Online-Journals.org

iJIM International Journal of Interactive Mobile Technologies

iJIM | eISSN: 1865-7923 | Vol. 20 No. 8 (2026) | OPEN ACCESS

https://doi.org/10.3991/ijim.v20i08.59947

PAPER

VoxAdapt: Adaptive Multi-Scale 3D Object Detection for Real-Time Mobile Applications and Edge Systems

Daham Pathiraja (✉), **Indika Perera**

University of Moratuwa, Moratuwa, Sri Lanka

daham.23@cse.mrt.ac.lk

ABSTRACT

Deploying real-time 3D perception capabilities on mobile and edge platforms, such as autonomous robots, drones, and intelligent sensing systems, requires balancing detection accuracy against strict computation and memory constraints. Existing voxel-based LiDAR perception pipelines rely on fixed voxel sizes that must be manually tuned for each dataset and sensor, limiting their adaptability across deployment environments. We introduce VoxAdapt, an adaptive multi-scale 3D object detection framework that treats voxel-scale values as learnable parameters updated via a surrogate gradient pathway, enabling task-driven optimization without requiring differentiation using discrete voxel indexing. Unlike prior approaches that adapt feature aggregation under fixed discretization, VoxAdapt allows voxel resolutions to be adjusted during training in response to the detection objectives and resource constraints. Experiments on the KITTI benchmark demonstrated that VoxAdapt enables robust detection of small, sparsely sampled objects that challenge fixed-scale methods while maintaining competitive performance on larger objects with minimal computational overhead. These results highlight the potential of learning adaptive geometric representations to support efficient and deployable 3D perception systems for real-time mobile and edge applications.

KEYWORDS

mobile edge computing, real-time 3D perception, intelligent sensing systems, adaptive deep learning, resource-constrained deployment, LiDAR-based object detection, multi-scale representations, mobile robotics, interactive mobile systems

1 INTRODUCTION

In recent years, 3D object detection has evolved to overcome sparse and unevenly distributed LiDAR point cloud data. Voxel-based techniques, such as VoxelNet [1], partition space into fixed voxels; however, a fixed voxel size that is hand-tuned tends to fail to balance high detail near object boundaries with efficiency in backgrounds. SECOND [2] boosted efficiency by using sparse 3D convolutions, enabling the scaling of voxel-based pipelines. Subsequently, techniques such as Voxel-FPN [3] and

Pathiraja, D., and Perera, I. (2026). VoxAdapt: Adaptive Multi-Scale 3D Object Detection for Real-Time Mobile Applications and Edge Systems. *International Journal of Interactive Mobile Technologies (iJIM)*, 20(8), pp. 87–102. https://doi.org/10.3991/ijim.v20i08.59947

Article submitted 2025-12-02. Revision uploaded 2026-02-09. Final acceptance 2026-02-24.

HVNet [4] added multi-scale feature aggregation, while others attempted voxel-point hybrid designs [5] and compensation schemes [6]. However, most techniques continue to depend on manually selected voxel sizes and predefined aggregation hierarchies. Similar efficiency–adaptivity trade-offs have been explored in interactive mobile intelligent systems, where learning-based mechanisms are used to dynamically adapt model behavior under resource constraints, rather than relying on fixed configurations [7].

Recently, transformer-based architectures have been applied to LiDAR detection [8], [9], [10], offering powerful global context modeling. However, these methods primarily focus on sequence modeling and do not explicitly address voxel rigidity or the challenge of scale selection. In contrast, our approach aims to make voxel-scale selection learnable through a surrogate gradient pathway without requiring differentiation via discrete voxel indexing. This learnable multi-scale voxelization balances efficiency and accuracy, which aligns with the broader trend toward adaptive three-dimensional (3D) representations [11].

To achieve this, we propose VoxAdapt, a framework that treats voxel scale values as learnable parameters updated via a surrogate gradient pathway jointly with detection network training. Unlike prior work, VoxAdapt learns both the assignment policy (i.e., which scale to use for each point) and the geometric parameters (i.e., the values of each scale). Our method combines a lightweight scale-prediction network (ScaleNet) with differentiable Gumbel-Softmax relaxation and trainable scale parameters (nn.Parameter). Because voxel indexing involves a non-differentiable floor operation, gradient signals reach voxel-scale parameters through Feature-wise Linear Modulation (FiLM) [12], which conditions output features on the predicted scales. Features from multiple learned scales are fused according to soft assignment probabilities, adaptively allocating finer resolutions to the object boundaries and coarser scales to the homogeneous regions. By introducing minimal computational overhead while preserving detection accuracy, VoxAdapt follows a design philosophy consistent with deep learning systems deployed at the mobile edge, where real-time responsiveness and computational efficiency are critical requirements [13]. This architecture introduces a minimal computational overhead while maintaining real-time performance, making it suitable for mobile robots, drones, and edge devices.

The remainder of this paper is organized as follows. Section 2 reviews the literature on voxel-based 3D object detection, multiscale representations, and adaptive voxelization techniques. Section 3 presents the research methodology, describing the VoxAdapt framework, including the learnable scale parameters, differentiable scale assignment via ScaleNet, and weighted multi-scale feature fusion. Section 4 reports the experimental results on the KITTI benchmark, including quantitative comparisons, ablation studies, and computational efficiency analyses. Finally, Section 5 concludes the paper by summarizing the main contributions, highlighting the implications of learnable voxelization for mobile and edge deployments, and outlining future research directions.

2 LITERATURE REVIEW

2.1 3D object detection methods

Voxel-based methods. Voxel-based approaches represent 3D point clouds as regular grids, thereby enabling the use of 3D convolution for feature extraction. VoxelNet [1] pioneered end-to-end learning directly from raw LiDAR points by combining voxel feature encoding with detection heads. SECOND [2] improved efficiency through sparse 3D convolutions, whereas PointPillars [14] introduced a pseudo-image representation that reduced the computational cost. Despite their success, these methods rely on fixed voxel sizes (commonly 0.05–0.2 m), which can lead

to information loss when the object geometry is not well aligned with the voxel boundaries, particularly for small or irregularly shaped objects.

Point-based methods. Point-based networks operate directly on raw point clouds without voxelization, thereby preserving the fine geometric details. Notable examples include PointNet [15], PointNet++ [16], PointRCNN [17], and 3DSSD [18]. These approaches avoid the quantization error introduced by voxels; however, their computational cost scales poorly with dense point clouds, often processing many irrelevant points. Techniques such as semantics-augmented set abstraction [19] mitigate this by emphasizing foreground points; however, real-time performance remains challenging in complex scenes.

Hybrid approaches. Hybrid methods combine voxel-and point-based processing to leverage the advantages of both representations. For instance, some studies integrate voxel-based feature extraction with point-based refinement, whereas others fuse LiDAR with RGB images [20] or radar (e.g., MCHFormer [10]). These approaches generally improve accuracy and flexibility; however, the underlying voxelization still relies on predefined fixed voxel scales, limiting their adaptability to objects of varying sizes or sparsity patterns.

