



Article

An Efficient and General Framework for Aerial Point Cloud Classification in Urban Scenarios

Emre Özdemir ^{1,2,*} , Fabio Remondino ² and Alessandro Golkar ¹¹ Skolkovo Institute of Technology (Skoltech), 143026 Moscow, Russia; a.golkar@skoltech.ru² 3D Optical Metrology (3DOM) Unit, Bruno Kessler Foundation (FBK), 38123 Trento, Italy; remondino@fbk.eu

* Correspondence: emre.ozdemir@skoltech.ru

Abstract: With recent advances in technologies, deep learning is being applied more and more to different tasks. In particular, point cloud processing and classification have been studied for a while now, with various methods developed. Some of the available classification approaches are based on specific data source, like LiDAR, while others are focused on specific scenarios, like indoor. A general major issue is the computational efficiency (in terms of power consumption, memory requirement, and training/inference time). In this study, we propose an efficient framework (named TONIC) that can work with any kind of aerial data source (LiDAR or photogrammetry) and does not require high computational power while achieving accuracy on par with the current state of the art methods. We also test our framework for its generalization ability, showing capabilities to learn from one dataset and predict on unseen aerial scenarios.

Keywords: aerial point cloud; classification; AI; machine learning; deep learning



Citation: Özdemir, E.; Remondino, F.; Golkar, A. An Efficient and General Framework for Aerial Point Cloud Classification in Urban Scenarios. *Remote Sens.* **2021**, *13*, 1985. <https://doi.org/10.3390/rs13101985>

Academic Editor: Daniela Poli

Received: 26 March 2021

Accepted: 14 May 2021

Published: 19 May 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, point cloud processing techniques are extensively investigated by the research community for various applications [1,2]. Among these, aerial point cloud classification methods hold an important place, as assigning a meaning to 3D points allows for the widespread use of such geospatial information. There are many studies presented in the literature focusing on a semantic interpretation of 3D point clouds based on different techniques [3–5] and for various approaches [6–9]. To the best of our knowledge, many of the current 3D classification solutions are confined to either specific data (e.g., only LiDAR) or scenarios (indoor vs. outdoor, terrestrial vs. aerial). This is due to the complexity of the 3D classification process, the different data structure, the need of specific training data, as well as generalization problems.

This paper introduces our work dedicated to aerial point cloud classification. A framework (TONIC: efficienT classification Of urbaN poInt Clouds) was developed with the motivation to realize an efficient (in terms of power consumption, memory requirement, and training/inference time), reliable and generalizable method for the semantic enrichment of urban point clouds. Such enriched point cloud could be preparatory for 3D building modeling [10], change detection [11], planning [12], etc. The presented framework can cope with point clouds from any data source (LiDAR or aerial photogrammetry), any point cloud density and even absence of RGB information.

Our major contributions can be summarized as follows:

1. analyze the effects of point clouds' density characteristics (overall density and density variations) for point cloud classification, and find an optimal overall density based on classification accuracy (Section 3.1.1);
2. deliver a novel point cloud classification approach based on the combination of classical Machine Learning (ML) and Deep Learning (DL) techniques, i.e., using a shallow and custom-designed convolutional neural network (CNN) supported by handcrafted features (Section 3.1.2);

3. classify heterogeneous point clouds produced by different sensors and methods (Section 3.2) in an efficient and low-consuming manner (Section 4.3);
4. generalize the method to unseen data (Section 4.4).

Based on the way we treat the hand-crafted features with CNNs, we consider our work to be categorized as an architectural innovation. To our knowledge, there are no other methods using our approach (Section 3.1.3), where we generate patches (a 2D matrix per point) using extracted features and coordinates of the neighboring points and process them as images aiming to predict a class label per point (represented by a patch). Therefore, we consider our approach as a new relationship between aerial 3D point clouds and CNNs [13].

In the following sections, related works are presented in Section 2, the workflow is outlined in Section 3, whereas results and validations are presented in Section 4. Discussions and conclusions will wrap up the article.

2. Related Work

Artificial Intelligence (AI) usage in data classification has evolved throughout the years (Figure 1), and this trend is clearly visible in point cloud processing [14–16]. AI-based processing of point clouds involves segmentation, classification, and object detection procedures. ML methods do not require large training data in general. However, more and more point cloud data becoming available, facilitating the development of new DL approaches [2,16,17]. DL methods commonly outperform ML methods in various fields and tasks with the capabilities of representation learning [17]. The benefits include more generalized model with possible higher accuracy compared to ML. In some cases, hand-crafted features are included to give a boost to the DL method [18,19], yet it is not a common trend in the AI community as seen in the literature [2,15,20,21].

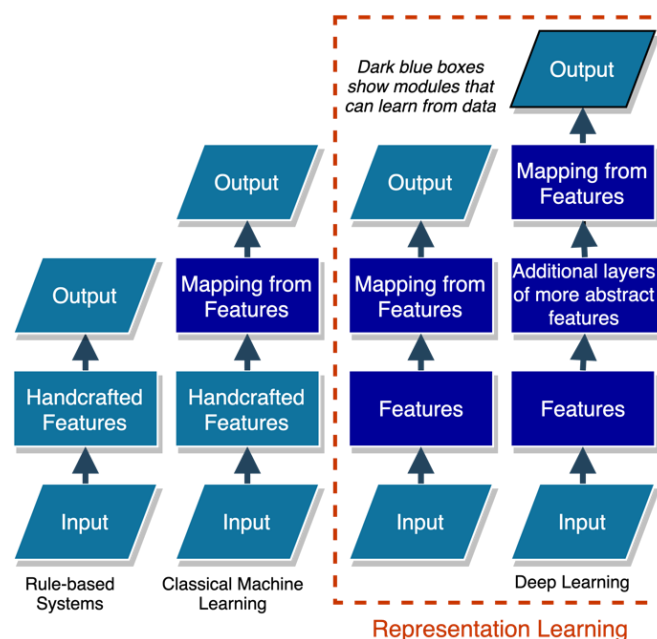


Figure 1. AI approaches for data classification (adapted from Goodfellow et al. [22]).

As our method can be categorized in between ML and DL, the next sections report some related works in both fields.

2.1. Classical Machine Learning (ML)

ML methods for point cloud classification are focused on labeling each point individually via their feature vectors. These features are extracted using neighboring points and they are pre-defined by the person who handcrafts them. In many cases, the computational efficiency of the ML algorithms is helpful, and in this way, some problems can be solved

with high accuracy [19]. However, there is a limit to the extent of feature handcrafting and therefore how good the results can be [23].

Weinmann et al. [24] analyzed features' impact in the classification of terrestrial laser scanning (TLS) data, showing how few and versatile features can outperform the increased number of features. Hackel et al. [25] presented a TLS classification method able to handle a varying point density. Thomas et al. [26] focused on utilizing multiscale spherical neighborhoods for indoor and outdoor LiDAR point clouds. Zhang et al. [27] developed a classification method that combines surface growing along with support vector machine for a classification based on segments and implemented connected components-based refinement to cope with noises in classification. Li et al. [28] applied label smoothing in order to improve classification results.

The current challenges and limitations of ML methods for point cloud classification include feature engineering (especially for irregular/noisy point clouds), solving complex problems, being able to exploit more data and transfer learning adaptation. We consider these challenges and limitations to be rooted in the nature of ML methods.

2.2. Deep Learning (DL)

With the recent developments in technology, DL is becoming more and more popular for different data processing necessities [29]. Being able to learn from data (Figure 1) and thanks to the increasing number of available datasets, DL methods are the current state-of-the-art in many applications, including point cloud classification.

There are various studies focusing on different perspectives of DL-based point cloud processing, including 3D shape classification, 3D object detection [30], and 3D point cloud classification [31]. 3D point cloud classification methods have been developed with different approaches based on graph recursive neural network (RNN), point convolution, point-wise multi-layer perceptron (MLP), etc. [17].

Aerial point cloud classification has been studied by many researchers with varying approaches: Qi et al. developed PointNet++ [32], a semantic segmentation approach, where the point cloud is processed with varying neighborhoods and local geometry is learned by the network as well as global geometry. Yousefhusien et al. [33] developed a 1D-CNN based method which can learn global and local geometric features from a given point cloud for classification. Özdemir et al. proposed the combination of handcrafted features with DL to boost the classification process [34]. Li et al. [21] developed a geometry-attentional deep neural network (DNN) based on geometry-aware convolutions. Wen et al. [35] developed a graph-attention based approach which includes edge and density attentions as well as graph global attention. Huang et al. [36] built a DNN solution based on PointNet++ including hierarchical data augmentation. Li et al. [37] proposes Dance-Net, which introduces a density-aware convolution module capable of approximating typical convolutions on an irregular 3D point cloud. Winiwarter et al. [38] developed their method based on multi-scale implementation of PointNet++. Chen et al. [39] proposed a network using PointNet++ architecture with modified local and global feature extraction capabilities as well as a focal loss to improve the performance of the original network. Thomas et al. [40] proposed kernel point convolution (KPConv) that can handle point clouds without any transitional representations.

The current challenges and limitations of DL methods for point cloud classification include data irregularity, uneven distribution and density, noise/outlier presence, availability of large training sets, lack of generalization to various contexts, and explainability. The reasons for such challenges are due to the intrinsic nature of DL methods, the general training on specific contexts and datasets, and their lack of clear transparent explanation and reasons for a certain result or failure.

