Published online: 29 October 2024 DOI: 10.54254/2977-3903/12/2024138

F-PointNet multi-modal 3D object detection based on RGB images and LiDAR point cloud

Yikun Luo

University of Bristol, Beacon House, Queens Road, Bristol, BS8 1QU, United Kingdom

Yikunluo2001@163.com

Abstract. This project explores the integration of image and point cloud data for 3D object detection using the F-PointNet model, aiming to enhance accuracy and reliability in autonomous driving applications. F-PointNet leverages multimodal data from RGB cameras and LiDAR to improve environmental perception and object localisation under varied operational conditions. Employing a rigorous methodology, the model incorporates preprocessing and network components such as frustum rotation and T-net adjustments to refine the detection process. Experiments were conducted on the KITTI dataset, which included applying both random and designated perturbations, and assessing their impact on the model's performance. Results show that random perturbations generally outperform designated ones, especially in complex scenarios, by enhancing the model's adaptability and capability for generalisation. This study highlights the critical role of methodological innovations and data perturbation strategies in advancing 3D object detection technologies, suggesting that further research is needed to optimise these approaches for broader applications. Furthermore, this research contributes to the development of autonomous systems, emphasising the importance of robust and accurate 3D object detection in enhancing the safety and reliability of autonomous vehicles.

Keywords: 3D Object Detection, F-PointNet, Multimodal Data Fusion

1. Introduction

1.1. Background

Increasingly, car companies are adopting autonomous driving technology, with vehicle detection on the road being a critical aspect of this innovation. The autonomous driving system's 'eyes' are responsible for detecting objects, which guarantees the safe operation of the system [1]. There are four main types of 3D vehicle identification algorithms, which are the most important aspect of autonomous driving: (1) mono image-based, (2) stereo image, (3) LiDAR (Light identification and Ranging), and (4) a fusion of the mono image and LiDAR [2]; these four algorithms are also the basis of numerous more models that have been constructed. Despite the work on these algorithms, several issues remain in detecting objects that make it impossible to fully trust autonomous driving systems. For example, issues like imbalances [3] and inaccurately identifying items with similar shapes [4] can negatively affect the model's overall performance.

The SAE (Society of Automotive Engineers) published autonomous driving standards in 2014, classifying driving into six categories, ranging from 0 to 5 [5]. An accurate object detection system is therefore required to reduce traffic accidents since it enables vehicles to perceive road conditions instead of humans [6]. However, conditional automation (level 3) is still the state of even the most sophisticated self-driving automobiles [5]. This implies that drivers of autonomous cars still can't depend entirely on their vehicles.

In this project, the F-PointNet model is employed for multimodal 3D object detection. Initially, 2D object region proposals are generated in the RGB image using a CNN. Each 2D region is then extruded into a 3D viewing frustum, from which a point cloud is derived from depth data. Finally, Frustum PointNet predicts an oriented and amodal 3D bounding box for the object from the points within the frustum.

1.2. Aims and objective

This project focuses on multimodal 3D object detection by integrating data from sensors like RGB cameras and LiDAR. F-PointNet improves accuracy by combining multimodal data, providing richer environmental information. This fusion enhances detection reliability in complex environments and ensures robust performance, even when some sensors deliver suboptimal data, offering greater resilience and adaptability. The project's goal is to achieve multimodal 3D object recognition by RGB image and point cloud data. The specific objectives of the project are as follows:

- **Frustum proposal**. Perform 2D detection on the RGB image to generate a 2D bounding box. The project also generates a viewing frustum based on the 2D bounding box and combines it with the point cloud data to define a 3D cone area.
- **3D instance segmentation**. Using the PointNet model, the point cloud data within the 3D viewing frustum is segmented to extract points associated with identified objects.
- **3D box estimation**. This method estimates the object's amodal-oriented 3D bounding box using a box regression PointNet in conjunction with a preprocessing transformer network.

The remainder of this report comprises an overview of methods related to 3D object detection (Chapter 1.3), including a detailed description of the target method (Chapter 2.1), a review of related work (Chapter 2.2), and results (Chapter 2.3). It concludes with discussions of the findings and suggestions for future work (Chapter 3).

1.3. Literature review

Three categories [1] can be used to categorise the current work on 3D detection: image approaches, point methods, and fusion methods. However, image-based 3D detection is rarely used in practical applications.

