Integrated Computer-Aided Engineering 29 (2022) 241–258
DOI 10.3233/ICA-220681
IOS Press

Object detection using depth completion and camera-LiDAR fusion for autonomous driving

Manuel Carranza-García*, F. Javier Galán-Sales, José María Luna-Romera and José C. Riquelme
Division of Computer Science, University of Sevilla, Sevilla, Spain

Abstract. Autonomous vehicles are equipped with complimentary sensors to perceive the environment accurately. Deep learning models have proven to be the most effective approach for computer vision problems. Therefore, in autonomous driving, it is essential to design reliable networks to fuse data from different sensors. In this work, we develop a novel data fusion architecture using camera and LiDAR data for object detection in autonomous driving. Given the sparsity of LiDAR data, developing multi-modal fusion models is a challenging task. Our proposal integrates an efficient LiDAR sparse-to-dense completion network into the pipeline of object detection models, achieving a more robust performance at different times of the day. The Waymo Open Dataset has been used for the experimental study, which is the most diverse detection benchmark in terms of weather and lighting conditions. The depth completion network is trained with the KITTI depth dataset, and transfer learning is used to obtain dense maps on Waymo. With the enhanced LiDAR data and the camera images, we explore early and middle fusion approaches using popular object detection models. The proposed data fusion network provides a significant improvement compared to single-modal detection at all times of the day, and outperforms previous approaches that upsample depth maps with classical image processing algorithms. Our multi-modal and multi-source approach achieves a 1.5, 7.5, and 2.1 mean AP increase at day, night, and dawn/dusk, respectively, using four different object detection meta-architectures.

Keywords: Autonomous driving, data fusion, deep learning, object detection, transfer learning

1. Introduction

Autonomous driving is attracting growing attention due to its potential to revolutionize mobility, improve road safety, and reduce environmental pollution. Although significant progress in computer vision has been achieved over the last few years [1–3], developing reliable perception systems for autonomous vehicles remains challenging [4–6]. Among the problems to be addressed, object detection is an essential perception task that has received considerable interest in the literature. Recently, many self-driving car companies such as Waymo Open Dataset [7], nuScenes [8], or PandaSet [9] have publicly released high-quality detection datasets. This increase in the amount and quality of data has allowed researchers to push the state-of-the-art in this field, with deep learning-based models as the main approach. However, the area of data fusion for object detection has not yet been explored in depth. What, how, and when to fuse remains an important challenge to be studied [10].

Self-driving vehicles need to be accurate and robust enough to operate safely in complex scenarios, such as mixed urban traffic or adverse weather conditions. Therefore, these modern vehicles are equipped with multiple sensors to better perceive the environment, such as red-green-blue (RGB) cameras, light detection and ranging (LiDAR), or radar. Since each type of sensor has its own limitations, multiple sensing modalities can be fused to exploit their complementary properties. For instance, RGB cameras provide high-resolution semantic information but cannot provide optimal perfor-

*Corresponding author: Manuel Carranza-García, Division of Computer Science, University of Sevilla, Sevilla, Spain. E-mail: mcarranzag@us.es.

ISSN 1069-2509/$35.00

mance with poor illumination at night. LiDAR devices have been used in several areas of knowledge, such as work safety [11], health [12,13] or construction [14]. Their depth sensors can provide useful geometric information even at night, but they generate low-resolution sparse point clouds [15]. However, how to efficiently combine multi-sensor data to leverage their advantages for object detection remains an open question. Furthermore, the real-time requirements of this context increase the complexity of multi-modal approaches, which have received considerably less attention than uni-modal proposals. Therefore, there is a need for more studies on data fusion neural networks to increase the reliability of real-time object detection for self-driving vehicles.

As supported by previous studies in the literature, LiDAR information is essential to detect objects in situations of low illumination, such as nighttime [15]. Therefore, in this work, our aim is to design a better RGB and LiDAR fusion network to enhance the performance of the 2D object detection task for autonomous driving. We propose a novel data fusion method, investigating how to efficiently integrate the depth information into existing deep learning architectures that have been traditionally used for RGB data. For the experimental study, the Waymo dataset is used, which is the most extensive and diverse self-driving dataset in terms of geographic coverage and weather conditions. This dataset provides a unique opportunity to explore the performance of multi-modal detection under different illumination situations. To the best of our knowledge, this work is the first that studies RGB-D 2D object detection using the large-scale Waymo dataset. The presented study can be divided into two fundamental parts, the LiDAR depth completion method and the data fusion network for 2D object detection.

The sparse nature of LiDAR data presents several challenges since their projection into the 2D space leaves many pixels without information. This work proposes a deep learning-based depth completion model, combining sparse depth maps with camera images to obtain dense depth maps in a supervised manner. This encoder-decoder depth completion network is fast, lightweight, and can be integrated as a module into the pipeline of traditional 2D detection architectures. The network is trained using the KITTI dataset [16], which is the only existing autonomous driving dataset with labeled dense depth maps. Then, we use transfer learning to directly infer the Waymo dataset's depth maps with no additional supervision. The experimental study compares our proposal to the usual approach of related studies, which is to upsample the depth maps using classical image processing algorithms such as bilateral filtering [17]. These algorithms only use sparse projections for the depth completion task and often fail to preserve the structure of objects due to the lack of semantic cues. In contrast, our depth completion network uses RGB guidance to better reconstruct the scene.

With the obtained dense LiDAR depth maps, we perform a thorough study of data fusion at different stages using popular detection networks such as Faster R-CNN [18], ATSS [19], RetinaNet [20], or YOLO [21]. We explore early and mid-fusion approaches and compare the dual-modal and uni-modal detection performance. The diversity of the Waymo dataset allows for examining the influence of combining RGB images with accurate depth information under three different lighting conditions: day, night, and dawn/dusk. The analysis also includes the proposed models' efficiency (computation time and memory cost), given that real-time performance is essential in this context. This multi-source and multi-modal study is expected to contribute to our understanding of how to effectively fuse depth with RGB camera data with a small computational overhead for 2D object detection.

In summary, the main contributions of this work are the following:

- An efficient LiDAR sparse-to-dense completion network that can be plugged into existing object detectors to increase the robustness of detection under all lighting conditions.
- A novel data fusion (RGB+LiDAR) object detection architecture that significantly improves the performance over the large-scale Waymo Open Dataset by using transfer learning from KITTI for LiDAR depth completion.
- A thorough analysis of the influence of the quality of LiDAR depth maps on the object detection downstream task with different illumination conditions (day, night, dawn/dusk).

The rest of the paper is organized as follows: Section 2 reviews relevant related work; Section 3 describes the materials and methods proposed in the study; Section 4 presents the results obtained from the experimental study and discusses the main findings; Section 5 presents the conclusions and potential future work.

2. Related work

2.1. Deep learning for 2D object detection

Object detection recognizes and localizes objects belonging to different classes in an image. As for many

computer vision tasks, deep learning models are currently state-of-the-art for this task. There are two main approaches in terms of 2D object detection networks: two-stage, such as Faster R-CNN [18], or one-stage models, such as RetinaNet [20] or YOLO [21].

In general, two-stage architectures run slower but achieve higher accuracy than one-stage models. Two-stage architectures divide the process into a category-agnostic region proposal stage and the classification and refinement stage. In contrast, one-stage detectors directly infer category-specific box candidates. Both approaches share a common structure: a convolutional backbone as a feature extractor, such as ResNet [22], and a dual prediction head for regression and classification. Commonly, these models use Feature Pyramid Networks (FPN) in the feature extractor to detect objects at multiple scales [23].

The mentioned models rely on predefined reference boxes, known as anchors or fiducial points, to generate detections. However, other approaches have proposed anchor-free architectures. For instance, FCOS [24] and CenterNet [25] directly regress the bounding box from key points, such as the object's center. Another recent trend in the literature is to combine convolutional feature extractors with attention-based layers, as seen in Transformer networks [26].

In the object detection field, novel proposals are usually validated using the general-purpose COCO benchmark [27], and often ignore the computational efficiency of the models. Recent performance improvements have been achieved at the cost of introducing significantly more computational overhead. For example, using more advanced feature extractors such as ResNeXt [28], cascading detectors as in Cascade R-CNN [29], or with ensembles [30]. However, the autonomous driving field has specific requirements in terms of high latency and limited computational resources. Therefore, top-performing models in COCO may not be suitable for this real-time application.

Nevertheless, with the increase in data, there have been more studies on camera-only object detection for autonomous driving. In a previous paper, we performed an experimental review of state-of-the-art detectors over the Waymo Open Dataset, analyzing the trade-off between accuracy and speed [31]. Several works have proposed anchor optimization methodologies considering the perspective of the vehicle's cameras [32,33]. There have also been efforts to improve vehicle detection in adverse weather with visibility enhancement techniques [34]. Other studies have focused on specifically improving the detection of pedestrians [35], small objects [36], or traffic signs [37].

2.2. LiDAR depth completion

Depth completion is the problem of inferring a dense depth map of a 3D scene given an image and a sparse depth map from sensors, such as LiDAR. To date, research on depth completion in the autonomous driving field has focused primarily on KITTI, which is the only labeled outdoor dataset for this task [16].

Existing studies can be classified into depth-only and multiple-input methods that include RGB features in the pipeline. Among depth-only proposals, some defend using classical image processing techniques, such as bilateral filtering, to solve this task [17]. In contrast, others introduce sparse invariant convolutions to upsample the raw LiDAR map [38]. Recent works have proposed light-weight networks using depthwise convolutions [39]. However, these methods are less accurate due to the sparse nature of the data and the lack of semantic cues [40].

Therefore, recent studies have developed more advanced deep learning models that combine both inputs and achieve superior performance. Several works have developed pseudo-depth completion networks guided by RGB features, fusing information at multiple stages [41,42]. Furthermore, given the lack of dense ground truth depth labels, a novel self-supervised training framework was presented in [43]. However, it achieves significantly lower precision than the supervised network. A more accurate approach proposed a convolutional network with depth-normal constraints and recurrent refinement stages for noise reduction [44]. More recently, Tang et al. developed a guided network to predict kernels for depth feature extraction [45]. Hu et al. designed a two-stage encoder-decoder network with geometric convolution, which is more efficient than previous models [46].

