rectify each other using Cross-Modal Feature Rectification units if one of the modalities is providing noisy information. After feature rectification, same level features are fused, and a similar structure continues throughout the network until the decoder.

FuseSeg [36] is a fusion method that is built as an extension to SqueezeSeg [22], the same method that was used for point cloud segmentation in the XYZDIRGB [21]. SqueezeSeg obtains information about reflectance, range and three-dimensional coordinates by using spherical projection. The camera image is processed through a CNN, features are extracted in

multiple layers, and they are concatenated with LiDAR representation using point correspondences between the two.

CLFusion [13] is a two-stream encoder-decoder method used for point cloud segmentation. It consists of 4 different parts: camera, LiDAR, fusion and supervision. Camera and LiDAR are processed by separate CNNs, in which features are gradually fused in the encoder and each layer uses skip connections to the corresponding decoder layers. Following the fusion of corresponding layers in the camera and LiDAR pipelines, these fused features are fed into the Swin Transformer [37] to obtain attention-based features. Then convolutional and attention-based features are injected back into the camera and LiDAR networks using sensor-specific weights. Finally, the combined features are upsampled in the decoder to produce the output. A visual representation of the CLFusion architecture is shown in Figure 7. CLFusion can be trained self-supervised on unlabeled or partially labeled datasets using a pretrained network for creating pseudo-labels. The network used is a modified version PIDNet [38], and it works by performing semantic segmentation in real-time with a layer that assigns confidence values to predictions. These confidence-based calculations are then used for filtering, allowing only high confidence labels to end up in the training set.

MFSA-Net [24] is a mid-fusion method that contains three different modules: DDSA3D, CATSR and perception-aware loss module. The method projects 3DPC to 2-dimensional space, and the nearest neighbors to each point are located using DDSA3D. The same method is used for utilizing feature- and Euclidian distances to gain information about local contextual feature information among the points. The extracted features are injected into the LiDAR stream in a two-stream network that is based on cross-attention. After both modalities have been processed through the network, the method uses a perceptual-aware loss module to calculate confidence values for each branch in the network, considering that camera pixels in the middle versus the edges of an object have a large difference in semantic relevance.

3.2.3 Methods for Late Fusion

PCR6+ Rule-based fusion (PCR6+RF) [29] uses 2 identical CNN-based encoder-decoder architectures for segmentation to produce softmax probability distributions for both camera and LiDAR inputs. Both softmax outputs are converted to basic belief assignments (BBA), which are a more flexible alternative to traditional probabilities. They allow belief to be assigned to single outcomes or combinations of outcomes. They are then merged using the PCR6+ fusion rule, which is designed to handle conflicting information from multiple sources. It identifies

areas where the methods disagree and shares the uncertainty depending on how confident each method was in its prediction. The result is a belief distribution for each pixel, and classes with the highest belief can be selected as the final prediction for that pixel.

Entropy-weighted decision fusion (EDF) [29] uses similar CNN-based encoder-decoder architectures as PCR6+ method for generating softmax probability distributions on camera and LiDAR inputs. Instead of converting outputs to belief assignments like PCR6+ Rule-based fusion [29], it makes early decisions based on the class with the highest softmax value for each input source. Each decision is then assigned a confidence weight calculated using Shannon entropy (lower entropy indicates higher confidence). The final class for each pixel is determined by averaging the two decisions, weighted by their respective entropies.

Semi-Supervised LiDAR-Camera Fusion (SSLCF) [30] is a fusion method based on the fully convolutional network FCN-ResNet50 [39]. The method contains 3 separate networks: camera, LiDAR and fusion. Features are extracted from the camera and LiDAR branch during the first 4 stages of the FCN, and then they are concatenated in the fusion network, which will proceed with the final prediction. This method takes advantage of semi-supervised learning by training the fusion network branch on labeled data and using the trained network to create predictions or *proxy labels* for unlabeled data. These labels are then considered as the ground truth when training the single modality networks, reducing the need for manual annotation.

