# TransFusion: Multi-modal Fusion Network for Semantic Segmentation

Abhisek Maiti
a.maiti@utwente.nl

Sander Oude Elberink
s.j.oudeelberink@utwente.nl

George Vosselman
george.vosselman@utwente.nl

University of Twente, The Netherlands

## Abstract

*The complementary properties of 2D color images and 3D point clouds can potentially improve semantic segmentation compared to using uni-modal data. Multi-modal data fusion is however challenging due to the heterogeneity, dimensionality of the data, the difficulty of aligning different modalities to the same reference frame, and the presence of modality-specific bias. In this regard, we propose a new model, TransFusion, for semantic segmentation that fuses images directly with point clouds without the need for lossy pre-processing of the point clouds. TransFusion outperforms the baseline FCN model that uses images with depth maps. Compared to the baseline, our method improved mIoU by $4\%$ and $2\%$ for the Vaihingen and Potsdam datasets. We demonstrate the capability of our proposed model to adequately learn the spatial and structural information resulting in better inference.*

## 1. Introduction

The utilization of multimodal data has become increasingly vital in the field of Earth Observation due to advancements in sensing technologies. The increasing availability of remote sensing data, such as high-resolution imagery and point clouds, has made it possible to gather a wealth of information about a scene in spectral, textural, and 3D domains. The fusion of these two forms of data potentially results in a more complete and accurate scene representation, as each form of data provides unique information that complements the other.

Semantic segmentation is a key task in computer vision and image analysis, where the goal is to assign semantic labels to each pixel in an image. This involves dividing an image into multiple segments, each corresponding to a specific class. As multi-modal data potentially provides a better representation of the scene, semantic segmentation tasks may benefit from the appropriate use of multi-modal data when available.

For 3D point cloud semantic segmentation, the input features are typically tied to the point cloud, with the segmentation task being performed on the point cloud itself, resulting in a segmented 3D point cloud. If a 2D segmentation label is needed, the segmented point cloud can be projected onto a plane. Alternatively, a planar representation, such as Digital Surface Model (DSM), can be generated from the point cloud in the pre-processing step to be fused with the respective image [40]. This allows for 2D semantic segmentation using a common convolutional neural network [10]. On the other hand, the point cloud and corresponding image-like features can be fused in the 3D feature space [13], transformed into a grid-shaped 3D feature space using voxelization [50], and then segmented in 2D through dimensionality reduction using a suitable model [1]. Therefore in both cases, subsequent models cannot harness the full potential of the information present in the point clouds. SPLATNet demonstrated a novel approach to seamlessly learn joint 2D-3D features from image and point cloud, respectively. This approach performs the learning on a higher dimensional lattice generated from each modality through interpolation analogous to voxelization.

This research addresses the aforementioned issues by introducing a novel 2D semantic segmentation architecture to fuse point clouds and images directly. We adapt Transformer and FCN-based networks for the fusion. Therefore, we refer to this proposed network as TransFusion. Our contributions through TransFusion are as follows:

- TransFusion does not require any lossy pre-processing of the point cloud to generate 3D voxels or 2D projections
- TransFusion accepts point clouds irrespective of spatial sparsity or variable point density
- TransFusion has no theoretical bounds regarding the number of points per sample
- TransFusion allows seamless mapping between 3D and 2D feature space while being end-to-end learnable.

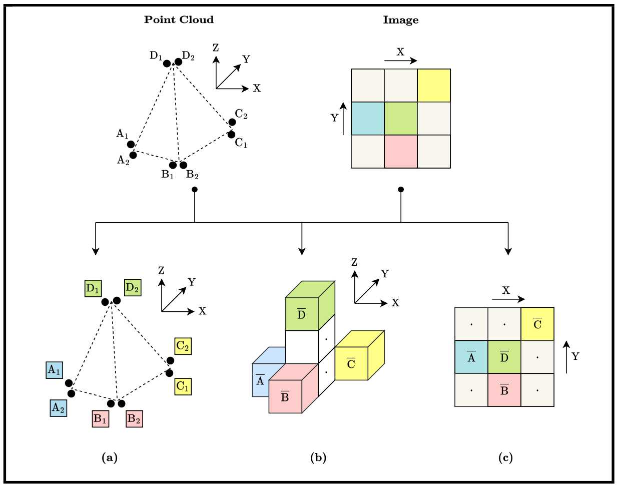


Figure 1. Important 2D-3D joint representations of point clouds and image. (a) Point cloud with projected image features, (b) Voxel with overlayed image features, (c) Depth map or surface model stacked combined with image

## 2. Related Works

This section will discuss joint 2D-3D feature representations in relation to mono-images and point clouds and the corresponding research. This section will be concluded with a brief discussion of the current state-of-the-art semantic segmentation methods leveraging the direct fusion of point clouds and mono-imagery. Deep learning based point cloud and image fusion architectures generally adopt one or more of the three pronounced representations shown in Fig. 1 for depicting joint 2D-3D features.

**Point Representation:** This type of representation is common for architectures focusing on point-wise 3D semantic segmentation or 3D object detection tasks. In many cases, data is available where the image is already projected on 3D points, i.e., each point in the point cloud has corresponding RGB values. Examples of this type of data are [6, 12, 20]. Many point cloud segmentation architectures are designed to work with these point clouds [9, 49]. Models like [3, 35], originally designed to learn point features, can be further modified to work with these colored point clouds with [18]. However, some networks with 3D vision tasks are designed to take input point clouds and images separately in their original form [47] and then project 2D features into 3D feature space for feature fusion and subsequent prediction in 3D [42].

Point clouds have a few other specialized representations, such as index or tree-based representations, graph-based representations, etc. [8]. A colored point cloud or point cloud with projected image features preserves all of its original properties. These alternative representations can also be generated from such point clouds [8]. Therefore networks designed for such representations apply to colored point clouds [48].

