IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 5 September 2023, accepted 22 September 2023, date of publication 4 October 2023, date of current version 10 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3321912

SURVEY

Key Indicators to Assess the Performance of LiDAR-Based Perception Algorithms: A Literature Review

CHIRANJEEVI KARRI[1], JOSÉ MACHADO DA SILVA[1,2], AND MIGUEL VELHOTE CORREIA[1,2], (Member, IEEE)

[1]Department of Electrical and Computer Engineering, Faculty of Engineering—University of Porto (FEUP), 4200-465 Porto, Portugal
[2]INESC TEC—INESC Technology and Science, 4200-465 Porto, Portugal

Corresponding author: Chiranjeevi Karri (kchiranjeevi@fe.up.pt)

This work was supported by the European Structural and Investment Funds in the FEDER Component through the Operational Competitiveness and Internationalization Programme (COMPETE 2020) (THEIA: Automated Perception Driving; Funding Reference: POCI-01-0247-FEDER-047264) under Project 047264.

ABSTRACT Perception algorithms are essential for autonomous or semi-autonomous vehicles to perceive the semantics of their surroundings, including object detection, panoptic segmentation, and tracking. Decision-making in case of safety-critical situations, like autonomous emergency braking and collision avoidance, relies on the outputs of these algorithms. This makes it essential to correctly assess such perception systems before their deployment and to monitor their performance when in use. It is difficult to test and validate these systems, particularly at runtime, due to the high-level and complex representations of their outputs. This paper presents an overview of different existing metrics used for the evaluation of LiDAR-based perception systems, emphasizing particularly object detection and tracking algorithms due to their importance in the final perception outcome. Along with generally used metrics, we also discuss the impact of Planning KL-Divergence (PKL), Timed Quality Temporal Logic (TQTL), and Spatio-temporal Quality Logic (STQL) metrics on object detection algorithms. In the case of panoptic segmentation, Panoptic Quality (PQ) and Parsing Covering (PC) metrics are analysed resorting to some pretrained models. Finally, it addresses the application of diverse metrics to evaluate different pretrained models with the respective perception algorithms on publicly available datasets. Besides the identification of the various metrics being proposed, their performance and influence on models are also assessed after conducting new tests or reproducing the experimental results of the reference under consideration.

INDEX TERMS Perception algorithms, metrics, deep learning, object detection, panoptic segmentation, autonomous driving.

I. INTRODUCTION

Human drivers successfully complete their driving tasks by 1) being aware of their current situation, including their steering angle, speed, location, and acceleration; 2) sensing the conditions of surrounding obstacles; 3) formulating a future course of action that will ensure their safety; and 4) operating the steering wheel and brakes to control the vehicle. But human errors like distraction, weariness, speeding, breaking traffic laws, and poor judgment on other drivers actions are the reasons for the majority of road accidents.

A 2018 study by the National Highway Transportation Safety Administration (NHTSA) [1] states that about 94% of all car accidents are caused by human errors. Later, safety experts claimed that this statistic was made up. Nevertheless, research has confirmed that human failure is the main cause of road accidents and that the introduction of some sort of automation contributes to reduce accidents statistics. Assistance and partly automated systems may prevent weaknesses in human capacities and increase safety in routine

The associate editor coordinating the review of this manuscript and approving it for publication was Junho Hong.

human driving cases with supervision, warnings and lateral or longitudinal support [2]. For example, the number of accidents caused by driver-error and skidding fell from about 2.8 (per 1000 cars) in 1998/1999 to 2.21 in 2000/2001, after Mercedes-Benz introduced Electronic Stability Control (ESC) as a standard in all cars [2].

A significant potential to lower errors and thereby achieve higher road safety, collision-free, profitability, and traffic control is achieved by automating the driving task [3]. The Society of Automotive Engineers (SAE International) defines six levels of driving automation going from no driving automation (Level 0) to full driving automation (Level 5) [4]. While Level 1 corresponds to basic driver assistance, such as using cruise control on highways, Levels 2 and higher include ADAS (Advanced Driver-Assistance System) features, where sensors and a computer are used to sense and analyze the surroundings to make decisions based on the proximity of objects. Within Level 3 the vehicle can handle most of the driving tasks, but the driver must still be ready to take control in certain situations. In Level 4 vehicles can operate in self-driving mode without human interaction in most circumstances, but a human still has the option to manually override. The difference to Level 3 is that Level 4 vehicles can take control in case of anomaly or system failure. At level 5, the vehicle does not require any human intervention, becoming a truly autonomous vehicle (AV).

Globally, these systems are being explored in order to realize their enormous potential, resorting to sensors like LiDAR (Light Detection And Ranging), cameras, ultrasound/sonar, RADAR (Radio Detection and Ranging), and GPS (Global Positioning System) to extract information from the surrounding environment [5]. Inertial measurement units (IMU) are also used to measure the vehicle's linear acceleration and angular velocity, providing information on the vehicle's current location and orientation (relatively to a known starting location).

Among all these sensors, LiDAR is currently the one that deserves the highest attention from industry. They show fast response, high resolution and high accuracy, high surface sample density, can be used day and night, and are economically accessible. According to a report by Grand View Research [6], the global LiDAR market size was worth US$ 1.81 billion in 2021 and is expected to increase with a compound annual growth rate (CAGR) of 9.8% from 2022 to 2030, with the ADAS segment expected to show the highest CAGR (13.9%) over this period, owing to the use of LiDAR to power ADAS systems up to Level 3.

A. DATA PROCESSING AND NAVIGATION

The control systems translate this sensory data into a two- or three-dimensional representation of the environment, determine the best navigation routes after identifying other vehicles, cyclists, pedestrians, traffic signs, stop signs, and generic obstacles, and manage the vehicles longitudinal and lateral motions simultaneously [7].

Recent developments in image processing and machine learning techniques make it simpler to implement these tasks. Object detection and tracking [8], object classification [9], semantic segmentation/instance segmentation [10], and localisation [11] are eventually the most useful operations for the perception of vehicle surroundings. The perception and motion planning modules are the most difficult assignments. The major function of the perception module is to comprehend/abstract the environment by processing data from sensors [12].

Within ADAS, object detection is a computer vision approach that enables the recognition and finding of objects in an image captured using a camera or/and LiDAR [13]. With the inclusion of features for identification and localization, object detection can be used to identify, localize, and count objects in a scene and label them appropriately. The process of tracking objects involves taking a collection of initial objects, giving each one a special identification (ID), and then following each object as it moves across the frames of a video while still keeping the ID assigned. Object classification is a part of object detection that helps to classify the objects in the image. An advanced method of image segmentation, called instance segmentation, deals with locating instances of things and defining their bounds [14].

Object detection, tracking, and classification are all tasks performed by a perception module. This serves as the framework for driving assistance and organizing AV's future mobility. The observation of an AV's status, including location, speed, and momentum, is required for its localization. For an approximate state estimation, one can resort to a GPS system [15]. Perception entails monitoring the conditions of the nearby obstacles, such as their position, speed, momentum, and class. To identify, categorize, and track the nearby obstacles, researchers have proposed various machine learning algorithms for the analysis of data collected from LiDARs, RADARs, GPS, and cameras. In order to safely travel in a challenging environment, an AV plans its subsequent decisions using knowledge about its surroundings. An extremely difficult problem is the motion planning (or, to be more accurate, trajectory planning) of the vehicle, which entails determining the vehicle's future states (location, speed, and velocity) in continuously changing traffic conditions. The motion planning module uses the present and potential future states of the surrounding obstacles to ensure the vehicle's safe and effective movement through the dynamics of traffic conditions. To prevent accidents, extreme caution must be taken. The tricky duty of environmental perception can be greatly simplified by wireless communication among all vehicles on the route. Nevertheless, this scenario would only be possible if all vehicles on the road are interconnected through wireless connection. Hence, the performance and effectiveness of the ADAS core modules determine the AV's safety.

Several AVs employ various types of perception algorithms and sensors. LiDARs are used by certain developers, while cameras are the primary sensors for others. As a result, the

design of the employed sensors will automatically affect how the environment is seen. AV's performance is mostly dependent on the perception algorithms utilized for processing the data provided by sensors. To ensure public acceptance, AVs driving behavior must resemble that of human drivers. To do so, the performance of perception algorithms, which depend on diverse parameters, must be accurate; for example, object detection algorithms depend on the size of the dataset, the correctness of the respective labels, the accuracy of sensor devices and model hyper-parameters. The accuracy of semantic segmentation depends on the individual pixel and its correlation with neighboring pixels. For the sake of safety, the ultimate objective is to minimize the probability of occurrence of false positives (an outcome where the model predicts the occurrence of an event which did not actually occur) and false negatives (a non detected occurrence that actually occurred).

This paper presents a review of different metrics used to measure the performance of perception algorithms, including object detection & object tracking, semantic & panoptic segmentation, and metrics used for the evolution of LiDAR sensors. Besides the identification of the various metrics being proposed, their performance and influence on models are assessed as well, after conducting new tests or reproducing the experimental results of the reference under consideration. The metrics used to assess perception algorithms can be split into the following four:

- Point Cloud: A three-dimensional set of measures acquired by the LiDAR of the vehicle's surroundings
- Object detection: List of detected objects where each one has been assigned a class. The measured metrics are accuracy, precision, recall, F1-score, Intersection Over Union (IOU), area under the so-called receiver-operating characteristic (ROC) curve, Planning KL-Divergence (PKL), Timed Quality Temporal Logic (TQTL), and Spatio-temporal Quality Logic (STQL).
- Object tracking: Is the process of estimating each identified object's position, dimensions, velocity, and respective class. The used metrics are multiple object tracking (MOT) accuracy and MOT precision.
- Semantic Segmentation: A point cloud is segmented into subgroups to facilitate further processing or analysis of each segment. Upon segmentation, labels are assigned to pixels to identify objects, pedestrians, and other important elements in the point cloud. To assess it, Dice coefficients, precision, and recall are used.

B. REVIEW ON PERCEPTION ALGORITHMS

Numerous studies have been done to date that look into different facets of autonomous vehicle technology [16]. To the best of our knowledge, none of these studies offer a comprehensive view on metrics to assess the performance of perception algorithms for AVs; instead, the majority of them concentrate on just one aspect of the AVs. The authors of [17] provide a review of AVs in view of hardware architectures, simulation software, deep learning models, and computational resources used till 2023. A study of algorithms and hardware used in AV's visual perception systems is mentioned in [18]. A survey on the applications of AI techniques in the creation of AV's is given in [19] that includes virtual & augmented reality, high performance computing, big data, and advancements in 5G communication for AV's. Stages of development, obstacles, and trends for the practical implementation of an energy management plan for AVs based on connected and intelligent technologies are given in [20]. Issues with security, privacy, and trust are a few of the most important ones in the AV's domain, and a review of various technologies like information and communication technology, Blockchain, AI, etc. covering these issues is given in [21] and [22]. A thorough analysis of the literature on the factors influencing the use of AVs can be found in [23]. An analysis of recent advances in obstacle detection technologies is presented in [24]. A description of different sensors and deep learning models used for obstacle detection can be found in [25]. Overviews of sensor technologies and sensor fusion for AV's perception are provided in [26] and [27], respectively. To increase road safety, AV's performance must have a solid, reliable perception, so the authors in [28] outline recent developments, suggest potential avenues for next research, and list the benefits and drawbacks of various sensor and localization/mapping algorithm configurations. Also, some issues, including detection certainty, illumination and weather, sensor fault detection, and other difficulties pertaining to AVs, include algorithm effectiveness, reliance on prior data, and public perception. Principles, issues, and developments in automotive LiDAR and perception systems for AVs are discussed in [29] and an examination of usual procedures and new technologies is provided in [30]. Other reviews focused on AV's applications have been published on motion planning [31], object detection [32], semantic segmentation [33] techniques, analysis of deep learning methods for semantic segmentation of images and videos [34], and deep learning-based image recognition [35].

The rest of the paper is organized as follows: Section II describes the search strategy adopted to withdraw relevant sources and publications. Performance indicators that have been adopted for LiDAR devices are given in Section III. Moreover, Section IV elaborates on the metrics that have been proposed to evaluate object detection. Section V provides an overview of the benefits and restrictions of the current performance indicators for object tracking. In Section VI semantic, instance, and panoptic segmentation are introduced, and the respective metrics are described. Section VII gives a theoretical and practical explanation of metrics with their respective models. Finally, section VIII provides a summary of the paper and highlights the main conclusions as well as new developments to be considered.

II. THE ADOPTED SEARCH STRATEGY

A comprehensive literature review was made based on articles published in international journals and conferences

between 2013 and 2023. This review is mainly focused on metrics for perception algorithms by looking at the critical academic publications of Science Citation Index (SCI), Science Citation Expanded (SCIE), and Scopus. Conference articles presented at well-known organizations, universities, and platforms under the umbrella of IEEE, Springer, and Elsevier and indexed by Scopus were taken into account. Three database sources were explored for relevant articles, mainly IEEE Explore, Scopus, and Google Scholar. These three sources mainly cover articles published in IEEE, Inderscience, IGI Global, MDPI, Willy, and ScienceDirect. A combination of several keywords was used to search for relevant articles. For example, ''perception algorithm metrics'', ''autonomous vehicle metrics'', ''object detection metrics'', ''object tracking metrics'', ''semantic segmentation metrics'', and ''panoptic segmentation metrics''. In addition, different keywords were used depending on the technology used for perception algorithm metrics. Some publishers reserve a few journals, books, or special issues that cover the main technologies related to autonomous vehicles. For example, ScienceDirect launched a journal in 2021 with the title ''Autonomous Vehicles'', Springer publisher is maintaining a journal with the title ''Autonomous Intelligent Systems'', Wiley holds an open access book with the title ''Autonomous Vehicles: Using Machine Intelligence'', and IEEE is publishing the ''IEEE Transactions on Intelligent Vehicles'' journal. Only with the ''perception algorithm'' keyword, 2,153 publications were found in the IEEE database. But, with the combination of another keyword (''perception algorithm + metrics''), the count was reduced to 798. Another approach is to use year-wise filtering; for example, the exact count was further reduced to 254. In this way, irreverent and incomplete publications were filtered. Also, book chapters, case reports, and letters were disregarded.

A. FILTERING PROCESS

Five selection criteria were used to collect relevant articles for this review. Those are:

- The title and abstract of the articles were checked against the stacked eligibility criteria. Duplicates and publications that did not match the basic inclusion criteria were eliminated.
- To guarantee that the included articles were most relevant to today's perception algorithm metrics, only publications from 2013 to 2023 were considered.
- To attract more readers, only publications written in English were included.
- Publications that were unavailable or lacked a full text or abstract were also discarded.
- Publications relevant to state-of-the-art technologies were included.

After applying the above five filtering criteria, the final numbers of articles selected for review were 81, 65, 11, 31, and 19, respectively, for ''autonomous vehicle metrics'', ''object detection metrics'', ''object tracking metrics'', ''semantic segmentation metrics'', and ''panoptic segmentation metrics''. A tool that can be used to document the many phases of the literature search procedure is the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) [36]. The PRISMA flow diagram of the adopted search strategy can be seen in Fig. 1.

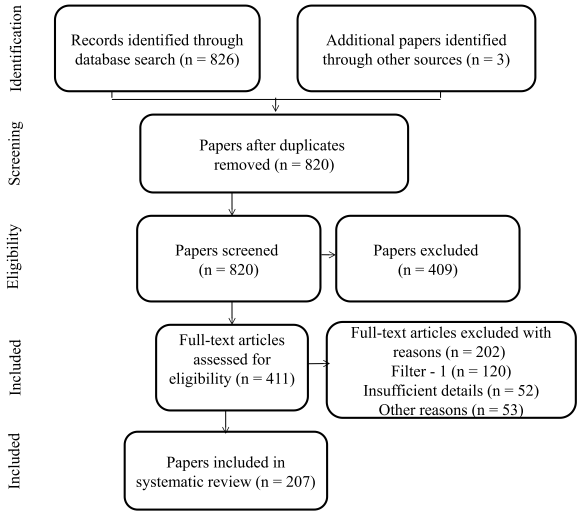


FIGURE 1. PRISMA diagram of the adopted bibliographic search strategy.

III. METRICS FOR LiDAR POINT CLOUD

To measure a LiDAR's point cloud performance, the point cloud distance is calculated by finding the minimum Euclidean distance between equivalent points in a reference cloud and in the captured point cloud. There are four distance metrics:

- Hausdorff Distance (HD): Is the largest of all Euclidean distances between any two points (x,y) in different point clouds [37]. More formally, the HD from P to Q is a maximin function, defined as (eq. 1)[1]

$$HD(P, Q) = \max \left\{ \sup_{x \in P} \inf_{y \in Q} d(x, y), \sup_{y \in Q} \inf_{x \in P} d(x, y) \right\} \quad (1)$$

  where x and y are points of P and Q, respectively, $d(x, y)$ is the Euclidian distance between x & y, and $'sup'$ & $'inf'$ are the supremum and infimum.
- Modified Hausdorff Distance (MHD): Is a modified version of HD proposed in [38] and uses the sum of the mean of the minimum distance between two sets of points; it is less prone to outliers. The MHD was found after extensive research into 24 various distance measures and their behavior in the presence of noise.
- Chamfer Distance (CD): When two point clouds are evaluated using the Chamfer Distance, each of the distances from a point in one cloud to all points in the

[1] https://pdal.io/en/2.4.3/apps/hausdorff.html

other cloud are taken into consideration. CD locates the closest point in the other point set and adds the square of the distance for each point in either cloud. The CD between two point clouds P and Q is given as in eq. 2.[2]

$$\mathrm{CD}(P,Q) = \frac{1}{|P|}\sum_{x\in P}\min_{y\in Q}\|x-y\|_2^2 + \frac{1}{|Q|}\sum_{y\in Q}\min_{x\in P}\|x-y\|_2^2 \quad (2)$$

where x and y are, respectively, points of P and Q.

