

Article

Not peer-reviewed version

---

# A Novel Method to Generate Auto-Labeled Datasets for 3D Vehicle Identification Using a New Contrast Model

---

[Guillermo S Gutierrez-Cabello](#) , [Edgar Talavera](#) \* , [Miguel Clavijo](#) , [Guillermo Iglesias](#) , [Felipe Jimenez](#)

Posted Date: 27 February 2023

doi: [10.20944/preprints202302.0438.v1](https://doi.org/10.20944/preprints202302.0438.v1)

Keywords: Auto-labeled; LiDAR, Point of View, Deep Learning



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

## Article

# A Novel Method to Generate Auto-Labeled Datasets for 3D Vehicle Identification Using a New Contrast Model

Guillermo S. Gutierrez-Cabello <sup>1,2</sup>, Edgar Talavera <sup>2,\*</sup>, Guillermo Iglesias <sup>2</sup>, Miguel Clavijo <sup>1</sup> and Felipe Jiménez <sup>1</sup>

<sup>1</sup> University Institute for Automobile Research (INSIA), Universidad Politécnica de Madrid, Spain

<sup>2</sup> Departamento de Sistemas Informáticos, Escuela Técnica Superior de Ingeniería de Sistemas Informáticos, Universidad Politécnica de Madrid, Spain

\* Correspondence: e.talavera@upm.es

**Abstract:** Auto-labeling is one of the main challenges in 3D vehicle detection. Auto-labeled datasets can be used to identify objects in LiDAR data, which is a challenging task due to the large size of the dataset. In this work, we propose a novel methodology to generate new 3D based auto-labeling datasets with a different point of view setup than the one used in the most recognized datasets (KITTI, WAYMO, etc.). The performance of the methodology has been further demonstrated with the development of our own dataset with the auto-generated labels and tested under boundary conditions on a bridge in a fixed position. The proposed methodology is based on the YOLO model trained with the KITTI dataset. From a camera-LiDAR sensory fusion, it is intended to auto-label new datasets while maintaining the consistency of the Ground Truth. The main contribution of this work is a novel methodology to auto-label autonomous driving datasets using YOLO as the main labelling system. The performance of this approach is measured retraining the contrast models of the KITTI benchmark.

**Keywords:** Auto-labeled; LiDAR, Point of View, Deep Learning

---

## 1. Introduction

Deep Learning (DL) algorithms in Autonomous Vehicles (AV) have evolved in recent years and 2D-based object detection has received a lot of attention due to the accurate performance achieved by models such as YOLO [1]. One of the reasons behind the performance of object detection models is the presence of large-scale labeled datasets such as COCO [2] with more than 220K labeled images of a total of 330k with 1.5M object instances, 80 object categories and 91 stuff categories.

However, 3D sensors, such as light detection and ranging (LiDAR), have attracted the interest of researchers, but due to the lack of labeled data, 3D-based object detection models have not yet reached the same performance as 2D-based ones. One of the leading datasets for 3D vehicle detection is KITTI [3], but its main weakness is its relatively small size, which contains around 15.000 images, which cannot be enough to train certain DL models. Recently, another 3D-labeled perception dataset has been published as WAYMO [4], with 2,030 video segments yielding a total of 390k frames. Because many frames correspond to the same vehicle in the same video, the WAYMO dataset lacks in its variability. In addition, both datasets have been developed in a very similar way, using images and point clouds taken from a vehicle with multiple sensors. This point of view, in addition to being the most used in AV systems, is relatively specific and cannot be used in other tasks that need different angles of vision.

DL models need large volumes of data for good performance [5], the availability of the data can be a critical factor in the results of DL algorithms. The datasets mentioned above have been labeled with much less information than the 2D-based ones and have also been presented with a fixed point of view. However, both datasets have been routinely selected by researchers due to the cost involved in annotating new 3D labels. The challenges facing the AV industry are widely different with different set-ups and configurations of the sensors, e.g. using cameras and LiDAR placed in the middle of

a roundabout or on a bridge that points over the road. As these datasets are not enough for new scenarios and the tedious and costly work that has to be done to traditionally build new datasets, a new approach, auto-labeled datasets, has to be adopted.

Instead of manually labeling 2D data, commercial tools or works based on [Convolutional Neural Networks \(CNNs\)](#) have been proposed to generate auto-labeled datasets [6–8]. As [CNNs](#) achieve contrasting results in 2D object detection, one of the main ideas that has been proposed is to auto-label datasets. There are works such as [6,9] where [LiDAR](#) point clouds are projected on the input images obtaining as output the object bounding box coordinates and the class as a 3D label, this way achieving 3D auto-generated labels. Despite the auto-labeling process in both works, the datasets have been developed with the same point of view as KITTI and WAYMO, so the bias in the datasets is still a problem. Nevertheless, as far as we are concerned, it is worth researching whether there is a way to use the models present in the literature to generate datasets with a new approach.

