# SEMANTIC3D.NET: A NEW LARGE-SCALE POINT CLOUD CLASSIFICATION BENCHMARK

Timo Hackel[a], Nikolay Savinov[b], Lubor Ladicky[b], Jan D. Wegner[a], Konrad Schindler[a], Marc Pollefeys[b]

[a] IGP, ETH Zurich, Switzerland - (timo.hackel, jan.wegner, konrad.schindler)@geod.baug.ethz.ch
[b] CVG, ETH Zurich, Switzerland - (nikolay.savinov, lubor.ladicky, marc.pollefeys)@inf.ethz.ch

**Commission II, WG II/6**

## ABSTRACT:

This paper presents a new 3D point cloud classification benchmark data set with over four billion manually labelled points, meant as input for data-hungry (deep) learning methods. We also discuss first submissions to the benchmark that use deep convolutional neural networks (CNNs) as a work horse, which already show remarkable performance improvements over state-of-the-art. CNNs have become the de-facto standard for many tasks in computer vision and machine learning like semantic segmentation or object detection in images, but have no yet led to a true breakthrough for 3D point cloud labelling tasks due to lack of training data. With the massive data set presented in this paper, we aim at closing this data gap to help unleash the full potential of deep learning methods for 3D labelling tasks. Our *semantic3D.net* data set consists of dense point clouds acquired with static terrestrial laser scanners. It contains 8 semantic classes and covers a wide range of urban outdoor scenes: churches, streets, railroad tracks, squares, villages, soccer fields and castles. We describe our labelling interface and show that our data set provides more dense and complete point clouds with much higher overall number of labelled points compared to those already available to the research community. We further provide baseline method descriptions and comparison between methods submitted to our online system. We hope *semantic3D.net* will pave the way for deep learning methods in 3D point cloud labelling to learn richer, more general 3D representations, and first submissions after only a few months indicate that this might indeed be the case.

## 1. INTRODUCTION

Deep learning has made a spectacular comeback since the seminal paper of (Krizhevsky et al., 2012), which revives earlier work of (Fukushima, 1980, LeCun et al., 1989). Especially deep convolutional neural networks (CNN) have quickly become the core technique for a whole range of learning-based image analysis tasks. The large majority of state-of-the-art methods in computer vision and machine learning include CNNs as one of their essential components. Their success for image-interpretation tasks is mainly due to (i) easily parallelisable network architectures that facilitate training from millions of images on a single GPU and (ii) the availability of huge public benchmark data sets like *ImageNet* (Deng et al., 2009, Russakovsky et al., 2015) and *Pascal VOC* (Everingham et al., 2010) for rgb images, or *SUN rgb-d* (Song et al., 2015) for rgb-d data.

While CNNs have been a great success story for image interpretation, it has been less so for 3D point cloud interpretation. What makes supervised learning hard for 3D point clouds is the sheer size of millions of points per data set, and the irregular, not grid-aligned, and in places very sparse structure, with strongly varying point density (Figure 1).

While recording is nowadays straight-forward, the main bottleneck is to generate enough manually labeled training data, needed for contemporary (deep) machine learning to learn good models, which generalize well across new, unseen scenes. Due to the additional dimension, the number of classifier parameters is larger in 3D space than in 2D, and specific 3D effects like occlusion or variations in point density lead to many different patterns for identical output classes. This aggravates training good, general classifiers and we generally need more training data in 3D than in 2D[1]. In contrast to images, which are fairly easy to annotate even for untrained users, 3D point clouds are harder to interpret. Navigation in 3D is more time-consuming and the strongly varying point density aggravates scene interpretation.

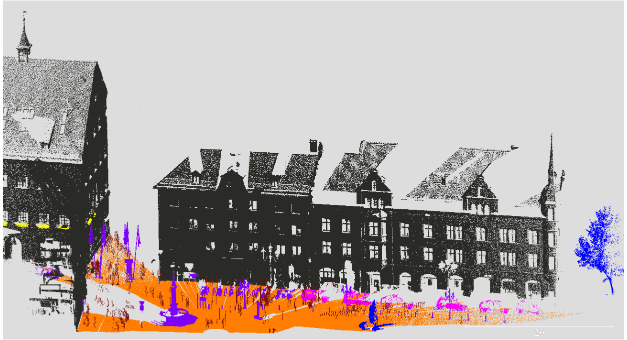


Figure 1: Example point cloud of the benchmark dataset, where colours indicate class labels.

In order to accelerate the development of powerful algorithms for point cloud processing[2], we provide the (to our knowledge) hitherto largest collection of (terrestrial) laser scans with point-level semantic ground truth annotation. In total, it consists of over $4 \cdot 10^9$ points and class labels for 8 classes. The data set is split into training and test sets of approximately equal size. The scans are challenging, not only due to their size of up to $\approx 4 \cdot 10^8$ points per scan, but also because of their high measurement resolution and long measurement range, leading to extreme density

[1] Note that the large number of 3D points of *semantic3d.net* ($4 \times 10^9$ points) is at the same scale as the number of pixels of the *SUN rgb-d* benchmark ($\approx 3.3 \times 10^9$ px) (Song et al., 2015), which aims at 3D object classification. However, the number of 3D points per laser scan ($\approx 4 \times 10^8$ points) is much larger than the number of pixels per image ($\approx 4 \times 10^5$ px).

[2] Note that, besides laser scanner point clouds, it is also more efficient to classify point clouds generated from SfM pipelines directly instead of going through all individual images to then merge results (Riemenschneider et al., 2014).

changes and large occlusions. For convenient use of the benchmark test we provide not only freely available data, but also an automated online submission system as well as public results of the submitted methods. The benchmark also includes baselines, one following the standard paradigm of eigenvalue-based feature extraction at multiple scales followed by classification with a random forest, the other a basic deep learning approach. Moreover, first submissions have been made to the benchmark, which we also briefly discuss.

## 2. RELATED WORK

Benchmarking efforts have a long tradition in the geospatial data community and particularly in ISPRS. Recent efforts include, for example, the *ISPRS-EuroSDR benchmark on High Density Aerial Image Matching*[3] that aims at evaluating dense matching methods for oblique aerial images (Haala, 2013, Cavegn et al., 2014) and the *ISPRS Benchmark Test on Urban Object Detection and Reconstruction*, which contains several different challenges like semantic segmentation of aerial images and 3D object reconstruction (Rottensteiner et al., 2013).

