remote sensing

MDPI

Article

GeoSparseNet: A Multi-Source Geometry-Aware CNN for Urban Scene Analysis

Muhammad Kamran Afzal [1,2,3], Weiquan Liu [1,2,*], Yu Zang [2], Shuting Chen [4], Hafiz Muhammad Rehan Afzal [5,6], Jibril Muhammad Adam [2], Bai Yang [3], Jonathan Li [7] and Cheng Wang [2]

[1] College of Computer Engineering, Jimei University, Xiamen 361021, China
[2] Fujian Key Laboratory of Sensing and Computing for Smart Cities, School of Informatics, Xiamen University, Xiamen 361005, China; kamran@stu.xmu.edu.cn (M.K.A.); zangyu7@xmu.edu.cn (Y.Z.); 23020170155980@stu.xmu.edu.cn (J.M.A.); cwang@xmu.edu.cn (C.W.)
[3] Center for Integrative Conservation & Yunnan Key Laboratory for Conservation of Tropical Rainforests and Asian Elephants, Xishuangbanna Tropical Botanical Garden, Chinese Academy of Sciences, Mengla 666303, China; kamran@xtbg.ac.cn (M.K.A.); baiyang@xtbg.ac.cn (B.Y.)
[4] Mathematics and Digital Science School, Chengyi College, Jimei University, Xiamen 361021, China; chenst2016@jmu.edu.cn
[5] The School of Life Sciences, Northwestern Polytechnical University, Xi'an 710072, China; rehan@nwpu.edu.cn (H.M.R.A.)
[6] School of Electrical Engineering and Computing, University of Newcastle, Callaghan, NSW 2308, Australia; c3249813@uon.edu.au (H.M.R.A.)
[7] Departments of Geography and Environmental Management and Systems Design Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada; junli@uwaterloo.ca
* Correspondence: wqliu@jmu.edu.cn

Abstract: The convolutional neural networks (CNNs) functioning on geometric learning for the urban large-scale 3D meshes are indispensable because of their substantial, complex, and deformed shape constitutions. To address this issue, we proposed a novel Geometry-Aware Multi-Source Sparse-Attention CNN (GeoSparseNet) for the urban large-scale triangular mesh classification task. GeoSparseNet leverages the non-uniformity of 3D meshes to depict both broad flat areas and finely detailed features by adopting the multi-scale convolutional kernels. By operating on the mesh edges to prepare for subsequent convolutions, our method exploits the inherent geodesic connections by utilizing the Large Kernel Attention (LKA) based Pooling and Unpooling layers to maintain the shape topology for accurate classification predictions. Learning which edges in a mesh face to collapse, GeoSparseNet establishes a task-oriented process where the network highlights and enhances crucial features while eliminating unnecessary ones. Compared to previous methods, our innovative approach outperforms them significantly by directly processing extensive 3D mesh data, resulting in more discerning feature maps. We achieved an accuracy rate of 87.5% when testing on an urban large-scale model dataset of the Australian city of Adelaide.

Keywords: deep learning; 3D meshes; urban-scale; remote sensing; Geometry-Aware; attention

check for updates

Citation: Afzal, M.K.; Liu, W.; Zang, Y.; Chen, S.; Afzal, H.M.R.; Adam, J.M.; Yang, B.; Li, J.; Wang, C. GeoSparseNet: A Multi-Source Geometry-Aware CNN for Urban Scene Analysis. *Remote Sens.* **2024**, *16*, 1827. https://doi.org/10.3390/rs16111827

Academic Editor: Massimiliano Pepe

Received: 7 March 2024
Revised: 13 April 2024
Accepted: 22 April 2024
Published: 21 May 2024

1. Introduction

The 3D scene or shape analysis of a concoction of dense, complex, curved surfaces, or irregular geometries of meshes in large urban environments including buildings, trees, cars, and other elements has been made possible by recent developments in 3D computer vision and photogrammetry [1]. To enable the use of these meshes in a variety of applications, such as smart urban planning, navigation systems, virtual reality, radiation estimation, noise modeling, and photovoltaic perspective, it is essential to extract semantic information from the mesh models [2,3].

For semantic classification of 3D mesh data, a trivial number of machine learning-based algorithms are available; and their primary focus is to process 3D point clouds [4,5].

A few recent studies, such as [6,7], also deal with deep learning for surface meshes, but they are only able to analyze single items or small inside scenes (such as a living room or kitchen). The Multi-View Stereo (MVS) modeling pipeline has limitations, making it difficult to recreate or classify these glass facades without causing geometric distortion and deformation. Classifying models seem extremely absurd and cluttered due to the deformed glass facades [8].

Textured urban meshes offer distinct advantages compared to point clouds and MVS modeling pipelines in classification. They excel in providing detailed surface representation, incorporating rich visual information through applied textures on geometric structures. Unlike point clouds, which comprise individual data points, textured meshes merge geometric data with surface textures. This fusion enables precise and robust classification in urban environments, capturing intricate details and enhancing feature recognition across various objects and structures within the scene [3]. This comprehensive representation arises from textured urban meshes' capability to integrate both geometric data and high-resolution textures. Unlike point clouds that solely capture spatial coordinates, meshes combine this information with surface textures, delivering a more realistic portrayal of buildings, streets, and urban landscapes. This amalgamation of geometry and texture elevates visualization, analysis, and simulation capabilities, thereby enhancing experiences in urban planning, navigation systems, and virtual reality applications. Notably, only a few studies, such as [1], have delved into the semantic classification of urban meshes.

Motivation

In accordance with Wang et al. [9], mesh models often suffer from topological and geometric errors due to incomplete or misclassified geometric shapes, especially in complex scenarios involving adhesions between distinct objects. To address this challenge, we have introduced GeoSparseNet, a neural network tailored to enhance the classification of 3D shapes within large-scale urban meshes, accommodating both planar and non-planar shapes. Our method adeptly extracts objects and shapes, enabling a more comprehensive understanding of the intricate geometric characteristics prevalent in urban environments. GeoSparseNet integrates a novel Large Kernel Attention (LKA) mechanism by employing edge collapse techniques. This innovative approach empowers the network to focus on crucial data features and relationships, potentially resulting in enhanced model accuracy and performance.

In this study, we use deep learning to provide a novel Geometry-Aware Multi-Source Sparse-Attention CNN (GeoSparseNet) for the classification of urban textured meshes. Our framework aims to improve three specific aspects of classification, which include:

- Quality of Classification: Urban environments pose challenges for semantic classification due to the complex interplay between man-made and natural elements. While typical algorithms excel in categorizing large continuous surfaces, they struggle with precise object delineation, especially in areas with subtle visual differences. Our method addresses this by employing a geometry-aware approach that effectively distinguishes between planar and non-planar surfaces using the Large Kernel Attention (LKA) edge collapse technique, thereby enhancing classification accuracy in such challenging spaces.
- Distinctive Geometric Features: Semantically classifying 3D data involves assigning single labels to individual points or objects in 3D space, focusing on labeling elements rather than coherent scene segmentation. However, utilizing attributes at the local component level, like groups of triangles (nodes, edges), limits the derived features to a confined area, lacking broader contextual relationships. Distinctive feature abstraction is crucial for effective classification predictions in mesh-based geometric learning. Our approach achieves both local and global geometric feature acquisition by employing LKA-based edge collapse, removing redundant edges while preserving vital ones through task-oriented pooling and unpooling operations.

- Efficiency: Current deep learning techniques encounter challenges in handling large-scale 3D data, especially concerning extensive urban environments. This limitation has been identified previously by Landrieu et al. [10] and further acknowledged in subsequent work like Hui et al. [11]. Following the principles established in prior research focused on improving efficiency, our method introduces distinctive local and non-local characteristics tailored for enhanced classification, leveraging Afzal et al. [12]'s Scaled Cosine Similarity Loss (SCSL). GeoSparseNet aims to advance semantic classification and improve object delineation within large-scale urban meshes.

The paper's structure can be outlined as follows: In Section 2, an in-depth literature review is presented, offering an extensive examination of the existing research and knowledge on the topic. Moving forward, Sections 3–6 are dedicated to different aspects of our study. Section 3 details our approach, Section 4 outlines the experiments we conducted, Section 5 provides a thorough discussion of our findings, and finally, in Section 6, we present our conclusions. This organization allows for a systematic and comprehensive exploration of the research process and outcomes.

2. Related Work

While there exists an extensive amount of research on semantic segmentation of urban large-scale data, Ulku et al. and Adam et al. [3,13] surveys proposed that in the realm of 3D meshes, semantic segmentation pertains to the classification of individual elements into specific categories. Rook et al. [14] propose an extension to the existing class definitions of Roof Surface and Wall Surface in CityGML to demonstrate that by expanding these definitions, automatic semantic labeling of a CityGML file with Level of Detail 1 and 2 becomes achievable. This section focuses solely on approaches tailored for the processing of urban large-scale 3D data, such as point clouds and meshes.

2.1. Semantic Classification of Urban Models

An essential stage in semantic classification involves feature extraction. Deep learning methods tend to perform more effectively when ample training data is available, as observed in studies [3,15,16]. These techniques often rely on contextual information for feature computation or learning. However, capturing effective global contextual features can be challenging [17–19].

