# LEARNING ON THE EDGE: BENCHMARKING ACTIVE LEARNING FOR THE SEMANTIC SEGMENTATION OF ALS POINT CLOUDS

M. Kölle*, V. Walter, S. Schmohl, U. Soergel

Institute for Photogrammetry, University of Stuttgart, Germany -
(michael.koelle, volker.walter, stefan.schmohl, uwe.soergel)@ifp.uni-stuttgart.de

**KEY WORDS:** Active Learning, Random Forest, Submanifold Sparse CNN, 3D Point Clouds, Semantic Segmentation.

**ABSTRACT:**

While most research in automatic semantic segmentation of 3D geospatial point clouds is concerned with enhancing respective Machine Learning (ML) models, we aim to shift the focus to be more of a data-centric nature. This means, we consider the creation of respective data sets that ML models learn from as key component, since even the most sophisticated model performs poorly when learning from suboptimal data. In this regard, the straightforward approach of providing labeled data abundantly is prohibitively expensive and just not scalable in times of high-frequency data acquistion cycles, where a dedicated training set should be available for each new epoch, as ML models often lack generalizability. As a remedy, we rely on Active Learning (AL), which is a cost-efficient and quick method to generate required training data at scale. Although AL has been (scarcely) applied in the geospatial domain before, a comprehensive evaluation of its capabilities, including benchmarking of achievable accuracies is lacking. Therefore, we apply the AL concept to both ISPRS' current point cloud benchmark data sets as well as to a third large scale National Mapping Agency point cloud. Respective experiments are conducted with both a feature-driven Random Forest classifcation approach and a data-driven Submanifold Sparse Convolutional Neural Network classifier. Our experiments verify that by labeling only a fraction of available training points (typically $\ll 1\,\%$), we can still reach accuracies that are at maximum only about 5 percentage points worse compared to leading benchmark contributions.

## 1. INTRODUCTION

Being capable to automatically interpret (geo)spatial 3D data enables a plethora of different applications, such as safe autonomous vehicle navigation through surrounding awareness, derivation of digital terrain models (Hui et al., 2019) or detection of significant changes in monitoring applications (Haala et al., 2020). To this end, supervised Machine Learning (ML) methods are often employed and have drawn considerable attention in research over the last decade. While conventional feature-driven classification approaches have achieved a rather mature state, the branch of data-driven Convolutional Neural Network (CNN) approaches, triggered by the introduction of *PointNet* (Qi et al., 2017), is a hot research topic currently. However, the main focus of ML, especially in the geospatial domain, has always been on the classification model rather than the careful generation of the training data set the model is supposed to learn from. For the latter, the long-held standard is that providing a sufficient training data set is an expert's burden and has to be completed before any model can be employed (Waldhauser et al., 2014). But only recently, Ng (2021) stressed that more emphasis should be given to the creation of training data sets, and recommended that ML system development should be more data-centric rather than model-centric.

One scheme following this mindset is Active Learning (AL) (Settles, 2009). In this iterative supervised ML approach, data annotation and training a respective model are no longer seen as two self-contained steps, but the machine represented by the ML model is actively involved in constructing the training set and is allowed to request labels for specific instances from one or more human labelers, known as the oracle. The basic idea behind determining such points is that predictive uncertainty of the model is directly correlated with informativeness and that by adding such points, the model improves, i.e., we seek to minimize epistemic uncertainty. With AL, the labeling effort can be focused only on those points that actually justify human involvement. Therefore, when realizing an AL framework, we build what we call hybrid intelligence systems (Vaughan, 2018), in which humans or *human processing units* work together with *electronic processing units*, so that both parties perform the tasks they are best at, i.e., human interpretation capabilities for data annotation and machine-based scanning through data highlighting potentially valuable instances.

Despite the great potential to perform cost-efficient data interpretation, for 3D point clouds, especially Airborne Laser Scanning (ALS) point clouds, respective AL-based approaches are scarce. One of the first methods to this end was proposed by Luo et al. (2018), who perform semantic segmentation of mobile laser scanning point clouds by means of a pair-wise conditional random field built upon an Random Forest (RF) classifier (Breiman, 2001) integrated into an AL loop. Also for the classification of terrestrial point clouds, AL-based solutions are presented by Wu et al. (2021), Shi et al. (2021) and Shao et al. (2022), each relying on superpoint regions as AL primitives (instead of single points) but differing in the sampling procedure.

An AL approach actually developed for ALS point clouds is presented by Hui et al. (2019), who formulate the generation of a Digital Terrain Model (DTM) as a binary classification problem where points are to be assigned to class *ground* or *non-ground*, but utilize an automated oracle based on both the current prediction and the distance to the approximated DTM level. To predict a more extensive class catalog, Li and Pfeifer (2019) combine AL built around an RF classifier with a semi-supervised learning scheme in which labels of an initially provided coarse training set are each propagated to the point

* Corresponding author

in an optimal neighborhood that exhibits the highest sampling score. A more typical AL scheme for semantic segmentation of ALS points clouds is pursued by Lin et al. (2020a,b). In this approach, AL operates on a regularily tiled point cloud and is designed to identify most-informative tiles, with tile scores obtained by averaging either point-based or segment-based sampling scores derived from a *PointNet++* classifier (the segments are obtained by a preceding unsupervised segmentation). Although this approach greatly minimizes labeling effort to most-informative tiles, costly full annotations of those are still expected. Kölle et al. (2020) and Kölle et al. (2021b) mitigate this issue by only requesting labels of single most-informative points, which are identified based on both RF and Submanifold Sparse Convolutional Neural Network (SCN) prediction scores. Furthermore, the assumption of an error-free Ground Truth (GT) oracle is lifted, as labels are provided directly by crowdworkers, thus completely excluding experts from the annotation process and actually forming a hybrid intelligence system.

While the aforementioned approaches have demonstrated great potential for cost-efficiently building ML models for a given data set as alternative to the conventional Passive Learning (PL) approach, they lack a comprehensive ranking of results compared to the current state of the art in semantic point cloud segmentation that allows for a fair ranking of AL. In this work, we aim to address this limitation and hope to thereby foster a wider dissemination of AL in the geospatial domain in the spirit of data-centric ML (Ng, 2021). Our contribution can thus be summarized as follows: i) We give a brief overview of AL, but particularly illustrate its working principle for ALS point clouds, followed by ii) a discussion of versatile add-ons for the key component of AL, namely the definition of a query function to identify most-informative samples, suitable for both data-driven and feature-driven classification approaches, and iii) we benchmark AL results for both an RF and SCN classifier by applying the respective methods to both ISPRS' semantic labeling challenges for 3D point clouds as well as to a typical National Mapping Agency (NMA) ALS cloud.

