Remembering Both the Machine and the Crowd When Sampling Points: Active Learning for Semantic Segmentation of ALS Point Clouds

Michael Kölle(✉), Volker Walter, Stefan Schmohl, and Uwe Soergel

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

Abstract. Supervised Machine Learning systems such as Convolutional Neural Networks (CNNs) are known for their great need for labeled data. However, in case of geospatial data and especially in terms of Airborne Laserscanning (ALS) point clouds, labeled data is rather scarce, hindering the application of such systems. Therefore, we rely on Active Learning (AL) for significantly reducing necessary labels and we aim at gaining a deeper understanding on its working principle for ALS point clouds. Since the key element of AL is sampling of most informative points, we compare different basic sampling strategies and try to further improve them for geospatial data. While AL reduces total labeling effort, the basic issue of experts doing this labor- and therefore cost-intensive task remains. Therefore, we propose to outsource data annotation to the crowd. However, when employing crowdworkers, labeling errors are inevitable. As a remedy, we aim on selecting points, which are easier for interpretation and evaluate the robustness of AL to labeling errors. Applying these strategies for different classifiers, we estimate realistic segmentation results from crowdsourced data solely, only differing in Overall Accuracy by about 3% points compared to results based on completely labeled dataset, which is demonstrated for two different scenes.

Keywords: Active Learning · Crowdsourcing · 3D point clouds · Classification · Labeling · Random Forest · Sparse 3D CNN

1 Introduction

A paramount requirement of supervised Machine Learning (ML) systems is labeled training data. Especially, since the renaissance of neural networks in the form of Convolutional Neural Networks (CNNs) there is an increasing demand for large pools of high-quality training data. In this context, huge effort was put in establishing massive annotated data corpora such as *ImageNet* [7] and *Cifar-10* & *Cifar-100* [18]. However, in the context of geospatial data such labeled

A. Del Bimbo et al. (Eds.): ICPR 2020 Workshops, LNCS 12667, pp. 505–520, 2021.
https://doi.org/10.1007/978-3-030-68787-8_37

datasets are rather scarce, which especially applies to 3D point clouds. One publicly available dataset is the *ISPRS Vaihingen 3D Semantic Labeling* benchmark (V3D) [25], which was manually annotated by experts. This annotation process is a highly labor-intensive and therefore costly task.

One method for significantly reducing the necessity of labeled training samples provided by human annotators is Active Learning (AL). The major goal of AL is to maintain the performance of a ML system, while only focusing on a subset of instances from a training pool inhering most information [28]. First AL approaches focused on Support Vector Machines (SVMs) [6], which are well suited for such approaches by design. Ertekin et al. [8] exploited the idea of SVMs of only focusing on points close to the decision boundary by sampling points to be labeled in the vicinity of already learned SVM hyperplanes. More general methods for detecting most informative points focus on the predicted a posteriori probability of a classifier making them more independent of the ML model used. A comprehensive overview of these methods of the pre-Deep Learning era is given by Settles [28]. When using CNNs, Gal and Ghahramani [10] recommend to form Monte Carlo dropout ensembles in order to overcome overestimation of a posteriori probabilities often observed in case of CNNs.

Regardless of the classifier used, AL selection criteria are typically designed for requesting the label of one specific data point per iteration step based on some informativeness measure [28]. However, retraining a classifier each time one individual point is added to the training pool is computationally expensive and will only marginally improve its performance, especially when employing CNNs. Because of this, most commonly batch-mode AL is preferred [15,24]. On the other hand, when adding multiple instances to the training pool based on one classification process, it is very likely that all sampled points are similar in terms of their representation in feature space. In order to increase the diversity of selected samples to boost convergence of the AL process multiple methods have been proposed [16,35].

While using AL for predicting land cover maps from hyperspectral imagery was studied extensively [26,29], only few investigations were conducted on applying AL for the semantic segmentation of Airborne Laserscanning (ALS) point clouds. Hui et al. [14] use a fully automated AL approach for filtering ground points to derive digital terrain models, focusing on a binary segmentation. Li and Pfeifer [19] rely on AL for predicting multiple land cover classes from ALS data by automatic propagation of labels from an initial training dataset without including human annotators. Luo et al. [23] present an approach for semantic segmentation of Mobile Mapping point clouds employing a voxel-based higher order Markov Random Field. Closest related to our method are the findings of Lin et al. [20], who employ the *PointNet++* architecture to ALS point clouds. The authors realize a tile-based approach, where in each iteration step most informative tiles are queried, fully labeled and added to the training pool.