**Voxelized Representation:** Voxels are a fixed-resolution volumetric representation of point clouds. This representation preserves coarser 3D structural information but loses fine-grained spatial and geometrical information [8]. Its regular grided structure makes this representation compatible with standard 3D convolutions. The most straightforward way to merge the voxelized point cloud and image is by adding the image features as extra channels to the voxel [38]. The grided structure is also convenient for transforming 3D features into 2D features [50], and vice versa [31] for easy feature-level fusion. Due to these reasons, voxel has been a popular choice for 3D and 2D-3D joint learning tasks [22, 25, 34]. A major issue with voxels is that the voxel's space is very sparse by nature, which leads to unnecessary computation if special care is not taken [19].

**Planar Representation:** The creation of 2D views from a point cloud involves projecting the cloud onto 2D planes and processing the resulting images with standard convolutions and pooling layers. This approach solves permutation and translation issues but loses 3D geometry information and struggles with point-wise label predictions [8]. In the past, these planar representations of point clouds have been generated in various ways depending on the context and the task [8]. Ma et al. [27] presented a network that leverages sparse depth with images. Some networks generate multiple sparse planar views at different depths for better learning and efficient computation [33]. However, a popular choice is to generate a dense 2D representation with pixel-wise depths. This way, the depth image can be stacked with the image as an additional channel, and any 2D Fully Convolutional Networks (FCN) like [4, 5] can subsequently perform segmentation [10]. Moreover, in the field of earth observation, it is fairly common to generate digital surface models (DSM) and digital elevation models (DEM) from various sources, including point clouds [30]. This makes combining images and depth features more organic in the context of remote sensing [23].

**Direct Fusion:** The integration of point cloud data with mono-image is a challenging process. Point clouds have an irregular, orderless, continuous structure, while images are discrete, ordered, and projective. To the best of our knowledge, SPLATNet [39] is the only recent seminal work that has successfully fused raw point cloud and image in an end-to-end manner for both 3D and 2D semantic segmentation. In the first step, the raw point cloud and image are separately interpolated to a respective permutohedral lattice using barycentric interpolation. Subsequently, spatial-aware learning is performed on the lattices using higher dimensional convolution with learnable kernels. Both point cloud features and image futures are fused in higher dimensional space. Finally, depending upon the task, segmentation labels are generated by projecting higher dimensional fused features into the respective dimensional space using barycentric interpolation [39]. This network outperformed various SOTA 3D segmentation networks at the time. This network's joint 2D-3D semantic segmentation capability has been tested on the RueMonge dataset [36]. SPLATNet demonstrated a novel approach to seamlessly learn joint 2D-3D features from image and point cloud. However, this method still involves interpolation, which is conceptually very similar to voxelization. Moreover, a density normalization [39] step is applied on the point clouds in order to deal with variable point density. Although it is better than discarding points through volumetric sampling, this process is still lossy and potentially reduces finer structural information.

**Multi-modal learning and Transformer:** Perceiver [17] is a deep learning model based on the Transformer architecture that can handle multiple modalities. It uses an asymmetric attention mechanism to iteratively distill inputs into a compact representation, allowing it to handle large inputs. It makes a few architectural assumptions about the input data, making it more versatile and applicable to a broader range of tasks.

The PerceiverIO [16] architecture is an extension of the Perceiver [17]. Perceiver has flexibility regarding the inputs. PerceiverIO adds the same level of flexibility and generalizability to the outputs. These properties of the PerceiverIO make it highly suitable for our direct point cloud and image fusion task.

## 3. Proposed Method

We aim to design a unified 2D semantic segmentation model capable of directly fusing point clouds and mono-images. Our model is agnostic to the source of the point cloud, such as LiDAR or stereo matching. However, in this work, we will focus on fusing aerial imagery and corresponding point cloud pairs. We adopt a feature-level fusion of the modalities with a late fusion strategy. First, the model derives from each modality separately in their respective feature extractor branches. These features are subsequently fused for final dense prediction using a segmentation head. We will refer to our proposed fusion network as TransFusion. The comprehensive architecture of TransFusion is presented in Fig. 2.

**Image Branch:** For dense feature extraction from images, we use an off-the-shelf FCN backbone. Over the years, FCNs have proven highly effective for dense prediction tasks. Current SOTA transfomer based image segmentation networks are relatively computationally expensive [46]. Moreover, recent research has demonstrated that computationally expensive attention layers might not provide significant additional benefits for various vision tasks, including segmentations [7, 41]. We adopt a typical encoder-decoder FCN network for image feature extraction. Here we use a lightweight ResNet [11] backbone as the encoder and DeepLabV3+ [5] without the final prediction head as our decoder. The encoder is responsible for extracting features from the input image.

**Point Cloud Branch:** The Design of our point cloud branch is inspired by PerceiverIO [16]. However, unlike PerceiverIO, we do not combine multi-modal data with variable embedding padded with learnable modality vectors. There are two main reasons for this. First, modality learning puts an additional burden on the model to learn and infer the source modality of each sample. Secondly, concatenating the inputs from the different modalities vastly increases the effective number of samples fed into the initial transformer block. Considering the $\mathcal{O}(n^2)$ complexity of the attention layers, this can get prohibitively expensive to compute for our use cases. Here we leverage the unique ability of the transformers to query higher dimensional latent to predict features in lower dimensions. Unlike PerceiverIO, we use separate branches for each modality eliminating the need for modality learning. We transform the point cloud $\mathcal{P} \in \mathbb{R}^{N \times C}$ to latent space $\mathcal{Z} \in \mathbb{R}^{A \times B}$. Subsequently, $n$ transformer blocks are applied on $\mathcal{Z}$ to obtain more refined latent features $\hat{\mathcal{Z}}$. Finally, a cross-attention is applied to query $\hat{\mathcal{Z}} \in \mathbb{R}^{A' \times B'}$ to predict features $\hat{\mathcal{P}} \in \mathbb{R}^{M \times G}$ at the dense pixel locations. Relative 2D pixel coordinates are encoded using the same positional encoding scheme as the point cloud and use the encoded coordinates as the query $(\mathcal{X})$ for this cross-attention module. Here, $N$ is the number of points in the point cloud, and $C$ represents each point's feature vector size. $A$, $A'$, $B$, and $B'$ are properties of the model which control the size of the latent spaces. $M$ represents the total number of pixels in the corresponding image, and $G$ is the desired feature size. The parameter $n$ indicates the number of attention blocks to apply on the initial latent space sequentially. Thus the purpose of the point cloud branch is to derive point cloud features at each pixel location of the corresponding image.