Before diving into the idea of creating a novel methodology to generate different auto-labeled datasets, it has been proposed to use [DL](#) model designs to detect objects over images like YOLO to generate new labels. Depending on the occlusion level of the object, sets 3 levels of difficulty, and the accuracy of the research, taken as F-score, for each difficulty is 0.957 when easy, 0.927 when moderate and 0.740 when hard.

These new auto-generated labels are the ones proposed to train models with new scenarios. But to conclude that the new labels work with new approaches, proven models have to be taken to analyze the performance using the new labels. Therefore, we have proposed to retrain the contrast models presented in the KITTI benchmark [10,11] such as [12–14] to be retrain with the new autogenerated labels. The performance obtained, although lower than that achieved with manually taken labels, is still reliable, with a small drop in accuracy expected compared to that achieved with manually taken labels. So, manually taken labels can be replaced by auto-generated ones and the models can continue to perform well.

Due to the minor deviation obtained with the proposed models, the slightly reduced accuracy is balanced by the enormous amount of data that can be generated at low cost. Therefore, the main contribution of this work is a novel methodology proposed to generate new 3D based auto-labeled datasets from boundary conditions and it has been further demonstrated with the development of our own 3D auto-labeled dataset with the point of view of sensors located at an elevated point above an infrastructure, in our case, a bridge over a highway.

One of the challenges facing researchers in the automotive field is to find datasets that fit a specific scenario. Traditionally, most of the work has been developed under the same conditions, i.e. they usually consist of a vehicle with different sensors placed in similar positions such as on the roof of the vehicle or on the hood, generating datasets with the same viewpoint. The main problem associated with this is that it is not possible to use these data for applications with different perspectives, thus requiring further research. Therefore, models have been developed that can automatically generate labeled datasets, avoiding the tedious task of manually labeling new data each time a system is developed.

Although there are datasets that have been manually labeled as KITTI [3], WAYMO [4], or nuScenes [15], the trend is evolving towards labeling automation. There are works such as [16] in which a not complete auto-labeling process is proposed. Annotators identify a bounding box in a [LiDAR](#) point cloud in a particular frame and then, using tracking techniques, automatically propagate it to the rest of the video frames. This technique does not completely reduce human workload, but it considerably increases the size of the labeled dataset at a low cost.

To achieve fully auto-labeled datasets, one of the types of models that can best handle this type of data is [CNNs](#). There are several examples such as [17], where GAN networks based on [CNNs](#) are used to perform data augmentation on a labeled dataset generating new labeled images from scratch. One of the main limitations of this approach is that it cannot work with point clouds, making it impossible to use this approximation in [LiDAR](#) based detection.

In [6] the YOLO architecture is also used, in this case re-trained with the KITTI dataset. In [7] a combination of complexYOLO [18] with PointRCNN [12] is proposed for the detection of objects in LiDAR data. The other approximation is followed in [19] where the SSD network is used to project the bounding box of a corresponding camera image onto the LiDAR point clouds. However, in spite of all the literature, there are already several applications that can handle the process of automatic label images, such as LabelBox, Image Labeler, RectLabel, etc.

Most of the previous auto-labeling models lack in their generalization, being only capable of work under certain circumstances, e.g. the point of view of a camera in the front of a car. Previous work such as [20] used traditional computer vision techniques and LiDAR point clouds to auto-label new datasets with different camera angles. This research is focused on using transfer learning techniques on models trained with KITTI, to use these new models to auto-label new datasets using different camera angles.

## 2. Methodology

In this paper, we propose a framework capable of tackling the following two problems: generate 3D based auto-labeled datasets for [AV](#) tasks and verify the auto-labeling process by evaluating the performance of the datasets regardless of the point of view in which the sensors, such as cameras or [LiDAR](#), are placed to generate the dataset.

To achieve our goal and propose a methodology, we first used the KITTI dataset as our base workbench, as it is a dataset used in multiple previous [AV](#) papers like [12–14] in which [DL](#) models applied to 3D object detection are presented on the point cloud generated by lasers. However, although the aforementioned works are recognized as the models that offer the best results with KITTI, in particular [7,8], these works use radar sensors as input to carry out the detection work. There are other classification works that allow vehicle detection through techniques such as object segmentation and by using transfer learning techniques [16,21,22]. Finally, other works such as [6,17], try to implement data augmentation techniques through GAN models applied to point cloud datasets.

