IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

*Digital Object Identifier 10.1109/ACCESS.2022.DOI*

# Multi-camera 3D Object Detection for Autonomous Driving Using Deep Learning and Self-Attention Mechanism

**ANANYA HAZARIKA[1] (Student Member, IEEE), AMIT VYAS[1] , MEHDI RAHMATI [1] (Senior Member, IEEE), and YAN WANG[2]**

[1]Department of Electrical Engineering and Computer Science, Cleveland State University, OH 44115 USA (e-mails: {a.hazarika, a.vyas62}@vikes.csuohio.edu and m.rahmati@csuohio.edu)
[2]Ford Motor Company, Dearborn, MI, USA (e-mail: {ywang21}@ford.com)

Corresponding author: Mehdi Rahmati (e-mail:m.rahmati@csuohio.edu).

"The authors would like to express their gratitude to Ford Motor Company for their collaboration, support, and sponsorship of this project under Ford Motor Company's university research program."

**ABSTRACT** In the absence of depth-centric sensors, 3D object detection using only conventional cameras becomes ill-posed and inaccurate due to the lack of depth information in the RGB image. We propose a multi-camera perception solution to predict the 3D properties of the vehicle obtained from the aggregated information from multiple static infrastructure-installed cameras. While a multi-bin regression loss has been adopted to predict the orientation of a 3D bounding box using a convolutional neural network, combining it with the geometrical constraints of a 2D bounding box to form a 3D bounding box is not accurate enough for all the driving scenarios and orientations. This paper leverages a vision transformer that overcomes the drawbacks of convolutional neural networks when there are no external LiDAR or pseudo-LiDAR pre-trained datasets available for depth map estimation, particularly in occluded regions. By combining the predicted 3D boxes from various cameras using an average weighted score algorithm, we determine the best bounding box with the highest confidence score. Comprehensive simulations for performance analysis are shown from the results obtained by utilizing the KITTI standard data generated from the CARLA simulator.

**INDEX TERMS** Autonomous vehicles, deep learning, object detection, vision transformer.

## I. INTRODUCTION

Recently, there has been a significant evolution in automotive technology, including driver assistants and automated driving systems, due to their ability to improve human life by emphasizing higher safety and convenience, boosting mobility, and reducing travel time, particularly in dense urban environments [1]. Thanks to the rapid advancement of deep learning and artificial intelligence in computer vision and robotics, autonomous vehicles are no longer science fiction since they can leverage the impressive outcomes of implementing deep learning. However, as the degree of autonomy increases, extensive testing and training are required [2] in order to be safer than human-controlled cars in perception, planning, and control subtasks [3], [4]. The visual perception of autonomous cars to detect the presence of other vehicles, pedestrians, and other entities, is highly dependent on 3D object detection and pose estimation techniques. Light Detection and Ranging (LiDAR)-based solutions [5] and their point cloud outputs are usually utilized to detect objects in many autonomous vehicle projects [6]–[8]. However, the collected data from LiDAR has to undergo several filtering processes, which include noise removal, downsampling, and transformation, before being applied to real-time models. These extensive processes make the involvement of LiDAR data very expensive. There exist some point cloud multi-view 3D object detection techniques that can also predict the 3D bounding boxes by fusing LiDAR and RGB images [9], [10]. On the other hand, using single-view RGB images can reduce sensor requirements and therefore be less expensive in real-world applications [11], [12]. Considering multi-view RGB images promises the potential for more accuracy in detection than the efforts based on single-view data. This aggregated perceptive information from multiple cameras placed at different locations can be helpful in sensing the overall activity and behavior of the environment, which in turn will be beneficial to detect an accurate model compared

to the single camera scenario [13].

To accurately detect and localize the vehicle, and depending on the application and environment, the sensor's processing and decision-making units [14] may be installed on the vehicle, on a roadside unit, or as a combination of both. In terms of cost, roadside unit installation is the best choice since it can provide the surroundings from a static observer point of view and remove the requirement for the camera and other sensors to be installed on the car, given that there are some vehicles that do not support this technology. While well-developed 2D object detection algorithms are capable of handling large variations in viewpoint and clutter, reliable and accurate 3D object detection remains an open problem despite some promising recent work in this domain [15]. Monocular depth estimation and 3D bounding box generation are among the most crucial challenges in autonomous driving when it comes to detecting the environment. In the absence of depth-centric sensors such as radar, LiDAR, or bird's eye view sensors, object detection using only conventional cameras become ill-posed and inaccurate due to the lack of depth information in the RGB image and hence, localizing the vehicle and generating a 3D bounding box becomes challenging in the 3D space. Furthermore, the geometrical relations between 2D and 3D make it challenging to regress directly using a single-view image. Although LiDAR-based solutions can fill the performance gap between 2D and 3D detections [16], however, it is a challenging task to recover a 7-DoF pose from a 4-DoF image without using any LiDAR sensor. Previous state-of-the-art methods utilize either some external depth estimation networks [17] or assume the prior information is available [18]. Moreover, occlusion can dramatically reduce the camera's ability to detect an object successfully by relying on the local information obtained from the convolutional filters. The only effective way to understand the occluded images due to dark or bad-weather conditions is by inspecting the entire image with a self-attentive mechanism. Considering the challenges faced by the camera's failure to detect occlusion, transformer [19] has been proved as a de facto standard model to detect occlusion successfully. Transformers are robust to occlusion as they are equipped with multi-head self-attention module to encode the images into patches where each attentive weights help to guide the network to perceive the interested region of interest.

In this paper, we propose a multi-camera solution where we detect and estimate 3D bounding boxes for a vehicle from multiple static, fixed cameras and utilize a weighted box selection algorithm to fuse the detected bounding boxes generated from different cameras. The key contributions of this work are as follows:

- We utilize a unique approach by employing a deep Convolutional Neural Network (CNN), specifically designed to map 2D bounding boxes to their 3D counterparts, providing a clear understanding of the object's orientation and dimensions. This architecture is applied across four distinct camera views, each casting a unique perspective on the same object, thereby creating independently generated 3D bounding boxes for the four cameras. We highlight the significant challenge of detecting objects in occluded regions when relying solely on camera-based systems. We utilize a weighted fusion algorithm that depends on the confidence scores of the individual bounding boxes for consolidating the four independent 3D bounding boxes into a single, definitive bounding box with the highest possible accuracy.
- We highlight the importance of integrating a transformer model for accurately detecting occlusion and depth in our scenario. To address these challenges, we leverage a Vision Transformer (ViT) which specializes in generating a pre-trained depth dataset and evaluating occluded images using a self-attentive module. The pre-trained depth dataset, derived from the ViT network, serves as the input for a subsequent depth-guided filtering module which is responsible for increasing the precision of 3D bounding box predictions, enhancing the robustness and reliability of our object detection system.
- We perform comprehensive simulations after fusing the results from multiple cameras and then provide a comparative analysis with the existing similar camera-based techniques on the KITTI standard data generated by the CARLA simulator for different scenarios.

The remainder of this paper is organized as follows. In Sect. II, we discuss the related work and the current literature in the field. In Sect. III, we describe the problem and propose the details of the solution. In Sect. IV, we present the simulation results in the autonomous driving simulator and the relevant comparisons. Finally, in Sect. V, we conclude the paper and provide insight into our future research.

## II. RELATED WORK

For many years, researchers have been studying 2D object detection techniques to produce 4-DoF axis-aligned bounding boxes with center coordinates and 2D size [20]. You Only Look Once (YOLO) is shown as a popular one-stage object detection model due to its ability to perform real-time object detection based on their already learned weights [21], [22]. YOLOv3 [23] utilizes logistic regression and scoring for each bounding box to achieve a better performance than its previous versions. However, YOLOv4 [24] has a faster training phase and is verified as an efficient tool for improving the accuracy of both the classifier and the detector. YOLOv5 [25] is shown to achieve better performance in detecting objects even from infrared images due to its strong real-time processing capabilities with high precision and faster convergence. YOLOv7 [26]–[28] is being claimed as the best algorithm with the best performance compared with the previous versions to overcome the problem of positioning accuracy. Considering multi-stage models, Faster R-CNN [29] has a Region Proposal Network (RPN) which is trained to generate high-quality region proposals for simultaneously predicting object bounds and scores at each location. A hybrid approach by combining Faster RCNN and YOLOv5, known as EnsembleNet, has been shown in [30] to improve the overall perfor-