For instance, addressing this obstacle and dealing with 3D urban scenes, they can be transformed into a 4-way rotationally symmetric (4-RoSy) area, employing methods such as Huang et al. [20] QuadriFlow as demonstrated in their TextureNet [21]. This approach yields a uniformly distributed orientation field of sampled points. The use of frame fields, as exemplified by Yang et al. [22] in the PFCNN project, enables the linking and alignment of tangent planes with the geometric characteristics of the mesh surface through the parallel transport technique. Furthermore, this allows for the consolidation of acquired features in multiple orientations.

Another revised deep learning model called LO-Net was presented by Li et al. [23] and significantly improved categorization accuracy for common road scene elements. This model created the necessary datasets for the study by extracting distinctive objects from preprocessed point clouds obtained from a mobile LiDAR system. Similarly, Wilk et al. [24] presented sub-algorithms employing CNNs as their fundamental feature for precise classification of oblique aerial scenes. However, the implementation in this instance is based on proprietary architecture. Zhang et al. [25] also introduced a mesh-based Dynamic Graph Convolutional Neural Network (DGCNN) tailored for the semantic segmentation of textured 3D meshes.

The studies by Geng et al. [26] and Peng et al. [27] provides a new viewpoint on creating and optimizing urban ecological networks in light of the tension between expanding ecological networks in densely populated metropolitan areas and the increasing demand on land resources. Their methods include source site selection and resistant surface development. It also looks at changes in landscape connectivity across different development scenarios.

2.2. Attention Mechanism

The attention mechanism can be seen as an adaptable selection process based on input features, and it was introduced by Mnih et al. [28] into computer vision through the concept of recurrent models of visual attention. This mechanism has proven advantageous in numerous visual tasks, including image classification [29,30], object detection [31,32], and semantic segmentation [33,34]. Guo et al. [35] notified that within the realm of computer vision, attention can be categorized into four fundamental types: temporal attention, spatial attention, channel attention, and branch attention, with hybrid forms like spatial and channel attention. Each of these attention types exerts a distinct influence on various visual tasks.

Derived from NLP [36,37], self-attention stands as a distinct attention mechanism. Its efficacy in apprehending extensive dependencies and flexibility has led to an increasingly pivotal role in computer vision [38–40]. A variety of deep self-attention networks, often referred to as vision transformers, [41–43], have consistently outperformed CNNs in various visual tasks, underscoring the substantial potential of attention-based models.

Henceforth, Basu et al. [44] introduce an attention-driven structure for mesh data, which demonstrates proven equivariance to the described transformations. Their approach utilizes relative tangential features, offering a straightforward, efficient, and equivariance-oriented substitute for utilizing raw node positions as inputs. Han et al. [45] introduce a novel method designed to capture extended dependencies using a temporal attention model inspired by the transformer architecture. Whereas, Milano et al. [46] propose a method that harnesses the strengths of both types of approaches while overcoming their constraints. They expanded a primal-dual framework rooted in the graph-neural-network domain to triangle meshes. This involves defining convolutions on two distinct graphs derived from an input mesh. Their approach incorporates features from both edges and faces of a 3D mesh as input, dynamically aggregating them via an attention mechanism.

While these approaches typically necessitate contextual information for feature computation or learning, capturing truly effective global contextual features can be a challenging endeavor. Our GeoSparseNet takes advantage of enhanced spatial relationships within the geometry-aware CNN by exploiting the large-attention-based edge collapse technique, at both local and global levels. This strategic approach significantly enhances the model's ability to grasp contextual information and acquire unique features specifically tailored for urban large-scale 3D mesh classification.

3. Materials and Methods

In this section, we provide notations and the architectural details required for the implementation of our proposed GeoSparseNet method during the training of Convolutional Neural Networks (CNN) on urban large-scale mesh dataset, utilizing a known modified loss function SCSL [12], specifically designed for 3D mesh models. We propose that maintaining geometric integrity in urban mesh data is essential for enhancing the accuracy of shape classification and segmentation tasks. Given that mesh edges bear a resemblance to pixels in a 2D image, we employed them as the fundamental units for executing all processes in GeoSparseNet. As illustrated in Figure 1, a mesh edge, denoted as e, represents the shared boundary of two neighboring triangles, leading to an optimized convolutional neighborhood of a constant size characterized by precisely four adjacent edges, namely $a, b, c,$ and d.

A mesh can also be stated as (V, L, E), where V is a collection of vertex positions, such as, $V = \{v_1, v_2, \ldots v_n\} \in \mathbb{R}^3$. For triangular meshes, the linkage is defined as L (triplets of vertices) that represents three edges $(E = e_1, e_2, e_3)$. Features like colors or normals are related to the mesh elements V, L and E. Feature edges E offer improved perception at the network layer and can preserve a geometric feature that is similarity-invariant, such as

RGB in photos. The initial geometric input features and convolutional adjacent instance connectivity parameters within the network are comparable to those in [12].

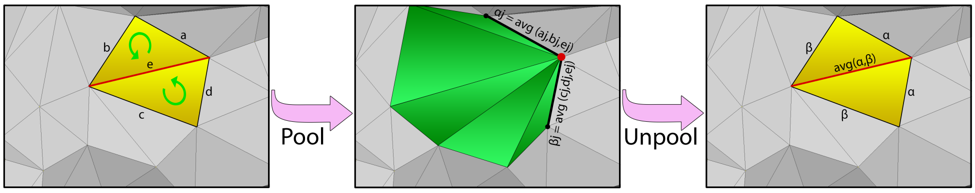

Figure 1. Feature aggregation using LKA-based edge collapse, during mesh pooling and unpooling.

3.1. GeoSparseNet Architecture

3.1.1. Convolution

Figure 1 illustrates how four consecutive neighbors are used to provide spatial support in order to get the convolution operation for mesh edges. The convolution operation, dot product between the consecutive neighbors (four edges e^1, e^2, e^3, e^4) and kernel k for any edge e may be expressed as follows:

$$k \cdot E + \sum_{j=1}^{4} e^j \cdot k_j \tag{1}$$

In this case, E corresponds to the j^{th} neighbor's edge feature of the convolution operation. Every filter is capable of activating on two possible edges, for instance, k_0 on a or c. Consequently, the four neighboring edges of edge e depicted in Figure 1, namely e^1, e^2, e^3, and e^4, could correspond to a, b, c, d, or c, d, a, b. The application of symmetric functions to the input data pairs has the potential to generate a new set of convolution neighbors while ensuring order invariance. The following equation represents the edges E:

$$E = (e^1, e^2, e^3, e^4) = (|a - c|, a + c, |b - d|, b + d) \tag{2}$$

Efficiently optimized batched operators, like conv2D, can be employed in the implementation by consolidating all edge features into a feature tensor of dimensions $C^N \times E^N \times 5$, where E^N represents the number of edges, C^N signifies the feature channels, and 5 accounts for the edge and convolution neighborhood, as described in Equation (2). Following this, the tensor is subjected to multiplication with a matrix containing convolution weights using the conventional General Matrix Multiply (GEMM) operation.

Like in image processing, after the process of convolution, a new group feature tensor is formed that contains an equal number of new features as the number of convolution kernels. For edge collapse, the procedure then uses a Large Kernel Attention (LKA) technique. It is noteworthy to emphasize that the updated connectivity dictates the new collection of convolution neighborhoods for the next convolution operation after each pooling phase.

3.1.2. Large Kernel Attention (LKA)

According to the input characteristics, the attention mechanism could be thought of as an adaptive decision-making system that can choose the discriminative features and automatically disregard noisy details. The creation of an attention map, which highlights the significance of various components, is a crucial stage in the attention mechanism. We must understand how various aspects relate to one another in different features relations.

There are two commonly used methods for establishing connections between various components. The first technique involves employing the self-attention mechanism [40,47] to capture long-range dependencies. Initially developed for NLP applications, self-attention encounters three key limitations when applied to computer vision tasks. Firstly, it disregards the inherent 2D/3D structures, treating them as 1D sequences. Secondly, its quadratic complexity makes it impractical to handle high-resolution images due to computational costs. Lastly, while it effectively achieves spatial adaptability, it overlooks adaptability within the channel dimension, thereby limiting its overall effectiveness in computer vision contexts.

The second method involves creating an attention map and constructing salience using large kernel convolution [30,48]. This method still has several glaring drawbacks. Considerable computational cost and parameterization are introduced by large-kernel convolution. Our LKA consists of four stages with decreasing output spatial resolution, $\frac{H}{4} \times \frac{W}{4}$, $\frac{H}{8} \times \frac{W}{8}$, $\frac{H}{16} \times \frac{W}{16}$, $\frac{H}{32} \times \frac{W}{32}$, respectively, forming a simple hierarchical structure. Here, H and W stand for the input mesh's height and width, respectively. In the context of large kernel attention in 3D meshes with decreasing output spatial resolution, the H and W parameters refer to the spatial dimensions of the attention maps at different stages of the network. When dealing with 3D meshes, the concepts of height, width, and depth can be interpreted differently compared to traditional 2D images. Exploiting large kernel attention and decreasing spatial resolution, the network operates in a hierarchical manner where attention is applied across different levels of detail. The attention mechanisms applied to these dimensions using LKA would involve over larger regions or receptive fields within the mesh, covering multiple vertices or nodes in both the height and width dimensions. As the network progresses through different stages, the output spatial resolution decrease, implying that the attention mechanism gradually reduces the detailed representation of the mesh while capturing more abstract and higher-level features.

