

---

# vFusedSeg3D

---

**Osama Amjad**

VisionRD

Pakistan

osama.amjad@visionrdai.com

**Ammad Nadeem**

VisionRD

Pakistan

ammad@visionrdai.com

## Abstract

In this technical study, we introduce VFusedSeg3D, an innovative multi-modal fusion system created by the VisionRD team that combines camera and LiDAR data to significantly enhance the accuracy of 3D perception. VFusedSeg3D uses the rich semantic content of the camera pictures and the accurate depth sensing of LiDAR to generate a strong and comprehensive environmental understanding, addressing the constraints inherent in each modality. Through a carefully thought-out network architecture that aligns and merges these information at different stages, our novel feature fusion technique combines geometric features from LiDAR point clouds with semantic features from camera images. With the use of multi-modality techniques, performance has significantly improved, yielding a state-of-the-art mIoU of 72.46% on the validation set as opposed to the prior 70.51%. VFusedSeg3D sets a new benchmark in 3D segmentation accuracy, making it an ideal solution for applications requiring precise environmental perception.

## 1 Introduction

vFusedSeg3D is a multi-model architecture that used both lidar point cloud and camera images to fuse features and output enhanced feature map that can be used to predict segmentation maps. Our work is inspired by Li et al. [2023], that also uses multimodalities to fuse camera and lidar features.

## 2 Architecture

Our architecture uses dual-modal feature extraction to take advantage of the capabilities of both image and LiDAR data, resulting in robust feature representation.

The architecture consists of two parallel feature extraction streams. We utilized DLA34, Yu et al. [2018] as the image side backbone for feature extraction due to its hierarchical feature aggregation and multiple levels of residual blocks for multi scale feature extraction. The Waymo Open Dataset sample includes a point cloud with 64 beams and 5 RGB camera images from the following viewpoints: front, front-left, front-right, side-left, and side-right. These 5 images are utilized in their entirety. After resizing for model input, the images have a resolution of 960x640 (width x height).

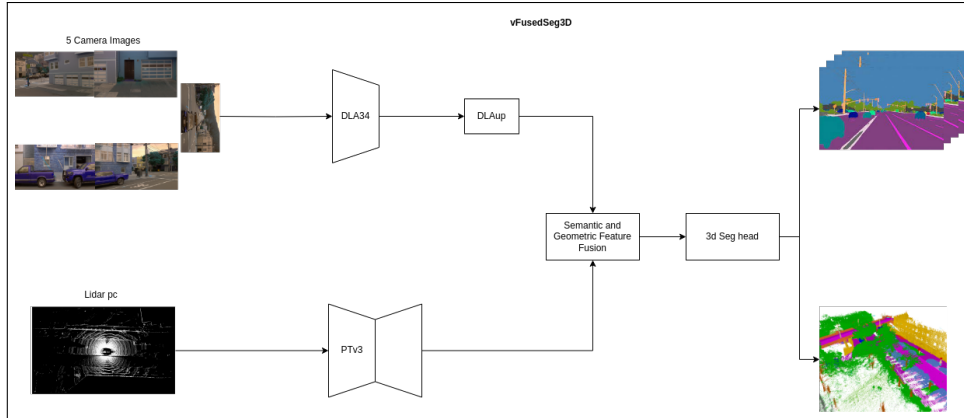


Figure 1: vFusedSeg3d architecture

The image encoder generates feature maps with dimensions of 16, 32, 64, 128, 256, and 512 feature channels. To facilitate detailed image feature extraction, a simple DLAup (Deep Layer Aggregation upsampling) module is employed as the neck of the image-side feature extraction. This module leverages residual connections to enhance the feature maps. This map is critical for subsequent fusion operations because this enable comprehensive scene knowledge by combining image-based information.

We used Point Transform v3 (PTv3), Wu et al. [2024] as the key feature extraction backbone on the LiDAR side. PTv3 is useful in processing 3D point cloud data, extracting spatial features that encapsulate the geometric complexities of the environment.