2.2 Multi-scale processing in 3D

Multiscale processing has been widely explored to improve 3D detection, particularly for objects of varying sizes. Static multiscale methods, such as Voxel-FPN [3] and SMS-Net [21], fuse features from multiple voxel scales to enhance detection, sometimes by combining data from BEV maps, RGB images, or radar. Although effective, these methods rely on manually tuned voxel resolutions, reducing flexibility and requiring significant engineering efforts for new datasets or sensor setups to achieve optimal performance.

Adaptive multi-scale strategies attempt to dynamically adjust processing according to the input properties. In 2D vision, Feature Pyramid Networks (FPNs) [22] employ hierarchical feature maps to address scale variations. In 3D approaches, such as MLCVNet [23] and dynamic sparse voxelization [24], self-attention, hierarchical pyramids, and density-based voxel refinement are employed to adapt features across different scales. Other methods, including multi-scale CNNs, Multi-scale Context Aggregation (MCA) [25], scale-aware multi-branch feature modeling as in TridentNet [26], and AFE-RCNN [27], focus on improving scene robustness and detecting small objects.

2.3 Gap and motivation

Despite the progress in multi-scale 3D detection, a fundamental limitation persists: existing methods treat voxel-scale values as fixed hyperparameters. Multiscale approaches learn to combine features from predefined scales, whereas attention-based methods learn to weight spatial regions; however, the underlying voxel sizes remain static. This decoupling isolates voxelization, a critical step determining spatial resolution and information loss, from task-driven optimization. When sensor characteristics, scene distributions, or detection targets change, practitioners must manually retune voxel scales through an expensive grid search.

This gap is particularly problematic for mobile and edge deployments with varying computational budgets. Fixed scales force a compromise: fine voxels capture small objects but increase memory usage, whereas coarse voxels are efficient but lose geometric details. Density-based refinement heuristics partially address this issue using handcrafted rules rather than learning from detection objectives.

We argue that voxel scales should receive gradient signals from the same loss that governs all other parameters. Because voxel indexing involves a non-differentiable floor operation, direct backpropagation through voxelization is not possible. VoxAdapt addresses this by introducing a surrogate gradient pathway: learnable scale parameters are combined with soft scale assignments via Gumbel-Softmax relaxation, and the resulting predicted scales condition the output features through FiLM modulation. This creates a differentiable path from the detection loss to the scale parameters, without requiring differentiation through discrete voxel indexing. This approach enables (1) reduced manual tuning of voxel sizes, (2) adaptive per-point scale assignment responding to detection objectives, and (3) the potential for transfer across sensor configurations. We acknowledge that VoxAdapt provides approximate gradient signals through a surrogate pathway rather than exact end-to-end differentiation and that we do not provide theoretical convergence guarantees.