## 2. METHODOLOGY

Our system to efficiently train ML models consists of three main components, namely the query function for sampling most-informative samples in context of the AL loop (Section 2.1), an appropriate ML model (Section 2.2), and our oracle capable of returning labels for selected instances (Section 2.3).

### 2.1 Setting-up the AL Loop

To initialize our pipeline (cf. Algorithm 1), we present a given unlabeled point cloud $U$ to our oracle $\mathcal{O}$, which can either be a simulated machine oracle or, more realistically, can be represented by human operators. The first task of the oracle is then to generate an initial (coarse) training set with $n_j$ samples for each of our $n_\Omega$ classes. When humans are asked to perform this task, they will naturally select samples that are fairly easy to label well away from respective class borders in object space. Using $L_{init}$ (cf. Algorithm 1), we can then train a respective ML model $M$ capable to perform semantic segmentation of 3D point clouds, and rely on it to derive predictions on the remaining unlabeled data set $U$, so that the loop can theoretically be terminated already at this point. However, if we aim at high-accuracy results, the loop/iteration is to be continued and thus the second and even more important task of the classifier is to

**Algorithm 1** AL loop for 3D point cloud classification

**Input**
- unlabeled point cloud $U = \{\boldsymbol{x}_u\}_{u=1}^{n_p}$
- numb. of samples $n^+$ to be labeled in each iteration step $i$
- definition of desired class catalog containing $n_\Omega$ classes
- access to oracle $\mathcal{O}$

initialize labeled training set $L = \{\}$
query $\mathcal{O}$ to generate initialization data set $L_{init} = \{(\boldsymbol{x}_r, c_r)\}_{r=1}^{n_j \cdot n_\Omega}$ containing $n_j$ samples per class
set $L = L \cup L_{init}$ and $U = U \setminus L_{init}$
initialize queried label set $L_i = \{\}$
**while** stopping criterion not met or labeling budget not exhausted **do**
    train the ML model $M$ using $L$
    predict on $U$ to get $\boldsymbol{p}(\boldsymbol{c}|\boldsymbol{x}_u)\ \forall\ \boldsymbol{x}_u \in U$
    derive a sampling score $s\ \forall\ \boldsymbol{x}_u \in U$
    select the $n^+$ samples with highest score and ask $\mathcal{O}$ for labels thus generating $L_i = \left\{\left(\boldsymbol{x}_i^+, c_i^+\right)\right\}_{i=1}^{n^+}$
    set $L = L \cup L_i$ and $U = U \setminus L_i$
    set $L_i = \{\}$
**end while**

**Output**
- final training set $L$
- trained model $M$
- full annotation of originally provided point cloud $U$ with points being either manually annotated by $\mathcal{O}$ or automatically labeled by $M$

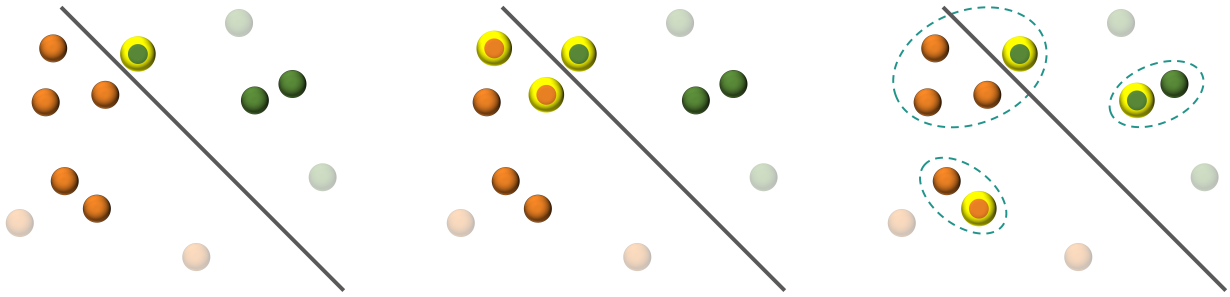
estimate the model's confidence by means of the predicted posterior probabilities $\boldsymbol{p}(\boldsymbol{c}|\boldsymbol{x})$ for each point $\boldsymbol{x} \in U$. To actually select most-informative points from this unlabeled pool, we rely on *entropy* sampling defined as:

$$\boldsymbol{x}_E^+ = \underset{\boldsymbol{x} \in U}{\operatorname{argmax}} \left( -\sum_{i=1}^{n_\Omega} p(c_i|\boldsymbol{x}) \cdot \log_2 p(c_i|\boldsymbol{x}) \right) \tag{1}$$

Generally speaking, this measure is designed to sample points in the vicinity of the current (perhaps suboptimal) class borders (cf. Figure 1*(a)*), i.e., we score aleatoric uncertainty, but especially in early iteration steps, epistemic uncertainty will also have a significant impact. This can be interpreted as mimicking the core idea of Support Vector Machines (SVMs), that is building separation hypotheses solely based on samples situated near class borders, essentially. However, when only uncertainty is scored, for ALS points clouds where we are typically confronted with heavily class-imbalanced data sets (e.g., consider the relative frequency of class *Car* vs. *Impervious Surface*), it is likely that classes that are underrepresented in the underlying data set are all the more underrepresented in our sampled training set. This is because (most likely) regions of class borders are populated by proportionally fewer representatives of such smaller classes. Thus, refinement of class borders with respect to these classes is likely to be neglected, eventually resulting in suboptimal separability. As a remedy, in each iteration step $i$ of our loop (cf. algorithm 1), we compute dynamic class weights $w_c$ based on the relative frequency of the number of samples of a specific class $n_c$ in our current training set $L$ with $n_L$ points.

$$w_c(i) = \frac{n_L(i)}{n_c(i)} \tag{2}$$

Those weight values are then multiplied by the predicted posterior probabilities, normalized, and inserted into the *entropy* formula in Equation 1. However, such AL sampling strategies are designed to add only one instance at a time, but re-training an ML model each time only one sample is added is both ineffi-



(a) *Entropy* sampling - emphasizing aleatoric uncertainty, but also epistemic uncertainty (especially in early iteration steps)

