

# Enhancing 3D Object Detection in Autonomous Vehicles Based on Synthetic Virtual Environment Analysis

Vladislav Li<sup>a</sup>, Ilias Siniosoglou<sup>b,c</sup>, Thomai Karamitsou<sup>d</sup>, Anastasios Lytos<sup>d</sup>, Ioannis D. Moscholios<sup>e</sup>, Sotirios K. Goudos<sup>f</sup>, Jyoti S. Banerjee<sup>g</sup>, Panagiotis Sarigiannidis<sup>b,c</sup>, Vasileios Argyriou<sup>a</sup>

<sup>a</sup>“institution –Kingston University” “department –Department of Networks and Digital Media” “city –Kingston upon Thames” “country –United Kingdom” “email –v.li@kingston.ac.uk, vasileios.argyriou@kingston.ac.uk”

<sup>b</sup>“institution –University of Western Macedonia” “department –Department of Electrical and Computer Engineering” “city –Kozani” “country –Greece” “email –isiniosoglou@uowm.gr, psarigiannidis@uowm.gr”

<sup>c</sup>“institution –MetaMind Innovations P.C.” “department –R&D Department” “city –Kozani” “country –Greece” “email –isiniosoglou@metamind.gr, psarigiannidis@metamind.gr”

<sup>d</sup>“institution –Sidroco Holdings Ltd.” “city –Nicosia” “country –Cyprus” “email –tkaramitsou@sidroco.com, alytos@sidroco.com”

<sup>e</sup>“institution –University of Peloponnese” “department –Department of Informatics and Telecommunications” “city –Tripoli” “country –Greece” “email –idm@uop.gr”

<sup>f</sup>“institution –Aristotle University of Thessaloniki” “department –Physics Department” “city –Thessaloniki” “country –Greece” “email –sgoudo@physics.auth.gr”

<sup>g</sup>“institution –Bengal Institute of Technology” “city –Kolkata” “country –India” “email –tojyoti2001@yahoo.co.in”

---

## Abstract

Autonomous Vehicles (AVs) rely on real-time processing of natural images and videos for scene understanding and safety assurance through proactive object detection. Traditional methods have primarily focused on 2D object detection, limiting their spatial understanding. This study introduces a novel approach by leveraging 3D object detection in conjunction with augmented reality (AR) ecosystems for enhanced real-time scene analysis. Our approach pioneers the integration of a synthetic dataset, designed to simulate various environmental, lighting, and spatiotemporal conditions, to train and evaluate an AI model capable of deducing 3D bounding boxes. This dataset, with its diverse weather conditions and varying camera settings, allows us to explore detection performance in highly challenging scenarios. The proposed method also significantly improves processing times while maintaining accuracy, offering competitive results in conditions previously considered difficult for object recognition. The combination of 3D detection within the AR framework and the use of synthetic data to tackle environmental complexity marks a notable contribution to the field of AV scene analysis.

**Keywords:** Augmented Reality, Object Detection, Scene Analysis, Scene Understanding, Object Recognition, Deep Learning, Feature Extraction.

---

## 1. Introduction

In the domain of autonomous driving, scene analysis and comprehension are fundamental for enabling vehicles to perceive and interact with their environment effectively [1] [2]. Autonomous vehicles (AVs) rely on advanced computer vision and machine learning (ML) algorithms to process data from multiple sensors such as cameras, LiDAR, and radar, allowing them to recognize

objects, navigate safely, and make real-time decisions. However, while 2D object detection methods have been widely adopted for these tasks, they are inherently limited in their ability to capture the full three-dimensional nature of the environment, which is crucial for accurately understanding object positions and interactions in real-world scenarios.

A major challenge faced by current AV systems is the transition from 2D to 3D object detection as mentioned in citefawole2024recent and citemao20233d. Projecting 3D bounding boxes into a three-dimensional environment is a more complex and computationally expensive task, especially when the system must handle diverse environmental conditions such as changes in lighting, weather, and sensor perspectives. Traditional 2D methods fall short when detecting objects in such varied conditions, leading to reduced accuracy and safety risks in AV applications. Furthermore, there is a need to efficiently integrate augmented reality (AR) into this process, which could further improve the system's ability to predict and overlay digital elements onto real-world environments for enhanced situational awareness.

This study aims to address these limitations by developing a novel 3D object detection solution that not only predicts accurate 3D bounding boxes but also improves processing times and performance across challenging conditions. Our approach introduces a multimodal architecture that extrapolates 3D information from 2D images, leveraging a synthetic dataset designed to mimic various real-world conditions such as lighting, weather, and camera viewpoints. The results of this work can be applied to improve AV systems' performance in dynamic environments, providing more robust and reliable object detection and localization.

For autonomous vehicles, scene analysis and comprehension play an important role. This includes a wide range of applications such as detecting other vehicles sharing the road, recognizing traffic signs, as well as detecting pedestrians, potential hazards, etc. This deeper understanding is instrumental in making autonomous decisions while integrating the augmented environment onto the vehicle's display systems like heads-up displays (HUDs). This extends to visual scene analysis which is the cornerstone of vehicle environment perception and interaction using advanced computer vision machine learning (ML) algorithms for controlling large amounts of data collected from sensors, such as cameras, LiDAR, and radar. The recognition and interpretation of the ever-changing surroundings allow the vehicle to make informed choices about navigation, safety, and interactions with other road users.

In order to drive safely and effectively, for example, the car must be able to recognise and identify other vehicles, as well as their position and relative speed. Equally important is the recognition of traffic signs and signals, ensuring compliance with traffic regulations and the seamless flow of traffic. Moreover, the accurate detection of pedestrians, cyclists, and potential obstacles is indispensable for avoiding accidents and ensuring the safety of all road users.

In the last decades, advances in computer vision have fostered the design and implementation of object recognition methods, increasing computational performance and lowering process time [3]. These technologies enable the vehicle's onboard computer systems to continuously learn and adapt, improving their ability to recognise and respond to an ever-evolving array of environmental conditions. An important milestone is that in the optimisation phase of such applications, the evaluation of AV image cognition systems can be performed in the virtual and augmented reality domains, utilising the same environment that is also used in virtual applications, like game development engines. As a result, current scene analysis technologies based on object recognition use complex computer vision techniques to detect and track objects in the real world. Examples of such technologies include the You Only Look Once (YOLO) model [4], homomorphic filtering and Haar markers [5] and the Single Shot Detector [6]. The use of Convolutional Neural Networks (CNNs) and Deep Learning (DL) led to faster and more accurate detection processes [7].

However, the AR experience could be improved by projecting 3D objects into the augmented reality space surrounding the user inferred from the real environment.

The aim of this study is to analyse a novel 3D solution that evaluates the performance of the 3D bounding box prediction in various conditions. This work proposes a novel architecture to efficiently produce 3D bounding boxes, superimposed onto the multivariate spatiotemporal view that technologies like advanced AR and AV cognition systems employ to perceive the three-dimensional environment. The produced system is further evaluated on the new synthetic dataset produced to encapsulate a variety of possible environmental conditions, like, camera view, lighting, weather, and sensor readings. Finally, this study evaluates the proposed architecture with other benchmark methods, providing a comparative dimension. The main contributions of this work are as follows:

- A multimodal architecture for efficient object detection and localisation for real-time scene analysis
- A methodology for predicting 3D bounding boxes on the three-dimensional environment, extrapolated from 2D images
- A Novel Synthetic Image dataset for object detection in AV applications with VR scene augmentation
- A comparative study of the efficacy and efficiency of the developed methodology against state-of-the-art techniques

The rest of this paper is organised as follows: 2 provides an overview of related work. 3 describes the proposed architecture. 4 presents results obtained using a novel synthetic image dataset. Finally, 5 concludes this work.

## **2. Overview of Previous Work**

### *2.1. Region-based Feature Extraction Algorithms*

An AR app identifies objects in the real world using ML and computer vision techniques with the goal of overlaying virtual objects in real-time. In recent years, the use of deep CNNs [8] has greatly enhanced the performance and accuracy of object detection and recognition in computer vision. In 2014, Girshick et al. introduced the Regions with CNN features (RCNN) method for object detection [9]. This approach involved first identifying potential object boxes through selective search and then rescaling each box to a fixed-size image for input into a CNN model trained on AlexNet [10] for feature extraction. The object was then detected using a linear SVM classifier, resulting in a significant improvement in mean Average Precision compared to previous methods, but also had a significant drawback of slow detection speed.

In 2014, Girshick et al. introduced the "Regions with CNN features" (RCNN) method for the purpose of object detection, as documented in their seminal work [9]. This pioneering approach signified a significant breakthrough in the realm of computer vision, particularly concerning the enhancement of object detection accuracy. The RCNN methodology employed a dual-stage process. Firstly, it commenced with the utilisation of "selective search" to identify prospective object boxes within an image. Selective search effectively partitioned the image into multiple regions or proposals that were posited as likely candidates harboring objects. These regions were thereby considered as candidate boxes for potential object localisation.

Subsequently, the next steps in the RCNN procedure entailed the resizing of the aforementioned candidate bounding boxes to fit into fixed-size images, rendering them ready for analysis. These standardised images are then subjected to a CNN-based processing, specifically pre-trained on the AlexNet model [10]. The principal role of this CNN was to perform feature extraction in order to discern and capture highly distinctive features of the object in question. Upon feature extraction, the final step of the RCNN methodology involved the employment of a linear Support Vector Machine (SVM) classifier. The SVM classifier was instrumental in effecting classification of the extracted features, thereby ascertaining the presence or absence of a given object within the candidate box. This classification process was the basis of object identification and localisation.