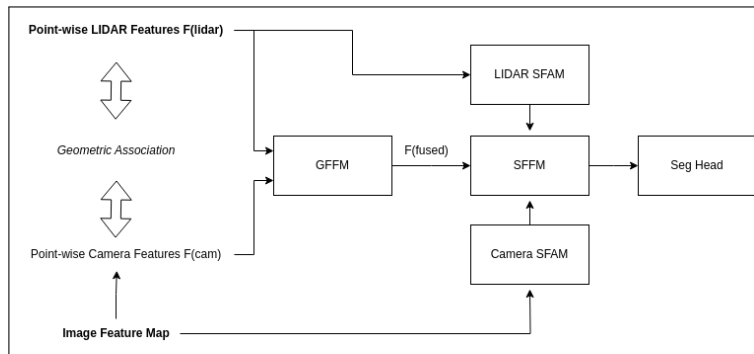


Figure 2: Lidar and camera feature fusion: Geometric Feature Fusion Module and Semantic Feature Fusion Modules as designed by Li et al. [2023] but with some modifications

Following feature extraction as shown in , both image and LiDAR feature maps go through a fusion process to combine semantic and geometric data. The fusion process is based on a modified version of MSeg3D’s feature fusion module that was designed specifically for this architecture. Point-wise LIDAR features and point-wise camera features are concatenated together subsequent learnable fusion using a Geometry-based Feature Fusion Module (GFFM) resulting out geometric fused features. Lidar and Camera features are aggregated using there respectiv Semantic Feature Aggregation Module (SFAM). Instead of voxel Features as used by MSeg3d, we utilized point-wise lidar features to be aggregated using LIDAR SFAM. These aggregated features are used as the input of the the Semantic Feature Fusion Module (SFFM), which has been significantly improved, with new layers added to optimise the integration of semantic cues across modalities. This modified fusion technique

allows for a more precise and detailed interpretation of the combined data, resulting in better performance in the following analytical tasks.

### 3 Training Strategy

Due to computational resource constraints, we trained the model sequentially and in segments. We initially approached this difficulty by training the LiDAR and image feature extraction components separately. This method allowed us to focus resources effectively, ensuring that each component was optimized to its full potential without overwhelming our hardware capabilities.

For the LiDAR (lidarBase) side, we trained the PTV3 model for 45 epochs with a batch size of 2. The training process spanned approximately 6 to 7 days, culminating in a highest achievable accuracy of 70.51% mIoU. In the original paper by Wu et al. [2024], the reported accuracy is around 71.3% mIoU.

The observed decrease in accuracy is primarily attributable to differences in grid sampling resolution and low batch size. The original study employed a finer grid sampling of 0.05 and batch size of 12 with multi-GPU training, whereas we used a coarser grid sampling of 0.1, due to limitations in our computational resources, and batch size of 2 with single GPU. Increasing the grid resolution beyond this value resulted in Out of Memory (OOM) errors, thus constraining our ability to achieve the higher accuracy reported in the original paper. Image side was trained for 10 epochs with batch size of 8.

Once the individual LiDAR and image models were trained and their parameters fine-tuned, we froze these models to conserve resources. This freezing process prevented any further modifications to the weights of the feature extraction models during the subsequent training phases, thereby reducing the computational load. This fusion model was trained for 25 epochs with batch size of 2. We did not use mini-batching for gradient accumulation, although it is a good idea, but due to time limitation for more experimentation, we trained our models on 2 batch size.

Subsequently, we directed our resources towards training the fusion model, which integrates the pre-trained and frozen image and LiDAR feature maps. This staged training approach not only addressed our resource limitations but also ensured that the integration layer—responsible for merging semantic and geometric features—received the dedicated attention needed for effective learning.

By compartmentalizing the training process in this manner, we managed to circumvent the limitations imposed by our hardware resources, allowing each part of the network to be meticulously trained and thereby enhancing the overall performance and efficiency of the model. This structured training strategy proved crucial in developing a robust multi-modal fusion system capable of high-accuracy performance in real-world applications.