mance of vehicle detection in dense traffic scenarios. Integration of convolution neural networks with 2D object detection algorithms in predicting 3D bounding boxes are witnessing an immense improvement in performance, leading to many advancements in autonomous driving and robotics [31]–[33]. Authors in [9] claim a high-accuracy 3D object detection model which utilizes a multi-view 3D network with RGB images and LiDAR point cloud as input for prediction of the 3D bounding boxes. Authors in [34] propose an RGB-D based solution in which RGB-D images are input in 2.5-D region proposal, and CNN for extracting the depth features in achieving an accurate 3D bounding box regression. Depth estimation is obtained by lifting the input image to point cloud representation, called pseudo-LiDAR, which is trained with LiDAR-based 3D detection network [35]. A multimodal vehicle detection system has been introduced in [36] that integrates the data from a 3D-LiDAR and a color camera in a ConvNet-based detector for improving vehicle detection. 3D object detection is performed with a single monocular image in [37] for generating a class-specific object proposal network to obtain high-quality detection by assuming prior ground-plane information. In [38], a drone-assisted model has been introduced for capturing images of the crowd by utilizing a deep-learning model from Root Mean Square Propagation (RMSProp) training algorithm which is a derivative of optimized ResNet architecture. A novel architecture comprising of newly developed derivatives of ResNet, DenseNet, and CNN combined into one global classifier for remote ship detection has been presented in [39]. The model introduced in [39] is designed for solving complex detection tasks in ships for several conditions which are trained accordingly. Authors in [15] present a method for 3D object detection and pose estimation from a single image by regressing 3D object properties using a deep CNN and then combining these estimates with geometric constraints provided by a 2D object bounding box to produce a complete 3D bounding box. Authors in [18] propose a 3D object detection framework based on a single RGB image for extracting the underlying 3D information in a 2D image and then determining the accurate 3D bounding box of the object without point cloud or stereo data. Authors in [40] propose a two-stage 3D object detection method aimed at getting the optimal solution of object location in 3D space by regressing two additional 3D object properties by a deep CNN and combined with cascaded geometric constraints between the 2D and 3D boxes. In [41], 3D object detection is performed from a single monocular image by first generating a set of class-specific object proposals, which are then run through a standard CNN pipeline to obtain high-quality object detection. In [42], the uncertainty issues faced in an autonomous vehicle due to sensor measurements are shown and an end-to-end context-aware solution is proposed by introducing the advantage of an Extended Kalman Filter (EKF) and machine learning to estimate the sensor uncertainty [43]. The authors in [44] present a multiple-object tracking system whose design is based on multiple Kalman filters dealing with observations from two different

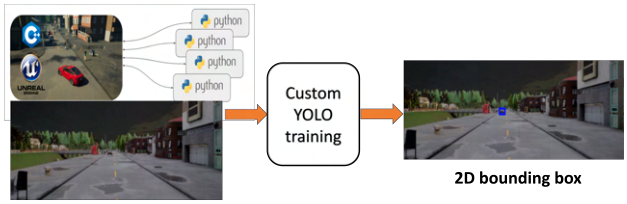



FIGURE 1: Usage of CARLA in dataset generation and training with the custom YOLOv3.

kinds of physical sensors. In [45], a method is proposed known as Deep Stereo Geometry Network (DSGN), which detects 3D objects on a differentiable volumetric representation known by 3D geometric volume, which effectively encodes 3D geometric structure for 3D regular space. To detect objects using a depth network in monocular images, depth-based estimation architecture is shown in [46] which employs stacks of Guided Upsampling Block (GUB) to build a cost-efficient encoder for high-resolution depth map generation. In [47], a set-based global loss has been introduced for bounding box predictions via bipartite matching and a transformer encoder-decoder architecture. A self-supervised depth estimation is shown in [48], which involves a simplified transformer for accurate depth estimation and an efficient source for deployment on GPU platforms. In our proposed solution, we extended the usage of CNN and transformer-generated depth network for the prediction of 3D bounding boxes.

## III. METHOD AND THE PROPOSED SOLUTION

In this section, we describe the challenges in 3D object detection for autonomous vehicles. We present the deep learning-based framework in Sect. III-A. In Sect. III-B, we explain our approach to perform back-projection of the predicted 3D bounding box using Deep Learning. In Sect. III-C, we discuss the transformer-based solution for depth estimation for the 3D bounding box detection in occluded scenarios. Finally, in Sect. III-D, we introduce the solution for fusing the detected object from multi-camera views.

### A. DEEP LEARNING-BASED FRAMEWORK

A deep learning-based approach can be considered in order to achieve a detection of the 3D bounding box in a multi-camera scenario, where the cameras are mounted at different static locations. We referred to the 3D detection based method in [15] for the generation of 3D bounding box from a 2D bounding box and the estimation of pose $(R, T)$ using Deep NN. To generate the 3D bounding box, the dimensions and rotation are obtained using L2 loss and multi-bin architecture, respectively. The 2D object detector has been trained to produce boxes that correspond to the bounding box of the projected 3D box with center $T = [t_x, t_y, t_z]^T$, dimension $D = [d_x, d_y, d_z]$, and orientation $R(\theta, \chi, \alpha)$, which is defined by the Azimuth, Elevation, and Roll angles, respec-

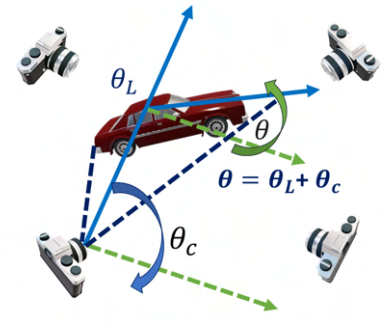



FIGURE 2: The orientation of the vehicle with respect to the camera in a multi-camera scenario. Here, $\theta$ is the global orientation of the vehicle and $\theta_L$ is the local orientation with respect to the angle of the camera denoted by $\theta_c$.

tively. Given the pose of the camera coordinate frame $(R, T)$ and the camera intrinsic matrix $\boldsymbol{K}$, the projection $x$ obtained for a 3D point $X_0 = [X, Y, Z, 1]^T$ is $x = \boldsymbol{K}[RT]X_0$.

We assume that the origin of the object coordinate frame is at the center of the 3D bounding box and the object dimensions $D$ are known. The coordinates of the 3D bounding box vertices can be described as $X_1$=$[d_x/2, d_y/2, d_z/2]^T$,$X_2$= $[-d_x/2, d_y/2, d_z/2]^T$,..,$X_8$ =$[-d_x/2, -d_y/2, -d_z/2]^T$. The constraint requirement for the 3D bounding box is that the 3D bounding box fits tightly into 2D detection window such that each side of the 2D bounding box needs to be touched by the projection of at least one of the 3D box corners. The point-to-point correspondence constraint results in the following equation:

$$x_{min} = \left( \boldsymbol{K} \quad [RT] \quad \begin{bmatrix} d_x/2 \\ -d_y/2 \\ d_z/2 \end{bmatrix} \right). \tag{1}$$

We can derive $x_{max}, y_{min}, y_{max}$ using a similar approach. In this way, we obtain the sides of the 2D bounding box which provides the four constraints on the 3D bounding box. The rotation matrix represents the orientation as a 3×3 orthogonal rotational matrix R derived from the Euler's rotation theorem in CARLA [49].

We describe the loss functions used by CNN for regressing the dimension and orientation of the 3D bounding box in this section. For regressing the dimension estimation, we use the L2 loss which estimate the residual relative to the mean parameter value computed over the training dataset. The loss for dimension estimation $L_{dim}$ is denoted as:

$$L_{dim} = \frac{1}{n} \sum (D^* - D - \delta), \tag{2}$$

where $D^*$ are the groundtruth dimensions of box from CARLA, D are the mean dimensions for objects of a certain category and $\delta$ is the estimated residual obtained in comparison with the mean predicted from the network.

As shown in Fig. 2, the local orientation $\theta_L$ varies with respect to the camera while the global orientation $\theta$ remains constant. Hence, multi-bin architecture regresses this $\theta$ by estimating a confidence probability for each bin such that the output angle lies inside the $i_{th}$ bin. We propose to regress the orientation for the 3D bounding box using a multi-bin architecture, shown in Fig. 3, where the orientation angle is divided or discretized into $n$ overlapping bins. The total loss is calculated as:

$$L_{multibin} = L_{con} + wL_{loc}, \tag{3}$$

where the confidence loss $L_{con}$ gives the softmax loss of each bin and the localization loss $L_{loc}$ tries to minimize the differences between the estimated and ground truth angles in each of the bins. The workflow of multi-camera deep learning-based 3D bounding box detection is shown in Fig. 4. The high-level pseudo-code describing the logic of the proposed framework is shown in Algorithm 1.