All these works are focused solely on the depth completion task using the KITTI dataset. These studies inspire our proposal, which is more focused on building a faster network that introduces a small overhead on the downstream object detection task, which is the problem addressed in this paper.

2.3. Data fusion for 2D object detection in autonomous driving

Given the limitations of individual sensors, multi-modal approaches for object detection have gained greater importance over the past years. A detailed review of existing multi-modal datasets for autonomous driving is provided in [10]. Premebida et al. carried out one of the first studies on the KITTI dataset using RGB and dense LiDAR fusion for pedestrian detection [47].

This work inspired several studies searching for optimal sensor data fusion architectures when using detectors originally designed for camera inputs [48,49].

There have also been efforts to develop efficient data fusion approaches with YOLO models, upsampling the sparse depth maps through bilateral filtering [50,51]. Other works have focused on evaluating the suitability of data fusion when simulating low-light environments [15] or adverse weather conditions [52]. In this case, the dense depth maps are obtained by simple interpolation, averaging values within a small neighborhood. More recent studies using the KITTI dataset have proposed combining bilateral filtering and trigonometric interpolation to upsample the depth channel before the fusion layer [53]. Furthermore, Geng et al. explored several dual-modal architectures for segmentation on KITTI, with dense depth maps also obtained using bilateral filtering [54].

The fact that Waymo provides synchronized LiDAR to camera projections has opened an interesting research area that a few studies have explored for vehicle [55] and pedestrian detection [56]. The first explores early fusion using an RGD approach in which the blue channel is replaced with sparse depth filled with zeros. Similarly, the latter concatenates RGB with raw sparse depth values, exploring early fusion on a YOLO-based detector with the Waymo dataset.

In all these studies, the sparse LiDAR depth maps are either kept raw or upsampled using morphological operations such as interpolation or bilateral filtering. In contrast, our proposal designs a supervised deep neural network to produce high-quality dense depth maps. The novelty of our approach is the introduction of an efficient sparse-to-dense depth completion network into the object detection pipeline. Furthermore, most 2D object detection studies in the sensor fusion area use KITTI. This dataset has a relatively low diversity since it only has recordings during daytime and sunny days. Our work studies object detection over the large-scale Waymo dataset, which has a much wider diversity in terms of weather and lighting conditions.

3. Materials and methods

This section first presents the autonomous driving datasets, followed by the LiDAR preprocessing method and the proposed data fusion architecture.

3.1. Autonomous driving datasets

This study uses two datasets for different purposes: the Waymo Open Dataset for the object detection task and the KITTI dataset for training the supervised LiDAR depth completion network.

3.1.1. Waymo 2D object detection dataset

The main goal of this study is to improve the performance of the 2D object detection task in self-driving vehicles. For this purpose, the Waymo Open Dataset has been selected [7]. This large-scale dataset contains more than a thousand driving scenes recorded across different urban areas (Phoenix, San Francisco, and Mountain View). It also includes recordings at different times of the day (day, night, and dawn/dusk) and weather conditions. This diversity of lighting situations is key for our study on the importance of including LiDAR depth information in the detection pipeline.

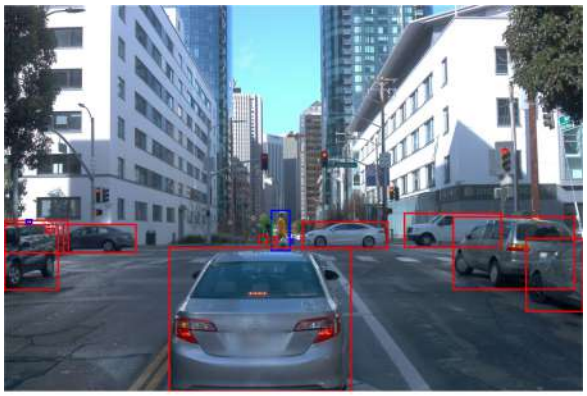
The detection task is a multi-class problem with three types of objects: vehicles, pedestrians, and cyclists. The vehicle is equipped with five cameras, three front with a resolution of 1920 $\times$ 1280 and two lateral of 1920 $\times$ 886. The images obtained from all cameras are considered a single dataset for evaluation purposes. Furthermore, this dataset offers synchronized LiDAR to camera projections, providing sparse depth maps that can be useful for detection under complex environmental conditions. Given the different perspectives of the cameras, the LiDAR projections are provided for each one of them. Figure 1 presents an example image of the dataset, including the sparse LiDAR projections. The mean of pixels with projected depth values in the dataset is less than 1%, which illustrates the sparsity of the LiDAR data.

The complete dataset contains over 1,150 driving scenes that capture synchronized LiDAR and camera data for 20 seconds, resulting in around 200 frames per scene. For computational reasons, we sample the dataset every ten frames. The training and validation division is provided directly by Waymo when downloading the dataset.

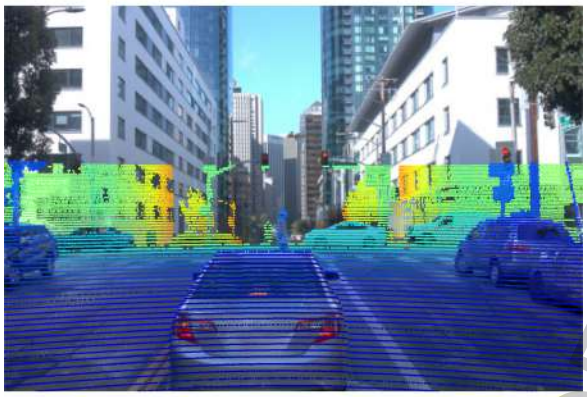
Of 99,190 images, about 75% are used for training and 25% for testing. Among the 950,080 labeled objects in the dataset, 78.07% are vehicles, 21.29% are pedestrians, and 0.64% are cyclists. As can be seen, there is a high imbalance between the number of vehicle instances and the other two classes, which is particularly severe for cyclists.

3.1.2. KITTI depth completion dataset

In order to obtain dense depth maps from sparse LiDAR projections with a supervised neural network, we need a fully labeled dataset to train a model for this task.

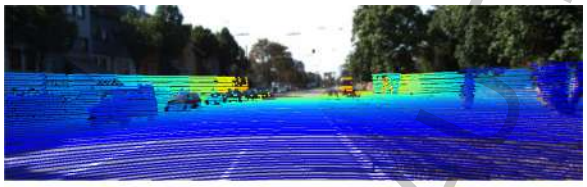


(a)

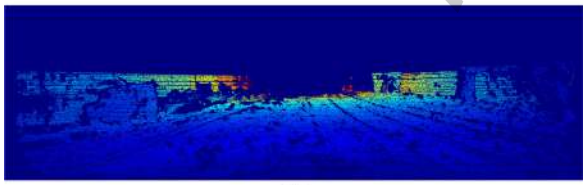


(b)

Fig. 1. Waymo 2D object detection dataset. (a) RGB camera image with object bounding boxes; (b) RGB camera image with liDAR sparse projections.



(a)



(b)

Fig. 2. KITTI depth completion dataset. (a) RGB camera image with LiDAR sparse projections; (b) Dense depth ground truth map.

KITTI is the only dataset with dense LiDAR depth annotated maps for autonomous driving [16]. It provides projections of 3D LiDAR points to the corresponding image frames, with about 5% of valid pixels. The ground truth dense maps have about 16% of pixels with depth values. The dataset contains over 90,000 images with a resolution of 1216 $\times$ 352. Figure 2 shows an example scene of this dataset, with the sparse projections and the dense depth map that has to be predicted.

3.2. LiDAR depth completion

This section describes the proposed depth completion method, which aims to provide better performance than traditional image processing algorithms in the downstream multi-modal object detection task. Our objective is to develop a multi-task network in which color and sparse depth maps produce dense depth maps, and that enhanced LiDAR information is again fused with RGB images for object detection. Therefore, our novel proposal focuses on building an efficient LiDAR depth completion network that can be easily integrated into an object detection architecture. Given the importance of LiDAR data to detect objects under adverse weather and illumination conditions, our goal is to obtain higher quality depth maps and enhance the detection performance.

As stated in Section 3.1.1, the recorded LiDAR data in the Waymo dataset is very sparse, with less than 1% of pixels having depth information. Furthermore, projections are often irregular and noisy around object boundaries. Given these issues, directly fusing sparse depth maps with RGB images may degrade the detection performance. Moreover, depth-only upsampling techniques such as bilateral filtering adopted in related studies fail to preserve the structure of many objects. Therefore, an encoder-decoder neural network has been built to solve the upsampling of a sparse depth map in a supervised manner. The proposal is inspired by the network designed in [46], but is simpler and more focused on efficiency. It adopts a single branch scheme to make it more suitable for this real-time application.

Figure 3 presents the proposed depth completion network (LDCNet) with a convolutional encoder-decoder architecture. The model inputs are the sparse LiDAR projections and the RGB camera image. Both inputs are stacked along their depth before the first layer. The network uses geometric convolutions to encode 3D geometric information by appending a 3D position map to the layer's input [46]. Before applying the standard convolution operation, three channels with coordinate information are concatenated to the input feature map [57]. These channels represent a position map

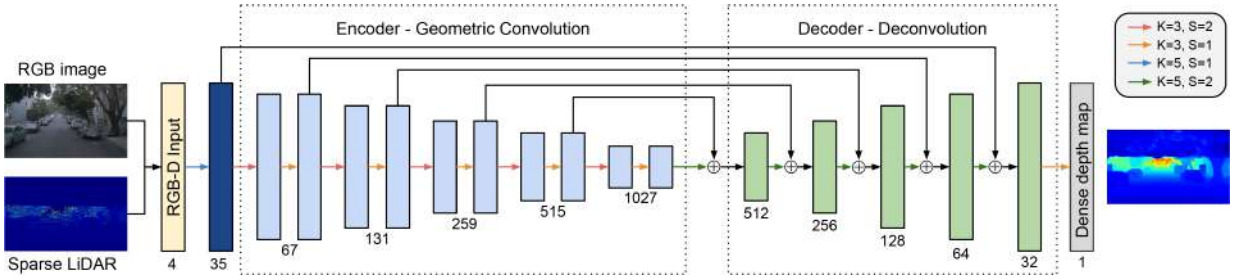


Fig. 3. Our proposed LiDAR Depth Completion network (LDCNet) that outputs a dense depth map combining a camera image and sparse LiDAR projections. The numbers below the maps indicate the number of channels. K and S are the kernel size and stride in the convolution, respectively.

with hard-coded coordinates in the x, y, and z-axis as follows:

$$X = \frac{(i - i_0)Z}{f_x},\ Y = \frac{(j - j_0)Z}{f_y},\ Z = D \quad (1)$$

where (i, j) are the pixel coordinates, and D are depth values obtained by pooling the LiDAR sparse projections. The rest of the values indicate the positional configuration of the camera. The values i_0 and j_0 are the optical centers and f_x and f_y refer to the focal length.