3 RESEARCH METHODOLOGY

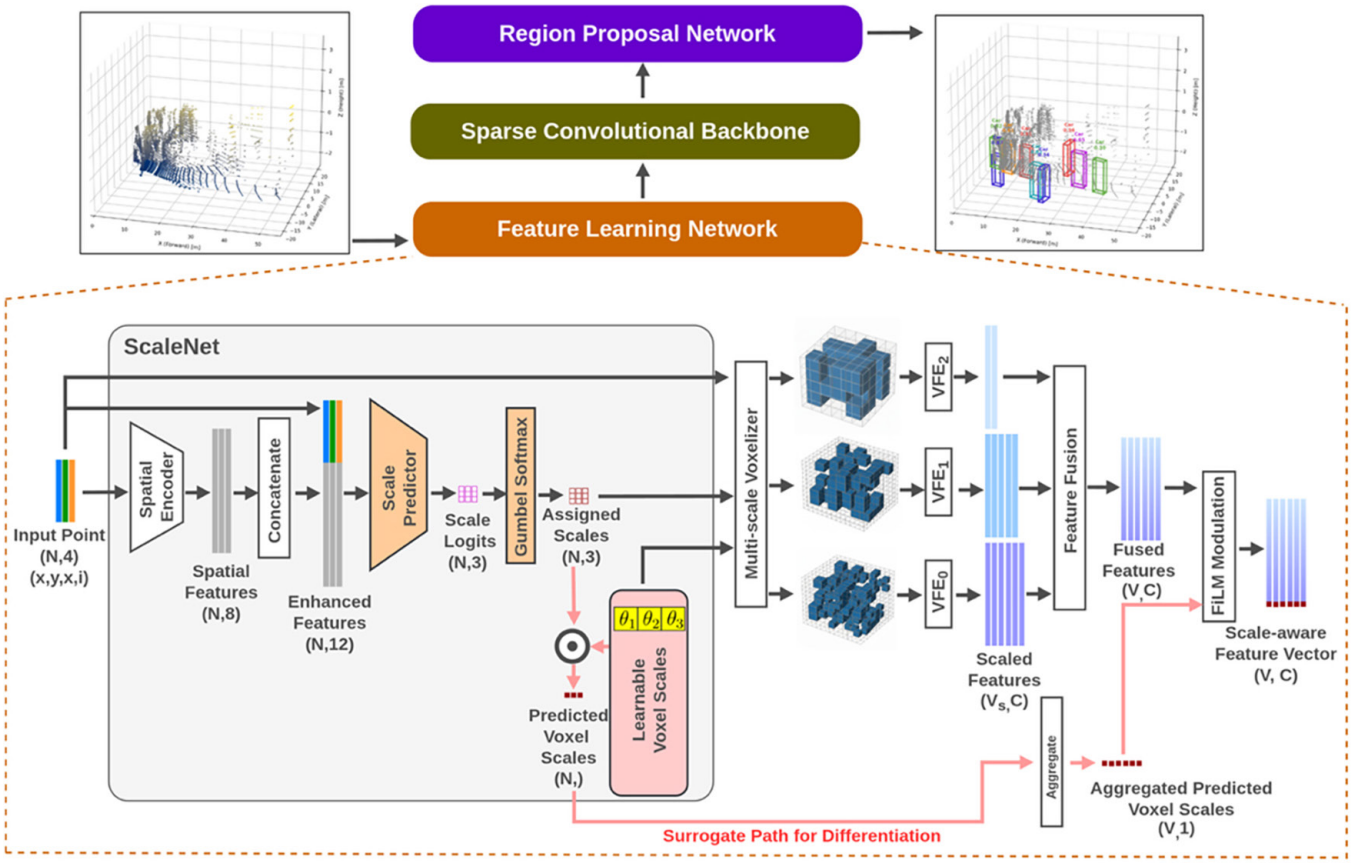

Fig. 1. VoxAdapt architecture. The feature learning network takes a raw point cloud $P \in R^{N\times 4}$ (coordinates and intensity) as input and processes it through ScaleNet, which comprises a spatial encoder producing spatial features (N, 8) that are concatenated with normalized input features to form enhanced features (N, 12). The scale predictor generates scale logits (N, K) over K learnable voxel scale parameters $\theta = \{\theta_1, \theta_2, \theta_3\} \in R^K$ ($K = 3$ in experiments). Gumbel–Softmax relaxation converts logits to soft scale assignments $\sigma \in R^{N\times K}$, which are combined with the learnable scales to compute per-point predicted voxel scales $\hat{s}_i = \sum_k \sigma_i^{(k)} \cdot \theta_k$ (shape (N,)). This surrogate path (shown in pink) enables end-to-end gradient flow to θ without requiring differentiation through discrete voxel indexing. Points are voxelized at each scale via the multi-scale voxelizer, and scale-specific VFEs (VFE_0, VFE_1, VFE_2) transform each voxel grid into scaled features (V_S, C). Feature fusion combines multi-scale representations into fused features (V, C), where V is the total number of non-empty voxels and $C = 64$ is the feature dimension. The aggregated predicted voxel scales $\bar{s} \in R^{V\times 1}$ condition the fused features via Feature-wise Linear Modulation (FiLM), producing scale-aware feature vectors $F_{out} \in R^{V\times C}$. The sparse convolutional backbone processes the resulting tensor to aggregate spatial context, and an RPN generates the final 3D detections

We propose VoxAdapt, a voxel-based 3D object detection framework that treats the voxel discretization scale as a trainable geometric parameter rather than a fixed preprocessing hyperparameter. Figure 1 illustrates the proposed architecture. Unlike prior voxel-based detectors that adapt feature aggregation or attention under a fixed discretization, VoxAdapt enables voxel geometry itself to evolve during training in response to detection objectives. Crucially, VoxAdapt does not assume the differentiability of voxel indexing; voxel discretization remains discrete, whereas voxel-scale parameters are optimized iteratively across training iterations via a surrogate gradient pathway, with updated scales applied in subsequent forward passes, where they directly modify voxel sizes and point-to-voxel assignments.

Let a LiDAR point cloud be defined as

$$P = \{p_i = (x_i, y_i, z_i, r_i) \mid i = 1, \ldots, N\}$$

where (x_i, y_i, z_i) are the Cartesian coordinates and r_i denotes the reflectance intensity. Conventional voxel-based detectors discretize space using a fixed voxel size $v \in R^+$, imposing a rigid trade-off between geometric fidelity and computational efficiency. Such discretization choices are manually tuned and decoupled from task supervision, limiting their adaptability across datasets, sensors, and deployment scenarios. To address this limitation, VoxAdapt introduces a set of learnable voxel-scale parameters

$$\theta = \{\theta_k\}_{k=1}^{K}, \theta_k > 0$$

where each θ_k defines the XY voxel size of one discretization branch, while the Z resolution remains fixed at 0.2 m for computational efficiency. Unlike multiscale methods that fix the voxel resolution a priori, VoxAdapt explicitly includes θ in the optimization problem as trainable parameters, enabling gradient-based updates during training. Because the hard assignment of points to voxel scales is non-differentiable, VoxAdapt employs ScaleNet, a lightweight neural network that predicts per-point scale preferences (see the ScaleNet block in Figure 1). ScaleNet consists of a spatial encoder and a scale predictor. The spatial encoder is a three-layer multilayer perceptron that maps each 3D point coordinate to an 8-dimensional spatial feature representation, using ReLU activations and dropout for regularization. Formally, the spatial feature for point i is computed as

$$h_i = \text{SpatialEncoder}(x_i, y_i, z_i) \in R^8$$

The input features are then normalized and concatenated with the spatial features to form $e_i = [\tilde{p}_i; h_i] \in R^{12}$, which the scale predictor MLP transforms into logits $l_i = f\phi(e_i) \in R^K$ for each scale. These logits are converted into soft scale-assignment weights using Gumbel-Softmax relaxation:

$$\sigma_i^{(k)} = \frac{\exp((l_i^{(k)} + g_k)/\tau)}{\sum_{j=1}^{K} \exp((l_i^{(j)} + g_j)/\tau)}$$

where $g_k \sim G\text{umbel}(0,1)$ and $\tau = 0.5$ is a fixed temperature parameter. We use a fixed temperature rather than annealing to maintain stable soft assignments throughout training, which empirically provides better gradient flow to the learnable scale parameters.

A key aspect of VoxAdapt is that each point receives its own predicted voxel scale through a weighted combination of the learnable scale parameters:

$$\hat{s}_i = \sum_{k=1}^{K} \sigma_i^{(k)} \cdot \theta_k$$

This operation produces a tensor of shape $(N,)$—one scale value per point, as shown in the surrogate path in Figure 1 (pink arrows). The computation multiplies the soft assignments (shape $N \times K$) element-wise with the broadcasted learnable scales (shape $1 \times K$), and then sums along the scale dimension to yield one predicted scale per point. This formulation ensures that each point's predicted scale is a differentiable function of both the soft assignments σ_i and the learnable scales θ_k, that points strongly assigned to scale k contribute more weight from θ_k, and that gradients flow through both σ and θ during backpropagation.

For each scale θ_k, voxel indices are computed using standard discrete voxelization:

$$c_i^{(k)} = \lfloor p_i / \theta_k \rfloor$$

This operation is non-differentiable and remains unchanged from prior voxel-based detectors. Each voxel grid is processed by a scale-specific Voxel Feature Encoder (VFE_0, VFE_1, VFE_2 in Figure 1), producing voxel-level features $f^{(k)} \in R^{V_k \times C}$ where V_k is the number of non-empty voxels at scale k and C is the feature dimension.

VoxAdapt fuses multi-scale features using Feature-wise Linear Modulation (FiLM), which creates a strong gradient pathway from the detection loss to the learnable scale parameters. First, the mean predicted scale across all points is computed:

$$\bar{s} = \frac{1}{N} \sum_{i=1}^{N} \hat{s}_i$$

The multi-scale features are concatenated as $f_{concat} = [f^{(1)}; f^{(2)}; \ldots; f^{(K)}]$ and processed through a fusion MLP with residual connections to produce f_{fused}. FiLM modulation is then applied using the mean scale:

$$\gamma = MLP_\gamma(\bar{s}) \in R^C, \quad \beta = MLP_\beta(\bar{s}) \in R^C$$

$$f_{out} = f_{fused} \odot (1 + \gamma) + \beta$$

where $\odot$ denotes element-wise multiplication (see FiLM Modulation block in Figure 1). This formulation produces output features of dimension (V, C) rather than $(V, C + 1)$, as scale information is incorporated through feature modulation rather than concatenation. The FiLM networks, implemented as a two-layer multilayer perceptron with an intermediate hidden dimension of $C/2$ and ReLU activation, provide a strong gradient pathway in which $\partial f_{out} / \partial \bar{s}$ exists through γ and β, enabling direct gradient flow to q. The multiplicative interaction allows scale information to modulate every feature channel, providing richer conditioning than simple concatenation.