**Algorithm 1** Multi-Camera Deep Learning-based 3D Bounding Box Detection

**Input:** Set $J$ number of Camera with images $I$ and the ground truth values of parameters from CARLA simulator

**Output:** Final 3D bounding box coordinates ($x$, $y$, $z$)

```
Initialize cameras in CARLA for scenarios of varying difficulty (easy, moderate, and hard)
for each camera j in 1 to J do
    for each frame i in 1 to I do
        Generate images I_i in all directions [back, front, side-right, side-left]
        Train Custom YOLO model using I_i; estimate bounding box parameters T, D and R to generate labels for ground-truth data
        if R_i ≥ R_th then
            Add I_i to Training_moderate set
        else
            Add I_i to Training_easy set
        end if
        Compute the constraints, [x_min, y_min, x_max, y_max] on the 3D bounding box based on D and R
        Apply L2 loss function to refine the dimension D
        Use a multi-bin architecture to estimate the orientation angle θ
        Formulate the 3D bounding box coordinates (x_j, y_j, z_j) for each camera j
    end for
end for
Employ Weighted Box Fusion (WBF) to fuse the 3D boxes using (12) and (13) for computing the final 3D box (x, y, z)
```

### *B. DEPTH ESTIMATION VIA PERSPECTIVE PROJECTION*

By retrieving distance information of the detected object relative to the camera, depth estimation can unlock several potentials in autonomous driving to perform multiple tasks

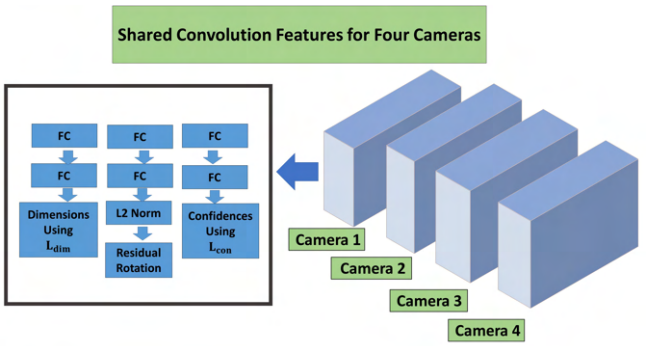



FIGURE 3: Multibin architecture for estimation of orientation and dimension from four cameras. For each camera, it has 3 branches which are used to estimate dimensions, residual rotation ($\cos\theta_i$ and $\sin\theta_i$ of each $i^{th}$ bin) and the confidence, respectively.

like perception, planning, and navigation. As we are not employing any other sensors or injecting any depth maps or pseudo-LiDAR points for depth estimation, we first try to obtain the mapping of the predicted 3D bounding box in the image plane in the form of pixels by using perspective projection. Mapping the 3D object into the image plane helps in determining the reliability and accuracy of object detection using camera sensors.

We consider the 3D coordinates of the predicted bounding box as the world coordinate frame, denoted by $X_w$, the camera coordinate frame is the one which is predicted using CNN and denoted by $X_c$ and $P$ is the image coordinate frame, respectively. The extrinsic calibration parameters are responsible for the transformation from world to camera coordinates, which is a standard 3D coordinate transformation as

$$X_c = \boldsymbol{M_{ex}}[X_w^T, 1]^T, \tag{4}$$

where $\boldsymbol{M_{ex}}$ is the extrinsic calibration matrix of the form

$$\boldsymbol{M_{ex}} = (\boldsymbol{R} - \boldsymbol{R}D), \tag{5}$$

where $\boldsymbol{R}$ is the rotation matrix and $D$ is the location in world coordinates of the center of projection of the camera.

The perspective projection $\tilde{X_c}$ can now be applied to the 3D point which is denoted as:

$$\tilde{X_c} = \frac{f}{x_{3,c}} X_c = \begin{pmatrix} \tilde{x}_{1,c} \\ \tilde{x}_{2,c} \\ f \end{pmatrix}, \tag{6}$$

where $f$ is the focal length of the camera.

The intrinsic calibration matrix ($\boldsymbol{M_{in}}$) is responsible for transforming the 3D image position to pixel coordinates, shown by $P{=}\frac{1}{Z}\boldsymbol{M_{in}}\tilde{X}_c$. Using this approach, we aim to find $Z$ by perspective projection which represents the depth of the static camera from the car. We formulate $P$ to find $Z$, which shows the depth of the static camera from the car. The drawback of regressing depth through the inverse projection of the predicted 3D bounding boxes for every frames is its time-consuming and ill-posed nature. As it will be shown in the results, inaccuracy in depth estimation. i.e., z-direction, for consecutive frames in turning or intersection of a road during inverse projection is a major source of error. Moreover, when there is an occlusion in an image, this method fails in providing an acceptable result. Though inverse depth parametrization obtained through perspective projection is used for the 3D reconstruction of multiple images and simultaneous localization, the predicted depth is not unique due to the occluded 3D scenes arising in hard scenarios that produce the same pictures on the image plane. Considering these drawbacks of perspective projection for depth estimation in occluded scenarios, transformers can be utilized to improve the quality of depth estimation by generating a depth network for occluded regions or those regions which are hard to identify the foreground objects from the background. We then input the pre-trained depth network into a fully convolutional single-stage 3D detection architecture [50] which is being explained in the next section. It does not require any additional pseudo-LiDAR pipelines [51] or any training datasets such as per-pixel depth estimates, 2D bounding box, or a 3D CAD model.

### C. ATTENTION-BASED VISION TRANSFORMER AS PRE-TRAINED DEPTH ESTIMATION

In the realm of monocular vehicle detection, a couple of challenges arise, particularly in areas characterized by occlusion and low light conditions. These complexities highlight the need for depth sensor technologies, such as LiDAR and radar. However, in tackling these demanding scenarios, we propose a novel approach that employs a self-attention-based Vision Transformer (ViT) [52], [53] to create a pre-trained, depth-aware network from single monocular images captured by each camera. The incorporation of this transformer-based depth network serves as an efficient surrogate for LiDAR or radar sensors, facilitating the generation of 3D bounding boxes for detected vehicles. Notably, this depth-aware network is enhanced for depth estimation, a critical factor in reconstructing the 3D bounding box, and offers significant advantages over traditional convolutional networks, particularly in terms of robustness to severe occlusion. This groundbreaking research utilizing the Vision Transformer offers substantial contributions to the field, providing cost-effective and highly precise solutions. These advancements have far-reaching implications and could potentially revolutionize a wide array of applications, spanning from autonomous vehicles to sophisticated remote sensing systems. Vision-based transformers rely on a self-attention mechanism that utilizes their representation as an encoder in the prediction of the depth map. The encoder of the transformer divides the entire image into non-overlapping patches of size $h$ and the features are extracted from those patches in the form of tokens. The relationship between the tokens is derived from self-attention by the application of sequential blocks of multi-headed self-attention (MHSA). The encoder extracts the embeddings of the patches from the image which are non-overlapping in

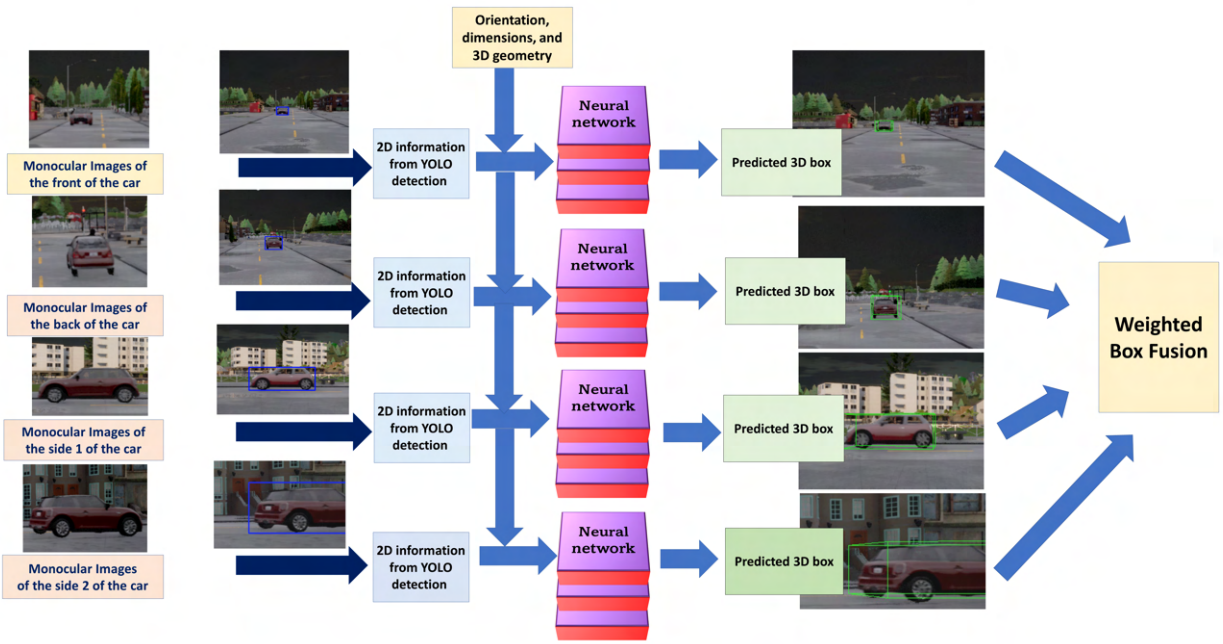



FIGURE 4: Workflow of deep learning based 3D bounding box detection from 4 different cameras with fusion using WBF. Figure shows that the 3D bounding box are generated for each of the four cameras from their estimated 2D bounding box through Deep-CNN layer by the regression of dimension and orientation. The four bounding boxes will be fused through a weighted algorithm to produce the final bounding box with the highest confidence.