- Earth Mover's Distance (EMD): It is also known as the Discrete Wasserstein distance [39]. It is a technique for determining the degree to which two multi-dimensional distributions differ in a feature space, where a ground distance is the measurement of the distance between individual features. The Earth Mover's Distance between two point clouds (P and Q) is calculated with eq. 3 [40].

$$EMD(P,Q) = \min_{\phi:P\to Q}\sum_{x\in P}|x-\phi(x)|_2 \quad (3)$$

where $\phi(x)$ is a bijective function $f : P \to Q$, i.e., a one-to-one (injective) and onto (surjective) mapping of P to Q.

A. LiDAR ACCURACY ASSESSMENT

Estimating a LiDAR's accuracy by finding the Root Mean Square Error (RMSE) between two point clouds is a typical practice. There are two different accuracy assessments: Absolute accuracy and Relative accuracy [41].

1) ABSOLUTE LiDAR ACCURACY

It refers to the vertical and horizontal precisions of data collected from a LiDAR. By comparing the collected LiDAR data with ground surveyed checkpoints, absolute accuracy is evaluated [41] with the condition that horizontal checkpoints, ground-level features, are well defined. Its horizontal placements are precisely measured in relation to the objects' geographic locations. On the other hand, vertical checkpoints do not have to be well defined. The term vertical accuracy refers to the vertical precision attained over the environment. There is no right way to choose the right checkpoint distribution. It typically depends on the geographic location of the objects and the environment under evaluation.

2) RELATIVE LiDAR ACCURACY

It is a metric to measure small variations in the point cloud [41] and the LiDAR calibration has an impact on it. There are two approaches to evaluate relative accuracy: The evaluation of data acquired by an autonomous vehicle with two different LiDARs at the same location is often known as ''within-swath accuracy''. It reveals the LiDAR system's level of stability; The evaluation of data obtained by an AV with two different LiDARs at different locations is often known as ''swath-to-swath accuracy''. In addition to these metrics, table 1 gives other metrics with their advantages and limitations.

[2]https://pdal.io/en/2.4.3/apps/chamfer.html

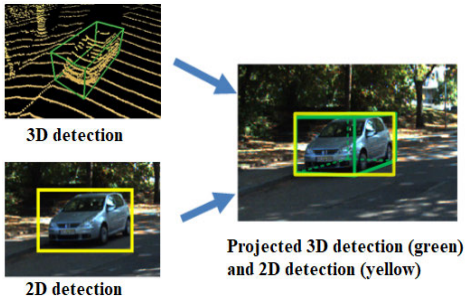


FIGURE 2. 3D object detection with camera and LiDAR.

IV. METRICS FOR OBJECT DETECTION

Autonomous vehicles require precise 3D vision of the surrounding environment, including other vehicles and all other relevant objects. Using 3D-based object detection, spatial path planning for object avoidance and navigation is possible, as opposed to 2D detection. With more output parameters required to indicate 3D-oriented bounding boxes around targets, 3D object detection is more difficult than 2D object detection, which has been extensively investigated in [47]. Moreover, the resolution of LiDAR data is often lower than that of video, which has a significant negative influence on accuracy at extended ranges. Three object detection modalities based on dataset dimensions that can be found in the literature are: 2D image based, 3D point cloud based, and fusion of both image and point cloud detection. Despite the advantage of not requiring LiDAR, 2D image-based approaches perform poorly as compared to those that use point clouds; therefore, here we concentrate on the first two categories.

As illustrated in Fig. 2, 2D object detection algorithms use RGB images as input and produce 2D axis-aligned bounding boxes with confidence scores, while 3D object detection algorithms work with 3D point clouds and produce classified, oriented 3D bounding boxes with confidence scores. The 3D bounding box in the LiDAR coordinates may be precisely projected into the image plane using the calibration settings of the camera and LiDAR after a fusion process. So, metrics to be considered in the case of object detection include 3D object detection using a camera, 3D object detection using LiDAR, fusion of both, and finally object tracking. In the following section, we discuss each of these individual metrics.

A. METRICS FOR 3D OBJECT DETECTION

To evaluate the effectiveness of object detection algorithms, intersection over union or the Jaccard index are used to compare the predicted and ground truth. As shown in Fig. 3, each ground truth box in the image is taken into consideration while calculating IOU for each prediction. Then, using a

TABLE 1. Metrics to characterize LiDAR point clouds.

Publication	Year	Main characteristics and distinguishing features	Strengths and limitations
Zhao et al., [42]	2020	Quality of 3D LiDAR point cloud assessed by direction representation of local point set and density information of spherical region.	Linear correlation between the metric and the vision task accuracy when applied for segmentation task
Triess et al., [43]	2020	Quality of 3D LiDAR point cloud assessed by obtaining relevant features learned from real-world and synthetic point clouds by training on a proxy classification task.	Metric more reliable and applicable on unseen data when tested for perception tasks.
Liu et al., [44]	2022	Introduced a no-reference (NR) quality metric called ResS-CNN, which is based on sparse convolutional neural networks (CNN), and developed a large-scale dataset with 104 reference point clouds and more than 22,000 distorted samples.	The ResSCNN metric has advantages over existing NR and Full Reference (FR) metrics in measuring the quality of LiDAR point clouds.
Kodors & Sergejs [45]	2017	A mathematical model to measure the quality of LiDAR data based on point density.	Dependence between building detection quality (Kappa coefficient or total accuracy) and point spacing.
Liu et al., [46]	2020	PointSSIM – Quality of LiDAR by taking structural similarity between reference and point cloud under evolution.	Best performance over the full reference metrics with high prediction accuracy under certain conditions.

greedy approach, predictions are matched with ground truth boxes after these IOUs have been thresholded to a certain value, often between 0.5 and 0.95 (the highest IOUs are matched first). Then it is determined whether a prediction is True Positive (TP), False Positive (FP), or False Negative (FN) with the aid of the IOU threshold value. It is crucial to keep in mind that a true negative (TN) result does not apply in the domain of object detection, because there are a limitless number of bounding boxes that should not be detected in any image. With the help of TP, FP, and FN, a confusion matrix is obtained as shown in Fig. 4. With the known confusion matrix, calculate the precision and recall. These metrics are also used for segmentation purposes, so we are defining specificity even if it is not important for object detection.

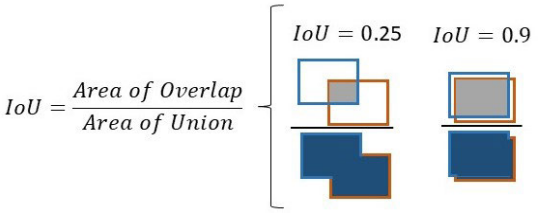


FIGURE 3. Intersection over union (IOU).

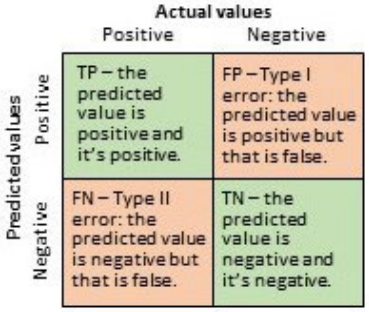


FIGURE 4. Confusion matrix.

1) PRECISION

Precision is the ratio of true positives to all positive predictions (true plus false predictions). For instance, if the model identified 100 trees and 90 of them were accurate, the precision would be 90%.

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

2) RECALL AND SPECIFICITY

Recall is also called true positive rate or sensitivity, and it gives the percentage of positive voxels in the label or ground truth that are positive. The specificity, or true negative rate, gives the percentage of negative voxels (background) in the ground truth detection that are further detected as negative by the assessed detection.

$$Recall = \frac{TP}{TP + FN} \qquad Specificity = \frac{TN}{TN + FP} \tag{5}$$

3) F1-SCORE

The F1-score is particularly suited for imbalanced datasets. It gives the harmonic mean of precision and recall.

$$F1 - score = \frac{2 \times precision \times recall}{precision + recall} \tag{6}$$

4) ACCURACY

Accuracy is the proportion of valid predictions, including true positives and true negatives, among the total number of analyzed cases.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

5) MEAN AVERAGE PRECISION (mAP)

One of the issues with object detection is the diversity of classes involved, e.g., car, tanker, pedestrian, bicycle, and bus. The average precision $AP(i)$ is the average of the individual class precision of the *ith* image in a total of F images in the dataset, and the average of all such precisions is mAP.

$$mAP = \frac{\sum_{i=1}^{F} AP(i)}{F} \tag{8}$$

The precision & recall curve can be viewed as a trade-off between TP and FN. The precision will be high if a detector's confidence level is such that its FP is low. Unfortunately, many positives may be missed in this situation, resulting in a high FN and consequently a low recall. The recall will increase if one accepts more positives, but the FP may increase as well, lowering the precision. A competent object detector should, however, locate every ground truth object (FN = 0; high recall) and recognize only pertinent objects (FP = 0; high precision). Hence, a specific object detector can be deemed effective if its precision maintains a high level while its recall increases, i.e., the precision and recall will still be high even if the confidence threshold changes. As a result, a high Area Under the Curve (AUC) usually denotes both good precision and strong recall. Unfortunately, in real-world situations, the precision-recall plot frequently has a zigzag shape, making it difficult to determine an exact AUC. This is avoided by preprocessing the precision recall curve to eliminate the zigzag behavior before estimating the AUC. Basically, there are two methods to do this: 11-point interpolation and all-point interpolation [48].

6) 11-POINT INTERPOLATION (AP_{11})

The highest precision whose recall value is greater than a particular value is taken into consideration in this definition of AP rather than the precision seen at each recall level [48]. The highest accuracy values at a set of 11 equally spaced recall levels [0, 0.1, 0.2, . . . , 1] are averaged to determine the precision recall curve form in the 11-point interpolation [49]. It is calculated with eq. 9.

$$AP_{11} = \frac{1}{11} \sum_{T \epsilon (0,0.1,.....1)} MP_{11} \tag{9}$$

where MP_{11} is the first 11 maximum precision values.

7) ALL-POINT INTERPOLATION (AP_{all})

Here, the AP is generated after interpolating the precision at each level, using the highest precision whose recall value is greater or equal to the particular value, as opposed to using the precision observed at only a few places [48].

For a better understanding of the 11-point and all-point interpolations, let's take an example of an object detection case [48] whose precision and recall curve is shown in Fig. 5. From this figure, the obtained average precision values are 26.84% and 24.56% with the 11-point and all-point interpolations, respectively.

The average precision computation has significant flaws due to the N-point interpolation techniques currently in use. It is impossible to accurately assess the model's performance because of these mistakes, which lead to average precision distortion. To address these problems, an enhanced interpolation was proposed in [48] by taking the position of the interpolation point from the middle and dynamic parameter selection in determining the interpolation interval's area. They observed that the average precision distortion is reduced by over 90% to only 0.04%.

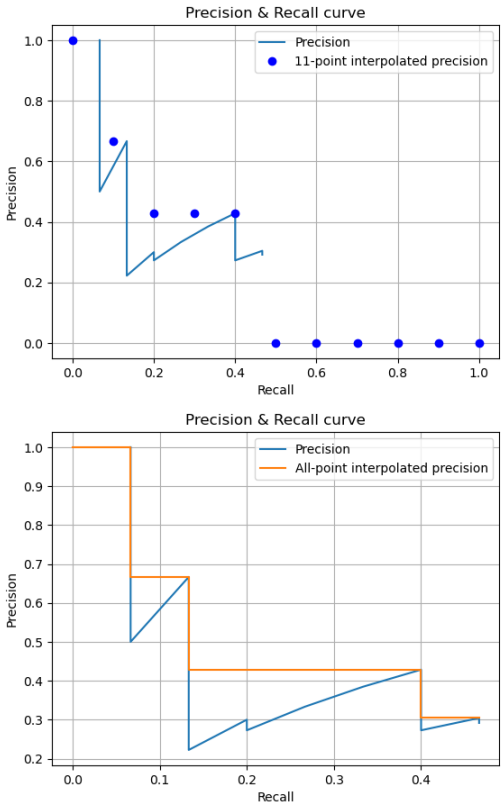


FIGURE 5. Representations of 11-point and all-point interpolations.

8) AVERAGE RECALL (AR)

The aggressiveness of object detectors for a particular class is measured using another assessment metric called average recall [50]. The assessed detector confidences are not included in the computation of AR, in contrast to the average precision. Because of this, the confidence threshold is effectively set to 0, and all detections are considered positive [51]. By including all recall results acquired for IOU thresholds in the span [0.5,1] the AR metric is evaluated by considering a wide range of IOU thresholds. The least reasonable IOU according to most metrics is 0.5, which can be read as an imprecise positioning of an object. An IOU of 1 corresponds to the exact location of the identified object. Consequently, the model is assessed under the presumption that the item placement is extremely accurate by averaging recall values that fall within the range [0.5,1].

9) MEAN AVERAGE RECALL (mAR)

Although AR is generated separately for each class, analogous to how mAP is computed, a single AR value can be determined by taking into account the mean AR across all classes [52], that is:

$$mAR = \frac{1}{N} \sum_{i=1}^{N} AR_i \tag{10}$$

where N is the number of IOU thresholds under consideration.

B. NUSCENES DETECTION SCORE (NDS)

Perhaps the most often used metric for object detection is mAP with a predefined IOU threshold [53]. The nuScenes detection task, such as estimations of position, shape, velocity, and inclination, cannot be fully measured with mAP. They are separated by specifying thresholds for each error category, as in the ApolloScape [54] 3D automobile instance challenge. In this challenge, the number of thresholds is 103, which leads to complicated, arbitrary, and unpredictable mAP. To overcome these limitations, a nuScenes detection score was introduced in [55], which combines the various errors into a scalar value by taking the weighted sum of mAP and several True Positive Metrics (TPM), such as translation, orientation, rotation, attribute, and velocity errors, which are defined as follows:

- Average Scale Error (ASE): Calculated as IOU after aligning centers and orientation.
- Average Translation Error (ATE): Euclidean center distance in 2D in meters.
- Average Orientation Error (AOE): Smallest yaw angle difference between prediction and ground truth in radians. The orientation error is evaluated at 360 degrees for all classes except barriers, where it is only evaluated at 180 degrees. Orientation errors for cones are ignored.
- Average Velocity Error (AVE): The absolute velocity error is measured in m/s. Velocity errors for barriers and cones are ignored.
- Average Attribute Error (AAE): Calculated as an attribute classification accuracy. Attribute errors for barriers and cones are ignored.

The TPM metrics are defined per class and take a mean over classes to calculate mATE, mASE, mAOE, mAVE and mAAE. For example, mTPE over C classes is calculated with eq. 11 and NDS is calculated with eq. 12. Here, TPE stands for a set of five mean true positive metrics and is calulated with eq. 12,[3] the second half of the equation measures the quality of the detection in terms of box position, size, orientation, and velocity, and the first half of NDS is dependent on the detection performance. The range of each metric lies between 0 and 1 because mAVE, mAOE, and mATE can not be larger than 1.

$$mTPE = \frac{1}{C}\sum_{c\varepsilon C} TPE_c \tag{11}$$

$$NDS = \frac{1}{10}[5mAP + \sum_{mTPE\epsilon TP} (1 - min(mTPE))] \tag{12}$$

C. PLANNING KL-DIVERGENCE (PKL)

The computer vision community uses variations of accuracy and precision as the gold standard to assess the performance of perception algorithms. These metrics are widely used since they are basically task-independent and, usually, are aimed at finding zero false positives or negatives of any object detection algorithm. These metrics have the drawback of ignoring objects' position, velocity, and speed. The orientation, location, and environmental characteristics are not taken into account by mAP and NDS. Jonah Philion proposed a novel measure, PKL [56], for 3D object detection that integrates perception performance analysis with driving performance. The main concept of PKL is to analyze detections using a planner that has been trained to plan a driving trajectory using its semantic observations, or detections. If the perception algorithm is flawless, PKL will always return the best result when tested on the nuScenes dataset [57] which is publicly available for indeed researchers. Test results demonstrated that the intuitive ranking of the significance of identifying each car in a scene is induced by the PKL metric, which outperformed traditional metrics [57]. They offer a server for comparing competing object detectors using planning-based metrics, in order to encourage the creation of new perception algorithms that are more in line with the requirements of autonomous driving in the real world.

When a planner is provided with a detection from a detector rather than a human-labeled detection, PKL evaluates the discrepancies between the planner's planning and perception efficiency [56]. It is usually positive, and lower detection performance is correlated with higher PKL scores. An ideal detector is one with a PKL of 0. Several environments for nuScenes detection are used to illustrate the advantage of PKL over mAP. The planner learns how to go through the scenarios by studying a lot of data collected from a human driven system. The local semantic map and the detected bounding boxes serve as conditions for the planner.