The outcomes of the RCNN approach bore substantial significance. It led to a marked augmentation in the mean Average Precision metric, a pivotal gauge of the efficacy and precision of object detection algorithms. Effectively, it surpassed antecedent methods in its competence to identify objects within images, marking a substantial progression in the arena of computer vision.

Nevertheless, it is worth acknowledging that the RCNN method suffered from a comparatively lengthy detection timeframe which can majorly impact the overall performance. Its sequential operations, such as selective search, CNN-based feature extraction, and SVM classification, made it very computationally intensive and took a long time to process, which limited its usefulness in situations where real-time object detection was needed.

So, while the RCNN approach made it easier to find objects, it required a lot of computing power and time, which meant that more research had to be done to make it work faster. Regardless, its creation was a major turning point in the history of object detection algorithms. It paved the way for later innovations and sped up progress in areas like robotics, autonomous vehicle systems, and many types of computer vision applications.

In an effort to tackle the persistent challenge of slow detection speed in object recognition and localisation, He et al. presented the Spatial Pyramid Pooling Network (SPPNet) as an innovative solution in their seminal work [11]. This architectural paradigm marked a notable milestone in the evolution of computer vision, offering a profound remedy to a long-standing predicament in the field.

The basis of the SPPNet's success lay in its strategic incorporation of a Spatial Pyramid Pooling (SPP) layer, a pivotal component that revolutionised the object detection process. The distinctive feature of this SPP layer was its ability to generate a fixed-length representation that remained invariant to alterations in image size and scale. This attribute had far-reaching implications, particularly in terms of mitigating overfitting issues that had previously plagued object recognition systems. After this initial feature extraction step, the SPPNet employed a sub-region pooling mechanism. This operation entailed dividing the image into spatial bins, enabling the aggregation of features from each bin to create fixed-length representations that were conducive for detector training.

One of the most notable outcomes of this innovative approach was a remarkable acceleration in processing speed, especially during testing. The SPPNet method proved to be a significant leap forward, with testing times ranging from 24 to 102 times faster than the previously established RCNN approach. This acceleration in speed held profound implications for real-time and time-sensitive applications, particularly in contexts like autonomous vehicles, robotics, and augmented reality.

In 2015, Girishick improved the previous two architectures with Fast RCNN [12]. This network trains both a detector and a bounding box regression simultaneously with the same



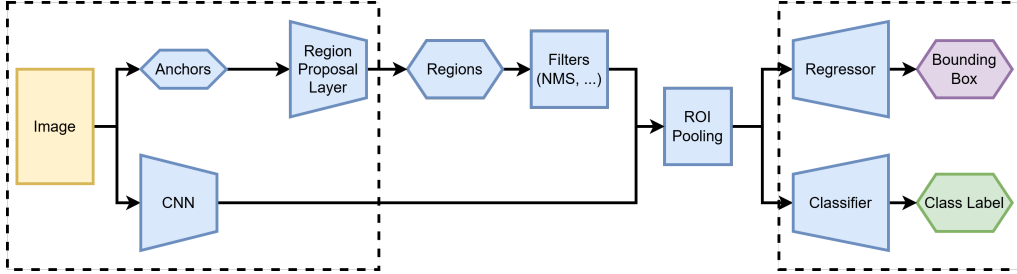


Figure 1: FRCNN architecture.

configuration. However, the speed limitation persisted. The same year, Ren et al. introduced the Faster RCNN detector [13], which was the first deep learning detector to almost achieve real-time detection through end-to-end training. This architecture employed the Region Proposal Network (RPN) to speed up the detection process, and several variants have been proposed since then to reduce computational redundancy [14], [15], [16]. In particular, Cao et al. (2020) [17] introduced the D2Det method based on the Faster R-CNN framework, which processes Region of Interest (ROI) features through two stages: high-density local regression and discriminant ROI pooling. The method replaces the Faster RCNN offset regression with a local dense regression block. Girishick furthermore introduced an enhancement to the existing architectural paradigms in the form of the Fast RCNN [12]. This novel network configuration entailed the simultaneous training of both an object detector and a bounding box regression component, all within the same unified architecture. However, it is noteworthy that the issue of computational speed constraints persisted despite this development.

The Fast RCNN model builds upon the existing state-of-the-art, enhancing efficiency. It exhibits the capability to concurrently train two fundamental components within the same system, i) an object detector and ii) a bounding box regression module, incorporated under the same framework. This integrated approach was a significant stride towards a more streamlined and coherent training process. Nevertheless, the overarching challenge of computational speed constraints persisted as an obstinate issue in the field.

In the same time, Ren et al. introduced the Faster RCNN detector [13], a groundbreaking endeavor that charted a course toward the realization of real-time object detection through the prism of end-to-end training. The Faster RCNN architecture marked a seminal turning point in the pursuit of swifter detection capabilities. At its core, it introduced the Region Proposal Network (RPN), a component specifically designed to expedite the object detection process. The RPN's mandate involved the generation of region proposals, a facet that greatly enhanced the network's adeptness in efficiently discerning objects within complex scenes.

The introduction of the Faster RCNN model had an indelible impact on the landscape of computer vision. It not only ushered in the possibility of near real-time object detection but also spurred a wave of innovative architectural variants. These variations, with an overarching focus on curtailing computational redundancy [14], [15], [16], explored diverse avenues to further amplify the velocity and efficiency of object detection while preserving precision.

Among these progressive adaptations, the D2Det method, introduced by Cao et al. in 2020 [17], stands out as an exemplar of innovation based on the Faster R-CNN framework. The D2Det method harnesses a sophisticated two-stage process for handling Region of Interest (ROI) features. In the initial phase, high-density local regression is employed to finetune the localization

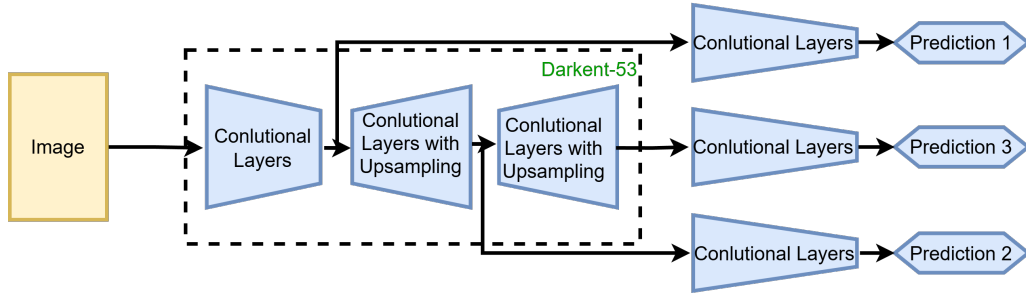


Figure 2: YOLO architecture.

of objects, infusing a heightened degree of precision into the detection process. Subsequently, in the second stage, a discriminant ROI pooling mechanism extracts distinctive features from the ROIs. Notably, D2Det departs from the Faster RCNN's offset regression by adopting a local dense regression block, thus augmenting the precision and robustness of the object detection process.

The progress made by researchers and the path from Fast RCNN to Faster RCNN and beyond show that the goal of real-time object recognition is still being worked on and will continue to get better. There is a chance that these advances will change many areas, such as autonomous systems, surveillance, robots, and augmented reality. At the cutting edge of progress in computer vision and deep learning is the never-ending search for faster, more accurate, and more efficient ways to find objects.

The methodologies discussed above fall under the classification of two-stage detectors due to their characteristic two-step process: initially generating regions of interest (ROIs) and subsequently executing detection and recognition. In 2016, Joseph et al. introduced a noteworthy departure from this convention, presenting a one-stage detector known as You Only Look Once (YOLO) [18]. YOLO epitomized a pioneering paradigm shift in the realm of object detection, manifesting as a single network architecture capable of processing the entirety of an image within a solitary step, which resulted in substantially expedited processing times.

## 2.2. Segmentation-based Feature Extraction

The YOLO methodology operates by segmenting the image into distinct regions and concurrently predicting bounding boxes for each of these regions. This one-step processing paradigm marked a significant departure from the multi-step procedures of its two-stage counterparts.

YOLO was a game-changer in the field, representing a revolutionary approach to object detection. Unlike its two-stage counterparts, YOLO employed a single neural network architecture, capable of processing an entire image in a solitary pass. This unique design offered a significant advantage in terms of processing speed, effectively reducing detection times. The core principle underlying YOLO's functionality involved the division of the image into discrete regions, with the network making concurrent predictions for bounding boxes associated with each region. This approach eliminated the need for sequential processing, offering a substantial boost in efficiency. In the subsequent years, YOLO underwent iterations with the introduction of YOLO v2 and v3, aimed at enhancing prediction accuracy [19], [20], while subsequent versions, v5 through v8 focus on prediction efficiency, accuracy, speed and deployment optimisation.

To enhance the accuracy of bounding box localization,  $DIoU$  loss is employed due to its demonstrated improvement in performance when used with the YOLO algorithm [21].  $DIoU$  represents an advancement of the IoU metric (1), specifically targeting the optimization of bounding box predictions.