In computer vision, very large-scale benchmark data sets with millions of images have become standard for learning-based image interpretation tasks. A variety of datasets have been introduced, many tailored for a specific task, some serving as basis for annual challenges for several consecutive years (e.g., *ImageNet, Pascal VOC*). Datasets that aim at boosting research in image classification and object detection heavily rely on images downloaded from the internet. Web-based imagery has been a major driver of benchmarks because no expensive, dedicated photography campaigns have to be accomplished for dataset generation. This enables scaling benchmarks from hundreds to millions of images, although often weakly annotated and with a considerable amount of label noise that has to be taken into account. Additionally, one can assume that internet images constitute a very general collection of images with less bias towards particular sensors, scenes, countries, objects etc., which allows training richer models that generalize well.

One of the first successful attempts to object detection in images at very large scale is *tinyimages* with over 80 million small ($32 \times 32$ $px$) images (Torralba et al., 2008). A milestone and still widely used dataset for semantic image segmentation is the famous Pascal VOC (Everingham et al., 2010) dataset and challenge, which has been used for training and testing many of the well-known, state-of-the-art algorithms today like (Long et al., 2015, Badrinarayanan et al., 2015). Another, more recent dataset is *MSCOCO*[4], which contains 300,000 images with annotations that allow for object segmentation, object recognition in context, and image captioning. One of the most popular benchmarks in computer vision today is the *ImageNet* dataset (Deng et al., 2009, Russakovsky et al., 2015), which made Convolutional Neural Networks popular in computer vision (Krizhevsky et al., 2012). It contains $> 14 \times 10^6$ images organized according to the WordNet hierarchy[5], where words are grouped into sets of cognitive synonyms.

The introduction of the popular, low-cost gaming device Microsoft Kinect gave rise to several, large rgb-d image databases. Popular examples are the *NYU Depth Dataset V2* (Silberman et al., 2012) or *SUN RGB-D* (Song et al., 2015) that provide labeled rgb-d images for object segmentation and scene understanding. Compared to laser scanners, low-cost, structured-light rgb-d sensors have much shorter measurement ranges, lower resolutions, and work poorly outdoors due to interference of the infrared spectrum of the sunlight with the projected sensor pattern.

To the best of our knowledge, no publicly available dataset with laser scans at the scale of the aforementioned vision benchmarks exists today. Thus, many recent Convolutional Neural Networks that are designed for Voxel Grids (Brock et al., 2017, Wu et al., 2015) resort to artificially generated data from the CAD models of ModelNet (Wu et al., 2015), a rather small, synthetic dataset. As a consequence, recent ensemble methods (e.g., (Brock et al., 2017)) reach performance of over 97% on ModelNet10, which clearly indicates a model overfit due to limited data.

Those few existing laser scan datasets are mostly acquired with mobile mapping devices or robots like *DUT1* (Zhuang et al., 2014), *DUT2* (Zhuang et al., 2015), or *KAIST* (Choe et al., 2013), which are small ($< 10^7$ points) and not publicly available. Publicly availabe laser scan datasets include the *Oakland dataset* (Munoz et al., 2009) ($< 2 \times 10^6$ points), the *Sydney Urban Objects data set* (De Deuge et al., 2013), the *Paris-rue-Madame database* (Serna et al., 2014) and data from the *IQmulus & TerraMobilita Contest* (Vallet et al., 2015). All have in common that they use 3D LIDAR data from a mobile mapping car which provides a much lower point density than a typical static scan, like ours. They are also relatively small, such that supervised learning algorithms easily overfit. The majority of today's available point cloud datasets comes without a thorough, transparent evaluation that is publicly available on the internet, continuously updated and that lists all submissions to the benchmark.

With the *semantic3D.net* benchmark presented in this paper, we aim at closing this gap. It provides the largest labeled 3D point cloud data set with approximately four billion hand-labeled points, comes with a sound evaluation, and continuously updates submissions. It is the first dataset that allows full-fledged deep learning on real 3D laser scans that have high-quality, manually assigned labels per point.

## 3. OBJECTIVE

Given a set of points (here: dense scans from a static, terrestrial laser scanner), we want to infer one individual class label per point. We provide three baseline methods that are meant to represent typical categories of approaches recently used for the task.

*i) 2D image baseline:*

Many state-of-the-art laser scanners also acquire color values or even entire color images for the scanned scene. Color images can add additional object evidence that may help classification. Our first, naive baseline classifies only the 2D color images without using any depth information, so as to establish a link to the vast literature on 2D semantic image segmentation. Modern methods use Deep Convolutional Neural Networks as a workhorse. Encoder-decoder architectures, like SegNet (Badrinarayanan et al., 2015), are able to infer the labels of an entire image at once. Deep architectures can also be combined with Conditional Random Fields (CRF) (Chen et al., 2016). Our baseline method in Section 3.1 covers image-based semantic segmentation.

*ii) 3D Covariance baseline:* A more specific approach, which takes advantage of the 3D information, is to work on point clouds