1.3.1. Methods based on images

The Multi-level Fusion Based 3D Object Detection from Monocular Images [7] method uses deep CNNs to extract features from single-camera images. It employs a multi-level fusion strategy to progressively refine these features, aligning 2D image data with 3D spatial representations. By estimating depth from monocular images, it effectively detects object location and dimensions from a single viewpoint. This approach is valuable for tasks like autonomous driving or robotic navigation, where accurate depth perception is needed with minimal hardware.

Another innovative method is DETR3D [8], DETR3D uses a transformer architecture to detect 3D objects from multi-camera 2D images. It projects 3D queries onto image planes, focusing on relevant regions to gather 2D features from different viewpoints. The transformer attention mechanism combines this information to accurately predict 3D object positions and dimensions, performing well without depth sensors.

1.3.2. Methods based on point cloud data from LiDAR

The voxel grid-based approach involves processing the point cloud using a 3D convolutional neural network after it has been divided into a voxel grid. For example, VoxelNet [9] converts LiDAR point clouds into voxel grids, processing each voxel with 3D CNNs to extract features. A voxel feature encoding layer then aggregates these features for object detection and 3D bounding box estimation. The huge 3D search space and costly issues with 3D convolution make this method computationally expensive to use [10]. Furthermore, SECOND [7] optimises 3D object detection in point clouds by using sparse 3D convolutions, focusing on key areas in voxel grids to reduce computational load and improve speed. This balance of efficiency and accuracy makes it suitable for real-time tasks like autonomous driving. Unlike voxel-based methods, point-based approaches process raw point cloud data directly, preserving its geometric properties.

PointNet [11] is a groundbreaking model for point cloud processing that processes each point independently through an MLP and uses max pooling to create a global feature representation. However, while it is effective for tasks like point cloud classification and segmentation, it lacks the ability to capture local structures, limiting its precision for complex object recognition and spatial understanding. PointNet++ [12] builds on PointNet by adding a hierarchical structure to better capture local features in point clouds. It uses point set abstraction and feature propagation to group points into local clusters and extract features using MLPs. With a self-attention mechanism, it identifies local structures at multiple scales, improving performance in tasks like hierarchical point cloud classification and semantic segmentation. PointCNN [13] adapts CNNs for point cloud data by using a unique 'X-Conv' operation to organise unordered points in local neighbourhoods, allowing the application of traditional convolution filters. This approach captures local structures and provides rotational invariance, making PointCNN effective for tasks like detailed object classification and precise point cloud segmentation.

RangeDet [14] introduces a LiDAR-based 3D object detection approach that uses the range view - a spherical projection of point clouds - retaining native spatial and depth information better than point cloud or bird's-eye view formats. By applying CNNs

directly to the range view, it captures detailed spatial relationships and depth cues, simplifying preprocessing and improving object detection accuracy, particularly for smaller and distant objects. This makes RangeDet especially effective for autonomous driving. BirdNet [15] is a 3D object detection framework for autonomous driving that transforms LiDAR data into a bird's-eye view (BEV). This simplifies the 3D scene into a 2D plane while retaining spatial and depth information. By applying CNNs to BEV images, BirdNet excels in detecting, classifying, and localising objects, making it effective for navigating complex environments and enhancing autonomous driving safety. Finally, VeloFCN [16] uses fully convolutional networks (FCNs) to detect vehicles in 3D point clouds, particularly from LiDAR systems. Instead of relying on anchor boxes or region proposals, it applies 3D convolutional layers directly to the raw point cloud data, extracting spatial features and enabling end-to-end learning. This approach efficiently segments vehicle points from non-vehicle points, allowing for accurate real-time vehicle detection while reducing computational complexity.

1.3.3. Methods based on the fusion method

Point cloud data offers the object's spatial geometry information, such as its position, shape, and distance, while the image provides details about the object's look and texture. The integration of point cloud and image data maximises the use of various sensors and has the benefit of enhancing the performance of specific item detection (such as far-off objects and occlusions) [10].

MV3D [17] combines images from multiple camera angles to detect objects in 3D space around a vehicle. Using CNNs, it processes these images to generate predictions about object locations, sizes, and types, improving detection accuracy by addressing the limitations of single-view systems. This makes MV3D especially effective in complex driving scenarios, enhancing safety through better spatial awareness.