**Feature Fusion:** The purpose of this module is to fuse the features generated by each feature extractor branch of the respective modality. Our approach is to first refine the features from each modality with the weights derived from the other modality and then fuse them. In this regard, popular activation functions have been used in various studies for context modeling of the image features [2,44]. We adopt the principle of context modeling and use softmax to derive modality refinement weights. Initially, the $M \times G$ features

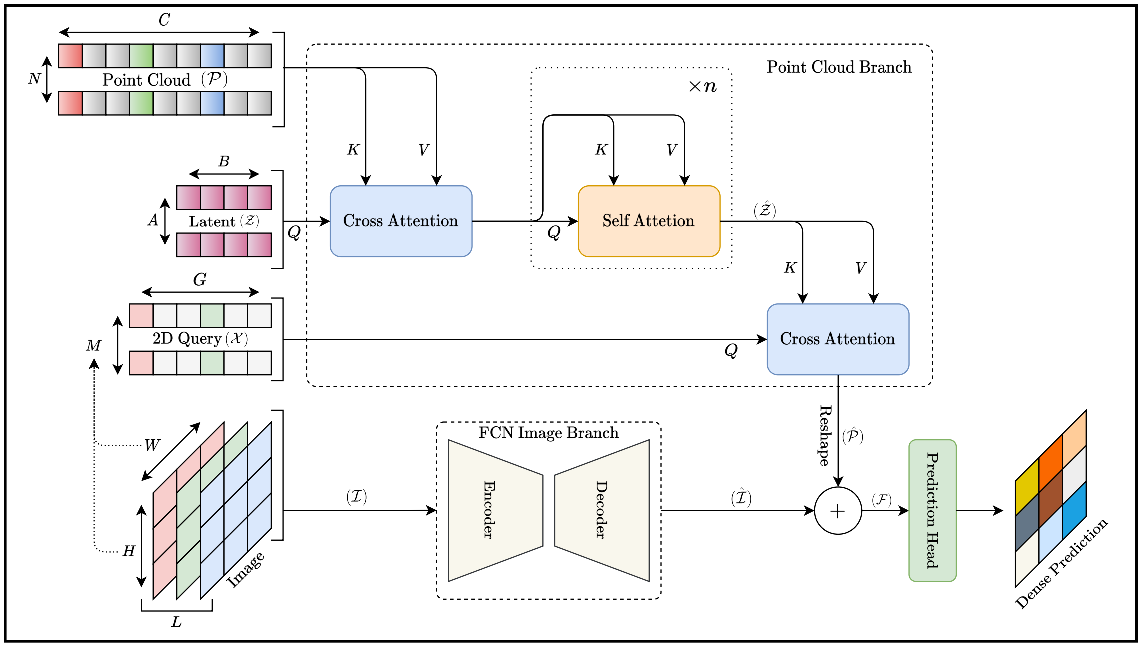


Figure 2. Overview of the proposed model. The architecture consists of two branches. The image branch includes an FCN network, and the point cloud branch comprises an attention-based architecture. Here the [+] represents the feature fusion module.

from the point cloud are reshaped into $H \times W \times G$ for dimensional compatibility with the image features ($\hat{\mathcal{I}}$) of the same shape. The image height and width are $H$ and $W$.

$$\mathcal{F} = \hat{\mathcal{I}} \cdot \sigma(\hat{\mathcal{P}}) + \hat{\mathcal{P}} \cdot \sigma(\hat{\mathcal{I}}) \tag{1}$$

We fuse the features as shown in Eq. (1), where $[\cdot]$ represents Hadamard product, and $\sigma$ denotes softmax operation along the feature dimension. Here the Hadamard product with softmax weights from the other modality refines the respective features before being summed to generate the fused features. Finally, for dense prediction, a residual layer with $1 \times 1$ convolution is applied on the fused feature $\mathcal{F}$.

## 4. Experiments

We design the experiments to compare our proposed fusion method with conventional methods of predicting 2D dense semantic labels using mono-images and depth maps. We train the proposed model with images and raw point clouds as inputs. We do not use the nDSMs in our fusion method. However, the baseline FCN segmentation model is trained on images that include nDSMs. We use the DeepLabV3+ as our baseline and FCN image branch of our fusion model. Finally, we compare the results of the proposed model with the baseline performance. In this section, we first introduce the benchmark datasets we will use for the experiments. Subsequently, we present the data preparation and model training strategies.

### 4.1. Datasets

For all of our experiments, we use two open benchmark datasets for 2D semantic segmentation released by ISPRS. These datasets are the Potsdam dataset [14] and the Vaihingen dataset [15]. Both datasets contain very high-resolution airborne images, DSMs, and point clouds.

**Potsdam Dataset:** The Potsdam dataset [14] contains ortho-rectified imagery. Dense urban features typically dominate scenes. Each image has a near-infrared (NIR) band and three visible bands Red-Green-Blue (RGB). Corresponding point clouds generated from stereo-matching and derived normalized DSM (nDSM) are available with the data. The semantic labels are available as images where each class is represented with a unique value. The images, nDSMs, and labels have a spatial resolution of 5 cm. There are 38 tiles of $6000 \times 6000$ images with corresponding labels. We use the officially designated 24 images set for training. We evaluate our models on 14 tiles designated for the test in the initial dataset release.