[3] http://www.ifp.uni-stuttgart.de/ISPRS-EuroSDR/ImageMatching/index.en.html

[4] http://mscoco.org/

[5] https://wordnet.princeton.edu/

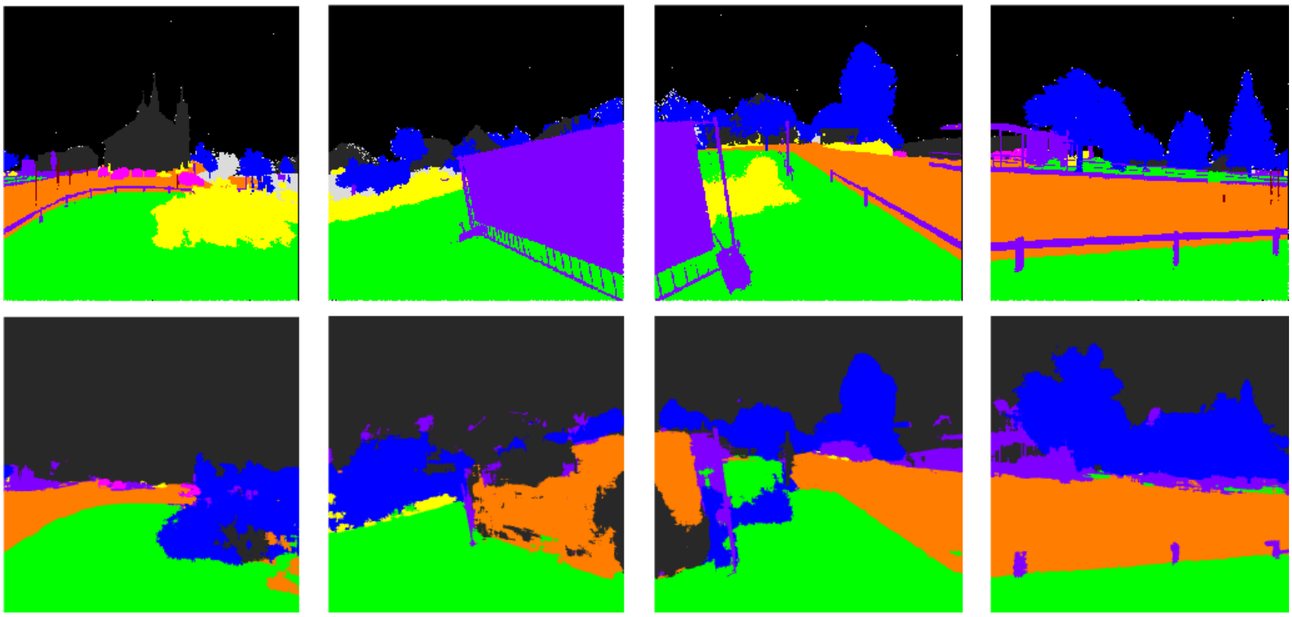


Figure 2: *Top row:* projection of ground truth to images. *Bottom row:* results of classification with the image baseline. *White:* unlabeled pixels, *black:* pixels with no corresponding 3D point, *gray:* buildings, *orange:* man made ground, *green:* natural ground, *yellow:* low vegetation, *blue:* high vegetation, *purple:* hard scape, *pink:* cars

directly. We use a recent implementation of the standard classification pipeline, i.e., extract hand-crafted features from 3D (multiscale) neighbourhoods, and feed them to a discriminative learning algorithm. Typical features encode surface properties based on the covariance tensor of a point's neighborhood (Demantké et al., 2011) or a randomized set of histograms (Blomley et al., 2014). Additionally, height distributions can be encoded by using cylindrical neighborhoods (Monnier et al., 2012, Weinmann et al., 2013). The second baseline method (Section 3.2) represents this category.

*iii) 3D CNN baseline:* It is a rather obvious extension to apply deep learning also to 3D point clouds, mostly using voxel grids to obtain a regular neighbourhood structure. To work efficiently with large point neighborhoods in clouds with strongly varying density, recent work uses adaptive neighbourhood data structures like octrees (Wu et al., 2015, Brock et al., 2017, Riegler et al., 2017) or sparse voxel grids (Engelcke et al., 2017). Our third baseline method in Section 3.3 is a basic, straight-forward implementation of 3D voxel-grid CNNs.

### 3.1 2D Image Baseline

We convert color values of the scans to separate images (without depth) with cube mapping (Greene, 1986). Ground truth labels are also projected from the point clouds to image space, such that the 3D point labeling task turns into a pure semantic image segmentation problem of 2D RGB images (Figure 2). We chose the associate hierarchical fields method (Ladicky et al., 2013) for semantic segmentation because it has proven to deliver good performance for a variety of tasks (e.g., (Montoya et al., 2014, Ladický et al., 2014)) and was available in its original implementation.

The method works as follows: four different types of features – texton (Malik et al., 2001), SIFT (Lowe, 2004), local quantized ternary patters (Hussain and Triggs, 2012) and self-similarity features (Shechtman and Irani, 2007) – are extracted densely per image pixel. Each feature category is separately clustered into 512 distinct patterns using standard K-means clustering, which corresponds to a typical bag-of-words representation. For each pixel in an image, the feature vector is a concatenation of bag-of-word histograms over a fixed set of 200 rectangles of varying sizes. These rectangles are randomly placed in an extended neighbourhood around a pixel. We use multi-class boosting (Torralba et al., 2004) as classifier and the most discriminative weak features are found as explained in (Shotton et al., 2006). To add local smoothing without loosing sharp object boundaries, we smooth inside superpixels and favor class transitions at their boundaries. Superpixels are extracted via mean-shift (Comaniciu and Meer, 2002) with 3 sets of coarse-to-fine parameters as described in (Ladicky et al., 2013). Class likelihoods of overlapping superpixels are predicted using the feature vector consisting of a bag-of-words representation of each superpixel. Pixel-based and superpixel-based classifiers with additional smoothness priors over pixels and superpixels are combined in a probabilistic fashion in a conditional random field framework as proposed in (Kohli et al., 2008). The most probable solution of the associative hierarchical optimization problem is found using the move making (Boykov et al., 2001) graph-cut based algorithm (Boykov and Kolmogorov, 2004), with appropriate graph construction for higher-order potentials (Ladicky et al., 2013).

### 3.2 3D Covariance Baseline

The second baseline was inspired by (Weinmann et al., 2015). It infers the class label directly from the 3D point cloud using multiscale features and discriminative learning. Again, we had access to the original implementation. That method uses an efficient approximation of multi-scale neighbourhoods, where the point cloud is sub-sampled into a multi-resolution pyramid, such that a constant, small number of neighbours per level captures the multi-scale information. The multi-scale pyramid is generated by voxel-grid filtering with uniform spacing.

The feautre set extracted at each level is an extension of the one decribed in (Weinmann et al., 2013). It uses different combinations of eigenvalues and eigenvectors of the covariance per point-neighborhood to different geometric surface properties. Furthermore, height features based on vertical, cylindrical neighbourhoods are added to emphasize the special role of the gravity direction (assuming that scans are, as usual, aligned to the vertical).