The number of output channels C is growing as resolution decreases. We suggest breaking down the large-kernel convolution process to capture long-range relationships in order to get around the aforementioned drawbacks and use the benefits of self-attention and large-kernel convolution. A large-kernel convolution may be broken down into three parts, as mentioned by Guo et al. [48], channel convolution (1×1), spatial long-range convolution (depth-wise dilation), and a spatial local convolution (depth-wise coupling). To be more precise, we may break down a $K \times K$ convolution; standard convolution: $conv2D = 1 \times 1$, depth-wise convolution: $D_c = (2d_c - 1) \times (2d_c - 1)$, and depth-wise dilation convolution: $D_d = \left(\frac{K}{D_d} \times \frac{K}{D_d}\right)$. We are able to capture long-range relationships with little processing effort and parameters by using the aforesaid decomposition. We can evaluate an edge's relevance and create an attention map after we have the long-range link. Figure 2 illustrates the LKA module and can be expressed as:

$$\mathbb{A} = conv2D_{1\times 1}(D_d(D_c(f))) \tag{3}$$

$$F_{out} = \mathbb{A} \otimes f \tag{4}$$

$f \in R^{H\times W\times C}$ is the input feature in this case. $\mathbb{A} \in R^{H\times W\times C}$ stands for attention map. Each feature's relevance is indicated by its value in the attention map. $\otimes$ indicates element-wise product. Unlike sigmoid and softmax, which are typical attention strategies, LKA does not need an extra normalizing function [48]. Additionally, we think that adaptively modifying output depending on input feature—rather than the normalized attention map—is one of the primary attributes of attention approaches. The LKA that Guo et al. [48] suggested combines the benefits of self-attention and convolution. It considers the broad receptive field, linear complexity, dynamic process, and local contextual information. LKA further accomplishes flexibility in the channel dimension along with the spatial dimension. It is noteworthy that in deep neural networks, distinct channels often represent distinct objects as explained by Guo et al. [35], and that flexibility in the channel dimension is crucial for tasks involving 3D meshes (refer to Figure 2).

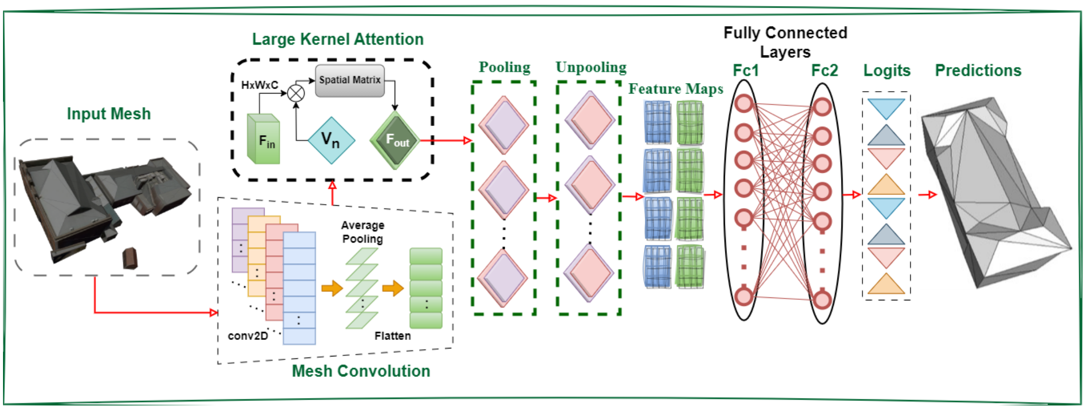

Figure 2. An end-to-end illustration of our novel GeoSparseNet architecture.

3.1.3. Pooling

By defining three fundamental procedures that together broaden the concept of pooling, we expand the use of traditional pooling to irregular data: First, determine the pooling area based on proximity. Second, Aggregate features in every pooling area using LKA-based edge collapse. Lastly, For the cumulative features, restructure the adjacency of newly merged regions.

Adjacency is implicit when pooling on regular imagery data, therefore the kernel size selection immediately determines the pooling area. The new adjacency is again intrinsically specified since characteristics in each area are combined, such as max or average, to produce an equally distributed grid space. The traditional pooling in images already has unfolded its generalization ability in many 2D computer vision tasks. Another unique use of generalized pooling is mesh pooling, in which adjacency depends on the underlying mesh topology. In contrast to images, where 2 × 2 pooling often has a natural reduction factor of 4, several edge collapse dispositions are needed for standard mesh pooling. In our settings, each of these edge collapse operations among five targeted edges will transform into two edges with the assistance of LKA. Every pooling process enables to direct the network to acquire the required mesh depth by adding an LKA-based hyperparameter that indicates the array of objective edges in the final mesh. According to [49], to obtain mesh adjacency information during runtime, certain data structures that are recurrent must be accessed and updated.

To enable the network to choose which areas of the mesh are important for solving the job, we rank the edge-collapse order according to the edge characteristics' magnitude (using a priority queue). Because of this, the network might collapse in certain locations that are less significant to the loss in an uneven manner.

By using a priority queue, we rank the edge-collapse order according to the magnitude of the edge characteristics determined by the LKA, enabling the network to determine which areas of the mesh are necessary to complete the job. As a result, the network can collapse in certain locations that are less significant to the loss in an uneven manner. Remember that while collapsing an edge that is next to two faces, the network eliminates three edges (Figure 1) since both faces are combined into one edge. Three edges create each face in a mesh: the minimal edge, its two adjoining peers, and the essential edge that needed to collapse using red color along with the neighboring peers in black (Figure 1). Each facet's three edges' features are combined into a single edge feature by averaging the overall feature channels.

The $l_2 - norm$ of an edge, or its feature magnitude obtained by LKA, establishes the edge collapse precedence. The features are then merged by blending operations for every single one of the incidence triangles of the most minimal edge feature e in an urban mesh to create two new feature vectors LKA-based edge collapse, α and β. The following lists for both triangles, the edge attributes in channel indices j are given below:

$$\alpha_j = avg(a_j, b_j, e_j), \;\; and, \;\; \beta_j = avg(c_j, d_j, e_j) \tag{5}$$

Subsequent edge collapses are taken into account while updating the half-edge data structure. Lastly, it is noteworthy that not all edges are collapsible. The network prevents edge collapses that produce non-manifold faces since they go against the notion of four convolution neighborhoods. Therefore, an edge is not considered capable of collapsing if it has two border points or three nodes on its 1-ring junction.

3.1.4. Unpooling

The process of pooling has a partial inverse called unpooling. Pooling layers decrease feature activation resolution by encoding or compressing information while unpooling layers enhance feature activation resolution by decoding or uncompressing information. In order to increase feature activation, the pooling operation keeps track of the merging operations' history (such as max locations). Because unpooling lacks learnable parameters, to recover the initial resolution which had been eliminated during the pooling process, it is often employed in conjunction through convolutions. When paired alongside the convolution operations, the unpooling effectively becomes a learnable process.

To enhance the quality of the mesh topology and edge features, every unpooling layer has a corresponding layer of pooling. By recording the connection before pooling, the unpooling layer restores the upsampled topology (prior to mesh pooling). Keep in mind that, similar to images, upsampling the connection is a reversible process. We keep a graph that records the adjacencies between the original edges (before pooling) and the new edges (after pooling) for the sake of computing unpooled edge features. A weighted mixture of the pooled edge features renders for each unpooled edge feature. Figure 1 illustrates the usual unpooling scenario.

3.1.5. Loss Function

According to Afzal et al. [12], a loss function for meshes has to employ a bigger value η for feature vectors and weights rather than 1 for softmax to properly converge on a training dataset. Just adding a scaling layer ℓ and a cosine layer thereafter will immediately do this. A single learnable parameter $\ell = \eta^2$ is included in this scale layer. This could have a set value based on the various class numbers. Nevertheless, they preferred that the network learn the parameter by back-propagation in their study rather than adding another hyperparameter. Since their loss defines the modified softmax loss function as follows, we may thus benefit from it:

$$L'_{SCSL} = -\frac{1}{K}\sum_{p=1}^{K} \log \frac{e^{\left(\ell\, \tilde{W}_{y_p}^{T}\, \tilde{f}(\tilde{x}_p) + b_{y_p}\right)}}{\sum_{q=1}^{J} e^{\left(\ell\, \tilde{W}_{q}^{T}\, \tilde{f}(\tilde{x}_p) + b_q\right)}} \tag{6}$$

Here, in the above Equation (6), $f(\tilde{x})$ denotes the normalized $f(x)$, and $\tilde{W}$ denotes the weights. ℓ represents the scale layer of the learnable parameter.

4. Results

This section documents the extensive tests we conducted to substantiate our proposal for the novel GeoSparseNet for city models. Firstly, the network settings and database that we utilized in our tests to apply the new GeoSparseNet are included in this section. Subsequently, detailed reports on assessments of the suggested GeoSparseNet in an urban mesh classification job are provided. Lastly, we provide a succinct analysis of the benefits and drawbacks of the suggested GeoSparseNet.