All previously discussed works describe efficient means to reduce the total amount of necessary labels, but these labels are typically still provided by an expert. Our goal is not only to reduce effort of experts but to completely shift

and outsource labeling effort to non-experts, namely to the crowd. It is already proven that crowdsourcing is well suited for annotating geospatial data [31,32]. This enables running a fully automated human-in-the-loop [3] pipeline exploiting capabilities of the online crowdsourcing platform *Microworkers* [13] as described by Kölle et al. [17]. Such hybrid intelligence systems were also discussed by Vaughan [30] for combining the individual strengths of both parties.

While in Kölle et al. [17] we mainly concentrated on the performance of the human operator given by the crowd, the emphasis of this work lies on the role of the machine. Therefore, our contributions can be summarized as follows: i) We focus on a deeper understanding of applying AL to geospatial data represented by ALS point clouds. ii) This includes a detailed comparison of different selection strategies provided in literature, which we enhance by different methods for faster convergence. These strategies are applied for both a feature-driven Random Forest (RF) [4] and a data-driven CNN approach. iii) While in literature usually receiving true labels from an oracle is assumed, this hardly holds true for actual labeling of data by experts and is especially unrealistic in case of paid crowdworkers, where labeling errors are inevitable. We therefore test the robustness of our approach, address it using a special sampling strategy and estimate results, which are realistic for the crowd to reach.

2 Methodology

In typical ML scenarios Passive Learning (PL) is applied, where a previously labeled data pool is used for training. In contrast to this, in AL a model is actively involved in establishing such a training dataset. Precisely, after an initial training step the classifier points out instances, which carry most information and are therefore a reasonable addition to the training dataset justifying annotation effort by a human operator. Thus, the inherent hypothesis is that only a small subset of the dataset is required for sufficiently training a classifier.

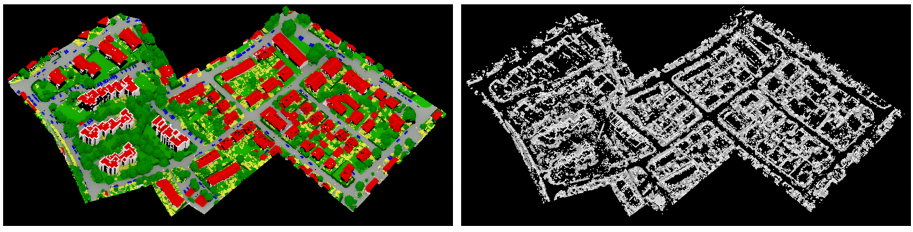

Fig. 1. Visualization of Support Vectors of V3D training dataset (*right*) compared to reference labeling (*left*). (Color figure online)

This is also the idea underlying the SVM. When training such a model, Support Vectors are determined, which define class-separating hyperplanes in feature space. Only these Support Vectors are afterwards used in inference, which means

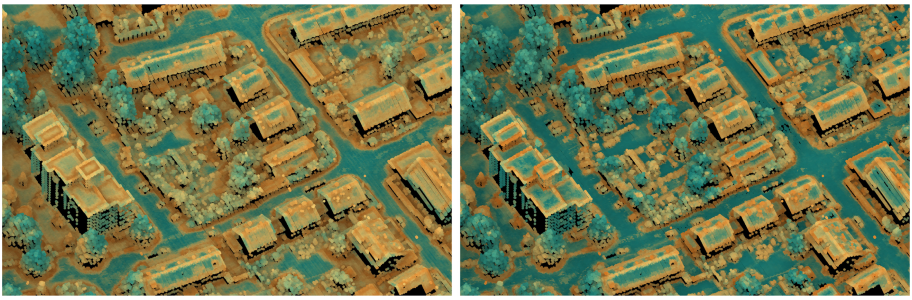

Fig. 2. Derived pointwise scores of the first iteration step (*left*) and the last iteration step (*right*). Color scale ranges from *dark blue* = low sampling priority to *orange* = high sampling priority. Points missing on the right were selected and added to T. (Color figure online)

