Object Recognition for Autonomous Vehicles from Combined Color and LiDAR Data

Lian Kang and Pierre Payeur
University of Ottawa, Ottawa, Ontario, CANADA

Abstract

In recent years, autonomous driving vehicles have garnered substantial attention in both the commercial and scientific domains. A key challenge faced by these vehicles is the accurate detection and recognition of objects within complex real-world road environments, essential for their real-time decision-making capabilities. While color imaging has traditionally provided rich information, the utilization of LiDAR scanners presents advantages such as high-quality data collection under varying lighting conditions and the provision of precise spatial information with an extensive range. By combining data from color cameras and LiDAR scanners, the potential for object detection in autonomous driving is expanded, opening up new avenues for advancement. This paper introduces a novel 3D object detector that leverages a bird's-eye view map generated from a LiDAR point cloud along with RGB images as input data. It employs focal loss and Euler angle regression techniques to enhance object detection performance. Through ablation experiments, the achieved improvements are evaluated. Experimental results demonstrate the framerate and performance of the proposed 3D object detector, surpassing 46 frames per second and achieving an average precision exceeding 90%. Additionally, a more compact version of the detector is introduced, processing the same input data three times faster while maintaining reasonably high accuracy.

Key Words: Object recognition, deep learning, LiDAR, autonomous vehicles.

1 Introduction

The advent of artificial intelligence has propelled autonomous driving vehicles into the realm of possibility, garnering substantial investments from major players like Tesla and Waymo. Detecting and recognizing pedestrians, vehicles, and other objects on the road is a crucial task in autonomous driving systems, ensuring safe and informed driving decisions. As a result, there has been a significant focus on developing efficient object detection and recognition technologies.

Deep learning methods have revolutionized the field of object detection and recognition, demonstrating remarkable advancements. Recent research has emphasized the inclusion of depth information in detection models, surpassing the limitations of traditional 2D mapping approaches for autonomous driving. Light detection and ranging (LiDAR) scanners, unlike stereo cameras and active depth sensors, offer consistent and high-precision spatial information unaffected by ambient lighting conditions. Consequently, LiDAR scanners have gained popularity for 3D object detection in outdoor environments. However, while LiDAR scanners provide shape and location information of objects in the real world through point cloud data, they lack texture and color information. To fully exploit 3D location with color and texture information, the point cloud from LiDAR scanners often needs to be preprocessed and combined with RGB images.

This paper represents an extended version of [6] providing further insights into the methodology employed, along with conducting comprehensive experimental ablation studies and robustness validation. The aim is to delve deeper into the intricacies of the proposed approach and to evaluate its performance under various scenarios and conditions. By conducting these additional analyses, a more comprehensive understanding of the methodology's strengths, limitations, and overall effectiveness can be achieved. The key original contributions of this research include the integration of Euler angle regression into the DarkNet-53 convolutional neural network [14] to create a 3D object detector capable of classifying and localizing cars, pedestrians, and cyclists using LiDAR point clouds and RGB images from real-world road scenes. To optimize training and testing efficiency, the LiDAR point cloud is transformed into a bird's-eye view (BEV) map using coordinate system transformation and height thresholding. Furthermore, the proposed architecture incorporates a focal loss [10] and a generalized intersection over union (GIoU) loss [15] to address biased data and to enhance the model's performance. The solution is trained and evaluated on real-world data provided by the KITTI vision benchmark suite [4], demonstrating its object recognition capabilities.

2 Literature Review on 3D Object Detection from LiDAR Data

In recent years, significant advancements in artificial intelligence have been made in the field of 3D object detection and recognition for autonomous driving. To fully support 3D object detection, which involves an important component of localization in the environment, the utilization of not only traditional RGB or grayscale images but also depth information

* Email: lkang018@uottawa.ca, ppayeur@uottawa.ca.

is required as it provides the spatial coordinates of each pixel. Consequently, the need for larger and more complex training and testing datasets arises to build performing deep learning models, posing higher demands on data processing and computing capabilities.

Two-stage detectors, such as MV3D-Net [3] and AVOD [7], have gained prominence in this field. MV3D-Net incorporates both RGB images and LiDAR point clouds as input. It combines a 3D object proposal network and a region-based fusion network to efficiently generate 3D candidate boxes from bird's-eye view (BEV) maps and front-view perspectives derived from the point cloud. AVOD shares similarities with MV3D-Net but utilizes a feature pyramid network (FPN) instead of a VGG16 based network for feature extraction. FPN helps maintain feature map resolution and preserves both low-level and high-level information, leading to enhanced detection accuracy, particularly for small objects.

Single-stage detectors, such as PIXOR [23] and PointPillars [8], have also made notable contributions to 3D object detection. PIXOR discretizes the point cloud by equally spaced units and encodes reflectivity to create a regular representation. It employs a fully convolutional network (FCN) to estimate the position and heading angle of the target relative to the sensor. PIXOR achieves a high detection framerate of over 28 frames per second (FPS). On the other hand, PointPillars directly aggregates points falling into each grid, forming 'Pillars'. The learned feature vector is then mapped back to grid coordinates, resembling an image-like representation.

In addition to these methodologies, researchers have integrated the Transformer architecture [19] into 3D detection algorithms. VoTr [11] employs a voxel-based Transformer as a 3D backbone network for object detection from point cloud data. It utilizes self-attention mechanisms to establish long-range relationships between voxels. To improve attention span without excessive computational overhead, VoTr introduces local attention, dilated attention, and fast voxel query. The versatility of the Transformer architecture is demonstrated through variants such as VoTr-SSD and VoTr-TSD [11], which leverage SSD and R-CNN backbones, respectively.

In summary, notable progress has been made in 3D object detection and recognition for autonomous driving applications through various approaches. Two-stage detectors, such as MV3D-Net and AVOD, offer efficient fusion of RGB images and LiDAR point clouds, while single-stage detectors like PIXOR and PointPillars provide rapid detection capabilities. Additionally, the incorporation of the Transformer architecture, exemplified by VoTr, shows promise in capturing long-range dependencies in point cloud data.

3 Point Cloud Preprocessing and Registration with RGB Data

A 3D point cloud is the default representation of the data collected by a LiDAR scanner. The LiDAR scanner used to collect point clouds for the KITTI dataset [4] and considered in this research is the Velodyne's HDL-64E. It is a 64-channel multi-beam mechanical LiDAR that continuously rotates the head to achieve dynamic 3D scanning. It covers a 360° horizontal and 26.9° vertical field of view [4]. Although the data provided by LiDAR reports an accurate 3D location, it does not contain color information. Therefore, in this work, both the color images provided by a collocated RGB camera, and the 3D point clouds provided by the LiDAR are used.

There are two major ways to process a 3D point cloud: one is directly processing a 3D matrix, while an alternative approach involves projecting the 3D map to its corresponding 2D representation. In recent work using LiDAR point clouds for object detection and recognition [5, 9, 26], researchers directly train the detector on the 3D point clouds [18] with the consequence that convolution operations are more time and memory intensive compared to when information is encoded in 2D. Alternatively, some models like MV3D-Net [3] opt for converting the point cloud to a front view and a BEV map, which is more efficient than processing the 3D point cloud directly, while not lowering the accuracy.

The detector proposed in this work uses BEV maps that are initially encoded as 2-channel bidimensional grids where each pixel of a 2D map contains respectively a 'cumulated height' parameter associated with 3D points mapped to the pixel, and a 'cumulated intensity' estimate provided by the LiDAR sensor along with every 3D point measurement. The BEV maps converted from point cloud data collected by the LiDAR scanner are further combined with registered front forward view images collected by the RGB camera as inputs. This section details how the LiDAR point clouds are preprocessed and converted to the BEV maps, as shown in Figure 1.

3.1 Data Registration

The BEV map is a graphical representation of the point clouds from a bird's-eye view. It is obtained by projecting the discrete LiDAR point cloud on a plane perpendicular to the height direction. Therefore, a BEV map forms an equivalent representation of the 3D location information contained in the LiDAR point cloud, hence it is more efficient when a large amount of data is being processed. The front forward view and color information is provided from corresponding RGB images. Therefore, all the required information can be obtained by combining the BEV maps converted from LiDAR point clouds and the RGB images.

The 3D point cloud and RGB images are obtained from different sensors and must be registered before the two sets of data are used together as input. With the help of the calibration matrices from the dataset, the registration of the LiDAR point clouds and RGB images is performed using matrix transformation to align LiDAR coordinate axes and origin to the RGB coordinates.