In the proposed LDCNet, the encoder contains one conventional convolutional block and five geometric convolutional blocks. These blocks have residual connections and subsequently reduce the spatial dimensions of the feature maps. Each convolutional operation is followed by batch normalization and ReLU activation. The decoder has five deconvolution layers and one convolutional block. Convolutional kernels of size 3×3 are used in the encoder, which is the usual dimension in ResNet architectures [31]. The decoder uses a slightly larger kernel of 5×5 to better recover the structure of the objects in the upsampling process. The spatial upsampling and downsampling are achieved using convolutions with a stride of two. Residual connections are employed within the encoder and between the encoder and decoder symmetrical maps. For training the model, the loss is the mean squared distance between the predicted depth values and the actual ground truth. The calculation only considers valid pixels with values different from zero.

As explained in the next section, the obtained dense depth maps are fused with RGB images for object detection. With the proposed LDCNet, we carry out an inductive transfer learning setting [58]. Consider depth completion the source task with KITTI as the source domain, and object detection the target task with Waymo as the target domain. The aim is to achieve higher performance on object detection on Waymo by transferring knowledge from a depth completion model trained on another dataset. The LDCNet is trained using the KITTI depth completion dataset, and it is directly used to obtain depth maps for object detection with the Waymo dataset without additional supervision. The fully convolutional architecture allows for this zero-shot transfer learning, even though the images from both datasets have different dimensions and aspect ratios, 1216×352 in KITTI versus 1920×1280 in Waymo. However, the geometric convolutional layers of LDCNet have to be adapted. The values of the coordinates in the position maps change according to the new dimensions.

Given the depth completion literature analysis provided in Section 2.3, we have designed a robust experimental framework to compare our proposal with existing methods. As a baseline, we study the performance when fusing raw sparse LiDAR depth projections with camera data, which is the common methodology followed in related studies that use the Waymo dataset [55,56]. Furthermore, our method is compared to the classical image processing algorithm (CIPA) proposed in [17]. CIPA only uses LiDAR data and concatenates a series of image processing operations to fill empty pixels, such as inversion, dilation, hole filling, bilateral filtering, and median/gaussian blurring. These operations are the usual approach to obtain upsampled depth maps in studies proposing RGB and LiDAR fusion for 2D object detection [50–54].

3.3. Data fusion for object detection

The next step is to fuse the enhanced LiDAR depth map with camera images to perform object detection.

RGB images provide important semantic cues, but LiDAR sensors are not affected by adverse lighting conditions. By fusing high-resolution color and depth information, we aim to exploit the complementary properties of both modalities and improve detection accuracy. For this purpose, we propose a novel data fusion method on top of state-of-the-art object detection meta-architectures such as Faster R-CNN [18],

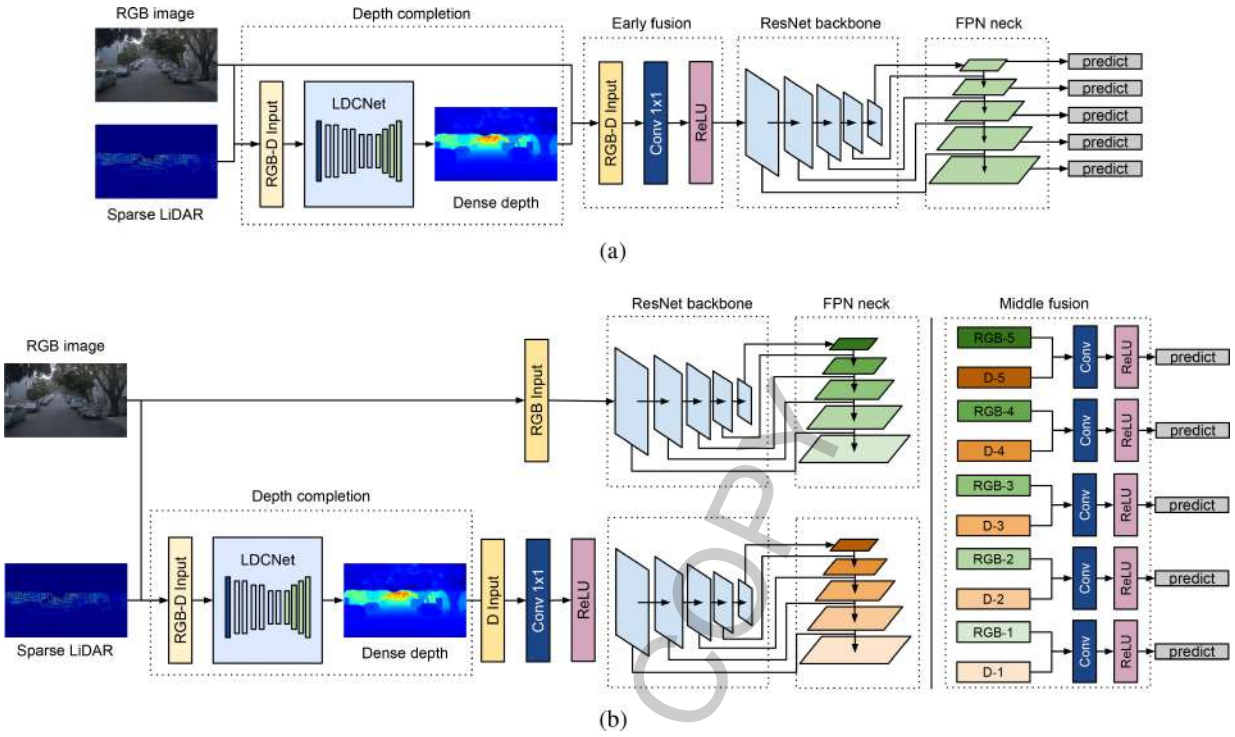


Fig. 4. Different data fusion approaches for object detection explored in this study. These examples illustrate the data fusion process in the single-stage RetinaNet detector. (a) Early fusion architecture; (b) Middle fusion architecture.

ATSS [19], RetinaNet [20] or YOLOF [21]. In order to prove the effectiveness of our proposal, the study includes four architectures with diverse approaches and different speed/accuracy trade-offs. Faster R-CNN is a two-stage detector, while ATSS, RetinaNet, and YOLOF are single-stage. Furthermore, Faster R-CNN, ATSS, and RetinaNet employ feature pyramid networks (FPN) to obtain multi-scale features for detection. In contrast, YOLOF proposes using one-level features to improve efficiency. ATSS is based on RetinaNet, but with an adaptive training sample selection.

Despite their differences, the general pipeline of all these detectors has two fundamental parts: the feature extraction network and the detection head. Given this structure, there are several possibilities to carry out the multi-sensor fusion. The stage at which LiDAR and RGB information is fused can significantly affect the efficiency of the network and the detection performance. Figure 4 illustrates how the proposed LDCNet can be integrated into state-of-the-art object detection meta-architectures to fuse data from both modalities. It presents the data fusion process using an efficient one-stage RetinaNet model with ResNet-50 and a single FPN neck.

Following previous studies [54], we explore early and mid-fusion methodologies in the backbone of the detection architecture. The novelty of our approach is the addition of a depth completion network that enhances the quality of the LiDAR data. Therefore, as seen in Fig. 4, the data fusion happens in two different parts of the network, in the LDCNet and the feature extractor of the detector.

The proposed fusion layer comprises a 1×1 2D convolution followed by the ReLU activation function. Using 1×1 convolutions is less expensive than larger kernels, hence more suitable for fusing data efficiently. Before the convolutional layer, the feature maps from both modalities are combined by concatenation.

Formally, feature maps are tridimensional matrices. Given two feature maps A and B of size $W \times H \times R$ and $W \times H \times D$, respectively. Denote $\copyright$ as their concatenation along the third dimension (depth), the resulting feature maps C can be defined as:

$$C = A \copyright B$$

where

$$c_{ijk} = \begin{cases} a_{ijk} & 1 \leqslant k \leqslant R, 1 \leqslant i \leqslant W, 1 \leqslant j \leqslant H \\ b_{ijk} & R < k \leqslant R + D, 1 \leqslant i \leqslant W, 1 \leqslant j \leqslant H \end{cases} \quad (2)$$

In particular, the RGB input f_0^R is a tridimensional matrix of size $W \times H \times 3$, and the LiDAR input f_0^D is a matrix of size $W \times H \times 1$. The concatenation of both feature maps is a matrix of size $W \times H \times 4$. In general, we denote the tridimensional matrices f_l^R and f_l^D as the feature maps of RGB and LiDAR modalities, respectively, in the l^{th} layer of the network. Mathematically, the proposed fusion operation $\oplus$ can be expressed as follows:

$$f_{l+1} = f_l^R \oplus f_l^D = G_l\left(f_l^R \copyright f_l^D\right) \tag{3}$$

The feature maps f_{l+1} are obtained applying the operation G_l on the concatenation of RGB and LiDAR features from layer l.

In general, $G_l(\cdot)$ is the feature transformation at layer l using 1×1 convolution and the ReLU layer. Consider the feature maps f_{l+1} of size $W \times H \times N$, and f_l of size $W \times H \times M$. The elements of f_l are neurons denoted as a_l^{xyz}, with $x = 1 \dots W$, $y = 1 \dots H$, $z = 1 \dots M$. The operator G_l calculates the values of the neurons of f_{l+1} depending on the neurons of f_l as follows:

$$a_{l+1}^{xyz} = ReLU\left(\sum_{m=1}^{M} w_l^z a_l^{xym} + b_l^z\right)$$
$$\forall x = 1 \dots W,\ \forall y = 1 \dots H,\ \forall z = 1 \dots N \tag{4}$$

where w_l^z stands for the weight in the convolutional kernel, and b_l^z is the bias.