3.2.4 Methods for Asymmetric Fusion

Panoptic FusionNet (PFN) [31] is a fusion method used for panoptic segmentation. It means that rather than focusing solely on semantic segmentation, the method can also detect individual instances of an object from the same class (instance segmentation) and create a combined result. It works by first processing camera and LiDAR in separate networks and extracting features from both sensors. Voxel features from LiDAR and 2D features from the camera are fused using a correspondence table, which associates corresponding sections from each sensor together. After fusion, the method has 3 parts: semantic head, instance head and a panoptic processing module. Semantic segmentation happens in the semantic head by concatenating voxel-wise global features with point-wise features and passing them through multiple MLPs.

Multi-Phase Fusion Network (MPFN) [34] is a fusion method that combines mid and late fusion. In this method, two separate networks are used to process LiDAR and camera data, the one used for LiDAR being the main network and the camera providing complementary

information. Before injecting the extracted image features to the point cloud, this method uses an attention-based feature fusion module to filter off irrelevant features. Finally, late fusion happens to both branches, where each pixel is given a confidence value and merged, resulting in the final output. The specialty of this method is that it takes advantage of weak supervision to address the lack of pixel-wise labels in the datasets used for training these networks. This is achieved through projecting LiDAR labels into the image space and using weak supervision to generate more labels.

LIF-Seg [32] is an asymmetric fusion method used for point cloud segmentation. It is a coarse-to-fine framework that involves 3 stages called coarse feature extraction, offset learning and refinement. In the first stage, LiDAR points are projected onto camera images and concatenated with the corresponding image context information. A LiDAR segmentation network is then used, leading to coarse features. The second stage takes care of alignment by first segmenting the camera image on a separate segmentation network and in the end projecting the segmentation result back into 3D space where it is concatenated with the coarse features from the previous stage. In the last stage, the fusion result from the second stage is fed into a similar segmentation network that was used in the first stage, creating the final prediction.

CMDFusion [33] is a bidirectional fusion method designed for 3D semantic segmentation. Bidirectional refers to the fact that this method uses a Bidirectional Fusion Block (BFB) to benefit from projecting a 2D knowledge branch into a 3DPC and vice versa. This way, features in the three-dimensional space can be enhanced directly and indirectly. The 2D-to-3D projection is used for the main 3D feature extraction (direct) and features from 3D-to-2D projection are used to enhance the main features (indirect). It also introduces a Cross-Modality Distillation (CMD) that allows the LiDAR network to be trained in a way that it can remember information from the camera network (2D knowledge branch). When camera images are not available in a certain direction, CMD can use the 3DPC to generate the information.

4 Discussion

In this section, we will discuss approaches to sensor fusion and potential future directions where the field is heading. Table 2 summarizes the advantages and disadvantages of sensor fusion approaches. The trends in the field that we will discuss, are mainly trying to address the limitations of current approaches.

4.1 Limitations of Existing Approaches

When carrying out the research for this literature review, it became obvious that mid-fusion is a popular approach in the current fusion methods. In the methods designed for complex scenarios, mid-fusion or a hybrid version of this approach appears to be a valid option. Mid-fusion offers the perfect balance, it is not as sensitive to misalignment as early fusion and still takes advantage of information available by the multiple modalities [3].

Table 2: Fusion Approaches Summary

	Advantages	**Disadvantages**
Early Fusion	Fusion happens on raw data, leveraging all the available information Requirements for memory and computing power are low since modalities are processed simultaneously	Changes to sensor configuration are difficult to execute, because it requires retraining the network Prone to data misalignment that can be caused by faulty calibration, sensor malfunction and sampling rate mismatch
Mid-Fusion	Can have a better perceptual understanding, because fusion happens at the feature-level, highly flexible	Choosing the optimal time to fuse is difficult
Late Fusion	Configuration can be changed easily, because every sensor is trained individually	High costs in computation and memory Does not take full advantage of synergy between sensors

The majority of the more complex methods included in the literature review have found unique ways of aligning multimodal data to minimize information losses and errors in the segmentation result. Still, sensor misalignment is a big issue, and it is difficult to do it perfectly. CNN and ViT hybrid mehtods were popular because the former is better at small details, while the latter captures larger features.