Since voxel indexing involves a floor operation, VoxAdapt does not attempt to differentiate through it. Instead, voxel-scale parameters are optimized using a surrogate gradient pathway. Applying the chain rule, the gradient of the loss with respect to each voxel scale parameter flows through the FiLM modulation:

$$\frac{\partial L}{\partial \theta_k} = \frac{\partial L}{\partial f_{out}} \cdot \frac{\partial f_{out}}{\partial \gamma} \cdot \frac{\partial \gamma}{\partial \bar{s}} \cdot \frac{\partial \bar{s}}{\partial \theta_k} + \frac{\partial L}{\partial f_{out}} \cdot \frac{\partial f_{out}}{\partial \beta} \cdot \frac{\partial \beta}{\partial \bar{s}} \cdot \frac{\partial \bar{s}}{\partial \theta_k}$$

The key term is:

$$\frac{\partial \bar{s}}{\partial \theta_k} = \frac{1}{N}\sum_{i=1}^{N} \sigma_i^{(k)}$$

which shows that the gradient $\partial \hat{s}_i / \partial \theta_k = \sigma_i^{(k)}$ is simply the soft assignment weight. Points strongly assigned to scale k contribute more to the gradient of θ_k, the factor $1/N$ distributes gradients uniformly across all points, and soft assignments from Gumbel-Softmax provide differentiable gradients even though the forward pass uses discrete voxelization. Figure 2 provides a detailed visualization of this surrogate gradient pathway, showing both the forward pass (main path and surrogate path) and the backward pass gradient flow.

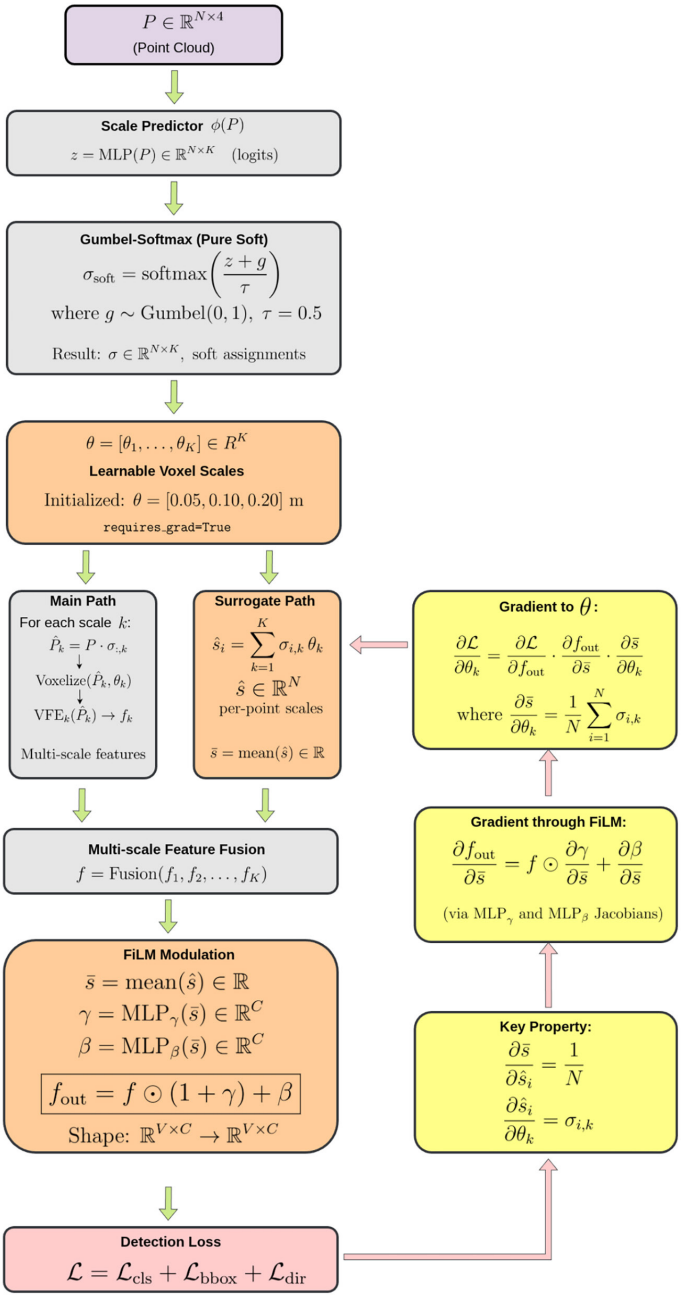

Fig. 2. Surrogate gradient pathway for learnable voxel scales. The forward pass shows the main processing path (left branch) where points are partitioned by soft assignments $\hat{P}_k = P \cdot \sigma_{:,k}$ and processed through scale-specific VFEs to produce multi-scale features f_k, alongside the surrogate path (center branch) computing per-point predicted scales $\hat{s}_i = \sum_{k=1}^{K} \sigma_{i,k}\theta_k \in R^N$ and mean scale $\bar{s} = mean(\hat{s})$. The Gumbel-Softmax operates in pure soft mode with $\sigma_{soft} = \text{softmax}((z+g)/\tau)$ where $g \sim Gumbel(0,1)$ and $\tau = 0.5$, producing soft assignments $\sigma \in R^{N\times K}$ that enable gradient flow while the main path uses discrete voxelization. Multi-scale features are fused via $f = \text{Fushion}(f_1, f_2, \ldots, f_K)$, and FiLM modulation conditions the output: $\gamma = MLP_{\gamma(\bar{s})} \in R^C$, $\beta = MLP_{\beta(\bar{s})} \in R^C$, yielding $f_{out} = f \odot (1+\gamma) +$ with shape $R^{V\times C} \to R^{V\times C}$. The backward pass (right, yellow boxes) shows gradient flow from the detection loss $L = L_{cls} + L_{bbox} + L_{dir}$ to the learnable scales θ via FiLM: $\frac{\partial f_{out}}{\partial \bar{s}} = f \odot \frac{\partial \gamma}{\partial \bar{s}} + \frac{\partial \beta}{\partial \bar{s}}$ (through MLP_γ and MLP_β Jacobians), with key properties $\frac{\partial \bar{s}}{\partial \hat{s}_i} = \frac{1}{N}$ and $\frac{\partial \hat{s}_i}{\partial \theta_k} = \sigma_{i,k}$, yielding $\frac{\partial L}{\partial \theta_k} = \frac{\partial L}{\partial f_{out}} \cdot \frac{\partial f_{out}}{\partial \bar{s}} \cdot \frac{\partial \bar{s}}{\partial \theta_k}$

The model is trained using a composite detection loss:

$$L_{det} = L_{cls} + \lambda_{reg} L_{reg} + \lambda_{dir} L_{dir}$$

where the focal loss is used for classification and the Smooth L1 loss for bounding box regression. Training uses the AdamW optimizer with a learning rate of 0.001 and weight decay of 0.01, cosine annealing learning rate schedule, batch size of 6, gradient clipping at norm 10, fixed temperature $\tau = 0.5$, initial learnable scales 0.05, 0.10, 0.20 m, and 10 training epochs.