that only those instances impact the performance of the trained model. When further pursuing the concept of SVM, we assume that most informative samples are always located in close proximity to the decision boundary. Instances located here are most demanding to classify since they incorporate features of two or more different classes. In context of ALS point clouds, such points naturally are also situated on class borders in object space, so that delineation lines of individual class occurrences can clearly be observed and compared to the reference labeling in Fig. 1. Only 21.68% of provided training points were considered Support Vectors supporting the hypothesis that only a fraction of points needs to be labeled. When used within PL, the SVM utilizes all instances in proximity of the separating hyperplanes until exhaustion of such points.

However, in AL typically significantly fewer labels are required [15,19,24]. Precisely, only a subset of these points closest to the decision boundary is constituted iteratively. In every iteration step only a limited number of points that currently represent most uncertainty and therefore most information (see Sect. 2.2) is drawn from an unlabeled training pool U, labeled by a so-called oracle $\mathcal{O}$ and added to the training pool T. After retraining a classifier C based on expanded T, C becomes much more certain when predicting on points of the remaining training data set $R = U \setminus T$, which are similar to those recently added. Therefore, sampling quasi-duplicates can be limited. Vice versa for our experiments presented in Sect. 3 up to 81.21% of instances selected within the AL procedure are actually Support Vectors. Selection of Non-Support Vectors mainly happens in early iteration steps where easy to interpret points are queried, which are however not included in T so far. The behavior of the AL process can also be traced in Fig. 2, where the model's uncertainty (measured by Eq. 1) of the initial iteration step is compared to that of the last step (30 iteration steps have been conducted). Both results underline that points in close proximity to class boundaries in object space are most complex for automatic interpretation, which persists throughout the complete iteration process. We can also observe that in total the model becomes more confident in its decisions (e.g. points on

roofs and vegetation becoming *dark blue*) and the uncertainty margin shrinks to an extent of close to class boundaries.

Based on this conceptional foundation, we now address the main components of AL: i) the employed classifier (Sect. 2.1), ii) the sampling strategy for detecting most informative instances (Sect. 2.2) and iii) the employed oracle (Sect. 2.3).

2.1 Employed Classifiers

For enabling a feature-based RF classifier, a selection of handcrafted geometric and radiometric features is taken from literature [2,5,33] and used within this work (detailed description of features can be found in Haala et al. [12]). All features are computed for each point considering spherical point neighborhoods of 1, 2, 3 and 5 m, so that a multi-scale approach is realized. Employing the RF classifier within the AL scenario is straightforward for its pointwise functionality. We can simply transfer selected points from U to T and use points included in T as individual instances since point neighborhoods were already sufficiently taken into account in the preprocessing, i.e. feature computation. This is a fundamental difference to employing a CNN approach, which we oppose to the RF classifier.

In contrast to applying *PointNet++* as Lin et al. [20], we employ the voxel-based Sparse Convolutional Neural Network (SCNN) [11], transferred for usage on ALS point clouds by Schmohl and Sörgel [27]. Compared to this work, we train slightly shallower networks (4 *U-Net* levels), which are more stable when trained on such few labeled points. The obsolescent need of handcrafted features in Deep Learning is not necessarily advantageous in case of AL, since in every iteration step features need to be relearned or at least refined based on the newly added training points. We therefore also have to include their (non-labeled) surrounding points as input to the network for spatial context. Such points do not directly contribute to the training loss, but assist feature learning/refinement due to their passive presence. However, this is computationally more complex than computing features only one time in advance of the AL loop as for the RF. To reduce training effort, we initialize the network weights and learning rate in each AL iteration step by adopting respective values from the previous one, yielding faster convergence. For each step, we establish an ensemble of 5 differently weight-initialized models.

For dynamic adaption of learning rate and early stopping of the training procedure, a validation dataset is required. In case of AL it is not reasonable to exclude a pre-defined area of the training dataset for this, since the spatial distribution of labeled points in the training set is not known before. Therefore, in each iteration step we randomly pick 20% of points of each class from T and use it to validate our model. Consequently, our validation dataset is more related to the training dataset than in PL, but consists only of most informative points, which are more demanding for classification than conventional validation datasets mitigating this issue.