The methods require a lot of data for training, and many of the methods in the literature review have already implemented ways to address this problem, like CLFusion [13], that implements self-supervised learning, or SSCLF [30] and MPFN [34], that have developed their own ways of using weakly supervised learning. There is still a lot of work to be done, and the more labeled datasets for training will lead to better overall segmentation performance.

4.2 Trends and Potential Future Directions

There has been a clear trajectory in semantic segmentation from traditional machine learning methods to CNNs and recently the ViT. CNNs gained popularity due to automatic feature extraction, and now ViTs have become interesting because of their ability to understand the features globally. Driven by high success rate and strong performance of the ViTs, they are one of the main research focuses for visual tasks [15]. The downside of ViTs is that they require large, labeled datasets for training and especially in semantic segmentation where the focus is on pixel-wise predictions. A future direction in the field is to investigate ways of how to overcome this, and there are several potential solutions.

Weakly supervised and self-supervised learning methods are one of the promising solutions to the dataset problem [9], [40]. Weakly supervised refers to datasets that are only partially labeled or contain noisy data, while self-supervised means that the network can work on the data even when the data is completely unlabeled [1]. These methods are still in the development phase and have not yet been able to surpass fully supervised methods [40], but as seen in the literature review, methods like [13], [30], [34] are starting to implement these methods.

Another potential future direction to address the challenge of the methods requiring a large amount of labeled data is open vocabulary segmentation, which leverages visual language models (VLMs). CLIP-based [41] VLMs can perform segmentation using natural language and visual feature pairs without the need for pixel-wise labels. Examples of methods already doing this include LSeg [42], ZegFormer [43] and MaskCLIP [44]. There are examples of visual language models being used to create pseudo-labels for data that is used to train other

segmentation methods in a weakly supervised way (MaskCLIP+ [45], OVSeg [44]), similar to what CLFusion [13] proposed. The future potential of visual language models has a lot to offer.

Finally, researchers have begun exploring state space models (SSMs) for visuals tasks as an alternative to the computationally challenging ViT. Implementing the Mamba architecture for vision [46] has shown promising results, by offering notable improvements in terms of efficiency and segmentation accuracy, while still capturing long-range dependencies in the data. However, this is a very recent development, gaining momentum only after the release of the original Mamba architecture [47], and research into its application on visual tasks is still in its early stages.

4.3 Conclusion

This thesis provided an overview of sensor fusion approaches (early, mid, late and asymmetric fusion) that could be used in autonomous driving systems equipped with cameras and LiDAR for semantic segmentation. The focus was on gathering new deep learning -based methods and organizing them by fusion approach, showcasing the working principles of each method. During the review, it was found that although ViTs have gained a substantial amount of research interest and are a key research focus in the field, many methods still depend on CNN-based fusion methods. Hybrid architectures combining the CNN and ViT were common, due to CNNs still being the dominant method for detecting smaller local features, while ViTs excel at capturing long-range dependencies. In the future, open vocabulary segmentation [48], weakly supervised and self-supervised methods have a lot of potential to drive this highly data dependent field forward.

5 References

[1] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image Segmentation Using Deep Learning: A Survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–1, 2021, doi: 10.1109/TPAMI.2021.3059968.

[2] D. Feng *et al.*, "Deep Multi-Modal Object Detection and Semantic Segmentation for Autonomous Driving: Datasets, Methods, and Challenges," *IEEE Trans. Intell. Transport. Syst.*, vol. 22, no. 3, pp. 1341–1360, Mar. 2021, doi: 10.1109/TITS.2020.2972974.

[3] C. Xiang *et al.*, "Multi-Sensor Fusion and Cooperative Perception for Autonomous Driving: A Review," *IEEE Intell. Transport. Syst. Mag.*, vol. 15, no. 5, pp. 36–58, Sep. 2023, doi: 10.1109/MITS.2023.3283864.