Compared with fixed-scale voxelization, VoxAdapt introduces K parallel voxelization branches during training while preserving a single sparse backbone. ScaleNet adds $O(N K)$ computational complexity, where N denotes the number of points and K the number of scales, which is negligible compared to the backbone cost of $O(V \cdot C^2)$ where V is the voxel count and C is the feature dimension. In practice, VoxAdapt increases the parameter count by approximately 6.3%, FLOPs by approximately 23.6%, memory usage by approximately 6.9%, and inference latency by approximately 11.9% relative to single-scale baselines, while maintaining real-time performance suitable for embedded deployment.

At the algorithmic abstraction level, existing adaptive voxelization methods differ fundamentally from VoxAdapt. Dynamic voxelization allows variable point occupancy but treats the voxel scale as a fixed hyperparameter, excluding it from optimization. DSVT and multiscale voxel-based detectors process points at predefined resolutions and adapt feature aggregation or attention; however, the discretization geometry remains static. Naïve multiscale approaches use multiple fixed scales with uniform weighting, providing no learning signal for scale parameters. In contrast, VoxAdapt explicitly includes voxel-scale parameters in the optimization problem, allowing the detection loss to influence the discretization of space rather than merely how features are processed after discretization. FiLM modulation creates a direct gradient path from the detection loss to the learnable voxel scales, enabling the network to discover task-optimal discretization during training. We acknowledge that VoxAdapt relies on a surrogate gradient pathway rather than direct differentiation through voxel indexing and that we do not provide theoretical convergence or stability guarantees. Experiments were conducted using a SECOND backbone, and generalization to other detector families remains future work.

4 RESULTS

This section presents the experimental results obtained by evaluating VoxAdapt on the KITTI 3D Object Detection benchmark [28]. The KITTI dataset consists of LiDAR point clouds captured using a Velodyne HDL-64E sensor across urban, highway, and rural environments, exhibiting significant variations in point density. Points were densely sampled near the sensor and became increasingly sparse beyond approximately 40 m, making the dataset suitable for assessing voxelization strategies under realistic operating conditions.

We followed the standard KITTI train/validation split of 3,712 and 3,769 samples, respectively, and adopted a 10-epoch training schedule to evaluate the convergence behavior. All reported results correspond to single training runs. Although this enables direct observation of training stability and convergence trends, we acknowledge that variance analysis across multiple random seeds remains future work. The detection performance was evaluated using Average Precision (AP) with 40 recall

positions (R40) at an IoU threshold of 0.7 for cars and 0.5 for cyclists and pedestrians across three difficulty levels: easy (fully visible objects with height ≥40 pixels), moderate (partially occluded objects with height ≥25 pixels), and hard (heavily occluded objects with height ≥25 pixels). Unless otherwise stated, a moderate difficulty level was used for primary comparison. Experiments were conducted on cars, cyclists, and pedestrians to evaluate the performance across object categories with different spatial extents and LiDAR point densities commonly encountered in real-world driving environments.

All experiments were implemented using MMDetection3D [29] with PyTorch 2.0 and the SECOND backbone and were trained on a single NVIDIA RTX 4070 SUPER GPU (12GB) with automatic mixed precision enabled. Three configurations were compared. The Single-Scale configuration used standard SECOND with HardSimpleVFE and a single voxel grid of resolution [0.1, 0.1, 0.2] m. The Naïve Multi-Scale configuration used three parallel voxel grids with fixed XY resolutions {0.05, 0.10, 0.20} m while keeping the Z resolution fixed at 0.2 m; features from the three grids were combined using uniform one-third weighting without any learned assignment. VoxAdapt employed learnable voxel scales with adaptive XY resolutions optimized via a surrogate gradient pathway using ScaleNet and Gumbel-Softmax relaxation ($\tau = 0.5$), while maintaining the same fixed Z resolution of 0.2 m. All models were trained using AdamW (learning rate 0.001, weight decay 0.01), a cosine annealing learning rate schedule, batch size 6, and gradient clipping at norm 10.

Table 1 presents the results of car detection. All three configurations achieved comparable performance across difficulty levels. The naïve multi-scale baseline slightly outperformed the single-scale configuration, indicating that manually selected voxel scales are sufficient for large objects with dense LiDAR returns. VoxAdapt maintained competitive accuracy, demonstrating that adaptive voxelization does not negatively affect the detection performance in scenarios where fixed discretization is already effective.

Table 1. Car detection AP@0.70 (R40)

Method	Easy	Moderate	Hard	Mean
Single-Scale	86.47	77.96	73.63	79.35
Naïve Multi-Scale	88.26	78.36	73.67	80.10
VoxAdapt	88.12	77.24	71.85	79.07

The cyclist detection results are presented in Table 2. The single-scale configuration performed best across the difficulty levels. This behavior suggests that the relatively consistent geometry and moderate point density of cyclists reduce the need for adaptive scaling of voxels. VoxAdapt remained functional but did not provide additional benefits in this regime.

Table 2. Cyclist detection AP@0.50 (R40)

Method	Easy	Moderate	Hard	Mean
Single-Scale	86.65	80.14	78.34	81.71
Naïve Multi-Scale	88.86	85.20	78.34	85.91
VoxAdapt	80.72	74.55	73.00	76.09

The pedestrian detection results reported in Table 3 show qualitatively different behaviors that highlight the primary contribution of this work. Both the single-scale

and naïve multi-scale configurations failed to converge, producing zero AP across all difficulty levels. In contrast, VoxAdapt successfully recovered the detection performance, achieving stable and consistent results. This outcome demonstrates that multi-scale voxelization alone is insufficient when the scale selection is not learned. The naïve multiscale baseline uses the same fixed voxel scales {0.05, 0.10, 0.20} m as VoxAdapt but assigns points uniformly across scales without learning. Its complete failure—identical to the single-scale configuration—shows that the learned scale assignment mechanism, rather than the mere presence of multiple voxel grids, enables pedestrian detection.