3.2 Mapping 3D Points within Region of Interest to 2D Pixels

In the dataset considered, the point cloud of each scene represents approximately 1.9 MB, which requires significant memory space, highly increases computation for both training and testing, and reduces detection efficiency. Therefore,

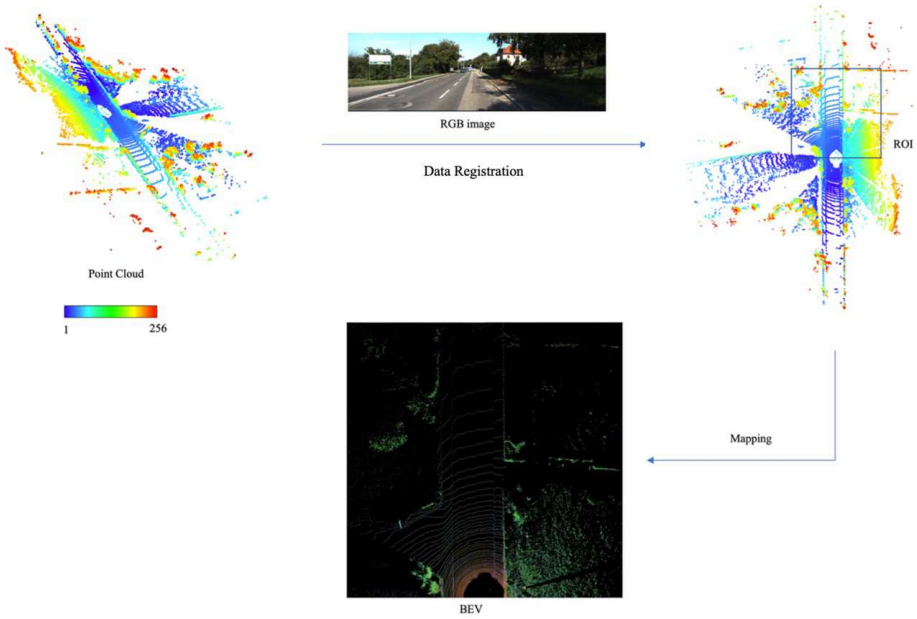

Figure 1: 3D point cloud preprocessing and registration with RGB image.

detection is focused over a pre-selected region of interest (ROI) in the point cloud. To balance the model's efficiency while covering all the annotated target objects in the corresponding RGB image, the ROI of the point cloud is manually set as a rectangular area that spans 40m on either side of the LiDAR scanner, and 80m in front of it. The point cloud data collected from the LiDAR scanner are 3D points with real (x, y, z) values that carry depth information. The registered points within the ROI carrying real number values are then mapped into integer values (u, v) that represent the pixel location on the discretized bird's-eye view (BEV) map.

3.3 Recording the Height and Intensity Information in the BEV Map

Once the coordinates (u, v) that represent each pixel in the BEV map are obtained, the height information represented by the values on the Z-axis and the intensity information represented by a 4th parameter contained in the source point cloud matrix are extracted from the 3D point cloud to be encoded in the corresponding BEV map.

Inspired by the representation adopted in PIXOR [23], a vertical ROI on the Z-axis is selected to support a height thresholding operation that is applied to preserve only data from the point cloud that are within the selected height of the ROI. Next, the height coordinates (on Z-axis) within the ROI are rescaled to the]0, 255[range, and the height coordinates exceeding the ROI are forced to 0 or 255. Finally, the height values of the points in the point cloud that are mapped into the same 2D pixel position on the BEV map are cumulated and recorded to the "cumulated height" channel of the BEV map. Compared with using the maximum height of each pixel position [1], the cumulation method appears to be less affected by changes in the elevation of the objects due to the characteristics of the environment, such as the slope of the road. Unlike MV3D-Net [3] that manually selects multiple ranges on the Z-axis and accumulate the values of the points within the ranges to generate multiple height channels for each 2D point in the BEV map, the proposed method performs a single height thresholding operation to create one height channel. This contributes to improve the efficiency of the detection process.

Similarly, the intensity information already contained in the point cloud as the 4th value for each 3D point is extracted. For all 3D points contained within the selected height ROI and falling within the same 2D pixel position on the BEV map, these intensity values are accumulated and recorded to the 'cumulated intensity' channel of the BEV map. The resulting preprocessed point cloud data leads to a 2-channel bidimensional BEV map which is used as part of the input for the proposed 3D object detector along with the corresponding 3-channel RGB image input.

4 3D Object Detector Architecture

The proposed method for 3D object detection combines the BEV map generated from a LiDAR point cloud and the associated RGB image information to form a single 5-channel (height, intensity, R, G, B) input for every 2D pixel coordinate in the BEV map, as depicted in Figure 2a. The BEV map part of the input represents the bird's-eye view over the detection range, with a cumulated height channel and cumulated intensity channel, as detailed in Section 3. The RGB image is subsampled and padded with $[R, G, B] = [128, 128, 128]$ to match the size of the BEV map, as shown in Figure 2a. It forms the RGB component of the input, which represents the front forward view as found in autonomous driving, with three

different color channels (R, G, B). Doing so preserves both the 3D information collected by the LiDAR scanner in the distribution of feature points in the 2D BEV map and the color information collected by the RGB camera.

The output of the proposed model represents the detection and recognition confidence over three object classes (car, pedestrian, cyclist), with prediction matrices at three different scales. The prediction matrices are used to draw bounding boxes (B-Box) around detected objects and to label them with their respective classification. Given the importance of making fast decisions in autonomous driving, any improvement in object detection speed while maximizing object detection and classification accuracy is prioritized. For this reason, a single-stage detection model is proposed in this paper.

As shown in Figure 2, the backbone of the proposed object detection model is based on DarkNet-53 [14], modified to include the preprocessing stage of the BEV maps and the RGB images described in Section 3. The detection head is based on the YOLOv3 anchor regression method, modified by adding BEV variables and rotation angle regression. For the latter, the rotation angle encoding uses the Euler representation, as inspired by complex YOLO [17].

4.1 Detection Head with Euler Angle Regression

Through the backbone convolutional neural networks and the FPN layers feature maps at three different scales are extracted from the input. As shown in Figure 2d, a detection head is used to generate the detection and recognition results based on these feature maps.

Within the detection head, each feature map is divided into grid cells. For each grid cell there are three anchors at different scales. These anchors represent priors for bounding boxes (B-Box). They are equivalent to a reference frame for the predicted B-Box. Based on this reference, the predicted B-Box generated by the detection head only needs to be fine-tuned with respect to the corresponding anchor. As a result, for every anchor, there is a prediction matrix that contains the parameters used for regression during training and the detection result. The output prediction matrix of the proposed detection head contains the B-Box prediction matrix for both the RGB front forward view and the BEV, a confidence score, and classification scores over the 3 object classes considered.

To adapt to the different perspectives of the predicted input, based on Yolov3 [14], the B-Box prediction matrix for the proposed detection head is modified. It is divided into two parts: one for the front forward view, another one for the BEV, as shown in Figure 3. The prediction matrix contains 14 parameters (i.e., N = 14 in Figure 2d). The confidence score p_0 indicates the confidence that the predicted B-Box contains an object. If this predicted B-Box corresponds to the background, then this value should be 0. The classification scores p_1, p_2, p_3 represent the probability that the category of the predicted B-Box corresponds to 'car', 'pedestrian', or 'cyclist' respectively. For the final output, only the B-Box with p_0 higher than a detection threshold will be kept and the classification shows $\max(p_1, p_2, p_3)$.

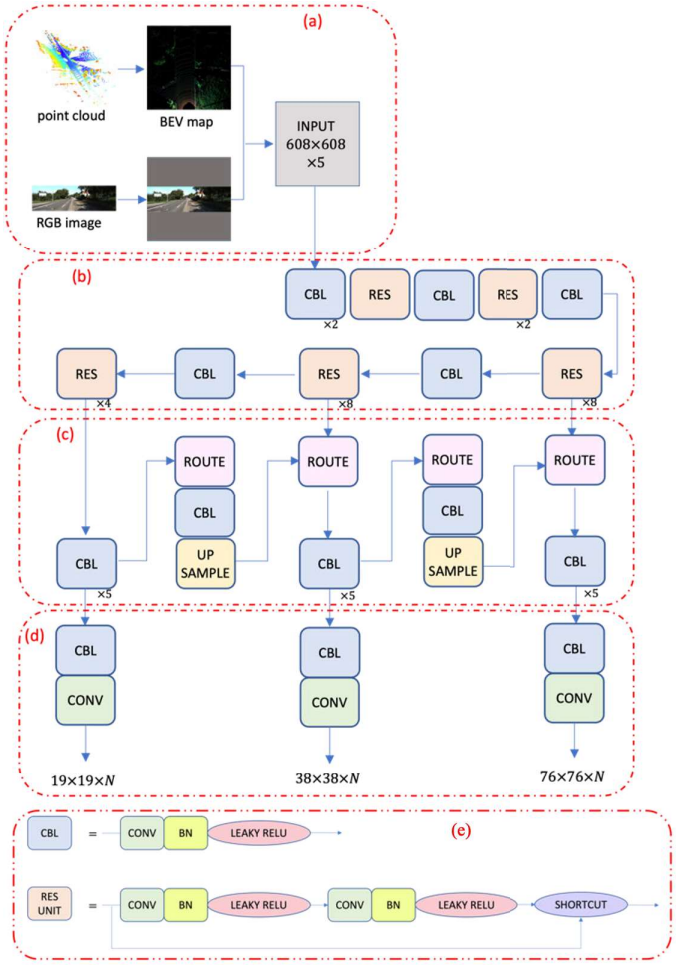

Figure 2: Proposed 3D object detection model architecture: (a) preprocessing stage to convert a 3D point cloud and corresponding remapped RGB image into a 5-channel 2D BEV map; (b) backbone of the proposed model (DarkNet-53); (c) feature pyramid network (FPN); and (d) detection head with outputs prediction matrices at three different scales; with (e) details of the respective structure of CBL (top) and res unit (bottom)

Considering that objects of primary interest in the context of autonomous driving, such as cars, pedestrians, and cyclists can generally be assumed to stand or move on the ground, the front forward view bounding box (B-Box) surrounding a detected object on the RGB image can be represented by 4 parameters. These are the center coordinates, (t_{cx}, t_{cy}), and the width and height, (t_{cw}, t_{ch}), of the B-Box as shown in Figure 3a.