[4] K. Huang, B. Shi, X. Li, X. Li, S. Huang, and Y. Li, "Multi-modal Sensor Fusion for Auto Driving Perception: A Survey," Dec. 16, 2024, *arXiv*: arXiv:2202.02703. doi: 10.48550/arXiv.2202.02703.

[5] Y. Guo, G. Nie, W. Gao, and M. Liao, "2D Semantic Segmentation: Recent Developments and Future Directions," *Future Internet*, vol. 15, no. 6, p. 205, Jun. 2023, doi: 10.3390/fi15060205.

[6] M. Cordts *et al.*, "The Cityscapes Dataset for Semantic Urban Scene Understanding," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 3213–3223. doi: 10.1109/CVPR.2016.350.

[7] Z. Xiao *et al.*, "Research Advances in Deep Learning for Image Semantic Segmentation Techniques," *IEEE Access*, vol. 12, pp. 175715–175741, 2024, doi: 10.1109/ACCESS.2024.3496723.

[8] B. Marsh, A. H. Sadka, and H. Bahai, "A Critical Review of Deep Learning-Based Multi-Sensor Fusion Techniques," *Sensors*, vol. 22, no. 23, p. 9364, Dec. 2022, doi: 10.3390/s22239364.

[9] G. Rizzoli, F. Barbato, and P. Zanuttigh, "Multimodal Semantic Segmentation in Autonomous Driving: A Review of Current Approaches and Future Perspectives," *Technologies*, vol. 10, no. 4, p. 90, Jul. 2022, doi: 10.3390/technologies10040090.

[10] N. Aherwadi, U. Mittal, J. Singla, N. Z. Jhanjhi, A. Yassine, and M. S. Hossain, "Prediction of Fruit Maturity, Quality, and Its Life Using Deep Learning Algorithms," *Electronics*, vol. 11, no. 24, p. 4100, Dec. 2022, doi: 10.3390/electronics11244100.

[11] V. Badrinarayanan, A. Kendall, and R. Cipolla, “SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017, doi: 10.1109/TPAMI.2016.2644615.

[12] Y. Duan, W. Zhang, P. Huang, G. He, and H. Guo, “A New Lightweight Convolutional Neural Network for Multi-Scale Land Surface Water Extraction from GaoFen-1D Satellite Images,” *Remote Sensing*, vol. 13, no. 22, p. 4576, Nov. 2021, doi: 10.3390/rs13224576.

[13] T. Wang, R. Song, Z. Xiao, B. Yan, H. Qin, and D. He, “CLFusion:3D Semantic Segmentation Based on Camera and Lidar Fusion,” in *2024 IEEE International Symposium on Circuits and Systems (ISCAS)*, Singapore, Singapore: IEEE, May 2024, pp. 1–5. doi: 10.1109/ISCAS58744.2024.10558356.

[14] K. Han *et al.*, “A Survey on Visual Transformer,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 87–110, Jan. 2023, doi: 10.1109/TPAMI.2022.3152247.

[15] H. Thisanke, C. Deshan, K. Chamith, S. Seneviratne, R. Vidanaarachchi, and D. Herath, “Semantic segmentation using Vision Transformers: A survey,” *Engineering Applications of Artificial Intelligence*, vol. 126, p. 106669, Nov. 2023, doi: 10.1016/j.engappai.2023.106669.

[16] A. Dosovitskiy *et al.*, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale,” Jun. 03, 2021, *arXiv*: arXiv:2010.11929. doi: 10.48550/arXiv.2010.11929.

[17] J.-H. Bang *et al.*, “CA-CMT: Coordinate Attention for Optimizing CMT Networks,” *IEEE Access*, vol. 11, pp. 76691–76702, 2023, doi: 10.1109/ACCESS.2023.3297206.

[18] X. Yuan, S. Wang, Y. Xie, S. Q. Xie, C. Wang, and T. Xiong, “Object-Based Semantic Fusion Algorithm of Lidar and Camera via Inverse Projection,” *IEEE Trans. Instrum. Meas.*, pp. 1–1, 2025, doi: 10.1109/TIM.2025.3548241.

[19] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, “LSD: A Fast Line Segment Detector with a False Detection Control,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 4, pp. 722–732, Apr. 2010, doi: 10.1109/TPAMI.2008.300.