D. TIMED QUALITY TEMPORAL LOGIC (TQTL)

The accuracy of perception algorithms was examined using TQTL. It is a formal language for expressing the desired spatio-temporal features of a perception algorithm when processing a video, and it is an extension of Timed Propositional Temporal Logic (TPTL) [58]. The evaluation of a perception algorithm typically involves comparing its performance against labels that represent the real world. TQTL provides an alternative metric that can provide relevant information even in the absence of ground truth labels, making it a helpful tool for assessing perception quality. The phrases ''I'm always hungry,'' ''I'll get hungry eventually,'' and ''I'll be hungry until I eat something'' can be taken as examples of TQTL. A temporal logic with modalities related to time is linear temporal logic (LTL), also known as linear-time temporal logic (LTTL). ''A condition will ultimately be true'', ''a condition will not be true until another fact becomes true'', etc, are a few examples of formulae that can be encoded in LTL to describe the future of pathways. Variables are used in TPTL to calculate the time intervals between two occurrences. For instance, TPTL permits specifying a time limit for the occurrence of an event 'E', whereas LTL only permits stating that each event 'B' is eventually followed by event E. Data stream, frames and data objects, information retrieval function, set of

[3]https://ar5iv.labs.arxiv.org/html/1903.11027

FIGURE 6. Multi-object tracking: ground truth (red) and prediction (blue).

objects function, and scoring function are the rules that make up TQTL.

To elaborate on the effectiveness of TQTL in our object detection problem, we consider the work in [59], in which object detection algorithms such as YOLO and SqueezeDet were trained on different datasets with the same settings [60], such as window frame range, for analysis and to know the impact of TQTL in measuring the performance of detection models. Following are the findings that are observed when using the TQTL metric in addition to other metrics: 1) Both object detection models mistakenly label bikes as pedestrians on multiple occasions. In some cases, the autonomous vehicle plane is orthogonal to the image plane, which leads to a cyclist looking like a pedestrian. This might suggest that there aren't enough images of the cyclist taken right in front of or behind the car in the KITTI dataset. 2) Both algorithms identify objects sporadically, which means they quickly lose faith in their predictions. 3) It has been noted that SqueezeDet finds a number of ''phantom'' items with high confidence before swiftly losing faith in these incorrect predictions.

E. SPATIO-TEMPORAL QUALITY LOGIC (STQL)

Autonomous vehicles perception algorithms are essential to their ability to recognize and track objects in the environment as well as comprehend the semantics of their surroundings. The results of these algorithms are then applied to decision-making in safety-critical situations, like autonomous emergency braking and accident avoidance. It is vital to keep an eye on these perceptual systems while they are in use. The outputs of perception systems are represented in high-level, sophisticated ways, making it difficult to test and validate these systems, particularly during runtime. Authors in [61] introduced PerceMon, a tool for runtime monitoring that can keep track of any specifications in timed quality temporal logic and its extensions with spatial operators. STQL is an extension of TQTL that includes a set of operations on and reasoning about high-level topological entities like bounding boxes that are present in perceptual data. These two are extensions of Metric Temporal Logic (MTL) [62]. In STQL, specifications define a set of operations on the spatial artifacts, like bounding boxes, produced by vision systems, together with operators to reason about classes of objects and discrete IDs. For perception algorithms, the correctness properties can be expressed using TQTL and STQL. PerceMon [61] is an effective online monitoring tool for STQL standards, and it is interconnected with the Robot Operating System (ROS) [63] and the CARLA simulation environment [64].

F. OBJECT DETECTION COMPETITIONS

World-famous competitions for object detection are the VOC PASCAL challenge [65], COCO [66], ImageNet object detection challenge [67], Google open images challenge [68] and Lyft [69]. All these competitions provide their code to calculate average precision, or mean AP, but the Lyft 3D object detection for autonomous vehicles challenge uses the AP averaged over 10 different thresholds, the so-called AP@50:5:95 metric. Submissions for the COCO detection challenge are graded based on metrics divided into four primary categories.

- Average Precision (AP): Several IOUs are used to evaluate the AP. It can be calculated for 10 IOUs that change in percentage by 5% increments from 50% to 95%; this value is typically stated as AP@50:5:95. It can also be assessed using just one IOU value; the most typical values are 50% and 75%, which are reported as AP50 and AP75, respectively.
- AP Across Scales: The AP is calculated for objects of three sizes: small (322 pixels or less in area), medium (322 pixels to 962 pixels), and large (962 pixels or more in area).
- Average Recall (AR): The maximum recall values for an image with a specified number of detections (1, 10, or 100) are used to estimate the AR.
- AR Across Scales: The same three sizes of objects used in the AP across scales are used to determine the AR, which are typically given as AR-S, AR-M, and AR-L, respectively.

V. METRICS FOR MULTI-OBJECT TRACKING (MOT)

It is a process of finding different objects in a video that are of interest, following them in later frames by giving them a distinctive ID, and keeping track of these distinct IDs as the objects move around in the video in later frames, as in Fig. 6. MOT separates a single continuous video into discrete frames at a predetermined frame rate (frames per second). The results of MOT are:

- Detection: Identification of the objects in each frame.
- Localization: locating things in each frame through localization.
- Association: determination of whether items appear to be the same or different in different frames.

By comparing a tracker's predictions to the actual set of tracking results, one may assess the performance of MOT algorithms. Metrics for MOT evaluation must have two important characteristics: 1. MOT evaluation metrics must account for five different types of MOT errors; 2. Error

TABLE 2. Metrics for object detection and tracking.

Paper	Year	Metric	Main characteristics and distinguishing features	Strengths and limitations
Wang et al., [70]	2019	Accuracy	The most straightforward way to assess an object detection model is to calculate the straightforward ratio of the number of accurate predictions to the total number of predictions.	For balanced or nearly balanced datasets, where the proportion of TP and TN is almost equal, accuracy works very well but fails when there is an imbalance in TP and FP.
Padilla et al., [49]	2020	Precision and Recall	These metrics work on TP and FP.	Work better on unbalanced datasets but sometimes fail to measure models' performance due to dependence on fixing the IOU threshold.
Padilla et al., [49]	2020	F1 score	Shows the balance between TP and FP of a model	Useful when the classes are imbalanced and there is a serious downside to predicting false negatives.
Li et al., [71]	2014	Mean average precision	It is the area under the curve (AUC) of the precision and recall curve; Its value lies between 0 to 1 — higher mAP values correspond to better performing models.	It is the most commonly used metric for the object detection problem, but due to the zigzag behavior of precision and recall curves, it is often difficult to calculate the AUC.
Zhang et al., [48]	2022	Interpolated mAP	To avoid the zigzag behavior of the precision and recall curve, precision values are interpolated towards maximum based on the respective recall value	The interpolation of mAP may lead to misclassifications of object classes in some specific cases.
Huang et al., [55]		NDS	Its measurement includes predicted and ground truth bounding boxes, as well as rotation, velocity, position, and translation of objects under test	It overcomes the drawbacks of mAP and interpolated mAP, but it is included only in the nuScenes dataset dashboard out of the publicly available AV datasets.
Philion et al., [56]	2021	PKL	An advanced object detection metric that is based on training the planner on the dataset such that it identifies the object classes that are missed by the object detection model during testing	Its performance depends on the set of rules adopted to train the model planner, and thus fails when rules direct the planner in the wrong direction. Also, it is only included in the nuScenes dataset dashboard.
Balakrishnan et al., [58]	2019	TQTL	It works particularly well for the problem of object tracking and is measured based on the object under test's appearance frame by frame (time).	Up to date, it is not included in any AV's dataset dashboard, and more emphasis on this metric is required
Balakrishnan et al., [61]	2021	STQL	It is also useful for object tracking tasks, in which to be computed, it takes both the time and spatial appearance of the object under test	Up to date, it is not included in any AV's dataset dashboard, and more emphasis on this metric is required
Gao et al., [72]	2021	Decoupled IOU	Fixing IOU threshold is a difficult problem. A Decoupled IOU Regression divides the IOU into two parts: the proportion of the object area in the detected bounding box and the completeness of the detected object area.	Significant improvement in the performance of the model on the MS COCO benchmark.

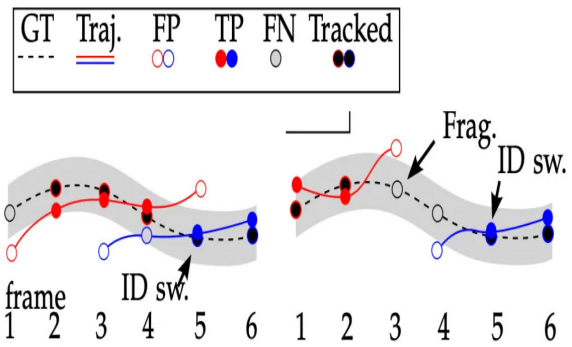


FIGURE 7. Five errors of MOT [73].

kinds should be distinguishable, and MOT evaluation metrics should be monotonic. The five errors are:

- False negative or miss: when there is a ground truth but the prediction is wrong, the result is a false negative or miss.
- False positive: if a tracker prediction exists but there is no ground truth, it is a false positive.
- Merge or ID switch: when two or more object tracks are switched as they pass by one another, this is known as a merge or ID switch.
- Deviation: deviation after re-initializing an object track with a changed track ID.
- Fragmentation: when a track abruptly stops being tracked yet the ground truth track still exists.

In the first part of Fig. 7, an ID switch occurs when the mapping switches from the previously assigned red track to the blue one. In the second part, a track fragmentation is counted in frame N because the target is tracked in frames $N-2$ and $N-1$, then interrupts, and then reacquires its tracked status at a later point with a different ID. Researchers must be aware of many evaluation measures, but selecting the right one is crucial. Determining the contribution of various faults to the final score requires an understanding of each evaluation metric. Understanding the various flaws that go into the evaluation metrics has a significant impact on how to raise MOT ratings and where future research should go. Typical MOT metrics include: Track-mAP, Multi-Object Tracking Accuracy (MOTA), Multi-Object Tracking Precision (MOTP), Safety Score (S), Identification F1-score (IDF1), Higher-Order Tracking Accuracy (HOTA), Detection Accuracy (DetA), Association Accuracy (AssA), Localization, detection error, and association error.

A. LOCALIZATION

Localization measures the spatial alignment between a predicted detection and the actual detection [74]. The

localization accuracy, given by the localization IOU (Loc-IOU), is often used in evaluation metrics. It is calculated as the ratio of the overlap (intersection) between two given detections to the total area covered by both (union). The average Loc-IOU of all matching predicted and real-world detection pairs in the entire dataset is known as localization accuracy (LocA) and is measured with eq.13 over C classes of TP predictions.

$$LocA = \frac{1}{TP} \sum_{C \varepsilon TP} \text{Loc-IOU}(C) \tag{13}$$

B. DETECTION ACCURACY (DetA)

The proportion of the set of predicted detections to the set of all ground-truth detections measures the detection accuracy. This metric is also often expressed by the Detection IOU (Det-IOU), which considers the intersecting predicted and ground-truth detections, after establishing a localization threshold, e.g. Det-IOU > 0.5, as a criterion to accept that an intersection occurs [74].

$$DetA = \text{Det-IOU} = \frac{TP}{TP + FN + FP} \tag{14}$$

When a prediction overlaps with more than one ground truth or vice-versa, the Hungarian algorithm is used to identify a one-to-one match between the predicted detection and ground truth.

C. ASSOCIATION ACCURACY (AssA)

The average alignment between matched trajectories, averaged over all TP detections over C classes, is known as AssA.[4]

$$AssA = \frac{1}{TP} \sum_{C \varepsilon TP} \text{Ass-IOU}(C) \tag{15}$$

D. TRACK-mAP

It matches trajectory-level predictions and ground reality. It requires a trajectory similarity score, *Str*, between trajectories and a threshold, *tr*, with the result that trajectories are only matched if the trajectory similarity score is higher than the threshold. *Str* is calculated by adding the sum of the spatial intersections of all the box intersections divided by the sum of the spatial union of all the box intersections. Track-mAP is non-monotonic and is calculated similarly to *mAP* by knowing false positive and true positive predictions. It is oriented towards quantifying association, and performs both matching and association at the trajectory level. Some difficulties with Track-mAP are:

- Sometimes Track-mAP has numerous overlapping outputs, and some of them have low confidence scores, making it difficult to understand tracking outputs with this method. As a result, the final score for each trajectory is obscured by the implicit confidence ranking, making it difficult to analyze and visualize the results.
- As a result of this metric's high threshold of 0.5 for a trajectory to be considered a positive match, it ignores significant advancements in localization, association, and detection. Any increase in detection and association is not evident in metric scores since even with the best tracking, more than half of its best guess predictions will be reported as errors in Track-mAP.
- The trajectories used by Track-mAP measurements combine association, detection, and localization in a way that makes the error type non-differentiable and non-separable.

[4]https://autonomousvision.github.io/hota-metrics/

E. MULTI-OBJECT TRACKING ACCURACY: MOTA

MOTA continues to be the most accurate measurement that most closely matches human visual evaluation. Matching is carried out at the detection level while calculating MOTA. If the predicted detection (*prDets*) and the ground truth detection (*gtDets*) are sufficiently comparable in space to compute TP, FP, and FN, then one-to-one mapping is created for each frame. When a tracker inadvertently switches object identities or when a track is lost and re-initialized with a different identity, it is called an TD Switch (*IDSW*) in MOTA and is calculated with eq. 16.

$$MOTA = 1 - \frac{FN + FP + IDSW}{gtDets} \tag{16}$$

F. MULTI-OBJECT TRACKING PRECISION (MOTP)

The overlap between all accurately matched predictions and their ground truth is averaged by MOTP. It takes the collection of TP and averages the similarity score (S). In addition to avoiding causing an ID switch to maximize the MOTP score, it matches *prDets* with *gtDets* that have similarity scores above the threshold. The MOTP's behavior is significantly influenced by the threshold. Because MOTP primarily measures the detector's localization accuracy, it doesn't reveal much about the tracker's actual performance. Evaluation measures for tracking systems such as precision in localizing objects, accuracy in recognizing objects, selecting the threshold value, and reliability in tracking objects over time are addressed by MOTP and MOTA. The MOTP is calculated with eq. 17

$$mTP = \frac{1}{TP} \sum_{TP} S \tag{17}$$

G. SAFETY SCORE (S)

To calculate the safety score of an object tracking model, one should give equal importance to precision and accuracy. The tracking safety score (S_t) is the average of MOTA and MOTP. Similarly, object safety score (S_D) is the average of Multi-object Detection Precision (MODP) and Multi-object Detection Accuracy (MODA), which is calculated with eq. 18

$$S_D = \frac{MODA + MODP}{2} \tag{18}$$

H. IDENTIFICATION F1-SCORE (IDF1)

It is used as a supplemental metric on the MOTChallenge[5] benchmark because it places more emphasis on measuring association accuracy than detection accuracy. Unlike MOTA, which matches objects at an object detection level across time, *IDF*1 determines whether trajectories are present by computing one-to-one mapping between ground truth and prediction trajectories. *IDF*1 is the proportion of accurately identified detections over the mean number of ground-truth detections.

$$ID - Recall = \frac{IDTP}{IDTP + IDFN} \tag{19}$$

$$ID - Precision = \frac{IDTP}{IDTP + IDFP} \tag{20}$$

$$IDF1 = \frac{IDTP}{IDTP + 0.5\ IDFN + 0.5\ IDFP} \tag{21}$$

where, *IDFN* (Identity False Negative) and *IDTP* (Identity True Positive) are calculated based on similarity and dissimilarity between *gtID* and *prID*, respectively. *IDFP* (Identity False Positive) characterizes the remaining *prID* trajectories that are not matched with any *gtID*. A high *IDF*1 score provides information regarding good detection or association, but it also predicts the overall number of distinct objects in a scene. Moreover, it does not assess the trackers' localisation precision. *IDF*1 combines ID-Precision and ID-Recall.

I. HIGH ORDER TRACKING ACCURACY (HOTA)

A single unifying metric called *HOTA* explicitly assesses tracking-related errors, including precise detection, association, and localization. All assessment metrics, including MOTA, IDF1, and HOTA use the Jaccard Index, or IOU score, which assesses their spatial similarity; incorrect predictions are penalized. Three IOU scores can be combined to form the HOTA. It breaks the evaluation of the tracking error into the three subtasks, i.e., detection, association, and localization, and uses an intersection over union formulation to determine a score for each. The total HOTA score is then calculated by combining these three IOU values for each subtask as in Eq 22. The Hungarian algorithm [75] is used to generate a bijective mapping between each pair of *gtDet* and *prDet* in order to identify the match that maximizes the total matching score. In order to gain insight into the various kinds of tracking errors that trackers are generating, HOTA decomposes into a series of sub-metrics that allow independent examination of various tracking errors.

$$HOTA = \text{Det-IOU} + \text{Ass-IOU} + \text{Loc-IOU} \tag{22}$$

where Det-IOU, Ass-IOU and Loc-IOU are *IOU* scores of detection, association, and localization, respectively.

J. DETECTION ERROR

A detection error occurs when a tracker either incorrectly anticipates detections in the ground truth or incorrectly predicts detections that are present in the ground truth. Other types of detection errors include detection recall (measured by FNs) and detection precision (measured by FPs).

[5]https://motchallenge.net/

K. ASSOCIATION ERROR

It occurs when trackers assign two detections with distinct *gtIDs* with the same *prID*, or two detections with identical *gtIDs* with different *prIDs*. Association error can also be divided into association recall errors (measured by FNs) and association precision errors (measured by FPs).