(b) Batch-mode AL with batch size 3 in combination with *entropy* sampling. Except for the point closest to the decision border, 2 quasi-duplicates get selected

(c) Batch-mode AL with batch size 3 in combination with *entropy* sampling and applied diversity criterion. *Dotted lines* represent formed *k-means* clusters

Figure 1. Comparison of different sampling strategies to select most informative points. *Transparent* points represent the current training data defining the decision boundary. *Yellow border lines* indicate samples with highest scores.

cient and statistically questionable (especially in case of a CNN model). Thus, AL is usually applied in batch-mode, where multiple $n^+$ samples are selected and presented to the oracle $\mathcal{O}$ for labeling. But in this case, it is likely that queried points are too similar to each other with respect to their representation in feature space (cf. Figure 1*(b)*). Thus, sampling such quasi-duplicates essentially wastes labeling resource. To get the most out of a fixed labeling budget, we therefore follow the recommendation of Zhdanov (2019) and compute a weighted *k-means* clustering with $n^+$ clusters according to:

$$\sum_{\boldsymbol{x}_i \in U} \sum_{j=1}^{n^+} s_i \left\| \boldsymbol{x}_i - \boldsymbol{\mu}_j \right\| \rightarrow \min \qquad (3)$$

where $\boldsymbol{\mu}$ are cluster centers

By explicitly considering the (weighted) *entropy* scores $s$ in clustering, we can guarantee that in this *Diversity in Feature Space* (*DiFS*) method, we sample a batch of points that is both as informative and as diverse as possible in order to boost the convergence of the loop (cf. Figure 1*(c)*). After determining the points to be added to the training set, the oracle $\mathcal{O}$ is asked to annotate these points, so that the ML model can be re-trained based on this expanded training set to complete the first training cycle. This iteration continues until a certain stopping criterion is reached (e.g., a fixed labeling budget, a certain number of iteration steps, or a more sophisticated stopping criterion as discussed by Bloodgood and Vijay-Shanker (2009)) and eventually results in an optimal training set tailored to the specific ML model $M$.

To get an intuition of the working principle of AL, it is worthwhile to examine samples that have been identified as informative within the loop. As humans, we tend to utilize the object space to this end. However, AL queries are based on the representation of instances in a high-dimensional feature space, which should then be the focus of such an analysis. But since such a representation is hard for humans to interpret, we should apply a re-mapping of high-dimensional spaces to 2D for visualization. For this, we rely on the non-linear *t-SNE* mapping (van der Maaten and Hinton, 2008) that aims to keep relative distances between samples based on their similarity. This is exemplarily applied to the feature description of the ISPRS Vaihingen 3D (V3D) data set (used features and the data set is briefly introduced in Sections 2.2 and 3, respectively) and yields the 2D feature space visualization in Figure 2. For exemplary points selected within an AL loop launched for this data set, we trace back respective point in feature space to object space. In this regard, Figure 2 corresponds well to our expectation that human-selected points in the initialization phase are typically easy for the machine to interpret, as they are situated well away from class borders in feature space and populate centers of rather homogeneous regions (cf. Figure 2*(a)* & *(f)*). Points sampled within the loop, on the other hand, naturally stem from inhomogeneous regions in feature space that correspond to spots near class borders in object space (cf. Figure 2*(b)-(e)*). Thus, as previously mentioned, AL can in fact be interpreted as emulating the working principle of SVMs, but focusing not only on selecting most-informative points but also on avoiding to sample quasi-duplicates. This typically minimizes labeling effort to only a really small fraction of available training points (Mackowiak et al., 2018; Kellenberger et al., 2019).

### 2.2 The ML Model

Although the basic assumption is that even the simplest classifier can perform well just by tailoring an appropriate training set to it (Ho and Baird, 1997; Stork, 1999), still the achievable performance will be partly determined by the suitability of the employed model. To demonstrate generalizability of results, we thus rely on both a representative of the feature-driven domain, an RF classifier, and a representative of the data-driven domain, a 3D-convolution-approximating, voxel-based SCN classifier, which is based on the work of Schmohl and Sörgel (2019). For an ML model to be successfully incorporated into AL, it *i)* needs to be capable to learn from sparsely labeled data, *ii)* must be suitable reliably assessing its uncertainty - especially, its epistemic uncertainty, which we seek to minimize, and *iii)* has to be provided with/needs to be capable of inferring, explicit point-wise feature vectors to guarantee diversity within sampled batches.

For the RF classifier, the latter requirement is met by design, as we utilize hand-crafted features. Precisely, we use a set of both geometric (structural tensor features, orientation of fitted plane, roughness, height above ground etc.) and radiometric features (LiDAR inherent features and color information) evaluated for multi-scale spherical neighborhoods, as described in the work of Haala et al. (2020). Also, learning from sparsely labeled data (challenge *i)*) can be straightforwardly implemented for the RF, as we simply reduce the list of samples provided