In this work, we study the data fusion at two different stages in the detection network, early and middle. In the early fusion approach, the depth map obtained from LDCNet and the RGB image are stacked along their depth, as can be observed in Fig. 4. Those four channels pass through the fusion layer that combines both modalities and generates three channels that are the input of the detection network. The feature extraction backbone (ResNet with or without FPN) obtains features from the fused map, and the detection head generates the bounding-box predictions. Early fusion can jointly process the information from both modalities, saving computation time and memory. However, this scheme is more sensitive to the quality of the input data.

Given an object detection network with L layers in the feature extraction backbone, the early fusion approach can be described as:

$$f_{l+1} = C_L(C_{L-1}(\dots C_l(C_1(f_0^R \oplus f_0^D)))) \tag{5}$$

where f_0^R and f_0^D are the RGB and LiDAR raw inputs, respectively. C_l is any convolutional feature transformation applied in layer l of the backbone network (ResNet and FPN), with $l \in [1, 2, \dots, L]$.

In contrast, middle fusion allows the network to learn feature representations of both modalities and fuse them at intermediate layers. This approach is less efficient than early fusion because the feature extractor runs for both modalities separately. As can be seen in Fig. 4b, two independent backbones process the RGB image and the depth map from LDCNet. Note that the depth map channels are increased from one to three using a 1×1 convolution before being fed to the ResNet. In this case, the data fusion happens in the multi-scale feature maps obtained from the feature pyramid network of both branches. The feature maps with the same spatial dimensions are concatenated along their depth and fused using the same fusion layer as in early fusion (1×1 convolution and ReLU). Finally, the fused features are used to predict the bounding boxes.

With the same premises mentioned before, the middle fusion process is formulated in Eq. (6). Denote l^* as the intermediate layer where features are fused, with $l^* \in [2, \dots, L-1]$:

$$f_{l^*}^R = C_{l^*}^R(C_{l^*-1}^R(\dots C_1^R(f_0^R)))$$
$$f_{l^*}^D = C_{l^*}^D(C_{l^*-1}^D(\dots C_1^D(f_0^D)))$$
$$f_{l+1} = C_L(C_{L-1}(\dots C_{l^*+1}(f_{l^*}^R \oplus f_{l^*}^D))) \tag{6}$$

It is important to mention the difference between the selected meta-architectures in the middle fusion case. Figure 4b illustrates the data fusion in the single-stage RetinaNet architecture. In the two-stage Faster R-CNN detector, the process is the same except that the fusion happens at the region proposal network, while the second-stage network remains unmodified. In contrast, YOLOF does not use feature pyramid networks, and only one fusion operation happens at the single-scale feature map.

3.4. Implementation details

The code implemented for this study uses the PyTorch MMDetection toolbox [59] and is publicly available at [60]. Except for the additional layers required for fusing RGB and LiDAR data, the rest of the training hyper-parameters are consistent for all experiments and follow the original implementation provided in the MMDetection repository. Given that this real-time application requires efficient models, all experiments use the ResNet-50 backbone network. All models are trained with mixed-precision using the default $1 \times$ learning rate schedule, 12 epochs with a learning rate decay of 1/10 at epochs 8 and 11. The SGD optimizer is used with learning momentum 0.9 and weight de-

cay 1e-4. This training configuration is recommended in popular object detection repositories and many related studies [61]. The batch size is 4, split across two NVIDIA RTX 2080Ti 12 GB GPUs. Scale augmentation is not applied during training, only random horizontal flip. With this setup, training the largest model (Faster R-CNN middle fusion) takes about two days.

The LiDAR depth completion network is separately trained, and the frozen weights are used for inference inside the object detection pipeline. The proposed LDCNet is trained for 25 epochs using the KITTI dataset, with a batch size of 4. This training uses the Adam optimizer with an initial learning rate of 0.001 and exponential decay.

4. Results and discussion

4.1. Experimental setup and evaluation metrics

Firstly, this section presents the experimental study carried out for the design of the sparse-to-dense depth completion network using the KITTI dataset. The performance of our proposed LDCNet is compared to other state-of-the-art models, and the sensitivity analysis to determine its architecture is reported. For the depth completion problem on KITTI, the evaluation metric is the root mean squared error (RMSE) of the distances between the predicted and actual depth of valid pixels.

After analyzing the depth completion task, we present the results obtained on the 2D object detection task using multi-modal data with the Waymo dataset. The average precision (AP) evaluates the detection accuracy. The AP metric computes the area under the precision-recall curve using numerical integration, as shown in Eq. (7). Given N recall (r) thresholds, the calculation is the sum of the precision (p) at every threshold k multiplied by the change in recall $\Delta r(k)$.

$$AP = \int_0^1 p(r)dr \approx \sum_{k=1}^{N} p(k)\Delta r(k) \tag{7}$$

For calculating the precision-recall curve, the intersection-over-union (IoU) is used to determine whether a prediction is a true positive or a false positive. The IoU is the area of overlap between a ground truth box and a predicted box. All detections matching an object box with IoU above a certain threshold are considered true positives, and false positives otherwise. In the Waymo dataset, the required IoU is 0.7 for vehicles and 0.5 for pedestrians and cyclists [7].

Table 1
Sensitivity analysis carried out for the design of the LDCNet using the KITTI depth completion validation set

N. blocks	Geom. Conv.	RMSE (mm)	Inf. time (ms)
3		893.9	**3.0**
3	✓	855.6	3.2
5		995.6	5.0
5	✓	**826.3**	5.1
7		981.2	5.6
7	✓	870.8	6.7

For the analysis of the experimental study, the Waymo dataset is divided into three subsets depending on the lighting conditions: day, night, and dawn/dusk. First, we evaluate the two different data fusion methodologies (early and mid-fusion) on all four detection architectures. Second, we analyze the effect of the quality of LiDAR depth maps on the detection performance and compare them against single-modal detection using only RGB images. Then, we study the difference in accuracy when using sparse depth maps, upsampled maps obtained through image processing algorithms, or the dense depth maps obtained from LDCNet. Finally, we evaluate the efficiency of all the studied models, both in terms of speed and memory requirements.

4.2. LiDAR depth completion

The first step of our study is the design of the depth completion network for enhancing the sparse LiDAR maps. As stated in Section 3.2, the LDCNet is trained on the KITTI depth completion dataset, and then transfer learning is used to obtain dense depth maps for Waymo and improve the precision over the object detection task. Although our main focus in this work is not the depth completion task itself, the performance of the proposed LDCNet is essential for the final object detection purpose. Therefore, it is important to previously evaluate our depth completion method on the KITTI validation set, which is the only labeled dataset available in this context.

Table 1 presents the sensitivity analysis carried out to determine the architecture of the LDCNet. The grid search involves experimenting with different numbers of blocks in the encoder-decoder network and whether using geometric or standard convolution. As can be seen, using geometric convolutions provides better results. Since the inference time is very similar for all configurations, we select the architecture with the lowest RMSE, which has five blocks.

Besides our sensitivity analysis, the quantitative differences between our proposed LDCNet and other state-of-the-art depth completions models are worth men-

Table 2
Performance of state-of-the-art depth completion models on the KITTI validation set. Our proposal is highlighted in bold

Model	RMSE (mm)	Inf. time (ms)	Mem. (GB)
PENet [46]	772.8	18.2	3.7
GuideNet [45]	777.7	21.7	5.1
PwP [44]	811.0	100.0*	–
LDCNet (ours)	**826.3**	**5.1**	**1.8**
RASP [42]	830.5	200.0*	–
Sparse-to-dense [43]	878.6	11.3	10.8
DepthNet [39]	991.9	90.0*	–
DFuseNet [62]	1206.7	47.0	3.7
CIPA [17]	1288.5	9.2	–
ADNN [63]	1350.0	40.0*	–

tioning. Table 2 presents a comparison over the KITTI validation set in terms of RMSE, inference time, and memory usage between our proposal and several methods found in the literature. For fair efficiency comparison, the models with publicly available code have been tested, and the time and memory usage have been measured on our server (NVIDIA RTX 2080Ti GPU). The models with a star sign do not have a public implementation, and the inference times have been taken from the official KITTI leaderboard.

The main motivation behind the design of the LDCNet is to build an efficient depth completion network that can be easily integrated into an object detection meta-architecture without a significant increase in the computational cost. As seen in Table 2, our proposal is the most efficient method while also being very competitive in depth completion accuracy. LDCNet is four times faster than the two most accurate models (PENet and GuideNet) and requires less memory. The inference time of object detection models on Waymo images varies, for instance, from 40 ms using YOLOF to 57 ms using Faster R-CNN. Considering this aspect, introducing a depth completion network in the pipeline that adds 20 ms, such as PENet, is not convenient for this real-time application. Therefore, these results support the advantages of LDCNet for the downstream object detection task for autonomous driving. Our proposal introduces a minimal computation time overhead and has a lower memory usage than other state-of-the-art depth completion models.

4.3. Multi-modal detection using LDCNet

Table 3 presents the average precision (AP) results obtained with the proposed data fusion network using the dense LiDAR depth maps obtained from LDCNet. We compare the performance of this multi-modal detection approach using early and middle fusion on all four meta-architectures studied. As found in previous studies with this dataset [31], the two-stage Faster R-CNN architecture provides the best detection accuracy. For instance, with early fusion at daytime, the AP of Faster R-CNN is 4.4 and 6.9 points better than RetinaNet and YOLOF. ATSS also provides competitive results, outperforming Faster R-CNN only in a few cases, such as the mean AP of dawn/dusk images.

Regarding the fusion stage of the two modalities, some differences can be observed between both approaches. In general, middle fusion provides a better AP at daytime and dawn/dusk, while early fusion obtains a better detection accuracy at night. For instance, with Faster R-CNN, the AP is slightly better using middle fusion with daylight, but early fusion provides a 1.4 mAP increase at night. Furthermore, with ATSS, the AP at daytime and dawn/dusk is over one point better with the middle fusion approach. In contrast, more vehicles can be detected at night when using early fusion in ATSS. The same conclusions can be derived from the results obtained by the RetinaNet model. These results suggest that, with bad illumination at night, it is better to jointly learn features from both modalities at an early stage.