L. SPATIO-TEMPORAL TUBE AVERAGE PRECISION (STT-AP)

All of the above-mentioned metrics are applied to an individual image or frame. The predictive accuracy at the level of the entire video may be relevant when working with videos. The STT-AP is an extension of the AP metric to assess video object detection models. Similar to AP, the accuracy of the detection is evaluated using a threshold above the IOU. Nevertheless, it broadens the conventional IOU definition to take into account the spatio-temporal tubes produced by the detection and the ground truth rather than utilizing two different kinds of overlaps (spatial and temporal). This metric is brief but evocative because it combines spatial and temporal localization. Spatio-temporal tube IOU (STT-IOU) is the ratio of ground truth to predicted spatio-temporal tube. This way, if the STT-IOU is equal to or higher than a specified threshold, a detection is treated as a TP.

VI. METRICS FOR SEMANTIC SEGMENTATION

The technique of grouping point clouds into various homogeneous regions, each containing points with similar characteristics, is known as 3D point cloud segmentation. As point cloud data has high levels of redundancy, irregular sample densities, and a lack of explicit structure, segmentation is difficult. These problems are addressed by several researchers in the field of robotics applications, including autonomous vehicles, self-driving cars, and navigation. There are three types of segmentation techniques that play a crucial role in relation to autonomous vehicles: semantic, instance, and panoptic segmentation [76]. These three are labeled differently based on the labeling of things/countable objects (trees, cars, pedestrians, etc.) and stuff/non-countable objects (road, gross, sky, etc.) in an image. For a better understanding and visual appearance of these three, see Fig. 8.

Every pixel in an image is assigned a class label using semantic segmentation, such as a person, flower, car, etc. Several objects belonging to the same class are treated as a single entity. Semantic segmentation methods that are frequently employed include Fully Connected Network (FCN) [77], DeconvNet [78], U-Net [79], and SegNet [80]. Comparatively speaking, instance segmentation treats several objects belonging to the same class as unique individual instances. Frequently used instance segmentation methods include PANet [81], Faster R-CNN [82], Mask R-CNN [83], and YOLACT [84].

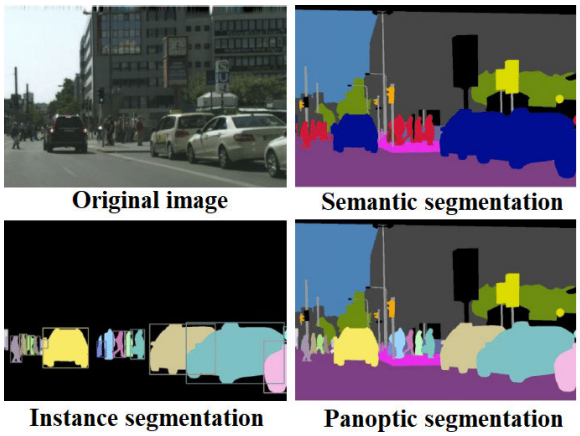


FIGURE 8. Three segmentation visual difference.

Each pixel in an image receives two labels from panoptic segmentation: a semantic label and an instance ID. The similarly marked pixels are regarded as being members of the same semantic class, and its instances are identified by their unique identifiers (IDs). The Mask R-CNN [83] approach is the foundation of most panoptic segmentation methods. The architectures that make up its backbone include VPSNet [85], EPSNet [86], FPSNet [87], and UPSNet [88].

A. EVALUATION METRICS

Each segmentation method evaluates the expected masks or IDs in a scene using a different set of evaluation measures. This is due to the diverse ways in which things and items are processed.

B. METRICS FOR SEMANTIC SEGMENTATION

The goal of establishing metrics for semantic segmentation is to score the similarity between the predicted (prediction) and annotated segmentation (ground truth). The mainly used ones are: Dice coefficient, Jaccard Index (or IOU), pixel accuracy, and mean accuracy.

1) DICE COEFFICIENT

It is equal to two times the intersection of the predicted (*Pseg*) and ground truth (*GTseg*), divided by the sum of *Pseg* and *GTseg* segmentation [89]. It is also referred to as the Sørensen-Dice coefficient, is equivalent to the F1 score, and is calculated with eq. 23.

$$Dice = 2|Pseg \cap GTseg|/(|Pseg| + |GTseg|) \tag{23}$$

Keep in mind that the area of the union of *Pseg* and *GTseg* in eq. 23 differs from the sum of the areas of *Pseg* and *GTseg* in eq. 25. Specifically, one is twice the other if there is 100% overlap. This is the reason why the multiplication by two is included in the Dice coefficient. They are both defined so that their values are 1 and 0, with 100% overlap and 0% overlap, respectively.

In general, most of the researchers are using IOU for the object detection evaluations and Dice for the semantic segmentation case, even if both have similar metrics. Which one to use depends on personal preferences and convention. In segmentation tasks, the Dice loss (eq. 24) is used as a loss function because it is differentiable where IOU is not differentiable. The IOU and Dice can be used as metrics to assess the model's performance, but only Dice loss is used as a loss function.

$$Dice\ loss = 1 - Dice\ coeff \tag{24}$$

2) JACCARD INDEX

The Jaccard Index, which measures how close the anticipated and actual masks are, is widely used in semantic segmentation. It is also commonly known as an IOU and is calculated after dividing the intersection's area by the union's area.

$$Jaccard = TP/(TP + FP + FN) \tag{25}$$

3) MEAN PIXEL ACCURACY (mPA)

The quantity of pixels accurately categorized in the resulting segmentation mask is known as pixel accuracy (*PAseg*). It might be the easiest statistic for assessing performance, but it may not genuinely take into account the model's performance. When there is a significant class imbalance in the dataset, the pixel accuracy metric always becomes skewed. Even with 90% accuracy, sometimes, qualitative performance would still be subpar. This statistic determines the proportion of pixels that can be accurately identified among all the pixels in the image.

$$PAseg = \frac{\sum_{J=1}^{C} TP_J}{\sum_{J=1}^{C} T_J} \tag{26}$$

where TP_J is the total of true positives observed in the J^{th} class and T_J is the total number of pixels labeled as J^{th} class. Since semantic segmentation involves numerous classes, the mean pixel accuracy serves as a representation of the class average accuracy as

$$mPA = \frac{1}{C}\frac{\sum_{J=1}^{C} TP_J}{\sum_{J=1}^{C} T_J} \tag{27}$$

4) AVERAGE HAUSDORFF DISTANCE (AHD)

It is a popular performance metric that determines the distance between two point sets. It is used to compare labels with detected or segmented images and to rate various detection/segmentation outcomes. The AHD is particularly well suited for segmentation involving complex boundaries and narrow segments. Unlike the Dice coefficient, AHD takes voxel localization information into account. The AHD between two point clouds *p*,*q* is calculated with eq. 28.

$$AHD(P, Q) = \frac{1}{|P|}\sum_{x \in Q} \min_{y \in P_z} d(P, Q) + \frac{1}{|Q|}\sum_{y \in Q_z} \min_{x \in Q} d(P, Q) \tag{28}$$

C. METRICS FOR INSTANCE SEGMENTATION

The typical evaluation statistic for instance segmentation is the average precision (*APinst*). For every instance of an item, the *APinst* metric employs the *IOUinst* on a pixel-by-pixel basis. The misclassification, the degree of prediction confidence, and the size of the regions all have an impact on the instance segmentation metrics that are currently in use. Hence, instead of focusing on confidence or semantics, in [90] a novel evaluation measure is provided that can be applied to input regions of any size and concentrates on the objects' distinguishing ability. The overlap between prediction and label might be fully (*TP*), partially (*PD*) or no overlap (*FN*). To deal with this issue, the intersection over a set (*IoS*) is introduced [90], which states the portion of a prediction (*Pinst*) that is contained in a ground truth (*GTinst*):

$$IoS(P, GT) = \frac{N(Pinst \cap GTinst))}{N(Pinst)} \tag{29}$$

Pinst is regarded as being contained in *GTinst* when $IoS(Pinst, GTinst)$ is greater than a specific threshold, and $N(h)$ specifies the number of points in h.

D. METRICS FOR PANOPTIC SEGMENTATION

As a brand-new task, panoptic segmentation was originally put forth in [76]. In this method, background classes are segmented using semantic segmentation, while foreground classes are segmented using instance segmentation. These two categories are also known as stuff/countable classes and things/non-countable classes, respectively. The Panoptic Quality (*PQ*) metric assesses anticipated masks and instance identifiers for both countable and non-countable objects in an image. The *PQ* combines segmentation quality (*SQ*) and recognition quality (*RQ*) criteria to create an evaluation that is consistent across all classes. The *SQ* score is the average *IOU* score of the matched segments, and the *RQ* score is the F1 score determined by applying the precision and recall values of the predicted masks. $PQ^{\dagger}$ is determined by converting each item class's *PQ* to its corresponding *IOU* and averaging the results across all classes like *PQ* does. These measures are also carried out independently on the two groups that make up the categories in panoptic segmentation, namely stuff and things. So metrics include PQ^{Th}, PQ^{St}, SQ^{Th}, SQ^{St}, RQ^{Th} and RQ^{St}.

$$PQ = \frac{\sum_{(popt,gopt)\epsilon TP} IOU(popt, gopt)}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|} \tag{30}$$

where *popt* and *gopt* are predicted and groud truth panoptic segmentation respectively. The *IOU* ratios for each of *TP* values are added together to calculate the panoptic quality as in eq. 30. Divide all *TP* and half of *FN* & *FP* at the bottom to achieve a happy medium between recall and precision. To understand this metric even better, consider it divided into two parts: *SQ* and *RQ* as in eq. 31

$$PQ = SQ \times RQ = \frac{\sum_{(popt,gopt)\epsilon TP} IOU(popt, gopt)}{|TP|} \times \frac{|TP|}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|} \tag{31}$$

The *SQ* metric measures how closely predicted segments reflect their underlying realities. When this value comes closer to 1, it signifies that *TP* projected segments are more closely aligned with their ground truth. That does not, however, explain any of the incorrect predictions. The *RQ* enters the picture at this point. This statistic combines accuracy and recall in an effort to assess how well models make accurate predictions. There is a need for confidence scores in order to rank predictions from highest to lowest, which will allow us to create a precision/recall graph. Unfortunately, as panoptic segmentation includes instance and semantic segmentation, there is a lack of definitive confidence scores for semantic predictions.

A fundamental *IOU* measure also has certain problems. With semantic segmentation, there is just one segment and one ground truth to compare for each class, although we can have many instances of the same class and multiple ground truths in panoptic segmentation. The problem of correctly matching the predicted segment to the appropriate ground truth is resolved by segment matching. It adheres to two fundamental tenets:

- No single pixel can simultaneously belong to two predicted segments or overlapping predictions.
- Only predicted segments whose *IOU* with the ground truth is greater than 0.5 can be matched with the ground truth.

1) PARSING COVERING (PC) METRIC

This metric is an extension of the covering metric [91] proposed in [92]. The covering metric is mostly useful for the evolution of class-agnostic segmentation. In some applications, such as portrait segmentation (referring to the process of segmenting a person in an image from its background) or autonomous driving (where near objects are more significant than far-off ones), one should pay more attention to large objects. This inspired the authors to propose the *PC* metric [92], which takes instances or objects' sizes into consideration. The following definition applies to *PC* calculation.

$$PC_i = \frac{1}{M_i} \sum_{R\epsilon s_i} |R| \max_{R'\epsilon s'_i} IOU(R, R') \tag{32}$$

$$M_i = \sum_{R\epsilon S_i} |R| \tag{33}$$

$$PC = \frac{1}{C} \sum_{i=1}^{C} PC_i \tag{34}$$

where S'_i and S_i are the predicted and ground truth segmentations of the i^{th} semantic class, respectively. M_i is the total

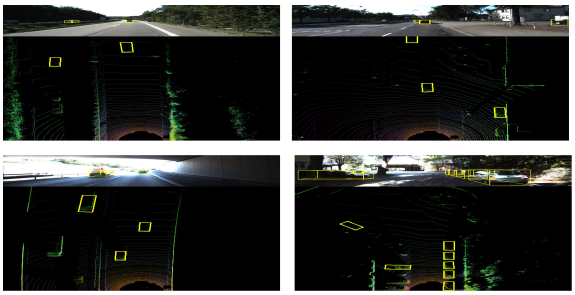


FIGURE 9. Outcome of Complex YOLOv4 on KITTI dataset.

number of pixels in the ground truth semantic class of S_i. PC_i is calculated in the same manner as the original covering metric, with the exception that only the ground truth S_i and predicted regions from S_i' are taken into account. Then, PC is calculated by averaging PC_i across C semantic classes. In order to assess image parsing outcomes, the PC is a straightforward extension of the covering. Covering does not penalize the erroneous positives, as was noted in [93]. The background class's coverage is not assessed, absorbing false positives from other classes. This won't happen in the case of image parsing because all classes and every pixel will be taken into consideration.

There is also no matching involved in PC, hence, there is no matching threshold, which is another significant distinction between PQ and PC. The segmentation of ''stuff'' classes nevertheless obtains a partial PC score if the segmentation is only partially accurate in an effort to treat ''thing'' and ''stuff'' equally. For instance, whether the model considers ''tree'' as ''stuff'' or ''thing,'' it will receive the same partial score by utilizing PC if one out of three equally-sized trees is perfectly segmented.

E. DATASET - PANOPTIC SEGMENTATION

1) SemanticKITTI [94]

It is a sizable dataset for driving scenes that can be used for panoptic and semantic point cloud segmentation [94]. Data was gathered in ''Germany'' using a Velodyne-HDLE64 LiDAR and is derived from the KITTI Vision Odometry Benchmark. The dataset consists of 22 sequences, which are divided into a training set (using sequence 08 as the validation set) and a test set (using sequences 11 to 21). After combining classes with varied movement statuses and ignoring classes with very few points, 19 classes are still available for training and evaluation.

2) NUSCENES [57]

It uses a 32-beam LiDAR sensor to gather 1,000 scenes with a 20-second duration. It comprises 40,000 frames in total, each of which is sampled 20 times per second. Additionally, they formally divided the data into a training set and a validation set. A total of 16 classes for the LiDAR semantic segmentation are left after combining comparable classes and deleting unusual classes. A cylindrical partition divides these point clouds from the two datasets into 3D representations with the dimensions 32, 360, and 480, where the three dimensions denote the height, angle, and radius, respectively.

3) CITYSCAPES [95]

It contains 5000 images of egotistical driving situations in metropolitan locations (2975 training sets, 500 validations, and 1525 tests). There are 19 classes with dense pixel annotations (97% coverage), of which 8 have instance-level segmentation.

4) ADE20k [96]

With an open-dictionary label set, it has over 25k images (20k for the training set, 2k for validation, and 3k for the test). In order to cover 89% of all pixels, 100 things and 50 stuff classes were chosen for the 2017 Places Challenge.[6]

5) MAPILLARY VISTAS [97]

It offers 25k street-view images in a variety of resolutions (18k for the training set, 2k for validation, and 5k for the test). The dataset has a 98% pixel coverage density annotation with 28 stuff and 37 things classes.

VII. RESULTS AND DISCUSSIONS

After the description of the various metrics that have been proposed for object detection, multi-object tracking, and panoptic segmentation model architecture, this section provides an evaluation of their performance and influence on models after reproducing previously published results or carrying out new tests with trained models from Github and testing them with the nuScence and KITTI datasets.

A. COMPLEX-YOLOV4 AND COMPLEX-YOLOV3

Due to its direct connection to environmental comprehension and subsequent creation of the foundation for prediction and motion planning, LiDAR-based 3D object detection is unavoidable for autonomous vehicles. A poorly stated problem for many other application fields besides autonomous vehicles, such as augmented reality, personal robots, or industrial automation, is the ability to infer highly sparse 3D data in real-time. The authors in [98] presented Complex-YOLO, a cutting-edge real-time 3D object identification network that exclusively works with point clouds. YOLOv2 [99], a quick 2D standard object detector for RGB images, is expanded in [98] by a network that uses a very sophisticated regression method to estimate multi-class 3D bounding boxes in the cartesian space. As a result, they suggest a particular Euler-region proposal network to calculate the object's posture by incorporating imaginary and real terms into the regression network. This eliminates singularities, which are caused by single-angle estimations, and results in a closed complex space. For the application of AVs, a comparison of complex-YOLO versions 3, 4, and 5 is given in [100]. Along with a theoretical description of complex YOLO, we

[6]https://places-coco2017.github.io/

TABLE 3. Metrics for segmentation.

Paper	Year	Metric	Main characteristics and distinguishing features	Strengths and limitations
Metrics for semantic segmentation				
Jha et al., [89]	2019	Dice	Is the ratio of intersection and union of predicted and ground truth	It is the most commonly used metric for semantic segmentation, especially in medical image segmentation. In place of it, IOU can also be useful but lacks gradient when used as a loss function when training the model
Jha et al., [89]	2019	Mean pixel accuracy	It determines the proportion of pixels that can be accurately identified among all of the pixels in the image	It is similar to mAP and has the same issue with the zigzag shape of the precision and recall curve
Metrics for instance segmentation				
Arase et al., [90]	2019	Average precision (AP)	It is calculated based on IOU calculation on a pixel-by-pixel basis	Its performance depends on misclassification, the degree of prediction confidence, and the regions' size
Metrics for panoptic segmentation				
Kirillov et al., [76]	2019	Panoptic Quality	It is a combination of segmentation quality and recognition quality	Its performance depends on how effectively the model works, for instance and semantic segmentation
Yang et al., [92]	2019	Parsing Covering	In addition to measures taken into account while calculating this metric, the size of the object is also considered	It is an extension of the covering metric, which is mostly useful for the evolution of class-agnostic segmentation.