Unlike the B-Box on RGB images, the BEV B-Box might not be parallel to the BEV map's coordinate axes, as exemplified in Figure 3b. To predict the relative rotation of the B-Box, as inspired by complex YOLO [17], a Euler representation of the rotation angle is added to the prediction matrix in the proposed detection model. Hence, the BEV prediction matrix obtained from the regression of the proposed model contains 6 variables.

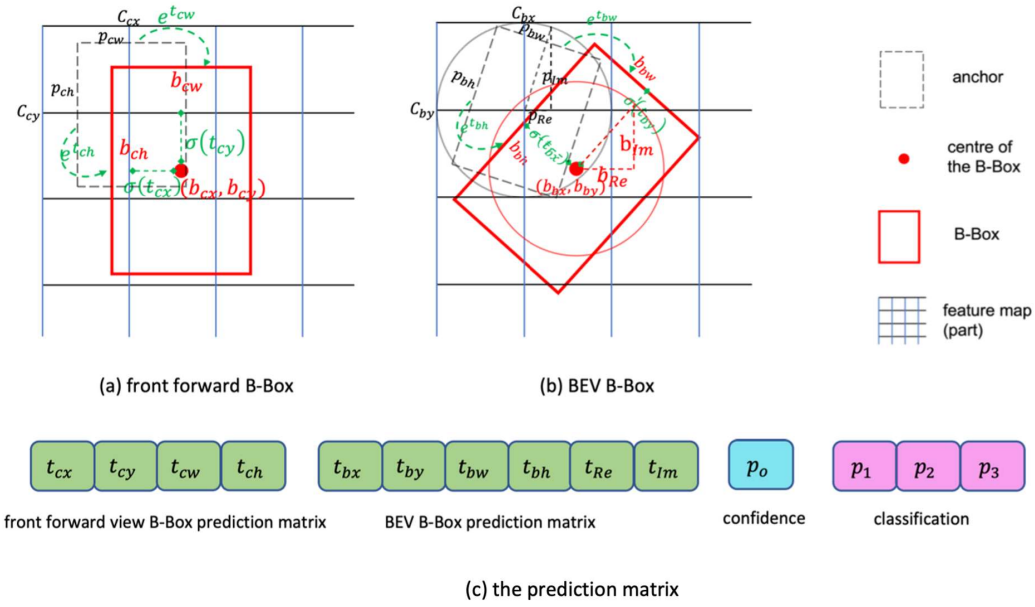

Figure 3: Converting the prediction matrix into B-Box: (a) front forward view B-Box; (b) BEV B-Box; and (c) prediction matrix of the detector with 14 parameters

Aside from the offsets of the B-Box center coordinate, (t_{bx}, t_{by}), and its width and height (t_{bw}, t_{bh}), the Euler representation of the rotation angle of the B-Box, (t_{Im}, t_{Re}), is also encoded in the 6 parameters.

4.2 Loss Functions

The loss function used in the proposed detector model consists of a combination of a classification loss, a B-Box regression loss, and a confidence loss. Compared to YOLOv3 [14], the regression loss uses Generalized Intersection over Union (GIoU) [15] instead of mean square error (MSE). To further optimize the performance of the detector, the Euler angle is added to the B-Box regression loss. For the classification loss, a focal loss [10] is added to address the imbalance problem observed in the KITTI vision benchmark suite [4] training dataset for the considered three classes.

4.2.1 Regression Loss. To match with the Euler angle regression network, a combination of GIoU [15] and Euler angle regression is used for the B-Box regression. The GIoU of the predicted B-Box and ground truth B-Box is computed as:

$$GIoU = \frac{I}{B^g \cup B^p} - \frac{A^c - (B^g \cup B^p)}{A^c} \quad (1)$$

Where B^g, B^p are the ground truth B-Box and the predicted B-Box respectively. I is the intersection of the ground truth and predicted B-Boxes, and $B^g \cup B^p$ is the union of the two B-Boxes. A^c represents the smallest convex shape that encloses both B^g and B^p. $A^c - (B^g \cup B^p)$ represents the area that is inside A^c but outside $(B^g \cup B^p)$. The GIoU loss for the front forward RGB view is represented by:

$$L_{GIoU} = 1 - GIoU \quad (2)$$

Since the B-Box prediction matrix contains 4 variables for the front forward view and 6 variables for the BEV, the B-Box regression loss is divided into two parts: GIoU loss (L_{GIoU}) for the front forward view prediction matrix, and an Euler defined GIoU loss (L^E_{GIoU}) for the BEV prediction matrix which is also computed with Equation (2) but on the BEV view.

4.2.2 Classification Loss. In YOLOv3 [14], cross entropy loss [25] is used for classification. In ideal circumstances, a non-biased training dataset helps the model learn the features for multi-class object detection and recognition, and cross entropy loss would be suitable. However, among the three classes considered in this research (car, pedestrian, cyclist), the training data provided by KITTI [4] contains 82% of the total objects in the class 'car', 13% in the 'pedestrian' class, and less than 5% in the 'cyclist' class. This represents a significant bias toward the 'car' category which must be addressed to achieve fair comparative results. Inspired by [10], a focal loss is used to substitute the cross entropy loss. This strategy represents a first usage of focal loss on BEV maps to the best of our knowledge. The classification loss is defined as:

$$L^F_{cla} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathrm{I}_{i,j}^{obj} \sum_{c \in \text{classes}} [\hat{p}_c (1 - p_c)^{\gamma} \log(p_c) + (1 - \hat{p}_c) \hat{p}_c{}^{\gamma} \log(1 - p_c)] \quad (3)$$

$$I_{i,j}^{obj} = \begin{cases} 1, & \text{if object} \in \text{the } j^{\text{th}} \text{ anchor} \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

Where γ is the relaxation parameter. The higher value of γ, the more 'focus' will be given to misclassified examples, and the less loss will be propagated from examples. S^2 represents the number of grid cells, which is equal to the size of the feature map. In the proposed model with three different scales, S^2 has sizes 19×19, 38×38, 76×76, respectively. B represents the B-Box. $I_{i,j}^{obj}$ is a binary value that indicates whether the j^{th} B-Box of the i^{th} grid cell's GIoU value is larger than the GIoU threshold. p_c and $\hat{p}_c$ are the ground truth and the prediction classification score for class c.

4.2.3 Confidence Loss. The confidence loss is used to measure the objectiveness of the B-Box. The proposed model uses focal loss as its confidence loss, defined as follows:

$$L_{con} = \sum_{i=0}^{S^2}\sum_{j=0}^{B} I_{ij}^{obj}\left[\hat{C}_i(1-C_i)^{\gamma}\log(C_i) + \left(1-\hat{C}_i\right)\hat{C}_i^{\gamma}\log(1-C_i)\right] + \lambda_{noobj}\sum_{i=0}^{S^2}\sum_{j=0}^{B} I_{ij}^{noobj}\left[\hat{C}_i(1-C_i)^{\gamma}\log(C_i) + \left(1-\hat{C}_i\right)\hat{C}_i^{\gamma}\log(1-C_i)\right] \quad (5)$$

$$I_{i,j}^{obj} = \begin{cases} 1, & \text{if object} \in \text{the } j^{\text{th}} \text{ anchor} \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

$$I_{i,j}^{noobj} = \begin{cases} 1, & \text{if the } j^{\text{th}} \text{ anchor is background} \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

$$\hat{C}_i = \hat{p}_i(c) \times (GIoU + GIoU_E) \quad (8)$$

$$C_i = \begin{cases} 1, & \text{if object} \in \text{the } j^{\text{th}} \text{ anchor} \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

If an object is detected in the B-Box, the confidence loss is $\sum_{i=0}^{S^2}\sum_{j=0}^{B} I_{ij}^{obj}\left[\hat{C}_i(1-C_i)^{\gamma}\log(C_i) + \left(1-\hat{C}_i\right)\hat{C}_i^{\gamma}\log(1-C_i)\right]$. $\hat{C}_i$ is the confidence score of the j^{th} prediction B-box in i^{th} grid cell, and C_i is the ground truth, that is whether the B-Box contains an object.

In realistic scenarios, most bounding boxes do not contain any object. This causes an imbalance problem where the background or negative samples are more frequently detected by the model than the objects of some positive samples. To alleviate this issue, the confidence loss is weighted down by a factor λ_{noobj}, which intervenes when no object is detected in the box (detected background only). In such a case, the confidence loss is $\lambda_{noobj}\sum_{i=0}^{S^2}\sum_{j=0}^{B} I_{ij}^{noobj}\left[\hat{C}_i(1-C_i)^{\gamma}\log(C_i) + \left(1-\hat{C}_i\right)\hat{C}_i^{\gamma}\log(1-C_i)\right]$, where I_{ij}^{noobj} is the complement of the binary value I_{ij}^{obj}, and λ_{noobj} weighs the loss down.

In summary, the loss function of the proposed 3D object detector combines the two GIoU regression losses (L_{GIoU} applied on the front forward view and the Euler angle loss L_{GIoU}^{E} applied on the BEV view), the focal classification loss (L_{cla}^{F}), and the confidence loss (L_{con}). The combination of all components leads to the general loss function:

$$L = \alpha_1 L_{cla}^{F} + \alpha_2 L_{GIoU} + \alpha_3 L_{GIoU}^{E} + \alpha_4 L_{con} \quad (10)$$

Where α_1, α_2, α_3, α_4 are the weights for each component of the loss function, which are empirically determined based on experimental results.