Furthermore, the KITTI dataset is published and can be publicly accessed. Thus, we will apply the 2D auto-labeling model to the raw KITTI dataset to project the bounding box of the images into point clouds, generating new automatic labels. Then the results of the auto-label dataset will be compared against the human KITTI annotated dataset. These results, measured with the F-Score, will show the difference between the human annotated labels and the auto-labels produced in our system.

Furthermore, we will compare the performance of [Artificial Neural Network \(ANN\)](#) models by training some models from OpenMMLab's OpenPCDet open-source project [23] with the new auto-labeled images. OpenPCDet is a general codebase for 3D object detection from point cloud and the way in which the models have been designed to work is by projecting the labels of the 2D images onto their corresponding 3D point cloud. In addition, these models have been shown to work effectively with KITTI and these will be used as a comparison for the performance of the models. In order to evaluate the performance of the proposed auto-label, we will train again from scratch the mentioned models with the new datasets. Moreover, as the main purpose is to demonstrate the good performance of the models with the auto-generated labels rather than to propose improvements such as a greater capacity to detect more vehicle classes, all tests have been performed only attending to the car class because it is sufficient to obtain clear results of the performance of the methodology.

Therefore, since the KITTI dataset is widely the most used in the field of [AV](#) object detection as it has been public since 2012 rather than Waymo, which has been available since 2020, KITTI will be used to develop a 3D auto-label methodology, which will then serve to label new datasets automatically. Thus, the presented framework demonstrates and tests an auto-labeling workflow using a well-documented dataset. This method, which demonstrates the possibility of exchanging annotator labels for auto-generated ones, can be applied to any dataset, including Waymo.

## 2.1. YOLO

The idea of auto-labeling the dataset has been proposed in some other works such as [6,7]. It has been shown that [CNNs](#) works well for that purpose when it comes to 2D labeling: Some of the most popular models used in traffic object detection are: [ssd](#) [19], [Yolo](#) [1] or [ComplexYolo](#) [18]. In particular, for our purposes, we have selected the last YOLO release, YOLOv5, as its [CNN](#) backbone for automatic 2D labeling of KITTI data.

YOLOv5 has been selected as the model to perform 2D labeling due to the fact that it is one of the models published in the literature in the object detection task, as shown by Ultralytics, comparing it with the literature benchmarks [24] as can be shown in Figure 1. YOLOv5 is published as a pre-trained model and can be used to predict where objects are located in 2D images, as well as to classify them. We use this information to project the generated bounding boxes onto the 3D point clouds and automatically build the necessary labels for our target by generating 3D labels.

## 2.2. KITTI

KITTI dataset is known to have been widely used in 3D [AV](#) research, but recently new datasets have appeared, such as [nusCense](#) or [Waymo](#), which are also a very common choice for the latest work. However, despite the emerging datasets, KITTI has been chosen for the measurement of the auto-labeling task because all the published models that will be used have been previously trained with KITTI, and we will be applying the auto-labeling task to the raw KITTI dataset. This will facilitate the comparison of the results between the proposed methodology and previous work.

KITTI is made up of 7,481 training images and 7,518 test images, as well as their corresponding point clouds, covering a total of 80,256 labeled objects. In addition, depending on the occlusion of the object, each label has the difficulty of viewing the object as easy, moderate and hard. The main idea is to take the raw data as a YOLOv5 input and use the generated bounding boxes as labels and to measure the YOLO labeling performance. The KITTI dataset labeled by annotators has been used as ground truth for comparison purposes.

## 2.3. Validation models

It is considered necessary that the models to be used in this methodology have been previously tested using KITTI. Thus, we have selected some models from OpenPCDet, in particular it was decided to test our methodology with PointPillar [13], PointRCNN [12], Voxel R-CNN [14] and PV-RCNN [25]. The main idea is to compare the performance of each model when using KITTI labels with respect to the labels generated by YOLOv5.

To measure the accuracy of each model as proposed in [26] it is possible to perform BEV or 3D object detection. BEV is a bird's eye view representation of all features captured from a camera perspective [27,28]. This method of representing transformed images offers some advantages, such as the lack of scale ambiguity and the absence of near occlusions due to its overhanging perspective. However, BEV tends to be sparse, which implies a non-efficient application of CNNs [13]. Furthermore, 3D detection and orientation estimation consist of three parts: First, 2D object detection measured in [Average Precision \(AP\)](#) is performed. Subsequently, [Average Orientation Similarity \(AOS\)](#) is performed to measure the 3D orientation estimation. And finally, pure classification and regression are evaluated [26]. The performance of the KITTI dataset, reported in each paper of each selected model, is shown in Table 2.