ImVoteNet [18] enhances 3D object detection by integrating image-based voting into the point cloud-based 'VoteNet' framework. It combines geometric details from 3D point clouds with texture and context from 2D images, using the images to cast votes on potential object locations. This hybrid approach improves detection accuracy and reliability, particularly in complex environments.

F-PointNet (Frustum PointNet) [19] enhances 3D object detection with RGB-D data by first using a 2D detector on RGB images to define 3D frustums around detected objects. Within these frustums, it processes point clouds using a PointNet-based architecture to segment and classify object points, generating accurate 3D bounding boxes. This method effectively combines colour and depth data for precise object detection, which is particularly useful in autonomous driving and robotics. F-ConvNet (Frustum ConvNet) [20] builds on F-PointNet by integrating CNNs to improve 3D object detection from RGB-D data. The use of convolutional layers enhances feature extraction and object segmentation within frustums, leading to more accurate detection and localisation. This approach also improves robustness, handling varying point cloud densities and environmental conditions more effectively.

RoIFusion [21] enhances 3D object detection by integrating LiDAR and vision sensor data. It combines LiDAR's precise depth information with the rich texture and colour from vision sensors by identifying and fusing Regions of Interest (RoIs) from both sources. This deep learning-based fusion improves detection accuracy and robustness, making it highly effective for autonomous driving applications.

DoBEM [22] uses CNNs to detect and localise vehicles in Bird's Eye View (BEV) elevation images. By processing these top-down 2D images, the model extracts key features like vehicle shapes, sizes, and positions, enabling accurate detection even in complex traffic scenarios. This combination of BEV imagery and CNNs offers fast and reliable vehicle detection for autonomous systems.

The methods for object detection and localisation can be categorised based on the type of data utilised: image-based methods [7-8], LiDAR point cloud-based methods [9], [12-16],[23] and methods that combine image and LiDAR point cloud data [17], [18-20],[22],[24]. The F-PointNet model offers a novel way to combine point cloud and image data for object identification applications. Moreover, this method integrates data from 2D pictures to give more extensive object detection by enhancing the PointNet [11] neural network.

2. Methodology and results

2.1. Methodology

The entire module can be divided into three modules: generating frustum regions, 3D instance segmentation and 3D bounding box prediction. Figure 1 shows the basic process of the model. The entire project implementation process can be divided into data preprocessing (Sec 2.2.1), model construction (Sec 2.2.2), model training and testing (Sec 2.2.3), and model performance evaluation (Sec 2.2.4).

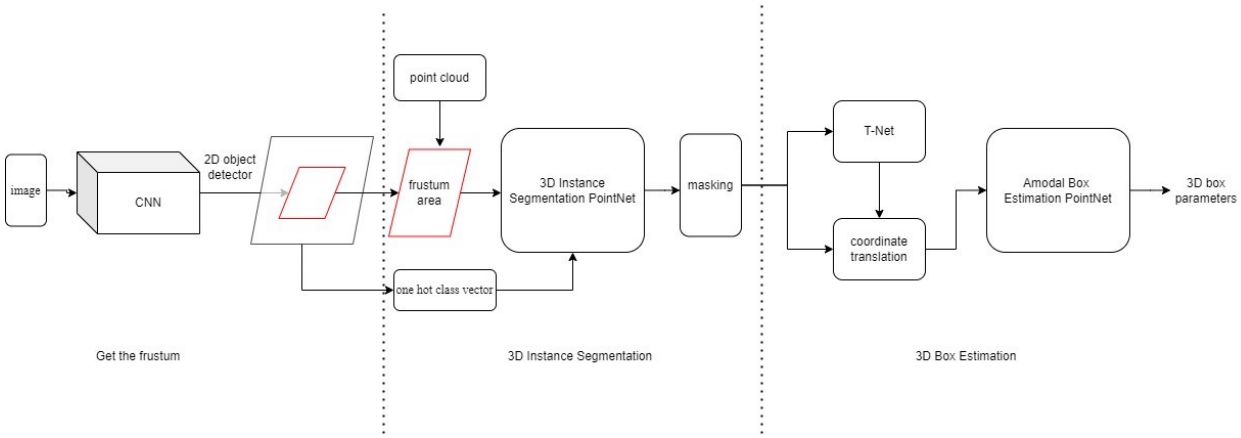

Figure 1. The F-PointNet model is constructed based on the following principle: each 2D bounding box that defines a 3D cone region is initially used as a 2D detector acting on an RGB image. Next, PointNet is chosen to implement 3D instance segmentation and non-modal 3D bounding box estimate based on point clouds, depending on the 3D point clouds in these viewing frustum regions. Consequently, three components comprise the overall model implementation: frustum proposal, 3D instance segmentation and 3D box estimation.