[20] A. Wang *et al.*, “YOLOv10: Real-Time End-to-End Object Detection,” Oct. 30, 2024, *arXiv*: arXiv:2405.14458. doi: 10.48550/arXiv.2405.14458.

[21] K. E. Madawy, H. Rashed, A. E. Sallab, O. Nasr, H. Kamel, and S. Yogamani, “RGB and LiDAR fusion based 3D Semantic Segmentation for Autonomous Driving,” Jul. 17, 2019, *arXiv*: arXiv:1906.00208. doi: 10.48550/arXiv.1906.00208.

[22] B. Wu, A. Wan, X. Yue, and K. Keutzer, “SqueezeSeg: Convolutional Neural Nets with Recurrent CRF for Real-Time Road-Object Segmentation from 3D LiDAR Point Cloud,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, QLD: IEEE, May 2018, pp. 1887–1893. doi: 10.1109/ICRA.2018.8462926.

[23] A. T. Candan and H. Kalkan, “U-Net-based RGB and LiDAR image fusion for road segmentation,” *SIViP*, vol. 17, no. 6, pp. 2837–2843, Sep. 2023, doi: 10.1007/s11760-023-02502-5.

[24] Y. Duan *et al.*, “MFSA-Net: Semantic Segmentation With Camera-LiDAR Cross-Attention Fusion Based on Fast Neighbor Feature Aggregation,” *IEEE J. Sel. Top. Appl. Earth Observations Remote Sensing*, vol. 17, pp. 19627–19639, 2024, doi: 10.1109/JSTARS.2024.3472751.

[25] J. Gu, M. Bellone, T. Pivoňka, and R. Sell, “CLFT: Camera-LiDAR Fusion Transformer for Semantic Segmentation in Autonomous Driving,” *IEEE Trans. Intell. Veh.*, pp. 1–12, 2024, doi: 10.1109/TIV.2024.3454971.

[26] J. Zhang, H. Liu, K. Yang, X. Hu, R. Liu, and R. Stiefelhagen, “CMX: Cross-Modal Fusion for RGB-X Semantic Segmentation With Transformers,” *IEEE Trans. Intell. Transport. Syst.*, vol. 24, no. 12, pp. 14679–14694, Dec. 2023, doi: 10.1109/TITS.2023.3300537.

[27] Z. Zhuang, R. Li, K. Jia, Q. Wang, Y. Li, and M. Tan, “Perception-Aware Multi-Sensor Fusion for 3D LiDAR Semantic Segmentation,” in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada: IEEE, Oct. 2021, pp. 16260–16270. doi: 10.1109/ICCV48922.2021.01597.

[28] G. Krispel, M. Opitz, G. Waltner, H. Possegger, and H. Bischof, “FuseSeg: LiDAR Point Cloud Segmentation Fusing Multi-Modal Data,” Dec. 19, 2019, *arXiv*: arXiv:1912.08487. doi: 10.48550/arXiv.1912.08487.

[29] D.-V. Giurgi, J. Dezert, T. Josso-Laurain, M. Devanne, and J.-P. Lauffenburger, “Fusion of Semantic Segmentation Models for Vehicle Perception Tasks,” in *2024 27th International Conference on Information Fusion (FUSION)*, Venice, Italy: IEEE, Jul. 2024, pp. 1–8. doi: 10.23919/FUSION59988.2024.10706336.

[30] L. Caltagirone, M. Bellone, L. Svensson, M. Wahde, and R. Sell, “Lidar–Camera Semi-Supervised Learning for Semantic Segmentation,” *Sensors*, vol. 21, no. 14, p. 4813, Jul. 2021, doi: 10.3390/s21144813.

[31] H. Song, J. Cho, J. Ha, J. Park, and K. Jo, “Panoptic-FusionNet: Camera-LiDAR fusion-based point cloud panoptic segmentation for autonomous driving,” *Expert*

Systems with Applications, vol. 251, p. 123950, Oct. 2024, doi: 10.1016/j.eswa.2024.123950.