3. Method and Data

3.1. Methodology

The methodology builds upon [34,41] and refines its performance, adding reliability, computational efficiency, and generalization capabilities. In particular, improvements in the feature extraction approach and deep neural network design are afterwards reported.

During our development and testing, we used publicly available libraries including Point Cloud Library [42] for point cloud processing, TensorFlow v2.5.0-rc1 with Keras [43,44], NumPy [45], and Scikit-learn [46] packages for classical machine learning.

3.1.1. Point Cloud Downsampling

One of the key characteristics of a point cloud is its overall density. The overall density may differ based on data acquisition (i.e., flight altitude, camera, and lens specifications, LiDAR sensors, etc.) or data processing (i.e., dense image matching parameter settings, filtering, etc.) options. A point cloud's overall density may be as low as 1 pts/m² or as high as 1000 pts/m², which hampers the classification needs. For instance, with an overall density of ~2 pts/m², it becomes difficult or even impractical to seek highly detailed classes, such as traffic lights, cars or sidewalk bars. In addition to overall density, another key characteristic is the density variation within the point cloud. Like overall density, the density variation can be as low as 1 pts/m² or as high as 1000 pts/m². The higher the density variation, the less consistent the level of detail is expected within the point cloud. The density variation has significant impact on results for our approach as we use handcrafted features (Section 3.1.2). Higher density variation causes less consistency for the features.

Considering these issues, we adapted a voxel-based downsampling approach which offer the following main advantages:

1. Speed-up the entire process significantly due to data reduction,
2. Make a better use of features given the consistent density within the point cloud due to lower density variations,
3. Achieve similar overall density characteristics among different point clouds and favor generalization possibilities,
4. Noise reduction in the point cloud due to voxel grid filtering.

The downsampling is done with Point Cloud Library's built-in voxel grid filter, which downsamples points falling in the same voxel by computing their centroid. We defined the voxel sizes for downsampling as the multiplication of points resolution and leaf coefficient. The minimum, mean, median, maximum, and standard deviation of number of nearest neighboring points are computed with a k-nearest neighborhood (knn) within 1.45 times the points resolution. This 1.45 coefficient is decided based on the neighboring distances in an image/patch (see Section 3.1.3).

3.1.2. Multi-Scale Feature Extraction

Neighborhood selection has always been a challenging decision when it comes to feature extraction. There are alternative approaches in the literature tackling this task [25,47–49]. In our case, instead of using a single search radius or a constant number of points (k-nearest neighbors), a multi-scale feature extraction for different neighborhood coverages is implemented. Normally, we apply three different radii in order to tackle with the different geometric aspects of the classes.

The extracted features include eigenvalues ($\lambda_1 > \lambda_2 > \lambda_3$) derived from the principal component analysis implementation of Point Cloud Library [42], eigenvectors-based surface normal estimations, covariance features (linearity, sphericity, omnivariance), as well as geometrically computed features (local elevation change, local planarity, vertical angle, and height above ground). The covariance features are computed following [24] and the others are computed based on direct geometrical computations rather than principal component analysis (Table 1).

Table 1. Employed handcrafted features.

Linearity	$(\lambda_1 - \lambda_2) / \lambda_1$
Sphericity	λ_2 / λ_1
Omnivariance	$(\lambda_1 * \lambda_2 * \lambda_3)^{1/3}$
Local Elevation Change	$z_{max} - z_{min}$
Local Planarity	$\frac{1}{n} \sum_{k=0}^n L_2(p_k, \vec{P})$
Vertical Angle	$\cos^{-1} \left(\frac{\vec{P}_{xy} \cdot \vec{p}}{(\ \vec{P}_{xy}\) * (\ \vec{p}\)} \right)$
Height Above Ground	$pr_z - pl_z$

The employed non-eigen features include: local elevation change (i.e., the difference in the minimum and maximum z -coordinates in the neighborhood); local planarity (i.e., the average distance between the neighboring points (p_k) to their best-fit plane (\vec{P})); vertical angle (i.e., the angle between the normal vector of a point (\vec{p}) and xy -plane (\vec{P}_{xy})); height above ground (i.e., the difference between z -coordinates of the point (pr_z) and the possible lowest point (pl_z)). The possible lowest point is the point which is hypothetically representing the ground and is extracted as given in Algorithm 1, in which we used the k -d tree algorithm implemented in the Point Cloud Library [42].

Algorithm 1: Possible lowest point identification

Input: 3D point cloud

Initialization: Iterate through the input point cloud and get minimum z -coordinate (z_{min})

- 1: **for** each real point ($pr: \{pr_x, pr_y, pr_z\}$) in the cloud **do**
 - 2: generate a pseudo point (pp) with coordinates pr_x, pr_y, z_{min}
 - 3: find the nearest neighbor point for pp in the input cloud using k -d tree
 - 4: retrieve the z -coordinate of the found point (fp_z)
 - 5: update the z -coordinate of the pp with fp_z
 - 6: search for the nearest neighbor point for pp in the input cloud using k -d tree, possible lowest point ($pl: \{pl_x, pl_y, pl_z\}$) is the found point
 - 7: **end for**
-

3.1.3. Classification

The developed classification framework (Figure 2) relies on the abovementioned handcrafted features as well as self-learned ones. Besides, if available, we also utilize additional sensor data as features (i.e., intensity, number of returns and return number for LiDAR clouds, color information for photogrammetric clouds) and local neighborhood point coordinates. The framework includes both 2DCNN and a 3DCNN which are applied depending on the data and tasks as reported in the discussions (Section 5).

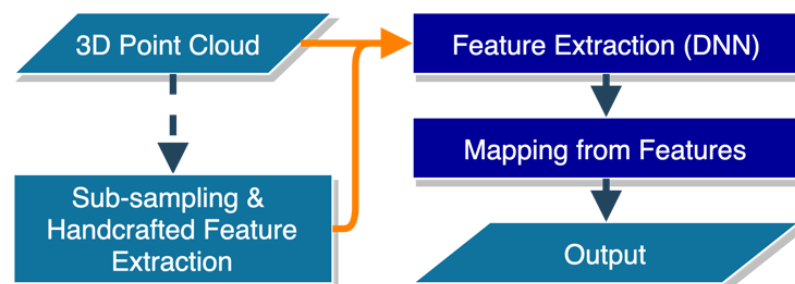


Figure 2. The proposed classification framework based on deep learning (modules that can learn from the given data are shown with dark blue boxes).

We include features and coordinates of the local neighbor points designing a data matrix (2D patch) for each point (Figure 3). The included coordinate values are first zero-centered around the point of interest, then scaled to the range of $[0,1]$ by dividing to maximum values and clipping, forming the patch-wise scaled coordinates. We sort the patch rows based on the coordinates, which we observed to provide fractionally better results. The 2D patch is then treated as an image, and classification is handled with an object detection approach.

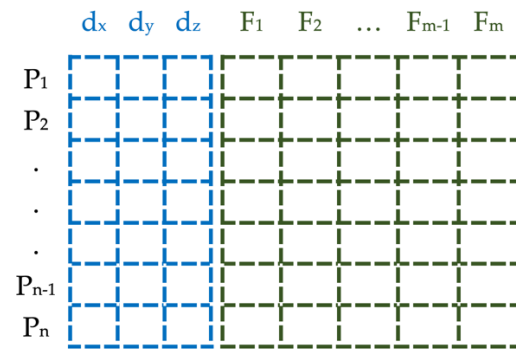


Figure 3. 2D patches generated for each 3D point of the cloud: P_n denotes points, $d_{x,y,z}$ denotes patch-wise scaled coordinates (blue cells) and F_m denotes the available/computed features (green cells).

The employed network structure is a 2DCNN architecture as shown in Figure 4. The network receives the 2D patches described in Figure 3 and processes them like an image-based object detection network outputting the class probabilities.

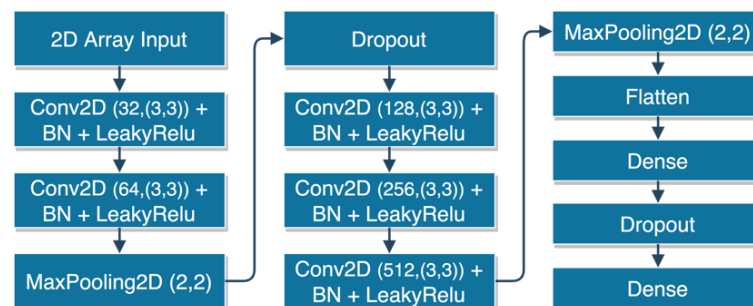


Figure 4. Our 2DCNN structure (BN stands for batch normalization layer).

Contrary to rendering-based methods or voxel-based methods [17], our CNN methods use pseudo images as shown in Figure 5.

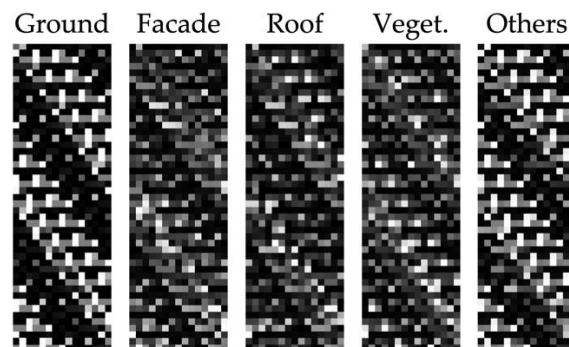


Figure 5. An example of 2D patch visualization (transposed for better picturing).