Note that we do not make use of color values or scanner intensities. These are not always available in point clouds, and we empirically found that they do not improve the results of the method. As classifier, we use a random forest, where optimal parameters are found with grid search and five fold cross-validation. Please refer to (Hackel et al., 2016) for details.

### 3.3 3D CNN Baseline

We design our baseline for the point cloud classification task following recent VoxNet (Maturana and Scherer, 2015) and ShapeNet (Wu et al., 2015) 3D encoding ideas. The pipeline is illustrated in Fig. 3. Instead of generating a global 3D voxel-grid prior to processing, we create $16 \times 16 \times 16$ voxel cubes per scan point[6]. We do this at 5 different resolutions, where voxel size ranges from 2.5 cm to 40 cm (multiplied by powers of 2) and encode empty voxel cells as 0 and filled ones as 1. The input to the CNN is thus encoded in a multidimensional tensor with $5 \times 16 \times 16 \times 16$ cube entries per scan point.

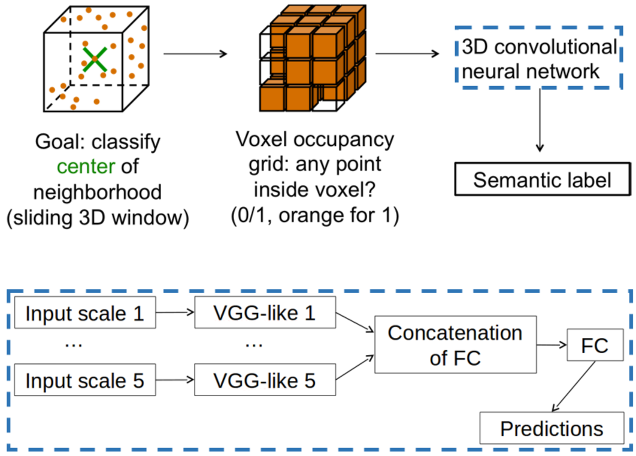


Figure 3: Our deep neural network pipeline.

Each of the five scales is handled separately by a VGG-like network path which includes convolutional, pooling and ReLU layers. The 5 separate network paths are finally concatenated into a single representation, which is passed through two fully-connected layers. The output of the second fully-connected layer is an 8-dimensional vector, which contains the class scores for each of the 8 classes in this benchmark challenge. Scores are transformed to class conditional probabilities with the soft-max function.

Before describing the network architecture in detail we introduce the following notation:

$c(i, o)$ stands for convolutional layers with $3 \times 3 \times 3$ filters, $i$ input channels, $o$ output channels, zero-padding of size 1 at each border and a stride of size 1. $f(i, o)$ stands for fully-connected layers. $r$ stands for a ReLU non-linearity, $m$ stands for a volumetric max pooling with receptive field $2 \times 2 \times 2$, applied with a stride of size 2 in each dimension, $d$ stands for a dropout with 0.5 probability, $s$ stands for a soft-max layer.

Our 3D CNN architecture assembles these components to a VGG-like network structure. We choose the filter size in convolutional layers as small as possible ($3 \times 3 \times 3$), as recommended in recent work (He et al., 2016), to have the least amount of parameters per layer and, hence, reduce both the risk of overfitting and the computational cost.

For the 5 separate network paths that act on different resolutions, we use a VGG-like (Simonyan and Zisserman, 2014) architecture:

$$(c(1, 16), r, m, c(16, 32), r, m, c(32, 64), r, m).$$

The output is vectorized, concatenated between scales and the two fully-connected layers are applied on top to predict the class responses:

$$(f(2560, 2048), r, d, f(2048, 8), s).$$

For training we deploy the standard multi-class cross-entropy loss. Deep learning is non-convex but it can be efficiently optimized with stochastic gradient descent (SGD), which produces classifiers with state-of-the-art prediction performance. The SGD algorithm uses randomly sampled mini-batches of several hundred points per batch to iteratively update the parameters of the CNN. We use the popular adadelta algorithm (Zeiler, 2012) for optimization, an extension of stochastic gradient decent (Bottou, 2010).

We use a mini batch size of 100 training samples (i.e., points), where each batch is sampled randomly and balanced (contains equal numbers of samples per class). We run training for 74,700 batches and sample training data from a large and representative point cloud with 259 million points (sg28_4). A standard pre-processing step for CNNs is data augmentation to enlarge the training set and to avoid overfitting. Here, we augment the training set with a random rotation around the z-axis after every 100 batches. During experiments it turned out that additional training data did not improve performance. This indicates that in our case we rather deal with underfitting (as opposed to overfitting), i.e. our model lacks the capacity to fully capture all the evidence in the available training data[7]. We thus refrain from further possible augmentations like randomly missing points or adding noise.

Our network is implemented in C++ and Lua and uses the Torch7 framework (Collobert et al., 2011) for deep learning. The code and the documentation for this baseline are publicly available at `https://github.com/nsavinov/semantic3dnet`.

## 4. DATA

Our 30 published terrestrial laser scans consist of in total $\approx$ 4 billion 3D points and contain urban and rural scenes, like farms, town halls, sport fields, a castle and market squares. We intentionally selected various different natural and man-made scenes to prevent over-fitting of the classifiers. All of the published scenes were captured in Central Europe and describe typical European architecture as shown in Figure 4. Surveying-grade laser scanners were used for recording these scenes. Colorization was performed in a post processing step by deploying a high resolution cubemap, which was generated from camera images. In general, static laser scans have a very high resolution and are able to measure long distances with little noise. Especially compared to point clouds derived via structure-from-motion pipelines or Kinect-like structured light sensors, laser scanners deliver superior data quality.

Scanner positions for data recording were selected as usually done in the field: only little scan overlap as needed for registration, so that scenes can be recorded in a minimum of time. This free

[6] This strategy automatically centers each voxel-cube per scan point. Note that for the alternative approach of a global voxel grid, several scan points could fall into the same grid cell in dense scan parts. This would require scan point selection per grid cell, which is computationally costly and results in (undesired) down-sampling.

[7] Note that our model reaches hardware limits of our GPU (TitanX with 12GB of RAM) and we thus did not experiment with larger networks at this point.