Regardless of the fact that PointPillars of the selected models is the only model that in the paper has been published the performance measured in BEV. This metric has been considered an interesting point by us due to the advantages previously mentioned, because it is considered later on when comparing the performance with auto-generated labels. It will be important and explanatory to count with more than one metric used to measure the state-of-the-art 3D object detection models based on

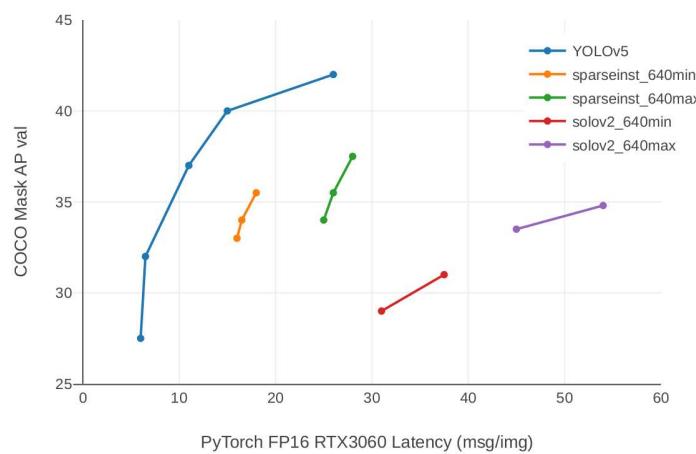
point clouds. This new metric will provide a more complete view of the performance of the proposed algorithm, making a comparison with the **state-of-the-art (SOTA)** models possible.

In addition to the fact that the models shown have all been trained with KITTI, two of them, Voxel-RCNN and PV R-CNN, have also been tested with WAYMO and their performance has also been proven to be good. The aim of this work is to demonstrate that the auto-labeling method does not depend on the way the dataset has been obtained, in terms of how and from what point of view the objects have been obtained. In addition, it is proved that the models, existing or to be developed, can work in different scenario settings (usually focused on the point of view from which the data are captured).

#### 2.4. Auto-label model

For the labeling task we determined that the YOLOv5 model, pre-trained by Ultralytics, was a good tool to use based on the comparison shown in the literature benchmark, so this model with its weights have been selected as the correct ones to perform this process.

Regarding the auto-labeling task, it is decided to use the YOLOv5 model, using the pre-trained weights in the Ultralytics implementation[24]. It was decided to use this particular model as a result of its performance reported in the benchmark literature presented on the Ultralytics main page (Figure 1).



**Figure 1.** v7.0 - YOLOv5 literature Realtime Instance Segmentation [24].

To evaluate the accuracy of YOLOv5 labeling we followed the same procedure as in [26], in Table 1 shows the results of our evaluation on the YOLOv5 model with images from the KITTI raw dataset [11].

**Table 1.** auto-labeling performance of the proposed YOLOv5 model with KITTI raw dataset images.

	Easy	Moderate	Hard
F-Score	0.957	0.927	0.740

Once the 2D data are labeled, we propose to label the 3D point clouds by projecting the labels onto the corresponding KITTI point clouds and use them as ground truth for the 3D detection models.

#### 2.5. Models trained from scratch

Normally, because of the training of the **Artificial Intelligence (AI)** models, the datasets are divided into training and test data; KITTI is also divided into a training set and a test set, but the official KITTI website has a validation tool with its own validation set. As the KITTI test set has been used to measure the performance of the models that have been trained from scratch with our auto-generated labels, the comparison with the results shown in each paper is not fair because it is obtained by submitting

their models to the official benchmark tool with a specific test selected by authors. Therefore, to make a proper comparison, the pre-trained models in OpenPCD have been taken to measure their performance with the KITTI test set, whose results are slightly different [2](#). The main reason for the performance differences is not only the set used in the validation, is that an [AOS](#) has been calculated by recalling 40 positions, as recently recommended on the official KITTI website [\[29\]](#).

### 3. Results

This section describes the different tests carried out on the model to guarantee and demonstrate its correct operation. Thus, two different steps have been defined in order to ensure two different validation approaches:

- *First step*: validate the error with which the images obtained from the sensors are self-labelled.
- *Second step*: validate that the self-labelled datasets generated are efficient enough to be used as training datasets for DL models from the literature.

#### 3.1. First step Validation

To check whether the target of generating an auto-labeled dataset has been met, it is necessary to consult for each model in Table [2](#) the rows about the OpenPCDet performance and the rows with our auto-generated labels. Not only is the accuracy of the auto-generated labels comparable with manual human annotators, but the performance shown by models trained from scratch with auto-generated labels is also comparable with the models trained with KITTI. Table [2](#) shows a slight deviation between the results obtained using manual labels with respect to the auto-generated labels proposed in the present work.