Our framework accommodates also a 3DCNN as it has shown good performance for image classification task and it takes advantage of inter-channel correlation as well as spatial correlation [50]. In order to use the aforementioned 2D patches in a 3DCNN, we fold them along the features' axis (vertical axis in Figure 5) and produce 3D patches. In this way, a 2D patch with dimensions of e.g., 40×15 becomes $40 \times 5 \times 3$. The applied 3DCNN architecture (Figure 6) is a slightly modified version of the 2DCNN architecture presented in Figure 4. Visualizing our 3D patches gives the color images shown in Figure 7.

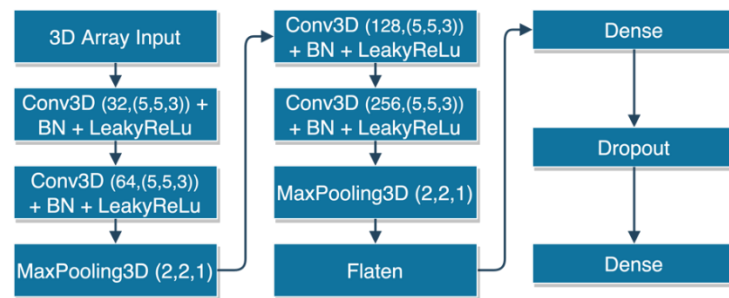


Figure 6. Our 3DCNN structure.

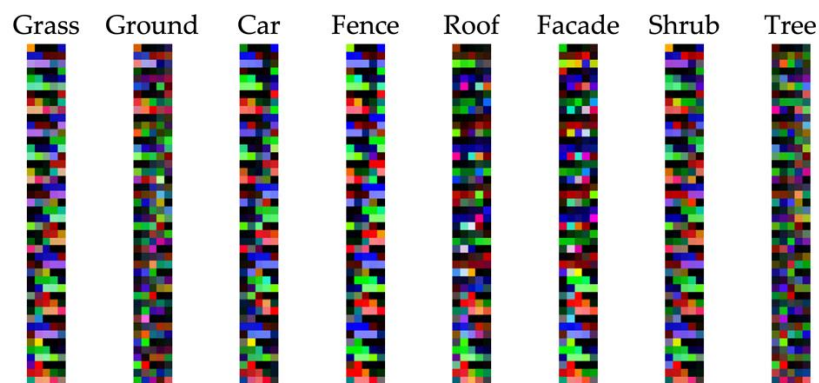


Figure 7. An example of generated 3D patches (transposed for better picturing).

3.2. Employed Data

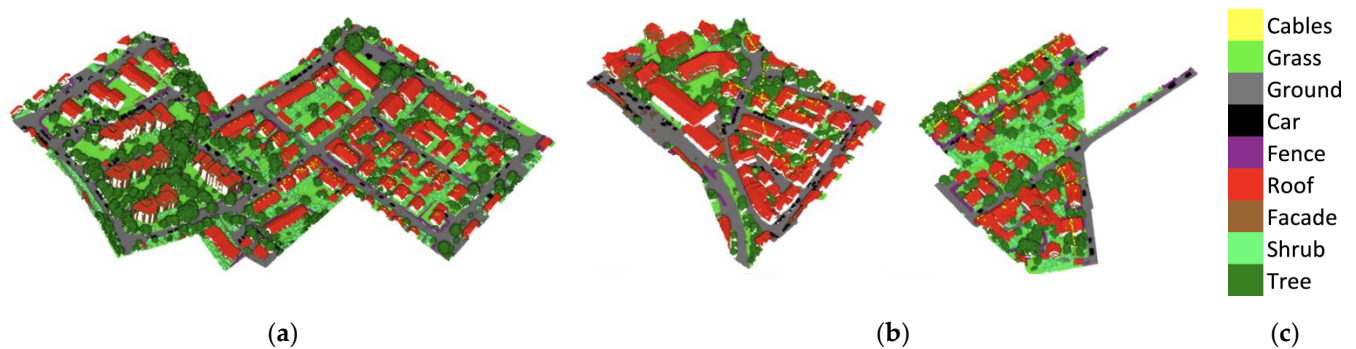
Hereafter, a large variety of dataset is presented (Table 2). These data are all used to evaluate the developed classification methodology and they include: ISPRS Vaihingen, DALES, LASDU, Bordeaux, and 3DOMCity. All point clouds have ground truth labels, and except for Bordeaux, all the employed point clouds are publicly available. Each dataset has different characteristics in terms of density, resolution (i.e., average distance between a point and its nearest neighbor), source, available sensor data, and classes. We consider the density as the average number of points per m^2 on the ground, whereas resolution is the average of the distances between each point and its nearest neighbor.

3.2.1. ISPRS 3D Semantic Labeling Contest Dataset (ISPRS Vaihingen)

ISPRS 3D Semantic Labeling Contest Dataset [51] has been one of the most widely used datasets for urban-scale point cloud classification benchmarking. The dataset includes LiDAR points, intensities, number of returns, return numbers as well as near infrared (NIR) orthophoto including NIR, green and blue channels (Figure 8).

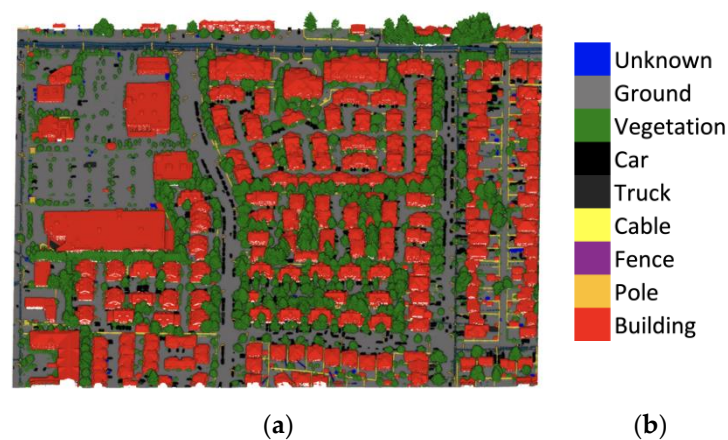
Table 2. Summary of the used datasets (L: LiDAR, OP: Oblique Photogrammetry, lab: laboratory).

Dataset (Year)	Source	Points	Density (pts/m ²)	Resolution (m)	Coverage (m × m) (Tiles)	RGB	Classes
ISPRS Vaihingen (2013) [51]	L	1,165,598	4	0.258	Train: 383 × 405 Test: 374 × 402	NIR	9
DALES (2020) [52]	L	497,632,442	35	0.116	500 × 500 (40 tiles)	No	8
LASDU (2020) [53,54]	L	3,080,856	3	0.484	1071 × 1285	No	5
Bordeaux (2020) [55]	L + OP	10,230,941	25	0.173	704 × 739	Yes	5
3DOMCity (2019) [56]	OP (lab.)	22,825,024	14000	0.158mm	0.813 × 0.811	Yes	6

**Figure 8.** ISPRS Vaihingen point cloud with its nine classes. (a) Training, (b) test sections and (c) legend.

3.2.2. DALES Dataset

The Dayton Annotated LiDAR Earth Scan (DALES) dataset [52] is a new, large-scale benchmark for the semantic segmentation of point clouds. The dataset includes number of returns and return numbers as LiDAR features, yet it lacks LiDAR intensity and color information (Figure 9).

**Figure 9.** (a) DALES point cloud with its eight classes (b) legend.

3.2.3. LASDU Dataset

The Large-Scale Aerial LiDAR Point Clouds of Highly Dense Urban Areas (LASDU) dataset [53,54]. The dataset provides intensity, number of returns and return numbers as LiDAR features, but it does not include any color information (Figure 10).

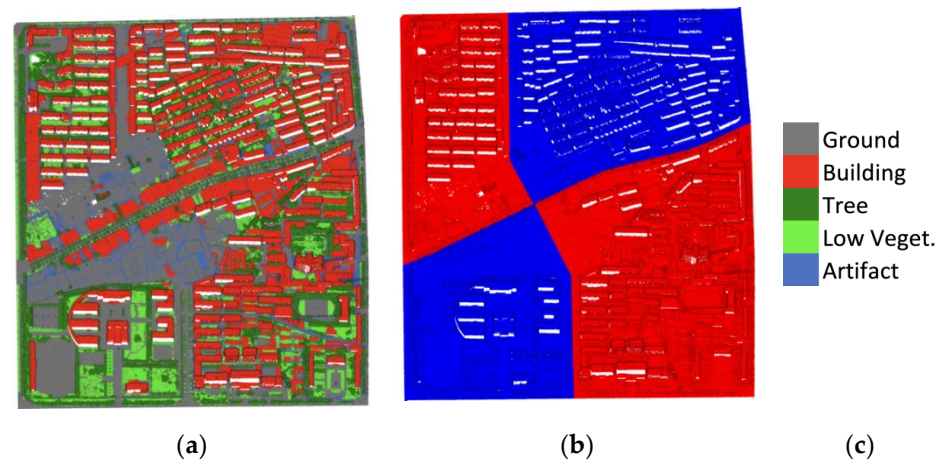


Figure 10. LASDU point cloud with its five classes (a). Training (red) and test (blue) sections (b). (c) Legend.