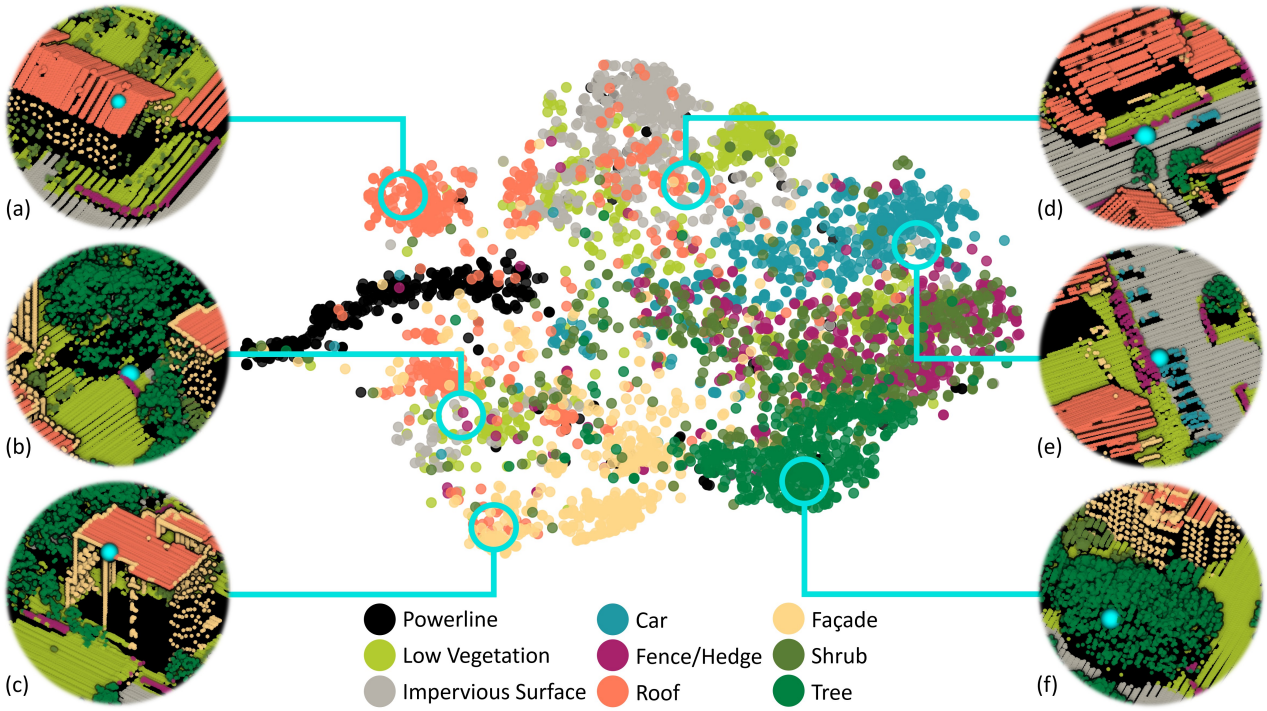



Figure 2. *t-SNE* embedding of feature vectors of the V3D data into 2D space. Exemplary regions where AL points originate from are indicated by *blue circles*. For each region, a representative is traced back to object space and colored *blue*. While points *(a)* and *(f)* were selected by human operators in the initialization step, the remaining examples were actively queried in the course of the AL loop.

for training. Furthermore, we argue that the predicted (pseudo) posterior probability of the RF is well suited to assess epistemic uncertainty, as it is the result of averaging over multiple bagging ensemble members and thus satisfies condition *ii)*.

As for the representative of the data-driven domain, the aforementioned challenges are more complex to overcome. Usually, ML models compute the loss over all labeled instances (or voxels in our case). However, dealing with sparse annotations, not every voxel carries a label, but should still be presented to the network to enable it to derive meaningful geometric descriptors (at least if it lies within the receptive field of one of the few labeled voxels, i.e., if it describes the neighborhood of labeled cells). Thus, to address *i)*, we modify the loss function so that unlabeled "background" voxels are ignored in loss calculation, but still contribute in training due to their passive presence. To address *ii)*, we employ a so-called deep ensemble, where each ensemble member is trained on the same training set but they differ in the randomly initialized weight values. In inference, we then compute the average over all ensemble-wise posterior probabilities to reliably estimate epistemic uncertainty (Jospin et al., 2022).

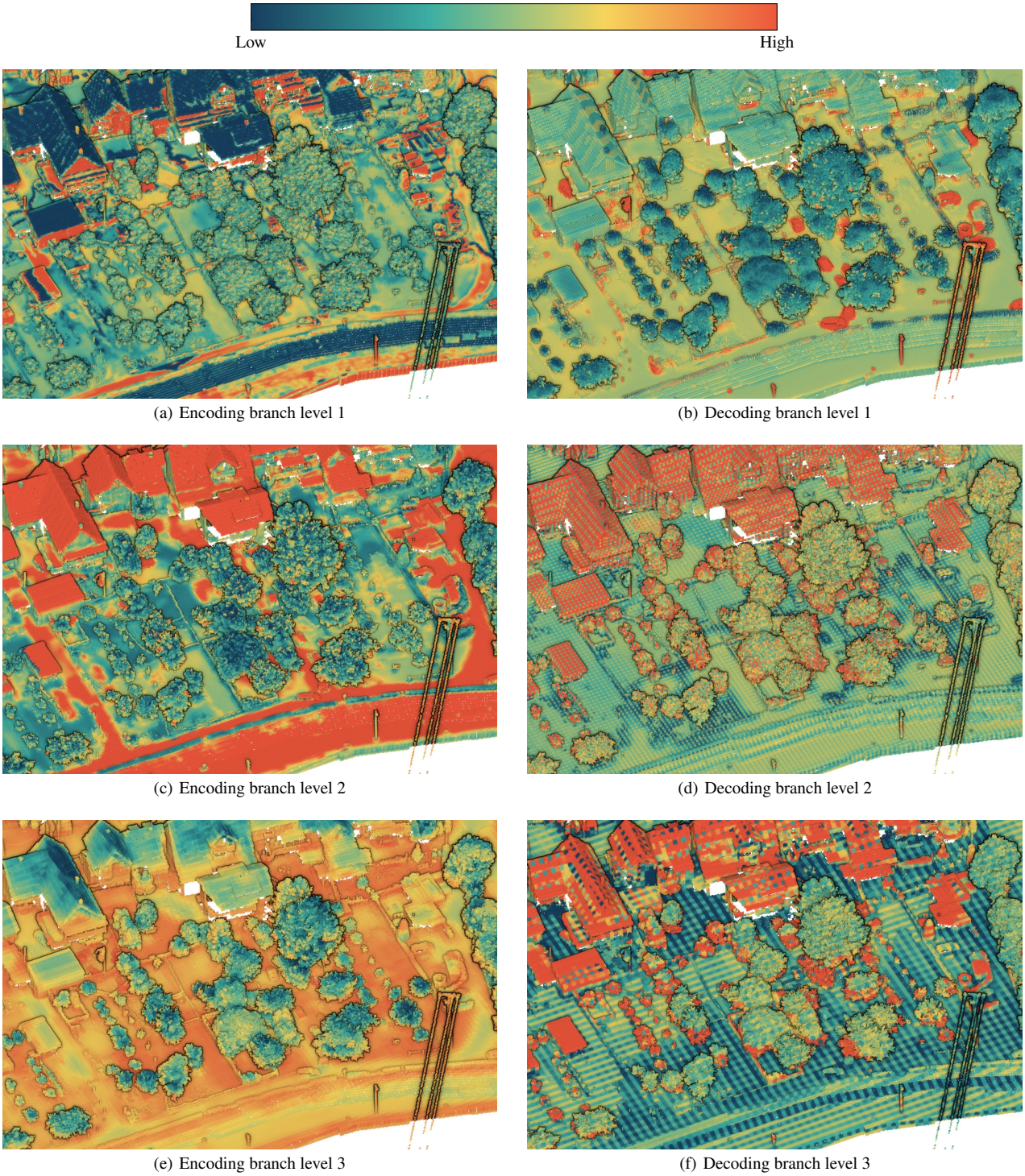
Although the network implicitly utilizes self-taught features, for *iii)*, we need to find a way to explicitly output point-wise feature vectors. To do so, we concatenate filter responses of the different levels of our 3-level U-Net like architecture from both the encoding and decoding branch to obtain a multi-scale description of our input points. However, at deeper levels, the original input voxel cloud is represented in a more abstract manner at a lower resolution than the input. As a remedy, we assign respective features of deeper levels to all voxels at the original resolution that have been aggregated into this specific cell. As can be seen from Figure 3, this often leads to a voxelated representation where upsampled filter responses from deeper encoding levels are smoother than their counterparts from decoding levels (although stemming from the same lower resolution). This is due to retrieving features in the decoding branch directly at the deconvolutional layer, essentially incorporating the resolution of the previous deeper level, which is contrary to the encoding branch where features are retrieved after a series of 3D convolutions at the last layer of an encoding level.