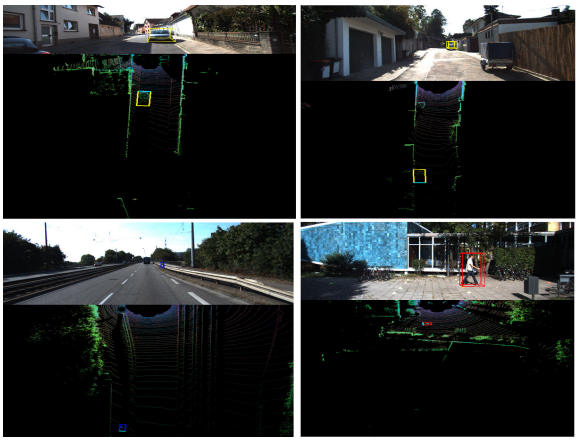


FIGURE 10. Outcome of Complex YOLOv3 on KITTI dataset.

produced the results of trained YOLOv4[7] and YOLOv3[8] on the KITTI dataset. Figs. 9 and 10 show the outcomes of YOLOv4 and YOLOv3 on the KITTI dataset, respectively. Keep in mind that the predictions in Fig. 9 are only based on aerial images created from point clouds.

B. POINTPILLARS 3D OBJECT DETECTION

A new encoder called PointPillars uses PointNets [101] to train itself how to represent point clouds in the form of vertical columns (pillars). PointPillars predicts positioned 3D boxes of vehicles, pedestrians, bicycles, etc., using point clouds as input. There are three primary phases: A point cloud is first transformed into a sparse pseudo-image by a feature encoder network, then the pseudo-image is processed into a high-level representation by a 2D convolutional backbone, and finally a detection head detects and regresses 3D boxes. While any common 2D convolutional detection architecture can employ the encoded features, it also uses a lean downstream network.

One of the earliest techniques to use PointNets for object detection with LiDAR point clouds is VoxelNet [102]. Here, voxels are subjected to PointNets before being processed by a group of 3D convolutional layers, a 2D backbone, and a detection head. This makes end-to-end learning a possibility, but VoxelNet is cumbersome; it takes 225 ms of inference time (4.4 Hz) for a single point cloud, which is slower than prior work [101]. This issue was resolved in Frustum PointNet [103] and the speed of detection increased further with a detector called SECOND [104].

In Tables 4 and 5, FP16 denotes the adoption of the Mixed Precision (FP16) in training. Using 8 Titan XP GPUs with a batch size of 2, PointPillars are trained with the nuScenes dataset using mixed precision training [101]. Without this mixed-precision training, out-of-memory (OOM) errors would result. On the nuScenes dataset, the loss scale

TABLE 4. Per-class results: PointPillars with SECFPN (FP16) network as backbone.

Object Class	AP	ATE	ASE	AOE	AVE	AAE
Car	0.797	0.207	0.161	1.527	0.228	0.144
Truck	0.548	0.258	0.224	1.577	0.124	0.541
Bus	0.7	0.423	0.175	1.387	1.01	0.2
Trailer	0.962	0.37	0.396	1.642	0.056	1
Construction_vehicle	0	1	1	1	1	1
Pedestrian	0.764	0.143	0.272	1.455	0.26	0.045
Motorcycle	0.085	0.467	0.505	1.77	0.116	0.087
Bicycle	0.172	0.257	0.251	1.427	0.645	0
Traffic_cone	0	1	1	NaN	NaN	NaN
Barrier	0.427	0.725	0.395	0.872	NaN	NaN

[7]https://github.com/maudzung/Complex-YOLOv4-Pytorch
[8]https://github.com/ghimiredhikura/Complex-YOLOv3

TABLE 5. Per-class results: PointPillars with FPN (FP16) network as backbone.

Object Class	AP	ATE	ASE	AOE	AVE	AAE
Car	0.792	0.217	0.341	1.513	0.219	0.136
Truck	0.409	0.803	0.371	1.588	0.152	0.393
Bus	0.655	0.505	0.45	1.275	0.831	0.112
Trailer	0.892	0.455	0.64	1.617	0.03	1
Construction_vehicle	0	1	1	1	1	1
Pedestrian	0.868	0.142	0.343	1.439	0.228	0.053
Motorcycle	0.202	0.31	0.349	1.549	0.104	0.066
Bicycle	0.26	0.242	0.29	1.502	0.583	0.019
Traffic_cone	0.008	0.687	0.41	NaN	NaN	NaN
Barrier	0.485	0.466	0.974	0.8	NaN	NaN

TABLE 6. Per-class results: SSN for multi-class object detection from point clouds.

Object Class	AP	ATE	ASE	AOE	AVE	AAE
Car	0.846	0.193	0.152	0.134	0.192	0.128
Truck	0.685	0.253	0.196	0.017	0.118	0.526
Bus	0.771	0.358	0.14	0.031	0.747	0.09
Trailer	0.989	0.232	0.287	0.011	0.015	1
Construction_vehicle	0	1	1	1	1	1
Pedestrian	0.842	0.15	0.256	0.263	0.246	0.085
Motorcycle	0.428	0.259	0.254	1.091	0.104	0.058
Bicycle	0.406	0.225	0.247	0.437	0.607	0
Traffic_cone	0	0.24	0.391	NaN	NaN	NaN
Barrier	0.749	0.589	0.217	0.044	NaN	NaN

for PointPillars is precisely calibrated to prevent the loss from being excessive. Experiments show that 32 is more stable than 512, while loss scale 32 occasionally causes NaN problems. This is the reason for NaN in Tables 4, 5 and 6 for some classes.

C. POINTPILLARS - FEATURE PYRAMID NETWORK (FPN)

By using a top-down approach and lateral connections, FPN mixes semantically robust features with high-resolution ones and semantically weak features with low-resolution ones. In FPN, a feature pyramid is generated quickly from a single input image scale and has extensive features at all levels without losing representational power, speed, or memory. Other concurrent works, such as the deconvolutional single Shot setector [105], also employ this strategy. The feedforward computation of the backbone ConvNet is the bottom-up pathway as in Fig. 11. Every stage in FPN has its own pyramid level, and the final layer of each stage's output will serve as the reference set of feature maps for lateral connections. The feature maps from higher pyramid levels are upsampled to produce high resolution features that are geographically coarser but semantically stronger. To be more precise, for ease of use, the spatial resolution is upsampled by a factor of two using the nearest neighbor. Each lateral connection combines feature maps from the top-down and bottom-up pathways that are of the same spatial size. To specifically decrease the channel dimensions, 1×1 convolutions are applied to the feature maps from the bottom-up pathway, and through element-wise addition, the feature maps from the top-down and bottom-up pathways are combined.

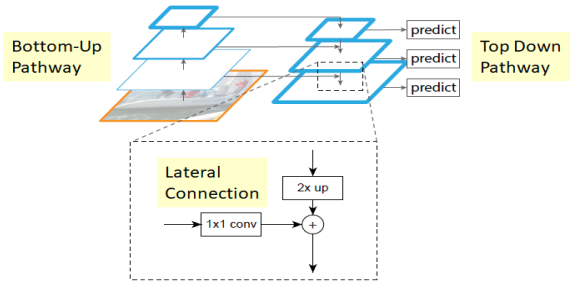


FIGURE 11. Architecture of FPN [105].

D. POINTPILLARS - SECOND FEATURE PYRAMID NETWORK (SECFPN)

Robot vision and autonomous driving are two examples of applications that make use of RGB-D or LiDAR-based object detection. Since a while ago, point cloud LiDAR data processing has made use of voxel-based 3D convolutional networks to improve information retention. Yet, issues such as sluggish inference speed and poor orientation estimation performance persist. In order to considerably speed up both training and inference, [104] investigated an enhanced sparse convolution approach for such networks. In order to increase the performance of orientation estimation, a new type of angle loss regression was introduced. Also, presented a fresh method for data augmentation that can boost convergence performance and speed. The SECFPN network maintains a high inference speed while delivering cutting-edge performance on the KITTI 3D object detection benchmark, as shown in Table 4.

Fig. 12 shows the components of SECOND detector. A raw point cloud is fed into the SECOND detector, which then transforms into voxel features and coordinates before applying two VFE (voxel feature encoding) [102] layers and a linear layer. A sparse CNN is then used. Lastly, the detection is produced by a Region Proposal Network (RPN) [106]. To extract voxel-wise features, VFE is used. A VFE layer uses FCN made up of a linear, a batch normalization (BatchNorm), and a rectified linear unit (ReLU) layer to extract pointwise information from all the points in a single voxel. All atomic operations relating to the convolution kernel elements are gathered by sparse convolution and saved as computation instructions in a rulebook. Fully convolutional networks that anticipate object limits and objectness scores at each place are known as RPNs. To provide top-notch regional proposals, the RPN receives comprehensive training.

The primary distinction between the shape-aware grouping heads and the original SECFPN heads is that the former groups objects of comparable sizes and shapes together while designing shape-specific heads for each group. Longer strides and more convolutions are seen in heavier heads, which are made for handling heavy things. Smaller heads are made for handling light objects. Keep in mind that the outputs could contain feature maps of various sizes; therefore, the solution must also include an anchor generator that is appropriate for feature maps.

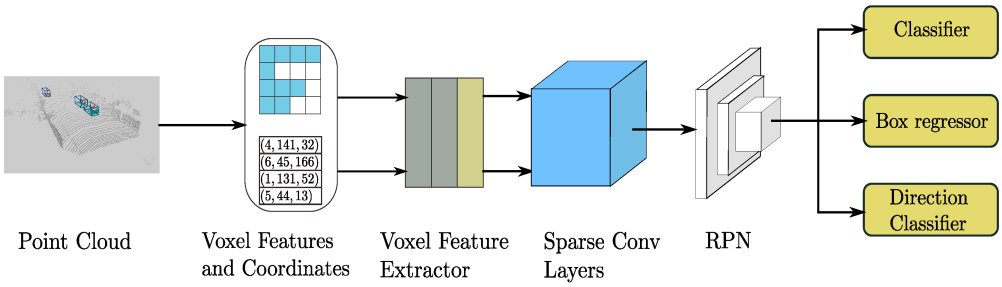


FIGURE 12. Architecture of SECFPN [104].

TABLE 7. nuScenes detection score obtained with models.

Metrics	SECFPN (FP16)	FPN (FP16)	SSN
mAP	0.4456	0.457	0.5716
mATE	0.485	0.4827	0.35
mASE	0.438	0.5168	0.3138
mAOE	1.4065	1.3649	0.3364
mAVE	0.4299	0.3933	0.3787
mAAE	0.3772	0.3473	0.361
NDS	0.4498	0.4545	0.6118
Eval time	1.6s	2.1s	1.6s

E. SHAPE SIGNATURE NETWORKS (SSN) FOR MULTI-CLASS OBJECT DETECTION FROM POINT CLOUDS

Finding and classifying objects from point clouds that fall into different categories is the goal of multi-class 3D object detection. Shape information is one feature that can help with multi-class discrimination but is underutilized because point clouds are, by their very nature, sparse, unstructured, and noisy. So, authors in [107], proposed 3D shape information from point clouds using a unique shape signature. By including a convex hull, symmetry, and Chebyshev filter, the proposed shape signature is not only efficient and compact but also noise-resistant, acting as a soft constraint to enhance the feature capability of multi-class discrimination. The created shape signature network is composed of explicit shape encoding objectives, shape-aware grouping heads, and pyramid feature encoding for 3D object detection. In this review paper, we employed shape-aware grouping heads of SSN as the backbone in PointPillars, and results are produced on nuScenes as in Table 6. Finally, Table 7 shows the evaluation time and mean average of true positive metrics obtained with SECFPN (FP16), FPN (FP16) and SSN.

F. POINT CLOUD DISTANCE METRIC

To assess the quality of a match between two point clouds, we used various distance measures. To visualize point clouds, open3D[9] was employed. Standard methods offered in Numpy and Scipy are used to create the distance metrics. We generated one point cloud randomly with 100 points, and it was shifted along [x,y,z] axis to generate another point cloud as in Fig. 13. For each point cloud, the measured nearest neighbor

FIGURE 13. Two point clouds for distance measure.

TABLE 8. Distance metrics with 3 shifts.

Distance Metric	S1	S2	S3
HD	0.7167	0.7558	1.0306
MHD	0.3155	0.3583	0.6571
CD	0.2336	0.2829	0.8598
EMD	0.4895	0.7845	0.9841

distances can be shown as a distribution in Fig. 14. Because point clouds have varying degrees of spatial resolution, accuracy, and outlier characteristics, the distributions may differ. We opted for three shifts along [x,y,z] axis for calculation of distances, i. e., $S1 = [0.50, 0.50, 0.1]$, $S2 = [0.40, 0.60, 0.2]$ and $S3 = [0.80, 0.10, 0.6]$. The measured Hausdorff Distance, modified Hausdorff Distance, Earth Mover's Distance, and Chamfer Distance between two point clouds with shifts S1, S2 and S3 are tabulated in Table 8 and their respective codes are publicly available.[10]

G. PLANNING KL-DIVERGENCE (PKL)

In order to explain the effectiveness of PKL over NDS, we used a trained MEGVII [108] point cloud 3D object detection model. In this model, sparse 3D convolutions [109] were used to extract rich semantic features, which were subsequently input into a class-balanced multi-head network. Class-balanced sampling and augmentation techniques were used to address the significant class imbalance problem inherent in autonomous vehicles, and balanced grouping heads improved the results for groups with comparable forms. Classes (car, bicycle, pedestrian, etc.,) with comparable shapes or sizes can cooperate with one another according to

[9]http://www.open3d.org/

[10]https://github.com/UP-RS-ESP/PointCloudWorkshop-May2022/tree/main/2_Alignment

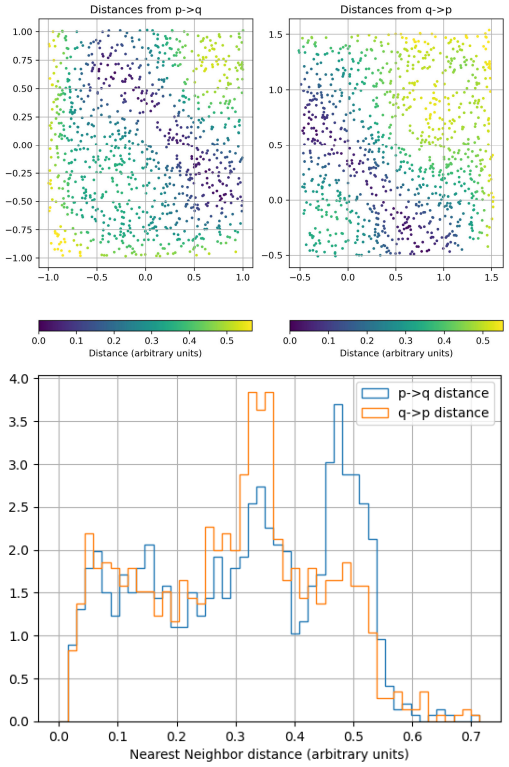


FIGURE 14. Visualizing distances in space and as distribution.

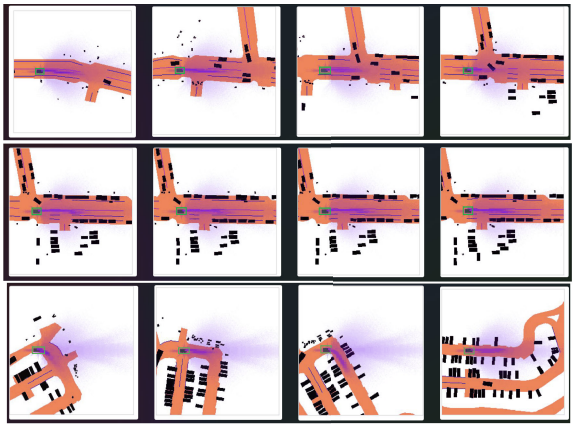


FIGURE 15. Some examples of predictions.

the multi-group head network's design, whereas categories with dissimilar shapes or sizes stop interfering with one another. Sub-manifolds and standard 3D sparse convolutions make up the 3D Feature Extractor. The outputs of the 3D Feature Extractor have a 16:1 downscale ratio and are flattened along the output axis before being input into the region proposal network that follows to produce 8:1 feature maps and the multi-group head network that follows to produce the final predictions. According to the grouping specification, the number of groups in the head is set. The main goal of PKL is to mark the false positives and false negatives of the object detection model as in Fig. 15. In general, a falsely detected parked vehicle will not lead to dangerous maneuvers by the AV, while a FP in front of it will. Metrics like mAP and NDS treat both of these cases in a similar way and rank the MEGVII object detection model, but PKL treats them in different cases and assigns a rank to the MEGVII object detection model accordingly. A model with a higher PKL value shows the worst performance as in Fig. 16 and a lower one shows better performance as in Fig. 17.