**Vaihingen Dataset** The Vaihingen dataset contains 33 tiles of different sizes with labels. This dataset also contains predominantly urban scenes. However, the images have 3 bands NIR-R-G with an approximate spatial resolution of 8 cm. The corresponding point clouds are acquired using airborne LiDAR. Associated nDSMs derived from these point clouds are also available with the data. Like the Potsdam dataset, we follow the officially designated training set of 16 images for model training and a designated test set of 17 for model validation.

### 4.2. Data Preparation

Although our method does not require any projection or aggregation, the point clouds must be prepared appropriately before passing them into the model. For both datasets, point clouds are available with absolute heights. Therefore we first classify the points into the ground and non-ground points using the SMRF filter [32]. Then we normalize the point cloud by deducting the height of the ground points, preserving the relative height above ground. We consider points within a quarter of the image pixel distance from each other as duplicates and remove them to reduce unnecessary computations.

Transformers are order-agnostic. Therefore appropriate positional encoding of the inputs is required to preserve the spatial relationships. First, the 3D coordinates are normalized in the closed range of $[-1, 1]$. we apply the Fourier feature positional encoding similar to [29]. Then frequency is linearly varied from the lowest sampling frequency to the Nyquist sampling frequency. We also concatenate the abso-lution position with the Fourier encoding to obtain the final positional encoding. The positional encoding ($\mathbf{X}$) of a single dimension is shown in Eq. (2). Here, $x$ is the absolute position. Lowest and Nyquist frequencies are denoted by $f_0$ and $f_n$, respectively.

$$\mathbf{X} = [x, \sin{(2\pi f_0 x)}, \cos{(2\pi f_0 x)}, \cdots, \cos{(2\pi f_n x)}] \quad (2)$$

We encode each dimension of the point cloud and concatenate them to obtain a combined positional embedding. Considering the compatibility with images, we use the dimensions of the image patches as the sampling frequency in their respective dimensions. We pre-calculate the global maximum for the Z - axis and use that as our sampling frequency. Subsequently, Nyquist frequencies are calculated as half of the respective sampling frequencies following the sampling theory.

We use the image patch size of $512 \times 512$ for all experiments. For the 2D query, we first extract the relative pixel coordinates from the images and normalize them in the $[-1, 1]$ range. We then apply the same positional encoding on the normalized pixel coordinates.

### 4.3. Implementation Details

For our experiments, we use 6 self-attention blocks in the point cloud branch to refine the latent space. Following a series of preliminary experiments, we set the number of latents to 512 with 480 latent channels. For all multi-head attention modules, we use 4 heads. We do not use feature widening using the MLP layers. The number of channels of the features generated by the point cloud branch is set to 64. For regularization, we set the dropout rate to 0.2. We initialize the latent space gaussian distribution with 0 mean and 0.02 standard deviation.

We use ResNet-34 as the feature extractor for our DeepLabv3+ based image branch. We do not use pre-trained weights and train the models from scratch for each experiment. We adjust the number of input channels accepted by the image branch depending on the number of channels present in the images of the respective dataset. We use the same model used in the image branch as our baseline. However, unlike the baseline, the final segmentation head is not used in TransFusion.

### 4.4. Training Strategy

The proposed model contains multiple input branches where each branch deals with a specific modality. In practice, it is challenging to train such models appropriately with a conventional end-to-end training strategy. Therefore we adopt a two-step training strategy. First, for a few epochs, we only train the point cloud branch on the point clouds as an auto-encoder with an ad-hoc decoder. Subsequently, we train the entire model end-to-end, mapping images and point clouds to the 2D labels. Since the learning progression of CNNs and transformers are quite different, we employ two different optimizers for each branch catering to their individual learning. We use AdaBound optimizer [26] for the image branch and LAMB optimizer [45] with a low initial learning rate for the point cloud branch.

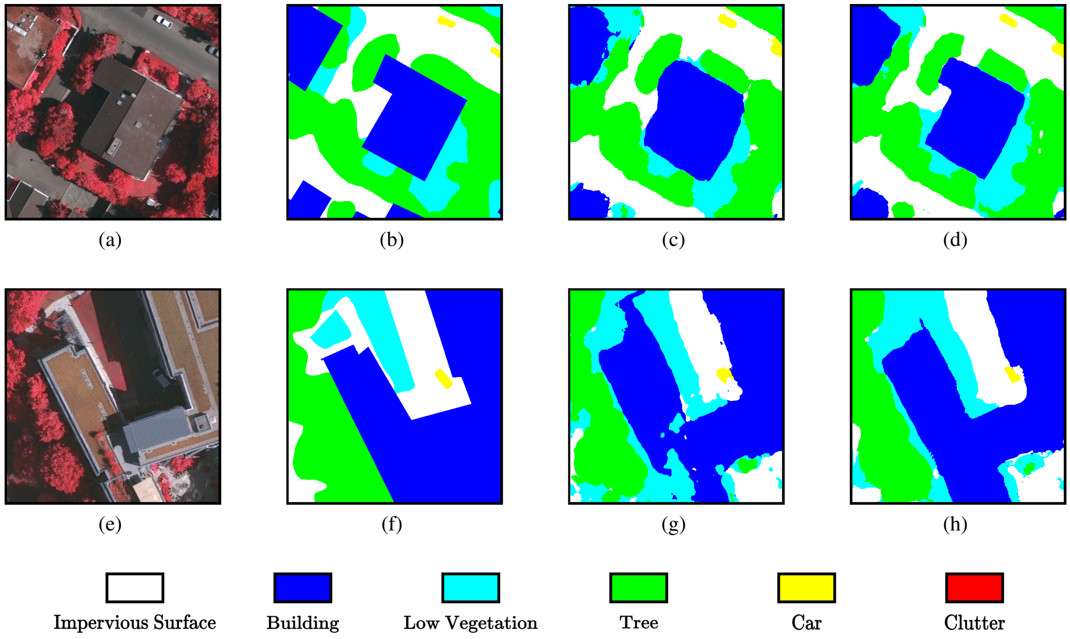
For the baseline, we follow the normal training strategy. For fairness, we use the same AdaBound optimizer [26] applied on the image branch of our proposed network. We use OneCycleLR [37] learning rate scheduling strategy in all the experiments. To find the approximate optimal initial learning rate, we use a fast grid search with exponentially varying the learning rate. We use Stochastic Weight Averaging for the last few epochs during the training to attain better generalizability.