4.1. Network Configurations

System settings: The experiments were conducted using NVIDIA GeForce RTX 2080Ti GPUs on an Ubuntu operating system.

Network settings: Kernel size $k = 5$, Adam optimizer with learning rate $lr = 0.0002$, $\lambda = 0.01$, $stride = 1$, and $padding = 1$.

Parameters settings: Pooling resolution $pool\_res = [800, 650, 400, 320]$, number of convolution filters $ncf = [128, 256, 512, 1024]$, and iterations $niter = 500$.

4.2. Adelaide Dataset

In this study we have included a 3D city model dataset of a full scan of City Adelaide Australia [50] to demonstrate the substantial power of GeoSparseNet to handle all types of triangular mesh data in a classification task. The dataset comprises on large number of sub-folders and files containing mesh models and metadata portraying block maps of buildings in the whole of Adelaide City. The urban model utilizes coordinate System-MGA Zone 54, with origins direct approximately for,

City Blocks: Origin(X) 278,434.0000, Origin(Y) 6,130,514.0000, Altitude(AHD) 125.0000

Balls: Origin(X) 281,063.3990, Origin(Y) 6,132,895.2440, Altitude(AHD) 44.9000

Rotunda: Origin(X) 278,434.0000, Origin(Y) 6,130,514.0000, Altitude(AHD) 125.0000

Vic_Pk_Fountain: Origin(X) 280,771.9397, Origin(Y) 6,132,324.6420, Altitude(AHD) 45.0633

4.3. Evaluation of GeoSparseNet

We first discussed the parameter for attention sparsity.

$$\mu_j = S_\lambda\left(W_j^T f(x)\right) \tag{7}$$

where, μ_j is the j^{th} element of the sparsely fully connected S layer with weights $_j$ and S is

$$S_\lambda(w) = \begin{cases} w & \because \lambda \leq w \\ 0 & \because w < \lambda \end{cases} \tag{8}$$

here, λ is a parameter for sparse connections. The achieved edge collapse features are finally weighted by LKA scores Equation (4).

A thorough ablation study was performed on GeoSparseNet by varying the attributes and parameters of the network. This analysis allowed us to determine the best settings for the network to improve robustness and accuracy in the classification of urban large-scale triangular meshes. The parameters affecting the overall performance and accuracy of the network are listed in Table 1 below:

Table 1. Ablation study of hyper-parameters.

pool_res	ncf	Accuracy (%)
600, 450, 300, 180	128, 256, 512, 512	83.7
600, 450, 300, 180	128, 256, 512, 1024	84.5
700, 550, 400, 280	128, 256, 512, 512	86.1
700, 550, 400, 280	128, 256, 512, 1024	87.2
800, 650, 400, 320	128, 256, 512, 512	86.7
800, 650, 400, 320	128, 256, 512, 1024	87.5

Please note that the pooling resolution and number of convolution filters are the main hyper-parameters that show significant changes in terms of accuracy when using GeoSparseNet for classification tasks in urban large-scale triangular meshes.

Figure 3 shows the distributions of evaluation scores on the 5 classes from the 3D city model of the Adelaide dataset for $\lambda \in [0, 1]$. Compared with no sparsity ($\lambda = 0$), Accuracy is higher for λ belongs to $(0, 0.05)$, while is lower for λ belongs to $(0.07, 1)$. From these observations, GeoSparseNet benefits from the sparsity, resulting in improved prediction performance. In our experiment, we adopted the sparsity $\lambda = 0.01$ for simplicity.

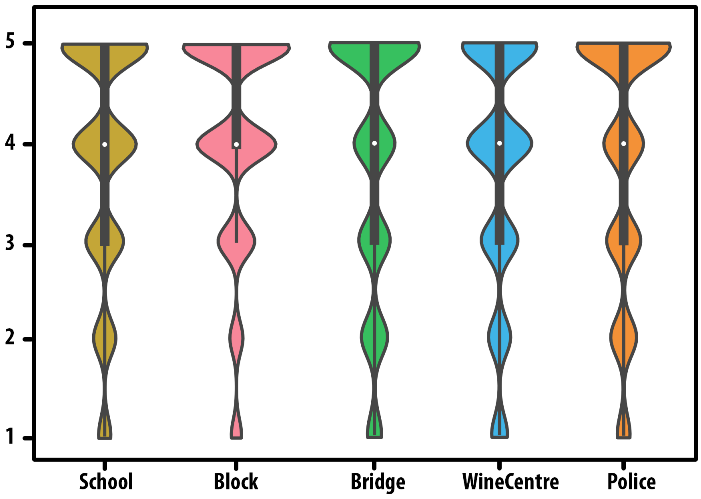

Figure 3. Distributions of evaluation scores on the five classes (black lines: Quartiles, while points: Medians).

For the network parameters in GeoSparseNet, we here investigated the best performance by varying the pooling resolution *pool_res* and number of convolution filters ncf on the 3D city model dataset. In Figure 4, GeoSparseNet has higher accuracies when $pool\_res = [800, 650, 400, 320]$ with $ncf = [128, 256, 512, 1024]$. While GeoSparseNet delivers comparable accuracies for $pool\_res = [800, 650, 400, 320]$ with $ncf = [128, 256, 512, 512]$ and $pool\_res = [700, 550, 400, 280]$ with $ncf = [128, 256, 512, 1024]$ with slightly less computational cost.

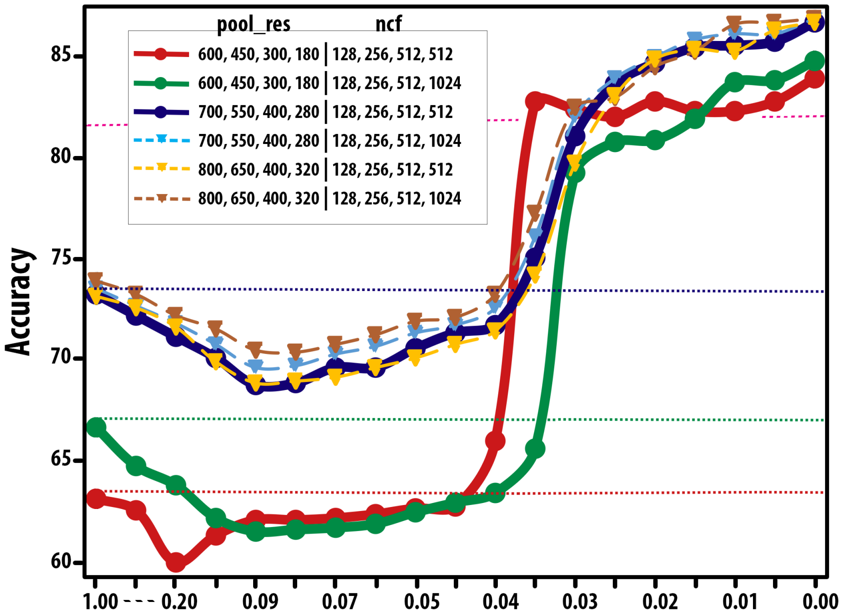

Figure 4. The effects of the λ parameter, *pool_res* and ncf on overall accuracies.

Among various structures and classes in the Adelaide Dataset, five classes are shown below for comparisons. The green curve in Figure 5 and the last column in the Table 2, represent average values. GeoSparseNet highlights its enhanced performance by demonstrating a discernible rise in accuracy across all classes.

Table 2. Classification comparisons result on five classes.

Method	**Class** (%)					**Average**
	School	**Block**	**Bridge**	**WineCenter**	**Police**	**(%)**
PointNet [51]	84.6	85.7	81.2	82.9	84.1	83.7
PointNet++ [52]	85.8	86.8	83.7	85.3	85.9	85.5
DIFD-Net [53]	86.7	86.6	83.8	85.6	85.3	85.6
PointCNN [54]	87.5	87.7	84.8	86.6	86.9	86.7
Y-Net [55]	87.8	87.6	85.1	85.7	87.3	86.7
MeshNet [56]	87.7	87.9	86.3	86.8	86.8	87.1
GeoSparseNet (ours)	87.8	88.2	87.1	86.8	87.6	87.5

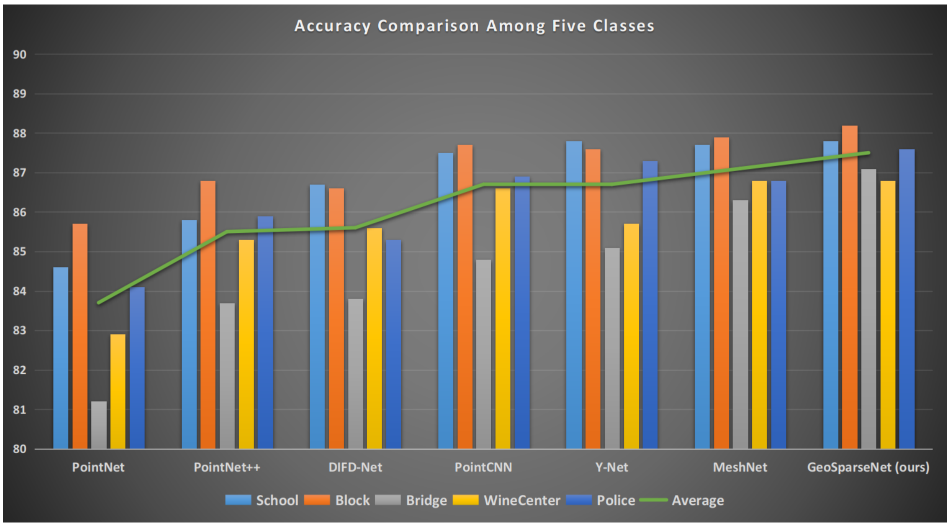

Figure 5. Accuracy comparisons of predictions on five classes.