Figure 2 shows the input and output of the point cloud data coordinate system, as well as the transformation of points in different coordinate systems.

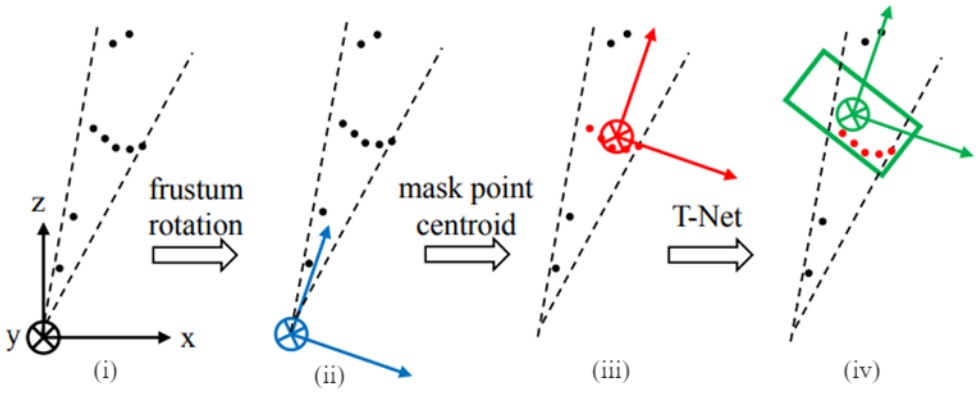

Figure 2. Artificial points (represented as black dots) are used to demonstrate different coordinate systems: (i) the standard camera coordinate system; (ii) the frustum coordinate system, which is adjusted to a centred view following rotation (Sec 2.1.1); (iii) the mask coordinate system, where the centroid of object points is positioned at the origin (Sec 2.1.2); (iv) the object coordinate system as predicted by T-Net (Sec 2.1.3).

2.1.1. Generate frustum

A critical component of F-PointNet is the FPN [26] 2D detector, a fully convolutional network used to generate 2D bounding boxes from images. These bounding boxes are then utilised to extract corresponding frustums from the 3D point cloud, facilitating further 3D analysis and object recognition. Figure 3 illustrates the complete model structure of the FPN.

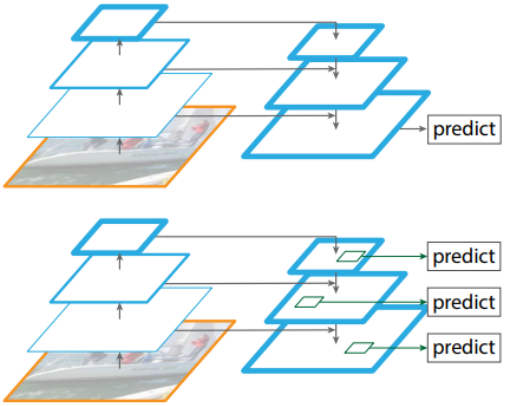

Figure 3. The top image shows a top-down approach with skip connections, where predictions are made at the most granular level. The bottom image shows how the FPN features a similar structure but utilises it as a feature pyramid to enable independent predictions at all levels.

FPN (Feature Pyramid Network) uses a single convolutional neural network to build a pyramid-shaped feature hierarchy, enabling object detection at multiple scales. Unlike traditional single-scale networks, FPN combines high-level semantic information with detailed high-resolution features through a top-down architecture, improving its ability to detect objects of various sizes, especially smaller ones.

The frustum is defined by the near and far planes specified by the range of a depth sensor, delineating the 3D search space for objects [19]. Given that the resolution of data produced by current 3D sensors, particularly real-time depth sensors used in autonomous driving technologies, remains lower than that of 2D images captured by commercial cameras, this method employs well-established 2D object detectors to identify object regions in RGB images and classify the objects. The detected 2D bounding boxes are utilised to determine the boundaries of the frustum. Subsequently, all points within the frustum are collected to form a frustum point cloud, which is then used to complete 3D instance segmentation.