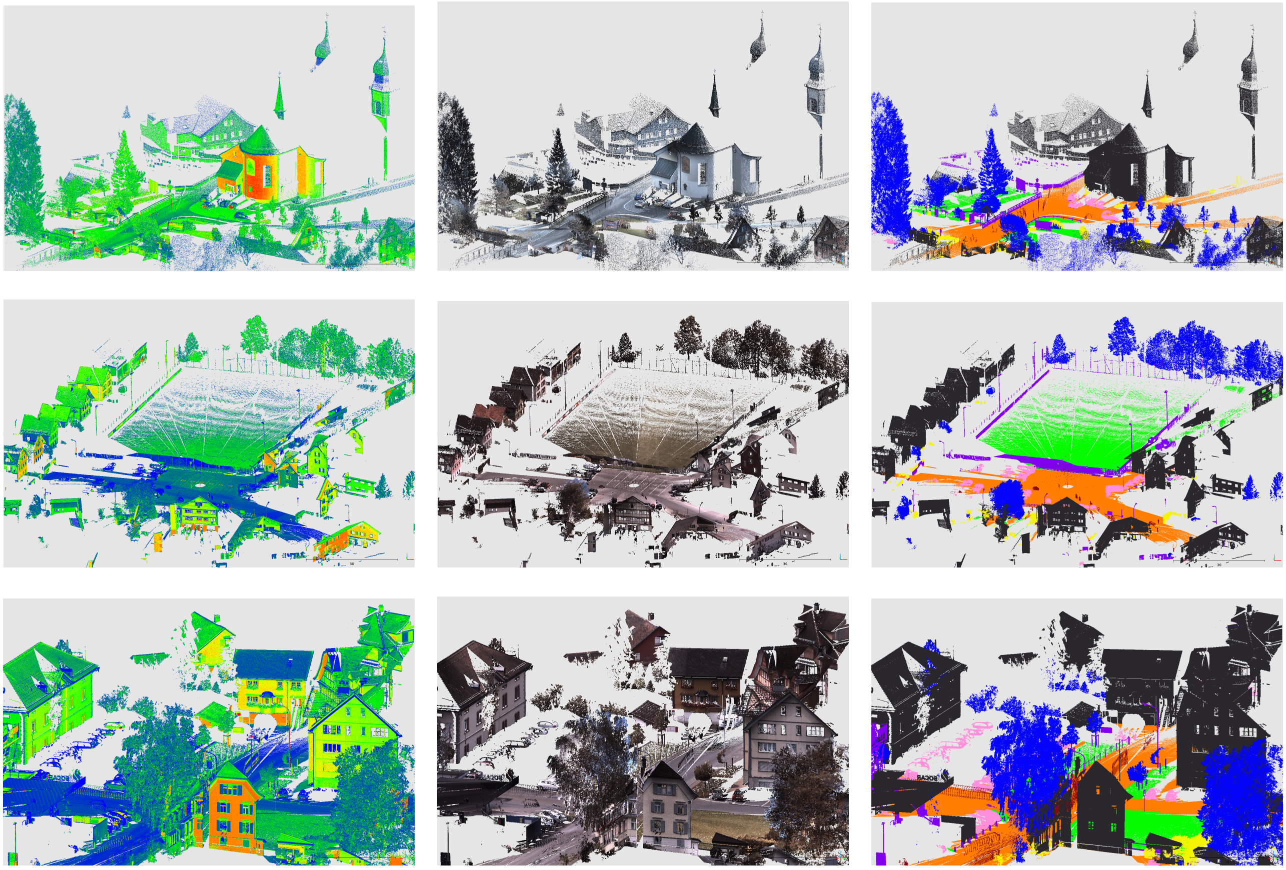


Figure 4: Intensity values (left), rgb colors (middle) and class labels (right) for example data sets.

choice of the scanning position implies that no prior assumption based on point density and on class distributions can be made. We publish up to 3 laser scans per scene that have small overlap. The relative position of laser scans at the same location was estimated from targets.

We use the following 8 classes within this benchmark challenge, which cover: *1)* man made terrain: mostly pavement; *2)* natural terrain: mostly grass; *3)* high vegetation: trees and large bushes; *4)* low vegetation: flowers or small bushes which are smaller than 2 m; *5)* buildings: Churches, city halls, stations, tenements, etc.; *6)* remaining hard scape: a clutter class with for instance garden walls, fountains, banks, etc.; *7)* scanning artifacts: artifacts caused by dynamically moving objects during the recording of the static scan; *8)* cars and trucks. Some of these classes are ill-defined, for instance some scanning artifacts could also go for cars or trucks and it can be hard to differentiate between large and small bushes. Yet, these classes can be helpful in numerous applications. Please note that in most applications class *7*, scanning artifacts, gets filtered with heuristic rule sets. Within this benchmark we want to deploy machine learning techniques instead, and thus do not perform any heuristic pre-processing.

In our view, large data sets are important for two reasons: *a)* Typically, real world scan data sets are large. Hence, methods which have an impact on real problems have to be able to process a large amount of data. *b)* Large data sets are especially important when developing methods with modern inference techniques capable of representation learning. With too small datasets, good results leave strong doubts about possible overfitting; unsatisfactory results, on the other hand, are hard to interpret as guidelines for further research: are the mistakes due to short-comings of the method, or simply caused by unsufficient training data?

### 4.1 Point Cloud Annotation

In contrast to common strategies for 3D data labelling that first compute an over-segmentation followed by segment-labeling, we manually assign each point a class label individually. Although this strategy is more labor-intensive, it avoids inheriting errors from the segmentation approach and, more importantly, classifiers do not learn hand-crafted rules of segmentation algorithms when trained with the data. In general, it is more difficult to label a point cloud by hand than images. The main problem is that it is hard to select a 3D point on a 2D monitor from a set of millions of points without a clear neighbourhood/surface structure. We tested two different strategies:

**Annotation in 3D:** We follow an iterative filtering strategy, where we manually select a couple of points, fit a simple model to the data, remove the model outliers and repeat these steps until all inliers belong to the same class. With this procedure it is possible to select large buildings in a couple of seconds. A small part of the point clouds was labeled with this approach by student assistants at ETH Zurich.