5 Experimental Results

5.1 Performance Evaluation

The performance of the proposed architecture is compared with other popular 3D object detectors that also use the KITTI LiDAR data as input. Table 1 summarizes the mean average precision (mAP) performance and framerate achieved while considering three difficulty levels for the object detection process, as defined in the KITTI evaluation metrics. The category 'easy' represents cases where low objects occlusion occurs and with objects' B-Box height reaching over 40 pixels in the front forward RGB image. 'Moderate' cases involve objects that are at least partially visible and taller than 25 pixels. Finally, 'hard' cases correspond to significantly occluded objects that are difficult to observe in images.

In terms of the detector framerate, the experimental evaluation demonstrates that the proposed model reaches 46.4 frames per second (FPS) in an implementation using a single NVIDIA Tesla V100 GPU. This is more than 3 times faster than the transformer-based detector VoTr-SSD, 3 to 4 times faster than two-stage detectors with similar detection accuracy, and faster than PointPillars by over 4 FPS with similar or exceeding accuracy.

The proposed detector also demonstrates superior precision performance compared to other models listed in Table 1, including the recently introduced transformer architectures. In this case, it outperforms the dominant two-stage F-ConvNet detector by a significant margin of over 3%. Performance gains are particularly visible in cases categorized as 'easy', as illustrated in the corresponding results shown in Figure 4. For visualization purposes, 3D B-Boxes are plotted around the detected objects over the testing RGB images (top part of results), and the corresponding 2D BEV B-Boxes are plotted over the BEV maps (lower part of results). The color coding of the B-Boxes represents the class of the detected objects: yellow for 'car', red for 'pedestrian', and blue for 'cyclist'.

Compared with other detectors that use both the LiDAR point cloud and RGB images, such as MV3D-Net [3], AVOD [7], PIXOR [23], MMF [9], F-PointNet [12] and F-ConvNet [20], the proposed detector demonstrates higher mAP on 'moderate' and 'hard' samples. As shown in Table 1, some models that use only a LiDAR point cloud as input reach slightly higher mAP on 'hard' cases compared with models that combine LiDAR point clouds with RGB images as input. Sample experimental results achieved with the proposed LiDAR+RGB single-stage detector are presented for 'moderate' and 'hard' cases in Figures 5 and 6, respectively.

Table 1: Comparison of performance among 3D object detectors using the KITTI LiDAR dataset

	Method	Data	Framerate (FPS)	mAP (%)		
				Easy	Moderate	Hard
Transformer	VoTr-SSD [11]	LiDAR	14.7	87.86	78.27	76.93
	VoTr-TSD [11]	LiDAR	7.2	89.04	84.04	78.68
Two Stages	MV3D-Net [3]	LiDAR + RGB	2.7	86.49	78.98	72.23
	AVOD [7]	LiDAR + RGB	10.0	89.74	84.81	78.12
	F-PointNet [12]	LiDAR + RGB	5.7	91.16	84.61	74.77
	F-ConvNet [20]	LiDAR + RGB	1.9	**91.44**	85.84	76.11
	Fast Point R-CNN [2]	LiDAR	15.3	90.87	**87.71**	80.51
	MMF [9]	LiDAR + RGB	12.2	86.81	76.75	68.41
	STD [24]	LiDAR	10.0	89.93	86.20	79.42
Single Stage	VoxelNet [26]	LiDAR	4.2	87.95	78.39	71.29
	SECOND [22]	LiDAR	19.7	89.33	82.87	78.51
	PointPillars [8]	LiDAR	**41.9**	90.07	86.56	**82.81**
	SA-SSD [5]	LiDAR	24.4	88.75	79.79	74.16
	PIXOR [23]	LiDAR + RGB	28.6	86.78	80.75	76.77
	Proposed detector	***LiDAR + RGB***	***46.4***	***94.71***	***87.33***	***81.52***

Globally, when examining performance over all classes and independently from the 'easy', 'moderate', or 'hard' categorization of sample test cases, the mAP over all object classes reaches 90.26%, with the average precision (AP) for each specific class corresponding to 97.94% for 'car', 82.72% for 'pedestrian', and 90.13% for 'cyclist' respectively. Statistical details about the performance achieved are detailed in Table 2 when considering 1500 pairs of the LiDAR point cloud and the corresponding RGB images including all three classes of objects considered.

5.2 Ablation Studies with Different Loss Functions

The proposed detector merges Euler angle regression to the DarkNet-53 backbone to improve the detection accuracy and uses Euler angle regression loss and GIoU loss to optimize the training. To reduce the bias caused by the imbalance in the number of samples in each class of the dataset, focal loss [10] is also used as the classification loss and the confidence loss. To evaluate if these methods improve the performance of the proposed model, ablation studies are designed to test the performance of different regression loss and to verify that the proposed Euler angle regression does contribute to increase the detection accuracy of the proposed model. The results of the ablation experiments are listed in Table 3 where various combinations of loss functions are considered as the components of Equation (10).

The ablation studies with different regression loss show that the consideration of Euler angle regression increases the detection accuracy of the proposed object detector. Compared to MSE, GIoU loss also shows better performance on the regression of the proposed model. Finally, when comparing with the use of cross entropy as the classification loss, focal loss significantly increases the average precision (AP) for the 'pedestrian' and 'cyclist' classes although it slightly decreases AP for 'car'. Hence, it is concluded that focal loss does contribute to better balance the AP over all considered classes.

5.3 Masked BEV Map

In another set of experiments conducted as part of this research, a mask in the form of an empty rectangle was superimposed over the bird's-eye view map to selectively exclude certain information from the detector's input. This investigation aimed to determine whether the proposed model can handle damaged data. The experiments demonstrate that the proposed detector remains functional even when presented with damaged data, with only a marginal decrease in mean average precision (mAP) of less than 5%. Our observations revealed that in most cases if a target object is completely hidden or missing in the BEV map data, the proposed detector fails to detect the object reliably, as illustrated in Figure 7. This indicates that the detector cannot operate with reasonable accuracy solely based on the RGB image input. Conversely, if a randomly positioned mask partially occludes an object, as depicted in Figure 8, it generally does not significantly affect the detection and recognition outcome, as long as partial information about the target object remains available in the BEV map. Finally, when the mask covers a background area without occluding any target object of interest, as shown in Figure 9, the detection result remains unaffected.

6 Mini 3D Object Detector

As part of the continuous development of deep convolutional neural networks and aiming at always pursuing higher accuracy,

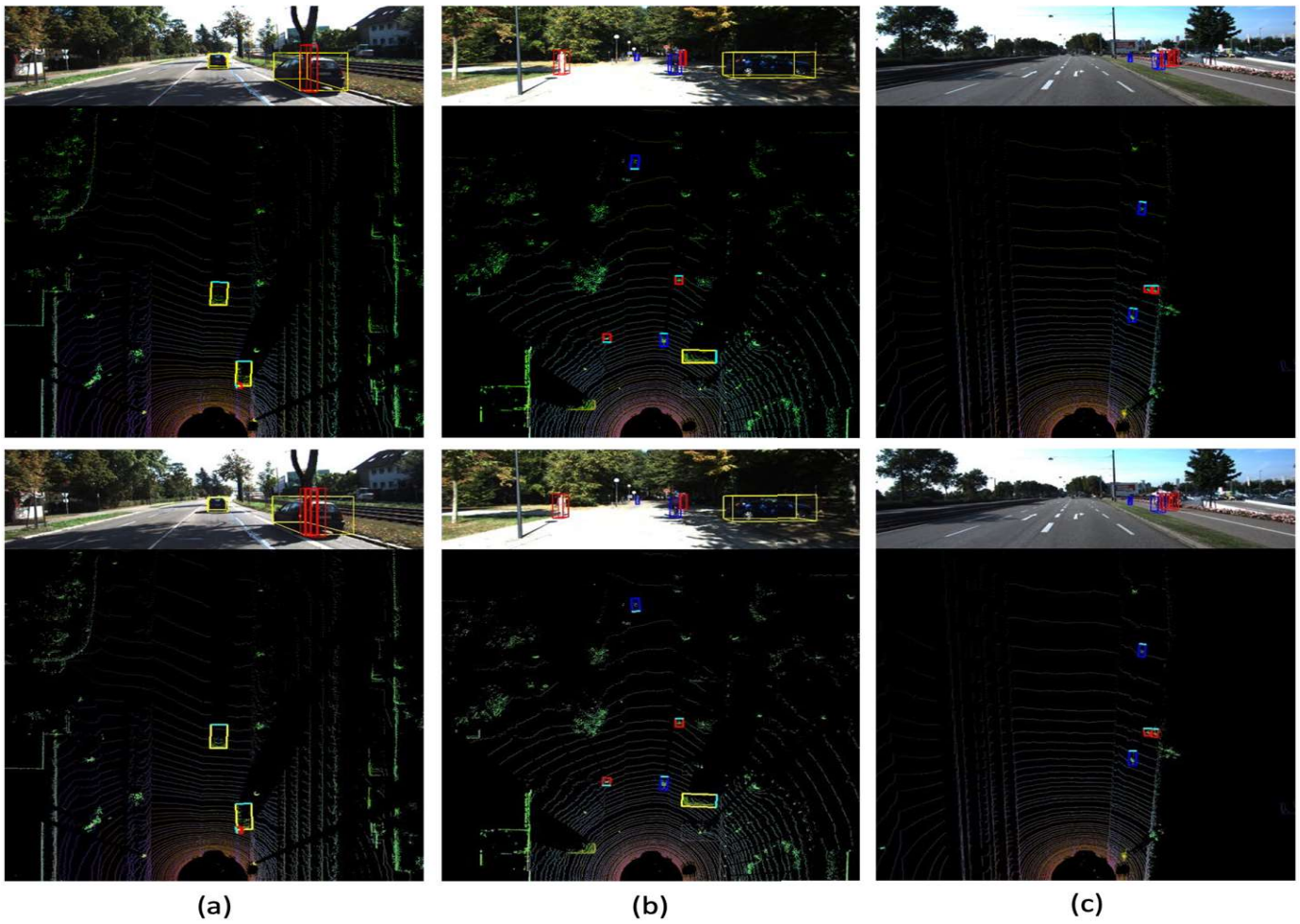

Figure 4: Samples of 'easy' scenes in the KITTI testing dataset. In each case (a,b,c), the upper row shows ground truth bounding boxes, and the bottom row shows detection results achieved with the proposed detector [yellow = car, red = pedestrian, blue = cyclist]