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}} = IoU(B_p, B_r) = \frac{B_p \cap B_r}{B_p \cup B_r} \quad (1)$$

and the distance

$$Loss_{IoU} = 1 - IoU \quad (2)$$

In this context,  $B_p$  and  $B_r$  represent the predicted and actual bounding boxes, respectively. The term  $DIoU$  improves upon  $IoU$  by factoring in the square of the diagonal  $d_B$  of the smallest bounding box  $B_o$  that encompasses both  $B_p$  and  $B_r$ . Therefore, the equation is as follows:

$$DIoU = IoU - \frac{\sqrt{(B_p^2) - (B_r^2)^2}}{d_B^2} \quad (3)$$

and the resulting loss function.

$$L_{DIoU} = 1 - DIoU = 1 - IoU - \frac{\sqrt{(B_p^2) - (B_r^2)^2}}{d_B^2} \quad (4)$$

This function, by addressing the issue of non-intersecting bounding boxes in terms of  $IoU$ , aids in accelerating the model's convergence.

While YOLO excelled in terms of speed, it encountered challenges related to localization accuracy. This trade-off spurred further research efforts to fine-tune the model. To redress this trade-off and enhance the localization accuracy, Liu et al. introduced the Single Shot MultiBox Detector (SSD) in 2016 [22]. The SSD method was different from the one-stage paradigm because it used both multi-reference and multi-resolution detection strategies. This made it possible to find objects at different sizes across different network layers. This architecture can accommodate objects of diverse sizes and magnitudes within the image, mitigating the aforementioned accuracy compromise.

In 2018, Lin et al. presented RetinaNet [23], marking a significant advancement in one-stage object detection. The key innovation within RetinaNet was the introduction of a novel loss function termed "focal loss." This loss function, which differs from the cross-entropy loss, was created to give more attention to instances that kept getting incorrectly labeled during the training process. This heightened attention to challenging examples during training resulted in an enhanced level of prediction accuracy, outstripping the performance of its one-stage counterparts.

### 2.3. Anchor-free Inference

In contemporary developments within the domain of object detection, there is a noteworthy shift towards anchor-free methodologies [24]. These novel approaches, in contrast to conventional techniques, emphasize the inference of bounding box corners, rather than reliance on pre-defined bounding boxes. A prominent exemplar of this trend is the CenterNet, an innovative framework introduced by Zhou et al. [25]. Notably, CenterNet has distinguished itself as a state-of-the-art solution for 3D Lidar-based detection and tracking, showcasing its versatility in diverse applications.

CenterNet can be perceived as an evolution of the CornerNet, another anchor-free approach to bounding box detection that represents objects as pairs of keypoints, specifically the top-left and bottom-right corners. These corner keypoints are extracted through a technique known as corner pooling, which was introduced by the same authors [26]. A critical stride in the advancement from CornerNet to CenterNet was the introduction of a central keypoint, a concept that facilitated the association of corner keypoints with objects depicted in images. This novel approach has demonstrated superior performance compared to conventional anchor-based solutions, such as Faster RCNN and YOLO, marking a significant advancement in object detection.

Continuing the trajectory of innovation, in 2020, Perez-Rua and colleagues introduced the Open-ended CenterNet (ONCE) [27]. ONCE improved CenterNet’s abilities by letting it find objects from classes where there were not many examples in its training dataset. This is an impressive achievement that could be useful in situations involving many types of objects.

Additionally, object detection techniques have begun to explore the capabilities of transformers, as paved by the DETection TRansformer (DETR) method introduced by Carion et al. [28]. This exploration leverages the advantages of transformer architectures, which have gained prominence in natural language processing, and integrates them into the object detection domain. What sets DETR apart is its simplicity, coupled with performance that rivals other sophisticated detection techniques employed in the field. Subsequently, Zhu et al.[29] proposed the Deformable DETR system. This system builds on the specific objective detecting small objects. This enhancement aimed to achieve state-of-the-art performance, underscoring the commitment of the scientific community to continuously refine and advance object detection methodologies to meet the evolving demands of real-world applications.

#### 2.4. 3D Object Detection

The field of 3D object detection in autonomous vehicles has witnessed significant advancements, driven by the need for accurate environmental perception to ensure safe navigation. Traditional single-modal detection methods, which typically utilise either LiDAR or camera data, as stated in [30], have limitations. Camera-based systems often lack sufficient depth information, leading to challenges in accurately detecting objects in three-dimensional space, particularly in complex environments. Conversely, LiDAR-based methods, while providing precise spatial data, are hindered by issues such as point cloud sparsity and low resolution, especially in occluded or distant scenarios. Authors in [30] have increasingly focused on multi-modal 3D object detection, which combines the strengths of various sensors to enhance detection performance. By fusing depth information from LiDAR with the rich texture and color data from cameras, multi-modal approaches can significantly improve the accuracy and reliability of object detection systems. However, the integration of heterogeneous data presents unique challenges, including the need for effective data representation, alignment, and fusion techniques. Researchers have proposed various methodologies to address these challenges, moving beyond traditional fusion strategies to more sophisticated frameworks that leverage the complementary characteristics of different modalities.

On the other hand, single-modality 3D object detectors offer several advantages over multi-modal approaches, particularly in terms of simplicity, cost, and computational efficiency. According to [31], single-modality detectors eliminate the need for complex fusion algorithms that are necessary when combining data from multiple sensors like LiDAR and cameras. This leads to reduced computational overhead and easier implementation, making single-modality systems more efficient and faster, especially for real-time applications like autonomous driving and robotics. Additionally, these detectors are cost-effective, as they only require one type of

sensor, significantly reducing hardware and maintenance costs. Single-modality detectors can also focus on leveraging the strengths of a specific sensor, such as the precision of LiDAR in depth estimation or the rich visual detail provided by cameras. While multi-modal systems can offer increased robustness, the added complexity often does not justify the performance gains in environments where a single modality is sufficient for high accuracy.

The development of comprehensive datasets, such as KITTI [32], nuScenes [33], and Waymo [34], has been crucial in facilitating research in the area of 3D object detection, providing benchmarks for evaluating the performance of single- and multi-modal detection algorithms under diverse driving conditions. In this context, the analysis of synthetic virtual environments emerges as a promising avenue for further enhancing 3D object detection. By simulating various driving scenarios and conditions, synthetic environments can generate diverse training data that improves model robustness and adaptability. This approach addresses the limitations of existing detection methods, essentially contributing to the development of safer and more efficient autonomous driving systems. The integration of synthetic virtual environment analysis into the 3D object detection pipeline represents a significant step forward, offering new insights and methodologies that can enhance the overall performance of autonomous vehicles in real-world applications.

The advancement of autonomous vehicles is significantly dependent on the efficacy of their perception systems, particularly in the realm of 3D object detection. Authors in [35] review recent findings on 3D object detection methodologies, with a specific focus on the integration of synthetic virtual environments to enhance detection capabilities. 3D object detection is a critical component that enables vehicles to accurately perceive their surroundings, facilitating informed decision-making and trajectory planning. Recent studies categorize detection methods into three primary types: image-based, point cloud-based, and multi-modal approaches. Multi-modal techniques, which integrate data from various sensors such as LiDAR and cameras, have demonstrated superior robustness against environmental variations, as highlighted by [35]. The use of synthetic virtual environments has emerged as a promising strategy to augment training datasets for these detection algorithms. By simulating diverse scenarios that are often challenging to capture in real-world settings—such as varying weather conditions, lighting scenarios, and complex object interactions—researchers can create comprehensive datasets that significantly improve the robustness of detection systems. For instance, [36] emphasize the development of efficient real-time 3D object detection frameworks that leverage synthetic data for training, leading to enhanced accuracy and reduced costs associated with real-world data collection. However, challenges persist, particularly concerning the domain gap between synthetic and real-world data, which can result in performance degradation when models are deployed outside their training conditions. Future research should prioritize bridging this gap through techniques such as domain adaptation and transfer learning, ensuring that models trained in virtual environments can generalize effectively to real-world scenarios. Overall, the integration of synthetic virtual environments in the development of 3D object detection systems presents a valuable opportunity to enhance the capabilities of autonomous vehicles, paving the way for more robust detection algorithms that are better equipped to handle the complexities of real-world driving conditions. Continued exploration in this area will be essential for advancing the state of autonomous driving technology.

### 3. Methodology

This section delves into the methodology, presented in this work, for detecting 3D objects utilizing ML algorithms and the CenterNet architecture. The methodology employs offsets to

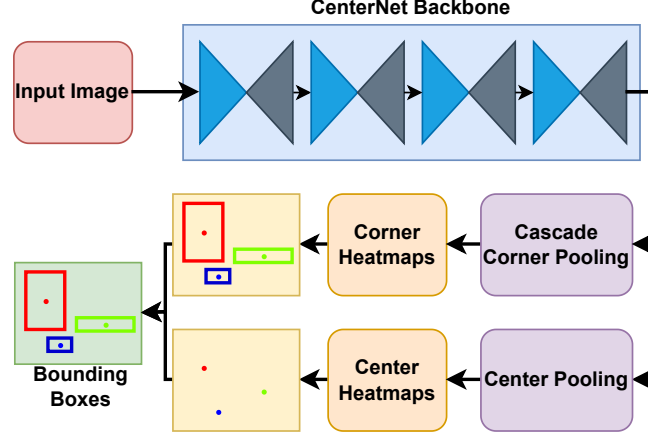


Figure 3: CenterNet architecture.

refine corner positions and heatmaps for predicted key-pairs and corners, in relation to their corresponding object categories. An anchor-free module produces heatmaps, where embedding methods establish significant connections between anticipated object classes and predicted corner values. The methodology further investigates the 3D bounding box regression process, which is leveraged to approximate attributes such as i) depth, ii) dimensions, and iii) orientation. The employed ML methods and algorithms enhance these to enable the thorough and precise detection of 3D objects in the respective scene. The following methodology plays a critical role in attaining accurate and comprehensive object detection.