**Annotation in 2D:** The user rotates a point cloud, fixes a 2D view and draws a closed polygon which splits a point cloud into two parts (inside and outside of the polygon). One part usually contains points from the background and is discarded. This procedure is repeated a few times until all remaining points belong to the same class. At the end all points are separated into different layers corresponding to classes of interest. This 2D procedure works well with existing software packages (Daniel Girardeau-Montaut, CloudCompare, 2016) such that it can be outsourced to external labelers more easily than the 3D work-flow. We used this procedure for all data sets where annotation was outsourced.

| Method | $\overline{IoU}$ | $OA$ | $t[s]$ | $IoU_1$ | $IoU_2$ | $IoU_3$ | $IoU_4$ | $IoU_5$ | $IoU_6$ | $IoU_7$ | $IoU_8$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| HarrisNet | **0.623** | 0.881 | unknown | 0.818 | 0.737 | **0.742** | **0.625** | **0.927** | 0.283 | 0.178 | **0.671** |
| DeepSegNet | 0.516 | **0.884** | unknown | 0.894 | **0.811** | 0.590 | 0.441 | 0.853 | **0.303** | **0.190** | 0.050 |
| TMLC-MS | 0.494 | 0.850 | 38421 | **0.911** | 0.695 | 0.328 | 0.216 | 0.876 | 0.259 | 0.113 | 0.553 |
| TML-PC | 0.391 | 0.745 | unknown | 0.804 | 0.661 | 0.423 | 0.412 | 0.647 | 0.124 | 0.0* | 0.058 |

Table 1: Semantic3d benchmark results on the full data set: 3D covariance baseline *TMLC-MS*, 2D RGB image baseline *TML-PC*, and first submissions *HarrisNet* and *DeepSegNet*. *IoU* for categories (1) man-made terrain, (2) natural terrain, (3) high vegetation, (4) low vegetation, (5) buildings, (6) hard scape, (7) scanning artefacts, (8) cars. * Scanning artefacts were ignored for 2D classification because they are not present in the image data.

| Method | $\overline{IoU}$ | $OA$ | $t[s]$ | $IoU_1$ | $IoU_2$ | $IoU_3$ | $IoU_4$ | $IoU_5$ | $IoU_6$ | $IoU_7$ | $IoU_8$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| TMLC-MSR | **0.542** | **0.862** | 1800 | **0.898** | **0.745** | 0.537 | **0.268** | **0.888** | **0.189** | **0.364** | **0.447** |
| DeepNet | 0.437 | 0.772 | 64800 | 0.838 | 0.385 | **0.548** | 0.085 | 0.841 | 0.151 | 0.223 | 0.423 |
| TML-PCR | 0.384 | 0.740 | unknown | 0.726 | 0.73 | 0.485 | 0.224 | 0.707 | 0.050 | 0.0* | 0.15 |

Table 2: Semantic3d benchmark results on the reduced data set: 3D covariance baseline *TMLC-MSR*, 2D RGB image baseline *TML-PCR*, and our 3D CNN baseline *DeepNet*. *TMLC-MSR* is the same method as *TMLC-MS*, the same goes for *TMLC-PCR* and *TMLC-PC*. In both cases *R* indicates classifiers on the reduced dataset. *IoU* for categories (1) man-made terrain, (2) natural terrain, (3) high vegetation, (4) low vegetation, (5) buildings, (6) hard scape, (7) scanning artefacts, (8) cars. * Scanning artefacts were ignored for 2D classification because they are not present in the image data.

## 5. EVALUATION

We follow Pascal VOC challenge's (Everingham et al., 2010) choice of the main segmentation evaluation measure and use *Intersection over Union* ($IoU$)[8] averaged over all classes. Assume classes are indexed with integers from $\{1, \ldots, N\}$ where $N$ is an overall number of classes. Let $C$ be an $N \times N$ confusion matrix of the chosen classification method, where each entry $c_{ij}$ is a number of samples from ground-truth class $i$ predicted as class $j$. Then the evaluation measure per class $i$ is defined as

$$IoU_i = \frac{c_{ii}}{c_{ii} + \sum\limits_{j \neq i} c_{ij} + \sum\limits_{k \neq i} c_{ki}}. \tag{1}$$

The main evaluation measure of our benchmark is thus

$$\overline{IoU} = \frac{\sum\limits_{i=1}^{N} IoU_i}{N}. \tag{2}$$

We also report $IoU_i$ for each class $i$ and overall accuracy

$$OA = \frac{\sum\limits_{i=1}^{N} c_{ii}}{\sum\limits_{j=1}^{N} \sum\limits_{k=1}^{N} c_{jk}} \tag{3}$$

as auxiliary measures and provide the confusion matrix $C$. Finally, each participant is asked to specify the time $T$ it took to classify the test set as well as the hardware used for experiments. This measure is important for understanding how suitable the method is in real-world scenarios where usually billions of points are required to be processed.

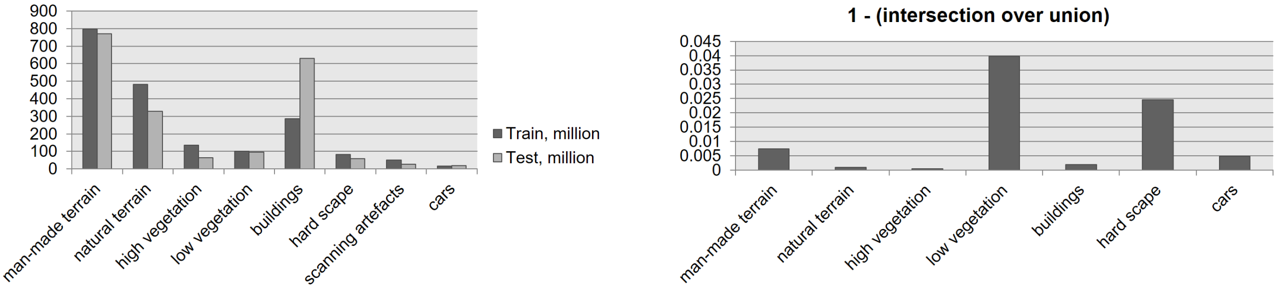
For computational demanding methods we provide a reduced challenge consisting of a subset of the published test data. The results of our baseline methods as well as submissions are shown in Table 1 for the full challenge and in Table 2 for the reduced challenge. Of the three published baseline methods the covariance based method performs better than the CNN baseline and the color based method. Due to its computational cost we could only run our own deep learning baseline *DeepNet* on the reduced data set. We expect a network with higher capacity to perform much better. Results on the full challenge of two (unfortunately yet unpublished) 3D CNN methods, *DeepSegNet* and *HarrisNet*, already beat our covariance baseline by a significant margin (Table 1) of 2 respective 12 percent points. This indicates that deep learning seems to be the way to go also for point clouds, if enough data is available for training. It is a first sign that our benchmark already starts to work and generates progress.