## 5. Results and Discussion

| | Surf. | Bld | Veg. | Tree | Car | mIoU |
|---|---|---|---|---|---|---|
| Baseline | 73.70 | 81.44 | 56.75 | 72.27 | 57.16 | 68.26 |
| Ours | **80.47** | **87.74** | **63.22** | **73.51** | **58.46** | **72.68** |

Table 1. Metrics from the Vaihingen experiment. Here class wise IoUs are reported along with the mIoU. The best values are marked in bold. **Surf.**: Impervious Surface, **Bld**: Building, **Veg.**: Low Vegetation.

**Quantitative Analysis:** We use mean intersection-over-union (mIoU) as the overall indicator of performance. Additionally, we compare the quantitative results using class-


(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 3. Qualitative comparison of a pair of images from the Vaihingen experiment. (a) and (e) represent images of two different subregions, wherein (b) and (f) are the respective ground truths (labels). (c) and (g) depict the baseline prediction, whereas (d) and (h) are the predictions from our proposed method, TransFusion.

wise IoU. We start our evaluation by analyzing the metrics from the Vaihigen experiments shown in Tab. 1.

Our proposed method improves the mIoU by more than 4% over the baseline. Significant improvements in IoU are observed in all the classes. The highest improvement of 6.76% in IoU is observed in the Impervious Surface class followed by Building, Low Vegetation classes with more than 6% improvement in respective IoUs. The IoU of the Tree and Cars class improved slightly more than 1%, which is the lowest among all the classes.

| | Surf. | Bld | Veg. | Tree | Cars | mIoU |
|---|---|---|---|---|---|---|
| Baseline | 78.92 | 87.11 | 68.12 | 70.03 | 74.49 | 75.73 |
| Ours | **79.01** | **89.18** | **69.38** | **71.29** | **78.51** | **77.47** |

Table 2. Metrics from the Potsdam experiment. Here class wise IoUs are reported along with the mIoU. The best values are marked in bold. **Surf.**: Impervious Surface, **Bld**: Building, **Veg.**: Low Vegetation.

We also observe overall and class-wise improvements in the IoU score for the Potsdam dataset in Tab. 2. In this case, we observe approximately 2% improvement in the mIoU. The improvement in IoU is also evident in each of the classes. The highest IoU improvement of approximately 4% is observed for the car class, followed by the building class with 2% improvement in the IoU. The least improvement is observed for the Impervious Surface class. The IoU improvement of approximately 1% is observed for both The Low Vegetation and Tree classes.

**Qualitative Analysis:** we found that classified images with and without fusion using our proposed framework resemble well with the original labels. The fusion of point clouds and optical imagery improved class prediction compared to the baseline with optical bands with nDSM. The improvements in quantitative metrics are well reflected in the model predictions. In the case of Vaihingen, we observe that our proposed model better preserves building shapes and edges compared to the baseline. Despite learning from the nDSMs, the baseline tends to prioritize visual features resulting in misclassifications in the presence of rooftop gardens. Our proposed model does not suffer from such an issue. This also emphasizes the added benefits of learning from the point cloud along with the image features. Similar improvements to the building inferences can also be observed in the Potsdam dataset. Generally, the building shapes and edges from the fusion model closely resemble the ground truth compared to the baseline. We observe significantly fewer visual artifacts from the fusion model than the baseline. Significant improvement of IoU in the car class is well reflected in the Figs. 4e to 4h.

Evidently, point clouds are comparatively more beneficial than interpolated depth maps. We suspect this can be attributed to two main reasons. Primarily, the structural information point cloud in the vertical direction is lost due to

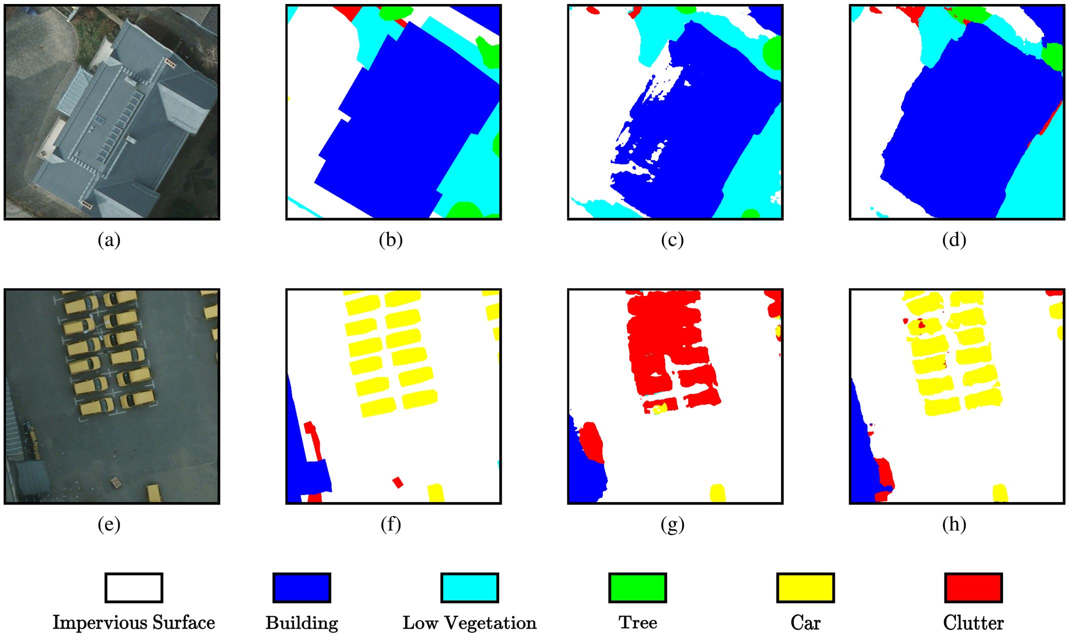


Figure 4. Qualitative comparison of a pair of images from Potsdam experiment. (a) and (e) represent images of two different subregions, Here (b) and (f) are the respective ground truths (labels). (c) and (g) depict the baseline prediction, and (d) and (h) are the predictions from our proposed method, TransFusion.

planar projection. Secondly, applying spatial interpolation to eliminate sparsity from the depth maps introduces additional uncertainty and noise. In contrast, our fusion model leverages the complete information of the point clouds without synthetic data and associated noise.