### 3.1. Methodology Overview

The proposed architecture incorporates a multitude of smaller components. The first component predicts a 2D bounding box common to the standard object detection task. It achieves it by generating and processing Heatmaps, Embeddings, and Offsets [26]. The three outputs are then further processed in Cascade Corner Pooling and Centre Pooling [37] components which infer the final positions of the 2D bounding box. The next component then estimates depth using a Multi-Scale Deep Network [38], in two stages, the first stage collects global information, and, the second stage refines the global information to produce a more precise prediction. The next component estimates the 3D dimensions and orientation of the object using CNNs. The 3D dimensions are regressed directly from feature map outputs using fully-connected layers, however, the orientation is formulated as a classification task and regressed accordingly using the hybrid discrete-continuous MultiBin loss [39]. The final 3D bounding box is constrained by the predicted 2D bounding box with an assumption that the 2D bounding box has been trained to match the position of the 3D bounding box.

### 3.2. Cascade Corner & Center Pooling Spatial Localisation

The methodology presented in this study, depicted in Figure 3, is fundamentally grounded in an anchor-free approach. Within this framework, keypoint descriptors play a fundamental role in representing crucial elements of object detection, encompassing essential entities such as the top-left, bottom-right, and center points. This method places a strong emphasis on the

precise determination of both corner and center keypoints, a task of utmost significance, as it significantly influences the accuracy and reliability of object localization and recognition.

The basis of this approach lies in the meticulous discernment and characterization of keypoint descriptors and which serve as distinctive markers of object structure and spatial attributes. These descriptors capture not only the spatial coordinates but also the semantic information associated with the objects in question. This careful attention to detail contributes to the heightened accuracy and reliability of tasks related to object localization and recognition. These capabilities are of particular importance in various domains including autonomous systems, surveillance, computer vision research, and where the ability to accurately and consistently identify and locate objects is a central requirement.

To confirm the corner keypoints of the bounding box, the methodology includes an AI model prepared with a Cascade Corner Pooling module. This module is chargeable for calculating the most summed response alongside the limits of the function map. Moreover, it extends its scope of analysis to encompass the internal instructions inside the function map. This approach showcases the stableness and robustness of the method, mainly in the face of function-stage noises and variations. The Cascade Corner Pooling module represents a complicated architectural innovation, optimized to provide a comprehensive and noise-tolerant mechanism for an appropriate localization of corner keypoints.

Subsequently, the methodology employs a dedicated AI model to infer the center keypoints. This task is facilitated by the addition of a Center Pooling module. This module calculates the maximum summed response along both horizontal and vertical directions within the feature maps. This bidirectional analysis is instrumental in pinpointing the central keypoints of objects, a task that is important for more accurate object localization and recognition.

The idea behind this methodology is based in the work of Duan et al. [37] that defines the utilization of keypoint descriptors, coupled with data-driven methodologies for determining corner and center keypoints. This method is characterized by its robustness, accuracy, and adaptability, making it well-suited for a wide variety of applications, including but not limited to autonomous vehicle systems, robotics, and computer vision research.

### 3.3. 3D Bounding Box Inference

The determination of keypoints within this framework is achieved through a systematic process that entails the utilization of Heatmaps derived from a set of feature maps generated by the Cascade Corner Pooling and Center Pooling modules. These Heatmaps serve as a representation of the approximate positions of keypoint entities, which are classified into three distinct categories: top-left, bottom-right, and center points. Notably, the Heatmaps are configured with a dimensionality of  $C$  channels, where  $C$  corresponds to the number of object classes under consideration. Additionally, they possess dimensions mirroring those of the input image, denoted as  $H \times W$ , with  $H$  denoting the image's height and  $W$  representing its width.

One of the main keypoints in this methodology is the generation of Associative Embeddings, which serve as a means to establish a link between individual keypoints and their respective object classes. These Embeddings  $E = e_1, e_2, \dots, e_n$  are computed by the AI model and disregard the need for a "ground-truth" label. Instead, their significance lies in the relative differences, which facilitate the grouping of object detections based on their associated Embeddings. Each detected embedding  $e_i$  generated by the AI network is accompanied by a numerical value, denoted as a "tag"  $t_i$ , which plays an instrumental role in the subsequent grouping of detections. The premise is that detections with similar tags should be effectively clustered together in a

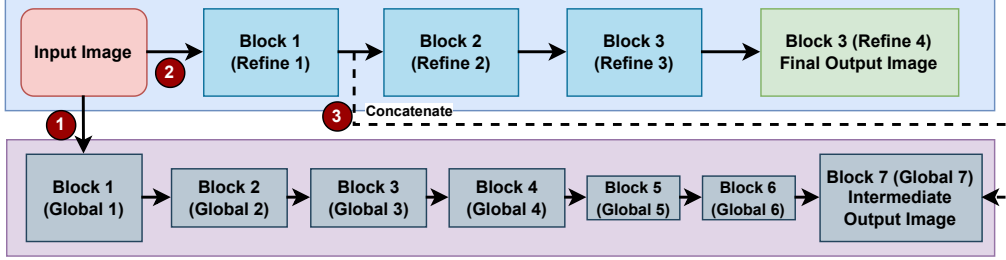


Figure 4: Abstract representation of the Multi-Scale Deep Network

respective tag cluster  $C_i$ . To break this down, let's assume a set of embedding detection  $D$  incorporating detections  $d_1, d_2, \dots, d_n$  where  $D = d_1, d_2, \dots, d_n$ , produced by the model. Each detection  $d_i$  is associated with an embedding  $e_i$  and a numerical tag  $t_i$ , producing an Associated Embedding  $E(di) \rightarrow (e_i, t_i)$ , where  $E$  is the embedding function. The produced tags are clustered assuming a clustering function  $S(t_i, t_j)$ , highlighting if a set of detections are similar when  $S$  exceeds a threshold  $\theta$ . This relationship can be summarised as shown in Eq. 5.

$$\begin{aligned} &\forall d_i, d_j \in D, \\ &\text{if } S(t_i, t_j) > \theta, \\ &\text{then } (d_i, d_j) \in C_l \subseteq C(D) \end{aligned} \quad (5)$$

The output of the model necessitates certain adjustments to optimize the fit of the predicted object to the actual object within the image. In response to this requirement, Offsets are introduced to facilitate these adjustments. Let's assume the set of predicted keypoint locations in the featuremap  $P_k = (x_p, y_p)$  and a set of offset vectors  $O = (o_x, o_y)$  with  $o_x$  and  $o_y$  each offset vector corresponding to a unique predicted keypoint  $(x_p, y_p)$  in  $P_k$ . Each Offset map serves as a spatial mapping of keypoint locations within the feature map space to their corresponding positions on the input image, conveyed in pixel coordinates. This process aids in predicting a set of actual keypoint locations  $A_k = (x_a, y_a)$  in the resulting image following a correlation of  $A_{ki} = P_{ki} + O_i$  for each  $i$ -th keypoint adjusted by its corresponding offset to align with the actual keypoint location in the image. The application of Offsets represents an essential mechanism for fine-tuning the localization of keypoints and enhancing the overall precision of object detection within the framework.

To achieve the successful detection of 3D objects, the methodology entails the prediction of center keypoints, incorporating additional information encompassing depth, 3D dimensions, and orientation, as visually depicted in Figure 5. The depth component is a transformed output, derived from Eigen et al.'s approach [38], and it is presented as an additional scalar value associated with each center keypoint.

The depth prediction mechanism comprises two principal modules, as illustrated in Figure 4. The initial component, the Global Coarse-Scale Network [38], takes the input image and endeavors to predict the depth of the entire scene at a global scale. This global-level prediction serves as the foundation for subsequent refinements. The refinement process is carried out by the Local Fine-Scale Network [38], which receives the output from the Global Coarse-Scale Network and fine-tunes the initial coarse predictions. This fine-tuning process is vital for aligning the depth predictions with local details, including the edges of objects and walls.





Figure 5: The network output for 3D object detection. From left to right: 3D dimensions (metres), depth (metres), orientation (degrees).

In assessing the quality of the depth predictions, the methodology employs a Scale Invariant Error metric (SIE), which can be seen in Eq. 6.

$$SIE = \frac{1}{n} \sum_{i=1}^n \left( \log d_i - \log \hat{d}_i - \frac{1}{n} \sum_{j=1}^n (\log d_j - \log \hat{d}_j) \right)^2 \quad (6)$$

Where  $n$  is the number of data points,  $d_i$  is the true value for the  $i$ -th data point and  $\hat{d}_i$  is the predicted value for the  $i$ -th data point. This metric computes the per-pixel differences between the predicted depth map and the ground truth. Notably, the calculations are performed in a logarithmic space, a choice made to address the potential issue of the average scale of the scene influencing the error measurements. This approach ensures a more robust and scale-invariant assessment of the quality of the depth predictions, ultimately contributing to the accuracy of the 3D object detection process.