3.2.4. Bordeaux Dataset

The Bordeaux dataset [55] is collected with a Leica CityMapper hybrid sensor. The data include LiDAR's intensity, number of returns and return numbers as well as colors retrieved from the photogrammetric dense cloud (Figure 11).

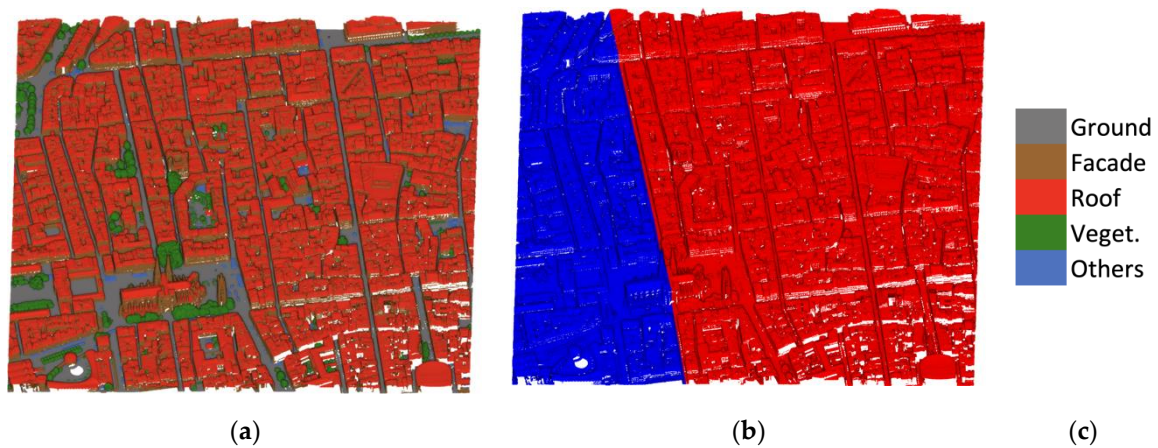


Figure 11. Bordeaux point cloud with its five classes (a). The dataset is separated (b) into training (~70%, blue) and test (~30%, red). (c) Legend.

3.2.5. 3DOMCity Benchmark

The 3DOMCity benchmark [56] includes several tasks: image orientation, dense image matching and point cloud classification. Here we use the very dense point cloud generated from oblique photogrammetry (Figure 12) for validating the presented classification framework.

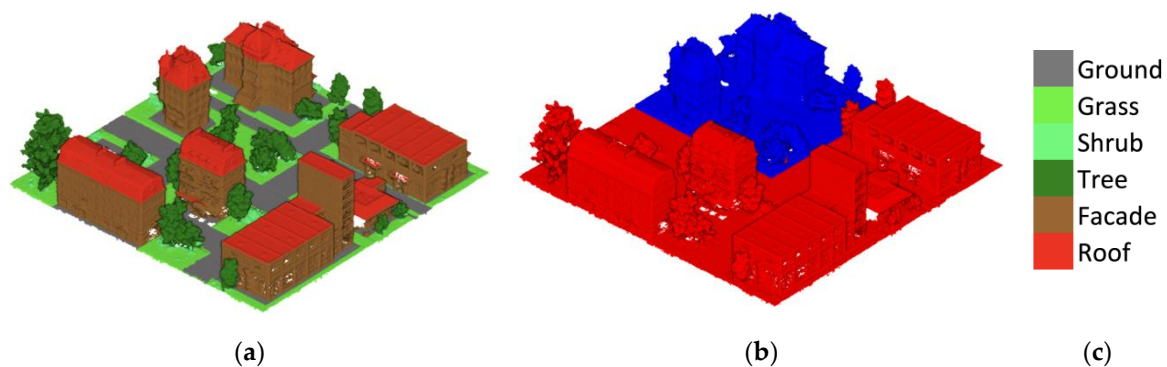


Figure 12. 3DOMCity point cloud and its six classes (a). The dataset is separated (b) into training (~70%, blue) and test (~30%, red). (c) Legend.

4. Results

In this section, quantitative results (namely F1 scores, intersection over union—IoU and overall accuracy—OA) for the proposed TONIC classification frameworks are reported, along with comparison with the state-of-the-art methods and generalization capability. Before detailing accuracy metrics for each dataset of Table 2, the developed downsampling approach and its effects are shortly presented.

For the training of our deep learning models, we preferred F1 score as loss function, and used it along with stochastic gradient descent (SGD) optimizer. We set the patience for 15 epochs for early stopping, observing the validation loss.

4.1. Point Cloud Downsampling

Downsampling—if exaggerated—can remove too many points, causing lower accuracy due to loss of detail as well as insufficient data for training a neural network. On the other hand, minimizing the downsampling will not be helpful for reaching a sufficient data reduction goal, lowering density variation (standard deviation of knn in the Table 3) and improving computational speed. Therefore, we held some experiments in order to get an overall understanding of the downsampling process. We used a random forest classifier for these experiments, due to its fast prediction capabilities and as the focus is on understanding the influence of the density on the classification. The experiments were held for ISPRS Vaihingen, LASDU and Bordeaux datasets, using various leaf coefficients for each dataset (no downsampling means leaf coefficient = 1) till resolution reaches ~1.0 m. The number of experiments therefore depends on the initial resolution of the dataset. The classification results are shared in terms of weighted F1 score, average F1 score and overall accuracy.

Table 3. Density analysis on the ISPRS Vaihingen dataset.

Leaf Coeff.	% of Points	Resol. (m)	Min. knn	Mean knn	Median knn	Max. knn	Std. knn	Weigh. F1	OA
1	100	0.258	2	11	5	106	12.99	0.788	0.788
2	50	0.434	2	4	5	15	1.51	0.798	0.796
4	20	0.732	2	4	4	13	1.31	0.776	0.788
6	11	1.07	2	4	3	13	1.37	0.768	0.770

Tests with the ISPRS dataset were held not only with training and test data but including the rest of the available tile (Table 2) in order to have a larger region rather than a combination of two relatively smaller regions. Tests (Table 3) show that the best result for this dataset was achieved with leaf coefficient of 2, i.e., a points resolution of ~0.43 m. The weighted F1 and the OA metrics do not show a major change for the next step of downsampling.

Downsampling tests on the LASDU entire dataset show that classification results (Table 4) reached a weighted F1 score fractionally better with the original dataset, while the overall accuracy is observed to be better with the downsampling with a leaf coefficient of 2.

Table 4. Density analysis on LASDU dataset.

Leaf Coeff.	% of Points	Resol. (m)	Min. knn	Mean knn	Median knn	Max. knn	Std. knn	Weigh. F1	OA
1	100	0.484	2	4	4	83	1.45	0.826	0.814
2	48	0.792	2	5	4	12	1.24	0.823	0.821
4	15	1.440	2	5	5	14	1.31	0.805	0.816

The tests with the Bordeaux dataset were also held on the entire dataset (Table 5). As the initial resolution is below 0.2 m, we skipped leaf coefficient 2 and continued with 4. The best weighted F1 score is achieved with leaf coefficient of 6, where the resolution is ~0.74 m.

Table 5. Density Analysis on Bordeaux dataset.

Leaf Coeff.	% of Points	Resol. (m)	Min. knn	Mean knn	Median knn	Max. knn	Std. knn	Weigh. F1	OA
1	100	0.173	2	6	2	61	6.01	0.930	0.928
4	23	0.528	2	4	2	14	1.44	0.922	0.942
6	12	0.737	2	4	4	12	1.31	0.940	0.940
8	8	0.946	2	4	4	12	1.32	0.935	0.935

Considering the achieved results in terms of F1 score, standard deviation for knn, and data reduction, resolution of 0.7–0.8 m can be considered for all datasets as the most suitable. The downsampling time is much lower compared to the time gained during feature extraction, model training and prediction times. Moreover, this data reduction approach is helpful for generalization purposes as it allows to get very similar density characteristics among different point clouds and to deal with large amount of data (Section 4.4).

Based on these findings, all datasets were downsampled with the same approach. The original number of points compared to downsampled versions is given in Table 6, showing reductions between 52–94%. The differences in ratios among datasets are observed to be caused mainly by the original point cloud densities.

Table 6. Number of points in each dataset before and after the downsampling procedure.

Dataset	ISPRS Vaihingen	DALES	LASDU	Bordeaux	3DOMCity
# of original points	1,165,598	497,632,442	3,080,856	10,230,941	22,825,024
# of downsampled points	236,603	27,652,837	1,465,068	1,264,690	2,075,937
% of kept points	0.203	0.056	0.476	0.124	0.091

4.2. Classification Results

In this section we report details of the classification results for each dataset with accuracy metrics in Tables 7–11 and visualization of the classes in Figures 13–17.

Table 7. Average and weighted F1 and IoU scores for 2DCNN and 3DCNN classifiers on the ISPRS Vaihingen dataset (LV: low vegetation). The OA is also reported.