It is worth noting that the overall improvement observed in the Potsdam dataset is lower than in the Vaihingen dataset. We suspect that the main reason is the source of the point clouds. The point clouds in Potsdam dataset are generated using stereo matching. However, the point cloud in the Vaihingen dataset was acquired using airborne LiDAR with potentially better vertical accuracy and comparatively less noise. The points generated from stereo matching represent mainly the outer surfaces of objects such as trees and low vegetation, whereas the LiDAR point clouds capture the 3D nature of those objects better. The superiority of the LiDAR point cloud has been observed in past studies [28].

**Point Cloud Feature Learning:** We investigate the learning capability of the point cloud branch of our fusion model. We generate the saliency maps from the features generated by the points cloud branch. The saliency maps for two patches of the Vaihingen dataset are shown in Figs. 5e and 5f. We also show nadir views of the corresponding point clouds in the Figs. 5c and 5d respectively. We observe that the point cloud branch generally pays more attention to the regions with high elevation. The planar and linear structures are quite prominent in the saliency maps. In both scenes, waterbodies are present on the left side of the tiles. There are no points present in these large regions. However, the model learns the spatial relationship despite the apparently missing information. There are missing points corresponding to a building Fig. 5c, possibly due to due to surface characteristics of the roof material. The predicted boundary of that building from TransFusion is overlayed in black on Fig. 5e. Although the point cloud branch fails to highlight the missing building like the other buildings in the saliency maps the features from the image branch complement this resulting in a relatively precise delineation of the building footprint. This dynamic feature importance can be attributed to our feature fusion module.

## 6. Ablation Study

We carried out various ablation studies to test the robustness of our proposed fusion model. First, we compared our fusion module with two common approaches, addition and concatenation. Tab. 3 shows the observations from this experiment. We use six self-attention layers in the point cloud branch for this experiment.

First, we vary the number of self-attention layers to determine the optimal number. We iteratively increase the number of self-attention blocks and measure the respective performance in terms of mIoU. Our observations are presented in Tab. 4. Interestingly, the model performance does not monotonically increase with the number of layers. This can be potentially attributed to depth inefficiency of self-

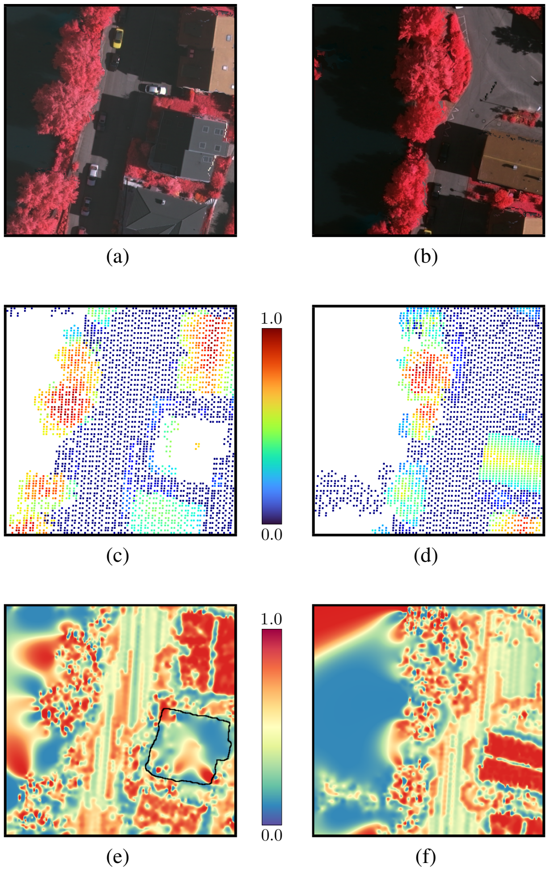


Figure 5. Learning capability of the point cloud branch. (a) and (b) represent images of two different subregions. (c) and (d) are the nadir view of respective point clouds. The corresponding saliency maps are depicted in (e) and (f). The boundary of the missing building in (c) is denoted in black in (e). This is extracted from the prediction of TransFusion.

| Fusion | Vaihingen | Potsdam |
|---|---|---|
| $\hat{\mathcal{I}} + \hat{\mathcal{P}}$ | 69.03 | 73.21 |
| $\hat{\mathcal{I}} \mid \hat{\mathcal{P}}$ | 70.14 | 73.57 |
| Ours | **72.68** | **77.47** |

Table 3. Performance of TransFusion with respect to different feature fusion strategies. The best score is highlighted in bold. $\hat{\mathcal{I}}, \hat{\mathcal{P}}$ are image feature and point cloud feature respectively. Element-wise addition and concatenation along feature dimension are respectively denoted by $+$, $|$ symbols.

attention [21], subject to further research.