Different behavior can be seen with the YOLOF meta-architecture. With this model, early fusion outperforms middle fusion in all lighting conditions. We hypothesize that this difference between the detection meta-architectures is because of the absence of feature pyramid networks in YOLOF. The LiDAR features can be less important during the daytime than the visual cues provided by RGB images. Therefore, the FPN can help to better assign the importance of depth features at different scales. In contrast, middle fusion is ineffective with the single-scale feature map in YOLOF, and the detection performance is degraded.

Overall, these results show that our depth completion network (LDCNet) effectively transforms the sparse LiDAR projection into a dense map with rich depth information for detection. However, selecting the optimal data fusion method highly depends on the detection model used and the time of the day in which objects have to be detected. Given the results from Table 3, our selected approach is to adopt early fusion at night and middle fusion when there is better illumination (at daytime and dawn/dusk). This proposed data fusion architecture offers a robust performance throughout different times of day with a small computational overhead, as analyzed in Section 4.6.

4.4. Importance of accurate LiDAR depth completion for object detection

This section evaluates the influence of different LiDAR inputs on the multi-modal object detection task.

Table 3
AP results fusing RGB images with the LDCNet dense depth maps at different stages (early and middle) using the four detection meta-architectures. Best results are highlighted in bold

		Day				Night				Dawn/dusk			
Model	LiDAR fusion	Veh	Ped	Cyc	Mean	Veh	Ped	Cyc	Mean	Veh	Ped	Cyc	Mean
Faster R-CNN	Early	55.5	68.5	**49.7**	57.9	**55.9**	**61.0**	**64.5**	**60.5**	**62.6**	62.8	**40.2**	55.2
	Middle	**55.7**	**68.8**	49.6	**58.0**	55.2	**61.0**	61.2	59.1	**62.6**	**64.3**	38.9	**55.3**
ATSS	Early	54.9	67.5	47.0	56.5	**56.4**	61.5	59.4	59.1	61.9	63.0	40.8	55.2
	Middle	**55.9**	**68.7**	**48.5**	**57.7**	55.5	**61.7**	**62.9**	**60.0**	**62.3**	**64.6**	**42.5**	**56.5**
RetinaNet	Early	51.4	65.6	**43.4**	**53.5**	**54.8**	59.9	**57.1**	**57.3**	58.6	60.7	**38.5**	**52.6**
	Middle	**51.9**	**66.5**	42.1	**53.5**	53.3	**60.0**	51.6	55.0	**59.2**	**61.1**	36.6	52.3
YOLOF	Early	**47.8**	**60.4**	**44.2**	**50.8**	**51.6**	**58.2**	**58.2**	**56.0**	**54.7**	**55.3**	39.6	**49.9**
	Middle	47.1	60.1	44.1	50.4	49.5	54.1	53.8	52.5	54.0	55.1	**40.5**	**49.9**

Table 4
AP results fusing RGB images with different LiDAR inputs on the four studied meta-architectures. Best results are highlighted in bold

		Day				Night				Dawn/dusk			
Model	LiDAR input	Veh	Ped	Cyc	Mean	Veh	Ped	Cyc	Mean	Veh	Ped	Cyc	Mean
Faster R-CNN	RGB only	55.6	68.7	**50.0**	**58.1**	53.3	52.1	53.9	53.1	62.0	62.6	36.9	53.8
	Raw sparse	54.7	67.4	46.9	56.3	54.6	56.2	58.7	56.5	61.5	62.4	37.5	53.8
	CIPA	54.7	67.1	49.7	57.2	53.7	54.3	58.8	55.6	60.7	61.3	**39.0**	53.7
	LDCNet	**55.7**	**68.8**	49.6	58.0	**55.9**	**61.0**	**64.5**	**60.5**	**62.6**	**64.3**	38.9	**55.3**
ATSS	RGB only	55.5	67.8	46.0	56.4	53.8	52.6	51.7	52.7	61.7	62.4	38.0	54.0
	Raw sparse	54.6	66.5	42.2	54.4	55.4	56.9	54.8	55.7	60.8	61.6	36.5	53.0
	CIPA	53.0	65.5	44.8	54.4	**56.8**	59.6	57.5	58.0	60.4	61.2	39.9	53.6
	LDCNet	**55.9**	**68.7**	**48.5**	**57.7**	56.4	**61.5**	**59.4**	**59.1**	**62.3**	**64.6**	**42.5**	**56.5**
RetinaNet	RGB only	51.6	65.8	**43.1**	**53.5**	52.0	50.9	51.9	51.6	58.4	59.7	36.3	51.5
	Raw sparse	50.6	63.6	40.5	51.6	52.6	54.1	56.2	54.3	57.6	59.1	32.0	49.6
	CIPA	50.3	63.6	40.8	51.6	52.9	57.5	53.3	54.6	56.8	58.5	34.4	49.9
	LDCNet	**51.9**	**66.5**	42.1	**53.5**	**54.8**	**59.9**	**57.1**	**57.3**	**59.2**	**61.1**	**36.6**	**52.3**
YOLOF	RGB only	46.9	59.1	39.6	48.5	48.0	47.9	41.6	45.8	53.5	52.8	32.6	46.3
	Raw sparse	46.4	58.4	42.4	49.1	49.5	53.0	49.7	50.7	52.9	53.9	34.3	47.0
	CIPA	46.9	59.9	43.0	49.9	51.0	57.9	55.0	54.6	**54.5**	**55.4**	38.0	49.3
	LDCNet	**47.1**	**60.1**	**44.1**	**50.4**	**51.6**	**58.2**	**58.2**	**56.0**	54.0	55.1	**40.5**	**49.9**

The accuracy of the four meta-architectures is investigated when using sparse LiDAR depth maps, upsampled maps from classical image processing algorithms (CIPA), and dense depth maps from the proposed LDCNet. Table 4 presents the results obtained using these three different depth inputs and compares them to camera-only detection. All dual-modal models use early fusion at night and middle fusion with day and dawn/dusk images.

As expected, single-modal RGB detection suffers a significant performance drop under adverse lighting conditions. For instance, with the best performing model (Faster R-CNN), the mean AP drops from 58.1% in the daytime to 53.1% at night. A similar decrease can be observed in the other models when using only camera image inputs. The main strength of our proposal, fusing RGB images with accurate dense LiDAR depth maps, is that it consistently enhances accuracy under different lighting conditions. The detection precision improves not only at nighttime but also with day and dawn/dusk images, when there is better illumination and the importance of LiDAR is not as obvious. With the infinite diversity of driving situations and environmental conditions, leveraging the complementary properties of the multi-modal sensors is essential for the perception systems of autonomous vehicles.

At night, our multi-modal approach improves RGB-only by 7.4, 6.4, 5.7, and 10.2 points on Faster R-CNN, ATSS, RetinaNet, and YOLOF, respectively. The AP increase in the minority classes, pedestrians and cyclists, is particularly interesting. As can be observed, our proposed data fusion network detects up to 10% more pedestrians and cyclists. The improvement is lower but still very significant regarding the vehicle class, ranging from 2.6 to 3.6 points. At dawn/dusk, following the order of detectors in Table 4, the mAP is enhanced by 1.5, 2.5, 0.8, and 3.6 with dual-modal detection. With daytime images, fusing both modalities provides an important advantage over RGB-only on two out of the four models studied, ATSS and YOLOF.

Furthermore, our approach is better than fusing with the other LiDAR inputs considered. As can be seen in Table 4, using LiDAR sparse depth maps leads to similar results as CIPA, with this last method obtaining slightly better outcomes than raw data. With both approaches, there is a significant decay of AP at daytime and dawn/dusk compared to RGB detection. For instance, day mAP drops 1.8 (sparse) and 0.9 (CIPA) points on Faster R-CNN. In contrast, the AP using LDCNet remains comparable to or even better than camera-based detection thanks to the better quality depth information. These results support the superiority of our LDCNet when upsampling depth maps. They indicate that, with good illumination, dual-modal detection can perform worse if sparse LiDAR data is not accurately preprocessed.

All LiDAR inputs improve detection at nighttime. For instance, introducing raw sparse depth inputs already provides an AP increment of about 3 points. However, the greatest performance improvement at night is obtained with LDCNet inputs. Our depth completion network outperforms CIPA by 4.9, 1.4, 2.7, 1.0 mAP with the different detectors studied. At dawn/dusk, LDCNet also provides more accurate detections, with AP increases ranging from 1.5 to 3 points.

Overall, our proposed method outperforms both raw data and CIPA inputs in all times of day situations. Furthermore, it considerably enhances the performance of RGB-only models. We achieve these improvements thanks to our study of the optimal data fusion architecture and the better quality depth maps generated by LDCNet.

Figure 5 presents a qualitative comparison between our dual-modal detection approach and RGB-only models. This figure illustrates typical examples of objects detected using the proposed RGB-LiDAR fusion, but not when using only camera images as input. The selected examples include diverse lighting conditions such as night, dawn, or rain. They also include all three types of objects with different dimensions.

For instance, the first row of Fig. 5 illustrates two cyclists at night that are not detected with RGB-only detectors. In the first case, the object is classified as a pedestrian. In the second case, the object is missed due to its small size and bad illumination. The associated depth maps clearly show that LiDAR provides essential information to detect these cyclists in the dark. Furthermore, the second and third rows in Fig. 5 display cars and pedestrians that are very close to the vehicle but are not detected with camera images given their dark colors. These missed detections may compromise the safety of the system, and illustrate the need for fusing RGB and LiDAR at night to increase the robustness of the detection model.

The rest of the scenes in Fig. 5 illustrate various situations in which LiDAR is fundamental for detecting small and occluded pedestrians. Even with better illumination conditions, the LDCNet dense maps help identify pedestrians wearing clothes with colors similar to the road (fourth row) and pedestrians occluded by other objects (fifth row). In summary, this qualitative analysis supports the results presented in Table 4. It demonstrates the suitability of the proposed LiDAR depth completion method, which helps to build a more reliable multi-modal detection system.