**Table 2.** Comparison of the performance of the models between using the KITTI dataset and KITTI with auto-generated labels by YOLOv5.

Model	Source	BEV			3D Detection		
		Easy	Moderate	Hard	Easy	Moderate	Hard
PointPillars	Lang et al. <a href="#">[13]</a>	88.35	86.10	79.83	79.05	74.99	68.30
	OpenPCDet	92.03	88.05	86.66	87.70	78.39	75.18
	OpenPCDet - Auto-labeled	93.55	84.70	81.56	85.06	75.00	71.43
Voxel R-CNN	Deng et al. <a href="#">[14]</a>	-	-	-	90.90	81.62	77.06
	OpenPCDet	93.55	91.18	88.92	92.15	85.01	82.48
	OpenPCDet - Auto-labeled	95.78	90.99	88.77	92.54	83.28	82.21
PointRCNN	Shi et al. <a href="#">[12]</a>	-	-	-	85.94	75.76	68.32
	OpenPCDet	94.93	89.12	86.83	91.49	80.65	78.13
	OpenPCDet - Auto-labeled	95.57	90.96	88.80	92.19	82.89	80.33
PV-RCNN	Shi et al. <a href="#">[25]</a>	-	-	-	-	81.88	-
	OpenPCDet	93.02	90.32	88.53	92.10	84.36	82.48
	OpenPCDet - Auto-labeled	92.50	86.93	85.83	90.09	78.25	75.38

Looking through Table [2](#), the results show that there is a slight difference between each model taken from the literature in KITTI and from scratch models that have been trained with auto-generated labels. These differences may be even smaller without doing anything particularly different, since the weights of each model's networks are extensively investigated for best performance for each model using the KITTI dataset and labels, running as many trainings as needed for this purpose. However, the purpose of this work is not to optimize the results, but to show that the use of the mentioned labels is reliable, so we performed testing using the same configuration as the original models without much work to fine tune the network.

A more detailed table analysis shows that the easier the object is to locate, the better the performance of the auto-generated labels. Breaking down the results into easy, moderate and hard, the average difference between the original is around 1.2% on easy, 2.1% on moderate and 3.16% on

hard respect [Bird's Eye View \(BEV\)](#) metrics. These small differences show the model's efficiency in obtaining similar results to the original model.

This variability on the results is justified because of the difference between the ground truth bounding boxes present on the KITTI dataset and the auto-generated labels of our methodology. As can be seen in Figure 2, there is a slight variation in the bounding boxes of both methods. The error calculation on the model performance also takes into account the position and dimension of the bounding box, which can lead to reduced performance when comparing the KITTI ground truth against our model results. This penalty on the error is caused by the location and dimension of the bounding box and not by vehicle detection errors, which produce the above-mentioned result variation.

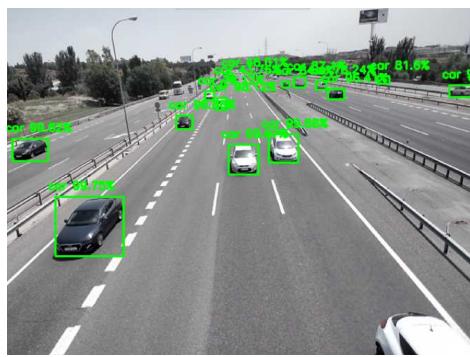


**Figure 2.** KITTI dataset image with two different bounding boxes representation: orange for the ground truth and green for the one generated with auto-labeled model respectively.

### 3.2. Second step Validation

As shown in the first validation step, the auto-labeling algorithm provides similar results to manual labeling. However, it has not been demonstrated whether an auto-labeled dataset with our model is capable of being used to train a [DL](#) model under boundary circumstances.

Therefore, using the same [DL](#) models from the literature as in Section 3.1 to demonstrate its performance using data collected from a different scenario and location than the one proposed in public datasets (KITTI or WAYMO). Thus, to prove the main contribution of this work, it is necessary to further test the performance of the models in a scenario in which they have been re-trained from scratch with our own dataset with the auto-generated labels and the idiosyncrasy of a particular point of view at the height over the road, Figure 3.



**Figure 3.** YOLOv5 labels from bridge point of view.

#### 3.2.1. Set up configuration

In this section, the different tools and configurations used to carry out the tests and with which the results of this section have been obtained will be presented. Thus, the specifications are divided into different sections that contemplate: the scenario from which the data will be captured, the computer equipment used and the sensor elements used.