researchers are motivated to propose various strategies to increase object detection framerate, especially in speed sensitive context such as autonomous vehicles navigation. Ideally, a detector should be compact from both the memory requirement and amount of calculation perspectives, mainly because of the availability of limited hardware resources that can be embedded on autonomous platforms. These constraints motivate the development of deep convolutional neural architectures well adapted for widespread deployment on embedded devices. Therefore, although the framerate of the original detector introduced in Section 4 reaches up to 46.4 FPS, we still wish to explore the design of a lightweight network that involves fewer feature matrices to perform even faster on the same 3D object detection and recognition tasks.

6.1 Mini Detector Architecture

The proposed mini detector merges the structure of tiny-YOLO [16] and the proposed detector from Section 4 to generate feature maps at 2 different scales, as shown in Figure 10. The main difference of the mini detector compared to its full-size version is the backbone and FPN. The mini detector uses a DarkNet-19 [13] based backbone, modified to adapt to the input of the BEV map and corresponding RGB image. Compared to the 53-layers backbone of the original detector, the mini detector's backbone only has 19 layers, that is about 1/3 the depth. Moreover, with the mini detector implementation, only two different scales of feature maps are generated and passed to the detection head, compared to three in the initial version, to further reduce the calculation load and minimize the depth of the mini detector model. The detection head then converts the reduced size feature maps into prediction result. Otherwise, the detection head uses the same design as in the proposed full-size model and the same combined loss function, Equation (10), for training.

Being more compact, the mini detector's framerate can reach

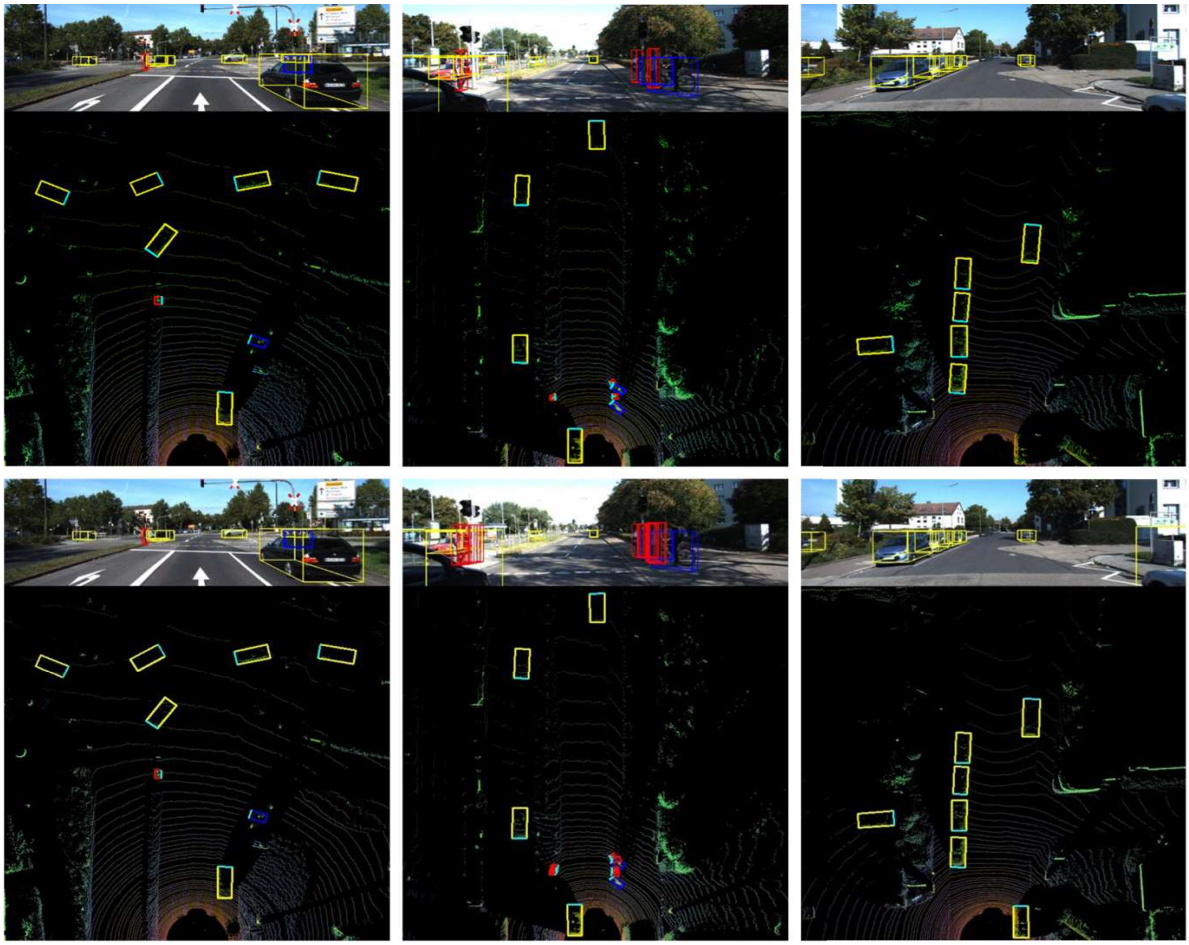

Figure 5: Samples of 'moderate' scenes in the KITTI testing dataset. In each case (a,b,c), the upper row shows ground truth bounding boxes, and the bottom row shows detection results achieved with the proposed detector [yellow = car, red = pedestrian, blue = cyclist]

up to over 3 times that of the original full-size detector. Therefore, it is better suited for real-time autonomous driving applications running on mobile devices, while offering a satisfactory compromise on detection performance.

6.2 Experimental Results with the Mini Detector

For a fair comparison of performance with the two proposed detectors, the mini detector is trained and tested on the same software and hardware environment as the original full-size detector presented in Section 4. Moreover, the training and testing dataset remains the same. Table 4 presents the detection and recognition results of both the mini detector and the full-size detector on 1500 pairs of the LiDAR point cloud and the corresponding RGB images.

As shown in Table 4, the mini detector achieves a good performance on detecting 'cars' with AP higher than 0.97, but lower AP is observed on detecting the 'pedestrian' and 'cyclist' targets. This is explained by the fact that the training dataset used for both proposed models is imbalanced, with less than 20% of positive samples exemplifying the pedestrian and cyclist classes. Although FPN and focal loss [10] are used to reduce the impact of the data imbalance, with fewer layers and less features extracted in the mini detector model, the testing performance is more severely impacted by the data imbalance than with the full-size detector. Figures 11 and 12 visually compare the performance against ground truth labels of both versions of the detector by displaying the predicted bounding boxes over the front forward RGB image and corresponding BEV map for detected objects belonging to the three considered classes.

Conversely, the framerate of the mini detector reaches up to 158.97 frames per second, which is 3.4 times faster than the full-size detector when testing in the same environment, while a 7.5% reduction of the mAP is observed overall on all three combined classes. Comparing with alternative compact implementations of objects detectors, as shown in Table 5, the proposed mini detector exhibits relatively high detection