4.5. Comparison between dense LiDAR inputs

Figure 6 presents a visual comparison of the different LiDAR depth maps used in the experimental study. Unlike KITTI, the Waymo images have significantly higher resolution, and the LiDAR projections are extremely sparse. The second row of Fig. 6 illustrates the sparse depth maps, with less than 1% of pixels with valid values. In this figure, dark blue colors illustrate pixels with depth values near zero, hence belonging to objects close to the LiDAR sensor. In contrast, lighter blues, yellow and red colors, in this order, indicate distant depth values belonging to far-away objects. As seen in the third row, it is difficult for the CIPA approach to upsample the sparse map accurately. These image processing algorithms still leave many pixels without information, propagating many zero values all over the image, especially due to mirror reflection on vehicles. The very dark pixels within the boundaries of the shape of distant vehicles illustrate this issue.

In contrast, our proposed LDCNet better preserves the structure of objects in the scene. Our neural network approach can better reconstruct the shape of thin elements, such as traffic signal poles. Thanks to the RGB guidance in the depth completion process, LDCNet can also avoid glass reflection issues and model faraway small objects more accurately. Furthermore, it can infer objects' shapes that span beyond the LiDAR device height range, such as trees, buildings, or tall traffic signals. It is essential to highlight that the supervised LDCNet was trained on the KITTI dataset, and used to infer the Waymo dense maps without additional supervision. The obtained results show the capacity of generalization of the network and the advantages of this transfer learning approach compared to classical algorithms, such as bilateral filtering. Our approach allows

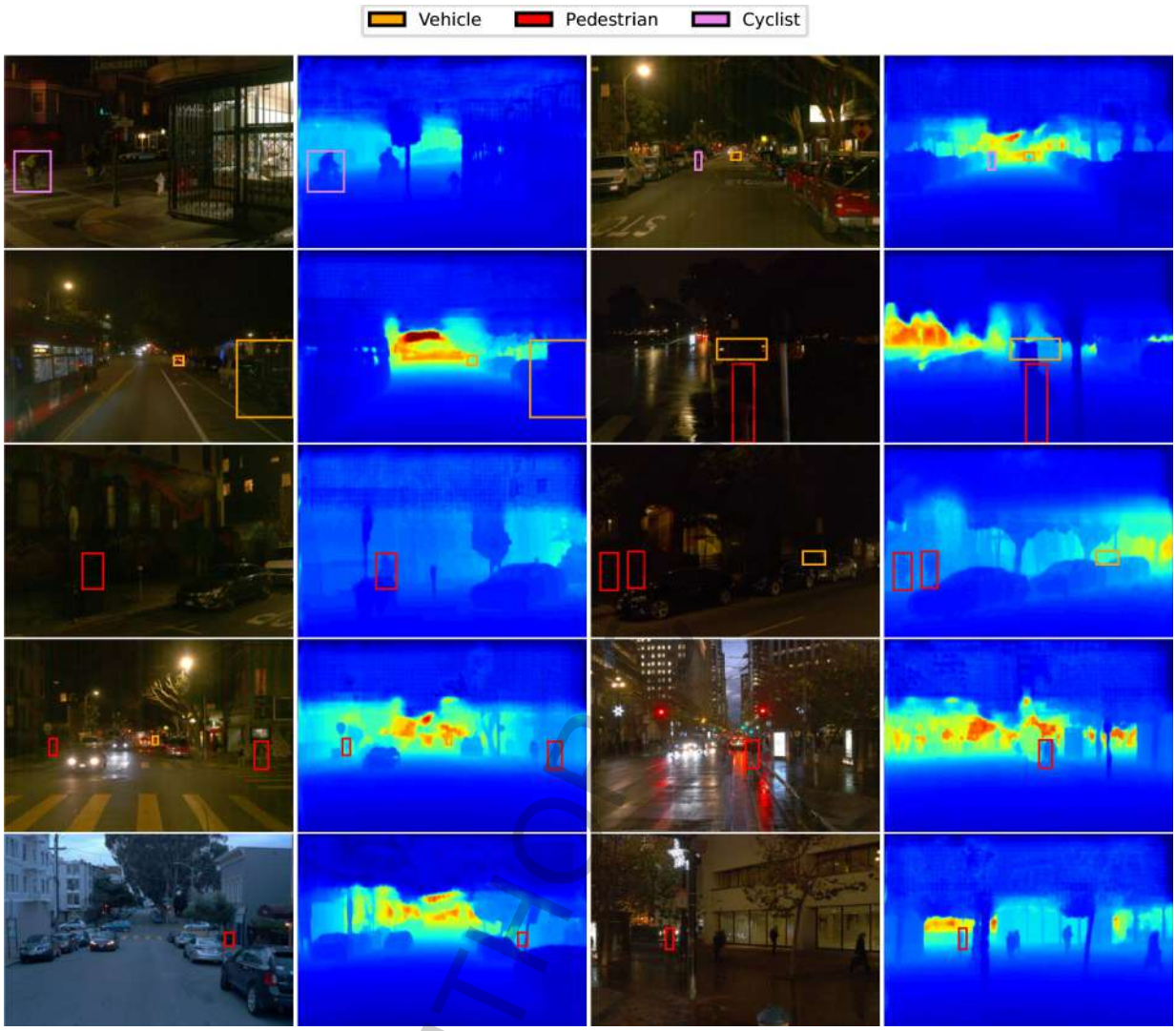


Fig. 5. Typical examples illustrating the advantages of the proposed dual-modal detection approach. The highlighted objects are detected by our RGB-LDCNet fusion network, but the RGB-only models fail to detect them. For each scene, the camera images and the dense LDCNet depth maps are displayed.

Table 5
AP detection results using LiDAR only inputs

Model	Input (LiDAR only)	Veh.	Ped.	Cyc.	Mean
Faster R-CNN	CIPA	28.2	45.0	20.6	31.3
	LDCNet	**30.0**	**50.3**	**23.3**	**34.5**
ATSS	CIPA	28.5	47.5	24.2	33.4
	LDCNet	**30.8**	**50.2**	**25.1**	**35.4**

leveraging the information of a different dataset from a similar domain to solve this specific task efficiently and effectively.

In addition to the visual comparison, a quantitative analysis between both dense LiDAR inputs is provided in Table 5. The detection performance using the depth maps obtained from CIPA and LDCNet is evaluated to provide more evidence of the superiority of our proposal. This comparison uses LiDAR-only inputs for detection without fusing them with RGB images. As can be seen in Table 5, LDCNet inputs outperform CIPA by 3.2 and 2.0 mAP using Faster R-CNN and ATSS models, respectively. The difference in depth completion accuracy between LDCNet and CIPA translates to better performance on the downstream object detection task on Waymo, which is the final aim of our study. These experiments further support the results presented in Table 4, where LDCNet outperformed other data fusion approaches.

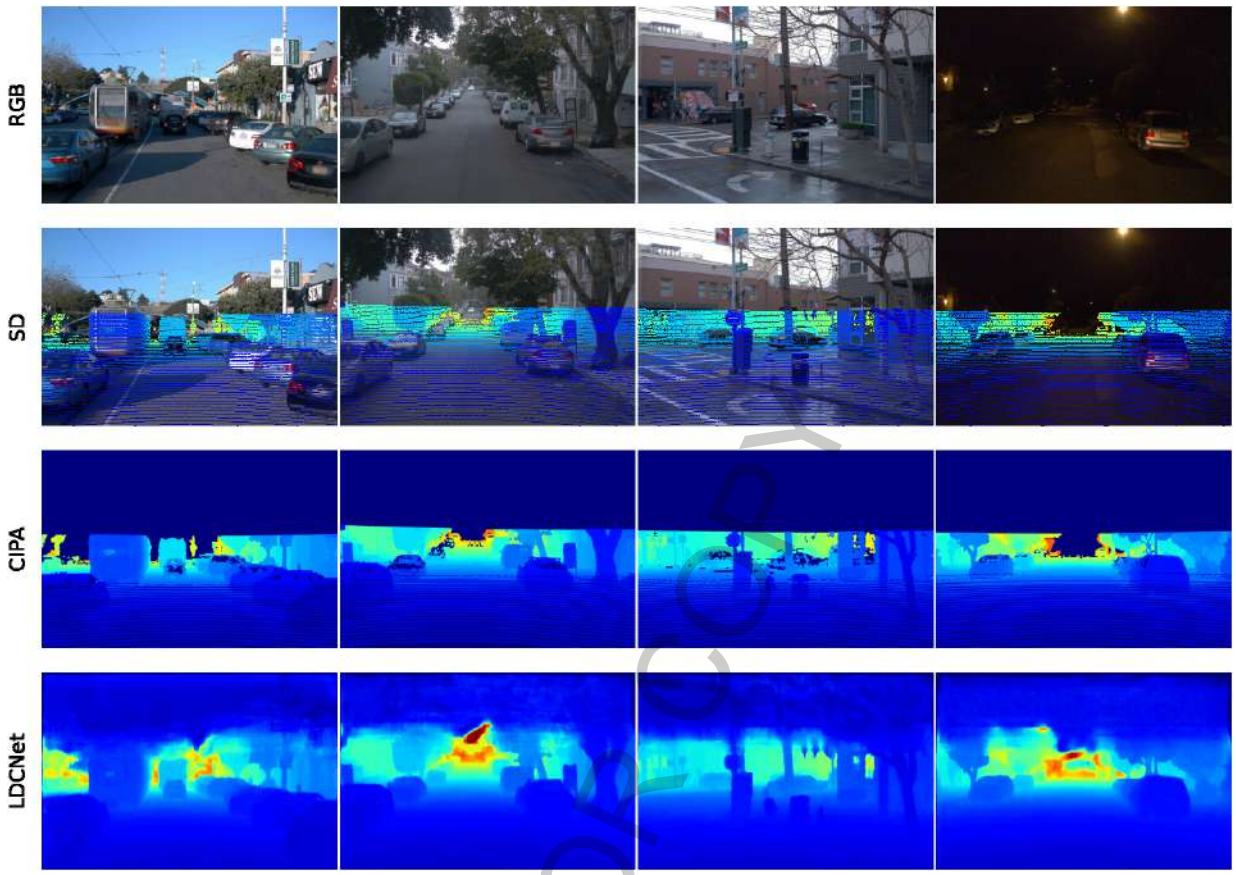


Fig. 6. Illustrations of the different LiDAR depth maps that can be fused with RGB images for object detection on the Waymo dataset. From top to bottom: RGB camera image, raw sparse LiDAR depth projections (SD), upsampled maps obtained using classical image processing algorithms (CIPA), and dense depth map obtained with the proposed LDCNet.