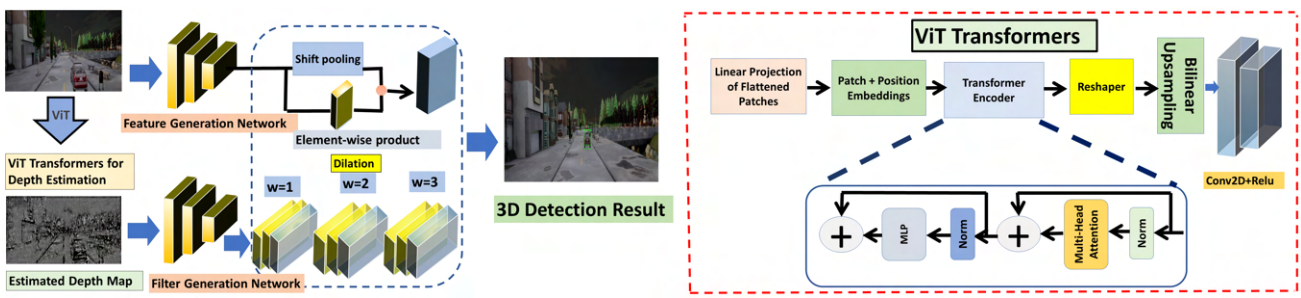



FIGURE 5: The proposed transformer-based depth-guided network for 3D bounding box generation.

nature by processing them with a size of $h^2$ from the image. The extraction is done by applying a ResNet for using the pixel features of the resulting map in the form of tokens. We add a learnable layer to the image embeddings to gain information from their representation. The decoder of the transformer is responsible for receiving all the outputs from the transformer blocks in the encoder by assembling the tokens into image-like representations at various resolutions. ViT shows more accuracy in the monocular depth estimation when being compared with a convolutional network which is shown in Table 1. We apply a scale and shift-invariant trimmed loss operating on an inverse depth prediction. ImageNet [54] pretained weights are used for the initialization of the encoder, but the decoder is initialized randomly. Multi-objective optimization and Adam are being used together with a learning rate of 1e-5 for the backbone and 1e-4 for the decoder. There are three output heads used for reducing the feature dimension in half and for upsampling the predictions to the input resolutions. The depth maps generated from each ViT transformer act as a guided source for learning the local dynamic depth-wise dilated kernels from each of the camera images without the usage of any depth-based sensors. As shown in Fig. 5, the estimated depth network obtained by ViT is the input to a depth-guided filtering module introduced in [50]. This module is a two-branch network consisting of a feature generation network and a filter generation network. ResNet-50 [55] is the backbone of the feature extraction network, and we have pretrained them on our custom Carla-generated dataset. The first three blocks of the ResNet-50 is in the filter generation network so as to reduce computational

costs. Depth-wise local convolution (DLCN) [50] is utilized by the filtering module, which possesses a set of global feature volume filters for operating at its corresponding channels of the ViT generated depth map. The feature volume filters are then converted into location-specific filters for applying the depth-wise and local convolutions to the feature maps. To overcome the huge intra-class and inter-class differences, a dilation rate has been utilized to obtain different sizes of receptive fields by an adaptive function. To solve the issue of the scale-sensitive and meaningless local structure of 2D convolutions, the depth-filtering module has assigned different kernels for different pixels and adaptive receptive fields on a different channel.

The losses used in this network are a classification loss, a 2D regression loss, a 3D regression loss, and a 2D-3D corner loss. The loss is defined by:

$$L = (1 - s_t)^{\gamma}(L_{class} + L_{2d} + L_{3d} + L_{corner}), \quad (7)$$

where $\gamma$ is the classification score. $L_{class}$, $L_{2d}$, $L_{3d}$, and $L_{corner}$ are the classification loss, 2D regression loss, 3D regression loss, and 2D-3D corner loss, respectively. Smooth L1 regression losses have been used for 2D/3D regression.

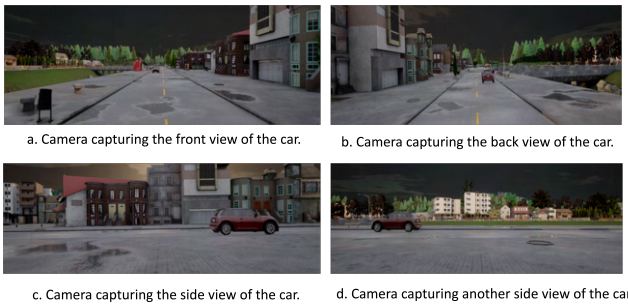
### D. WEIGHTED FUSION OF THE DETECTED BOXES: THE MULTI-CAMERA PERSPECTIVE

In our proposed work, we consider the fusion of the bounding boxes obtained from multiple static cameras to ensure that the object is detected in an accurate way, with an aim to overcome the problems faced due to incomplete observations, recurring detection of the same object, or errors in detection. Fusion can help in identifying those cameras that are able to detect the vehicle accurately. The less the error in the x, y, and z directions of the bounding box, the more accuracy in detection is achieved by the cameras.

For our multi-camera work, we utilize the Weighted Box Fusion (WBF) algorithm, initially introduced in [56]. There are several steps which will be discussed in this section. The first step is to select samples of predicted bounding boxes from four cameras and sort them in decreasing order of their confidence scores for each camera. For each frame, the predicted bounding boxes from four different cameras are matched accordingly under the condition (IoU $\geq THR$) in an iterative manner so as to capture the maximum overlapping between the predicted bounding boxes, and the match produces an optimal output in the form of a fused bounding box for every frame. Here, $THR$ denotes the threshold set for each camera, and we consider our $THR$ to be $0.7$. We need to recalculate the confidence score and coordinates of that bounding box from a camera for a particular frame when it fails to match with the corresponding boxes from the three different cameras using the following formulas given as:

$$C = \frac{\sum_i^M C_i}{M}, \quad (8)$$

$$X_{1,2} = \frac{\sum_i^M C_i * X_{1,2_i}}{\sum_i^M C_i}, \quad (9)$$



a. Camera capturing the front view of the car. b. Camera capturing the back view of the car.

c. Camera capturing the side view of the car. d. Camera capturing another side view of the car.

FIGURE 6: (a) Camera is placed on a pole in front of the car at 180°. (b) Camera is placed at the back of the car. (c) The side camera which is at 90° angle. (d) The fourth camera (at 90°) on the side capturing the car.

$$Y_{1,2} = \frac{\sum_i^M C_i * Y_{1,2_i}}{\sum_i^M C_i}, \quad (10)$$

$$Z_{1,2} = \frac{\sum_i^M C_i * Z_{1,2_i}}{\sum_i^M C_i}. \quad (11)$$

Here, (8) denotes the new confidence score of the bounding boxes obtained by setting the confidence score for the fused box as the average confidence of all boxes from M=4 cameras. Moreover, (9), (10), and (11) show the new coordinates of the unmatched bounding box from a particular camera. Then, we need to re-scale the confidence scores of the four bounding boxes so as to give weight to more prominent boxes using (12) or (13) where $N$ is the number of models.

$$C = C * \frac{min(M, N)}{N}, \quad (12)$$

$$C = C * \frac{M}{N}. \quad (13)$$

Then, we generate the fused box from the weighted sums of the coordinates of the bounding boxes from four different cameras, where the weights are equivalent to the confidence scores for the corresponding boxes. Thus, the fused box has the major contribution of that bounding box of a particular camera with a larger confidence.

## IV. EVALUATION AND RESULTS

In this section, we discuss the generation of datasets and the evaluation techniques used for analyzing the results for different scenarios. In Sect. IV-A, we focus on the utilization of CARLA in preparing the custom datasets. In Sect. IV-B, we describe the methods used for the generation of 2D and 3D bounding boxes. Finally, in Sect. IV-C, we present our simulation results and provide the corresponding discussions.