In the image, each selected Frustum direction may not be consistent, which will cause the position of the point cloud to change, causing errors. In order to ensure the consistency of each Frustum, as shown in Figure 2 (ii), the central axis is orthogonal to the plane of the image through a rotation operation. Normalisation operations can be used to improve the selection invariance of the algorithm.

2.1.2. D instance segmentation

Objects in photos typically appear to be occluded, as shown in Figure 4, which presents a significant obstacle to 3D detection. This will significantly impact the direct use of 2D depth maps to return 3D distances. Nevertheless, Frustum point clouds - which serve as depth information - replace 2D images in F-PointNet. Since the occlusion issue won't interfere with the point cloud data, PointNet can be used to process Frustum point cloud data to fix the occlusion problem. In this phase, F-PointNet performs 3D instance segmentation on the Frustum point cloud using the PointNet++ network, and it predicts the likelihood that each point belongs to each class.

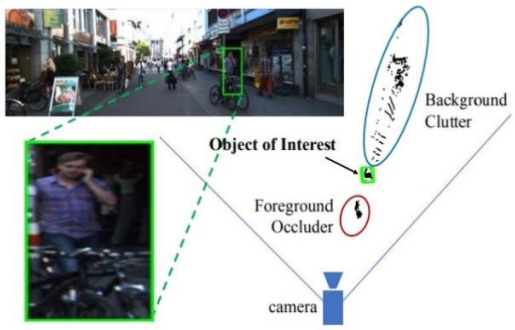

Figure 4. Challenges in detecting 3D objects within a frustum point cloud include the following: In the left image, there's an RGB image showing a proposed region for a person. On the right, there's a bird's eye view of LiDAR points within an extruded frustum based on a 2D bounding box. Here, we observe a significant dispersion of points, including foreground occluders like bikes and background clutter such as buildings.

Considering that a prior knowledge object is naturally separated in a 3D scene, this is simpler than segmentation on a 2D image because objects that are not adjacent in 2D may also be very close in pixels [19]. Therefore, a 3D instance segmentation method is adopted; this method is based on the PointNet segmentation network and performs segmentation in the view frustum.

In the F-PointNet model, the network initially defines a viewing frustum in 3D space to capture the point cloud of the target object. Each view frustum typically encompasses only the intended target but may also include points from irrelevant areas such as the ground, vegetation or other occluded or background objects. The network is trained to handle these challenges, effectively recognising the geometry of specific object classes despite the occlusion and clutter. For more efficient 3D instance segmentation, F-PointNet leverages semantic information provided by 2D detectors. For instance, if the target is identified as a car, the segmentation network preprocesses the data to focus on car-like geometries. Specifically, the model incorporates semantic category information into a one-hot-encoded vector (a k-dimensional vector for predefined k categories) and merges this vector with the point cloud's intermediate features. This approach enhances the model's accuracy in segmenting 3D instances, enabling it to precisely identify and segment points associated with the objects of interest.

Semantic segmentation can be used to filter each point cloud, and all point clouds that are anticipated to belong to a particular class are retrieved and referred to as 'masking'. For the time being, the algorithm's translation invariance needs to be improved by normalising these point cloud coordinates. In this particular implementation, the Frustum coordinate system is further localised intuitively by deducting the centre of mass of Frustum from the point XYZ value. This stage only involves data processing; the point cloud is not rotated or translated in any way. The steps are displayed in Figure 2 (iii).

2.1.3. D box estimation

Using the segmented object points situated within the 3D mask coordinate system, this component predicts the amodal-oriented 3D bounding box of the object. It employs a box regression PointNet in conjunction with a preprocessing transformer network for this estimation process.

T-Net [27] introduces a high-order tensor architecture, replacing traditional multiple tensors with a single advanced tensor for network parameters. This approach reduces memory usage and computational demands by using tensor decomposition, improving efficiency during training and inference. T-Net is particularly relevant for fully convolutional networks in tasks like semantic segmentation, offering a more streamlined yet powerful alternative to conventional FCN frameworks.