Fig. 15 shows some examples of pretrained planner predictions on the nuScenes test dataset. The pre-trained planner can be found at the link shown below.[11] The ground truth, predictions, and PKL in which the reported MEGVII detections perform the worst according to the PKL measure are shown in Fig. 16. The objects in front of the ego vehicle appear to be traveling backwards due to a FP that appears in front of the ego vehicle. Because of this, the planner anticipates that the ego vehicle will halt instead of moving ahead, which would incur a severe penalty under the PKL metric. The time interval where MEGVII performs the best under PKL is depicted in Fig. 16. The car to the left of the ego is consistently detected in the time sequence. Although there are a number of FP human detections in the scene, the task of waiting at the light is unaffected by these detections, so the scene still functions properly. Recognizing the people on the sidewalk accurately is an essential subtask for some downstream tasks, such as autonomous vehicles. Our objective is not to promote the use of PKL exclusively for object detector evaluation, but rather to suggest PKL as an alternative to task-agnostic metrics that do not take into consideration the environment in which perceptual errors occur. In Fig. 17, the green-colored object is the ego car, the red color is a FP, and the pink color is a FN.

H. TIMED QUALITY TEMPORAL LOGIC (TQTL)

In this section, the impact of TQTL on two object detection models is discussed. We used pre-trained weights found in the code repository run by the creators of the original SqueezeDet[12] and YOLOv3[13] was assessed. Both models are trained on the KITTI object detection dataset with a total of nine classes, for example, cycle, van, misc., etc. Both models were trained for 1000 epochs on a GPU-compatible device, and it took 9 and 12 hours to train SqueezeDet and YOLOv3. A portion of the KITTI raw dataset was used to monitor the data streams produced by these two models in comparison to the TQTL specifications [59].

One of the specifications verifies that if the object detection algorithms identify bicycles in any frame x with a probability greater than 0.7, the likelihood that the object is a cyclist won't drop below 0.6 over the following 5 frames. With this specification, it is noted that SqueezeDet and YOLOv3 mistakenly classify bikers as pedestrians, as illustrated in Fig. 18. This might be explained by the fact that the bicycle is less obvious in the images when the rider's path is practically

[11] https://github.com/nv-tlabs/planning-centric-metrics
[12] https://github.com/BichenWuUCB/squeezeDet
[13] https://github.com/ghimiredhikura/Complex-YOLOv3

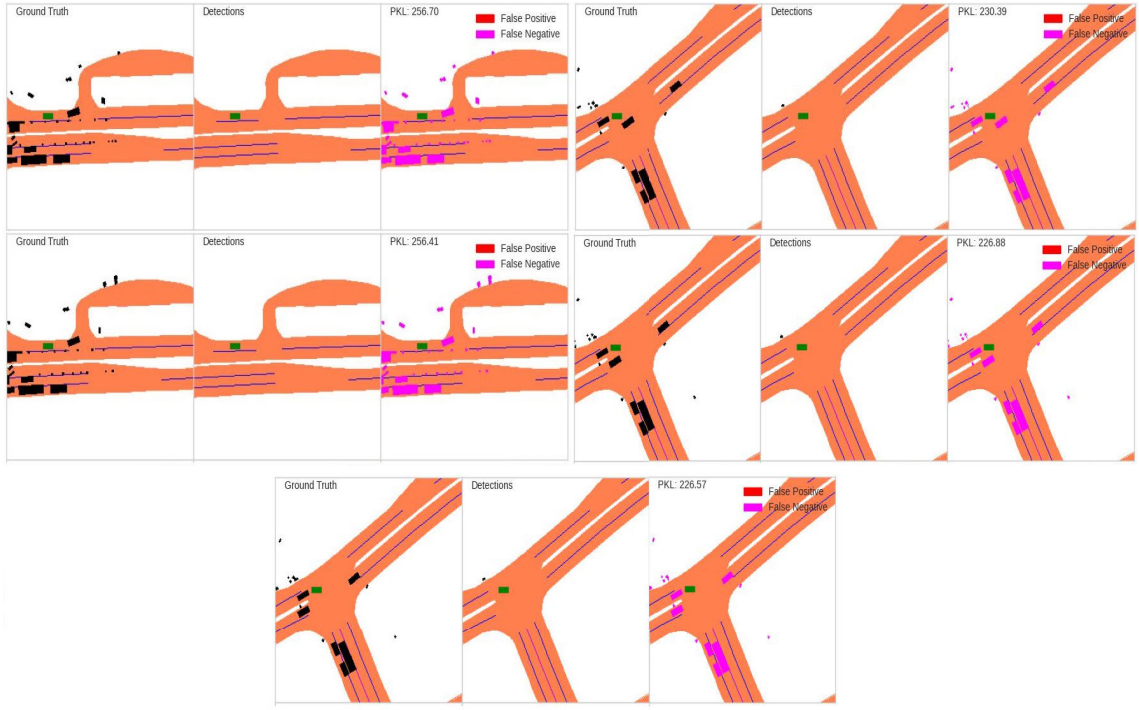


FIGURE 16. Predictions with PKL not equal to zero.

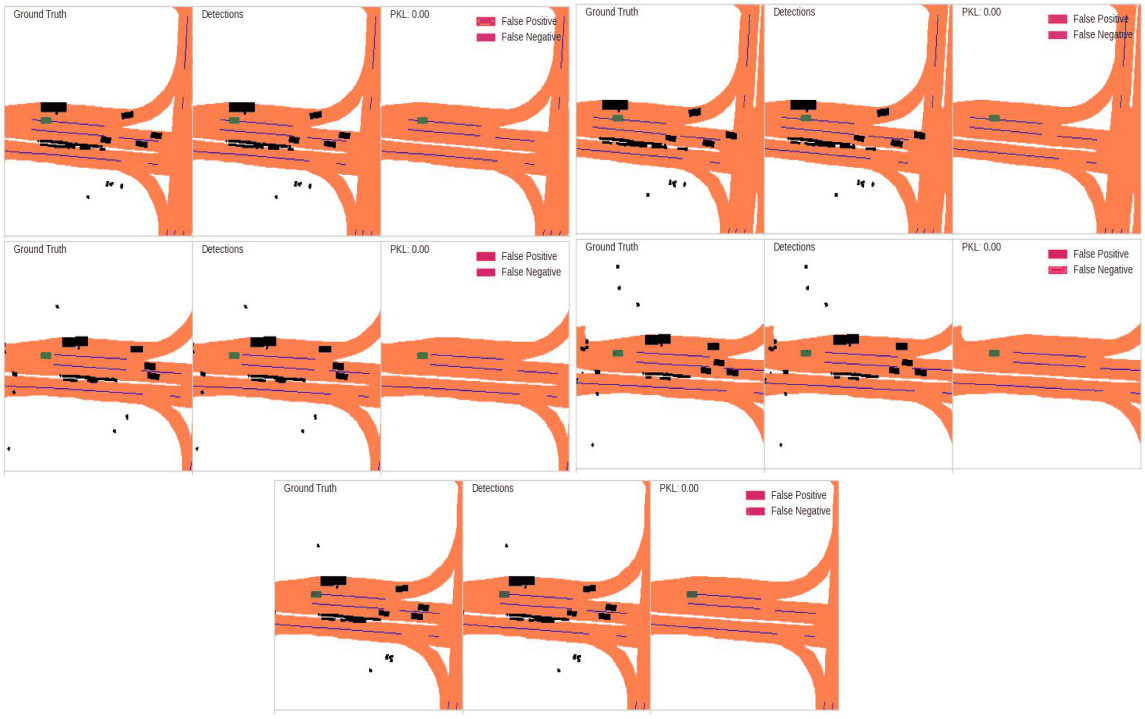
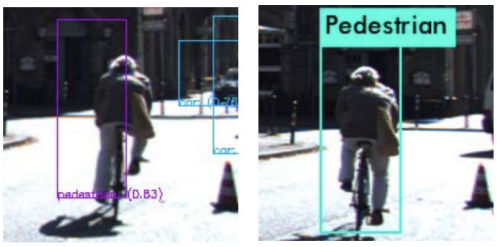


FIGURE 17. Predictions with PKL equal to zero.

straight ahead of the camera, making the cyclist appear to be a pedestrian. This may lead to a violation of this specification. So, the authors of the paper [59] defined another specification that monitors whether the algorithms can detect any cyclists when there are actually at least one. This specification indicates that if an object is detected, it is either a cyclist with



a) SqueezeDet b) YOLOv3

FIGURE 18. A cyclist is misclassified twice with both SqueezeDet and YOLOv3.

a probability greater than 0.6 or there is another object, less than 40 pixels away, that is a pedestrian with a probability greater than 0.6.

The ability of TQTL to compare object characteristics across frames is demonstrated by these specifications. As seen in Fig. 19, the first specification is broken since the cyclist that YOLOv3 mistakenly classified as a pedestrian with a fair amount of confidence. As seen in Fig. 19, we can see that the requirement set forth by the second specification is being violated since the likelihood of the cyclist being correctly identified as a cyclist or a pedestrian is dropping below 60%. This is the reason for the negative robustness observed when measured against the second specification. As the cyclist is incorrectly classified as a pedestrian with high confidence in the stream in Fig. 20, SqueezeDet violates specification one. This demonstrates that the algorithm incorrectly labels the cyclist as a pedestrian in images like YOLOv3, where the cyclist is moving nearly parallel to the direction of the camera. Even if we used a second specification to keep an eye on this misclassification, the algorithm continues to break the property. This is caused by ''phantom'' objects that SqueezeDet had a high likelihood of detecting but then unexpectedly failed to do so. Fig. 20 is an illustration of this. With these results, we were able to locate intriguing examples of bad quality perception algorithm outputs localized to a set of frames using TQTL. When a perception algorithm is being debugged, such information can be quite helpful, especially if it is being used in a situation where safety is crucial.

I. SPATIO-TEMPORAL QUALITY LOGIC (STQL)

The PerceMon framework monitors and broadcasts all the data from the simulator, including information from the autonomous vehicle's cameras, using the ROS wrapper for CARLA [61] as shown in Fig. 21. Perception modules, such as the YOLO object detector [110] and the DeepSORT object tracker [111], use the image data to broadcast processed data. These perception modules publish information that can be used by other perception modules, controllers, online PerceMon monitors, and other controllers to follow objects that are recognized and possibly avoid collisions. Fig. 21 depicts an overview of the architecture. The bounding boxes of images are detected with the YOLO object detector, and DeepSORT assigns an ID to each of the sets of detections that the object detector makes. Then, using Kalman filters and cosine association measures, it tries to follow each item that has been marked over multiple frames. PerceMon successfully detects false negatives and false positives in object detectors by creating two specifications in STQL. Those are:

- Consistent detection: If an object is far from the margins in the current frame and has a high confidence value, it must have existed in the preceding frame with a similar high confidence value.
- Smooth object trajectories: Every object in the current frame must have a bounding box that overlaps with the equivalent bounding box in the previous frame by at least 30%.

With the help of these two specifications, PerceMon keeps an eye on the aforementioned attributes for the situations shown in Fig. 22, as well as on how long it takes to compute the satisfaction values of the aforementioned properties. The object detector finds more objects as more passive or non-adversarial vehicles are included in each scenario. PerceMon can therefore empirically evaluate how long it takes to compute the satisfaction value in the monitor because the runtime for the STQL monitor grows exponentially with the number of item IDs.

J. PANOPTIC SEGMENTATION

In this article, we consider some of the models used for panoptic segmentation, and the respective results are tabulated in Table 9. In this table, the superscripts '*Th*' and '*St*' stand for 'thing' and 'stuff'. A unified panoptic segmentation network (UPSNet) was suggested for the panoptic segmentation task in the paper [88]. It contains a Mask R-CNN and deformable convolution-based instance segmentation and semantic segmentation, respectively, on top of a single backbone residual network. Much more significantly, it supports a parameter-free panoptic head that resolves panoptic segmentation through pixel-wise classification. It first makes use of the logits from the prior two heads before creatively expanding the representation to allow prediction of an additional unknown class that aids in more effectively resolving conflicts between semantic and instance segmentation. It also addresses the difficulty brought on by the variable number of instances and allows end-to-end back propagation to the bottom modules. The results shown in Table 9 were obtained using the model pretrained on the COCO dataset (examples UPSNet-COCO and UPSNet-101-M-COCO). In the DeeperLab [92] model, a single pass of a fully convolutional network is used to produce the per-pixel semantic and instance predictions. A fusion algorithm then fuses these predictions into the final image parsing (panoptic segmentation) output. In this network, Xception-71 was used as the backbone network. In VPSNet [85], a video panoptic segmentation was performed that predicates bounding boxes, pixel-wise classification, and the respective IDs. Experimental results were obtained for two datasets with the proposed performance

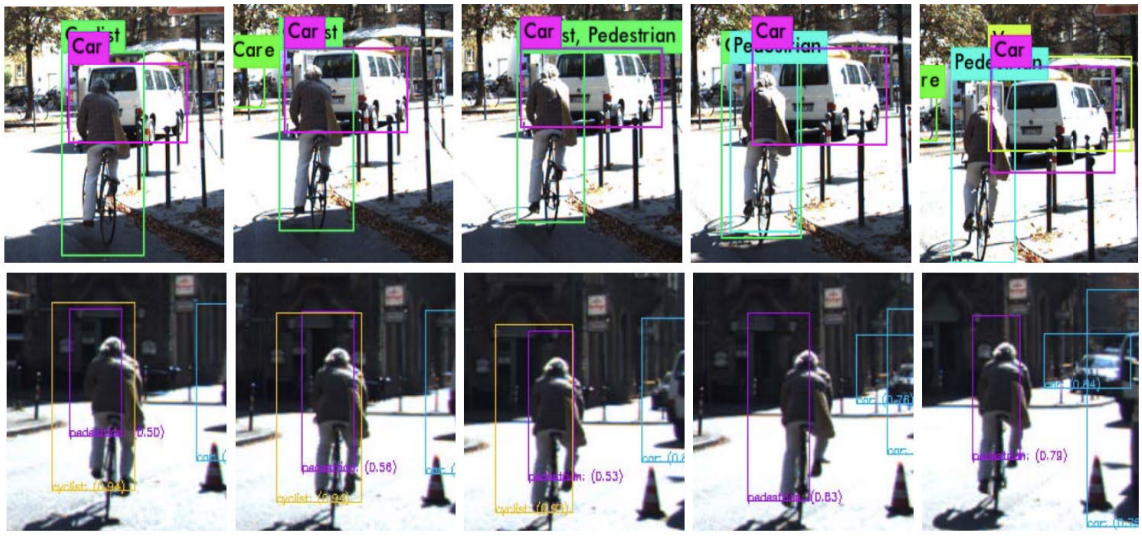


FIGURE 19. a) With high confidence (greater than 75%), YOLOv3 incorrectly labels the cyclist as a pedestrian; b) The possibility that YOLO will detect a cyclist varies from 0 to 75%.

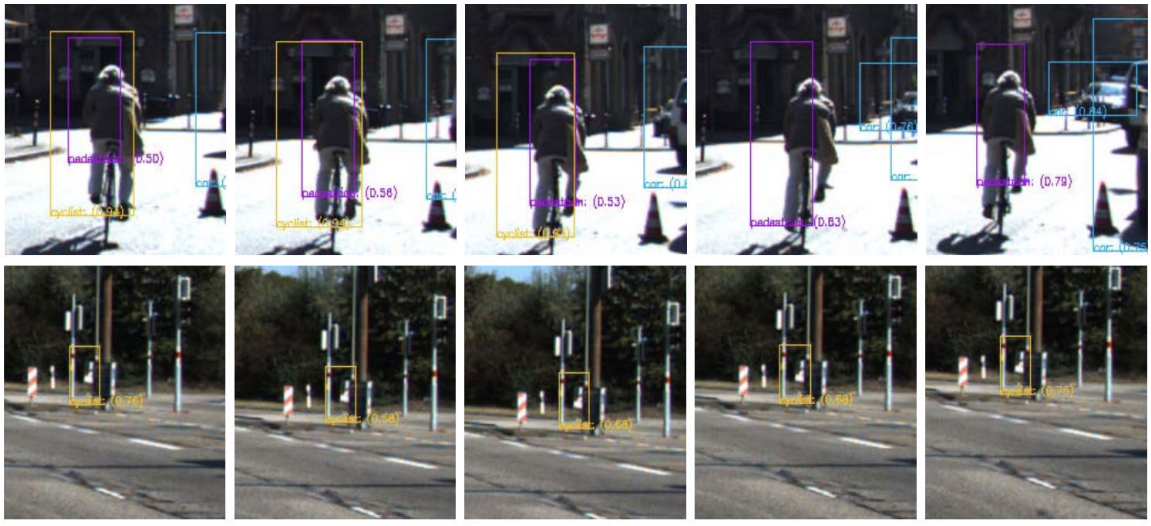


FIGURE 20. a) With high confidence (greater than 75%), SqueezeDet incorrectly labels the cyclist as a pedestrian; b) Occasionally, SqueezeDet detects an erroneous cyclist with a probability ranging from 55% to 75%.

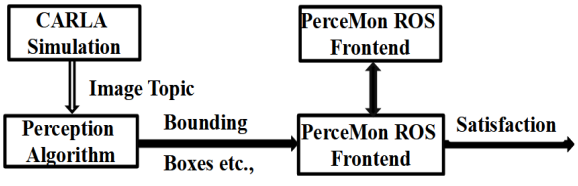


FIGURE 21. PerceMon architecture.

metrics. In Table 9, additional terms COCO and VP indicate that models are pretrained on COCO and VIPER datasets [116], respectively. EfficientPS [112] consists of a shared backbone that effectively encodes and integrates multiscale information with rich semantic content. It has a new variation of Mask R-CNN as the instance head and a new semantic head that coherently collects fine and contextual features. Additionally, it has a brand-new panoptic fusion module that combines the output logits from both heads of architecture to produce the output for panoptic segmentation.