For further analysis, we present a comparative assessment of prediction classification labels between MeshNet [56] and GeoSparseNet using confusion matrices. Figure 6 visually demonstrates the distinct predictive performances of both models when incorporating geometric and photometric features (as shown in (a) and (b) respectively). Notably, when both types of features are utilized together, we observe significant differences in how the models classify certain labels. Particularly in the case of MeshNet (a), there is a noticeable confusion between the school and police classes. This confusion might stem from two probable reasons: Firstly, there could be instances where distinguishing between these classes during the annotation process proves challenging. Secondly, within this configuration, the dominance of photometric features over geometric ones might contribute to this confusion. Upon evaluating GeoSparseNet (b), our findings corroborate the argument that our method generates more discriminant features, significantly impacting all five classes. Notably, GeoSparseNet demonstrates a reduced tendency for confusion between classes, suggesting its ability to maintain clearer distinctions without an over-reliance on specific features.

The suggested technique showcased significant improvements over original photogrammetry mesh models, generating lightweight polygonal meshes with more concise descriptions and labels through the utilization of LKA-based edge collapse pooling and unpooling processes. Notably, elements like baseboards, window ledges, roof tiles, and chimneys may be perceived as insignificant details based on the data's intended use. For instance, previous methods often overlooked these details, which could be crucial for simulation, quantitative surveying, or accurate classification of building types.

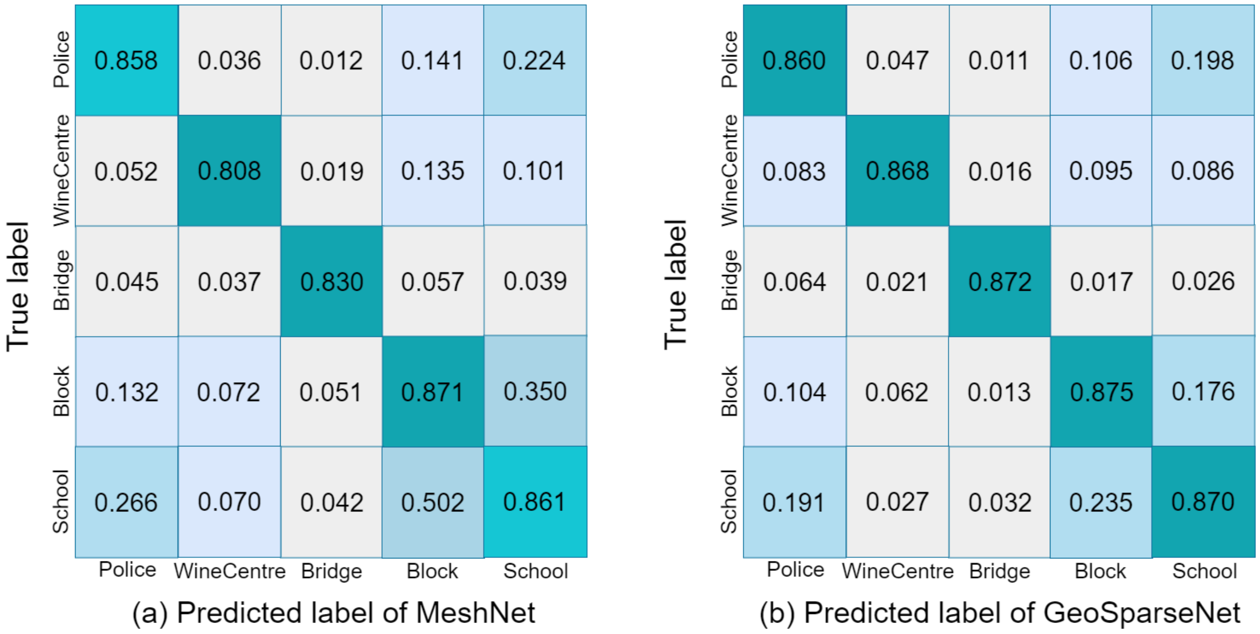

Figure 6. Comparison of Confusion Matrices incorporating geometric and photometric features between MeshNet (**a**) and GeoSparseNet (**b**).

In essence, our classification approach addresses these limitations by effectively addressing flaws present in smooth areas, resulting in highly precise labeled meshes. This advancement is noteworthy, particularly when considering real-world models generated from aerial photogrammetry methods that tend to include numerous cumbersome surfaces. Groger et al. [57], mitigated representations achieved through similar to our approach meet the standards outlined in CityGML LOD2 models. This indicates that our methodology successfully produces meshes meeting the quality benchmarks set for 3D city models, signifying a significant advancement in generating accurate, detailed, and purpose-driven mesh representations.

4.4. Computational Complexity

The complexity of a deep neural network model depends on several factors, including the number of layers, the number of neurons per layer, the type of activation functions used, and the size of the training dataset. In general, the complexity of a deep neural network increases as the number of layers and neurons per layer increases. This can result in overfitting when the model becomes too specialized to the training data and performs poorly on new and unseen data. On the other hand, if the model is too simple, it may not capture the underlying patterns in the data and result in underfitting. The choice of activation functions and the optimization algorithm used to train the model also affect the complexity. In addition, the size of the training dataset can impact the complexity, as larger datasets typically require more complex models to effectively learn the underlying patterns in the data.

For feedforward neural networks, the time complexity is typically $\mathbb{O}(n \times m \times l)$, where n is the number of inputs; m is the number of neurons in the largest layer; and l is the number of layers, as in the case expressed in Equation (11). This assumes that each neuron has a constant time complexity for processing inputs and computing the output. The time complexity for training a deep neural network can be much higher, as it involves computing the gradients of the loss function with respect to the model parameters and updating the parameters based on the gradients. This process is typically repeated multiple times for different batches of the training data until convergence.

Figure 7 (we took fifty epochs) shows the time complexity of GeoSparseNet compared to various $pool\_res$ and ncf. Hereunder's Big-$\mathbb{O}$ notation was employed to calculate the

GeoSparseNet run-time complexity by simply omitting $\mathbb{T}\,\Delta_{\vartheta}$ as a directive of 1, where $\mathbb{T}$ denotes time.

$$\mathbb{T}_{gradient\,decent} = k_{gradient\,iterations} \cdot \mathbb{T}_{weights} \tag{9}$$

Let us assume that k gradient iterations occur:

$$\mathbb{O}\left(\mathbb{T}_{gradient\,decent}\right) = k \cdot k^{4} = k^{5} \tag{10}$$

Here, we consider k layers, each with k neurons. The gradient descent is performed for k iterations, and the total run-time of back-propagation can be calculated as:

$$\mathbb{O}\left(k^{5}\right) \because \forall k \geq 1 \tag{11}$$

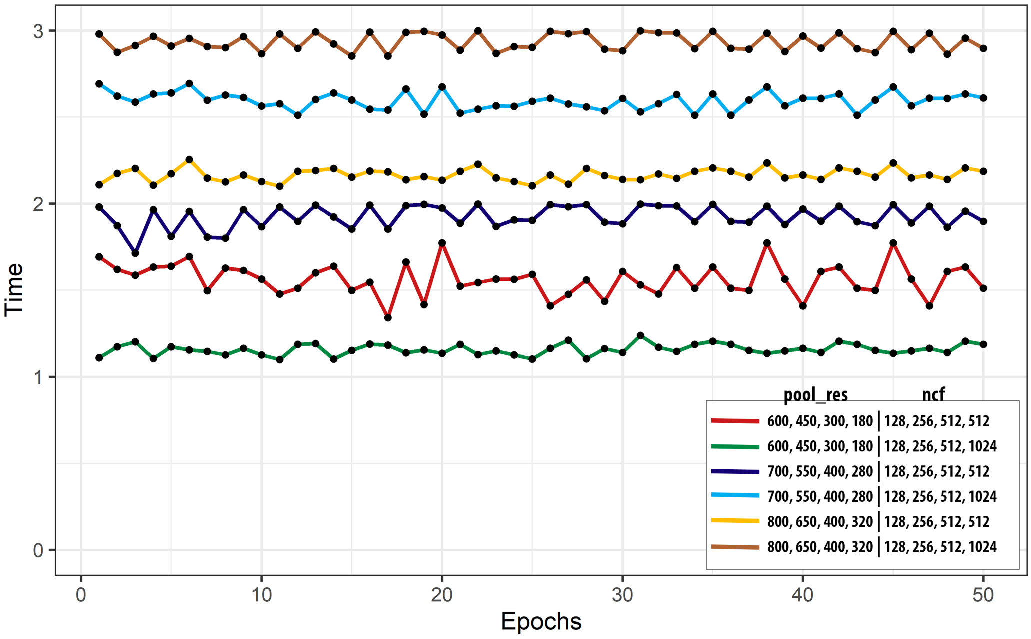

Figure 7. Computational complexity of our method.