### A. DATA GENERATION

We utilize CARLA to generate our custom dataset required for training in 3D bounding box generation. Car Learning to Act (CARLA), an open source autonomous driving simulator [49] was built to serve as a modular and flexible API

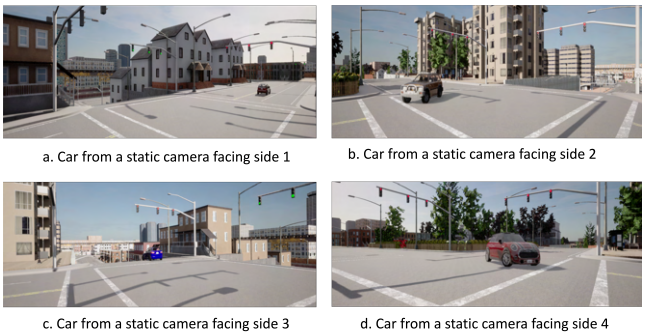



FIGURE 7: a) Capturing the back of the car at one of the turnings. b) Capturing the front of the car at another turning angle. c) Capturing back of the car after a turn. d) Capturing the front of the car at the turning.

for addressing a large range of tasks involved in the problem of autonomous driving. It consists of a scalable client-server architecture which is in charge of almost everything related to the simulation itself, including sensor rendering, physics computation, world-state and actor updates, and much more. With the aim of providing realistic results, the best fit would be running the server with a dedicated GPU, especially when dealing with tasks related to machine learning. The client-side is made up of client modules that are in charge of controlling the logic of actors on the scene and setting world conditions. The dataset has been generated using the Unreal Engine and CARLA, and the labeling of the CARLA-generated dataset is done in KITTI format. As shown in Fig. 1, we utilize CARLA to generate dataset for our training and testing purposes. The datasets have different camera configurations and they were divided into three different categories based on the different camera placement strategies. Given that our dataset is generated using the CARLA simulator, we have set the image resolution to an optimal size of 1284 x 384 to ensure the production of high-quality frames, which is crucial for effective 3D object detection. The high resolution assists in capturing intricate details within the simulated environment, thereby enhancing the precision and reliability of our detection model. We train the Carla-generated dataset with YOLOv3 to produce 2D bounding boxes. These bounding boxes are then regressed into 3D bounding boxes. The dataset is generated under three distinct scenarios, each characterized by varying levels of truncation and occlusion: $(i)$ zero-angle rotations (easy), $(ii)$ smooth or sharp turns (moderate), and $(iii)$ dark or occluded regions (hard). To generate tightly fitting bounding boxes for these diverse scenarios, we utilize unique training strategies suitable to each scenario's specific challenges. For the easy and moderate scenarios, we consider the truncation level of the bounding boxes, which is determined by examining their rotation angle. The threshold of the rotation value, denoted by $(R_{th})$, representing the vehicle's rotation around the Y-axis relative to the camera coordinate system is used to distinguish between the easy and moderate scenarios. $R_{th}$ guides the simulation process, helping to categorize the generated data into appropriate scenarios. Once categorized, this data is saved and subsequently used for both training and testing purposes. This approach ensures a robust and adaptive training process that can effectively handle a wide range of driving scenarios. For the hard scenario, we utilize a ViT transformer for generating a pre-trained depth network which acts as an input to a depth-guided filtering module along with the existing CARLA-generated dark or occluded datasets for detecting the vehicles using only multi-cameras. We utilize the ViT transformer in hard scenario (see Fig. 5) because this transformer helps to overcome the major flaw of CNN's pooling layers for occluded regions, which fails to extract valuable information by ignoring the relationship between the occluded part of images and the whole. The CARLA uses Unreal Engine coordinates to get the vehicle location and then transforms them to 2D-plane coordinates. However, the generation of 3D bounding boxes in the CARLA simulator can be a computationally intensive task, especially when dealing with large datasets or high-resolution images. This may result in an increase in processing time and memory usage, affecting the overall performance and efficiency of the system. To optimize the processing time and memory usage, we have utilized a computationally powerful GPU consisting of GPU acceleration libraries such as CUDA for boosting up the processing time in bounding boxes generation. Our proposed approaches consist of several robust layers to reduce overfitting and computation complexity in order to improve the accuracy of the bounding box detection.

### B. MULTI-CAMERA BOUNDING BOX GENERATION

**2D and 3D Bounding Box Prediction:** Usually the detection of the 3D bounding box is done for the cars from a moving vehicle where the camera is not static with reference to the detected objects and the angles to all the cars remains constant [15]. However, in our proposed solution, we extend this for the prediction of 3D bounding boxes by placing four different cameras in fixed positions under two different angular conditions as shown in Fig. 6 and Fig. 7. For easy and moderate scenarios, the 2D bounding box is generated in real-time by training the YOLOv3 model on our custom CARLA generated dataset. With the usage of transfer learning, we made modifications in the convolution layers of YOLOv3.cfg file by changing the number of classes in the YOLO layer and filters in the convolution layer. We then trained this custom model using darknet53.conv.74 weights (initial YOLO weights) and annotations are obtained in YOLOv3 format based on the labelling of one class (i.e car). YOLOv3 utilizes a total of 106 layer fully convolutional (FC) network to perform the detection of the 2D bounding box by additionally stacking another 53 FC layer network for detection task with the initial 53 FC layer of network trained in our custom datasets. For the hard scenario, the 2D dimensions of the bounding box are predicted from the regression head of the single-stage detector with prior-based 2D-3D anchor boxes [21], [57]. These anchor boxes are first

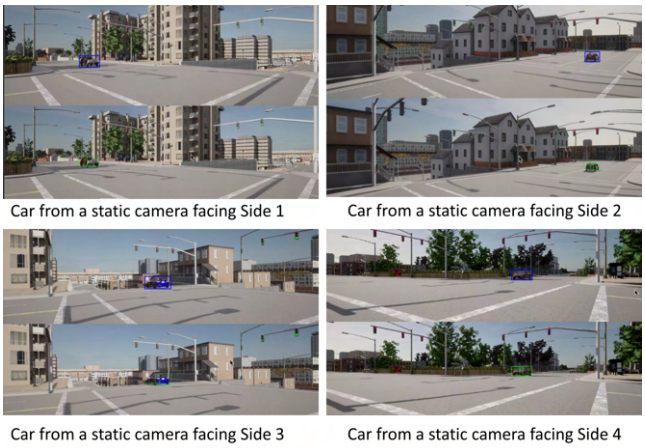



FIGURE 8: Results showing the detected 2D and 3D bounding box for no turning (easy) scenarios.


FIGURE 9: Results showing the detected 2D and 3D bounding box for smooth or sharp turning (moderate) scenarios.

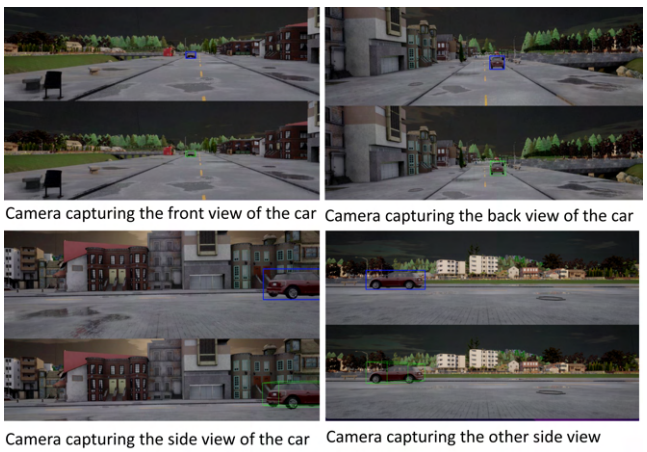



FIGURE 10: Results showing the detected 2D and 3D bounding box for dark or occluded regions (hard) scenarios.

defined on the 2D space as defined in [57], and then it uses the corresponding priors to calculate the part of it in 3D space.

We utilize a VGG16 CNN in easy and moderate scenarios to generate a 3D bounding box by regressing the orientation and dimension using the multi-bin approach and L2 loss respectively [15]. Neural-net takes input images of size 224x224 and predicts the orientation and relative dimension of that object to the class average. We perform training with over 100 images for the four cameras and the network is trained at a learning rate of 0.0001. The 3D bounding box is then predicted from those estimated YOLOv3 generated 2D bounding box when their IoU scores exceed a threshold of 0.7. For the hard scenario, the 3D predicted bounding box is an outcome of 2D-3D anchor-based transformation obtained from the feature generation network of the depth-guided filtering module introduced in [50]. Figs. 8, 9, and 10 show the 2D and 3D bounding box of the detected vehicle obtained during simulations for easy, moderate and hard scenarios. In this paper, we employ two optimal camera configurations to provide a comprehensive 360-degree view of the vehicle. These configurations are outlined as follows: (i) In the first scenario, depicted in Fig. 6, we strategically position cameras in four distinct directions to track the vehicle's movement. These cameras are aligned in such a way that they simultaneously capture the front, back, and both sides of the same car. They are mounted with the front and back cameras angled directly towards the car at 0 degrees, while the side cameras observe the car from a 90-degree angle. (ii) In the second scenario, as illustrated in Fig. 7, we install the four cameras at four different arrangements to capture the vehicle's front and back as it makes a turn. Specifically, two cameras are positioned to record the car's front and back before the turn, while the remaining two cameras capture the vehicle's front and back as it navigates the intersection turn.