Table 3. Pedestrian detection AP@0.50 (R40)

Method	Easy	Moderate	Hard	Mean
Single-Scale	0.00	0.00	0.00	0.00
Naïve Multi-Scale	0.00	0.00	0.00	0.00
VoxAdapt	**44.05**	**40.87**	**38.41**	**41.11**

To understand why fixed-scale methods fail for pedestrians while VoxAdapt succeeds, Table 4 reports the training-loss progression for pedestrian detection. All three configurations started with comparable loss values in the range of 4.3–4.7 at iteration 50, indicating similar initial optimization behavior. However, their training trajectories diverge substantially from each other. The Single-Scale configuration exhibited early and recurring numerical instability, with NaN losses appearing at iterations 100, 500, and 950. These failures arise from sparse gradient signals when pedestrians occupy only one or two voxels at a fixed 0.1 m XY resolution, leading to unstable optimization. The Naïve Multi-Scale configuration initially appeared more stable, converging to reasonable loss values through iteration 500, but ultimately experienced numerical instability at iteration 950. This indicates that multiple fixed voxel scales alone do not resolve the underlying sparsity problem, as a uniform one-third assignment still routes many points to voxel resolutions with inadequate point density. In contrast, VoxAdapt stabilizes after the initial training phase and maintains consistent convergence throughout the training by routing sparse pedestrian points to voxel resolutions that aggregate sufficient points per voxel.

Table 4. Training loss behavior for pedestrian detection

Iteration	Single-Scale	Naïve Multi-Scale	VoxAdapt	Iteration
50	4.29	4.66	4.41	50
100	NaN	2.80	2.92	100
150	2.49	2.43	2.49	150
500	NaN	2.10	2.05	500
950	NaN	NaN	1.77	950

Performance improvements occurred rapidly within the first two epochs and then stabilized, as shown in Table 5. The sharp improvement from epoch 1 to epoch 2 (15.89% → 39.61% Moderate AP) coincided with the rapid adjustment of learned voxel scales. During this phase, the network discovered that pedestrians required coarser voxelization than initially configured, shifting effective XY resolutions from

{0.05, 0.10, 0.20} m toward approximately {0.17, 0.23, 0.40} m to aggregate scattered points into voxels with sufficient density for stable optimization.

Table 5. Epoch-wise pedestrian AP@0.50 (R40) for VoxAdapt

Epoch	Easy	Moderate	Hard
1	17.34	15.89	14.86
2	44.40	39.61	36.47
5	40.09	37.13	34.93
9	44.41	40.75	37.86
10	44.05	40.87	38.41

Table 6 summarizes the computational requirements. Compared with single-scale voxelization, VoxAdapt added 6.3% parameters, 23.6% FLOPs, 6.9% memory usage, and 11.9% inference latency. Compared with naïve multi-scale voxelization, VoxAdapt was more efficient, reducing FLOPs by 18.7% and inference latency by 7.8%. This efficiency advantage arises because VoxAdapt selectively routes points to the appropriate voxel scales rather than processing all points uniformly at all resolutions.

Table 6. Computational overhead

Method	Parameters	FLOPs	Memory	Inference
Single-Scale	4.8M	12.3G	2.75 GB	42 ms
Naïve Multi-Scale	5.2M	18.7G	3.12 GB	51 ms
VoxAdapt	5.1M	15.2G	2.94 GB	47 ms

To complement the quantitative evaluation, Figures 1 and 2 present qualitative detection examples that illustrate VoxAdapt's performance on representative KITTI scenes.

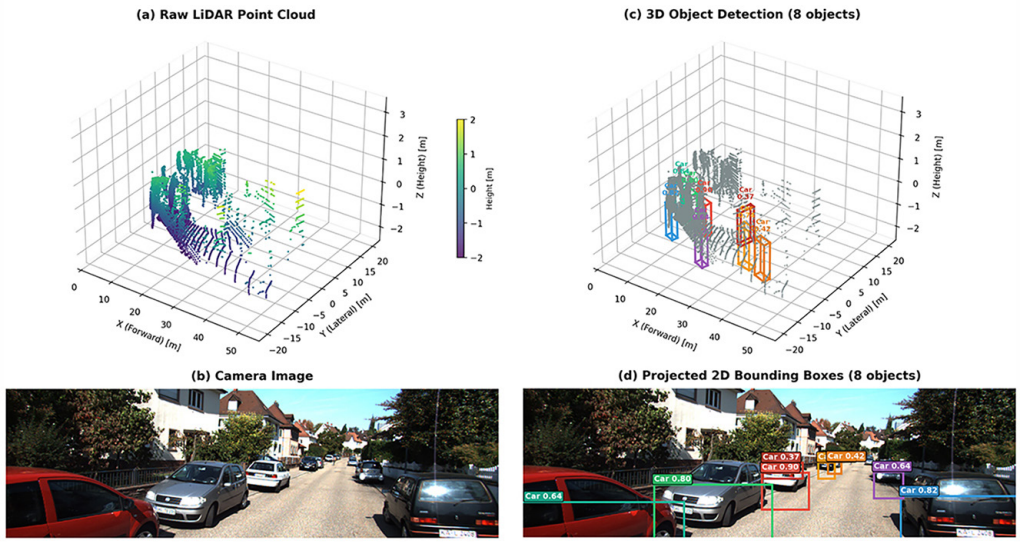

Fig. 3. Car detection results on a residential street scene: (a) Raw LiDAR point cloud colored by height showing the characteristic sparse-to-dense distribution with distance; (b) Corresponding camera image; (c) 3D object detection with 8 detected vehicles shown as oriented bounding boxes; (d) Projected 2D bounding boxes overlaid on the camera image with confidence scores ranging from 0.37 to 0.90

Figure 3 demonstrates VoxAdapt's car detection capability on a scene containing multiple parked vehicles at varying distances. The model successfully detected eight vehicles with confidence scores ranging from 0.37 to 0.90. Higher confidence scores (0.64–0.90) were observed for nearby vehicles where LiDAR point density was sufficient, while more distant or partially occluded vehicles received moderate confidence scores (0.37–0.54). The 3D bounding boxes accurately captured vehicle orientations, demonstrating robust localization even for vehicles positioned at oblique angles to the sensor.