4.5. Results Visualization

To evaluate the robustness, efficacy, and adaptability of our novel GeoSparseNet, we conducted an evaluation using polygonal mesh models derived from an urban large-scale 3D model dataset of the city of Adelaide, Australia [50]. We conducted a comparison using a single building from the city model, specifically focusing on various buildings in a classification task. This comparison experiment highlighted the sensitivity of geometric feature-preserving using GeoSparseNet. Our approach, benefiting from a smoothing classification, yielded shipshape and more precise building detection results. The geometric features demonstrated efficient retrieval of sharp, polished features, even if there are small roof patches. In contrast, the method listed in Table 2, lacking feature-based constraints, tended to flatten adjacent non-coplanar regions, resulting in a smaller number of geometric areas compared to our method.

Figure 8 shows the zoomed mesh object of block-1 from the city model and Figure 9 illustrates our pooled mesh and classification results of a single building in block-1 from the Adelaide City dataset. Whereas, Figure 10 provide our predictions on various building structures from the full scan of Adelaide City, and the red arrow in Figure 11 indicates block-1 among numerous blocks in this urban large-scale city mesh model. In summary, our proposed GeoSparseNet is capable of transformation invariance exhibiting overall superior accuracy over these urban large-scale meshes in a classification task.

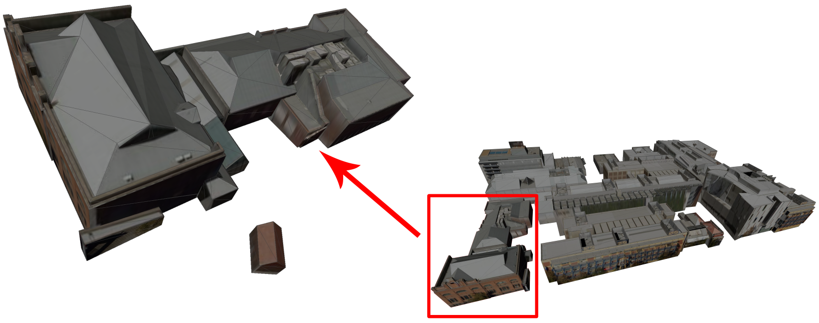

Figure 8. Zoomed mesh object of block-1 of city Adelaide (Figure 11) [50], Australia.

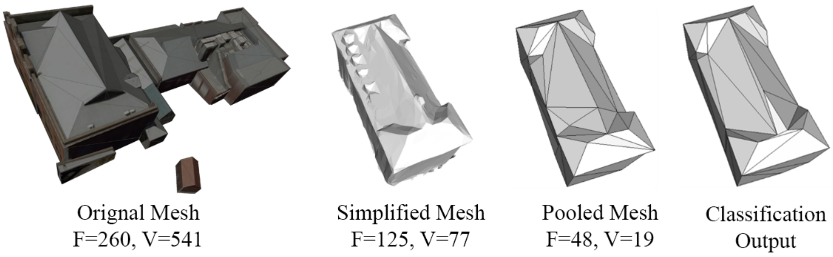

Figure 9. Note that illustration of the last column, our proposed trained model has an improved ability to remove redundant edges while preserving important ones with precise detection results of a single building in block-1 among various classes in a mesh classification task of the city model. (F: face number; V: vertex number).

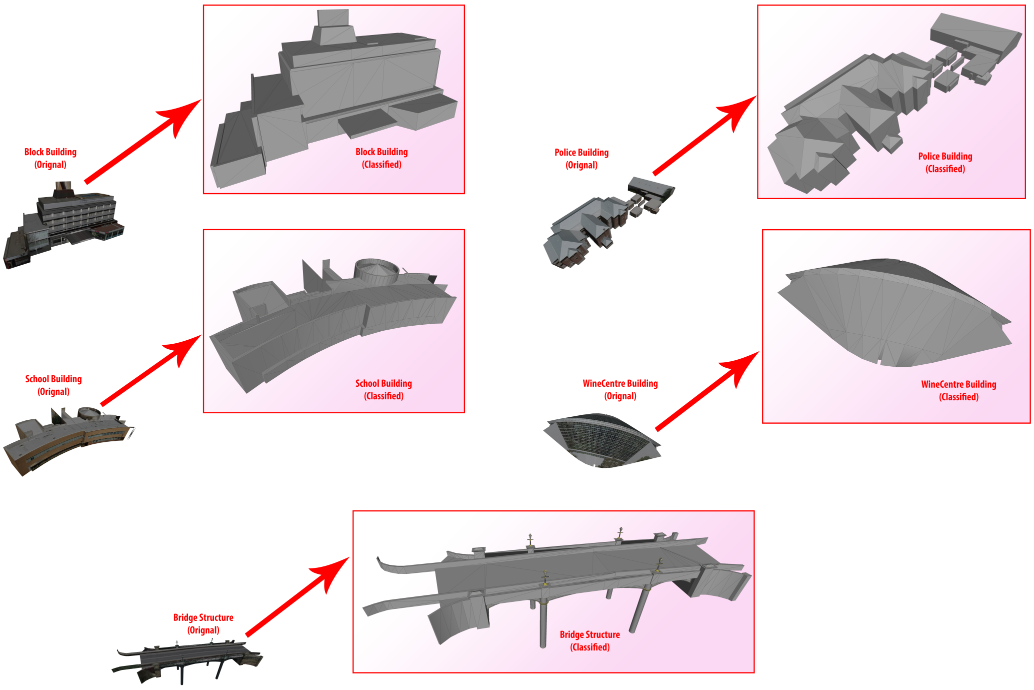
Block Building (Orignal)
Block Building (Classified)
Police Building (Orignal)
Police Building (Classified)
School Building (Orignal)
School Building (Classified)
WineCentre Building (Orignal)
WineCentre Building (Classified)
Bridge Structure (Orignal)
Bridge Structure (Classified)

Figure 10. The classification predictions of 5 classes from the Large-scale 3D City Model Adelaide.

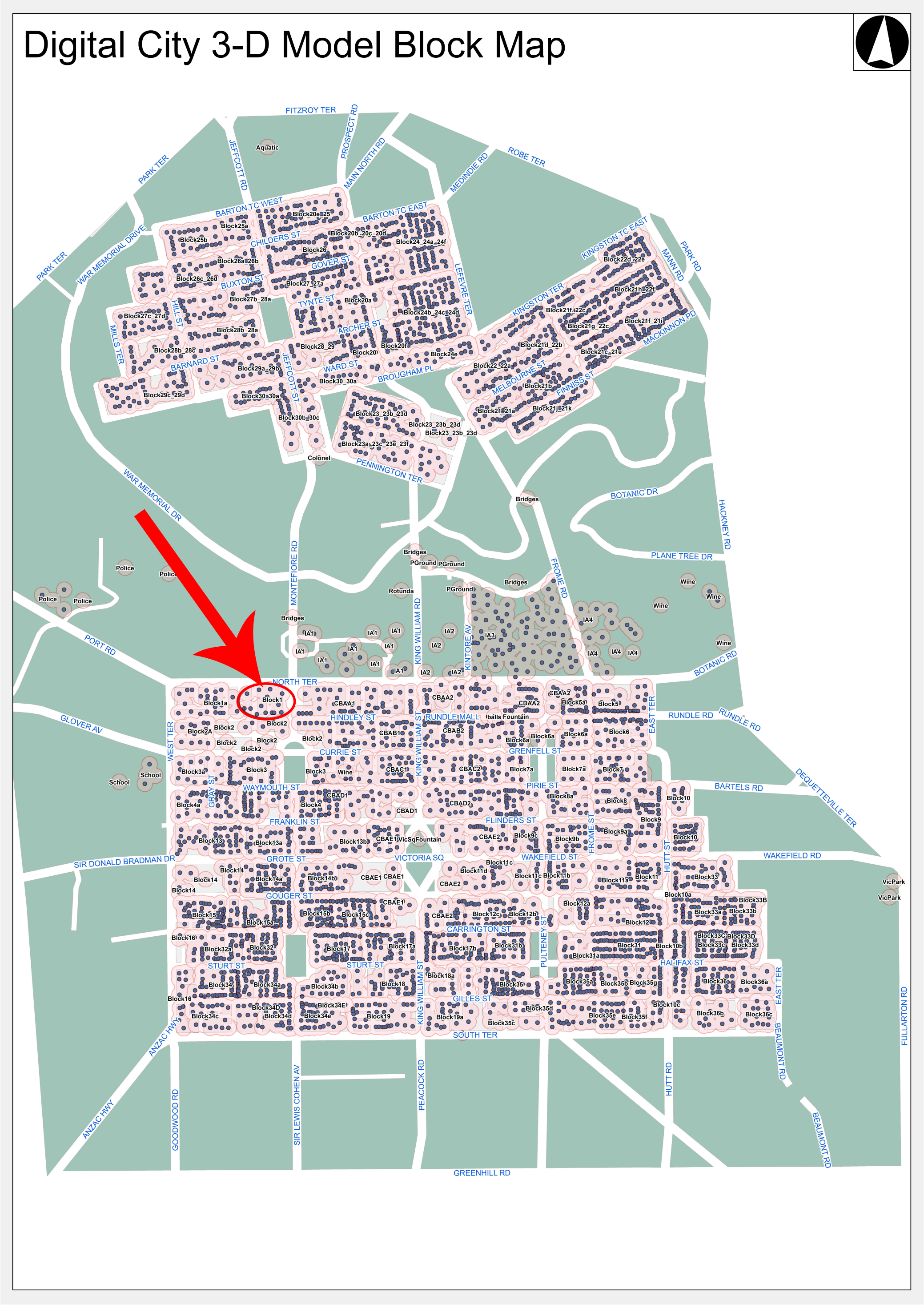

Figure 11. Full city scan of Adelaide [50], Australia.