PSPNet [76] contains a Mask R-CNN and FPN based instance and semantic segmentation, respectively. Surprisingly, this basic framework not only continues to work well for instance segmentation but also produces a fast, efficient way for semantic segmentation. RTPS [113] used dense detection and a global self-attention mechanism. This model presents a unique parameter-free mask-building technique that effectively makes use of data from the object detection and semantic segmentation subtasks to significantly reduce computational complexity. Because of the network's straightforward data flow and lack of feature map

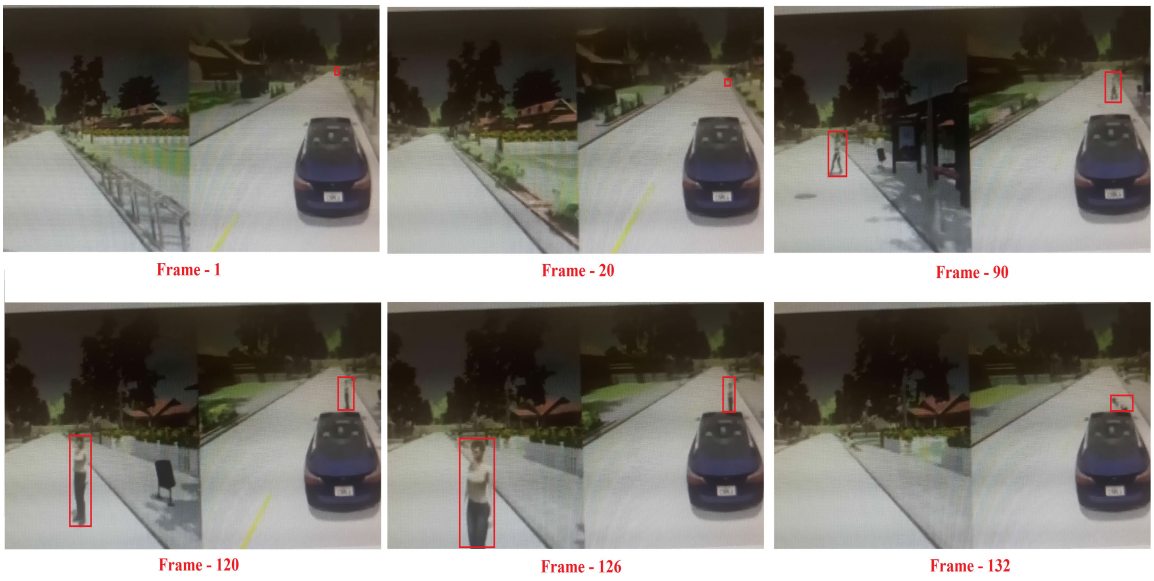


FIGURE 22. PerceMon results with STQL [61].

TABLE 9. Results of different panoptic segmentation models within the Cityscapes dataset.

Models	PQ	SQ	RQ	PQ^{Th}	PQ^{St}	$mIOU$	AP	PC
UPSNet [88]	59.3	79.7	73	54.6	62.7	75.2	33.3	—-
UPSNet-COCO [88]	60.5	80.9	73.5	57	63	77.8	37.8	—-
UPSNet-101-M-COCO [88]	61.8	81.3	74.8	57.6	64.8	79.2	39	—-
DeeperLab-Xception-71 [92]	56.53	—-	—-	—-	—-	—-	—-	75.63
VPSNet-Base+COCO [85]	60.6	—-	—-	57	63.2		—-	—-
VPSNet-Fuse+VP [85]	62.2	—-		58	65.3			—-
EfficientPS [112]	63.9	81.5	77.1	60.7	66.2	79.3	38.3	—-
PSPNet + M-RCNN [76]	61.2	80.9	74.4	54	66.4	36.4	80.9	—-
RTPS + ResNet-50-FPN [113]	58.8	—-	—-	52.1	63.7	77	29.8	—-
PanopticDepth + R-50 [114]	64.1	58.8	68.1	62	55	67.1	—-	—-
Panoptic-DeepLab [115]	42.7	78.1	52.5	35.9	51.6	56.8	17.2	—-

resampling, significant hardware acceleration is possible. The PanopticDepth [114] model was designed with the dynamic convolution technique, which helps to predict depth and segmentation masks for each instance instead of predicting depth for all pixels at a time. The Panoptic-DeepLab [115] contains a dual Atrous Spatial Pyramid Pooling (ASPP) and dual-decoder for instance segmentation and semantic segmentation, respectively. In order to extract a denser feature map, it uses atrous convolution in the final block of a network backbone. The context module uses the ASPP together with a lightweight decoder module that only uses one convolution at a time during each upsampling stage.

VIII. CONCLUSION

In autonomous driving and advanced driver-assistance systems, perception algorithms play a significant role in observing the surrounding environment for safe, secure, and collision-free motion. The performance of these algorithms depends on several factors, and selecting the most accurate and robust one is a crucial task. Thus, after training, performance has to be defined and evaluated with metrics based on ''unseen'' test data. This is achieved by resorting to testing methods that compare their output against the ground truth (annotated data) included within the dataset, and provide detailed test reports including statistics, correlations, outliers, etc.

This paper presents an overview of the main four perception performance assessment approaches: point cloud quality analysis, object detection, object tracking, and panoptic segmentation. Different metrics and their advantages and disadvantages over different models are also discussed, with particular emphasis on state-of-the-art metrics used for performance measures of object detection, object tracking, and panoptic segmentation algorithms. Actually, object tracking is intimately related to object detection, as tracking implies detecting the same object through frames and estimating or predicting its positions and other details of a moving object.

The following main conclusions can be drawn from the conducted experiments:

- LiDAR point cloud: The originality of the environment captured with LiDARs depends on many factors, such as the functional characteristics of the sensors used, environmental weather and lighting conditions, speed of the ego vehicle, etc. To measure the LiDAR device's

accuracy with respect to a reference LiDAR, distance accuracy is the most commonly used metric. In this review, we evaluated four different distance metrics, and among them, we observed that the Earth Mover's Distance metric gives better dissimilarity between two point clouds or distributions generated with two different LiDARs.

- Object detection: For autonomous vehicles, the most important perception algorithm is object detection. So in this review, we have given importance to metrics that measure the performance of object detection algorithms. The effectiveness of object detection models depends on many factors, such as the speed and size of the object, the size of the dataset, an imbalance in the class of objects, etc. In the literature, several metrics exist to measure the performance of object detection models, but these metrics have their own advantages and disadvantages. The most commonly used metric for object detection is mean average precision, but it ignores the object's position, velocity, speed, and orientation. So, the nuScenes detection score was introduced, which covers all four of these in calculating the model's performance. To understand this effect, we consider shape signature networks and PointPillers (with backbone networks such as feature pyramid network and SECOND feature pyramid network) as an object detection model. With these models, we explained the effect of the nuscene detection score in measuring the model's performance over the mean average precision. Also, it was observed that mAP and NDS also fail to identify the false positives and false negatives of the object detection model. So PKL was introduced and tested on the MEGVII point cloud 3D object detection model. With this test, it was observed that PKL is capable of distinguishing a parked vehicle from a vehicle in front of the autonomous vehicles by giving different confident scores, while mAP and NDS fail to distinguish both cases. We also observed that a higher PKL value shows the worst performance of the model (the ideal value would be zero). In addition to these metrics, TQTL and STQL were introduced as object detection metrics. The TQTL metric considers time, and STQL considers both time and space in evaluating the model's performance. To know the impact of TQTL on queezeDet and YOLOv3 models, we performed experiments on pre-trained models of both on the KITTI dataset. It could be observed that both models fail to detect pedestrians and cyclists when they are exactly opposite to autonomous vehicles. But TQTL identifies these false positives by introducing two specifications in terms of time frames for video. Similarly, STQL was used to track or monitor the performance of the YOLO object detector, followed by the DeepSORT object tracker. STQL successfully detects false negatives and false positives in object detectors by creating two specifications in terms of the spatial and time frames of the video.
- Panoptic segmentation: It is a cascaded combination of semantic and instance segmentation; thus, metrics used for both are useful for panoptic segmentation. The most commonly used metric for semantic segmentation is the Dice coefficient. In the literature, two metrics for panoptic segmentation could be found, parsing covering and panoptic quality metrics, which are a combination of segmentation quality and recognition quality. A table of models with these metrics is presented.

Different methods exist that allow us to evaluate the performance of LiDAR data perception algorithms. The diversity and specificity of driving conditions and vehicles' surrounding situations, requires the rigorous application of various methods to fully evaluate the algorithms' capabilities and ensure the highest levels of dependability and safety of autonomous driving and advanced driver-assistance systems. Other methods, not reported here, exist or are being developed to tackle these requirements. These include, e. g., testing in dynamic scenarios, measure of the signal to noise ratio of both distance and beam intensity, and under moisture, mechanical and other environmental influences. On the other hand, a higher diversity of datasets is needed so that the most realistic evaluation conditions are available as input.

REFERENCES

[1] S. Singh, "Critical reasons for crashes investigated in the national motor vehicle crash causation survey. (Traffic safety facts crash. Stats)," Nat. Highway Traffic Saf. Admin., Washington, DC, USA, Tech. Rep. DOT HS 812 506, Mar. 2018.

[2] T. Winkle, "Safety benefits of automated vehicles: Extended findings from accident research for development, validation and testing," in *Autonomous Driving: Technical, Legal and Social Aspects*, M. Maurer, J. C. Gerdes, B. Lenz, and H. Winner, Eds. Berlin, Germany: Springer, 2016, pp. 335–364, doi: 10.1007/978-3-662-48847-8_17.

[3] C. Katrakazas, "Developing an advanced collision risk model for autonomous vehicles," Ph.D. dissertation, Transp. Study Group, Loughborough Univ., Loughborough, U.K., 2017.

[4] R. K. Jurgen. (2013). *Autonomous Vehicles for Safer Driving*. [Online]. Available: https://api.semanticscholar.org/CorpusID:106544090

[5] M. Wood, C. Knobel, N. Garbacik, P. Robbel, D. Boymanns, D. Smerza, M. Maass, M. Löhning, L. Dalong, and D. Radboud, "Safety first for automated driving. white paper of different car manufacutres and suppliers," Aptiv Services US, LLC, AUDI AG, Bayrische Motoren Werke AG, Beijing Baidu Netcom Sci. Technol. Co., Ltd, ContinentalTeves AG Co oHG, Daimler AG, FCA US LLC, HERE Global B.V., Infineon Technol. AG, Intel, Volkswagen AG, Tech. Rep., 2019.

[6] Gran View Research, "LiDAR market size, share & trends analysis report by product type (airborne, terrestrial, mobile & UAV), by component, by application, by region, and segment forecasts, 2022–2030," Gran View Res., San Francisco, CA, USA, Tech. Rep. 978-1-68038-344-7, 2022. [Online]. Available: https://www.grandviewresearch.com/industry-analysis/lidar-light-detection-and-ranging-market

[7] *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, SAE Int., Warrendale, PA, USA, vol. 4970, no. 724, pp. 1–5, 2018.

[8] M. Zohaib, M. Ahsan, M. Khan, and J. Iqbal, "A featureless approach for object detection and tracking in dynamic environments," *PLoS ONE*, vol. 18, no. 1, Jan. 2023, Art. no. e0280476.

[9] L. Wang, Z. Song, X. Zhang, C. Wang, G. Zhang, L. Zhu, J. Li, and H. Liu, "SAT-GCN: Self-attention graph convolutional network-based 3D object detection for autonomous driving," *Knowl.-Based Syst.*, vol. 259, Jan. 2023, Art. no. 110080.

[10] H. Liu, Z. Xu, D. Wang, B. Zhang, G. Wang, B. Dong, X. Wen, and X. Xu, "PAI3D: Painting adaptive instance-prior for 3D object detection," in *Computer Vision–ECCV 2022*. Tel Aviv, Israel: Springer, Oct. 2023, pp. 459–475.

[11] M. Dyberg, A. Hedvall, J. Hultenheim, O. P. Ruiz, S. Rahmanian, G. Rangaraju, E. Troillet, and E. Z. Raheem, "Towards autonomous driving on KTH campus: AD-EYE," KTH, School Ind. Eng. Manag. (ITM), 2023, p. 71.

[12] A. P. Sligar, "Machine learning-based radar perception for autonomous vehicles using full physics simulation," *IEEE Access*, vol. 8, pp. 51470–51476, 2020.

[13] J. You and Y.-K. Kim, "Up-sampling method for low-resolution LiDAR point cloud to enhance 3D object detection in an autonomous driving environment," *Sensors*, vol. 23, no. 1, p. 322, Dec. 2022.

[14] S. Capy, G. Venture, and P. Raksincharoensak, "Pedestrians and cyclists' intention estimation for the purpose of autonomous driving: A systematic review," *Int. J. Automot. Eng.*, vol. 14, no. 1, pp. 10–19, 2023.

[15] M. N. Sharath, N. R. Velaga, and M. A. Quddus, "A dynamic two-dimensional (D2D) weight-based map-matching algorithm," *Transp. Res. C, Emerg. Technol.*, vol. 98, pp. 409–432, Jan. 2019.

[16] S. B. Sarkar and B. C. Mohan, "Review on autonomous vehicle challenges," in *Proc. 1st Int. Conf. Artif. Intell. Cogn. Comput. (AICC*. Cham, Switzerland: Springer, 2019, pp. 593–603.

[17] R. Hussain and S. Zeadally, "Autonomous cars: Research results, issues, and future challenges," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1275–1313, 2nd Quart., 2019.

[18] W. Shi, M. B. Alawieh, X. Li, and H. Yu, "Algorithm and hardware implementation for visual perception system in autonomous vehicle: A survey," *Integration*, vol. 59, pp. 148–156, Sep. 2017.

[19] Y. Ma, Z. Wang, H. Yang, and L. Yang, "Artificial intelligence applications in the development of autonomous vehicles: A survey," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 2, pp. 315–329, Mar. 2020.

[20] P. Dong, J. Zhao, X. Liu, J. Wu, X. Xu, Y. Liu, S. Wang, and W. Guo, "Practical application of energy management strategy for hybrid electric vehicles based on intelligent and connected technologies: Development stages, challenges, and future trends," *Renew. Sustain. Energy Rev.*, vol. 170, Dec. 2022, Art. no. 112947.

[21] A. K. Tyagi and S. U. Aswathy, "Autonomous intelligent vehicles (AIV): Research statements, open issues, challenges and road for future," *Int. J. Intell. Netw.*, vol. 2, pp. 83–102, 2021.

[22] M. Girdhar, J. Hong, and J. Moore, "Cybersecurity of autonomous vehicles: A systematic literature review of adversarial attacks and defense models," *IEEE Open J. Veh. Technol.*, vol. 4, pp. 417–437, 2023.

[23] M. Alawadhi, J. Almazrouie, M. Kamil, and K. A. Khalil, "A systematic literature review of the factors influencing the adoption of autonomous driving," *Int. J. Syst. Assurance Eng. Manage.*, vol. 11, no. 6, pp. 1065–1082, Dec. 2020.

[24] X. Yu and M. Marinov, "A study on recent developments and issues with obstacle detection systems for automated vehicles," *Sustainability*, vol. 12, no. 8, p. 3281, Apr. 2020.

[25] E. Khatab, A. Onsy, M. Varley, and A. Abouelfarag, "Vulnerable objects detection for autonomous driving: A review," *Integration*, vol. 78, pp. 36–48, May 2021.

[26] E. Marti, M. A. de Miguel, F. Garcia, and J. Perez, "A review of sensor technologies for perception in automated driving," *IEEE Intell. Transp. Syst. Mag.*, vol. 11, no. 4, pp. 94–108, Jun. 2019.

[27] D. J. Yeong, G. Velasco-Hernandez, J. Barry, and J. Walsh, "Sensor and sensor fusion technology in autonomous vehicles: A review," *Sensors*, vol. 21, no. 6, p. 2140, Mar. 2021.

[28] J. Van Brummelen, M. O'Brien, D. Gruyer, and H. Najjaran, "Autonomous vehicle perception: The technology of today and tomorrow," *Transp. Res. C, Emerg. Technol.*, vol. 89, pp. 384–406, Apr. 2018.

[29] Y. Li and J. Ibanez-Guzman, "LiDAR for autonomous driving: The principles, challenges, and trends for automotive LiDAR and perception systems," *IEEE Signal Process. Mag.*, vol. 37, no. 4, pp. 50–61, Jul. 2020.

[30] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE Access*, vol. 8, pp. 58443–58469, 2020.

[31] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1135–1145, Apr. 2016.

[32] D. Feng, A. Harakeh, S. L. Waslander, and K. Dietmayer, "A review and comparative study on probabilistic object detection in autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 9961–9980, Aug. 2022.

[33] D. Feng, C. Haase-Schütz, L. Rosenbaum, H. Hertlein, C. Gläser, F. Timm, W. Wiesbeck, and K. Dietmayer, "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1341–1360, Mar. 2021.

[34] A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, and J. Garcia-Rodriguez, "A survey on deep learning techniques for image and video semantic segmentation," *Appl. Soft Comput.*, vol. 70, pp. 41–65, Sep. 2018.