Model	mIoU (%)
Our LidarBase	70.51
PTv3 Wu et al. [2024]	71.3
vFusedSeg3D	72.46

Table 1: Model Performance and mIoU

## 4 Final Results on VAL set

As demonstrated in Table 1, the model performance metrics are summarized based on their mean Intersection over Union (mIoU) percentages.

## 5 Training Hyper parameters

Training parameters used for model training are given in the Table 2.

Parameter	Value
<b>Augmentations</b>	<b>Lidar Side:</b> Global rotation around the Z-axis ( $[-\pi/4, +\pi/4]$ ), Global translation ( $\Delta x, \Delta y, \Delta z$ with $N(0, 0.5)$ ), Global scaling ( $[0.95, 1.05]$ ). <b>Image Side:</b> Scaling ( $[1.0, 1.5]$ ), Horizontal rotation ( $[-1^\circ, 1^\circ]$ ), Random cropping (size $H_{in}, W_{in}$ ), Color jitter (brightness, contrast, saturation, hue with parameters 0.3, 0.3, 0.3, 0.1), JPEG compression ratio ( $[30, 70]$ with probability 0.5).
<b>Image mean/std</b>	mean = $[0.40789654, 0.44719302, 0.47026115]$ , std = $[0.28863828, 0.27408164, 0.27809835]$
<b>Lidar enc Input Features</b>	5 (x, y, z, intensity, elongation)
<b>Point Cloud Serialization</b>	["z", "z-trans", "hilbert", "hilbert-trans"]
<b>Channels</b>	<b>Lidar Enc:</b> (32, 64, 128, 256, 512) <b>Lidar Dec:</b> (64, 64, 128, 256) <b>Image Enc:</b> (32, 64, 128, 256, 512) <b>Image Neck:</b> (32, 64, 128, 256)
<b>Point Cloud Range</b>	(-75.2, -75.2, -4, 75.2, 75.2, 2)
<b>Image Size</b>	(960, 640) (width, height)
<b>Optimizer</b>	AdamW
<b>Scheduler</b>	Cosine
<b>Criteria</b>	CrossEntropy, Lovasz
<b>Learning Rates</b>	Main LR: 8e-4 Block LR: 8e-5
<b>Weight Decay</b>	5e-2
<b>Batch Size</b>	2
<b>Epochs</b>	50

Table 2: Training Hyperparameters for vFusedSeg3D

## 6 Test Time Augmentations (TTA)

We only utilized Test Time Augmentations on lidar point clouds to boost our accuracies as mentioned on Table 3

Augmentation	Description
Global Scaling ( $\tau_{\text{scale}}$ )	A random scaling factor in $[0.95, 1.05]$ .
Random Flipping ( $\tau_{\text{flip}}$ )	Random flipping along the X, Y axis.
Global Rotation ( $\tau_{\text{rot}}$ )	Rotation around the Z axis with a random angle in $[-\frac{\pi}{4}, +\frac{\pi}{4}]$ .
Global Translation ( $\tau_{\text{tran}}$ )	Translation with a random vector $(\Delta x, \Delta y, \Delta z)$ .

Table 3: Test Time Augmentations (TTAs)

## 7 Experimental Setup

All experiments and training sessions were conducted on a single NVIDIA RTX 3090 graphics card, equipped with 24GB of VRAM. The GPU was paired with 32GB of CPU RAM.

### References

- Jiale Li, Hang Dai, Hao Han, and Yong Ding. Mseg3d: Multi-modal 3d semantic segmentation for autonomous driving. In *CVPR*, pages 21694–21704, 2023.
- Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler, faster, stronger. In *CVPR*, 2024.
- Fisher Yu, Dequan Wang, Evan Shelhamer, and Trevor Darrell. Deep layer aggregation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2403–2412, 2018.