We used the Large-scale 3D City Model Adelaide dataset's building mesh models, each having irregular planar area connectivity and attributes from different building architectures, to assess the effectiveness of the suggested strategy. Our method can certainly involve identifying and distinguishing different types of regions or elements within a three-dimensional space, regardless of whether they are coplanar or non-coplanar. The whole city block with its varied building structures is seen in Figure 11. Figure 8 provides an illustration of block-1 among various buildings, which is shown by the red arrow in Figure 11.

The evolution of the result on a block-1 building model produced by aerial photogrammetry methods is shown in Figure 9. The suggested technique is successful in recognizing contour and geometric features, structured areas, and flattening surfaces. There could be some overly smoothed or very curved areas in the original mesh models. Since our classification technique does not divide these curved regions, it uses LKA-based edge collapse operators to simplify unstructured sections, such as air conditioners on rooftops. A significantly reduced number of faces characterizes the structural elements, such as the building's major facade, which are much simplified using GeoSparseNet. By comparing, we can see that the model's triangle number has drastically decreased after the mesh pooling operation. The outcome demonstrates the ability of our approach to produce and categorize well-decimated meshes with respect to geometric precision (Figure 9), visual coherence, and classification (Figure 10).

Additionally, a comparison of classifications employing a single building from the city model (Figure 9) showed that our method's smoothing procedure produces cleaner and more accurate results for the classification prediction. The GeoSparseNet's ability of geometric features to capture crisp, clear features from minor details. Because the GeoSparseNet exploits LKA-based edge collapses pooling and unpooling constraints, resulting in better overall accuracy Table 2.

5. Discussion

A perfect classification for simplified mesh would have the same visual coherence as the original model and just the bare minimum of intricacies needed to accurately portray the object without sacrificing any of its features. The definition of minimum complexity is still somewhat unclear, however. It is challenging to come up with a standard by which to end the simplification process. We provide a set of data in Figure 9 to illustrate how the desired number of nodes affects the visual appearance. It is evident that the general structures of the models do not significantly alter when the vertex counts of the models exceed during LKA-based edge collapse operation. Upon further reduction of the objective edges, the model's structure may eventually get lost, signifying an ineffective simplification attempt. When the overall number of nodes is high, we see that changing one or two of them has minimal impact on the model. In the simplified approach, every edge modification will provide surprising outcomes when the number of nodes is lowered to a minimal amount. This is a decent negotiation, even if it may not be the ideal number of nodes.

Since polygonal forms predominate in these structures, piecewise planar buildings and segmentation are our future primary target objects. Large free-form features in structures will make our technique more computationally demanding, and the method's benefits could disappear. Additionally, as quadratic optimization-based classification of mesh simplification techniques often have problems with polished nodes that are not precisely positioned with the decoupled surfaces, the suggested solution is unable to ensure this and will be our future task. additionally, to enable users to effectively change and evaluate the models, we want to parse the simplified building models into a few semantic components for future work.

6. Conclusions

This work introduced a novel method 'GeoSparseNet' for the classification of dense urban triangular meshes. The approach updates the node locations after filtering the face normals using the multi-source geometry-aware CNN technique. Filtering may improve sharp edge features when denoising meshes on planar or non-planar areas. We have shown that our mesh classification procedure, which most prior techniques just disregarded, may considerably increase the performance of the classification algorithm for urban 3D meshes. To lower the number of mesh faces, an enhanced LKA-based edge collapse approach for pooling and unpooling is suggested for better classification predictions. For both planar and non-planar locations, the technique effectively preserves the architectural structures via the use of a multi-source geometry-aware CNN strategy. The experimental findings demonstrate the great generality and wide applicability of the suggested technique in urban 3D model classification tasks. The streamlined models have broad applicability in several graphics and mapping fields.

Author Contributions: Conceptualization, M.K.A.; methodology, M.K.A.; software, M.K.A.; validation, M.K.A., Y.Z. and C.W.; formal analysis, M.K.A., Y.Z. and C.W.; investigation, M.K.A. and H.M.R.A.; resources, W.L., Y.Z. and C.W.; data curation, H.M.R.A.; writing—original draft preparation, M.K.A.; writing—review and editing, M.K.A., W.L., Y.Z., S.C., J.M.A., H.M.R.A., B.Y., C.W. and J.L.; visualization, M.K.A. and H.M.R.A.; supervision, C.W.; co-supervision, Y.Z.; project administration, Y.Z. and C.W.; funding acquisition, Y.Z., W.L., S.C., C.W. and B.Y.; correspondence, W.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by China Postdoctoral Science Foundation (Funder: Liu W.; No. 2021M690094), National Natural Science Foundation of China (Funder: Zang Y.; No. 61971363), The Educational Project Foundation of Young and Middle-aged Teacher of Fujian Province of China (Funder: Chen S.; No. JAT210672), The 14th Five-Year Plan of the Xishuangbanna Tropical Botanical Garden, Chinese Academy of Sciences (Funder: Bai Y.; No. E3ZKFF7B), and FuXiaQuan National Independent Innovation Demonstration Zone Collaborative Innovation Platform (Funder: Wang C.; No. 3502ZCQXT2021003).

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Gao, W.; Nan, L.; Boom, B.; Ledoux, H. SUM: A benchmark dataset of semantic urban meshes. *ISPRS J. Photogramm. Remote Sens.* **2021**, *179*, 108–120. [CrossRef]
2. Besuievsky, G.; Beckers, B.; Patow, G. Skyline-based geometric simplification for urban solar analysis. *Graph. Model.* **2018**, *95*, 42–50. [CrossRef]
3. Adam, J.M.; Liu, W.; Zang, Y.; Afzal, M.K.; Bello, S.A.; Muhammad, A.U.; Wang, C.; Li, J. Deep learning-based semantic segmentation of urban-scale 3D meshes in remote sensing: A survey. *Int. J. Appl. Earth Obs. Geoinf.* **2023**, *121*, 103365. [CrossRef]
4. Hackel, T.; Wegner, J.D.; Schindler, K. Fast semantic segmentation of 3D point clouds with strongly varying density. *ISPRS J. Photogramm. Remote Sens.* **2016**, *3*, 177–184. [CrossRef]
5. Thomas, H.; Qi, C.R.; Deschaud, J.E.; Marcotegui, B.; Goulette, F.; Guibas, L.J. Kpconv: Flexible and deformable convolution for point clouds. In Proceedings of the IEEE/CVF, Seoul, Republic of Korea, 27 October–2 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 6411–6420.
6. Selvaraju, P.; Nabail, M.; Loizou, M.; Maslioukova, M.; Averkiou, M.; Andreou, A.; Chaudhuri, S.; Kalogerakis, E. BuildingNet: Learning to label 3D buildings. In Proceedings of the IEEE/CVF, Nashville, TN, USA, 20–25 June 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 10397–10407.
7. Fu, H.; Jia, R.; Gao, L.; Gong, M.; Zhao, B.; Maybank, S.; Tao, D. 3D-future: 3D furniture shape with texture. *Int. J. Comput. Vis.* **2021**, *129*, 3313–3337. [CrossRef]
8. Mao, Z.; Huang, X.; Xiang, H.; Gong, Y.; Zhang, F.; Tang, J. Glass façade segmentation and repair for aerial photogrammetric 3D building models with multiple constraints. *Int. J. Appl. Earth Obs. Geoinf.* **2023**, *118*, 103242. [CrossRef]
9. Wang, S.; Liu, X.; Zhang, Y.; Li, J.; Zou, S.; Wu, J.; Tao, C.; Liu, Q.; Cai, G. Semantic-guided 3D building reconstruction from triangle meshes. *Int. J. Appl. Earth Obs. Geoinf.* **2023**, *119*, 103324. [CrossRef]
10. Landrieu, L.; Boussaha, M. Point cloud oversegmentation with graph-structured deep metric learning. In Proceedings of the IEEE/CVF, Seoul, Republic of Korea, 27 October–2 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 7440–7449.
11. Hui, L.; Yuan, J.; Cheng, M.; Xie, J.; Zhang, X.; Yang, J. Superpoint network for point cloud oversegmentation. In Proceedings of the IEEE/CVF, Nashville, TN, USA, 20–25 June 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 5510–5519.
12. Afzal, M.K.; Adam, J.M.; Afzal, H.R.; Zang, Y.; Bello, S.A.; Wang, C.; Li, J. Discriminative feature abstraction by deep L2 hypersphere embedding for 3D mesh CNNs. *Inf. Sci.* **2022**, *607*, 1158–1173. [CrossRef]
13. Ulku, I.; Akagündüz, E. A survey on deep learning-based architectures for semantic segmentation on 2d images. *Appl. Artif. Intell.* **2022**, *36*, 2032924. [CrossRef]
14. Rook, M. Automatic Thematic and Semantic Classification of 3D City Models. Master's Thesis, TU Delft, Delft, The Netherlands, 2016.
15. Guo, Y.; Wang, H.; Hu, Q.; Liu, H.; Liu, L.; Bennamoun, M. Deep learning for 3d point clouds: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 4338–4364. [CrossRef]
16. Luo, Y.; Wang, H.; Lv, X. End-to-End Edge-Guided Multi-Scale Matching Network for Optical Satellite Stereo Image Pairs. *Remote Sens.* **2024**, *16*, 882. [CrossRef]