In our experiments, we observed that our model overfits quickly if the dataset is relatively small. This observation is in accordance with previous studies concerning the training of transformer based networks on small datasets [24, 43]. In the case of the Vaihingen dataset, this issue is particularly noticeable. Data augmentation during model training partly mitigates this. We used common pixel-level transforms along with various spatial transforms for image augmentation during the training of the baseline. Along with augmentation, we addressed the issue of overfitting by pre-training the point cloud branch and subsequently using an adequate warm start strategy during the joint training, as previously discussed in the training strategy section.

| Layers | Vaihingen | Potsdam |
|---|---|---|
| 1 | 67.12 | 71.27 |
| 2 | 69.43 | 69.89 |
| 3 | 68.89 | 75.51 |
| 4 | 70.92 | 76.83 |
| 5 | 71.19 | **77.47** |
| 6 | **72.68** | <u>77.39</u> |
| 7 | <u>72.17</u> | 73.32 |

Table 4. Performance of TransFusion with respect to a varying number of self-attention layers. The best score is in bold, and the next best score is underlined

## 7. Conclusion

In this study, we introduce a novel architecture to fuse images directly with point clouds and perform semantic segmentation in the 2D domain. One of the key advantages of our proposed model is that it does not require pre-processing of point clouds to make elevation maps. Thus the initial data will not be subject to lossy projection or interpolation. Our proposed model evidently outperforms the FCN baseline, which performs segmentation on images combined with depth maps. Our experiments also highlight that, for our proposed model, LiDAR point clouds are more beneficial than those generated from dense matching. As part of our model performance, the mIoU is improved by 4% and 2% for the Vaihingen and Potsdam datasets, respectively. Particularly, our method improved the IoUs more than 6% for the Impervious Surface, Building, and Low Vegetation classes in the Vaihingen data. In the Potsdam dataset, we see an improvement of 4% and 2% for the Car and Building classes, respectively. There are two main reasons for such improvements: a) structural information is fully retained without any loss in the vertical direction during planar projection, and b) no spatial interpolation is applied in point clouds. In addition, the saliency maps demonstrate that our point cloud branch adequately learns spatial and structural information, thus significantly helping in better inference. Compared to the common feature fusion schemes, the proposed feature fusion strategy adopts cross-modality feature refinement, which contributes toward better prediction.

## References

[1] Inigo Alonso, Luis Riazuelo, Luis Montesano, and Ana C. Murillo. 3D-MiniNet: Learning a 2D representation from point clouds for fast and efficient 3D LiDAR semantic segmentation. *IEEE Robotics and Automation Letters*, 5(4):5432–5439, Oct. 2020. 1

[2] Yue Cao, Jiarui Xu, Stephen Lin, Fangyun Wei, and Han Hu. Global context networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2020. 3

[3] R. Qi Charles, Hao Su, Mo Kaichun, and Leonidas J. Guibas. PointNet: Deep learning on point sets for 3D classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, July 2017. 2

[4] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation, 2017. 2

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation, 2018. 2, 3

[6] Meida Chen, Qingyong Hu, Zifan Yu, Hugues Thomas, Andrew Feng, Yu Hou, Kyle McCullough, Fengbo Ren, and Lucio Soibelman. STPLS3D: A large-scale synthetic and real aerial photogrammetry 3D point cloud dataset. In *33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022*. BMVA Press, 2022. 2

[7] Shoufa Chen, Enze Xie, Chongjian GE, Runjian Chen, Ding Liang, and Ping Luo. CycleMLP: A MLP-like architecture for dense prediction. In *International Conference on Learning Representations*, 2022. 3

[8] Yaodong Cui, Ren Chen, Wenbo Chu, Long Chen, Daxin Tian, Ying Li, and Dongpu Cao. Deep learning for image and point cloud fusion in autonomous driving: A review. *IEEE Transactions on Intelligent Transportation Systems*, 23(2):722–739, Feb. 2022. 2

[9] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Niessner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, July 2017. 2

[10] Foivos I. Diakogiannis, François Waldner, Peter Caccetta, and Chen Wu. ResUNet-a: A deep learning framework for semantic segmentation of remotely sensed data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 162:94–114, Apr. 2020. 1, 2

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2016. 3

[12] Qingyong Hu, Bo Yang, Sheikh Khalid, Wen Xiao, Niki Trigoni, and Andrew Markham. Towards semantic segmentation of urban-scale 3D point clouds: A dataset, benchmarks and challenges. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 2

[13] Sumin Hu, Seungwon Song, and Hyun Myung. Image projection onto flat LiDAR point cloud surfaces to create dense and smooth 3D color maps. In *2020 17th International Conference on Ubiquitous Robots (UR)*. IEEE, June 2020. 1

[14] ISPRS. 2D Semantic Labeling Dataset - Potsdam. https://www.isprs.org/education/benchmarks/UrbanSemLab/2d-sem-label-potsdam.aspx, 2016. 4

[15] ISPRS. 2D Semantic Labeling Dataset - Vaihingen. https://www.isprs.org/education/benchmarks/UrbanSemLab/2d-sem-label-vaihingen.aspx, 2016. 4

[16] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier J Henaff, Matthew Botvinick, Andrew Zisserman, Oriol Vinyals, and Joao Carreira. Perceiver IO: A general architecture for structured inputs & outputs. In *International Conference on Learning Representations*, 2022. 3

[17] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4651–4664. PMLR, 18–24 Jul 2021. 3

[18] Mingyang Jiang, Yiran Wu, Tianqi Zhao, Zelin Zhao, and Cewu Lu. Pointsift: A sift-like network module for 3D point cloud semantic segmentation, 2018. 2

[19] Georg Krispel, Michael Opitz, Georg Waltner, Horst Possegger, and Horst Bischof. FuseSeg: LiDAR point cloud segmentation fusing multi-modal data. In *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, Mar. 2020. 2

[20] Michael Kölle, Dominik Laupheimer, Stefan Schmohl, Norbert Haala, Franz Rottensteiner, Jan Dirk Wegner, and Hugo Ledoux. The Hessigheim 3D (H3D) benchmark on semantic segmentation of high-resolution 3D point clouds and textured meshes from uav lidar and multi-view-stereo. *ISPRS Open Journal of Photogrammetry and Remote Sensing*, 1:11, 2021. 2