[35] H. Fujiyoshi, T. Hirakawa, and T. Yamashita, "Deep learning-based image recognition for autonomous driving," *IATSS Res.*, vol. 43, no. 4, pp. 244–252, Dec. 2019.

[36] A. A. Selçuk, "A guide for systematic reviews: PRISMA," *Turkish Arch. Otorhinolaryngology*, vol. 57, no. 1, pp. 57–58, 2019.

[37] J. Ribera, D. Guera, Y. Chen, and E. J. Delp, "Locating objects without bounding boxes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 6479–6489.

[38] M.-P. Dubuisson and A. K. Jain, "A modified Hausdorff distance for object matching," in *Proc. 12th Int. Conf. Pattern Recognit.*, vol. 1, 1994, pp. 566–568.

[39] A. Savkin, Y. Wang, S. Wirkert, N. Navab, and F. Tombari, "LiDAR upsampling with sliced Wasserstein distance," *IEEE Robot. Autom. Lett.*, vol. 8, no. 1, pp. 392–399, Jan. 2023.

[40] D. Urbach, Y. Ben-Shabat, and M. Lindenbaum, "DPDIST: Comparing point clouds using deep point cloud distance," in *Computer Vision–ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham, Switzerland: Springer, 2020, pp. 545–560.

[41] *LiDAR Accuracy Assessment*. Accessed: Jan. 2023. [Online]. Available: https://up42.com/blog/author/rose-njambi

[42] H. Zhao, Q. Yang, W. Zhu, and Y. Xu, "A quality metric for 3D LiDAR point cloud based on vision tasks," in *Proc. IEEE Int. Symp. Broadband Multimedia Syst. Broadcast. (BMSB)*, Oct. 2020, pp. 1–5.

[43] L. T. Triess, C. B. Rist, D. Peter, and J. M. Zöllner, "A realism metric for generated LiDAR point clouds," *Int. J. Comput. Vis.*, vol. 130, no. 12, pp. 2962–2979, Dec. 2022.

[44] Y. Liu, Q. Yang, Y. Xu, and L. Yang, "Point cloud quality assessment: Dataset construction and learning-based no-reference metric," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 19, no. 2s, pp. 1–26, Jun. 2023.

[45] S. Kodors, "Point distribution as true quality of LiDAR point cloud," *Baltic J. Modern Comput.*, vol. 5, no. 4, pp. 362–378, Dec. 2017.

[46] E. Alexiou and T. Ebrahimi, "Towards a point cloud structural similarity metric," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2020, pp. 1–6.

[47] S. Pang, D. Morris, and H. Radha, "CLOCs: Camera-LiDAR object candidates fusion for 3D object detection," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 10386–10393.

[48] H. Zhang, A. Rogozan, and A. Bensrhair, "An enhanced N-point interpolation method to eliminate average precision distortion," *Pattern Recognit. Lett.*, vol. 158, pp. 111–116, Jun. 2022.

[49] R. Padilla, S. L. Netto, and E. A. B. da Silva, "A survey on performance metrics for object-detection algorithms," in *Proc. Int. Conf. Syst., Signals Image Process. (IWSSIP)*, Jul. 2020, pp. 237–242.

[50] J. Hosang, R. Benenson, P. Dollár, and B. Schiele, "What makes for effective detection proposals?" *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 4, pp. 814–830, Apr. 2016.

[51] R. Padilla, W. L. Passos, T. L. B. Dias, S. L. Netto, and E. A. B. da Silva, "A comparative analysis of object detection metrics with a companion open-source toolkit," *Electronics*, vol. 10, no. 3, p. 279, Jan. 2021.

[52] H. Zhang, M. Abualsaud, N. Ghelani, M. D. Smucker, G. V. Cormack, and M. R. Grossman, "Effective user interaction for high-recall retrieval: Less is more," in *Proc. 27th ACM Int. Conf. Inf. Knowl. Manage.*, 2018, pp. 187–196.

[53] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

[54] X. Huang, P. Wang, X. Cheng, D. Zhou, Q. Geng, and R. Yang, "The ApolloScape open dataset for autonomous driving and its application," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2702–2719, Oct. 2020.

[55] J. Huang, G. Huang, Z. Zhu, Y. Ye, and D. Du, "BEVDet: High-performance multi-camera 3D object detection in bird-eye-view," 2021, *arXiv:2112.11790*.

[56] J. Philion, A. Kar, and S. Fidler, "Learning to evaluate perception models using planner-centric metrics," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jul. 2020, pp. 14055–14064.

[57] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuScenes: A multimodal dataset for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 11621–11631.

[58] A. Balakrishnan, A. G. Puranic, X. Qin, A. Dokhanchi, J. V. Deshmukh, H. B. Amor, and G. Fainekos, "Specifying and evaluating quality metrics for vision-based perception systems," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, 2019, pp. 1433–1438.

[59] A. Dokhanchi, H. B. Amor, J. V. Deshmukh, and G. Fainekos, "Evaluating perception systems for autonomous vehicles using quality temporal logic," in *Runtime Verification*. Limassol, Cyprus: Springer, Nov. 2018, pp. 409–416.

[60] B. Wu, A. Wan, F. Iandola, P. H. Jin, and K. Keutzer, "SqueezeDet: Unified, small, low power fully convolutional neural networks for real-time object detection for autonomous driving," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 129–137.

[61] A. Balakrishnan, J. Deshmukh, B. Hoxha, T. Yamaguchi, and G. Fainekos, "PerceMon: Online monitoring for perception systems," in *Runtime Verification*. Cham, Switzerland: Springer, Oct. 2021, pp. 297–308.

[62] G. E. Fainekos and G. J. Pappas, "Robustness of temporal logic specifications for continuous-time signals," *Theor. Comput. Sci.*, vol. 410, no. 42, pp. 4262–4291, Sep. 2009.

[63] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: An open-source robot operating system," in *Proc. ICRA Workshop Open Source Softw.*, Kobe, Japan, vol. 3, nos. 3–2, 2009, p. 5.

[64] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. Conf. Robot Learn.*, 2017, pp. 1–16.

[65] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[66] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Computer Vision–ECCV 2014*. Zurich, Switzerland: Springer, Sep. 2014, pp. 740–755.

[67] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[68] T. Akiba, T. Kerola, Y. Niitani, T. Ogawa, S. Sano, and S. Suzuki, "PFDet: 2nd place solution to open images challenge 2018 object detection track," 2018, *arXiv:1809.00778*.

[69] S. Mandal, S. Biswas, V. E. Balas, R. N. Shaw, and A. Ghosh, "Lyft 3D object detection for autonomous vehicles," in *Artificial Intelligence for Future Generation Robotics*. Amsterdam, The Netherlands: Elsevier, 2021, pp. 119–136.

[70] K. Wang, B. Fang, J. Qian, S. Yang, X. Zhou, and J. Zhou, "Perspective transformation data augmentation for object detection," *IEEE Access*, vol. 8, pp. 4935–4943, 2020.

[71] K. Li, Z. Huang, Y.-C. Cheng, and C.-H. Lee, "A maximal figure-of-merit learning approach to maximizing mean average precision with deep neural network based classifiers," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 4503–4507.

[72] Y. Gao, Q. Wang, X. Tang, H. Wang, F. Ding, J. Li, and Y. Hu, "Decoupled IoU regression for object detection," in *Proc. 29th ACM Int. Conf. Multimedia*, 2021, pp. 5628–5636.

[73] A. Milan, L. Leal-Taixe, I. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," 2016, *arXiv:1603.00831*.

[74] J. Luiten, A. Osep, P. Dendorfer, P. Torr, A. Geiger, L. Leal-Taixé, and B. Leibe, "HOTA: A higher order metric for evaluating multi-object tracking," *Int. J. Comput. Vis.*, vol. 129, pp. 1–31, Oct. 2020.

[75] M. B. Wright, "Speeding up the Hungarian algorithm," *Comput. Oper. Res.*, vol. 17, no. 1, pp. 95–96, Jan. 1990.

[76] A. Kirillov, R. Girshick, K. He, and P. Dollár, "Panoptic feature pyramid networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 6399–6408.

[77] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2015, pp. 3431–3440.

[78] A. Mukherjee, S. Chakraborty, and S. K. Saha, "Detection of loop closure in SLAM: A DeconvNet based approach," *Appl. Soft Comput.*, vol. 80, pp. 650–656, Jul. 2019.

[79] N. Siddique, S. Paheding, C. P. Elkin, and V. Devabhaktuni, "U-net and its variants for medical image segmentation: A review of theory and applications," *IEEE Access*, vol. 9, pp. 82031–82057, 2021.

[80] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder–decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[81] K. Wang, J. H. Liew, Y. Zou, D. Zhou, and J. Feng, "PANet: Few-shot image semantic segmentation with prototype alignment," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Aug. 2019, pp. 9197–9206.

[82] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[83] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2961–2969.

[84] D. Bolya, C. Zhou, F. Xiao, and Y. J. Lee, "YOLACT: Real-time instance segmentation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Sep. 2019, pp. 9157–9166.

[85] D. Kim, S. Woo, J.-Y. Lee, and I. S. Kweon, "Video panoptic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 9859–9868.

[86] C.-Y. Chang, S.-E. Chang, P.-Y. Hsiao, and L.-C. Fu, "EPSNet: Efficient panoptic segmentation network with cross-layer attention fusion," in *Proc. Asian Conf. Comput. Vis.*, 2020, pp. 689–705.

[87] A. Xiao, X. Yang, S. Lu, D. Guan, and J. Huang, "FPS-net: A convolutional fusion network for large-scale LiDAR point cloud segmentation," *ISPRS J. Photogramm. Remote Sens.*, vol. 176, pp. 237–249, Jun. 2021.

[88] Y. Xiong, R. Liao, H. Zhao, R. Hu, M. Bai, E. Yumer, and R. Urtasun, "UPSNet: A unified panoptic segmentation network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 8818–8826.

[89] S. Jha, L. H. Son, R. Kumar, I. Priyadarshini, F. Smarandache, and H. V. Long, "Neutrosophic image segmentation with dice coefficients," *Measurement*, vol. 134, pp. 762–772, Feb. 2019.

[90] K. Arase, Y. Mukuta, and T. Harada, "Rethinking task and metrics of instance segmentation on 3D point clouds," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 4105–4113.

[91] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.

[92] T.-J. Yang, M. D. Collins, Y. Zhu, J.-J. Hwang, T. Liu, X. Zhang, V. Sze, G. Papandreou, and L.-C. Chen, "DeeperLab: Single-shot image parser," 2019, *arXiv:1902.05093*.

[93] Z. Zhang, S. Fidler, and R. Urtasun, "Instance-level segmentation for autonomous driving with deep densely connected MRFs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 669–677.

[94] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, J. Gall, and C. Stachniss, "Towards 3D LiDAR-based semantic scene understanding of 3D point cloud sequences: The SemanticKITTI dataset," *Int. J. Robot. Res.*, vol. 40, nos. 8–9, pp. 959–967, Aug. 2021.

[95] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3213–3223.

[96] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ADE20K dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 633–641.

[97] G. Neuhold, T. Ollmann, S. Rota Bulo, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 4990–4999.

[98] M. Simony, S. Milzy, K. Amendey, and H.-M. Gross, "Complex-YOLO: An euler-region-proposal for real-time 3D object detection on point clouds," in *Proc. Eur. Conf. Comput. Vis. (ECCV) Workshops*, 2018, pp. 1–14.

[99] T. Yang, D. Li, Y. Bai, F. Zhang, S. Li, M. Wang, Z. Zhang, and J. Li, "Multiple-object-tracking algorithm based on dense trajectory voting in aerial videos," *Remote Sens.*, vol. 11, no. 19, p. 2278, Sep. 2019.

[100] U. Nepal and H. Eslamiat, "Comparing YOLOv3, YOLOv4 and YOLOv5 for autonomous landing spot detection in faulty UAVs," *Sensors*, vol. 22, no. 2, p. 464, Jan. 2022.
[101] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum PointNets for 3D object detection from RGB-D data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Nov. 2018, pp. 918–927.
[102] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4490–4499.
[103] L. Wang, T. Chen, C. Anklam, and B. Goldluecke, "High dimensional frustum PointNet for 3D object detection from camera, LiDAR, and radar," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020, pp. 1621–1628.
[104] Y. Yan, Y. Mao, and B. Li, "SECOND: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, Oct. 2018.
[105] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "DSSD : Deconvolutional single shot detector," 2017, *arXiv:1701.06659*.
[106] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with Siamese region proposal network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8971–8980.
[107] X. Zhu, Y. Ma, T. Wang, Y. Xu, J. Shi, and D. Lin, "SSN: Shape signature networks for multi-class object detection from point clouds," in *Computer Vision–ECCV 2020*. Glasgow, U.K.: Springer, Aug. 2020, pp. 581–597.
[108] B. Zhu, Z. Jiang, X. Zhou, Z. Li, and G. Yu, "Class-balanced grouping and sampling for point cloud 3D object detection," 2019, *arXiv:1908.09492*.
[109] B. Graham, "Sparse 3D convolutional neural networks," 2015, *arXiv:1505.02890*.
[110] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, May 2016, pp. 779–788.
[111] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3645–3649.
[112] R. Mohan and A. Valada, "EfficientPS: Efficient panoptic segmentation," *Int. J. Comput. Vis.*, vol. 129, no. 5, pp. 1551–1579, May 2021.
[113] R. Hou, J. Li, A. Bhargava, A. Raventos, V. Guizilini, C. Fang, J. Lynch, and A. Gaidon, "Real-time panoptic segmentation from dense detections," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jul. 2020, pp. 8523–8532.
[114] N. Gao, F. He, J. Jia, Y. Shan, H. Zhang, X. Zhao, and K. Huang, "PanopticDepth: A unified framework for depth-aware panoptic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jul. 2022, pp. 1632–1642.
[115] B. Cheng, M. D. Collins, Y. Zhu, T. Liu, T. S. Huang, H. Adam, and L.-C. Chen, "Panoptic-DeepLab: A simple, strong, and fast baseline for bottom-up panoptic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 12475–12485.
[116] E. Shelhamer, K. Rakelly, J. Hoffman, and T. Darrell, "Clockwork ConvNets for video semantic segmentation," in *Computer Vision–ECCV*. Amsterdam, The Netherlands: Springer, Oct. 2016, pp. 852–868.

CHIRANJEEVI KARRI received the B.Tech. degree from JNTU Hyderabad, in 2003, and the M.Tech. and Ph.D. degrees from the Veer Surendra Sai University of Technology, India. Both the M.Tech. and Ph.D. dissertations focused on computer vision and biomedical image processing. He is currently a Postdoctoral Researcher with the University of Porto, Porto, Portugal. He has more than 18 years of teaching experience in the areas of computer vision, soft computing, machine learning, deep learning, and biomedical image processing. He did postdoctorals with the Computational Clinical Imaging Group (CCIG), Champalimaud Foundation Centre for the Unknown, Lisbon, Portugal, and the University of Beira-Interior, Portugal. Under his guidance, six Ph.D. scholars were awarded, and five are working on the submission of their thesis. He has presented many research papers related to biomedical image processing with soft computing techniques at international and national levels. He is a speaker at several conferences and a Visiting Faculty Member with Symbiosis University, India. He is an editor of international journals, the author of book chapters, and the conference chair at the international and national levels. His research interests include machine learning and deep learning for rectal and breast cancer detection, security, and computer vision. He has experience with Amazon Cloud Services. He acted as a mentor for students from New York, USA. He has a research collaboration with Taif University, Saudi Arabia. He attended summer school with Oxford University, London, U.K. He delivered several webinar talks and gave several guest lectures.

JOSÉ MACHADO DA SILVA received the Licenciado degree in electrical engineering and the Ph.D. degree in electrical and computer engineering from the Faculty of Engineering, University of Porto (FEUP), Portugal, in 1984 and 1998, respectively. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, FEUP, and a Research Manager and a Project Leader with INESC TEC, with teaching and research responsibilities, including analogue and mixed-signal electronics, VLSI design and test, signal processing, and biomedical electronics and instrumentation, in which domains he was a PI or a group leader of 15 national or international projects. He (co-)supervised or is supervising eight Ph.D. students and more than 40 M.Sc. dissertations. He is the co-editor of a book, and coauthored six book chapters, more than 60 papers published in international and national journals and conferences, and one patent. For more information visit the link (https://web.fe.up.pt/∼jms/).

MIGUEL VELHOTE CORREIA (Member, IEEE) received the Graduate degree in electrical and computer engineering and the master's and Ph.D. degrees in the fields of industrial automation and computer vision from the Faculty of Engineering, University of Porto (FEUP), in 1990, 1995, and 2001, respectively. He is currently an Associate Professor with the Department of Electrical and Computer Engineering, FEUP, where he has been taught, since 1998. Since March 2008, he has been a Senior Research Member with INESC TEC, Porto, and the Head of the Bio-Instrumentation Laboratory, Centre for Biomedical Engineering Research. He has participated in more than 20 funded research projects, (co)-supervised ten Ph.D.'s, coauthored over 150 research papers published in peer-reviewed journals and conference proceedings, and he is a co-inventor of three patents. His research interests include electronics and biomedical instrumentation, computational vision, and image and signal processing, with a focus on sensing methods, technologies, and data fusion for the measurement and analysis of human movement, perception, action, and performance. He is also a member of the Portuguese Official Engineers Association, the International Association of Pattern Recognition, through its Portuguese Chapter, and the Co-Founder of the Portuguese Experimental Psychology Association.