Table 1. The importance of three coordinate rotations

Frustum rot	Mask centralise	T-net	Accuracy
-	-	-	12.5
√	-	-	48.1
√	√	-	64.6
√	√	-	71.5
√	√	√	**74.3**

Even when a local coordinate transformation is utilised for the point cloud within the Frustum, there might be a significant discrepancy from the actual location of the target. In such cases, F-PointNet employs a component known as T-Net to deduce the object's true centre, as depicted in Figure 2 (iv). This module adjusts the local coordinates to recentre the predicted object's centre at the coordinate origin. The boundary of the object is parameterised by dimensions (x, y, height, width, length and angle). The accurate location of the object is computed by adding the initial local coordinates of the masked point cloud to the substantial correction from T-Net and a slight additional offset, resulting in the predicted centre of the object:

$$Cpred = Cmask + \Delta Ct\text{-}net + \Delta Cbox\text{-}net$$

$Cpred$ represents the final predicted bounding box or target area. This is the final output we want to get from the model, including the position and dimensions of the 3d bounding box. $Cmask$: represents the initial prediction based on the object mask; this is derived from an F-PointNet segmentation model that predicts a pixel-level mask covering the target object. $\Delta Ct\text{-}net$: represents the adjustment term produced by a specific network (called $t\text{-}net$), which is used to fine-tune or correct the prediction of the mask position. This can be a correction vector based on the learned object position deviation. $\Delta Cbo\text{-}ne$ə: represents an adjustment term calculated by another network (called $box\text{-}net$) specifically used to adjust the size and shape of the predicted bounding box. This is usually an initial prediction based on a mask to more accurately match the actual boundaries of the object. The size and orientation of the bounding box are then estimated through a combined approach of classification and regression.

2.2. Results

In the results section, the analysis is divided into three parts. The first part compares the model's performance after training using different preprocessing methods. The second part contrasts these results with those obtained from other methods. The third part discusses the strengths and limitations of the applied methods.

2.2.1. Comparison with different disturbances

During the data preprocessing stage, two data augmentation methods are employed to enhance the model's generalisation capabilities: random perturbation of 2D bounding boxes and specified perturbation of 2D bounding boxes. Consequently, these pre-processed datasets are utilised to train the model, after which the model's performance is assessed. The effectiveness of the two data augmentation methods is compared table 2 presents the accuracy for three detection categories evaluated on the ground plane, while table 3 details the accuracy of three categories of 3D bounding box predictions.

Table 2. 3D object localisation AP (bird's eye view) on the KITTI test set. The models from both methods are separately evaluated to predict the AP across three categories and three levels of difficulty.

Method	Car			Pedestrians			Cyclists		
	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
Designated disturbance	85.88	73.70	65.22	68.23	58.45	51.46	**79.86**	**57.60**	**53.66**
Random disturbance	**87.10**	**81.93**	**73.71**	**70.23**	**60.83**	**53.24**	74.04	56.00	52.63

Table 2 illustrates the impact of two data perturbation methods (designated and random) on the positioning accuracy of 3D objects in three categories (cars, pedestrians and cyclists). These results indicate that random perturbation generally outperforms designated perturbation, particularly in detecting automobiles, likely due to its enhancement of the model's adaptability to varying scenarios and improved generalisation. However, in easy bicycle detection tasks, designated perturbation performed slightly better, suggesting a scenario-specific advantage. Overall, increasing difficulty correlates with decreasing accuracy across all categories, reflecting the significant impact of factors like occlusion and background interference in complex scenes.

Table 3. 3D object detection 3D AP on the KITTI test set. The models from both methods are separately evaluated to predict the AP across three categories and three levels of difficulty.

Method	Car			Pedestrians			Cyclists		
	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
Designated disturbance	74.24	59.99	52.42	62.82	51.97	47.23	**76.27**	**53.91**	**49.59**
Random disturbance	**83.10**	**69.09**	**62.18**	**67.10**	**57.06**	**50.16**	68.89	52.22	48.07

Table 3 illustrates the impact of two data perturbation methods - designated and random - on the detection of cars, pedestrians and bicycles across three levels of difficulty: easy, medium and difficult.

The data indicates that random perturbation generally outperforms designated perturbation in detecting cars and pedestrians, particularly at medium and difficult levels; this suggests that random perturbations may enhance the model's ability to generalise in complex environments. For bicycle detection, random perturbation slightly reduced performance at the easy level, likely due to the introduced changes causing interference. Overall, increased difficulty led to reduced detection accuracy across all categories, reflecting the challenges posed by more complex detection scenarios.

2.2.2. Comparison with other methods

Table 4. 3D object detection 3D AP on the KITTI test set. Compared with the MV3D and DoBEM models, F-PointNet demonstrates superior performance.