## 6. BENCHMARK STATISTICS

Class distributions in test and training set are rather similar, as shown in Figure 5a. Interestingly, the class with most samples is *man-made terrain* because, out of convenience, operators in the field tend to place the scanner on flat and paved ground. Recall also the quadratic decrease of point density with distance to the scanner, such that many samples are close to the scanner. The largest difference between samples in test and training sets occurs for class *building*. However, this does not seem to affect the performance of the submissions so far. Most difficult classes, *scanning artefacts* and *cars*, have only few training and test samples and a large variation of possible object shapes. *Scanning artefacts* is probably the hardest class because the shape of artefacts mostly depends on the movement of objects during the scanning process. Note that, following discussions with industry professionals, class *hard scape* was designed as clutter class that contains all sorts of man-made objects except houses, cars and ground.

In order to get an intuition of the quality of the manually acquired labels, we also checked the label agreement among human annotators. This provides an indicative measure on how much different annotators agree in labeling the data, and can be viewed as an internal check of manual labeling precision. We roughly estimate the label agreement of different human annotators in areas where different scans of the same scene overlap. Because we cannot rule out completely that some overlapping areas might

[8] $IoU$ compensates for different class frequencies as opposed to, for example, *overall accuracy* that does not balance different class frequencies giving higher influence to large classes.



(a)

(b)

Figure 5: (a) Number of points per class over all scans and (b) ground truth label errors estimated in overlapping parts of adjacent scans.

have been labeled by the same person (labeling was outsourced and we thus do not know exactly who annotated what), this can only be viewed as an indicative measure. Recall that overlaps of adjacent scans can precisely be established via artificial markers in the scene. Even if scan alignments would be perfect without any error, no exact point-to-point correspondences exist between two scans, because scan points acquired from two different locations will never exactly fall onto the same spot. We thus have to resort to nearest neighbor search to find point correspondences. Moreover, not all scan points have a corresponding point in the adjacent scan. A threshold of 5 cm on the distance is used to ignore those points where no correspondence exists. Once point correspondences have been estblished, it is possible to transfer ground truth labels from one cloud to the other and compute a confusion matrix. Note that this definition of correspondence is not symmetric, point correspondences from cloud A in cloud B are not equal to correspondences of cloud B in cloud A. For each pair we calculate two intersection-over-union ($IoU_i$) values which indicate a maximum label disagreement of $< 5\%$. No correspondences can be found on moving objects of course, hence we ignored category *scanning artefacts* in the evaluation in Figure 5b.

## 7. CONCLUSION AND OUTLOOK

The *semantic3D.net* benchmark provides a large set of high quality terrestrial laser scans with over 4 billion manually annotated points and a standardized evaluation framework. The data set has been published recently and only has few submissions, yet, but we are optimistic this will change in the future. First submissions already show that finally CNNs are beginning to outperform more conventional approaches, for example our covariance baseline, on large 3D laser scans. Our goal is that submissions on this benchmark will yield better comparisons and insights into the strengths and weaknesses of different classification approaches for point cloud processing, and hopefully contribute to guide research efforts in the longer term. We hope the benchmark meets the needs of the research community and becomes a central resource for the development of new, efficient and accurate methods for classification in 3D space.

## ACKNOWLEDGEMENT

This work is partially funded by the Swiss NSF project 163910, the Max Planck CLS Fellowship and the Swiss CTI project 17136.1 PFES-ES.

## REFERENCES

Badrinarayanan, V., Kendall, A. and Cipolla, R., 2015. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. arXiv preprint arXiv:1511.00561.

Blomley, R., Weinmann, M., Leitloff, J. and Jutzi, B., 2014. Shape distribution features for point cloud analysis-a geometric histogram approach on multiple scales. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences.

Bottou, L., 2010. Large-scale machine learning with stochastic gradient descent. In: Proceedings of COMPSTAT'2010, Springer, pp. 177–186.

Boykov, Y. and Kolmogorov, V., 2004. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. Transactions on Pattern Analysis and Machine Intelligence.

Boykov, Y., Veksler, O. and Zabih, R., 2001. Fast approximate energy minimization via graph cuts. PAMI.

Brock, A., Lim, T., Ritchie, J. and Weston, N., 2017. Generative and discriminative voxel modeling with convolutional neural networks.

Cavegn, S., Haala, N., Nebiker, S., Rothermel, M. and Tutzauer, P., 2014. Benchmarking high density image matching for oblique airborne imagery. In: Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., Vol. XL-3, pp. 45–52.

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K. and Yuille, A. L., 2016. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. arXiv preprint arXiv:1606.00915.

Choe, Y., Shim, I. and Chung, M. J., 2013. Urban structure classification using the 3d normal distribution transform for practical robot applications. Advanced Robotics 27(5), pp. 351–371.

Collobert, R., Kavukcuoglu, K. and Farabet, C., 2011. Torch7: A matlab-like environment for machine learning. In: BigLearn, NIPS Workshop.

Comaniciu, D. and Meer, P., 2002. Mean shift: A robust approach toward feature space analysis. PAMI.

Daniel Girardeau-Montaut, CloudCompare, 2016. http://www.danielgm.net/cc/.

De Deuge, M., Quadros, A., Hung, C. and Douillard, B., 2013. Unsupervised feature learning for classification of outdoor 3d scans. In: Australasian Conference on Robitics and Automation, Vol. 2.