4.6. *Efficiency analysis*

This section analyzes the computational efficiency of all the architectures and the different data fusion methods explored. Table 6 presents the speed in frames per second (FPS) and the memory requirements of the studied models. These metrics are essential aspects to consider for autonomous driving, where fast predictions are needed to operate in real-time and the computational resources are limited. Furthermore, in order to facilitate the accuracy/speed comparison, Table 6 also provides a summary of the mean AP at different times of the day. We report the frames per second for inference with batch size one using an NVIDIA RTX 2080Ti GPU. The speed values of the models using RGB-LiDAR fusion include the depth completion method in the object detection pipeline.

As can be seen, in the early fusion approach, the addition of the LDCNet provides a significant AP improvement without sacrificing speed. For instance, in Faster R-CNN the computational overhead is minimal. The FPS drop from 17.4 at single-modal detection to 15.0 when LiDAR data is incorporated. These values illustrate that the proposed depth completion network is efficient and suitable for this application. LDCNet runs in only 5 ms, which is about 7% of the total time of Faster R-CNN when using the early fusion approach (66 ms). Note that the time employed by LDCNet is independent of the detection meta-architecture used, hence the analysis is similar for the rest of the models.

In contrast, the middle fusion approach is less efficient since features are extracted separately for both modalities. Compared to early fusion, there is a drop of about 5, 4, 4, and 8 FPS in Faster R-CNN, ATSS, RetinaNet, and YOLOF, respectively. Furthermore, as expected, dual-modal detection increases the memory usage by about 2 $\times$ with early fusion and 4 $\times$ with mid-fusion. It is also important to mention that the other alternative considered for depth completion, CIPA, is not practical for this application. CIPA requires about 40 ms

Table 6
AP summary, frames per second (FPS), and memory usage of the four detection meta-architectures with different data fusion approaches

Model	LiDAR fusion	AP_{Day}	AP_{Night}	$AP_{Dawn/Dusk}$	FPS	Mem. (GB)
Faster R-CNN	RGB only	**58.1**	53.1	53.8	17.4	2.8
	Raw sparse	56.3	56.5	53.8	16.2	3.8
	CIPA	57.2	55.6	53.7	9.9	3.8
	LDCNet-early	57.9	**60.5**	55.2	15.0	5.8
	LDCNet-middle	58.0	59.1	**55.3**	9.7	8.7
ATSS	RGB only	56.4	52.7	54.0	15.6	2.6
	Raw sparse	54.4	55.7	53.0	14.8	3.5
	CIPA	54.4	58.0	53.6	9.3	3.5
	LDCNet-early	56.5	59.1	55.2	13.8	5.5
	LDCNet-middle	**57.7**	**60.0**	**56.5**	10.1	7.0
RetinaNet	RGB only	**53.5**	51.6	51.5	16.5	2.0
	Raw sparse	51.6	54.3	49.6	15.8	2.9
	CIPA	51.6	54.6	49.9	10.0	2.9
	LDCNet-early	**53.5**	**57.3**	**52.6**	14.6	5.0
	LDCNet-middle	**53.5**	55.0	52.3	10.3	6.2
YOLOF	RGB only	48.5	45.8	46.3	25.2	1.7
	Raw sparse	49.1	50.7	47.0	23.3	2.6
	CIPA	49.9	54.6	49.3	12.1	2.6
	LDCNet-early	**50.8**	**56.0**	**49.9**	20.9	4.6
	LDCNet-middle	50.4	52.5	**49.9**	13.7	6.0

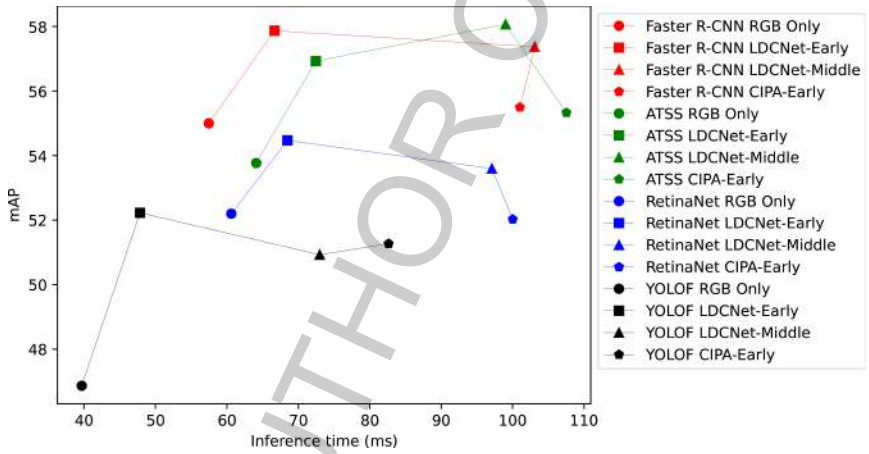


Fig. 7. Speed/accuracy trade-off of the studied uni-modal and dual-modal object detection architectures. The mAP displayed is the mean of the three times of day considered.

to upsample the sparse maps, which is even slower than the complete YOLOF detection architecture.

Figure 7 presents a graphical summary of the speed/accuracy trade-off of the uni-modal an dual-modal detection networks studied. The lines in the figure connect the same meta-architectures, and illustrate the mAP improvement obtained when fusing both modalities with early and middle approaches. In general, it can be seen that the proposed early fusion network is the most suitable option for efficiently improving the robustness of detectors. Using LDCNet and feeding a single backbone with both modalities introduces a small overhead, allowing all models to have a higher processing speed than the capture rate of Waymo's vehicle LiDAR sensor (10 Hz). In contrast, middle fusion runs slower than early fusion but provides a higher detection precision when there is daylight.

5. Conclusions

This study proposed a novel multi-modal network for object detection in autonomous driving. Our aim was to improve the robustness of detectors under adverse lighting conditions by fusing data from two different sensors, RGB cameras and LiDAR. For this purpose,

we designed a data fusion method and evaluated its performance at different times of the day over the Waymo Open Dataset, which is the most diverse self-driving dataset.

First, our proposal integrated an efficient sparse-to-dense depth completion network (LDCNet) into the detection pipeline. This supervised network allowed to upsample the resolution of sparse LiDAR projections using RGB guidance, and improved the quality of the depth information. The LDCNet was trained with the KITTI depth dataset, and the dense depth maps for Waymo were obtained using transfer learning. The experimental study demonstrated that this novel multi-source deep learning approach provided more accurate dense depth maps than classical image processing algorithms.

With the enhanced LiDAR data, we explored two different data fusion approaches, early and middle, using popular detection meta-architectures such as Faster R-CNN, ATSS, RetinaNet, and YOLOF. The performance of dual-modal detection was compared against single-modal RGB detection in terms of precision and computational cost. The study showed that incorporating the dense depth maps into the object detection pipeline is essential when there is bad illumination, but can also be beneficial with dawn/dusk or daytime images if sparse depth maps are accurately processed.

The improvement obtained with our proposal was consistent across four meta-architectures with different characteristics, with early fusion achieving higher detection precision at night and middle fusion at daytime and dawn/dusk.

Overall, the results showed that our LDCNet approach successfully leverages the complementary properties of both modalities for object detection. Furthermore, the efficiency analysis showed that the depth completion network could easily be integrated into traditional detection architectures with minimal computational overhead. Early fusion proved to be the most efficient alternative, obtaining a more reliable detection system while running at a speed comparable to single-modal camera detection.

In future studies, we aim to exploit the temporal nature of autonomous driving data by developing recurrent approaches for sequential perception. We plan to design novel adaptive fusion methods that can learn how to combine different modalities in a streaming fashion. Moreover, with the increasing amount of available self-driving data, further research should develop effective transfer learning and domain adaptation approaches. We are also interested in the advantages and promising results that could offer recent supervised algorithms, such as the enhanced probabilistic networks [64], or dynamic ensemble learning algorithms [65]. The extent to which more complex models and more diverse training data offer an advantage is an interesting question to be addressed. Finally, another important line of work is to further study the proposed models' efficiency and suitability when deployed on embedded devices in vehicles.

Funding

This research has been funded by FEDER/Ministerio de Ciencia e Innovación – Agencia Estatal de Investigación/Proyecto PID2020-117954RB-C22 and by the Andalusian Regional Government under the projects: BIDASGRI: Big Data technologies for Smart Grids (US-1263341), Adaptive hybrid models to predict solar and wind renewable energy production (P18-RT-2778). M.C.G receives funding from the predoctoral fellowship FPU18/00622.

References

[1] Wang Z, Zhao X, Xu Z, Li X, Qu X. Modeling and field experiments on autonomous vehicle lane changing with surrounding human-driven vehicles. Computer-Aided Civil and Infrastructure Engineering. 2020; 36: 877-889.

[2] Foresti GL, Scagnetto I. An integrated low-cost system for object detection in underwater environments. Integrated Computer-Aided Engineering. 2022; 29(2): 123-139.

[3] Yang T, Cappelle C, Ruichek Y, El Bagdouri M. Multi-object tracking with discriminant correlation filter based deep learning tracker. Integrated Computer-Aided Engineering. 2019; 26(3): 273-284.

[4] Wang Y, Hou S, Wang X. Reinforcement learning-based bird-view automated vehicle control to avoid crossing traffic. Computer-Aided Civil and Infrastructure Engineering. 2021; 36(7): 890-901.

[5] Zhao D, Li X, Cui J. A simulation-based optimization model for infrastructure planning for electric autonomous vehicle sharing. Computer-Aided Civil and Infrastructure Engineering. 2021; 36(7): 858-876.

[6] Chen S, Leng Y, Labi S. A deep learning algorithm for simulating autonomous driving considering prior knowledge and temporal information. Computer-Aided Civil and Infrastructure Engineering. 2019; 35: 305-321.

[7] Waymo Open Dataset: An autonomous driving dataset; 2019. (Accessed 28 March 2022). Available online: https://www.waymo.com/open.

[8] Caesar H, et al. Nuscenes: A multimodal dataset for autonomous driving. 2020; 11618-11628.

[9] Hesai, Scale. PandaSet: Public large-scale dataset for autonomous driving. 2019. (Accessed 7 February 2022). Available online: https://scale.com/open-datasets/pandaset.