Method	Car			Pedestrians			Cyclists		
	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
DoBEM [13]	7.42	6.95	13.45	-	-	-	-	-	-
MV3D [12]	71.09	62.35	55.12	-	-	-	-	-	-
Designated disturbance	74.24	59.99	52.42	62.82	51.97	47.23	**76.27**	**53.91**	**49.59**
Random disturbance	**83.10**	**69.09**	**62.18**	**67.10**	**57.06**	**50.16**	68.89	52.22	48.07

This table compares the performance of three different 3D object detection methods: DoBEM [13], MV3D [18] and F-PointNet in detecting cars, pedestrians and bicycles across various difficulty levels (easy, medium and hard).

2.2.3. Discussion

Performance evaluations on the KITTI test set compare 3D object localisation and detection using two data perturbation methods - designated and random - across difficulty levels (easy, medium and hard) for cars, pedestrians and bicycles. Random perturbation generally outperforms designated perturbation, especially in car detection, suggesting improved model generalisation in diverse scenarios. However, designated perturbation shows a slight advantage in easy-level bicycle detection. Accuracy decreases with difficulty due to obstacles and background interference.

In 3D detection, random perturbation excels in detecting cars and pedestrians at moderate and difficult levels, while its performance drops slightly for bicycles at the easy level. Comparisons of three 3D detection methods - DoBEM, MV3D, and F-PointNet - show that F-PointNet significantly outperforms the others. DoBEM has limited performance (13.45% for cars at difficult levels), while MV3D declines from 71.09% at the easy level to 55.12% at the difficult level. F-PointNet, especially with random perturbation, leads with 83.10% accuracy at the easy level.

Overall, random perturbation improves generalisation in complex environments, while designated perturbation may be better suited for specific tasks. F-PointNet's success highlights the effectiveness of a single-view approach in 3D detection without sensor fusion. Future research should focus on optimising these strategies for broader applications.

3. Conclusion and future work

3.1. Conclusion

The exploration of 3D object detection for autonomous driving emphasises the importance of innovation and data perturbation in improving performance. F-PointNet, especially with random perturbation, surpasses methods like MV3D and DoBEM by integrating image and point cloud data without requiring sensor fusion. This reduces complexity while boosting efficiency and accuracy. Random perturbation enhances adaptability in complex environments, while designated perturbation remains useful in predictable scenarios. Advancements in 3D detection, such as F-PointNet, are crucial for improving the safety and reliability of autonomous driving systems. Future research should focus on refining detection methods and optimising perturbation techniques to enhance precision and adaptability.

3.2. Future work

The study highlights limitations, particularly with designated perturbations and in less variable conditions where adaptability is underutilised. While random perturbation excels in complex scenarios, it slightly reduces performance in simpler tasks like easy-level bicycle detection. Future research should optimise perturbation methods to balance adaptability and accuracy, possibly exploring hybrid techniques. Additionally, improving the model's efficiency in processing multimodal data and refining neural network architectures could enhance detection accuracy and speed, which are crucial for real-time applications like autonomous driving.

References

[1] Karangwa, J., Liu, J., & Zeng, Z. (2023). Vehicle Detection for Autonomous Driving: A Review of Algorithms and Datasets. *IEEE Trans. Intell. Transp. Syst*., vol. 24, no. 11, pp. 11568-11594.

[2] Du, X., Ang, M. H., Karaman, S., & Rus, D. (2018). A General Pipeline for 3D Detection of Vehicles. *Proc. IEEE Int. Conf. Robot. Autom*., Brisbane, QLD, Australia, pp. 3194-3200, doi: 10.1109/ICRA.2018.8461232.

[3] Oksuz, K., Cam, B. C., Kalkan, S., & Akbas, E. (2021). Imbalance Problems in Object Detection: A Review. *IEEE Trans. Pattern Anal. Mach. Intell*., vol. 43, no. 10, pp. 3388-3415, doi: 10.1109/TPAMI.2020.2981890.

[4] Guo, X., Ji, Z., Feng, Q., Wang, H., Yang, Y., & Li, Z. (2023). URS: A Light-Weight Segmentation Model for Train Wheelset Monitoring. *IEEE Trans. Intell. Transp. Syst*., vol. 24, no. 7, pp. 7707-7716.