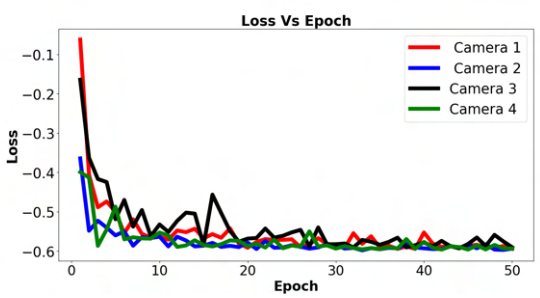



FIGURE 11: Loss vs epoch for the four cameras.

### C. SIMULATION RESULTS AND DISCUSSIONS

In this section, we present the simulation results and the corresponding discussions. We have computed the results for three scenarios, i.e., easy, moderate, and hard, based on different circumstances in driving. In order to generate a dataset using CARLA and train the neural networks on them in our proposed DNN-based and ViT transformer-based approaches, it is essential to meet the following re-

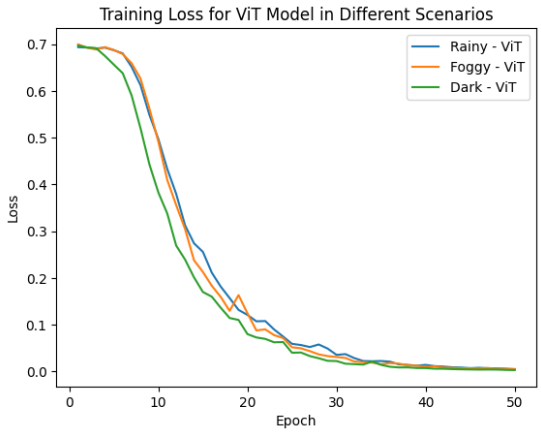



FIGURE 12: Comparison of training loss for several occluded scenarios utilizing the ViT transformer

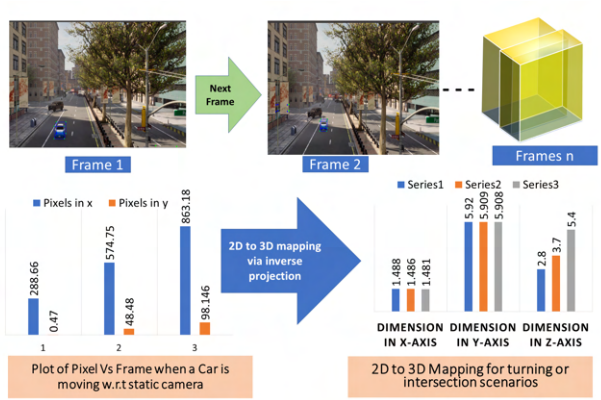



FIGURE 13: 2D-3D mapping for consecutive sample frames via inverse projection for the depth estimation process.

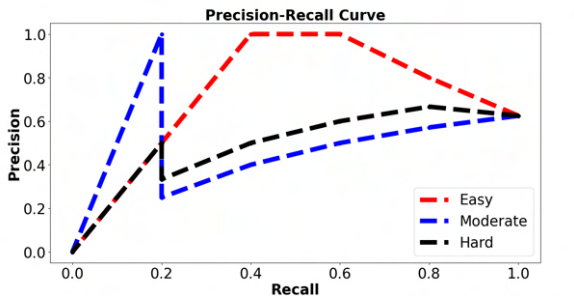



FIGURE 14: Precision vs recall for easy, moderate, and hard scenarios.

quirements: $(i)$ System Compatibility. CARLA is compatible with both Windows and Linux operating systems and it is built from the Unreal Engine, which supports cross-platform development. $(ii)$ Adequate GPU. A minimum of 6 GB GPU is required to run CARLA and to perform their training and testing for realistic simulations. However, it is recommended to have an 8 GB dedicated GPU for machine learning purposes. $(iii)$ Sufficient Disk Space. The installation of CARLA requires approximately 20 GB of disk space. $(iv)$ Processing time. These hardware and software specifications contribute directly to the processing time due to the large network architecture of our proposed approaches by enabling atleast 30 frames per second (fps) or more for a real-time performance. The training loss of the four different cameras is shown w.r.t to their epochs in Fig. 11. The loss curves of the four cameras are shown to undergo a steep decrease from their $4^{th}$-$5^{th}$ epoch, implying that the model performs better on our CARLA-generated custom training datasets. In Fig. 12, we leverage Carla's simulation settings to generate 3D detection data for challenging scenarios - fog, rain, and night. We train the Vision Transformer (ViT) models on this data to assess their performance. Figure 12 offers valuable insights by showing better capabilities of this model in detecting dark regions compared to rainy and foggy weather conditions. Figure 13 shows the 2D-3D mapping of the predicted bounding box for consecutive frames in a turning or intersection scenario. Utilizing the inverse projection method outlined in Section III-B and illustrated in Fig. 13, we determine the optimal distance for different frames where the distance exhibits fluctuations, ranging between 0.9 to 2 m. Due to the non-linearity of the depth w.r.t the static camera, the depth estimation by inverse projection is not applicable for all scenarios. It can be seen in Fig. 13 that the z-dimensions (denoting the depth w.r.t the camera) obtained during mapping are not the same for every frame from the same static camera. The linearity of depth for every frame is being inconsistent as the z-dimension in $2^{nd}$ frame varies by

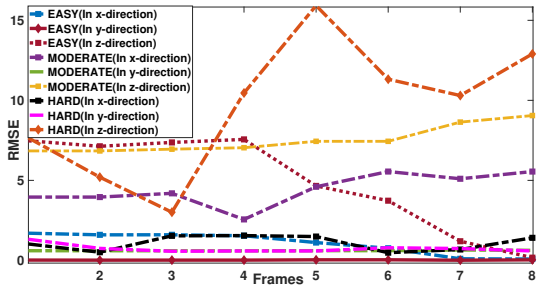



FIGURE 15: RMSE for easy, moderate, and hard scenarios.

| Method | Modality | Easy | Moderate | Hard |
|---|---|---|---|---|
| OFT-Net [58] | Camera | 2.50 | 3.28 | 2.27 |
| CaDDN [59] | Camera | 19.71 | 13.41 | 11.46 |
| AM3D [60] | Camera | 21.48 | 16.08 | 15.26 |
| MonoDTR [61] | Camera | 21.99 | 15.39 | 12.73 |
| MonoDETR [62] | Camera | 24.52 | 16.26 | 13.93 |
| MonoCon [63] | Camera | 22.50 | 16.46 | 13.95 |
| DD3D [51] | Camera | 23.19 | 16.87 | 14.36 |
| MonoDDE [64] | Camera | 24.93 | 17.14 | 15.10 |
| GCDR+DDA [65] | Camera + Pseudo-lidar | 25.21 | 17.25 | 13.53 |
| PS-fld [66] | Pseudo-Stereo Cameras | 23.74 | 17.74 | 15.14 |
| CIEF [67] | Depth-aware Camera | 31.55 | 20.95 | 17.83 |
| **Ours** | Multi-camera | **54.1** | **44.7** | **27.6** |

TABLE 1: Comparative results on camera-based techniques for 3D object detection.

0.9 and the z-dimension varies by 1.7 for $3^{rd}$ frame. Fig. 14 shows the relationship between the accuracy and precision of the easy, moderate and hard approaches. In Fig. 14, it shows a $95\%$ precision for a $40\%$ recall for easy approach, a precision of $37\%$ for a $40\%$ recall for moderate approach, and a precision of $48\%$ for a $40\%$ recall for hard approach. Fig. 15 shows the Root Mean Squared Error (RMSE) between the predicted and the ground truth of the 3D bounding box of some continuous samples of frames for the three different mentioned approaches. We consider RMSE, a widely used metric in statistics and machine learning to evaluate the quantitative predictions of our proposed approaches by calculating the square root of the mean of the squared differences between the predicted and actual dimensions of our proposed 3D bounding box.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(B_i - \hat{B}_i)^2}{N}}, \tag{14}$$

where, $B_i \in \{x_i, y_i, z_i\}$ is the dimension of ground-truth value of the 3D bounding-box for the $i^{th}$ frame, N is the total number of frames in the dataset, and $\hat{B}_i \in \{\hat{x_i}, \hat{y_i}, \hat{z_i}\}$ is the predicted 3D bounding-box dimension of $i^{th}$ frame. Average Precision (AP) is another popular evaluation metric for comparison between different monocular approaches to identify the average level of accuracy in the prediction of a tight bounding box from the overall detected bounding box. This metric is usually applied to situations where we need to represent the identified objects within an image with a bounding box. As machine learning models do not always generate perfect bounding boxes, there may be many bounding boxes detected for each object which are not tight enough to include areas containing the object. Considering these limitations, we utilize Precision to find the proportion of true positives (correctly identified objects) among all positive detections. To compute Average Precision, we first generate a precision-recall curve where the curve is created by plotting precision values (y-axis) against recall values (x-axis) at various thresholds. Recall, or sensitivity measures the proportion of actual positives that are correctly identified. After creating the precision-recall curve, the AP is calculated as the area under this curve (AUC). A higher AP score defines a model's ability to accurately detect objects within images. The three different approaches were evaluated using the AP shown in Table 1 at an Intersection-over-Union (IoU) value equal to 1.7. Table 1 shows that our proposed approaches outperform the existing monocular techniques by a range of $10\%$ to $24\%$.