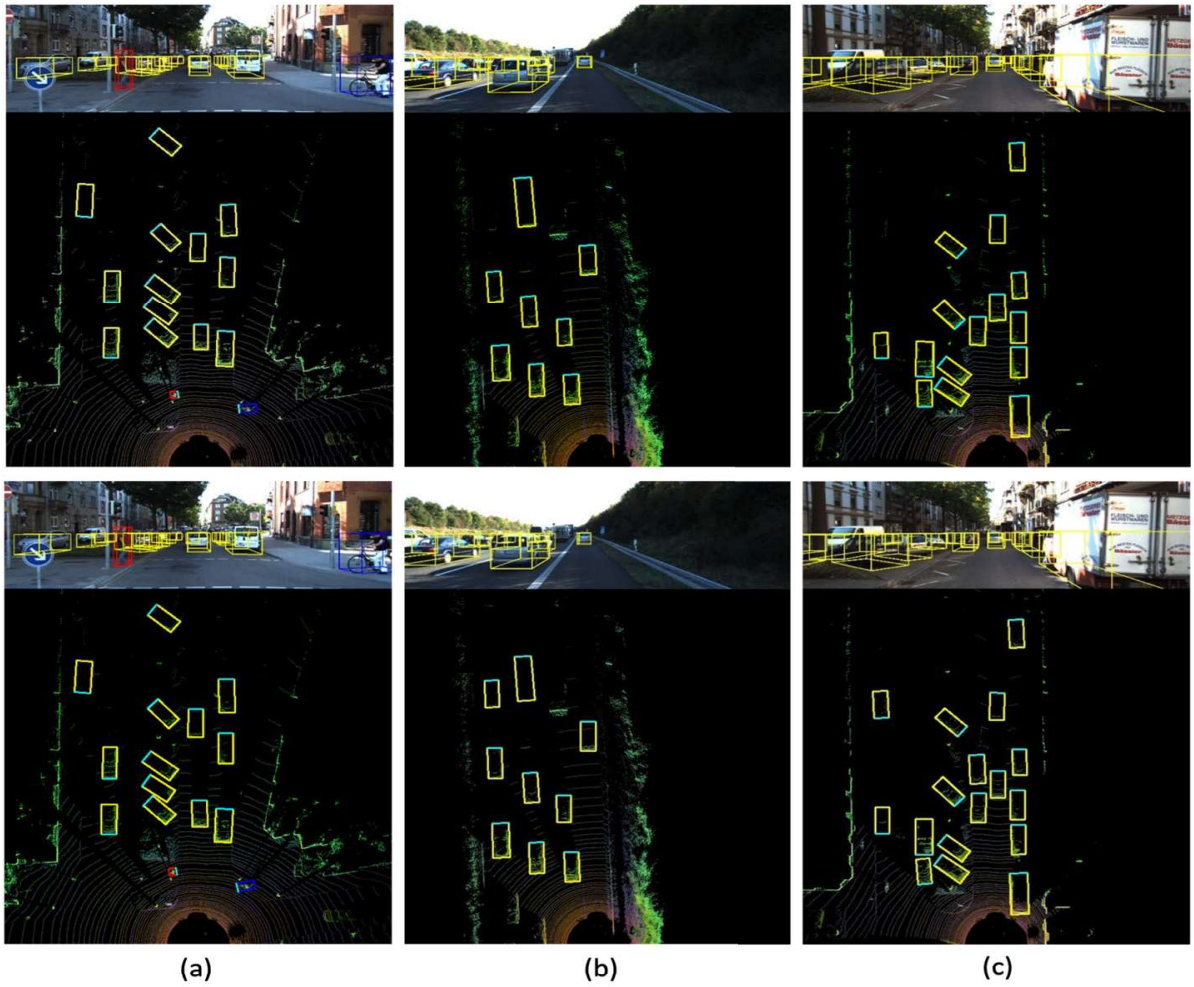

Figure 6: Samples of 'hard' scenes in the KITTI testing dataset. In each case (a,b,c), the upper row shows ground truth bounding boxes, and the bottom row shows detection results achieved with the proposed detector [yellow = car, red = pedestrian, blue = cyclist]

Table 2: Object detection performance on the KITTI LiDAR dataset

Class	Precision	Recall	AP	F1	mAP (%)	Framerate (FPS)
Car	90.65	98.68	97.94	94.50		
Pedestrian	63.89	93.17	82.72	75.80	**90.26**	**46.4**
Cyclist	79.51	95.24	90.13	86.67		

Table 3: Effect of different regression loss on the detection results

Classification Loss	Regression Loss	Confidence Loss	AP			mAP (%)
			Car	Pedestrian	Cyclist	
focal	MSE	focal	95.11	53.93	62.58	70.56
focal	GIoU	focal	96.78	64.61	83.83	81.74
focal	MSE + Euler	focal	96.88	78.48	90.96	88.77
focal	GIoU + Euler	focal	97.94	**82.72**	**90.13**	**90.26**
cross entropy	GIoU + Euler	focal	**98.03**	79.91	88.35	88.76

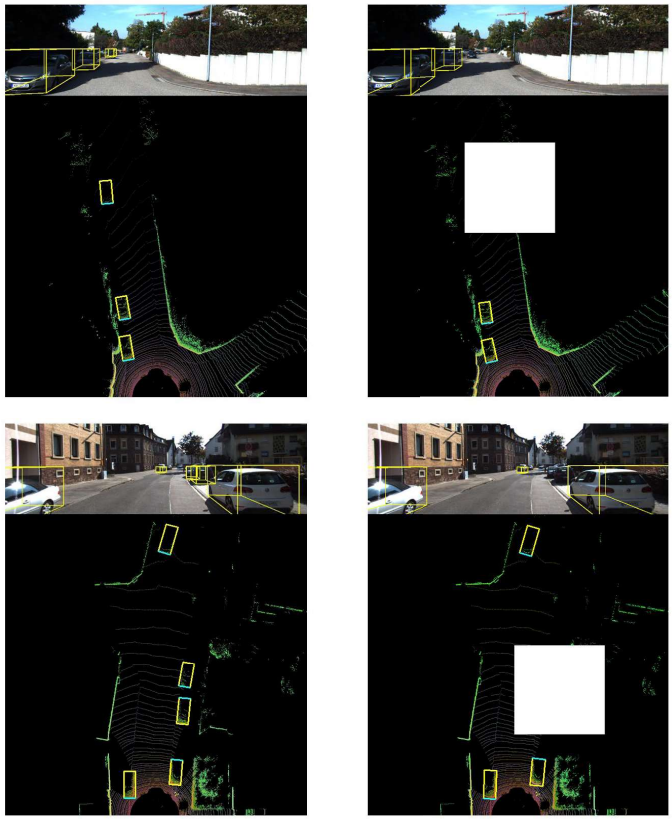

Figure 7: Objects detected on sample test cases with fully masked objects in the BEV map (left: without mask; right: with occluding mask)

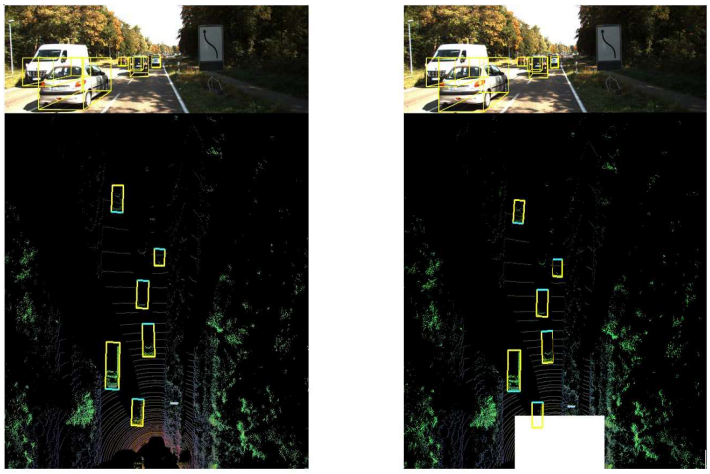

Figure 8: Objects detected on sample test case with partially masked object (left: without mask; right: with occluding mask)

accuracy when compared to tiny YOLO [16] and tiny SSD [21], which are designed for 2D image-based only object detection. While expanding the architecture to benefit from 3D information issued from LiDAR data, the proposed mini detector remains competitive with the performance of similar scale detectors reported in the literature. The mini detector also achieves significantly higher detection framerate.

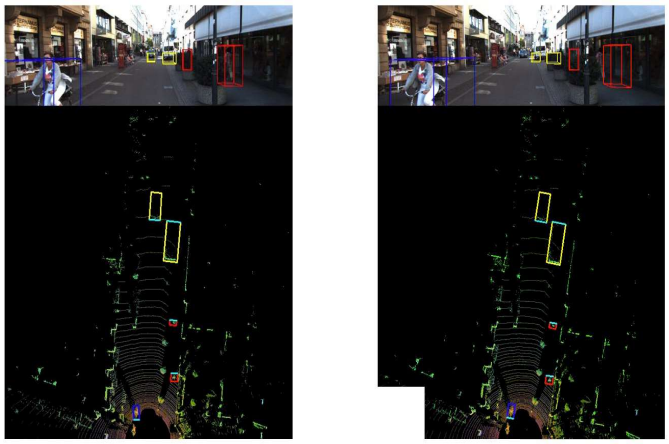

Figure 9: Objects detected on sample test case with mask over the background only (left: without mask; right: with occluding mask)

7 Conclusion and Future Work

This paper proposes two original formulations for 3D object detectors that leverage 3-dimensional location information from a LiDAR point cloud in combination with RGB images. With the objective to optimize the computation and memory usage of the detection models, a preprocessing step is performed to convert the point cloud data into a bird's-eye view (BEV) map. The latter emphasizes the height range of interest through height thresholding, while intensity values from the LiDAR sensor are accumulated and recorded on the corresponding pixel positions. When combined with color information from a registered frontal view RGB image, the process leads to a 5-dimensional BEV map that serves as input to the detectors.

The design of the proposed full-size detector model combines the GIoU loss and DarkNet-53 architecture from the YOLOv3 single-stage detector. Additionally, Euler angle orientation is incorporated into the detection head and an original formulation for a combined loss function is introduced. Experimental results reveal that the integration of Euler angle regression and GIoU losses enhances the performance of the proposed detector compared to the original YOLOv3 model, which utilizes MSE regression loss.