Obtained filter responses of the encoding branch in Figure 3 often resemble typical features utilized by feature-driven classifiers. For instance, Figure 3*(a)* is reminiscent of a verticality measure and Figure 3*(c)* seems to score flatness. However, both responses also appear to be impacted by radiometric features, as convolutions are performed over all available input channels. Also, the model tries to gradually enhance its context awareness with Figure 3*(e)* resembling height above ground, which can only be inferred from a wider spatial context. Contrary to the encoding branch, where the data is solely described by deriving descriptive features, in the decoding branch the model progressively develops its ability to recognize individual classes. In this regard, Figure 3*(f)* attempts to accentuate buildings, but also lower parts of high vegetation that are often geometrically similar (both are vertically oriented and noisy, either due to façade furniture or detailed branch structures), but are already far less emphasized in Figure 3*(d)*. Eventually, Figure 3*(b)* is clearly suited to extract points of a specific class, in this case class *Car*.

### 2.3 The AL Oracle

Another key component of AL is the formulation of an oracle capable of providing labels for selected points. In literature, an omniscient GT oracle $\mathcal{O}_O$ is often assumed, but this is unrealistic in real world scenarios where humans are tasked with point annotation. Thus, labeling errors should also be taken into account when simulating oracles. Respective errors can be either

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-1-W1-2023-945-2023 | © Author(s) 2023. CC BY 4.0 License.




(a) Encoding branch level 1 (b) Decoding branch level 1

(c) Encoding branch level 2 (d) Decoding branch level 2

(e) Encoding branch level 3 (f) Decoding branch level 3

Figure 3. Filter responses from selected filters at each level of our SCN, arranged in an order to match the SCN's $U$ shape.

purely random or systematic in nature (Lockhart et al., 2020). A noisy oracle $\mathcal{O}_N$ will always assign a fraction of points to any class, except the correct one. But more severely, a confused oracle will follow some distinct mapping function (e.g., always labeling façades as class *Roof*), which can be particularly harmful for classification approaches (Kölle et al., 2021b). Especially in AL, this becomes problematic since we sample points from class borders (both in feature and object space, cf. Figure 2), where selected points are often ambiguous and thus systematic errors can be the result of different class understanding. To avoid such errors, as recommended by Kölle et al. (2021b), we modify our sampling strategy slightly and consider the point originally queried by the machine (cf. Section 2.1) as seed point only, but instead select the neighboring point in a spherical neighborhood of radius $d_{RIU}$ with the lowest sampling score. This strategy, referred to as Reducing Interpretation Uncertainty (RIU), assumes that the distance to the class border correlates directly with annotation complexity and has proven an efficient means of minimizing systematic labeling errors (Kölle et al., 2021b).

## 3. DATA SETS

As our main goal is to benchmark AL in the domain of geospatial ALS point cloud semantic segmentation, we rely on both ISPRS' current benchmark data sets. These are the Vaihingen 3D Semantic Labeling Contest (V3D) as a typical ALS point cloud (Niemeyer et al., 2014) and the high-resolution Hessigheim 3D Benchmark (H3D) captured from an UAV (Kölle et al., 2021a). Although both data sets incorporate rich and challenging class catalogs, they cover only limited spatial regions. Therefore, we utilize as a third data set, an NMA ALS point cloud depicting the city center of Stuttgart (S3D), that is about 30 times larger in extent than the V3D data set, but contains a comparably small class catalog, as can be seen from Table 3. Nevertheless, it is well suited for evaluating the scalability of AL.

## 4. EXPERIMENTS

To assess the capabilities of AL for semantic point cloud classification, we derive a series of solutions for our three data sets that incorporate the different strategies and classifiers described in Section 2.1 & 2.2. We report results of pure weighted entropy sampling (*wE*) as well as the adapted variant with the *DiFS* sampling add-on. But to also give realistic estimates of accuracies to be expected in an AL scenario where *human processing units* are employed for labeling the queried points, we i) augment sampling with *RIU*, to reduce chances for encountering an oracle following a systematic error behavior, and ii) incorporate a noisy oracle $\mathcal{O}_N$ where 10 % of labels are randomly misclassified in each iteration step. In each of our AL runs, the initial data sets consist of $n_j = 10$ samples per class. Unless stated otherwise, we report AL results after 30 iteration steps with 300 points queried in each step, exclusively from the dedicated training set, predicting on the respective test splits (i.e., we adhere to the official data splits for the benchmark data sets). As for the incorporated ML models, the RF is parametrized by 100 binary decision trees with a maximum depth of 18 and a minimum number of samples at a node to justify a new split of 7. Respective features are computed for spherical neighborhoods of $r \in \{1, 2, 3, 5\}$ m. For the SCN classifier, we employ a deep ensemble of 5 networks, each operating on a 0.5 m voxelized input point cloud. To reduce computation time, networks of each iteration step start their training cycle based on the result of the previous iteration step and use the current decayed learning rate. Apart from these AL runs, we rely on both the PL results of our classifiers using the fully labeled training set and the PL result of the respective benchmark leader (for V3D & H3D) as baseline solutions.