[32] L. Zhao, H. Zhou, X. Zhu, X. Song, H. Li, and W. Tao, "LIF-Seg: LiDAR and Camera Image Fusion for 3D LiDAR Semantic Segmentation," *IEEE Trans. Multimedia*, vol. 26, pp. 1158–1168, 2024, doi: 10.1109/TMM.2023.3277281.

[33] J. Cen *et al.*, "CMDFusion: Bidirectional Fusion Network With Cross-Modality Knowledge Distillation for LiDAR Semantic Segmentation," *IEEE Robot. Autom. Lett.*, vol. 9, no. 1, pp. 771–778, Jan. 2024, doi: 10.1109/LRA.2023.3335771.

[34] X. Chang, H. Pan, W. Sun, and H. Gao, "A Multi-Phase Camera-LiDAR Fusion Network for 3D Semantic Segmentation With Weak Supervision," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 8, pp. 3737–3746, Aug. 2023, doi: 10.1109/TCSVT.2023.3241641.

[35] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," May 18, 2015, *arXiv*: arXiv:1505.04597. doi: 10.48550/arXiv.1505.04597.

[36] G. Krispel, M. Opitz, G. Waltner, H. Possegger, and H. Bischof, "FuseSeg: LiDAR Point Cloud Segmentation Fusing Multi-Modal Data," in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Snowmass Village, CO, USA: IEEE, Mar. 2020, pp. 1863–1872. doi: 10.1109/WACV45572.2020.9093584.

[37] Z. Liu *et al.*, "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," Aug. 17, 2021, *arXiv*: arXiv:2103.14030. doi: 10.48550/arXiv.2103.14030.

[38] J. Xu, Z. Xiong, and S. P. Bhattacharyya, "PIDNet: A Real-time Semantic Segmentation Network Inspired by PID Controllers," Apr. 07, 2023, *arXiv*: arXiv:2206.02066. doi: 10.48550/arXiv.2206.02066.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[40] Y. Mo, Y. Wu, X. Yang, F. Liu, and Y. Liao, "Review the state-of-the-art technologies of semantic segmentation based on deep learning," *Neurocomputing*, vol. 493, pp. 626–646, Jul. 2022, doi: 10.1016/j.neucom.2022.01.005.

[41] A. Radford *et al.*, "Learning Transferable Visual Models From Natural Language Supervision".

[42] B. Li, K. Q. Weinberger, S. Belongie, V. Koltun, and R. Ranftl, “Language-driven Semantic Segmentation,” Apr. 03, 2022, *arXiv*: arXiv:2201.03546. doi: 10.48550/arXiv.2201.03546.

[43] J. Ding, N. Xue, G.-S. Xia, and D. Dai, “Decoupling Zero-Shot Semantic Segmentation,” Apr. 15, 2022, *arXiv*: arXiv:2112.07910. doi: 10.48550/arXiv.2112.07910.

[44] Z. Ding, J. Wang, and Z. Tu, “Open-Vocabulary Universal Image Segmentation with MaskCLIP,” Jun. 08, 2023, *arXiv*: arXiv:2208.08984. doi: 10.48550/arXiv.2208.08984.

[45] C. Zhou, C. C. Loy, and B. Dai, “Extract Free Dense Labels from CLIP,” Jul. 27, 2022, *arXiv*: arXiv:2112.01071. doi: 10.48550/arXiv.2112.01071.

[46] H. Zhang *et al.*, “A Survey on Visual Mamba,” *Applied Sciences*, vol. 14, no. 13, p. 5683, Jun. 2024, doi: 10.3390/app14135683.

[47] A. Gu and T. Dao, “Mamba: Linear-Time Sequence Modeling with Selective State Spaces,” May 31, 2024, *arXiv*: arXiv:2312.00752. doi: 10.48550/arXiv.2312.00752.

[48] C. Zhu and L. Chen, “A Survey on Open-Vocabulary Detection and Segmentation: Past, Present, and Future,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 12, pp. 8954–8975, Dec. 2024, doi: 10.1109/TPAMI.2024.3413013.