To address the bias in detection results caused by imbalanced training data, focal loss is employed as the classification loss. Ablation studies demonstrate that focal loss partially compensates for data imbalance in the proposed models and improves the mean average precision (mAP) across different classes. Furthermore, experiments using masked BEV maps showcase the robustness of the proposed model to degraded sensor inputs. Overall, the proposed full-size model achieves a

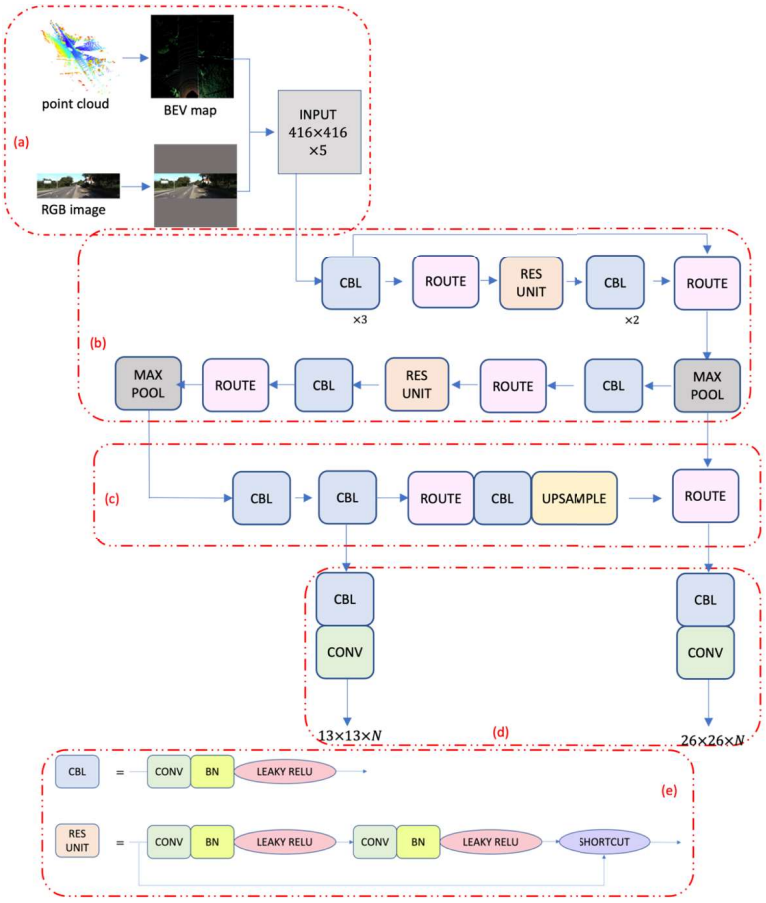

Figure 10: Proposed 3D object mini-detector architecture: (a) preprocessing converts point cloud and rescaled RGB image into a 5-channel 2D BEV map; (b) reduced backbone of proposed mini detector (DarkNet-19); (c) mini FPN; and (d) detection head with output prediction at 2 different scales; with (e) details of the respective structure of CBL (top) and Res Unit (bottom)

Table 4: Detection framerate, precision, recall, AP and F1 estimated on each class and overall mAP for the proposed mini detector compared with the full-size detector

	Class	Mini detector	Full-size detector
Framerate (FPS)		**158.97**	**46.4**
Precision	Car	89.22	90.65
	Pedestrian	47.83	63.89
	Cyclist	68.74	79.51
Recall	Car	95.72	98.68
	Pedestrian	62.31	93.17
	Cyclist	87.72	95.24
AP	Car	93.71	97.94
	Pedestrian	69.08	82.72
	Cyclist	85.44	90.13
F1	Car	92.47	94.50
	Pedestrian	38.38	75.80
	Cyclist	78.32	86.67
mAP (%)		**82.74**	**90.26**

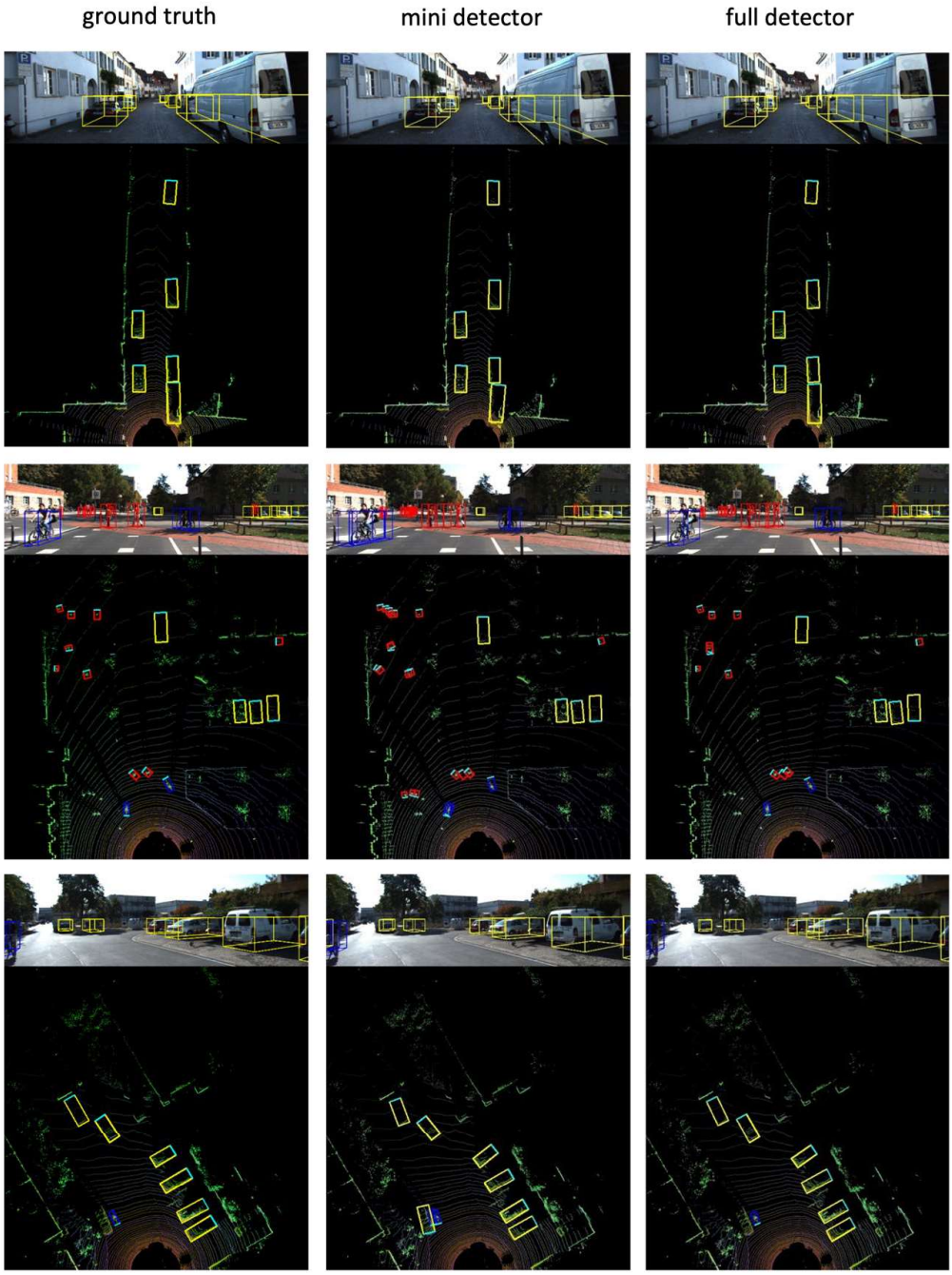

Figure 11: Three sample test cases comparing the ground truth (left), with results of the mini detector (center), and results of the full-size detector (right), with B-Boxes [yellow = car; red = pedestrian; blue = cyclist] superimposed over front forward RGB image (upper part) and over corresponding BEV map (lower part)

Table 5: Comparison of performance between lightweight detection models

Model	Size	Nb of trained parameters	Framerate (FPS)	mAP (%)
Tiny YOLO [16]	60.5 MB	-	133	57.1
Tiny SSD [21]	2.3 MB	1.13 M	-	61.3
Proposed mini detector	*44.9 MB*	*7.19 M*	***158.97***	***82.7***