As for the results for the V3D data set, we can firstly conclude from Table 1 that both our classifiers are well suited for the task at hand, as our PL results are on a level comparable to the top-performing benchmark submission, and are only worse by about 1 percentage point (pp) in Overall Accuracy (OA). However, we prefer comparing our AL-based runs to the PL result obtained with our classifiers, as these can be considered the limit of achievable accuracy for the specific model. Regarding the AL runs, it is evident that the *DiFS* sampling add-on contributes significantly to the improvement of the classification accuracy, so that the *wE+DiFS* strategy can be considered as optimal result from the point of the machine, performing less than 3 pp worse in OA compared to PL for both the RF and SCN classifier. However, in a realistic scenario with imperfect human operators as oracle, these accuracies are unlikely to be

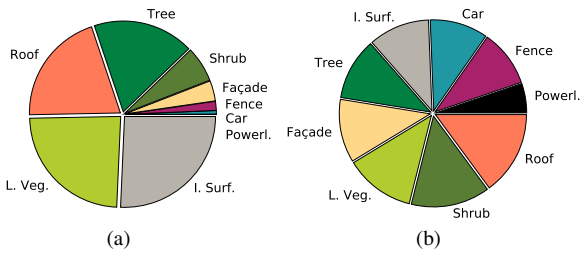



Figure 4. Comparison between the class distributions in the original V3D training data set *(a)* vs. the one obtained by AL after 30 iteration steps *(b)*.

achieved. Thus, we add the *RIU* technique with $d_{RIU} = 1.5$ m to minimize chances of systematic errors and consequently simulate only the effect of a noisy oracle $\mathcal{O}_N$. Such more realistic AL runs perform only marginally worse with a final loss of $< 5$ pp in OA compared to the best-performing PL benchmark submissions, but are far more cost-efficient since only 1.15 % of points from the training set require labeling.

With respect to the performance of individual classes, underrepresented categories such as *Powerline* or *Car* tend to perform better in AL than in their PL counterparts. This effect can be traced back to the generation of a training set in AL, which, thanks to the weighted sampling scheme (cf. Section 2.1), has a distribution that is close to that of an equal distribution, as clearly visible from Figure 4.

As for the RF classifier vs. the SCN classifier, results are rather similar, with the RF slightly outperforming the SCN. However, the two models differ significantly in computational complexity, which is due to their basic working principle. With the RF, features of each point only need to be computed once and can be kept throughout the iteration. But for the SCN, whenever new labels become available, we need to recompute or at least refine features of all points (voxels), which is inevitably computationally more expensive. Precisely, an RF-based AL iteration step can be completed in about 1 minute, whereas such a training cycle for the SCN takes about 50 times as long. Therefore, for AL, CNN-based approaches are a suboptimal choice - at least from a purely economic point of view.

Hence, for the high-resolution H3D data set incorporating a significantly larger voxel volume, we are compelled to ease the computational load by reducing the number of training cycles to 10 iteration steps, but then sampling 600 points in each step. We also slightly adapt our RF classifier to H3D's resolution and compute features for neighborhoods of $r \in \{0.125, 0.25, 0.5, 0.75, 1, 2, 3, 5\}$ m. Generally, results on H3D confirm our observations on V3D with final classification accuracies for *wE* + *DiFS* + *RIU* with an $\mathcal{O}_N$ oracle that are less than 3 pp worse compared to our classifier's optimal PL results and only require 0.12 ‰ (RF) and 0.08 ‰ (SCN) of available training points. We would like to emphasize that in such an ultra-high-resolution data set, due to spatial proximity of neighboring points, we always face a significant number of quasi-duplicates with respect to the representation of these points in feature space. This underlines the significance of *DiFS*, which is capable of improving OA values by $> 4$ pp and mF1 values by $> 5$ pp for both classifiers.

Since our two classifiers lead to similar accuracy levels for V3D and H3D, due to the aforementioned advantages in time complexity, we restrict ourselves to reporting solely RF-based AL

| Method | Sampl. Method | Oracle | F1-score | | | | | | | | | **mF1** | **OA** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Powerl. | L. Veg. | I. Surf. | Car | Fence | Roof | Façade | Shrub | Tree | | |
| **TP** | | | 61.99 | 88.83 | 91.22 | 66.72 | 40.66 | 93.61 | 42.62 | 55.87 | 82.57 | 69.34 | 85.24 |
| **RF** | | | | | | | | | | | | | |
| PL | | | 48.39 | **83.16** | **91.93** | 72.68 | 14.94 | **95.17** | **64.30** | 40.60 | **80.73** | 65.76 | **84.25** |
| AL | *wE* | $\mathcal{O}_O$ | 49.98 | 80.50 | 89.99 | 70.68 | 14.49 | 94.50 | 52.45 | 43.55 | 77.11 | 63.69 | 81.00 |
| | *wE+DiFS* | $\mathcal{O}_O$ | 61.90 | 80.53 | 90.24 | **73.12** | **28.58** | 94.14 | 57.08 | **43.55** | 78.99 | **67.57** | 82.43 |
| | *wE+DiFS+RIU* | $\mathcal{O}_O$ | 67.35 | 79.37 | 89.50 | 70.32 | 28.53 | 92.77 | 60.45 | 39.62 | 79.24 | 67.46 | 81.59 |
| | *wE+DiFS+RIU* | $\mathcal{O}_N$ | **68.85** | 79.44 | 90.16 | 69.43 | 27.44 | 92.64 | 58.06 | 36.66 | 77.00 | 66.63 | 81.17 |
| **SCN** | | | | | | | | | | | | | |
| PL | | | 42.11 | **81.40** | **91.11** | 72.15 | **41.22** | **94.10** | **59.65** | **48.87** | **83.88** | **72.92** | **83.86** |
| AL | *wE* | $\mathcal{O}_O$ | 65.17 | 78.29 | 88.96 | 68.86 | 25.32 | 88.39 | 49.58 | 34.49 | 76.81 | 63.99 | 79.07 |
| | *wE+DiFS* | $\mathcal{O}_O$ | 60.57 | 79.31 | 88.59 | 72.28 | 24.92 | 91.21 | 55.34 | 43.44 | 80.16 | 66.20 | 81.13 |
| | *wE+DiFS+RIU* | $\mathcal{O}_O$ | **63.02** | 79.52 | 89.62 | **75.03** | 26.33 | 91.18 | 54.41 | 38.45 | 78.27 | 66.20 | 80.91 |
| | *wE+DiFS+RIU* | $\mathcal{O}_N$ | 60.68 | 78.89 | 89.48 | 74.09 | 22.29 | 90.64 | 53.77 | 39.10 | 78.54 | 65.28 | 80.59 |