2.2 Selection Strategies

When applying the trivial strategy of sampling points by randomly picking, it is to be expected that a mixture of both most and low informative points will be selected causing prolonged convergence time of the iteration process. Furthermore, random sampling lacks applicability for highly inhomogeneous class distributions, which are common for ALS point clouds. More directed strategies aim at detecting points where the intrinsic confidence of the model is minimum based on the a posteriori probability $p(c|x)$ that point x belongs to class c. Since strategies such as Least Certainty Sampling and Breaking Ties [28] only consider a fraction of predictive information (provided that multi-class problem is to be solved), we decide to rely on Entropy (E). Points having greatest E are considered to be informative, since E is maximum for an equal distribution of a posteriori probabilities and minimum for one class having a $p(c|x)$ of 1:

$$x_E = \underset{x}{\operatorname{argmax}} - \sum_c p(c|x) \cdot \log p(c|x) \tag{1}$$

The aforementioned measures can be summarized as *Query-by-Uncertainty* [28]. When applying an ensemble classifier (e.g. RF), uncertainty can additionally be measured as disagreement between different models pursuing the idea of *Query-by-Committee*. This can be achieved by Vote Entropy (VE) [1], where we assume to have e ensemble members each predicting a posteriori probabilities for each class placed in $\mathbf{P}_e$. Each member is allowed to vote for one class (the one having highest $p(c|x)$). These votings are then evaluated for each class establishing a new distribution, which is normalized by the number of ensemble members N_e and evaluated using the entropy formula:

$$\begin{aligned} x_{VE} &= \underset{x}{\operatorname{argmax}} - \sum_c \frac{\sum_e D(\mathbf{P}_e, c)}{N_e} \cdot \log \frac{\sum_e D(\mathbf{P}_e, c)}{N_e} \\ \text{where } D(\mathbf{P}_e, c) &= \begin{cases} 1, & \text{if } \operatorname{argmax}(\mathbf{P}_e) = c \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{2}$$

The rationale of VE is that the class of one individual instance can be predicted with high confidence as long as most ensemble members vote for this class even if the maximum a posteriori probability is rather low.

For both VE and E, we can easily introduce a sampling method yielding a more equal distribution of classes in the created training dataset, which can be accomplished by individual class weighting. Precisely, these weights are calculated dynamically as ratio of the total number of points N_T currently present in T and the number of representatives of each class N_c at iteration step t ($w_c(t) = N_T(t)/N_c(t)$). These weights are then multiplied by the individual score of the respective class (E: $p(c|x)$, VE: $\sum_e D(\mathbf{P}_e, c)/N_e$) before inserting into the entropy formula and referred to as wE and wVE respectively.

For efficiency reasons, we aim at selecting and adding multiple points per iteration step to our training dataset according to pool-based AL. Since similar

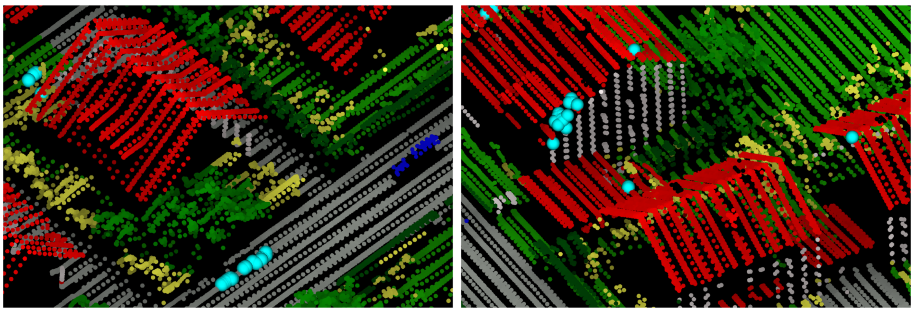

Fig. 3. Subsets of via E sampled points (*cyan*) from two exemplary iteration steps visualized in the training point cloud colorized according to reference data. (Color figure online)

points in feature space yield similar uncertainty scores, sampling quasi-duplicates inhering same information is likely, when only considering this score. In case of ALS point clouds such points typically appear as clusters in object space (see Fig. 3), which is why increasing diversity is related to increasing the distance between sampled points. Consequently, we consider the scores gained by any selection strategy as priority list for creating Diversity in Object Space (*DiOS*). Based on the order in this list, points are transferred from R to T if the distance to all points previously selected within this iteration step is greater than d_{DiOS}. While such methods are commonly realized in feature space [16,34], this procedure directly works in object space, which is of course mainly applicable for geospatial data where an interpretable object space is present.