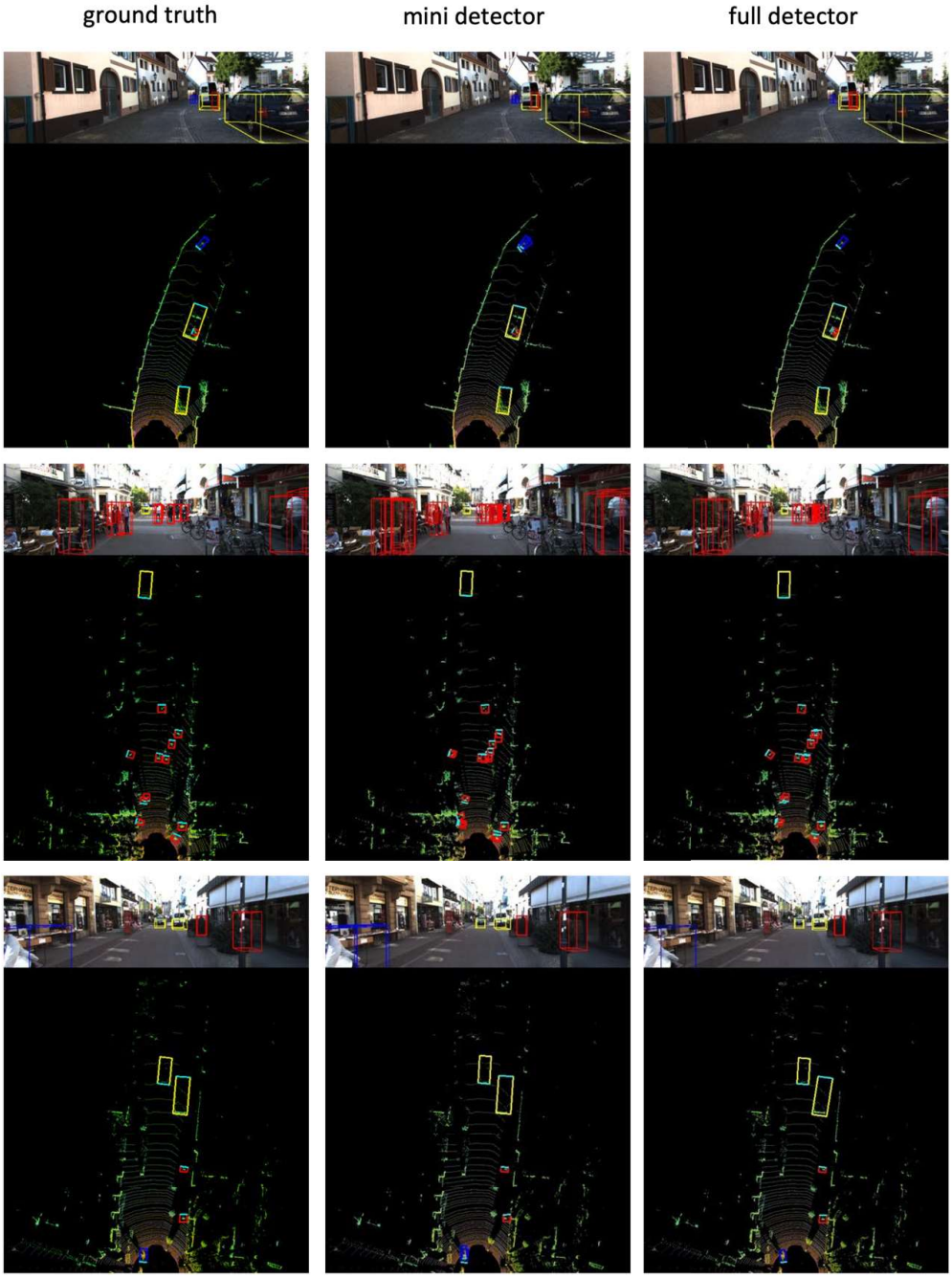

Figure 12: Additional sample cases comparing the ground truth (left), with results of the mini detector (center), and results of the full-size detector (right), with B-Boxes [yellow = car; red = pedestrian; blue = cyclist] superimposed over front forward RGB image (upper part) and over corresponding BEV map (lower part)

detection framerate of up to 46.4 frames per second (FPS) in a single GPU-based implementation with mAP exceeding 90%. Experimental results indicate that the model adapts well to real-life autonomous driving scenarios with varying levels of occlusions.

To explore faster and lighter detection models suitable for real-time and embedded vehicle applications, a mini detector is also introduced. By integrating a lightweight deep learning detector into the 3D data processing domain and leveraging key concepts from the full-size detector, a compact 19-layer network model is developed for 3D object detection and recognition, achieving mAP above 82%. Experiments demonstrate that

compared to the full-size detector, the mini detector requires approximately 2/3 of the training time and 1/3 of the testing time. This trade-off between processing time and accuracy allows for effective performance in time-critical applications. The proposed mini detector also shows superior performance in terms of both detection accuracy and framerate compared to other lightweight 3D detectors.

While this research brings significant contributions to 3D object recognition, some limitations remain and open areas for future research. First, it will be beneficial to develop a self-optimizing training model that can automatically adjust training parameters and feature maps to improve the generalization ability of the detectors. This will allow to adapt better to different operational conditions, such as occlusion, low resolution, and varying scene complexity. Second, a sensor-independent fusion framework is essential to ensure the safety of autonomous vehicles. Further research is needed to explore the signal coupling issues that may arise when fusing LiDAR scanner and camera inputs, especially in safety-critical environments.

Moreover, addressing the imbalance in the training dataset is crucial for improving object recognition accuracy. While we employed focal loss to reduce bias, there is still room for improvement, particularly in detecting cyclists and pedestrians, which are as important as detecting cars in autonomous driving scenarios. Future research will investigate methods to make the model less sensitive to the number of training samples or adjust the training dataset to improve balance in the number of samples from different classes.

References

[1] J. Beltrán, C. Guindel, F. M. Moreno, D. Cruzado, F. Garcia, and A. De La Escalera, "Birdnet: A 3D Object Detection Framework from Lidar Information," 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 2018.

[2] Y. Chen, S. Liu, X. Shen, and J. Jia, "Fast Point R-CNN," *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9775-9784, 2019.

[3] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-View 3D Object Detection Network for Autonomous Driving," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1907-1915, 2017.

[4] A. Geiger, P. Lenz, and R. Urtasun, "Are We Ready for Autonomous Driving? The Kitti Vision Benchmark Suite," *Proceedings of 2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3354-3361, 2012.

[5] C. He, H. Zeng, J. Huang, X.-S. Hua, and L. Zhang, "Structure Aware Single-Stage 3D Object Detection from Point Cloud," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11873-11882, 2020.

[6] L. Kang and P. Payeur, "3D Objects Detection and Recognition from Color and LiDAR Data for Autonomous Driving," *Proceedings of 38th International Conference on Computers and Their Applications,* 91:42-55, 2023.

[7] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3D Proposal Generation and Object Detection from View Aggregation," *Proceedings of 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1-8, 2018.

[8] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast Encoders for Object Detection from Point Clouds," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12697-12705, 2019.

[9] M. Liang, B. Yang, Y. Chen, R. Hu, and R. Urtasun, "Multi-Task Multi-Sensor Fusion for 3D Object Detection," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7345-7353, 2019.

[10] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2980-2988, 2017.

[11] J. Mao, Y. Xue, M. Niu, H. Bai, J. Feng, X. Liang, H. Xu, and C. Xu, "Voxel Transformer for 3D Object Detection," *Proceedings of the IEEE/CVF International Conference on Computer Vision,* pp. 3164-3173, 2021.

[12] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum Pointnets for 3D Object Detection from RGB-D Data," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 918-927, 2018.

[13] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7263-7271, 2017.

[14] J. Redmon and A. Farhadi, "Yolov3: An Incremental Improvement," *arXiv preprint arXiv:1804.02767,* 2018.

[15] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, "Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression," *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 658-666, 2019.

[16] K. C. Saranya, A. Thangavelu, A. Chidambaram, S. Arumugam, and S. Govindraj, "Cyclist Detection Using Tiny Yolo v2," *Soft Computing for Problem Solving*, 1057:969-979, 2020.

[17] M. Simon, S. Milz, K. Amende, and H.-M. Gross, "Complex-Yolo: An Euler-Region-Proposal for Real-Time 3D Object Detection on Point Clouds," *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pp. 197-209, 2018.

[18] V. A. Sindagi, Y. Zhou, and O. Tuzel, "Mvx-Net: Multimodal Voxelnet for 3D Object Detection," *2019* International Conference on Robotics and Automation (ICRA), 2019.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is All You Need," *Advances in Neural Information Processing Systems,* 30:5998-6008, 2017.

[20] Z. Wang and K. Jia, "Frustum Convnet: Sliding Frustums

to Aggregate Local Point-Wise Features for Amodal 3D Object Detection," 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 1742-1749, 2019.
[21] A. Womg, M. J. Shafiee, F. Li, and B. Chwyl, "Tiny SSD: A Tiny Single-Shot Detection Deep Convolutional Neural Network for Real-Time Embedded Object Detection," 15th Conference on Computer and Robot Vision (CRV), pp. 95-101, 2018.
[22] Y. Yan, Y. Mao, and B. Li, "Second: Sparsely Embedded Convolutional Detection," *Sensors,* 18:3377, 2018.
[23] B. Yang, W. Luo, and R. Urtasun, "Pixor: 2017 Real-Time 3D Object Detection from Point Clouds," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7652-7660, 2018.
[24] Z. Yang, Y. Sun, S. Liu, X. Shen, and J. Jia, "Std: Sparse-to-Dense 3D Object Detector for Point Cloud," *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1951-1960, 2019.
[25] Z. Zhang and M. Sabuncu, "Generalized Cross Entropy Loss for Training Deep Neural Networks with Noisy Labels," Advances in Neural Information Processing Systems, 2018.
[26] Y. Zhou and O. Tuzel, "Voxelnet: End-to-End Learning for Point Cloud Based 3D Object Detection," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4490-4499, 2018.

Lian Kang (photo not available) is a software engineer specializing in software development and artificial intelligence. She holds a Master's degree in Electrical and Computer Engineering from the University of Ottawa, with a focus on machine learning algorithms for computer vision tasks. Her research published in CATA 2023 and IJCA has significantly contributed to advancements in the field of object detection. With a strong background in software engineering and expertise in machine learning, Lian is passionate about leveraging AI technologies to solve real-world problems and drive innovation. Her dedication to cutting-edge research and practical applications makes her a valuable contributor to the field of artificial intelligence.

Pierre Payeur (photo not available) received the Ph.D. degree in Electrical Engineering from Université Laval, Canada. Since 1998, he is a Professor at the School of Electrical Engineering and Computer Science, University of Ottawa. He is the Director of the Sensing and Machine Vision for Automation and Robotic Intelligence research laboratory and a co-founder of the Vision, Imaging, Video Processing and Autonomous Systems research laboratory. He also served in various academic leadership roles while being extensively involved in industrial and international collaborations. His research interests include machine vision, 3D modeling, tactile sensing, automation, manipulator and mobile robotics, and computational intelligence for man-machine interfaces.