[5] SAE International (2018). Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles. *SAE Int*., vol. 4970, no. 724, pp. 1-5.

[6] Li, Z., Du, Y., Zhu, M., Zhou, S., & Zhang, L. (2022). A survey of 3D object detection algorithms for intelligent vehicles development. *Artif. Life Robot*., vol. 27, pp. 1-8.

[7] Xu, B., & Chen, Z. (2018). Multi-level fusion based 3d object detection from monocular images. *Proc. IEEE Conf. Comp. Vis. Pattern Recognit*., pp. 2345-2353.

[8] Wang, Y., Guizilini, V. C., Zhang, T., Wang, Y., Zhao, H., & Solomon, J. (2022). Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. *Conf. Robot Learn*., pp. 180-191. PMLR.

[9] Zhou, Y., & Tuzel, O. (2018). Voxelnet: End-to-end learning for point cloud based 3d object detection. *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*., pp. 4490-4499.

[10] Zhao, K. et al. (2022). 3D Vehicle Detection Using Multi-Level Fusion From Point Clouds and Images. *IEEE Trans. Intell. Transp. Syst*., vol. 23, no. 9, pp. 15146-15154, doi: 10.1109/TITS.2021.3137392.

[11] Qi, C. R. et al. (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*.

[12] Qi, C. R. et al. (2017). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Inf. Process. Syst*., vol. 30.

[13] Li, Y. et al. (2018). Pointcnn: Convolution on x-transformed points. *Advances in Neural Inf. Process. Syst*., vol. 31.

[14] Fan, L., Xiong, X., Wang, F., Wang, N., & Zhang, Z. (2021). Rangedet: In defense of range view for lidar-based 3d object detection. *Proc. IEEE/CVF Int. Conf. Comp. Vis*., pp. 2918-2927.

[15] Beltrán, J., Guindel, C., Moreno, F. M., Cruzado, D., Garcia, F., & De La Escalera, A. (2018). Birdnet: a 3d object detection framework from lidar information. 2018 21st Int. Conf. Intel. Trans. Sys. (ITSC), pp. 3517-3523. *IEEE*.

[16] Li, B. (2017). 3d fully convolutional network for vehicle detection in point cloud. 2017 IEEE/RSJ Int. Conf. Intell. Robots Sys. (IROS), pp. 1513-1518. *IEEE*.

[17] Chen, X., Wan, J., Li, B., & Xia, T. (2017). Multi-view 3D Object Detection Network for Autonomous Driving. *IEEE*. https://doi.org/10.1109/cvpr.2017.691

[18] Qi, C. R. et al. (2020). Imvotenet: Boosting 3d object detection in point clouds with image votes. *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit*., pp. 4404-4413.

[19] Qi, C. R. et al. (2018). Frustum pointnets for 3D object detection from rgb-d data. *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*., pp. 918-927.

[20] Wang, Z., & Jia, K. (2019). Frustum ConvNet: Sliding frustums to aggregate local point-wise features for amodal 3D object detection. *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst*. (IROS), pp. 1742-1749.

[21] Chen, C., Fragonara, L. Z., & Tsourdos, A. (2021). RoIFusion: 3D object detection from LiDAR and vision. *IEEE Access*, 9, 51710-51721.

[22] Yu, S. L., Westfechtel, T., Hamada, R., Ohno, K., & Tadokoro, S. (2017). Vehicle detection and localization on bird's eye view elevation images using convolutional neural network. 2017 *IEEE Int. Sym. Safety, Sec. Resc. Robot*. (SSRR), pp. 102-109. IEEE.

[23] Yan, Y., Mao, Y., & Li, B. (2018). SECOND: Sparsely embedded convolutional detection. *Sensors*, vol. 18, no. 10, pp. 3337.

[24] Zou, Z., Chen, K., Shi, Z., Guo, Y., & Ye, J. (2023). Object detection in 20 years: A survey. Proc. *IEEE*, 111(3), 257-276.

[25] Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*., pp. 2117-2125.

[26] Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. *Proc. IEEE Conf. Comput. Vis. Pattern Recognit*., pp. 2117-2125.

[27] Kossaifi, J., Bulat, A., Tzimiropoulos, G., & Pantic, M. (2019). T-net: Parametrizing fully convolutional nets with a single high-order tensor. *Proc. of the IEEE/CVF Conf. Comp. Vis. Pattern Recognit*., pp. 7822-7831.