## V. CONCLUSION AND FUTURE WORK

In this work, we presented a robust solution for an infrastructure-mounted multi-camera object detection system for detecting an autonomous vehicle in different scenarios. In order to create an environment-friendly ecosystem for LiDAR-free driving, we used the fixed location of multiple cameras to predict the dimensions and orientation of the 3D bounding box of the detected vehicle. The fusion of 3D bounding boxes was performed via a weighted fusion algorithm. Given the challenges of camera-based object detection arising due to their lack of depth information and the presence of occlusion, we utilized a ViT transformer to generate a pre-trained network to be fed into a depth-guided filtering module for the prediction of a 3D bounding box in difficult scenarios. This application can be used in autonomous vehicles in areas such as indoor garages or any other GPS-denied environment. Simulations and analysis were performed to show the efficiency of both models in different scenarios. However, the high computational requirements while training and deploying these models lead to additional expensive costs for fulfilling the demands for significant computational resources. So, we aim to introduce a cost-efficient strategy in our future work by enabling optimal sensor placement for maximum coverage from the combined field of view projected by each of the sensors with accurately high-object detection. This novel framework will be proposed for optimal sensor configuration through the selection of highly optimized locations and orientation of each sensor with the utilization of a minimal number of camera sensors. In the future, as we focus on real-time vehicle tracking, we will enable instant joining and leaving of the cameras for full coverage due to the uncertainty in the movement of the vehicle. Furthermore, our future aspirations include making contributions towards the development of a resilient 3D object detection model specifically designed to enhance accuracy amidst challenging conditions such as fog, rain, and darkness. The main goal of this detection model will be to refine the detection precision, ensuring reliable operation in diverse and adverse weather scenarios.

**Acknowledgements:** The authors would like to thank Ford Motor Company for their collaboration and sponsorship through the University Research Program (URP) Award.

## REFERENCES

[1] J. B. Manchon, M. Bueno, and J. Navarro, "From manual to automated driving: How does trust evolve?" Theoretical Issues in Ergonomics Science, vol. 22, no. 5, pp. 528–554, 2021.

[2] "Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles," published: 04-30-2021, Accessed: 05-07-2022. [Online]. Available: https://www.sae.org/standards/content/j3016-202104.

[3] K. Yoneda, N. Suganuma, R. Yanase, and M. Aldibaja, "Automated driving recognition technologies for adverse weather conditions," IATSS Research, vol. 43, no. 4, pp. 253–262, 2019.

[4] Y. Abdolahi, S. Yousefi, J. Tavoosi et al., "A new self-tuning nonlinear model predictive controller for autonomous vehicles," Complexity, vol. 2023, 2023.

[5] J. Leonard, J. How, S. Teller, M. Berger, S. Campbell, G. Fiore, L. Fletcher, E. Frazzoli, A. Huang, S. Karaman et al., "A perception-driven autonomous urban vehicle," Journal of Field Robotics, vol. 25, no. 10, pp. 727–774, 2008.

[6] Y. Wu, Y. Wang, S. Zhang, and H. Ogai, "Deep 3d object detection networks using lidar data: A review," IEEE Sensors Journal, vol. 21, no. 2, pp. 1152–1171, 2021.

[7] B. Yang, W. Luo, and R. Urtasun, "Pixor: Real-time 3d object detection from point clouds," in Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2018, pp. 7652–7660.

[8] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud

based 3d object detection," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 4490–4499.

[9] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3d object detection network for autonomous driving," in Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2017, pp. 1907–1915.

[10] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from rgb-d data," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 918–927.

[11] F. Chabot, M. Chaouch, J. Rabarisoa, C. Teuliere, and T. Chateau, "Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2040–2049.

[12] J. Ku, A. D. Pon, and S. L. Waslander, "Monocular 3d object detection leveraging accurate proposals and shape reconstruction," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2019, pp. 11 867–11 876.

[13] A. Gupta, A. Anpalagan, L. Guan, and A. S. Khwaja, "Deep learning for object detection and scene perception in self-driving cars: Survey, challenges, and open issues," Array, vol. 10, p. 100057, 2021.

[14] A. Choudhari, S. Talkar, P. Rayar, and A. Rane, "Design and manufacturing of compact and portable smart cnc machine," in Proceedings of International Conference on Intelligent Manufacturing and Automation, H. Vasudevan, V. K. N. Kottur, and A. A. Raina, Eds. Singapore: Springer Singapore, 2020, pp. 201–210.

[15] A. Mousavian, D. Anguelov, J. Flynn, and J. Kosecka, "3d bounding box estimation using deep learning and geometry," in Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, 2017, pp. 7074–7082.

[16] X. Weng and K. Kitani, "Monocular 3d object detection with pseudo-lidar point cloud," in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops, Oct 2019.

[17] C. Reading, A. Harakeh, J. Chae, and S. L. Waslander, "Categorical depth distribution network for monocular 3d object detection," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 8555–8564.

[18] B. Li, W. Ouyang, L. Sheng, X. Zeng, and X. Wang, "Gs3d: An efficient 3d object detection framework for autonomous driving," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 1019–1028.

[19] J. Heo, Y. Wang, and J. Park, "Occlusion-aware spatial attention transformer for occluded object recognition," Pattern Recognition Letters, vol. 159, pp. 70–76, 2022.

[20] M. A. Fischler and R. A. Elschlager, "The representation and matching of pictorial structures," IEEE Transactions on computers, vol. 100, no. 1, pp. 67–92, 1973.

[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779–788.

[22] J. Lee and K.-i. Hwang, "Yolo with adaptive frame control for real-time object detection applications," Multimedia Tools and Applications, vol. 81, no. 25, pp. 36 375–36 396, 2022.

[23] A. Farhadi and J. Redmon, "Yolov3: An incremental improvement," in Computer Vision and Pattern Recognition, vol. 1804. Springer Berlin/Heidelberg, Germany, 2018, pp. 1–6.

[24] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," arXiv preprint arXiv:2004.10934, 2020.

[25] S. Li, Y. Li, Y. Li, M. Li, and X. Xu, "Yolo-firi: Improved yolov5 for infrared image object detection," IEEE access, vol. 9, pp. 141 861–141 875, 2021.

[26] Y. Zhang, Y. Sun, Z. Wang, and Y. Jiang, "Yolov7-rar for urban vehicle detection," Sensors, vol. 23, no. 4, 2023.

[27] N. Ma, X. Zhang, H.-T. Zheng, and J. Sun, "Shufflenet v2: Practical guidelines for efficient cnn architecture design," in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 116–131.

[28] M. Hussain, H. Al-Aqrabi, M. Munawar, R. Hill, and T. Alsboui, "Domain feature mapping with yolov7 for automated edge-based pallet racking inspections," Sensors, vol. 22, no. 18, p. 6927, 2022.

[29] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," CoRR, vol. abs/1506.01497, 2015. [Online]. Available: http://arxiv.org/abs/1506.01497

[30] U. Mittal, P. Chawla, and R. Tiwari, "Ensemblenet: A hybrid approach for vehicle detection and estimation of traffic density based on faster r-cnn and yolo models," Neural Computing and Applications, vol. 35, no. 6, pp. 4755–4774, 2023.

[31] I. Misra, R. Girdhar, and A. Joulin, "An end-to-end transformer model for 3d object detection," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 2906–2917.

[32] B. Brown, E. Laurier, and E. Vinkhuyzen, "Designing motion: Lessons for self-driving and robotic motion from human traffic interaction," Proceedings of the ACM on Human-Computer Interaction, vol. 7, no. GROUP, pp. 1–21, 2023.

[33] M. Naya-Varela, S. Guerreiro-Santalla, T. Baamonde, and F. Bellas, "Robobo smartcity: An autonomous driving model for computational intelligence learning through educational robotics," IEEE Transactions on Learning Technologies, pp. 1–17, 2023.

[34] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning rich features from rgb-d images for object detection and segmentation," 2014.