[10] Feng D, Haase-Schutz C, Rosenbaum L, Hertlein H, Glaser C, Timm F, et al. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. IEEE Transactions on Intelligent Transportation Systems. 2021; 22(3): 1341-1360.

[11] Shen J, Yan W, Li P, Xiong X. Deep learning-based object identification with instance segmentation and pseudo-liDAR point cloud for work zone safety. Computer-Aided Civil and Infrastructure Engineering. 2021; 36(12): 1549-1567.

[12] Park SW, Park HS, Kim JH, Adeli H. 3D displacement measurement model for health monitoring of structures using a motion capture system. Measurement. 2015; 59: 352-362.

[13] Oh BK, Kim KJ, Kim Y, Park HS, Adeli H. Evolutionary learning based sustainable strain sensing model for structural health monitoring of high-rise buildings. Applied Soft Computing. 2017; 58: 576-585.

[14] Kalenjuk S, Lienhart W, Rebhan M. Processing of mobile laser scanning data for large-scale deformation monitoring of anchored retaining structures along highways. Computer-Aided Civil and Infrastructure Engineering. 2021; 36(6): 678-694.

[15] Rashed H, Ramzy M, Vaquero V, El Sallab A, Sistu G, Yogamani S. FuseMODNet: Real-time camera and liDAR based moving object detection for robust low-light autonomous driving. Proceedings – International Conference on Computer Vision Workshop, ICCVW. 2019; 2393-2402.

[16] Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? The KITTI Vision Benchmark Suite. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition. 2012; 3354-3361.

[17] Ku J, Harakeh A, Waslander SL. In defense of classical image processing: Fast depth completion on the CPU. 15th Conference on Computer and Robot Vision (CRV). 2018; 16-22.

[18] Ren S, He K, Girshick R, Sun J. Faster r-CNN: Towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017; 39(6): 1137-1149.

[19] Zhang S, Chi C, Yao Y, Lei Z, Li SZ. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2020; 9756-9765.

[20] Lin T, Goyal P, Girshick R, He K, Dollár P. Focal loss for dense object detection. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2020; 42(2): 318-327.

[21] Chen Q, Wang Y, Yang T, Zhang X, Cheng J, Sun J. You only look one-level feature. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2021; 13034-13043.

[22] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2016; 770-778.

[23] Lin T, Dollár P, Girshick R, He K, Hariharan B, Belongie S. Feature pyramid networks for object detection. in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017; 936-944.

[24] Tian Z, Shen C, Chen H, He T. FCOS: Fully convolutional one-stage object detection. in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). 2019; 9626-9635.

[25] Zhou X, Wang D, Krähenbühl P. Objects as points. CoRR. 2019; abs/1904.07850.

[26] Carion N, Massa F, Synnaeve G, Usunier N, Kirillov A, Zagoruyko S. End-to-end object detection with transformers. in: Computer Vision – ECCV 2020. 2020; 213-229.

[27] Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, et al. Microsoft COCO: Common objects in context. in: Computer Vision – ECCV 2014. 2014; 740-755.

[28] Xie S, Girshick R, Dollár P, Tu Z, He K. Aggregated residual transformations for deep neural networks. 2017.

[29] Cai Z, Vasconcelos N. Cascade r-CNN: Delving into high quality object detection. in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2018; 6154-6162.

[30] Buenaposada JM, Baumela L. Improving multi-class boosting-based object detection. Integrated Computer-Aided Engineering. 2021; 28(1): 81-96.

[31] Carranza-García M, Torres-Mateo J, Lara-Benítez P, García-Gutiérrez J. On the performance of one-stage and two-stage object detectors in autonomous vehicles using camera data. Remote Sensing. 2021; 13(1): 89.

[32] Carranza-García M, Lara-Benítez P, García-Gutiérrez J, Riquelme JC. Enhancing object detection for autonomous driving by optimizing anchor generation and addressing class imbalance. Neurocomputing. 2021; 449: 229-244.

[33] Wang Y, Liu Z, Deng W. Anchor generation optimization and region of interest assignment for vehicle detection. Sensors. 2019 03; 19: 1089.

[34] Hassaballah M, Kenk MA, Muhammad K, Minaee S. Vehicle detection and tracking in adverse weather using a deep learning framework. IEEE Transactions on Intelligent Transportation Systems. 2020; 1-13.

[35] Zhang S, Benenson R, Omran M, Hosang J, Schiele B. Towards reaching human performance in pedestrian detection. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2018; 40(4): 973-986.

[36] Lian J, Yin Y, Li L, Wang Z, Zhou Y. Small object detection in traffic scenes based on attention feature fusion. Sensors. 2021; 21(9).

[37] Arcos-García Á, Álvarez García JA, Soria-Morillo LM. Evaluation of deep neural networks for traffic sign detection systems. Neurocomputing. 2018; 316: 332-344.

[38] Uhrig J, Schneider N, Schneider L, Franke U, Brox T, Geiger A. Sparsity invariant CNNs. in: 2017 International Conference on 3D Vision (3DV). 2017; 11-20.

[39] Bai L, Zhao Y, Elhousni M, Huang X. DepthNet: Real-time liDAR point cloud depth completion for autonomous vehicles. IEEE Access. 2020; 12; 8: 1-1.

[40] Lu K, Barnes N, Anwar S, Zheng L. From depth what can you see? Depth Completion Via Auxiliary Image Reconstruction. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2020; 11303-11312.

[41] Gu J, Xiang Z, Ye Y, Wang L. DenseLiDAR: A real-time pseudo dense depth guided depth completion network. IEEE Robotics and Automation Letters. 2021; 6(2): 1808-1815.

[42] Lee S, Lee J, Kim D, Kim J. Deep architecture with cross guidance between single image and sparse liDAR data for depth completion. IEEE Access. 2020; 8: 79801-79810.

[43] Ma F, Cavalheiro GV, Karaman S. Self-supervised sparse-to-dense: Self-supervised depth completion from liDAR and monocular camera. in: 2019 International Conference on Robotics and Automation (ICRA). 2019; 3288-3295.

[44] Xu Y, Zhu X, Shi J, Zhang G, Bao H, Li H. Depth completion from sparse liDAR data with depth-normal constraints. in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). 2019; 2811-2820.

[45] Tang J, Tian F, Feng W, Li J, Tan P. Learning guided convolutional network for depth completion. IEEE Transactions on Image Processing. 2021; 30: 1116-1129.

[46] Hu M, Wang S, Li B, Ning S, Fan L, Gong X. PENet: Towards precise and efficient image guided depth completion. 2021

IEEE International Conference on Robotics and Automation (ICRA). 2021; 13656-13662.
[47] Premebida C, Carreira JA, Batista J, Nunes U. Pedestrian detection combining RGB and dense LIDAR data. in: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems. 2014; 4112-4117.
[48] Guo ZX, Liao WZ, Xiao YF, Veelaert P, Philips W. Deep learning fusion of RGB and depth images for pedestrian detection. in: 30th British Machine Vision Conference (BMVC), Proceedings. 2019; 1-13.
[49] Ophoff T, Van Beeck K, Goedemé T. Exploring RGB+depth fusion for real-time object detection. Sensors. 2019; 19(4).
[50] Kim J, Kim J, Cho J. An advanced object classification strategy using YOLO through camera and liDAR sensor fusion. in: 2019 13th International Conference on Signal Processing and Communication Systems (ICSPCS). 2019; 1-5.
[51] Li Y, Niu J, Ouyang Z. Fusion strategy of multi-sensor based object detection for self-driving vehicles. in: 2020 International Wireless Communications and Mobile Computing (IWCMC). 2020; 1549-1554.
[52] Pfeuffer A, Dietmayer K. Optimal sensor data fusion architecture for object detection in adverse weather conditions. in: 2018 21st International Conference on Information Fusion (FUSION). 2018; 1-8.
[53] Ouyang Z, Cui J, Dong X, Li Y, Niu J. SaccadeFork: A lightweight multi-sensor fusion-based target detector. Information Fusion. 2022; 77: 172-183.
[54] Geng K, Dong G, Yin G, Hu J. Deep dual-modal traffic objects instance segmentation method using camera and LIDAR data for autonomous driving. Remote Sensing. 2020; 12(20).
[55] Liu Z, Farrell J, Wandell BA. ISETAuto: Detecting vehicles with depth and radiance information. IEEE Access. 2021; 9: 41799-41808.
[56] Islam MM, Newaz AAR, Karimoddini A. A pedestrian detection and tracking framework for autonomous cars: Efficient fusion of camera and liDAR data. in: 2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC). 2021; 1287-1292.
[57] Liu R, Lehman J, Molino P, Petroski Such F, Frank E, Sergeev A, et al. An intriguing failing of convolutional neural networks and the coordconv solution. in: Advances in Neural Information Processing Systems. 2018.
[58] Pan SJ, Yang Q. A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering. 2010; 22(10): 1345-1359.
[59] Chen K, Wang J, Pang J, Cao Y, Xiong Y, Li X, et al. MMDetection: Open MMLab detection toolbox and benchmark. CoRR. 2019; abs/1906.07155.
[60] Carranza-García M. Multi-modal fusion for 2D object detection in autonomous driving. 2022. (Accessed 28 March 2022). https://github.com/carranza96/waymo-detection-fusion.
[61] He K, Girshick R, Dollar P. Rethinking imageNet pre-training. Proceedings of the IEEE International Conference on Computer Vision. 2019; 4917-4926.
[62] Shivakumar SS, Nguyen T, Miller ID, Chen SW, Kumar V, Taylor CJ. DFuseNet: Deep fusion of RGB and sparse depth information for image guided dense depth completion. in: 2019 IEEE Intelligent Transportation Systems Conference (ITSC). 2019; 13-20.
[63] Chodosh N, Wang CY, Lucey S. Deep convolutional compressed sensing for liDAR depth completion. in: Asian Conference on Computer Vision (ACCV). 2018.
[64] Ahmadlou M, Adeli H. Enhanced probabilistic neural network with local decision circles: A robust classifier. Integr Comput-Aided Eng. 2010; 17(3): 197-210.
[65] Alam KMR, Siddique N, Adeli H. A dynamic ensemble learning algorithm for neural networks. Neural Comput Appl. 2020; 32(12): 8675-8690.