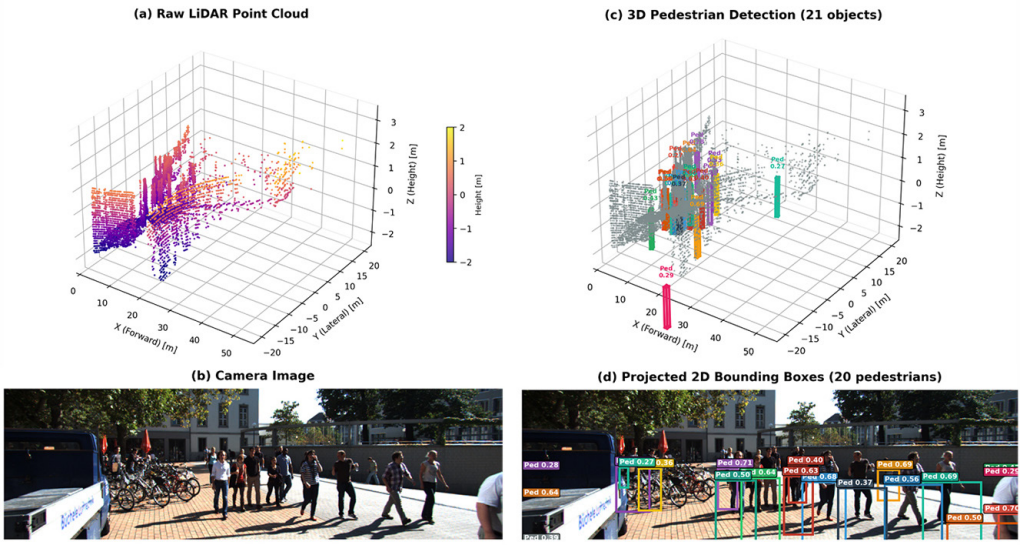

Fig. 4. Pedestrian detection results on a crowded urban scene: (a) Raw LiDAR point cloud showing sparse pedestrian signatures compared to background structures; (b) Corresponding camera image with a group of pedestrians; (c) 3D pedestrian detection with 21 detected objects; (d) Projected 2D bounding boxes showing 20 pedestrians with confidence scores ranging from 0.27 to 0.71

Figure 4 presents a challenging pedestrian detection scenario with a dense group of pedestrians at medium range (15–30 m). Despite the inherent sparsity of pedestrian LiDAR returns—typically only 5–15 points per person at this distance—VoxAdapt successfully detected 21 pedestrians in 3D space. The projected 2D bounding boxes show confidence scores ranging from 0.27 to 0.71, with higher confidence for pedestrians with clearer separation from the group. This result directly validates the quantitative findings: while single-scale and naïve multi-scale configurations fail entirely on pedestrian detection (0% AP), VoxAdapt's learned voxel scale assignment enables robust detection of these sparse, small objects. The contrast between Figures 1 and 2 illustrates the adaptive nature of VoxAdapt. For cars, the standard voxel resolution provides adequate point aggregation, resulting in high-confidence detections. For pedestrians, the learned coarser voxelization (approximately 0.2–0.4 m effective resolution) aggregates scattered points into voxels with sufficient density for stable feature extraction, enabling detection where fixed-scale methods fail.

The experimental results reveal a consistent pattern: adaptive voxelization provides the greatest benefit for small, sparsely sampled objects where fixed-scale methods fail entirely. Car detection shows minimal differentiation between methods because cars are large objects (approximately 3.9 × 1.6 × 1.56 m) that typically produce 30–360 LiDAR points at medium range (15–30 m), providing sufficient information for stable feature learning regardless of voxelization strategy. Cyclist detection favors single-scale voxelization because cyclists have consistent geometric

profiles and moderate point counts, reducing the need for adaptive scale selection. Pedestrian detection demonstrates the critical value of learned voxelization because pedestrians are small objects (approximately $0.8 \times 0.6 \times 1.7$ m) that produce 20–110 LiDAR points at medium range but as few as 10–25 points at far range (30–50 m); at fixed 0.1 m voxel resolution, these sparse points spread across the object volume can result in many voxels containing only one or two points, producing weak gradients that cause numerical instability, whereas VoxAdapt learns to use coarser scales (0.2–0.4 m) that aggregate scattered points into voxels with sufficient density for stable optimization.

The complete failure of the naïve multi-scale baseline—despite using the same voxel scales as VoxAdapt—provides strong evidence that learned scale assignment is the critical factor enabling robust pedestrian detection. Several limitations should be acknowledged. All experiments use KITTI, and cross-dataset generalization to nuScenes, Waymo, or other benchmarks requires future validation. While training dynamics are consistent across observations, formal variance analysis with multiple random seeds would strengthen statistical claims. Pedestrian detection shows 3–5% AP fluctuation between epochs (Table 5), suggesting room for more stable optimization strategies. VoxAdapt does not improve over baselines for larger objects, and future work could explore class-conditional scale learning or mixed strategies. Despite these limitations, the results demonstrate that VoxAdapt offers a practical and effective adaptive voxelization mechanism for real-world LiDAR-based perception systems operating under diverse geometric and environmental conditions.

5 CONCLUSION

We introduce VoxAdapt, a framework that treats voxel-scale values as learnable parameters updated via a surrogate gradient pathway jointly with detection network training. Unlike fixed-scale or naïve multi-scale approaches that adapt feature aggregation under fixed discretization, VoxAdapt allows voxel sizes to be adjusted during training in response to detection objectives—scale parameters receive gradient signals through FiLM modulation without requiring differentiation through discrete voxel indexing, with updated scales applied in subsequent forward passes.

The key conceptual contribution is demonstrating that geometric discretization parameters, traditionally treated as fixed preprocessing choices, can benefit from gradient-based optimization through surrogate pathways. This is particularly significant for sparse object detection, where fixed voxelization struggles but adaptive scales enable robust detection. VoxAdapt maintains real-time performance with minimal overhead, supporting deployment on resource-constrained platforms without manual sensor-specific tuning.

We acknowledge several limitations of this work: (1) VoxAdapt relies on surrogate gradients rather than true end-to-end differentiation through voxel indexing, providing approximate rather than exact gradient signals to scale parameters; (2) we do not provide theoretical convergence or stability guarantees for Gumbel–Softmax relaxation in sparse LiDAR settings; (3) evaluation is limited to the KITTI dataset and SECOND backbone—generalization to larger datasets (nuScenes, Waymo) and other detector families (PointPillars, CenterPoint, transformer-based architectures) remains future work; (4) reported results represent single training runs without variance analysis across random seeds; and (5) visualization of learned scale distributions, which could provide additional insight into ScaleNet's spatial assignment patterns, is not included in this work.

Two key research questions remain: whether the learnable voxelization paradigm generalizes to geometric discretization choices beyond scale, such as point cloud range boundaries, anchor configurations, or pillar dimensions, and how learned scale distributions evolve with increasing dataset complexity, including whether pre-trained scale parameters can accelerate adaptation to novel sensor configurations.

Future work will focus on extending VoxAdapt to enhance user interaction with portable devices such as smartphones, drones, wearable devices, and augmented reality systems. These platforms increasingly incorporate compact LiDAR sensors for applications including indoor navigation, accessibility assistance, and interactive gaming yet face severe computational and memory constraints. By enabling automatic adaptation of voxel scales to sensor characteristics and scene complexity, VoxAdapt can eliminate the need for manual tuning when deploying across diverse mobile hardware configurations. We plan to investigate lightweight variants of ScaleNet optimized for edge inference, integration with model compression techniques such as quantization and pruning, and real-time scale adaptation for dynamic environments encountered in pedestrian navigation and human-robot interaction. Additionally, we aim to evaluate VoxAdapt on emerging datasets captured by mobile-grade LiDAR sensors, which exhibit different point density distributions compared to automotive-grade systems. These directions will enable more intuitive and responsive 3D perception on handheld and wearable devices, improving user experiences in augmented reality, assisted navigation, and interactive spatial computing applications.