Table 1. Comparison of reachable accuracies [%] for different training approaches and oracles using RF and SCN for the V3D data set after 30 iteration steps. TP represents the result of the top-performing model of the benchmark challenge.

| Method | Sampl. Method | Oracle | F1-score | | | | | | | | | | | **mF1** | **OA** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | L. Veg. | I. Surf. | Car | U. Furn. | Roof | Façade | Shrub | Tree | Gravel | Vert. Surf. | Chim. | | |
| **TP** | | | 92.90 | 90.23 | 78.51 | 57.89 | 95.71 | 80.43 | 68.46 | 97.21 | 62.37 | 73.08 | 72.45 | 79.02 | 89.75 |
| **RF** | | | | | | | | | | | | | | | |
| PL | | | 89.97 | **88.17** | **63.76** | **49.18** | 95.59 | **78.08** | **65.86** | 95.36 | 47.34 | **59.63** | 80.52 | **73.95** | **86.87** |
| AL | *wE* | $\mathcal{O}_O$ | 87.04 | 79.33 | 49.48 | 42.15 | 93.17 | 74.72 | 63.22 | 95.12 | 46.65 | 27.40 | **85.50** | 67.62 | 81.63 |
| | *wE+DiFS* | $\mathcal{O}_O$ | **91.04** | 85.93 | 59.74 | 43.64 | **95.92** | 76.40 | 64.41 | **95.68** | **51.34** | 54.80 | 82.97 | 72.90 | 86.58 |
| | *wE+DiFS+RIU* | $\mathcal{O}_O$ | 88.38 | 85.97 | 55.68 | 44.07 | 93.75 | 75.64 | 66.46 | 95.56 | 49.69 | 55.53 | 63.59 | 70.39 | 84.84 |
| | *wE+DiFS+RIU* | $\mathcal{O}_N$ | 88.06 | 86.94 | 56.01 | 42.88 | 93.93 | 75.78 | 64.43 | 95.14 | 46.67 | 56.17 | 50.26 | 68.75 | 84.82 |
| **SCN** | | | | | | | | | | | | | | | |
| PL | | | **90.69** | **87.82** | 55.17 | **52.52** | **96.74** | **81.61** | **63.25** | **96.60** | 50.55 | **70.97** | 63.24 | **73.56** | **87.40** |
| AL | *wE* | $\mathcal{O}_O$ | 84.91 | 79.04 | 51.37 | 38.98 | 92.45 | 75.10 | 51.51 | 92.01 | 43.77 | 60.90 | 63.65 | 66.70 | 80.25 |
| | *wE+DiFS* | $\mathcal{O}_O$ | 88.28 | 82.06 | 68.27 | 40.25 | 95.01 | 77.68 | 56.81 | 95.66 | 49.91 | 70.09 | **74.64** | 72.61 | 84.35 |
| | *wE+DiFS+RIU* | $\mathcal{O}_O$ | 89.58 | 85.45 | **68.36** | 45.50 | 95.55 | 75.78 | 49.87 | 95.76 | 54.18 | 70.87 | 48.96 | 70.90 | 85.44 |
| | *wE+DiFS+RIU* | $\mathcal{O}_N$ | 89.29 | 83.03 | 63.64 | 39.06 | 94.78 | 73.93 | 51.50 | 95.24 | **54.59** | 67.10 | 54.31 | 69.68 | 84.43 |

Table 2. Comparison of reachable accuracies [%] for different training approaches and oracles using RF and SCN for the H3D data set after 30 iteration steps (RF) and 10 iteration steps (SCN), respectively. Furthermore, we report the result of the (at the time of writing this paper) top-performing TP model of the still ongoing benchmark challenge.

| Method | Sampl. Method | Oracle | F1-score | | | | **mF1** | **OA** |
|---|---|---|---|---|---|---|---|---|
| | | | U. Furn. | Ground | Building | Tree | | |
| PL | | | **75.30** | **98.63** | **96.82** | **93.97** | **91.18** | **95.51** |
| AL | *wE* | $\mathcal{O}_O$ | 67.70 | 98.19 | 96.12 | 93.31 | 88.83 | 94.63 |
| | *wE+DiFS* | $\mathcal{O}_O$ | 66.25 | 98.29 | 96.03 | 93.40 | 88.49 | 94.65 |
| | *wE+DiFS+RIU* | $\mathcal{O}_O$ | 62.19 | 97.87 | 94.81 | 91.90 | 86.69 | 93.47 |
| | *wE+DiFS+RIU* | $\mathcal{O}_N$ | 59.86 | 97.82 | 93.89 | 91.51 | 85.77 | 92.83 |

Table 3. Comparison of reachable accuracies [%] for different training approaches and oracles using RF for the S3D data set after 30 iteration steps.

runs for the large-scale S3D data set. As this data sets depicts a significantly larger scene with a plethora of representatives for each class, we are dealing with a much greater intra-class variety, which is further amplified by generalization through the rather coarse class catalog. Thus, the highest accuracies are achieved for S3D in the PL run. Especially class *Urban Furniture* suffers when learning from only limited training sets, as those fail to truthfully characterize the large variety of this quasi-class *Other*. Nevertheless, with the optimal configuration from the machine's point of view ($wE + DiFS$), we obtain a result that is less than 1 pp worse in OA than in PL, but only utilizing 0.23 ‰ of available training points (please note that the effect of boosting convergence by *DiFS* is not visible at this saturated state of the iteration after 30 iteration steps, but improves OA by $> 2$ pp at iteration step 10, for instance).

## 5. CONCLUSION

This work represents a first attempt to benchmark AL in the domain of ALS point cloud classification and underlines its great potential to minimize labeling effort and thus make ML methods broadly applicable. Although the accuracy of AL approaches is slightly worse compared to corresponding PL approaches, models can be flexibly set up for a given (new) data set with minimal labeling overhead, which is an important property in times of rapid data acquisition cycles. More significantly, our AL-based results emphasize that the long-held understanding of ML models requiring vast annotated data sets is not the key to success, but rather building a versatile (small) training set with most-informative (actively queried) samples. In this spirit, the geospatial community can benefit from the recommendation of Ng (2021) to focus more on the data-centric branch of ML research to really enable its true capabilities. This is especially the case, as the community lacks annotated data

sets at a level comparable to that of the computer vision community.