	Cables	LV	Ground	Car	Fence	Roof	Facade	Shrub	Tree	Average	Weighted	OA
2DCNN-F1	0.000	0.795	0.904	0.733	0.213	0.929	0.583	0.451	0.817	0.603	0.822	0.826
3DCNN-F1	0.301	0.781	0.896	0.688	0.207	0.902	0.536	0.413	0.802	0.614	0.804	0.806
2DCNN-IoU	0.000	0.660	0.825	0.579	0.119	0.867	0.411	0.291	0.690	0.494	0.719	0.826
3DCNN-IoU	0.177	0.641	0.811	0.525	0.116	0.822	0.367	0.261	0.670	0.488	0.693	0.806

Table 8. Average and weighted F1 and IoU scores for 2DCNN and 3DCNN classifiers on the DALES dataset. The OA is also reported.

	Ground	Veget.	Car	Truck	Cable	Fence	Pole	Building	Avg.	Weighted	OA
2DCNN-F1	0.962	0.927	0.666	0.000	0.903	0.530	0.468	0.911	0.671	0.937	0.938
3DCNN-F1	0.958	0.923	0.682	0.000	0.914	0.490	0.547	0.905	0.677	0.932	0.934
2DCNN-IoU	0.926	0.863	0.499	0.000	0.823	0.360	0.306	0.837	0.577	0.884	0.938
3DCNN-IoU	0.919	0.857	0.517	0.000	0.841	0.325	0.377	0.826	0.583	0.876	0.934

Table 9. Average and weighted F1 and IoU scores for 2DCNN and 3DCNN classifiers for the LASDU dataset. The OA is also reported.

	Ground	Building	Tree	LV	Artifact	Average	Weighted	OA
2DCNN-F1	0.887	0.935	0.860	0.691	0.360	0.746	0.851	0.846
3DCNN-F1	0.885	0.915	0.858	0.673	0.322	0.730	0.840	0.837
2DCNN-IoU	0.796	0.878	0.754	0.527	0.220	0.635	0.757	0.846
3DCNN-IoU	0.793	0.843	0.751	0.507	0.192	0.617	0.741	0.837

Table 10. Average and weighted F1 and IoU scores for 2DCNN and 3DCNN classifiers for the Bordeaux dataset. The OA is also reported.

	Ground	Facade	Roof	Vegetation	Others	Average	Weighted	OA
2DCNN-F1	0.972	0.819	0.956	0.986	0.708	0.888	0.943	0.944
3DCNN-F1	0.966	0.807	0.951	0.985	0.682	0.878	0.937	0.938
2DCNN-IoU	0.945	0.694	0.916	0.972	0.548	0.808	0.897	0.944
3DCNN-IoU	0.934	0.676	0.907	0.970	0.517	0.801	0.887	0.938

Table 11. Average and weighted F1 and IoU scores for 2DCNN and 3DCNN classifiers for the 3DOMCity dataset. The OA is also reported.

	Ground	Grass	Shrub	Tree	Facade	Roof	Average	Weighted	OA
2DCNN-F1	0.945	0.936	0.798	0.878	0.864	0.906	0.888	0.889	0.889
3DCNN-F1	0.954	0.938	0.777	0.864	0.866	0.887	0.881	0.883	0.883
2DCNN-IoU	0.897	0.880	0.664	0.782	0.761	0.828	0.802	0.802	0.889
3DCNN-IoU	0.913	0.883	0.635	0.760	0.763	0.796	0.792	0.793	0.883

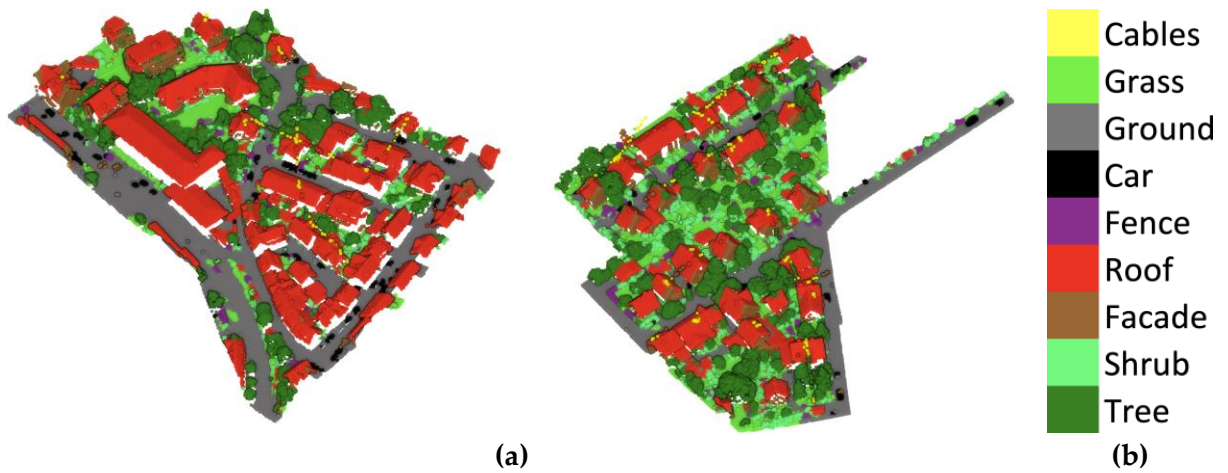


Figure 13. (a) Classification result of the ISPRS Vaihingen dataset with the 2DCNN approach. (b) Legend.

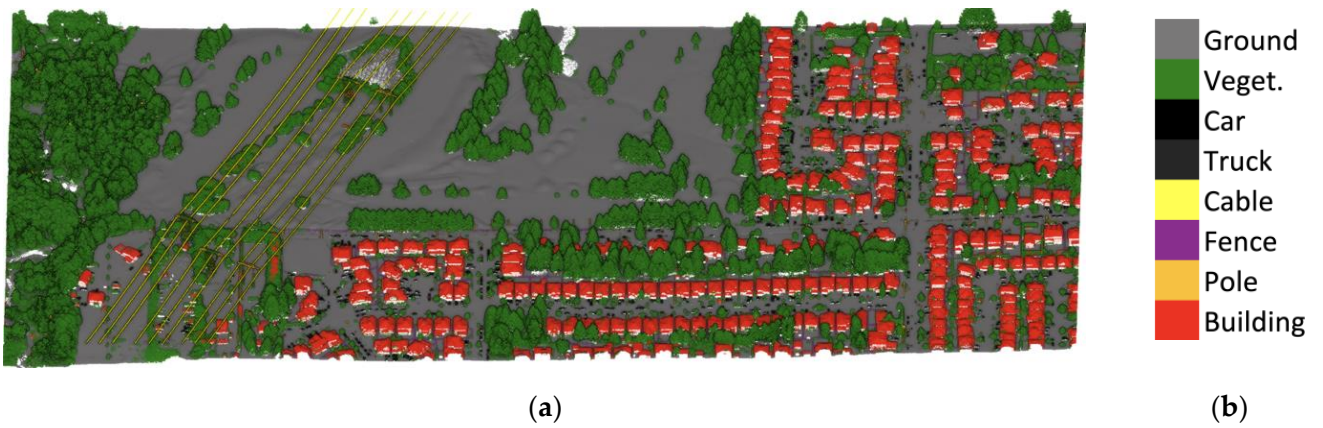


Figure 14. (a) Classification result of both DALES tiles with our 3DCNN, (b) legend.

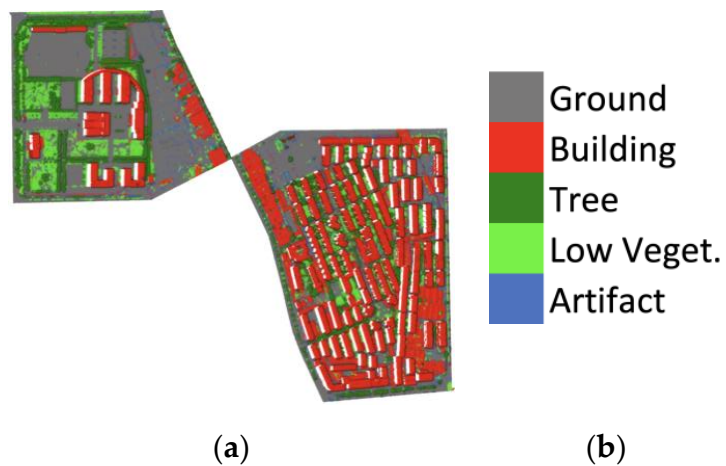


Figure 15. (a) 3DCNN classification results on the testing area of the LASDU dataset, (b) legend.



Figure 16. (a) Classification result on the testing area of the Bordeaux dataset with our 2DCNN, (b) legend.

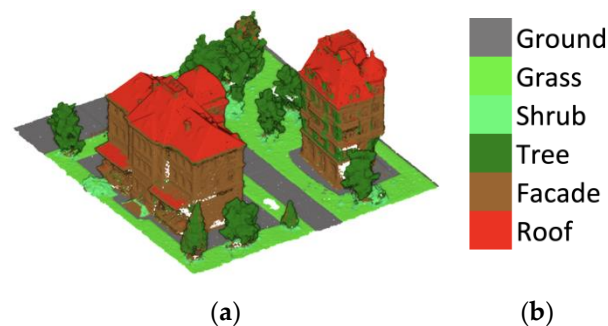


Figure 17. (a) Classification result on testing area of the 3DOMCity dataset with our 2DCNN, (b) legend.