The representation of a 3D bounding box in this context relies on three primary parameters, specifically the bounding box center, dimensions, and orientation. The center of the 3D bounding box is characterized by a set of three 3D coordinates, denoted as  $x$ ,  $y$ , and  $z$  (where the center  $B_c = (x, y, z)$ ). The dimensions of the 3D bounding box are governed by an additional triad of attributes, namely width  $w$ , height  $h$ , and length  $l$ , measured in meters, represented as  $B_d = (w, h, l)$ . These dimensions are directly regressed to their respective attributes through the application of a straightforward loss function measuring the distance of the regressed samples, like Mean Square Error, for the three-dimensional aspect of the samples, Eq. 7. The orientation of the 3D bounding box is defined by another set of three attributes, encompassing azimuth, elevation, and roll angles in degrees  $B_o = (\theta_{\text{azimuth}}, \theta_{\text{elevation}}, \theta_{\text{roll}})$ .

$$L_{MSE} = \frac{1}{n} \sum_{i=1}^n [(w_i - \hat{w}_i)^2 + (h_i - \hat{h}_i)^2 + (l_i - \hat{l}_i)^2] \quad (7)$$

The derivation of the three 3D coordinates is realized through the utilization of the MultiBin architecture [39]. In this architecture, each angle is treated as a distinct class, effectively framing the orientation prediction as a classification task. To account for the angular relationships between classes, the MultiBin architecture incorporates the computation of small offsets using trigonometric functions, specifically sine and cosine, applied to the angles. The outcome of this module encompasses three values for each class: the confidence associated with the class, the cosine difference of the angle, and the sine difference of the angle. Specifically, the angles are divided into  $N$  bins, each associated with a class  $C$  and bin  $i$ , calculating a confidence score

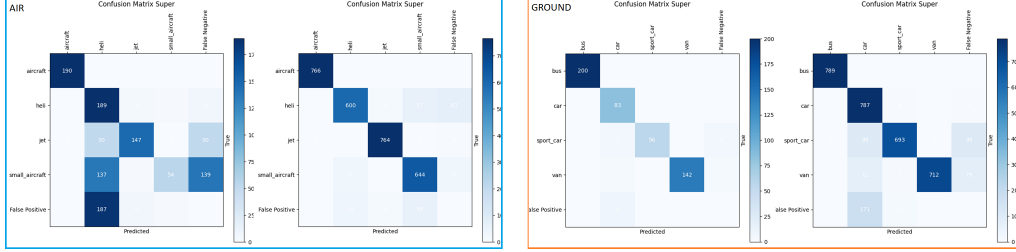


Figure 6: Two comparisons between initial fine-tuning and extensive fine-tuning of Air and Ground categories. From left to right: the blue box is grouping Air category (initial and extensive correspondingly), and the orange box groups Ground category (initial and extensive correspondingly).

$\text{Conf}_{C,i}$ , and trigonometric offsets,  $\cos(\Delta\theta_{C,i})$  and  $\sin(\Delta\theta_{C,i})$ , where  $\Delta\theta_{C,i} = \theta - \theta_{\text{center},i}$ . The final angle  $\hat{\theta}_C$  is estimated by combining these values, weighted by confidence scores, offering precise angular orientation predictions.

For the successful projection of a 3D bounding box onto a 2D image, the calculations necessitate the availability of a camera intrinsic matrix. This matrix plays an important role in ensuring the accurate alignment of the 3D bounding box with the 2D image. Furthermore, to enhance the precision and reliability of the 3D bounding box, it is constrained through the utilization of the 2D bounding box. This constraint mechanism contributes to the refinement of the 3D bounding box and enhances the accuracy of the overall object detection process.

### 3.4. Parameters, Model Complexity, Computational Burden, and Hardware Specifications

In this section, we assess the computational complexity of the proposed methodology by examining the total number of parameters employed in the model architecture. By quantifying the number of parameters, we can gain insight into the model’s resource requirements, such as memory usage and processing power. This evaluation is crucial because it provides a direct measure of the model’s scalability and efficiency, especially when deployed in real-world scenarios. A model with a high number of parameters may achieve better performance, but it also imposes greater computational demands, which can affect training time, inference speed, and hardware requirements. Therefore, understanding the parameter count is essential for balancing accuracy and computational feasibility, particularly in applications with limited resources or real-time constraints.

The proposed methodology leverages a series of experiments, see table 1, to comprehensively assess its performance. In the initial experiment, the model was subjected to 10 epochs of training with a batch size of 3, utilizing synthetic data that spanned four distinct categories. During this training phase, the learning rate was set to 0.0001, and the optimization process was facilitated by the Adam optimizer. Importantly, data augmentation techniques were not employed during this phase.

After the initial experiments detailed above, additional experiments were carried out to probe the impact of extended training on the performance of the models. These final experiments didn’t involve the utilization of fine-tuned models that had undergone 10 epochs of training but were done from the beginning. The choice of batch size for these experiments varied from 3 to 6, contingent on the memory limitations of the GPU.

Furthermore, to expand the scope and depth of the analysis, extended experiments were conducted. These experiments employed the same fine-tuned models, which had undergone 100 epochs of training, and maintained similar batch size ranges as the final experiments. This extended training duration aimed to provide a more comprehensive assessment of the models' performance, encompassing a broader range of training scenarios and conditions.

The proposed methodology incorporates several components, each of which contributes to the total parameter count, ultimately influencing the computational demands during both training and inference. Each component, whether it involves layers, filters, or other structural elements, adds to the overall complexity of the model, impacting how much memory and processing power are required to execute it efficiently. This cumulative parameter count is not only a factor in determining the time and resources needed to train the model but also plays a significant role in how quickly and efficiently it can make predictions during inference. As the number of components increases, the system's computational load rises, which can affect its suitability for large-scale or real-time applications. Therefore, the integration of these components must be carefully balanced to achieve optimal performance without overwhelming available computational resources.

- **Backbone Network:** The backbone of the architecture is based on DLA-34, which is responsible for generating feature maps from the input images. DLA-34, as used in the proposed framework, consists of approximately 15 million parameters. This network forms the foundation of the heatmap and embedding generation process used for object detection tasks [26].
- **Depth Estimation Module:** The depth estimation module employs a Multi-Scale Deep Network based on the work of [38]. This module contains around 30 million parameters, divided between its global coarse-scale network and the local fine-scale network, both of which are responsible for producing accurate depth predictions.
- **3D Bounding Box Regression:** The 3D bounding box regression module is responsible for predicting the dimensions and orientation of the detected objects. This component contains an additional 5 million parameters in the fully connected layers, while the orientation estimation module, which uses a MultiBin loss for angular classification and regression, adds 2 million parameters [39].

In total, the model contains approximately 52 million parameters. This high parameter count reflects the complexity of the architecture, comparable to other state-of-the-art models such as YOLO and Faster R-CNN [7]. While the computational burden is substantial, the use of techniques such as batch normalization and early stopping during training helps mitigate this issue, ensuring that the model can achieve real-time performance when deployed on modern GPUs. Moreover, given the model's ability to handle complex 3D object detection tasks across varying environmental conditions, the computational cost is justified by the improvements in accuracy and robustness, particularly in autonomous driving and augmented reality applications.

The experiments were conducted using two distinct hardware setups to evaluate the performance and generalisability of the proposed method. The primary experiments were performed on a desktop computer equipped with a desktop-grade NVIDIA GPU and an Intel Xeon CPU, providing a robust and high-performance environment for testing. Detailed specifications of this setup are provided in table 2 for reference. To assess the method's versatility and performance under different conditions, the experiments were also repeated on a portable computer featuring a laptop-grade NVIDIA GPU and an Intel Core-series CPU, which represents a less powerful

Table 1: Parameter Tuning

Experiment Phase	Epochs	Batch Size	Data Type	Learning Rate	Optimizer	Synth. Data	Notes
Initial	10	3	4 categories	0.0001	Adam	×	-
Additional	10	3-6	4 categories	0.0001	Adam	✓	GPU memory limitations
Extended	100	3-6	8 categories	0.0001	Adam	✓	Fine-tuning, performance assessment

Table 2: Hardware Specifications

Name	CPU	RAM	GPU
Desktop Computer	Xeon Gold 5122@3.60GHz	64GB	NVIDIA Titan Xp 12GB
Laptop Computer	i9-9880H@2.30GHz	32GB	NVIDIA 1650 Max-Q 4GB

but more common hardware configuration. The specifications for this portable setup are also documented in table 2, enabling a comprehensive comparison between the two platforms.

#### 4. Evaluation

The experiments aim to evaluate the performance of the proposed object detection methodology under various environmental conditions, leveraging both synthetic and real-world datasets. The main target of these experiments is to assess how well the model adapts to different conditions, including variations in camera angles, lighting, weather, and sensor types. In addition, a comparative study is conducted to benchmark the proposed method against state-of-the-art models like YOLOv3, Faster R-CNN, and RetinaNet. These experiments are essential to understand the robustness and generalization capabilities of the proposed detection model, which is evaluated using synthetic data generated through a 3D rendering engine and real-world data from the KITTI dataset. The KITTI dataset is a well-established benchmark in autonomous driving research, containing high-resolution images and LiDAR data from urban environments, making it highly suitable for evaluating the performance of object detection models. Its rich diversity in classes like cars, pedestrians, and cyclists, across various environments, provides a comprehensive platform for testing the robustness of models in both controlled (synthetic) and real-world settings.