6 ACKNOWLEDGEMENTS

This study was supported in part by the University of Moratuwa, Sri Lanka, through the Senate Research Committee (SRC) Conference and Publishing Grant.

7 REFERENCES

[1] Y. Zhou and O. Tuzel, "VoxelNet: End-to-end learning for point cloud based 3D object detection," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. https://doi.org/10.1109/CVPR.2018.00472

[2] Y. Yan, Y. Mao, and B. Li, "SECOND: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018. https://doi.org/10.3390/s18103337

[3] H. Kuang, B. Wang, J. An, M. Zhang, and Z. Zhang, "Voxel-FPN: Multi-scale voxel feature aggregation for 3D object detection from LIDAR point clouds," *Sensors*, vol. 20, no. 3, p. 704, 2020. https://doi.org/10.3390/s20030704

[4] M. Ye, S. Xu, and T. Cao, "HVNet: Hybrid voxel network for LiDAR based 3D object detection," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. https://doi.org/10.1109/CVPR42600.2020.00170

[5] J. Noh, S. Lee, and B. Ham, "HVPR: Hybrid voxel-point representation for single-stage 3D object detection," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2021*, 2021. https://doi.org/10.1109/CVPR46437.2021.01437

[6] T. Jiang, N. Song, H. Liu, R. Yin, Y. Gong, and J. Yao, "VIC-Net: Voxelization information compensation network for point cloud 3D object detection," in *IEEE International Conference on Robotics and Automation (ICRA 2021)*, Xi'an, China, 2021. https://doi.org/10.1109/ICRA48506.2021.9561597

[7] J. Hu and G. Jin, "An intelligent framework for english teaching through deep learning and reinforcement learning with interactive mobile technology," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 18, no. 9, pp. 74–87, 2024. https://doi.org/10.3991/ijim.v18i09.49289

[8] J. Mao *et al.*, "Voxel transformer for 3D object detection," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021. https://doi.org/10.1109/ICCV48922.2021.00315

[9] L. Fan *et al.*, "Embracing single stride 3D object detector with sparse transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2022. https://doi.org/10.1109/CVPR52688.2022.00827

[10] F. Cao *et al.*, "MCHFormer: A multi-cross hybrid former of point-image for 3D object detection," *IEEE Transactions on Intelligent Vehicles*, vol. 9, no. 1, pp. 383–394, 2024. https://doi.org/10.1109/TIV.2023.3323518

[11] A. Diego *et al.*, "Mobile application for continuous recognition and classification of sign language images through deep learning," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 19, no. 7, pp. 4–21, 2025. https://doi.org/10.3991/ijim.v19i07.52853

[12] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "FiLM: Visual reasoning with a general conditioning layer," in *AAAI Conference on Artificial Intelligence*, 2018. https://doi.org/10.1609/aaai.v32i1.11671

[13] J. Praveenchandar, S. Vinoth Kumar, A. Christopher Paul, M. A. Mukunthan, and K. Maharajan, "Deep learning algorithms in mobile edge with real-time abnormal event detection for 5G-IoT devices," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 17, no. 17, pp. 59–71, 2023. https://doi.org/10.3991/ijim.v17i17.42805

[14] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. https://doi.org/10.1109/CVPR.2019.01298

[15] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. https://doi.org/10.1109/CVPR.2017.16

[16] A. Geiger, P. Lenz, and R. Urtasun, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[17] S. Shi, X. Wang, and H. Li, "PointRCNN: 3D object proposal generation and detection from point cloud," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. https://doi.org/10.1109/CVPR.2019.00086

[18] Z. Yang, Y. Sun, S. Liu, and J. Jia, "3DSSD: Point-based 3D single stage object detector," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. https://doi.org/10.1109/CVPR42600.2020.01105

[19] C. Chen, Z. Chen, J. Zhang, and D. Tao, "SASA: Semantics-augmented set abstraction for point-based 3D object detection," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2022. https://doi.org/10.1609/aaai.v36i1.19897

[20] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, "Multi-view 3D object detection network for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017. https://doi.org/10.1109/CVPR.2017.691

[21] S. Liu, W. Huang, Y. Cao, D. Li, and S. Chen, "SMS-Net: Sparse multi-scale voxel feature aggregation network for LiDAR-based 3D object detection," *Neurocomputing*, vol. 501, pp. 555–565, 2022. https://doi.org/10.1016/j.neucom.2022.06.054

[22] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. https://doi.org/10.1109/CVPR.2017.106

[23] Q. Xie *et al.*, "MLCVNet: Multi-level context VoteNet for 3D object detection," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) 2020*, 2020. https://doi.org/10.1109/CVPR42600.2020.01046

[24] H. Wang *et al.*, "DSVT: Dynamic sparse voxel transformer with rotated sets," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. https://doi.org/10.1109/CVPR52729.2023.01299

[25] F. Koltun and V. Yu, "Multi-scale context aggregation by dilated convolutions," in *International Conference on Learning Representations (ICLR)*, San Juan, Puerto Rico, 2016.

[26] Y. Li, Y. Chen, N. Wang, and Z.-X. Zhang, "Scale-aware trident networks for object detection," in *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. https://doi.org/10.1109/ICCV.2019.00615

[27] F. Shuang, H. Huang, Y. Li, R. Qu, and P. Li, "AFE-RCNN: Adaptive feature enhancement RCNN for 3D object detection," *Remote Sensing*, vol. 14, no. 5, p. 1176, 2022. https://doi.org/10.3390/rs14051176

[28] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, RI, USA, 2012, pp. 3354–3361. https://doi.org/10.1109/CVPR.2012.6248074

[29] OpenMMLab, "MMDetection3D: OpenMMLab next-generation platform for general 3D object detection," 2020. [Online]. Available: https://github.com/open-mmlab/mmdetection3d

8 AUTHORS

Daham Pathiraja is a researcher at the Department of Computer Science and Engineering, University of Moratuwa, Sri Lanka. His research interests include 3D computer vision, deep learning for point cloud processing, adaptive voxelization, real-time 3D object detection, and efficient artificial intelligence models for edge and resource-constrained environments (E-mail: daham.23@cse.mrt.ac.lk).

Prof. Indika Perera is a Professor at the Department of Computer Science and Engineering, University of Moratuwa, Sri Lanka. His research interests include software engineering, software architecture and design, enterprise software systems, software process and management, and human–computer interaction, with a focus on software engineering processes for AI-based enterprise systems (E-mail: indika@cse.mrt.ac.lk).