**Scenario** In order to achieve the main objective of the current work, it is necessary to carry out a test in which data can be obtained from a fixed point of view on the infrastructure. This configuration is relevant for traffic supervision tasks, crossing monitoring, etc. In this way, our methodology is intended to validate with this new configuration widely used in the field of mobility. The choice of location is particularly important, as it has to be ensured that data acquisition is from a point of view as radically different from the traditional one (sensors placed on the roof of a vehicle).



**Figure 4.** Scenerio satellite image.

It is intended that our own auto-labeled dataset has the point of view of the sensors placed on a bridge and pointing to the road. In addition, it had to be ensured that the location was relevant to different case studies. To satisfy this requirement, the pedestrian bridge on km 6 of the A3 highway next to *Madrid* has been selected.

**Sensors** The sensors used in this experiment are a normal video camera and a **LiDAR**. The camera used is the Aukey PC-LM1E webcam characterized by providing a 640x480 resolution RGB image compressed in MJPG format at 30fps. Meanwhile, the **LiDAR** used is the OS1-64 from Ouster characterized mainly by its 64 laser layers, a scope of 120 metros, vertical field view of 45° and a frequency of 10Hz.



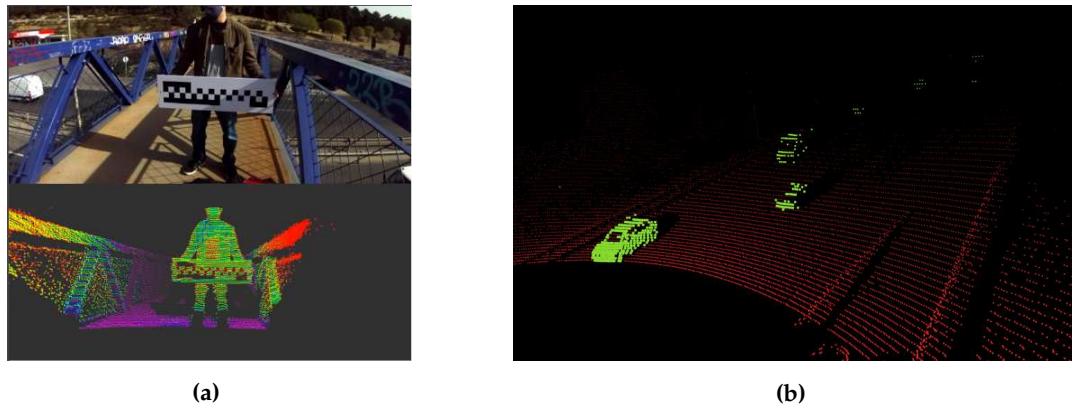
**Figure 5.** Sensors set up.

**Data processing** In order to carry out this experiment, **DL** models, in particular YOLOv5, had to be trained, and data obtained directly from the sensors had to be captured and processed. All data have been processed on a NVIDIA Jetson AGX Xavier with 32 GB memory and Ubuntu 20.04 installed.

As the Nvidia Jetson series is an ARM64 architecture, it is not possible to install the OpenPCDet project, as the authors explain in their installation guide, due to dependencies between the packages and the system. To overcome this problem, the docker image l4t-pytorch provided by NVIDIA has been installed as a container to run the OpenPCDet models.

### 3.2.2. Experiments

When sensor fusion is applied to develop models, it is necessary to calibrate them intrinsically and extrinsically. KITTI has its own calibration matrix for this purpose; however, for our new approach with a configuration different from KITTI, it is required to perform our own calibrations using a custom chessboard calibration system. The calibration procedure and results can be seen in Figure 6, in the same location described in section 3.2.1.



**Figure 6.** (a) Custom calibration board. (b) LiDAR view from the bridge.

The results obtained in this real-life scenario experiment are very similar to the results presented by the models in Table 2. As can be seen in a previous comparative (Table 3), in terms of average precision, the results are close to those of the corresponding model with auto-labeled KITTI as dataset. In this way, it is shown that the model is able to replicate the results in a very similar way.

**Table 3.** Training results of different DL models using an auto-labeled dataset from the scenario described in Section 3.2.1.

Model	BEV			3D Detection		
	Easy	Moderate	Hard	Easy	Moderate	Hard
PointPillars	93.05	84.03	81.50	79.99	65.53	61.03
Voxel R-CNN	95.39	91.19	88.56	92.45	83.33	81.95
PointRCNN	95.55	91.00	88.88	92.57	82.84	79.89
PV-RCNN	93.02	87.08	85.13	89.98	79.14	75.08

In the best of the results obtained by the Voxel R-CNN model, an accuracy above 88% in BEV and in 3D detection above 82% can be seen. This shows good results for vehicle detection, which could be improved by optimizing the networks used.