The experimental procedure followed a clear and systematic process. First, the synthetic dataset was created using a 3D rendering engine, with each category (Camera, Light, Weather, and Sensor) comprising specific parameters such as camera angles, lighting conditions, weather settings, and sensor types. The dataset was split into training, validation, and testing subsets, ensuring that the models were evaluated on both known and unseen data. After training, the models were evaluated on both synthetic and real-world data, including the KITTI dataset, which provided a real-world benchmark for performance evaluation. Additionally, experiments were repeated on a more performance limited device detailed in table 2. This step-by-step process ensured that the model’s performance was rigorously tested under controlled and real-world conditions, allowing for a fair comparison across different models.

##### 4.1. Metrics

The evaluation of the proposed method relied on the mean Average Precision (mAP), a standard metric to quantify object detection performance based on a user-defined set of criteria [40]. It is defined as the mean value of the average precision of the individual classes:

$$mAP = \frac{1}{n} \sum_{k=1}^n AP_k \quad (8)$$

where  $AP_k$  is Average Precision of class  $k$ , and  $n$  is the number of classes.

Additionally, confusion matrices were used to analyze the performance of the model across different object classes, offering a detailed view of detection accuracy. This combination of metrics provides a comprehensive evaluation of model performance across varied scenarios, highlighting both successes and challenges.

#### 4.2. Data Specification

The synthetic dataset used in this study was designed to emulate real-world conditions and contains images divided into four main categories: Camera, Light, Weather, and Sensor. Each category is further split into two subcategories, Air and Ground, where Air images contain aerial vehicles, and Ground images contain terrestrial vehicles. Approximately 3000 images per category were generated to comprehensively assess model performance under different environmental factors. The Camera category consists of images captured at various distances, elevation angles, and azimuth angles, while the Light category introduces different lighting conditions, such as variations in intensity and direction. The Weather category incorporates rainy and non-rainy conditions, including wind variations, to assess the model's performance under adverse conditions. Finally, the Sensor category includes images that simulate night and thermal vision to further test the model's robustness. In contrast, the KITTI dataset contains real-world images from urban environments, which also include LiDAR data and other sensor information. It provides a more realistic benchmark for evaluating the model's performance in object detection tasks, especially since it includes diverse objects such as cars, pedestrians, and cyclists in real-world driving scenarios. While the synthetic dataset allows for controlled experimentation, the KITTI dataset serves as an important benchmark for generalization to real-world data.

As it was mentioned above, there are four categories of sub-datasets a) Camera, b) Light, c) Weather, and d) Sensor. The Camera category represents images generated with different camera angles (point of view) and distances from an object in the City and the Desert scenes. Specifically, for the Air sub-category, there are images generated at 4 equal distances between 70 and 350 metres. For the Ground sub-category, there are images generated at 4 equal distances between 15 and 75 metres. In both categories the images were generated at 4 equal elevation angles between  $5^\circ$  and  $85^\circ$  degrees, and at 3 equal azimuth angles between  $0^\circ$  and  $240^\circ$  degrees. The other parameters such as light, image type, fog and rain were selected in such a way to prevent generating bias on the evaluation of the camera parameters. An overview of the synthetic data generation specification is presented in Table 3.

The Light category contains images generated using variable balanced lighting parameters covering the City and Desert scenes. In more detail, the Air and Ground sub-categories were generated with the light intensity set between 10% and 100% power at 3 equal steps. The light elevation angles were set between  $5^\circ$  and  $90^\circ$  degrees at 3 equal steps, the light azimuth angles were set between  $0^\circ$  and  $180^\circ$  degrees at 3 equal steps. The other parameters related to camera, weather, and sensors were selected randomly and uniformly in such a way to avoid bias on the evaluation of the model under the set of the light parameters.

The Weather category contains images generated using different balanced weather parameters covering the City and Desert scenes. The Air and Ground sub-categories were generated both with and without enabling rain. Furthermore, the rainy images included variations due to the

Table 3: Summary of Data Specifications

Data Category	Scene	Sub-Categories	Parameters	Details
Camera	City & Desert	Air, Ground	Distance, Elevation Angle, Azimuth Angle	Distances: 70-350m (Air), 15-75m (Ground); Elevation Angles: 5°-85°; Azimuth Angles: 0°-240°
Light	City & Desert	Air, Ground	Light Intensity, Elevation Angle, Azimuth Angle	Intensity: 10-100% (3 steps); Elevation Angles: 5°-90°; Azimuth Angles: 0°-180°
Weather	City & Desert	Air, Ground	Rain, Wind	Rain: Enabled/Disabled; Wind: 0 or 10 units
Sensor	City & Desert	Air, Ground	Night Vision, Thermal Vision	Night and Thermal Vision emulations
Real (KITTI)	Varied	Air, Ground	High-resolution Images, Lidar, Calibration	Object Detection, Tracking, 3D Scene Understanding

wind parameter that was selected to be 0 or 10 units of power. The other parameters were selected in such a way to avoid bias on the evaluation of the models in the weather category.

The night and thermal vision are the main attributes of the Sensor category. The night vision visualises an approximation of the effect of night vision goggles and the same approach was considered for the thermal vision. The Sensor category contains images generated using different balanced sensor image types covering the City and Desert scenes. Also, the Air and Ground sub-categories contain images emulating both night and thermal vision sensors. The other parameters again were selected uniformly preventing bias on the evaluation of the models for the sensor set of parameters.

Synthetic datasets can effectively mimic the generalisation capabilities of real-world data by leveraging digital twins and synthetic data generation tools built on game engines. Digital twins create virtual replicas of real-world environments, capturing detailed spatial, temporal, and func-



Figure 7: Real Dataset 3D Bounding Box Sample

tional characteristics that enable realistic simulation of real-world scenarios. When combined with the advanced rendering, physics, and animation capabilities of game engines, these tools can generate highly realistic and diverse datasets that mirror the complexity of actual environments. Such synthetic data can replicate intricate interactions, simulate varying conditions, and introduce controlled variations. This approach is particularly useful for creating scalable and cost-effective datasets while addressing limitations like bias or scarcity in real-world data collection and ensuring that the synthetic datasets encapsulate the variability and complexity of real-world data. Consequently, models trained on such datasets can generalise effectively, as they are exposed to a wide range of representative patterns and scenarios that mirror real-world applicability.

#### 4.3. Real Dataset Experiments

The KITTI dataset [32] which is a widely used benchmark dataset for research in computer vision and autonomous driving [41] was chosen as the Real dataset. It stands for "Karlsruhe Institute of Technology and Toyota Technological Institute" and was created by researchers from these institutions. This dataset is commonly referenced in academic publications related to tasks such as object detection, tracking, 3D scene understanding, and more. The specification of the dataset can be seen in Table 4.

The primary objective behind its inception is to foster the advancement of algorithms and technologies relevant to autonomous vehicles. The dataset is characterized by a comprehensive collection of diverse data modalities, encompassing high-resolution camera images, LiDAR point clouds, and calibration parameters. This dataset is used for a wide variety of tasks, including object detection, motion tracking, 3D scene analysis, and other such applications. An added feature of the KITTI dataset is the provision of image annotations for various object types, such as cars, pedestrians, and cyclists, thereby rendering it an important resource for the validation of AI models. Furthermore, the dataset encompasses a wide spectrum of real-world driving scenarios, variable weather conditions, and different times of day, thereby facilitating a comprehensive assessment of algorithm performance under diverse environmental conditions. It is worth noting that while the KITTI dataset is widely used in the research community, it does exhibit certain

Table 4: Features and Specifications of the 3D Object Detection KITTI Dataset

Feature/Specification	Description
<b>Data Type</b>	Images, Lidar data
<b>Tasks</b>	Stereo, Optical Flow, Visual Odometry, 3D Object Detection, Tracking
<b>Number of Images</b>	~15,000 images for object detection
<b>Image Resolution</b>	1242 x 375 pixels
<b>Sensors</b>	Inertial Navigation System (GPS/IMU): OXTS RT 3003, Laserscanner: Velodyne HDL-64E, Grayscale cameras, 1.4 Megapixels: Point Grey Flea 2 (FL2-14S3M-C), Color cameras, 1.4 Megapixels: Point Grey Flea 2 (FL2-14S3C-C), Varifocal lenses, 4-8 mm: Edmund Optics NT59-917
<b>Annotation Types</b>	Bounding boxes, 3D boxes, object type, truncation, occlusion levels
<b>Environments</b>	Urban, residential, road
<b>Classes</b>	Cars, vans, trucks, pedestrians, cyclists
<b>Ground Truth Availability</b>	Yes

Table 5: Comparison of Performance Results on the Real Datasets (% mAP)

Class	FRRCNN	RETINA	YOLOv3	PM
Car	64.67	77.09	69.01	<b>87.85</b>
Pedestrian	28.42	51.78	39.17	<b>60.85</b>
Cyclist	32.33	51.32	43.84	<b>48.69</b>
<b>Total</b>				
mAP	41.81	60.06	51.34	<b>65.80</b>

limitations, notably its relatively modest scale and the absence of data pertaining to certain object classes, e.g., motorcycles.

The performance of the proposed framework on the Real dataset could be observed in table 5. The results of the table comparing performance on real datasets reveal several key insights regarding the effectiveness of different models—FRRCNN, RETINA, YOLOv3, and the proposed PM model—across three object detection categories: Car, Pedestrian, and Cyclist. The proposed PM model demonstrates superior performance across all three categories, achieving the highest accuracy in detecting Cars (87.85%), Pedestrians (60.85%), and Cyclists (48.69%). This suggests that the PM model is particularly well-suited for detecting objects in real-world conditions, outperforming other models in each individual class. The strong performance in the Car category is especially notable, where PM significantly outperforms the other models, with a performance margin of over 10% compared to the second-best RETINA model (77.09%). This indicates that PM is highly capable of recognizing cars, likely due to better feature extraction or training strategies suited to this object class.