4.3. Comparisons with the State-of-the-Art

Our TONIC framework was compared to available state-of-the-art approaches in terms of performances and accuracy. We report the accuracy metrics and training times as available in the original publications.

In addition to the point cloud classification methods, we also compared our framework with EfficientNetB7, which is not only efficient but also provides high accuracy compared to the current state-of-the-art [57]. The network is provided within TensorFlow library, and we applied it as shown in Figure 18. Due its design, the EfficientNetB7 cannot process images smaller than 32×32 pixels. Therefore, we zero padded our patches to fill that requirement.

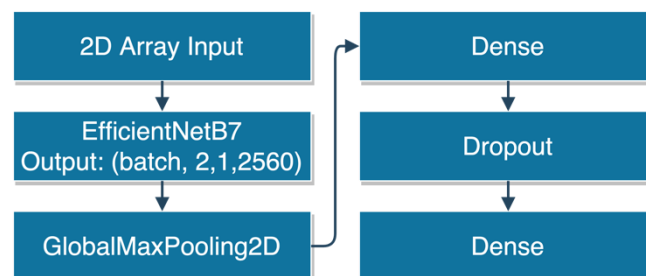


Figure 18. Implementation of EfficientNetB7 from TensorFlow library.

Table 12 reports a comparison for the ISPRS Vaihingen dataset in terms of computational efficiency and accuracy. It can be seen that our models can achieve accuracies on par with the current state-of-the-art methods, while requiring less power consumption, memory and training/inference time. In terms of inference performance for this dataset, our method takes less than 10 s on Nvidia RTX 2080Ti GPU. With the same hardware

configuration, EfficientNetB7 takes 23 s. Feature extraction for the dataset takes less than 10 s on Intel i9-8950HK Mobile CPU.

Table 12. Performance comparison between our methods and recent papers, ordered by OA. (* difference from the highest OA score in the table).

Method	GPU TFLOPS FP32	Training Time (Hours)	GPU Watt	GPU Memory	OA	GPU
Li et al. [37]	8.73	10	250	24 GB	0.839	Nvidia Tesla K80
Li et al. [21]	2 × 12.15	7	2 × 250	2 × 12 GB	0.835	2 × Nvidia Titan Xp
Wen et al. [35]	12.15	10	250	12 GB	0.832	Nvidia Titan Xp
Chen et al. [39]	14.13	2	250	32 GB	0.832	Nvidia Tesla V100
EfficientNetB7 [57]	13.45	1	250	11 GB	0.748 (−9.1% *)	Nvidia RTX 2080Ti
Ours (2DCNN)	13.45	0.15	250	11 GB	0.826 (−1.3% *)	Nvidia RTX 2080Ti
Ours (3DCNN)	13.45	0.5	250	11 GB	0.806 (−3.3% *)	Nvidia RTX 2080Ti

Other accuracy comparisons are executed using the DALES dataset and comparing our networks with IOU metrics with respect to current state-of-the-art methods (Table 13). KPConv and PointNet++ both outperform our methods by 3–4% in terms of OA.

Table 13. OA scores for 2DCNN and 3DCNN classifiers for DALES dataset.

	Ground	Vegetation	Car	Truck	Cable	Fence	Pole	Building	OA
KPConv [40]	0.971	0.941	0.853	0.419	0.955	0.635	0.750	0.966	0.978
PointNet++ [32]	0.941	0.912	0.754	0.303	0.799	0.462	0.400	0.891	0.957
Ours (2DCNN)	0.926	0.863	0.499	0.000	0.823	0.360	0.306	0.837	0.938
Ours (3DCNN)	0.919	0.857	0.517	0.000	0.841	0.325	0.377	0.826	0.934

Finally, accuracy comparisons with the LASDU dataset between our methods and available baselines are given in Table 14. Our methods outperform baseline methods in terms of average F1 score, by 1–4%, and OA, by 1–3%.

Table 14. Comparison of F1 scores and OA on the LASDU dataset with respect to current state-of-the-art methods.

	Ground	Building	Tree	LV	Artifact	Avg. F1	OA
Ours (2DCNN)	0.887	0.935	0.860	0.691	0.360	0.746	0.846
Ours (3DCNN)	0.885	0.915	0.858	0.673	0.322	0.730	0.837
PointNet++ [32]	0.877	0.906	0.820	0.632	0.313	0.710	0.828
HDA-PointNet++ [36]	0.887	0.932	0.822	0.652	0.369	0.733	0.844

4.4. Generalization Capability

To prove the generalization of our approach, we experimented with training and predicting on different datasets. In particular, we used the trained networks to run predictions on unseen datasets with overlapping sensor features (i.e., intensity, number of returns, return numbers, color information). To make use of the existing ground truth data for accuracy assessment, we applied class modifications in the prediction results and ground truth labels to match these two. Table 15 shows the prediction results on the ISPRS Vaihingen dataset of models trained on the DALES dataset. As the class structures of these two datasets do not completely match, we had to arrange them as shown in Table 16.

Table 15. Averaged and weighted F1 and IoU scores for 2DCNN and 3DCNN classifiers trained on DALES dataset and predicting on ISPRS Vaihingen dataset. The OA is also reported.

	Cable	Ground	Car	Fence	Building	Vegetation	Average	Weighted	OA
2DCNN DALES F1	0.130	0.926	0.399	0.001	0.759	0.397	0.435	0.753	0.779
3DCNN DALES F1	0.230	0.929	0.347	0.000	0.758	0.318	0.430	0.739	0.774
2DCNN DALES IoU	0.069	0.862	0.249	0.000	0.611	0.247	0.340	0.649	0.779
3DCNN DALES IoU	0.130	0.867	0.210	0.000	0.610	0.189	0.334	0.639	0.774

Table 16. Corresponding classes between ISPRS Vaihingen and DALES datasets along with their distributions.

ISPRS Vaihingen	Points (Test)	DALES	Points (Train)
Powerline	0.2%	Cable	29.7%
Low veget./Imp. Surface	38.9%	Ground	50.0%
Car	0.8%	Car/Truck	1.2%
Fence	1.9%	Fence	0.6%
Roof/Facade	22.7%	Building/Pole	0.9%
Shrub/Tree	35.5%	Vegetation	17.6%

The DALES dataset was also used to learn classes and then predict on the Bordeaux dataset. Table 17 shows classification results, along with class modifications in Table 18.

Table 17. Average and weighted F1 and IoU scores for the 2DCNN and 3DCNN classifiers trained on DALES dataset and predicted on Bordeaux dataset. The OA is also reported.

	Ground	Building	Vegetation	Others	Average	Weighted	OA
2DCNN DALES F1	0.965	0.978	0.947	0.434	0.831	0.968	0.969
3DCNN DALES F1	0.956	0.972	0.937	0.462	0.832	0.961	0.961
2DCNN DALES IoU	0.931	0.958	0.900	0.277	0.766	0.941	0.969
3DCNN DALES IoU	0.916	0.945	0.882	0.300	0.761	0.927	0.961

Table 18. Corresponding classes between Bordeaux and DALES datasets.

Bordeaux	Points (Test)	DALES	Points (Train)
Ground	13.9%	Ground	29.7%
Roof/Facade	80.2%	Building/Cable/Pole	50.0%
Vegetation	5.0%	Vegetation	2.1%
Others	1.0%	Car/Truck/Fence	18.2%

Similarly, the ISPRS Vaihingen was used to train the classifiers and then predict classes on the Bordeaux dataset with network trained on ISPRS Vaihingen. The accuracy metrics for these predictions are given in Table 19 and class modifications are given in Table 20.

Table 19. F1, IoU with 2DCNN and 3DCNN classifiers trained on ISPRS Vaihingen dataset and predicted on Bordeaux dataset. The OA is also reported.

	Ground	Facade	Roof	Vegetation	Others	Average	Weighted	OA
2DCNN ISPRS F1	0.921	0.690	0.923	0.759	0.400	0.739	0.878	0.882
3DCNN ISPRS F1	0.882	0.743	0.878	0.832	0.331	0.733	0.855	0.855
2DCNN ISPRS IoU	0.854	0.527	0.857	0.612	0.250	0.620	0.793	0.882
3DCNN ISPRS IoU	0.789	0.591	0.782	0.712	0.198	0.614	0.751	0.855

Table 20. Corresponding classes between Bordeaux and ISPRS Vaihingen datasets.

Bordeaux	Points (Test)	ISPRS Vaihingen	Points (Train)
Ground	13.9%	Ground	18.1%
Roof	43.6%	Roof	26.9%
Facade	36.6%	facade	5.1%
Vegetation	5.0%	Low	46.1%
Others	1.0%	Veget./Shrub/Tree	3.7%
		Cables/Car/Fence	

5. Discussions

The reported results (Section 4) show that alternatively 2DCNN and 3DCNN achieve the best accuracy in the various datasets. As there are no significant differences (Tables 21 and 22) and considering also the computational time and resources (Table 12), we recommend a 2DCNN approach with respect to a 3DCNN when the generalization capability is not needed. Results reported in Section 4.4 shows 3DCNN has better generalization capabilities especially for facade and cables classes.

Table 21. Summarized OA achieved in the different datasets considered in the evaluation procedures.