## ACKNOWLEDGEMENT

We thank the State Office for Spatial Information and Land Development Baden-Wuerttemberg for providing the ALS point cloud and orthophoto data to constitute S3D. This work profited from funding by the German Research Foundation as part of Germany´s Excellence Strategy – EXC 2120/1 – 390831618.

## References

Bloodgood, M., Vijay-Shanker, K., 2009. A method for stopping active learning based on stabilizing predictions and the need for user-adjustable stopping. *Proceedings of the Thirteenth Conference on Computational Natural Language Learning*, 39–47.

Breiman, L., 2001. Random Forests. *Machine Learning*, 45(1), 5–32.

Haala, N., Kölle, M., Cramer, M., Laupheimer, D., Mandlburger, G., Glira, P., 2020. Hybrid Georeferencing, Enhancement and Classification of Ultra-High Resolution UAV LIDAR and Image Point Clouds for Monitoring Applications. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, V-2-2020, 727–734.

Ho, T. K., Baird, H., 1997. Large-scale simulation studies in image pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(10), 1067–1079.

Hui, Z., Jin, S., Cheng, P., Ziggah, Y. Y., Wang, L., Wang, Y., Hu, H., Hu, Y., 2019. An Active Learning Method for DEM Extraction From Airborne LiDAR Point Clouds. *IEEE Access*, 7, 89366-89378.

Jospin, L. V., Laga, H., Boussaid, F., Buntine, W., Bennamoun, M., 2022. Hands-On Bayesian Neural Networks—A Tutorial for Deep Learning Users. *IEEE Computational Intelligence Magazine*, 17(2), 29–48.

Kellenberger, B., Marcos, D., Lobry, S., Tuia, D., 2019. Half a Percent of Labels is Enough: Efficient Animal Detection in UAV Imagery Using Deep CNNs and Active Learning. *IEEE Transactions on Geoscience and Remote Sensing*, 57(12), 9524–9533.

Kölle, M., Laupheimer, D., Schmohl, S., Haala, N., Rottensteiner, F., Wegner, J. D., Ledoux, H., 2021a. The Hessigheim 3D (H3D) benchmark on semantic segmentation of high-resolution 3D point clouds and textured meshes from UAV LiDAR and Multi-View-Stereo. *ISPRS Open Journal of Photogrammetry and Remote Sensing*, 1, 100001.

Kölle, M., Walter, V., Schmohl, S., Soergel, U., 2020. Hybrid Acquisition of High Quality Training Data for Semantic Segmentation of 3D Point Clouds Using Crowd-Based Active Learning. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, V-2-2020, 501–508.

Kölle, M., Walter, V., Schmohl, S., Soergel, U., 2021b. Remembering both the machine and the crowd when sampling points: Active learning for semantic segmentation of ALS point clouds. *ICPR International Workshops and Challenges 2021*, 505–520.

Li, N., Pfeifer, N., 2019. Active Learning to Extend Training Data for Large Area Airborne LiDAR Classification. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W13, 1033-1037.

Lin, Y., Vosselman, G., Cao, Y., Yang, M. Y., 2020a. Active and incremental learning for semantic ALS point cloud segmentation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 169, 73-92.

Lin, Y., Vosselman, G., Cao, Y., Yang, M. Y., 2020b. Efficient Training of Semantic Point Cloud Segmentation via Active Learning. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, V-2-2020, 243–250.

Lockhart, J., Assefa, S., Balch, T., Veloso, M., 2020. Some people aren't worth listening to: periodically retraining classifiers with feedback from a team of end users. *CoRR*, abs/2004.13152.

Luo, H., Wang, C., Wen, C., Chen, Z., Zai, D., Yu, Y., Li, J., 2018. Semantic Labeling of Mobile LiDAR Point Clouds via Active Learning and Higher Order MRF. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 56(7), 3631-3644.

Mackowiak, R., Lenz, P., Ghori, O., Diego, F., Lange, O., Rother, C., 2018. CEREALS - Cost-Effective REgion-based Active Learning for Semantic Segmentation. *British Machine Vision Conference 2018*.

Ng, A., 2021. The batch - weekly issue 84 [WWW Document]. URL: https://www.deeplearning.ai/the-batch/issue-84/ (accessed October 18, 2022).

Niemeyer, J., Rottensteiner, F., Soergel, U., 2014. Contextual classification of lidar data and building object detection in urban areas. *ISPRS Journal of Photogrammetry and Remote Sensing*, 87, 152–165.

Qi, C. R., Su, H., Kaichun, M., Guibas, L. J., 2017. PointNet: Deep learning on point sets for 3D classification and segmentation. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 77–85.

Schmohl, S., Sörgel, U., 2019. Submanifold Sparse Convolutional Networks For Semantic Segmentation of Large-Scale ALS Point Clouds. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2/W5, 77–84.

Settles, B., 2009. Active Learning Literature Survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison.

Shao, F., Luo, Y., Liu, P., Chen, J., Yang, Y., Lu, Y., Xiao, J., 2022. Active Learning for Point Cloud Semantic Segmentation via Spatial-Structural Diversity Reasoning. *CoRR*, abs/2202.12588.

Shi, X., Xu, X., Chen, K., Cai, L., Foo, C. S., Jia, K., 2021. Label-Efficient Point Cloud Semantic Segmentation: An Active Learning Approach. *CoRR*, abs/2101.06931.

Stork, D., 1999. Character and document research in the open mind initiative. *Proceedings of the Fifth International Conference on Document Analysis and Recognition.*, 1–12.

van der Maaten, L., Hinton, G., 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9, 2579–2605.

Vaughan, J. W., 2018. Making Better Use of the Crowd: How Crowdsourcing Can Advance Machine Learning Research. *Journal of Machine Learning Research*, 18(193), 1–46.

Waldhauser, C., Hochreiter, R., Otepka, J., Pfeifer, N., Ghuffar, S., Korzeniowska, K., Wagner, G., 2014. Automated classification of airborne laser scanning point clouds. *Solving Computationally Expensive Engineering Problems*, Springer International Publishing, 269–292.

Wu, T.-H., Liu, Y.-C., Huang, Y.-K., Lee, H.-Y., Su, H.-T., Huang, P.-C., Hsu, W. H., 2021. Redal: Region-based and diversity-aware active learning for point cloud semantic segmentation. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 15510–15519.

Zhdanov, F., 2019. Diverse mini-batch Active Learning. *CoRR*, abs/1901.05954. http://arxiv.org/abs/1901.05954.