## 4. Conclusions

One of the main challenges in building a model for auto-labeling auto-motion scenarios is dealing with high data variability. Autonomous vehicles must be capable of operating in a wide variety of environments and the data used to train the model should reflect this diversity. The problem associated with this challenge is growing due to the high demand for data needed to provide increasingly complex AI models with the data.

Regarding the results, the differences between models are not very large; it would be possible to obtain even better results by applying this method to a larger set of models. Considering these four models there is the presence, as can be seen, of statistical outliers such as PointPillars or PV-RCNN with much larger differences than the other two and than with respect to the mean. This is because these particular models are too dependent on the hyper-parameters used for their training. If better parameters were sought for these two or a more extensive set of models not dependent on the network hyper-parameters were used, these statistics could result in a smaller difference in the percentage

accuracy of the models trained with auto-generated labels with respect to the original ones, since the outliers would have less significance.

In general, building an auto-labeling model for autonomous vehicles is a complex task that requires a combination of machine learning, crowd-sourcing, and other techniques. It is important to carefully design and evaluate the model to ensure that it is capable of accurately labeling the data and supporting the development of effective autonomous vehicle systems.

This paper has presented an approach to perform this work using models already known from the literature and has shown that the errors produced in the process are trivial. To take the demonstration of their performance a little further, we have contrasted the results of training models using known data such as KITTI and self-labeled data, where, despite slightly worse results, the difference is negligible in contrast to the benefits of using these technologies.

Therefore, the paper presents a solution to the tedious job of labeling large datasets using a self-labeling model. As shown in the results presented, the model works correctly with negligible errors, and it has also been shown that models can be trained exclusively using these self-labeled datasets.

**Author Contributions:** For the development of this work, the authors have divided the work and contributed as follows: Conceptualization, Guillermo S. Gutierrez-Cabello and Miguel Clavijo; methodology, Edgar Talavera and Felipe Jiménez; software, Guillermo S. Gutierrez-Cabello; validation, Guillermo Iglesias, Miguel Clavijo and Edgar Talavera; writing—original draft preparation, Guillermo S. Gutierrez-Cabello and Guillermo Iglesias; supervision, Felipe Jiménez, Edgar Talavera and Miguel Clavijo. All authors have read and agreed to the published version of the manuscript.

**Funding:** Proyect PID2019-104793RB-C33 funded by MCIN/ AEI /10.13039/501100011033 and project PDC2022-133684-C32 funded by MCIN/AEI /10.13039/501100011033 and European Union Next GenerationEU/ PRTR.

**Data Availability Statement:** The datasets used for the experiments and comparisons of all the results of the paper have been obtained The KITTI Vision Benchmark Suite, A project of Karlsruhe Institute of Technology and Toyota Technological Institute at Chicago. References: [3,10,26]

**Acknowledgments:** Special thanks to Álvaro García González for his approach and work that led to the beginning of the development of this idea for the paper.

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

## References

1. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788. doi:10.1109/CVPR.2016.91.
2. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollar, P.; Zitnick, C.L. Microsoft COCO: Common Objects in Context. Computer Vision – ECCV 2014; Fleet, D.; Pajdla, T.; Schiele, B.; Tuytelaars, T., Eds.; Springer International Publishing: Cham, 2014; pp. 740–755.
3. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets Robotics: The KITTI Dataset. *International Journal of Robotics Research (IJRR)* **2013**.
4. Sun, P.; Kretzschmar, H.; Dotiwalla, X.; Chouard, A.; Patnaik, V.; Tsui, P.; Guo, J.; Zhou, Y.; Chai, Y.; Caine, B.; Vasudevan, V.; Han, W.; Ngiam, J.; Zhao, H.; Timofeev, A.; Ettinger, S.; Krivokon, M.; Gao, A.; Joshi, A.; Anguelov, D. Scalability in Perception for Autonomous Driving: Waymo Open Dataset. 2020, pp. 2443–2451. doi:10.1109/CVPR42600.2020.00252.
5. Roh, Y.; Heo, G.; Whang, S.E. A Survey on Data Collection for Machine Learning: A Big Data - AI Integration Perspective. *IEEE Transactions on Knowledge and Data Engineering* **2021**, *33*, 1328–1347. doi:10.1109/TKDE.2019.2946162.
6. Chadwick, S.; Maddern, W.; Newman, P. Distant Vehicle Detection Using Radar and Vision. 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 8311–8317. doi:10.1109/ICRA.2019.8794312.