	ISPRS Vaihingen	DALES	LASDU	Bordeaux	3DOMCity
2DCNN	0.826	0.938	0.846	0.944	0.889
3DCNN	0.806	0.934	0.837	0.938	0.883

Table 22. Summarized OA for the generalization tests.

Trained on	Predicted on	Model	OA
DALES	ISPRS Vaihingen	2DCNN	0.779
DALES	ISPRS Vaihingen	3DCNN	0.774
DALES	Bordeaux	2DCNN	0.969
DALES	Bordeaux	3DCNN	0.961
ISPRS Vaihingen	Bordeaux	2DCNN	0.882
ISPRS Vaihingen	Bordeaux	3DCNN	0.855

The results show that the developed framework also outperform state-of-the-art methods (Tables 13 and 14), with differences within few percent, but still very low memory and energy usage. If AI methods were to be used for real-world deployment or in production activities, manual quality control and corrections would still be required. In this case, few percentage of overall accuracy is not expected to make significant difference. However, in terms of efficiency, the reported numbers for training time and used computer resources (Table 12) clearly shows that our method can process the same dataset much faster compared to current state-of-the-art methods. We explain the time differences between our DL methods and the current state-of-the-art DL methods with the following two steps implemented: (i) data reduction with downsampling, (ii) multi-scale feature extraction before passing data to the DNN. Extracting some features outside the DNN allows us to utilize a shallower network compared to the state-of-the-art methods, which supports our efficiency goal. A shallower DNN is not only faster to train and predict, but also consumes less memory allowing us to run 1024 points per batch in 11 GB of GPU memory. We consider such efficiency to bring advantages of less energy consumption, lowering hardware costs, and speeding-up the production progress.

The reported tests on generalization ability (summarized in Table 22) show promising results. A key challenge here is to get the features from different datasets to share similar characteristics. Density variations among datasets make it more difficult to generalize without downsampling them. This step ensures the density characteristics among datasets

are more similar compared to their original versions (Tables 3–5). For instance, 3DCNN method achieves 82.6% OA (Table 7) on ISPRS Vaihingen dataset, yet, training the same model on DALES dataset achieves 77.9% OA (Table 15). Similarly, on Bordeaux dataset, 2DCNN classifier achieves 94.4% OA (Table 10) and training the same model on DALES dataset achieves 96.9% OA (Table 17). Furthermore, the models trained on ISPRS Vaihingen dataset achieved higher accuracies on Bordeaux even if the color information for these datasets are in different spaces (NIR, green, blue for ISPRS Vaihingen and red, green, blue for Bordeaux). Training on ISPRS Vaihingen, 2DCNN classifier achieved 88.2% OA, while 3DCNN achieved 85.5% OA on the Bordeaux dataset.

Comparing the results summarized in Tables 21 and 22, it can be seen a dataset has a significant effect on the accuracy. For instance, on ISPRS Vaihingen, our methods reach an OA of 81–83%, and, training the models on DALES, where our methods reach OA of 93–94%, they still reach OA of 77–78%. Likewise, when the same models—trained on DALES—predicting on Bordeaux they reach OA of 97%, in which they reach 94% OA. Training the models ISPRS Vaihingen and predicting on Bordeaux achieves 88% OA. Similarly, PointNet++ achieves 96% OA on DALES (Table 13) and 83% on LASDU (Table 14). Based on these numbers, it can be inferred that there is a high correlation between the used datasets for prediction and the achieved OA. We consider this can be caused by noise in the ground truth labels (i.e., wrong labelled points), noise in the point cloud (i.e., isolated point type noise) and type of classes in the data (more complex classes vs. simpler classes).

6. Conclusions

In this paper, we propose the TONIC framework for geospatial point cloud classification. It uses multi-scale handcrafted features together with DNN, not only reaching accuracy on par with the current state-of-the-art models, but also a more efficient approach in terms of resources and computational time (Table 12). We share our experiments on multiple datasets, including generalization experiments. The reported results also indicate that our framework is able to work with different data sources (namely photogrammetry and LiDAR). Considering the large amount of data available in the real world—as opposed to benchmarks—and the computational complexity of the state-of-the-art methods, it will not be surprising to see more research focusing on this aspect in the near future. Based on the accuracy metrics reported in Section 4.2, our framework can be useful for building extraction, powerline mapping, digital terrain model generation from point clouds, etc. The two proposed DNN approaches are observed to perform similarly within the same dataset, yet 3DCNN is observed to perform better specifically in cables and poles classes as seen in Tables 7 and 8. This makes our solution reliable and useful for many classification tasks with geospatial data.

The main limitation of our approach is its applicability to urban scenarios surveyed with aerial sensors (photogrammetry or LiDAR). Due to the used hand-crafted features, our approach might not be competitive in alternative cases other than aerial point cloud classification. As future work, we will investigate its deployment to indoor, UAV, and satellite-based dense point clouds and we will consider data augmentation, which we expect to improve our results.

Author Contributions: The article presents a research contribution that involved the authors in equal measure. E.Ö. dealt with the conceptualization, state of the art, methodology, data processing and results. F.R. and A.G. supervised the overall work and reviewed the paper, writing the introduction, aim, and conclusions. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The Vaihingen data set was provided by the German Society for Photogrammetry, Remote Sensing and Geoinformation (DGPF): <http://www.ifp.uni-stuttgart.de/dgpf/DKEP-Allg.html> (accessed on 25 March 2021). The authors would like to acknowledge Hexagon/Leica Geosystem for providing the Bordeaux dataset. The data set presented in this paper contains information licensed under the Open Government License—City of Surrey. We would like to thank the City of Surrey for generously providing the raw data presented in this paper. For more information about the raw data and other data sources like it please see their Open Data Site.

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

References

1. Liu, W.; Sun, J.; Li, W.; Hu, T.; Wang, P. Deep Learning on Point Clouds and Its Application: A Survey. *Sensors* **2019**, *19*, 4188. [[CrossRef](#)] [[PubMed](#)]
2. Bello, S.A.; Yu, S.; Wang, C.; Adam, J.M.; Li, J. Review: Deep Learning on 3D Point Clouds. *Remote Sens.* **2020**, *12*, 1729. [[CrossRef](#)]
3. Kanezaki, A.; Matsushita, Y.; Nishida, Y. RotationNet: Joint Object Categorization and Pose Estimation Using Multiviews from Unsupervised Viewpoints. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 5010–5019.
4. Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.E.; Bronstein, M.M.; Solomon, J.M. Dynamic Graph CNN for Learning on Point Clouds. *ACM Trans. Graph.* **2019**, *38*. [[CrossRef](#)]
5. Su, H.; Maji, S.; Kalogerakis, E.; Learned-Miller, E. Multi-View Convolutional Neural Networks for 3D Shape Recognition. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 945–953.
6. Huang, X.; Cao, R.; Cao, Y. A Density-Based Clustering Method for the Segmentation of Individual Buildings from Filtered Airborne LiDAR Point Clouds. *J. Indian Soc. Remote Sens.* **2019**, *47*, 907–921. [[CrossRef](#)]
7. Maltezos, E.; Doulamis, A.; Doulamis, N.; Ioannidis, C. Building Extraction from LiDAR Data Applying Deep Convolutional Neural Networks. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 155–159. [[CrossRef](#)]
8. Becker, C.; Rosinskaya, E.; Häni, N.; D’Angelo, E.; Strecha, C. Classification of Aerial Photogrammetric 3D Point Clouds. *Photogramm. Eng. Remote Sens.* **2018**, *84*, 287–295. [[CrossRef](#)]
9. Bittner, K.; D’Angelo, P.; Körner, M.; Reinartz, P. DSM-to-LoD2: Spaceborne Stereo Digital Surface Model Refinement. *Remote Sens.* **2018**, *10*, 1926. [[CrossRef](#)]
10. Özdemir, E.; Remondino, F. Segmentation of 3D Photogrammetric Point Cloud for 3D Building Modeling. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2018**, *XLII-4/W10*, 135–142. [[CrossRef](#)]
11. Qin, R.; Tian, J.; Reinartz, P. 3D Change Detection—Approaches and Applications. *ISPRS J. Photogramm. Remote Sens.* **2016**, *122*, 41–56. [[CrossRef](#)]
12. Urech, P.R.W.; Dissegna, M.A.; Girot, C.; Grêt-Regamey, A. Point Cloud Modeling as a Bridge Between Landscape Design and Planning. *Landsc. Urban Plan.* **2020**, *203*, 103903. [[CrossRef](#)]
13. Henderson, R.M.; Clark, K.B. Architectural Innovation: The Reconfiguration of Existing Product Technologies and the Failure of Established Firms. *Adm. Sci. Q.* **1990**, *35*, 9–30. [[CrossRef](#)]
14. Zhang, J.; Zhao, X.; Chen, Z.; Lu, Z. A Review of Deep Learning-Based Semantic Segmentation for Point Cloud. *IEEE Access* **2019**, *7*, 179118–179133. [[CrossRef](#)]
15. Xie, Y.; Tian, J.; Zhu, X.X. Linking Points with Labels in 3D: A Review of Point Cloud Semantic Segmentation. *IEEE Geosci. Remote Sens. Mag.* **2020**, *8*, 38–59. [[CrossRef](#)]
16. Griffiths, D.; Boehm, J. A Review on Deep Learning Techniques for 3D Sensed Data Classification. *Remote Sens.* **2019**, *11*, 1499. [[CrossRef](#)]
17. 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**, *1*. [[CrossRef](#)]
18. Ramanath, A.; Muthusrinivasan, S.; Xie, Y.; Shekhar, S.; Ramachandra, B. NDVI Versus CNN Features in Deep Learning for Land Cover Classification of Aerial Images. In Proceedings of the IGARSS 2019—2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; pp. 6483–6486.
19. Matrone, F.; Grilli, E.; Martini, M.; Paolanti, M.; Pierdicca, R.; Remondino, F. Comparing Machine and Deep Learning Methods for Large 3D Heritage Semantic Segmentation. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 535. [[CrossRef](#)]
20. LeCun, Y.; Bengio, Y.; Hinton, G. Deep Learning. *Nature* **2015**, *521*, 436–444. [[CrossRef](#)] [[PubMed](#)]
21. Li, W.; Wang, F.-D.; Xia, G.-S. A Geometry-Attentional Network for ALS Point Cloud Classification. *ISPRS J. Photogramm. Remote Sens.* **2020**, *164*, 26–40. [[CrossRef](#)]
22. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016; ISBN 0-262-33737-1.
23. Heipke, C.; Rottensteiner, F. Deep Learning for Geometric and Semantic Tasks in Photogrammetry and Remote Sensing. *Geo-Spat. Inf. Sci.* **2020**, *23*, 10–19. [[CrossRef](#)]
24. Weinmann, M.; Jutzi, B.; Mallet, C. Feature Relevance Assessment for The Semantic Interpretation of 3D Point Cloud Data. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2013**, *5*, 1. [[CrossRef](#)]