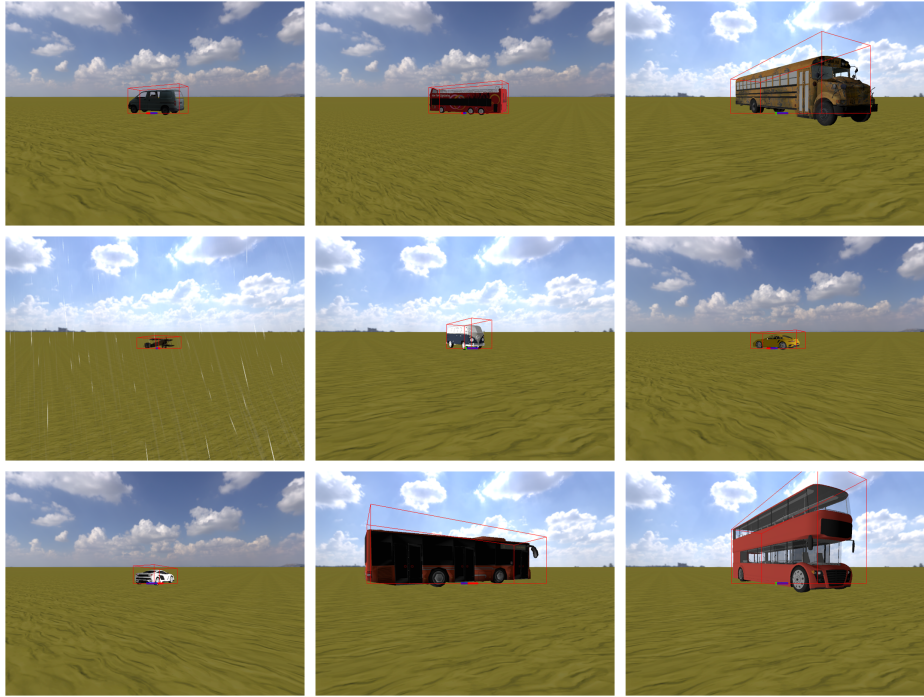


Figure 8: Example of 3D bounding box predictions. (Synthetic Data)

Table 6: Performance results on the Synthetic dataset (mAP)

Category	Sub-category	PM	FRRCNN	YOLOv3	RETINA
Air	Camera	<b>61.04%</b>	5.24%	44.82%	44.79%
	Light	<b>39.95%</b>	20.66%	63.58%	61.25%
	Weather	<b>88.71%</b>	5.35%	39.00%	45.57%
	Sensor	<b>51.90%</b>	4.97%	4.27%	7.95%
Ground	Camera	<b>74.66%</b>	17.95%	76.02%	88.81%
	Light	<b>33.82%</b>	32.54%	38.52%	87.12%
	Weather	<b>58.75%</b>	17.07%	66.32%	86.09%
	Sensor	<b>55.72%</b>	4.16%	7.64%	15.59%

In the Pedestrian class, PM again outperforms the other models, although the performance gap is narrower compared to the Car category. RETINA, which shows a strong second-place performance (51.78%), trails PM by approximately 9%. Pedestrian detection typically involves more variability in size and occlusion, making it a challenging category. PM’s performance indicates its robustness in handling this complexity, although there may still be room for improvement to reach higher accuracy.

For Cyclists, the PM model achieves the highest accuracy (48.69%), though the performance gap here is smaller compared to other classes. RETINA and YOLOv3 perform similarly, with results of 51.32% and 43.84%, respectively, while FRRCNN shows weaker performance (32.33%). This suggests that while PM is the most effective model overall, detecting cyclists remains more challenging due to the variability in appearance and size, indicating that further fine-tuning or data augmentation might be required to improve accuracy in this category.

When looking at the total mean Average Precision (mAP) across all categories, the PM model achieves the highest overall score (65.80%), outperforming RETINA (60.06%), YOLOv3 (51.34%), and FRRCNN (41.81%). The mAP metric highlights the superior generalization and robustness of the PM model across all object detection tasks. The improvement in mAP by nearly 6% over RETINA further underscores the advantage of PM in handling real-world datasets.

In summary, the PM model stands out in terms of performance, particularly in detecting cars, where it exhibits substantial accuracy gains. Its ability to consistently outperform other models across all categories, coupled with the highest total mAP, suggests that the PM model is more adaptable and effective in diverse object detection tasks. Nevertheless, some categories, such as cyclists, remain more challenging, and additional efforts to further enhance detection accuracy, particularly through data diversity or model fine-tuning, could help close performance gaps. These results suggest the PM model is highly promising for real-world applications but may benefit from further optimization for more challenging object classes.

#### 4.4. Synthetic Dataset Experiments

The proposed CenterNet model demonstrated strong and competitive performance across various categories, as evidenced by the results on the Synthetic dataset (Table 6). The model’s confusion matrices can be observed in Figures 3 and 6, illustrating the distinctions between performance on Air and Ground datasets. Several notable phenomena were observed during the experiments. The model’s detection accuracy for ground vehicles was significantly higher than

for aerial vehicles. This discrepancy is likely due to the more consistent size, shape, and proximity of ground vehicles, whereas aerial vehicles exhibit greater variability. In the Camera and Light subcategories, the model performed particularly well, benefitting from the predictable nature of lighting and camera angles. In contrast, the Weather and Sensor subcategories posed considerable challenges. Rain, wind, and other environmental factors in the Weather subcategory reduced detection accuracy, while the Sensor subcategory, which included night vision and thermal vision images, proved to be the most difficult for the model. The complexity of these specialized data types likely requires more focused training and fine-tuning.

The objective of this experiment was to evaluate the performance of the proposed CenterNet model for object detection in synthetic environments. The goal was to analyze the model’s ability to detect objects across different categories, particularly “Air” and “Ground,” as well as in subcategories like “Camera,” “Light,” “Weather,” and “Sensor.” The evaluation aimed to assess the effects of diverse environmental conditions on the model’s performance and determine where improvements could be made, especially in challenging scenarios like night vision and thermal imaging. The model excelled in several subcategories, particularly in the Camera and Light subcategories, where the predictability of features such as lighting conditions led to strong detection results. As seen in Table 6, the Light subcategory was among the easiest to detect, given its uniformity in visual parameters like object angles and lighting. On the other hand, more complex conditions, such as those found in the Weather and Sensor subcategories, introduced challenges that affected performance. The Weather subcategory achieved mid-level results due to factors like rain and wind, which added complexity to image detection. The Sensor subcategory, comprising specialized data types like night and thermal vision, proved the most difficult for the model, reflecting the need for further specialized training and fine-tuning to handle these complex image types effectively.

The underlying causes of these results are multifaceted. Ground vehicles tend to have more stable and consistent features, making them easier to detect, while aerial vehicles vary in size and shape, which complicates detection. In the Light subcategory, uniform lighting conditions made object features more predictable, contributing to the model’s strong performance. The drop in performance in the Weather subcategory can be attributed to the increased complexity introduced by environmental conditions like rain and wind. Similarly, the Sensor subcategory, with its night and thermal vision data, differs significantly from normal visual data, making it more challenging for the model to adapt without further specialized training.

To further improve performance, several recommendations can be made. Fine-tuning the model on sensor data, particularly night and thermal vision, could enhance its ability to handle complex image types. Additionally, augmenting the training dataset with real-world samples, such as those from the KITTI dataset, could reduce the gap between synthetic and real-world performance. Domain adaptation techniques could also improve the model’s generalisation from synthetic to real-world conditions, making it more robust for practical applications. Overall, the PM demonstrated strong performance, particularly in the Camera and Light subcategories, where it outperformed other models. However, the results in the Weather and Sensor subcategories suggest that further work is needed to improve the model’s robustness in handling complex environmental conditions like rain, wind, night vision, and thermal vision. With additional fine-tuning, expanded training data, and domain adaptation strategies, the model’s generalization capabilities can be further enhanced, making it more suitable for real-world applications in object detection. This experiment provides a solid foundation for future research aimed at improving object detection in challenging conditions.

Additionally, experiments were conducted on both synthetic and real datasets using a more

hardware-constrained device to evaluate the robustness and adaptability of the proposed method under limited computational resources. The results obtained were identical to those from experiments conducted on higher-performance hardware, demonstrating the method’s consistency across different platforms. However, the restricted GPU memory of the constrained device necessitated adjustments to the number of training epochs, ensuring that the experiments could be executed without exceeding memory limitations. Despite these adjustments, the inference performance remained unaffected, indicating that the method is not resource-dependent in this phase. Nevertheless, the reduced computational capacity led to an increase in the time required to complete each experiment, highlighting the impact of hardware limitations on the training process. However, inference speed stayed the same. This underscores the importance of considering hardware constraints when deploying the method in real-world scenarios.