17. Shen, S.; Xia, Y.; Eich, A.; Xu, Y.; Yang, B.; Stilla, U. SegTrans: Semantic Segmentation With Transfer Learning for MLS Point Clouds. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 6501105. [CrossRef]
18. Xia, Y.; Wu, Q.; Li, W.; Chan, A.B.; Stilla, U. A lightweight and detector-free 3d single object tracker on point clouds. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 5543–5554. [CrossRef]
19. Li, X.; Liu, Y.; Xia, Y.; Lakshminarasimhan, V.; Cao, H.; Zhang, F.; Stilla, U.; Knoll, A. Fast and deterministic (3 + 1) DOF point set registration with gravity prior. *ISPRS J. Photogramm. Remote Sens.* **2023**, *199*, 118–132. [CrossRef]
20. Huang, J.; Zhou, Y.; Niessner, M.; Shewchuk, J.R.; Guibas, L.J. Quadriflow: A scalable and robust method for quadrangulation. In *Computer Graphics Forum*; Wiley Online Library: Hoboken, NJ, USA, 2018; Volume 37, pp. 147–160.
21. Huang, J.; Zhang, H.; Yi, L.; Funkhouser, T.; Nießner, M.; Guibas, L.J. Texturenet: Consistent local parametrizations for learning from high-resolution signals on meshes. In Proceedings of the IEEE/CVF, Seoul, Republic of Korea, 27 October–2 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 4440–4449.
22. Yang, Y.; Liu, S.; Pan, H.; Liu, Y.; Tong, X. PFCNN: Convolutional neural networks on 3D surfaces using parallel frames. In Proceedings of the IEEE/CVF, Seoul, Republic of Korea, 27 October–2 November 2019; IEEE: Piscataway, NJ, USA, 2020; pp. 13578–13587.
23. Li, Y.; Wu, J.; Liu, H.; Ren, J.; Xu, Z.; Zhang, J.; Wang, Z. Classification of Typical Static Objects in Road Scenes Based on LO-Net. *Remote Sens.* **2024**, *16*, 663. [CrossRef]
24. Wilk, Ł.; Mielczarek, D.; Ostrowski, W.; Dominik, W.; Krawczyk, J. Semantic urban mesh segmentation based on aerial oblique images and point clouds using deep learning. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2022**, *43*, 485–491. [CrossRef]
25. Zhang, R.; Zhang, G.; Yin, J.; Jia, X.; Mian, A. Mesh-based DGCNN: Semantic Segmentation of Textured 3D Urban Scenes. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 4402812. [CrossRef]
26. Geng, J.; Yu, K.; Sun, M.; Xie, Z.; Huang, R.; Wang, Y.; Zhao, Q.; Liu, J. Construction and Optimisation of Ecological Networks in High-Density Central Urban Areas: The Case of Fuzhou City, China. *Remote Sens.* **2023**, *15*, 5666. [CrossRef]
27. Peng, B.; Yang, J.; Li, Y.; Zhang, S. Land-Use Optimization Based on Ecological Security Pattern—A Case Study of Baicheng, Northeast China. *Remote Sens.* **2023**, *15*, 5671. [CrossRef]
28. Mnih, V.; Heess, N.; Graves, A.; Kavukcuoglu, K. Recurrent models of visual attention. In Proceedings of the NeurIPS, Montreal, QC, Canada, 8–13 December 2014; Curran Associates, Inc.: New York, NY, USA, 2014; Volume 27.
29. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE/CVF, Salt Lake City, UT, USA, 18–23 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 7132–7141.
30. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of the ECCV, Munich, Germany, 8–14 September 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 3–19.
31. Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; Wei, Y. Deformable Convolutional Networks. In Proceedings of the ICCV, Venice, Italy, 22–29 October 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 764–773.
32. Hu, H.; Gu, J.; Zhang, Z.; Dai, J.; Wei, Y. Relation networks for object detection. In Proceedings of the IEEE/CVF, Salt Lake City, UT, USA, 18–23 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 3588–3597.
33. Yuan, Y.; Chen, X.; Wang, J. Object-contextual representations for semantic segmentation. In Proceedings of the ECCV, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 173–190.
34. Geng, Z.; Guo, M.H.; Chen, H.; Li, X.; Wei, K.; Lin, Z. Is attention better than matrix decomposition? *arXiv* **2021**, arXiv:2109.04553.
35. Guo, M.H.; Xu, T.X.; Liu, J.J.; Liu, Z.N.; Jiang, P.T.; Mu, T.J.; Zhang, S.H.; Martin, R.R.; Cheng, M.M.; Hu, S.M. Attention mechanisms in computer vision: A survey. *Comput. Vis. Media* **2022**, *8*, 331–368. [CrossRef]
36. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the NeurIPS, Long Beach, CA, USA, 4–9 December 2017; Curran Associates, Inc.: New York, NY, USA, 2017; Volume 30.
37. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
38. Wang, X.; Girshick, R.; Gupta, A.; He, K. Non-local neural networks. In Proceedings of the IEEE/CVF, Salt Lake City, UT, USA, 18–23 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 7794–7803.
39. Fu, J.; Liu, J.; Tian, H.; Li, Y.; Bao, Y.; Fang, Z.; Lu, H. Dual attention network for scene segmentation. In Proceedings of the IEEE/CVF, Seoul, Republic of Korea, 27 October–2 November 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 3146–3154.
40. Zhang, H.; Goodfellow, I.; Metaxas, D.; Odena, A. Self-attention generative adversarial networks. In Proceedings of the ICML, Long Beach, CA, USA, 9–15 June 2019; PMLR: Cambridge, MA, USA, 2019; pp. 7354–7363.
41. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
42. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF, Nashville, TN, USA, 20–25 June 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 10012–10022.
43. Wang, W.; Xie, E.; Li, X.; Fan, D.P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; Shao, L. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In Proceedings of the IEEE/CVF, Nashville, TN, USA, 20–25 June 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 568–578.

44. Basu, S.; Gallego-Posada, J.; Viganò, F.; Rowbottom, J.; Cohen, T. Equivariant mesh attention networks. *arXiv* **2022**, arXiv:2205.10662.
45. Han, X.; Gao, H.; Pfaff, T.; Wang, J.X.; Liu, L.P. Predicting physics in mesh-reduced space with temporal attention. *arXiv* **2022**, arXiv:2201.09113.
46. Milano, F.; Loquercio, A.; Rosinol, A.; Scaramuzza, D.; Carlone, L. Primal-dual mesh convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 952–963.
47. Yuan, Y.; Huang, L.; Guo, J.; Zhang, C.; Chen, X.; Wang, J. OCNet: Object Context Network for Scene Parsing. *arXiv* **2018** , arXiv:1809.00916.
48. Guo, M.H.; Lu, C.Z.; Liu, Z.N.; Cheng, M.M.; Hu, S.M. Visual attention network. *Comput. Vis. Media* **2023**, *9*, 733–752. [CrossRef]
49. Berg, M.; Cheong, O.; Kreveld, M.; Overmars, M. *Computational Geometry: Algorithms and Applications*; Springer: Berlin/Heidelberg, Germany, 2008.
50. Australia, L.G. [Dataset] Aerial Largescale 3D Mesh Model Dataset of the City of Adelaide Australia. 2022. Available online: https://data.sa.gov.au/data/dataset/3d-model (accessed on 6 March 2024).
51. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.
52. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Volume 30.
53. Liu, W.; Lai, B.; Wang, C.; Cai, G.; Su, Y.; Bian, X.; Li, Y.; Chen, S.; Li, J. Ground camera image and large-scale 3-D image-based point cloud registration based on learning domain invariant feature descriptors. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *14*, 997–1009. [CrossRef]
54. Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; Chen, B. Pointcnn: Convolution on x-transformed points. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018; Volume 31.
55. Li, Y.; Xu, L.; Rao, J.; Guo, L.; Yan, Z.; Jin, S. A Y-Net deep learning method for road segmentation using high-resolution visible remote sensing images. *Remote Sens. Lett.* **2019**, *10*, 381–390. [CrossRef]
56. Feng, Y.; Feng, Y.; You, H.; Zhao, X.; Gao, Y. Meshnet: Mesh neural network for 3d shape representation. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 8279–8286.
57. Gröger, G.; Kolbe, T.H.; Nagel, C.; Häfele, K.H. *OGC City Geography Markup Language (CityGML) Encoding Standard*; Open Geospatial Consortium: Arlington, VA, USA, 2012; p. 344.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.