[35] J. Beltrán, C. Guindel, F. M. Moreno, D. Cruzado, F. García, and A. De La Escalera, "Birdnet: A 3d object detection framework from lidar information," in 2018 21st International Conference on Intelligent Transportation Systems (ITSC), 2018, pp. 3517–3523.

[36] A. Asvadi, L. Garrote, C. Premebida, P. Peixoto, and U. J. Nunes, "Multimodal vehicle detection: fusing 3d-lidar and color camera data," Pattern Recognition Letters, vol. 115, pp. 20–29, 2018.

[37] M. Chen, H. Zhao, and P. Liu, "Monocular 3d object detection based on uncertainty prediction of keypoints," Machines, vol. 10, no. 1, 2022.

[38] M. Woźniak, J. Siłka, and M. Wieczorek, "Deep learning based crowd counting model for drone assisted systems," in Proceedings of the 4th ACM MobiCom Workshop on Drone Assisted Wireless Communications for 5G and Beyond, 2021, pp. 31–36.

[39] M. Woźniak, M. Wieczorek, and J. Siłka, "Deep neural network with transfer learning in remote object detection from drone," in Proceedings of the 5th International ACM Mobicom Workshop on Drone Assisted Wireless Communications for 5G and Beyond, 2022, pp. 121–126.

[40] J. Fang, L. Zhou, and G. Liu, "3d bounding box estimation for autonomous vehicles by cascaded geometric constraints and depurated 2d detections using 3d results," CoRR, vol. abs/1909.01867, 2019.

[41] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3d object detection for autonomous driving," in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 2147–2156.

[42] M. Alharbi and H. A. Karimi, "Context-aware sensor uncertainty estimation for autonomous vehicles," Vehicles, vol. 3, no. 4, pp. 721–735, 2021.

[43] A. Gupta, A. Choudhari, T. Kadaka, and P. Rayar, "Design and analysis of vertical vacuum fryer," in Proceedings of International Conference on Intelligent Manufacturing and Automation. Springer, 2019, pp. 133–149.

[44] D. Y. Kim and M. Jeon, "Data fusion of radar and image measurements for multi-object tracking via kalman filtering," Information Sciences, vol. 278, pp. 641–652, 09 2014.

[45] Y. Chen, S. Liu, X. Shen, and J. Jia, "Dsgn: Deep stereo geometry network for 3d object detection," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020.

[46] M. Rudolph, Y. Dawoud, R. Güldenring, L. Nalpantidis, and V. Belagiannis, "Lightweight monocular depth estimation through guided decoding," 2022. [Online]. Available: https://arxiv.org/abs/2203.04206

[47] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," CoRR, vol. abs/2005.12872, 2020. [Online]. Available: https://arxiv.org/abs/2005.12872

[48] J. Yang, L. An, A. Dixit, J. Koo, and S. I. Park, "Depth estimation with simplified transformer," 2022. [Online]. Available: https://arxiv.org/abs/2204.13791

[49] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," 2017. [Online]. Available: https://arxiv.org/abs/1711.03938

[50] M. Ding, Y. Huo, H. Yi, Z. Wang, J. Shi, Z. Lu, and P. Luo, "Learning depth-guided convolutions for monocular 3d object detection," 2019. [Online]. Available: https://arxiv.org/abs/1912.04799

[51] D. Park, R. Ambrus, V. Guizilini, J. Li, and A. Gaidon, "Is pseudo-lidar needed for monocular 3d object detection?" 2021. [Online]. Available: https://arxiv.org/abs/2108.06417

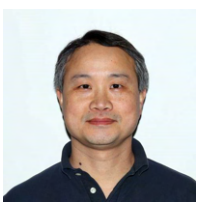
[52] R. Ranftl, A. Bochkovskiy, and V. Koltun, "Vision transformers for dense prediction," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 12 179–12 188.

[53] S. Khan, M. Naseer, M. Hayat, S. W. Zamir, F. S. Khan, and M. Shah, "Transformers in vision: A survey," ACM Computing Surveys (CSUR), 2021.
[54] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in Proceedings of the conference on computer vision and pattern recognition. IEEE, 2009, pp. 248–255.
[55] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778, 2016.
[56] R. Solovyev, W. Wang, and T. Gabruseva, "Weighted boxes fusion: Ensembling boxes from different object detection models," Image and Vision Computing, vol. 107, p. 104117, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0262885621000226
[57] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in Computer Vision – ECCV 2016. Springer International Publishing, 2016, pp. 21–37.
[58] T. Roddick, A. Kendall, and R. Cipolla, "Orthographic feature transform for monocular 3d object detection," published: 2018. [Online]. Available: https://arxiv.org/abs/1811.08188
[59] C. Reading, A. Harakeh, J. Chae, and S. L. Waslander, "Categorical depth distribution network for monocular 3d object detection," 2021. [Online]. Available: https://arxiv.org/abs/2103.01100
[60] X. Ma, Z. Wang, H. Li, P. Zhang, X. Fan, and W. Ouyang, "Accurate monocular object detection via color-embedded 3d reconstruction for autonomous driving," 2019. [Online]. Available: https://arxiv.org/abs/1903.11444
[61] K.-C. Huang, T.-H. Wu, H.-T. Su, and W. H. Hsu, "Monodtr: Monocular 3d object detection with depth-aware transformer," 2022. [Online]. Available: https://arxiv.org/abs/2203.10981
[62] R. Zhang, H. Qiu, T. Wang, Z. Guo, X. Xu, Y. Qiao, P. Gao, and H. Li, "Monodetr: Depth-guided transformer for monocular 3d object detection," 2022. [Online]. Available: https://arxiv.org/abs/2203.13310
[63] X. Liu, N. Xue, and T. Wu, "Learning auxiliary monocular contexts helps monocular 3d object detection," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 36, no. 2, 2022, pp. 1810–1818.
[64] Z. Li, Z. Qu, Y. Zhou, J. Liu, H. Wang, and L. Jiang, "Diversity matters: Fully exploiting depth clues for reliable monocular 3d object detection," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 2791–2800.
[65] H. Hu, M. Zhu, M. Li, and K.-L. Chan, "Deep learning-based monocular 3d object detection with refinement of depth information," Sensors, vol. 22, no. 7, 2022.
[66] Y.-N. Chen, H. Dai, and Y. Ding, "Pseudo-stereo for monocular 3d object detection in autonomous driving," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2022.
[67] Q. Ye, L. Jiang, and Y. Du, "Consistency of implicit and explicit features matters for monocular 3d object detection," 2022. [Online]. Available: https://arxiv.org/abs/2207.07933

ANANYA HAZARIKA (Student Member, IEEE) is a Second Year PhD student at Cleveland State University, Ohio. She has completed her M. Tech from Indian Institute of Information Technology, Guwahati. Her research interests include applications of AI/ML in wireless communication, ultra-low latency communication for extreme environments, reinforcement learning, and Bayesian optimization. She is the President of IEEE Women in Engineering (WIE) at Cleveland State University.

AMIT VYAS is a master's student in Computer Science at Cleveland State University, Ohio. He has completed his B. Tech from Manipal Institute of Technology, Karnataka, India. His research interests include Artificial Intelligence, Deep Learning, and Machine Learning in computer vision, 3D object detection, large Language Models, and NLP.

MEHDI RAHMATI (Senior Member, IEEE) is an Assistant Professor with the Department of Electrical Engineering and Computer Science at Cleveland State University, USA. He is the director of Intelligent Communications and Autonomous Systems Laboratory at Cleveland State University. He received his Ph.D. in Electrical and Computer Engineering from Rutgers University, NJ, in 2020. He has published numerous peer-reviewed conference and journal papers and has received many prestigious awards, including the IEEE Oceanic Engineering Society Young Professional Boost award in 2022-2023, the best demo award at the IEEE International Conference on Sensing, Communication and Networking (SECON'19), and the best paper award at the IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS'17). Currently, he works on wireless communications, Vehicle to Everything (V2X), 5G-enabled smart cities, and Internet of Things.



YAN WANG (Member, IEEE) received the B.S. degree in mechanical engineering in 1994, and the M.S. degree in mechatronics, in 1997, from Tsinghua University, Beijing, China, and the Ph.D. degree in mechanical engineering in 2001 from the University of California, Santa Barbara, CA, USA, focusing on controls. He has since been with Ford Research working on different control problems. He accumulated 15+ years of experience in actuator design and mechatronics controls, with interdisciplinary background in mechanical, electrical, and magnetic systems. He then moved to the field of adaptive and optimal control for automotive applications, focusing on automotive calibration and optimal/adaptive control problems. His main research interests include the application and realtime implementation of advanced/modern control methods, including robust control, adaptive control, system identification, optimal control, model predictive control, and data analytics and machine learning, on vehicle design, control, and calibration. His recent interests include adaptive DOE, AI/ML in system ID and adaptation, real time optimization, sensor fusion and diagnostics, and predictive optimization with preview, etc.