[21] Yoav Levine, Noam Wies, Or Sharir, Hofit Bata, and Amnon Shashua. Limits to depth efficiencies of self-attention. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 22640–22651. Curran Associates, Inc., 2020. 8

[22] Ying Li, Lingfei Ma, Zilong Zhong, Dongpu Cao, and Jonathan Li. TGNet: Geometric graph CNN on 3-D point cloud segmentation. *IEEE Transactions on Geoscience and Remote Sensing*, 58(5):3588–3600, 2019. 2

[23] Yansong Liu, Sankaranarayanan Piramanayagam, Sildomar T. Monteiro, and Eli Saber. Dense semantic labeling of very-high-resolution aerial imagery and LiDAR with fully-convolutional neural networks and higher-order CRFs. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, July 2017. 2

[24] Yahui Liu, Enver Sangineto, Wei Bi, Nicu Sebe, Bruno Lepri, and Marco De Nadai. Efficient training of visual transformers with small datasets. In A. Beygelzimer, Y. Dauphin,

P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. 8

[25] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-Voxel CNN for efficient 3D deep learning. In *Advances in Neural Information Processing Systems*, 2019. 2

[26] Liangchen Luo, Yuanhao Xiong, Yan Liu, and Xu Sun. Adaptive gradient methods with dynamic bound of learning rate, 2019. 5

[27] Fangchang Ma and Sertac Karaman. Sparse-to-Dense: Depth prediction from sparse depth samples and a single image. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2018. 2

[28] Evangelos Maltezos, Athanasia Kyrkou, and Charalabos Ioannidis. LiDAR vs dense image matching point clouds in complex urban scenes. In Kyriacos Themistocleous, Diofantos G. Hadjimitsis, Silas Michaelides, and Giorgos Papadavid, editors, *Fourth International Conference on Remote Sensing and Geoinformation of the Environment (RSCy2016)*. SPIE, Aug. 2016. 7

[29] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 5

[30] Xuran Pan, Lianru Gao, Bing Zhang, Fan Yang, and Wenzhi Liao. High-resolution aerial imagery semantic labeling with dense pyramid network. *Sensors*, 18(11):3774, Nov. 2018. 2

[31] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. ENet: A deep neural network architecture for real-time semantic segmentation, 2016. 2

[32] Thomas J. Pingel, Keith C. Clarke, and William A. McBride. An improved simple morphological filter for the terrain classification of airborne LiDAR data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 77:21–30, Mar. 2013. 5

[33] Charles R. Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J. Guibas. Frustum PointNets for 3D object detection from RGB-D data. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, June 2018. 2

[34] Charles R. Qi, Hao Su, Matthias NieBner, Angela Dai, Mengyuan Yan, and Leonidas J. Guibas. Volumetric and multi-view CNNs for object classification on 3D data. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2016. 2

[35] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint arXiv:1706.02413*, 2017. 2

[36] Hayko Riemenschneider, András Bódis-Szomorú, Julien Weissenberg, and Luc Van Gool. Learning where to classify in multi-view semantic segmentation. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 516–532, Cham, 2014. Springer International Publishing. 3

[37] Leslie N. Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates, 2017. 5

[38] Shuran Song and Jianxiong Xiao. Sliding shapes for 3D object detection in depth images. In *Computer Vision – ECCV 2014*, pages 634–651. Springer International Publishing, 2014. 2

[39] Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. SPLATNet: Sparse lattice networks for point cloud processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2530–2539, 2018. 2, 3

[40] F. Tarsha-Kurdi, T. Landes, and P. Grussenmeyer. Joint combination of point cloud and DSM for 3D building reconstruction using airborne laser scanner data. In *2007 Urban Remote Sensing Joint Event*. IEEE, Apr. 2007. 1

[41] Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. MLP-Mixer: An all-MLP architecture for vision, 2021. 3

[42] Zhixin Wang and Kui Jia. Frustum ConvNet: Sliding frustums to aggregate local point-wise features for amodal 3D object detection. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Nov. 2019. 2

[43] Peng Xu, Dhruv Kumar, Wei Yang, Wenjie Zi, Keyi Tang, Chenyang Huang, Jackie Chi Kit Cheung, Simon J. D. Prince, and Yanshuai Cao. Optimizing deeper transformers on small datasets, 2020. 8

[44] Lingxiao Yang, Ru-Yuan Zhang, Lida Li, and Xiaohua Xie. SimAM: A simple, parameter-free attention module for convolutional neural networks. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11863–11874. PMLR, 18–24 Jul 2021. 3

[45] Yang You, Jing Li, Sashank Reddi, Jonathan Hseu, Sanjiv Kumar, Srinadh Bhojanapalli, Xiaodan Song, James Demmel, Kurt Keutzer, and Cho-Jui Hsieh. Large batch optimization for deep learning: Training bert in 76 minutes, 2019. 5

[46] Bowen Zhang, Zhi Tian, Quan Tang, Xiangxiang Chu, Xiaolin Wei, Chunhua Shen, and Yifan liu. Segvit: Semantic segmentation with plain vision transformers. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 3

[47] Rui Zhang, Guangyun Li, Minglei Li, and Li Wang. Fusion of images and point clouds for the semantic segmentation of large-scale 3D scenes based on deep learning. *ISPRS Journal of Photogrammetry and Remote Sensing*, 143:85–96, Sept. 2018. 2

[48] Haoran Zhou, Yidan Feng, Mingsheng Fang, Mingqiang Wei, Jing Qin, and Tong Lu. Adaptive graph convolution for point cloud analysis. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, Oct. 2021. 2

[49] Wei Zhou, Xin Cao, Xiaodan Zhang, Xingxing Hao, Dekui Wang, and Ying He. Multi point-voxel convolution (mpvconv) for deep learning on point clouds, 2021. 2

[50] Yin Zhou and Oncel Tuzel. VoxelNet: End-to-end learning for point cloud based 3D object detection. In *2018*

*IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, June 2018. 1, 2