Demantké, J., Mallet, C., David, N. and Vallet, B., 2011. Dimensionality based scale selection in 3d lidar point clouds. The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. and Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, pp. 248–255.

Engelcke, M., Rao, D., Wang, D. Z., Tong, C. H. and Posner, I., 2017. Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks.

Everingham, M., van Gool, L., Williams, C., Winn, J. and Zisserman, A., 2010. The pascal visual object classes (voc) challenge. International Journal of Computer Vision 88(2), pp. 303–338.

Fukushima, K., 1980. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biological cybernetics 36(4), pp. 193–202.

Greene, N., 1986. Environment mapping and other applications of world projections. IEEE Computer Graphics and Applications 6(11), pp. 21–29.

Haala, N., 2013. The landscape of dense image matching algorithms. In: Photogrammetric Week 13, pp. 271–284.

Hackel, T., Wegner, J. D. and Schindler, K., 2016. Fast semantic segmentation of 3D point clouds with strongly varying point density. In: ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. III-3, pp. 177–184.

He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778.

Hussain, S. and Triggs, B., 2012. Visual recognition using local quantized patterns. In: European Conference on Computer Vision.

Kohli, P., Ladicky, L. and Torr, P. H. S., 2008. Robust higher order potentials for enforcing label consistency. In: Conference on Computer Vision and Pattern Recognition.

Krizhevsky, A., Sutskever, I. and Hinton, G. E., 2012. Imagenet classification with deep convolutional neural networks.

Ladicky, L., Russell, C., Kohli, P. and Torr, P., 2013. Associative hierarchical random fields. PAMI.

Ladický, L., Zeisl, B. and Pollefeys, M., 2014. Discriminatively trained dense surface normal estimation. In: European Conference on Computer Vision, pp. 468–484.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. and Jackel, L. D., 1989. Backpropagation applied to handwritten zip code recognition. Neural computation 1(4), pp. 541–551.

Long, J., Shelhamer, E. and Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440.

Lowe, D. G., 2004. Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision.

Malik, J., Belongie, S., Leung, T. and Shi, J., 2001. Contour and texture analysis for image segmentation. International Journal of Computer Vision.

Maturana, D. and Scherer, S., 2015. Voxnet: A 3d convolutional neural network for real-time object recognition. In: Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on, IEEE, pp. 922–928.

Monnier, F., Vallet, B. and Soheilian, B., 2012. Trees detection from laser point clouds acquired in dense urban areas by a mobile mapping system. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences.

Montoya, J., Wegner, J. D., Ladický, L. and Schindler, K., 2014. Mind the gap: modeling local and global context in (road) networks. In: German Conference on Pattern Recognition (GCPR).

Munoz, D., Bagnell, J. A., Vandapel, N. and Hebert, M., 2009. Contextual classification with functional max-margin markov networks. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE, pp. 975–982.

Riegler, G., Ulusoy, A. O. and Geiger, A., 2017. Octnet: Learning deep 3d representations at high resolutions.

Riemenschneider, H., Bódis-Szomorú, A., Weissenberg, J. and Van Gool, L., 2014. Learning where to classify in multi-view semantic segmentation. In: European Conference on Computer Vision, Springer, pp. 516–532.

Rottensteiner, F., Sohn, G., Gerke, M. and Wegner, J. D., 2013. ISPRS Test Project on Urban Classification and 3D Building Reconstruction. Technical report, ISPRS Working Group III / 4 - 3D Scene Analysis.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. and Fei-Fei, L., 2015. Imagenet Large Scale Visual Recognition Challenge. International Journal of Computer Vision 115(3), pp. 211–252.

Serna, A., Marcotegui, B., Goulette, F. and Deschaud, J.-E., 2014. Paris-rue-madame database: a 3d mobile laser scanner dataset for benchmarking urban detection, segmentation and classification methods. In: 4th International Conference on Pattern Recognition, Applications and Methods ICPRAM 2014.

Shechtman, E. and Irani, M., 2007. Matching local self-similarities across images and videos. In: Conference on Computer Vision and Pattern Recognition.

Shotton, J., Winn, J., Rother, C. and Criminisi, A., 2006. *TextonBoost*: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In: European Conference on Computer Vision.

Silberman, N., Hoiem, D., Kohli, P. and Fergus, R., 2012. Indoor segmentation and support inference from rgbd images. In: European Conference on Computer Vision, Springer, pp. 746–760.

Simonyan, K. and Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Song, S., Lichtenberg, S. P. and Xiao, J., 2015. Sun rgb-d: A rgb-d scene understanding benchmark suite. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 567–576.

Torralba, A., Fergus, R. and Freeman, W. T., 2008. 80 million tiny images: A large data set for nonparametric object and scene recognition. IEEE transactions on pattern analysis and machine intelligence 30(11), pp. 1958–1970.

Torralba, A., Murphy, K. and Freeman, W., 2004. Sharing features: efficient boosting procedures for multiclass object detection. In: CVPR.

Vallet, B., Brédif, M., Serna, A., Marcotegui, B. and Paparoditis, N., 2015. Terramobilita/iqmulus urban point cloud analysis benchmark. Computers & Graphics 49, pp. 126–133.

Weinmann, M., Jutzi, B. and Mallet, C., 2013. Feature relevance assessment for the semantic interpretation of 3d point cloud data. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences.

Weinmann, M., Urban, S., Hinz, S., Jutzi, B. and Mallet, C., 2015. Distinctive 2d and 3d features for automated large-scale scene analysis in urban areas. Computers & Graphics 49, pp. 47–57.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X. and Xiao, J., 2015. 3d shapenets: A deep representation for volumetric shapes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1912–1920.

Zeiler, M. D., 2012. Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701.

Zhuang, Y., He, G., Hu, H. and Wu, Z., 2014. A novel outdoor scene-understanding framework for unmanned ground vehicles with 3d laser scanners. Transactions of the Institute of Measurement and Control p. 0142331214541140.

Zhuang, Y., Liu, Y., He, G. and Wang, W., 2015. Contextual classification of 3d laser points with conditional random fields in urban environments. In: Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on, IEEE, pp. 3908–3913.