7. Dong, X.; Wang, P.; Zhang, P.; Liu, L. Probabilistic Oriented Object Detection in Automotive Radar. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2020.
8. Chen, Z.; Liao, Q.; Wang, Z.; Liu, Y.; Liu, M. Image Detector Based Automatic 3D Data Labeling and Training for Vehicle Detection on Point Cloud. 2019 IEEE Intelligent Vehicles Symposium (IV), 2019, pp. 1408–1413. doi:10.1109/IVS.2019.8813818.
9. Chen, X.; Ma, H.; Wan, J.; Li, B.; Xia, T. Multi-View 3D Object Detection Network for Autonomous Driving. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
10. Fritsch, J.; Kuehnl, T.; Geiger, A. A New Performance Measure and Evaluation Benchmark for Road Detection Algorithms. International Conference on Intelligent Transportation Systems (ITSC), 2013.
11. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets Robotics: The KITTI Dataset. *International Journal of Robotics Research (IJRR)* **2013**.
12. Shi, S.; Wang, X.; Li, H. PointRCNN: 3D Object Proposal Generation and Detection From Point Cloud. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019.
13. Lang, A.H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; Beijbom, O. PointPillars: Fast Encoders for Object Detection from Point Clouds. *CVPR 2019*.
14. Deng, J.; Shi, S.; Li, P.; Zhou, W.; Zhang, Y.; Li, H. Voxel R-CNN: Towards High Performance Voxel-based 3D Object Detection. AAAI, 2021.
15. Caesar, H.; Bankiti, V.; Lang, A.H.; Vora, S.; Lioung, V.E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; Beijbom, O. nuScenes: A multimodal dataset for autonomous driving. CVPR, 2020.
16. Patil, A.; Malla, S.; Gang, H.; Chen, Y.T. The H3D Dataset for Full-Surround 3D Multi-Object Detection and Tracking in Crowded Urban Scenes. 2019 International Conference on Robotics and Automation (ICRA), 2019, pp. 9552–9557. doi:10.1109/ICRA.2019.8793925.
17. Yu, W.; Sun, Y.; Zhou, R.; Liu, X. GAN Based Method for Labeled Image Augmentation in Autonomous Driving. 2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE), 2019, pp. 1–5. doi:10.1109/ICCVE45908.2019.8964902.
18. Simony, M.; Milzy, S.; Amendey, K.; Gross, H.M. 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, 2018.
19. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. Computer Vision – ECCV 2016; Leibe, B.; Matas, J.; Sebe, N.; Welling, M., Eds.; Springer International Publishing: Cham, 2016; pp. 21–37.
20. Neupane, B.; Horanont, T.; Aryal, J. Real-Time Vehicle Classification and Tracking Using a Transfer Learning-Improved Deep Learning Network. *Sensors* **2022**, *22*.
21. Imad, M.; Doukhi, O.; Lee, D.J. Transfer Learning Based Semantic Segmentation for 3D Object Detection from Point Cloud. *Sensors* **2021**, *21*. doi:10.3390/s21123964.
22. Vatani Nezafat, R.; Sahin, O.; Cetin, M. Transfer Learning Using Deep Neural Networks for Classification of Truck Body Types Based on Side-Fire Lidar Data. *Journal of Big Data Analytics in Transportation* **2019**, *1*. doi:10.1007/s42421-019-00005-9.
23. Team, O.D. OpenPCDet: An Open-source Toolbox for 3D Object Detection from Point Clouds, 2020.
24. Jocher, G. YOLOv5 by Ultralytics. 2020. doi:10.5281/zenodo.3908559.
25. Shi, S.; Guo, C.; Jiang, L.; Wang, Z.; Shi, J.; Wang, X.; Li, H. PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.
26. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? The KITTI vision benchmark suite. 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3354–3361. doi:10.1109/CVPR.2012.6248074.
27. Zhou, H.; Ge, Z.; Mao, W.; Li, Z. PersDet: Monocular 3D Detection in Perspective Bird's-Eye-View. *arXiv preprint arXiv:2208.09394* **2022**.
28. Ng, M.H.; Radia, K.; Chen, J.; Wang, D.; Gog, I.; Gonzalez, J.E. BEV-Seg: Bird's Eye View Semantic Segmentation Using Geometry and Semantic Point Cloud. *arXiv preprint arXiv:2006.11436* **2020**.

29. Simonelli, A.; Bulò, S.R.; Porzi, L.; Lopez-Antequera, M.; Kotschieder, P. Disentangling Monocular 3D Object Detection. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 1991–1999. doi:10.1109/ICCV.2019.00208.

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.