A more in-depth analysis of the differences between synthetic and real-world datasets reveals several factors that influence their effectiveness and applicability in various domains. While real-world datasets are rich in complexity, capturing the inherent noise, variability, and unpredictability of actual environments, they often suffer from biases, incomplete data, and challenges related to data collection [42], [43], [44]. In contrast, synthetic datasets offer a controlled environment where these biases can be mitigated, and data can be generated to specifically target areas that are underrepresented or difficult to capture in the real world, such as rare events, edge cases, sensory data, weather conditions, and camera angles, etc. However, synthetic datasets may lack the full diversity and nuance found in real-world data, especially when the data generation process cannot perfectly replicate the complex interactions and real-world uncertainties. Despite this, synthetic datasets can be valuable for training models in scenarios where real-world data is scarce, expensive, or ethically challenging to obtain.

## 5. Conclusion

This paper provides a comprehensive overview of existing methodologies and approaches within the realm of scene analysis, leveraged by autonomous vehicles, with a specific emphasis on their applicability in immersive environments. The research presented delves into an in-depth analysis of a 3D object detection model from the vantage point of the augmented reality domain. The architectural framework comprises a diverse set of components, each meticulously designed to tackle various intricacies related to the estimation of keypoints, the conversion of keypoints to 2D bounding boxes, and the inference of crucial spatial information. This information encompasses depth, 3D dimensions measured in meters, as well as orientation, encompassing azimuth, elevation, and roll angles. The collective contributions of these components culminate in a model that exhibits proficiency in the projection of 3D bounding boxes onto a 2D image.

To empirically evaluate the efficacy of the proposed architecture, a comprehensive testing regimen was conducted, utilizing a synthetic dataset in a comparative study. The outcomes of this evaluation reveal that the proposed model delivers competitive performance while demonstrating stability, particularly when tasked with the detection of distant objects. The evaluation and analysis of the proposed model were undertaken under diverse environmental conditions and with varying camera settings, establishing its versatility and robustness. Furthermore, to augment the comprehensiveness of the study, a novel and well-balanced synthetic dataset was created and curated, utilising a virtual environment. This dataset encompasses annotated data spanning a multitude of objects and environmental scenarios, providing a rich resource for subsequent validation, experimentation, and refinement.

## Acknowledgments

This work was funded by UK Research and Innovation (UKRI) under the UK government's Horizon Europe funding guarantee [grant number 10047653] and funded by the European Union [under EC Horizon Europe grant agreement number 101070181 (TALON)].

## References

- [1] M. I. Pavel, S. Y. Tan, A. Abdullah, Vision-based autonomous vehicle systems based on deep learning: A systematic literature review, *Applied Sciences* 12 (14) (2022) 6831.
- [2] J. Xiong, E.-L. Hsiang, Z. He, T. Zhan, S.-T. Wu, Augmented reality and virtual reality displays: emerging technologies and future perspectives, *Light: Science & Applications* 10 (1) (2021) 216.
- [3] Z. Zou, K. Chen, Z. Shi, Y. Guo, J. Ye, Object detection in 20 years: A survey, *Proceedings of the IEEE* (2023).
- [4] R. Anderson, J. Toledo, H. ElAarag, Feasibility study on the utilization of microsoft hololens to increase driving conditions awareness, in: *2019 SoutheastCon*, IEEE, 2019, pp. 1–8.
- [5] D. L. Gomes Jr, A. C. de Paiva, A. C. Silva, G. Braz Jr, J. D. S. de Almeida, A. S. de Araújo, M. Gattas, Augmented visualization using homomorphic filtering and haar-based natural markers for power systems substations, *Computers in Industry* 97 (2018) 67–75.
- [6] N. Dimitropoulos, T. Togias, G. Michalos, S. Makris, Operator support in human–robot collaborative environments using ai enhanced wearable devices, *Procedia Cirp* 97 (2021) 464–469.
- [7] Z.-Q. Zhao, P. Zheng, S.-t. Xu, X. Wu, Object detection with deep learning: A review, *IEEE transactions on neural networks and learning systems* 30 (11) (2019) 3212–3232.
- [8] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, *Proceedings of the IEEE* 86 (11) (1998) 2278–2324.
- [9] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.
- [10] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, Vol. 1, Ieee, 2001, pp. I–I.
- [11] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, *IEEE transactions on pattern analysis and machine intelligence* 37 (9) (2015) 1904–1916.
- [12] R. Girshick, Fast r-cnn, in: *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.
- [13] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, *Advances in neural information processing systems* 28 (2015).
- [14] J. Dai, Y. Li, K. He, J. Sun, R-fcn: Object detection via region-based fully convolutional networks, *Advances in neural information processing systems* 29 (2016).
- [15] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, J. Sun, Light-head r-cnn: In defense of two-stage object detector, *arXiv preprint arXiv:1711.07264* (2017).
- [16] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.
- [17] J. Cao, H. Cholakkal, R. M. Anwer, F. S. Khan, Y. Pang, L. Shao, D2det: Towards high quality object detection and instance segmentation, in: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11485–11494.
- [18] J. Redmon, S. Divvala, R. Girshick, 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.
- [19] J. Redmon, A. Farhadi, Yolo9000: better, faster, stronger, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.
- [20] J. Redmon, A. Farhadi, Yolo3: An incremental improvement, *arXiv preprint arXiv:1804.02767* (2018).
- [21] Z. Wang, L. Wu, T. Li, P. Shi, A smoke detection model based on improved yolov5, *Mathematics* 10 (7) (2022). doi:10.3390/math10071190.  
URL <https://www.mdpi.com/2227-7390/10/7/1190>
- [22] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg, Ssd: Single shot multibox detector, in: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I* 14, Springer, 2016, pp. 21–37.

- [23] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2980–2988.
- [24] X. Wu, D. Ma, X. Qu, X. Jiang, D. Zeng, Depth dynamic center difference convolutions for monocular 3d object detection, *Neurocomputing* 520 (2023) 73–81. doi:<https://doi.org/10.1016/j.neucom.2022.11.032>. URL <https://www.sciencedirect.com/science/article/pii/S092523122201414X>
- [25] X. Zhou, D. Wang, P. Krähenbühl, Objects as points, arXiv preprint arXiv:1904.07850 (2019).
- [26] H. Law, J. Deng, Cornernet: Detecting objects as paired keypoints, in: Proceedings of the European conference on computer vision (ECCV), 2018, pp. 734–750.
- [27] J.-M. Perez-Rua, X. Zhu, T. M. Hospedales, T. Xiang, Incremental few-shot object detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.
- [28] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part 1 16, Springer, 2020, pp. 213–229.
- [29] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, J. Dai, Deformable detr: Deformable transformers for end-to-end object detection (2021). arXiv:2010.04159.
- [30] L. Wang, X. Zhang, Z. Song, J. Bi, G. Zhang, H. Wei, L. Tang, L. Yang, J. Li, C. Jia, et al., Multi-modal 3d object detection in autonomous driving: A survey and taxonomy, *IEEE Transactions on Intelligent Vehicles* 8 (7) (2023) 3781–3798.
- [31] T. Karim, Z. R. Mahayuddin, M. K. Hasan, Singular and multimodal techniques of 3d object detection: Constraints, advancements and research direction, *Applied Sciences* 13 (24) (2023) 13267.
- [32] A. Geiger, P. Lenz, R. Urtasun, Are we ready for autonomous driving? the kitti vision benchmark suite, in: Conference on Computer Vision and Pattern Recognition (CVPR), 2012.
- [33] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, O. Beijbom, nuscenes: A multimodal dataset for autonomous driving, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 11621–11631.
- [34] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, et al., Scalability in perception for autonomous driving: Waymo open dataset, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 2446–2454.
- [35] Z. Song, L. Liu, F. Jia, Y. Luo, C. Jia, G. Zhang, L. Yang, L. Wang, Robustness-aware 3d object detection in autonomous driving: A review and outlook, *IEEE Transactions on Intelligent Transportation Systems* (2024).
- [36] H. Meng, C. Li, G. Chen, Z. Gu, A. Knoll, Er3d: An efficient real-time 3d object detection framework for autonomous driving, in: 29th IEEE International Conference on Parallel and Distributed Systems, 2023.
- [37] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, Q. Tian, Centernet: Keypoint triplets for object detection, in: Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 6569–6578.
- [38] D. Eigen, C. Puhrsch, R. Fergus, Depth map prediction from a single image using a multi-scale deep network, *Advances in neural information processing systems* 27 (2014).
- [39] A. Mousavian, D. Anguelov, J. Flynn, 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.
- [40] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft coco: Common objects in context, in: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13, Springer, 2014, pp. 740–755.
- [41] J. Mao, S. Shi, X. Wang, H. Li, 3d object detection for autonomous driving: A comprehensive survey, *International Journal of Computer Vision* 131 (2023) 1–55. doi:[10.1007/s11263-023-01790-1](https://doi.org/10.1007/s11263-023-01790-1).
- [42] H. Gao, J. Shao, M. Iqbal, Y. Wang, Z. Xiang, Cfpc: The curbed fake point collector to pseudo-lidar-based 3d object detection for autonomous vehicles, *IEEE Transactions on Vehicular Technology* (2024).
- [43] H. Gao, X. Yu, Y. Xu, J. Y. Kim, Y. Wang, Monoli: Precise monocular 3d object detection for next-generation consumer electronics for autonomous electric vehicles, *IEEE Transactions on Consumer Electronics* (2024).
- [44] H. Gao, D. Fang, J. Xiao, W. Hussain, J. Y. Kim, Camrl: A joint method of channel attention and multidimensional regression loss for 3d object detection in automated vehicles, *IEEE Transactions on Intelligent Transportation Systems* 24 (8) (2022) 8831–8845.