25. Hackel, T.; Wegner, J.D.; Schindler, K. Fast Semantic Segmentation of 3d Point Clouds with Strongly Varying Density. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2016**, *3*, 177–184. [[CrossRef](#)]
26. Thomas, H.; Goulette, F.; Deschaud, J.; Marcotegui, B.; LeGall, Y. Semantic Classification of 3D Point Clouds with Multiscale Spherical Neighborhoods. In Proceedings of the 2018 International Conference on 3D Vision (3DV), Verona, Italy, 5–8 September 2018; pp. 390–398.
27. Zhang, J.; Lin, X.; Ning, X. SVM-Based Classification of Segmented Airborne LiDAR Point Clouds in Urban Areas. *Remote Sens.* **2013**, *5*, 3749. [[CrossRef](#)]
28. Li, N.; Liu, C.; Pfeifer, N. Improving LiDAR Classification Accuracy by Contextual Label Smoothing in Post-Processing. *ISPRS J. Photogramm. Remote Sens.* **2019**, *148*, 13–31. [[CrossRef](#)]
29. Wang, X.; Zhao, Y.; Pourpanah, F. Recent Advances in Deep Learning. *Int. J. Mach. Learn. Cybern.* **2020**, *11*, 747–750. [[CrossRef](#)]
30. Yan, Y.; Yan, H.; Guo, J.; Dai, H. Classification and Segmentation of Mining Area Objects in Large-Scale Spares Lidar Point Cloud Using a Novel Rotated Density Network. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 182. [[CrossRef](#)]
31. Hackel, T.; Savinov, N.; Ladicky, L.; Wegner, J.D.; Schindler, K.; Pollefeys, M. SEMANTIC3D.NET: A New Large-Scale Point Cloud Classification Benchmark. *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.* **2017**, *IV-1/W1*, 91–98. [[CrossRef](#)]
32. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep Hierarchical Feature Learning on Point Sets in A Metric Space. *arXiv* **2017**, arXiv:1706.02413.
33. Yousefhusien, M.; Kelbe, D.J.; Ientilucci, E.J.; Salvaggio, C. A Multi-Scale Fully Convolutional Network for Semantic Labeling of 3D Point Clouds. *ISPRS J. Photogramm. Remote Sens.* **2018**, *143*, 191–204. [[CrossRef](#)]
34. Özdemir, E.; Remondino, F.; Golkar, A. Aerial Point Cloud Classification with Deep Learning and Machine Learning Algorithms. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *42*, 843–849. [[CrossRef](#)]
35. Wen, C.; Li, X.; Yao, X.; Peng, L.; Chi, T. Airborne LiDAR Point Cloud Classification with Global-Local Graph Attention Convolution Neural Network. *ISPRS J. Photogramm. Remote Sens.* **2021**, *173*, 181–194. [[CrossRef](#)]
36. Huang, R.; Xu, Y.; Hong, D.; Yao, W.; Ghamisi, P.; Stilla, U. Deep Point Embedding for Urban Classification Using ALS Point Clouds: A New Perspective from Local to Global. *ISPRS J. Photogramm. Remote Sens.* **2020**, *163*, 62–81. [[CrossRef](#)]
37. Li, X.; Wang, L.; Wang, M.; Wen, C.; Fang, Y. DANCE-NET: Density-Aware Convolution Networks with Context Encoding for Airborne LiDAR Point Cloud Classification. *ISPRS J. Photogramm. Remote Sens.* **2020**, *166*, 128–139. [[CrossRef](#)]
38. Winiwarter, L.; Mandlbürger, G.; Schmohl, S.; Pfeifer, N. Classification of ALS Point Clouds Using End-to-End Deep Learning. *PFG—J. Photogramm. Remote Sens. Geoinf. Sci.* **2019**, *87*, 75–90. [[CrossRef](#)]
39. Chen, Y.; Liu, G.; Xu, Y.; Pan, P.; Xing, Y. PointNet++ Network Architecture with Individual Point Level and Global Features on Centroid for ALS Point Cloud Classification. *Remote Sens.* **2021**, *13*, 472. [[CrossRef](#)]
40. 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 International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 6411–6420.
41. Özdemir, E.; Remondino, F. Classification of Aerial Point Clouds with Deep Learning. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *XLII-2/W13*, 103–110. [[CrossRef](#)]
42. Rusu, R.B.; Cousins, S. 3D Is Here: Point Cloud Library (PCL). In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 1–4.
43. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. *arXiv* **2015**, arXiv:1603.04467.
44. Chollet, F. *Others Keras*; GitHub: San Francisco, CA, USA, 2015.
45. Harris, C.R.; Millman, K.J.; van der Walt, S.J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N.J.; et al. Array Programming with NumPy. *Nature* **2020**, *585*, 357–362. [[CrossRef](#)] [[PubMed](#)]
46. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
47. Pauly, M.; Keiser, R.; Gross, M. Multi-Scale Feature Extraction on Point-Sampled Surfaces. *Comput. Graph. Forum* **2003**, *22*, 281–289. [[CrossRef](#)]
48. Weinmann, M.; Jutzi, B.; Hinz, S.; Mallet, C. Semantic Point Cloud Interpretation Based on Optimal Neighborhoods, Relevant Features and Efficient Classifiers. *ISPRS J. Photogramm. Remote Sens.* **2015**, *105*, 286–304. [[CrossRef](#)]
49. Grilli, E.; Farella, E.M.; Torresani, A.; Remondino, F. Geometric Features Analysis for The Classification of Cultural Heritage Point Clouds. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *42*, 541–548. [[CrossRef](#)]
50. Koundinya, S.; Sharma, H.; Sharma, M.; Upadhyay, A.; Manekar, R.; Mukhopadhyay, R.; Karmakar, A.; Chaudhury, S. 2D-3D CNN Based Architectures for Spectral Reconstruction from RGB Images. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Salt Lake City, UT, USA, 18–22 June 2018; pp. 957–957.
51. Niemeyer, J.; Rottensteiner, F.; Soergel, U. Contextual Classification of Lidar Data and Building Object Detection in Urban Areas. *ISPRS J. Photogramm. Remote Sens.* **2014**, *87*, 152–165. [[CrossRef](#)]
52. Varney, N.; Asari, V.K.; Graehling, Q. DALES: A Large-Scale Aerial LiDAR Data Set for Semantic Segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Seattle, WA, USA, 14–19 June 2020.
53. Ye, Z.; Xu, Y.; Huang, R.; Tong, X.; Li, X.; Liu, X.; Luan, K.; Hoegner, L.; Stilla, U. LASDU: A Large-Scale Aerial LiDAR Dataset for Semantic Labeling in Dense Urban Areas. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 450. [[CrossRef](#)]

54. Li, X.; Cheng, G.; Liu, S.; Xiao, Q.; Ma, M.; Jin, R.; Che, T.; Liu, Q.; Wang, W.; Qi, Y.; et al. Heihe Watershed Allied Telemetry Experimental Research (HiWATER). *Bull. Am. Meteorol. Soc.* **2013**, *94*, 1145–1160. [[CrossRef](#)]
55. Toschi, I.; Farella, E.M.; Welponer, M.; Remondino, F. Quality-Based Registration Refinement of Airborne Lidar and Photogrammetric Point Clouds. *ISPRS J. Photogramm. Remote Sens.* **2021**, *172*, 160–170. [[CrossRef](#)]
56. Özdemir, E.; Toschi, I.; Remondino, F. A Multi-Purpose Benchmark for Photogrammetric Urban 3D Reconstruction in a Controlled Environment. *ISPRS—Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2019**, *42*, 53–60. [[CrossRef](#)]
57. Tan, M.; Le, Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *arXiv* **2019**, arXiv:1905.11946 2020.