As a second method we resort to Diversity in Feature Space (*DiFS*) according to Zhdanov [35]. For this we aim at detecting clusters of similar points with regard to their representation in feature space. For focusing on most informative points, we additionally use the score of each instance derived by any of the aforementioned selection strategies as individual weight and combine both measures by running a weighted k-means clustering [21]. Afterwards, from every cluster formed, we sample the same number of instances with the highest scores. In order to reduce computational effort, for this procedure we only consider n_{DiFS} points having highest selection scores since we can assume that points yielding low scores will not improve our model.

Considering our ultimate goal of crowdworkers labeling selected points, we assume that increasing distance to the class boundary is helpful for a better and unambiguous interpretability and helps avoiding weariness of crowdworkers resulting in less labeling errors. As already seen in Fig. 2, in case of geospatial data analysis spatial distance to class boundary is closely related to distance to decision boundary. Therefore, we identify informative points by any of the aforementioned measures and consider neighboring points for labeling instead. Precisely, for Reducing Interpretation Uncertainty (*RIU*) we use a spherical neighborhood of radius d_{RIU} centered in a selected point (seed point) and search within this neighborhood for the lowest score. This point is then presented for labeling instead of the original seed point. This procedure is exemplary visualized

for different values of d_{RIU} (i.e. max. distance from the seed point) in Fig. 4 and demonstrates that distance to the class boundary can be efficiently increased.

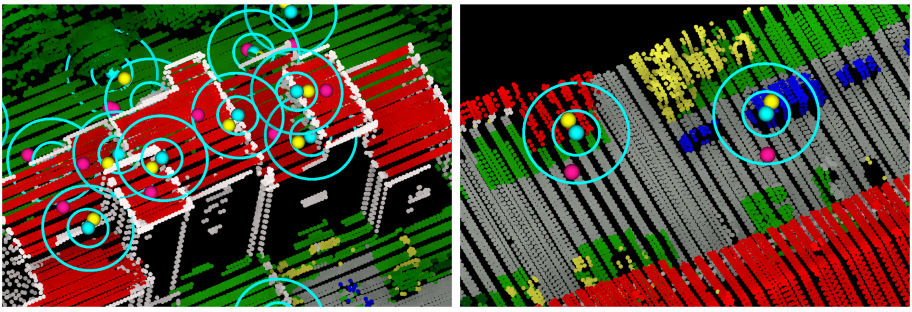

Fig. 4. Increasing distance to decision boundary. Instead of the seed point (*cyan*), we select a point further away from the class border, allowing maximum 3D radii d_{RIU} of 1.5 m (*yellow*) and 4 m (*pink*) indicated by respective circles. (Color figure online)

2.3 Employed Oracle

In the context of the proposed human-in-the-loop pipeline, previous sections focused on the role of the machine for querying most informative points, whereby respective labels are intended to be received from a human annotator. However, in most studies this operator is replaced by an omniscient oracle $\mathcal{O}_O$, which always labels correctly according to the reference data. Especially for paid crowdsourcing relying on non-experts this assumption is not justified [32]. Lockhart et al. [22] differentiate between two types of erroneous oracles, namely noisy and confused oracles ($\mathcal{O}_N$ and $\mathcal{O}_C$). The noisy oracle behavior $\mathcal{O}_N$ applies both to a human annotator, who has a well understanding of the task but randomly misclassifies some points, and to a crowdworker, who is not paying attention at all and, often observed in crowdsourcing [9], just picks classes randomly. A confused oracle on the other hand misclassifies points by always confusing the same classes (according to some distinct mapping), for instance *Fence/Hedge* vs. *Shrub* or *Roof* vs. *Façade*. This problem occurs especially in AL where focus lies on most informative points, which are situated on or near to class boundaries.

2.4 Datasets

We test our method on two different datasets of individual characteristics. A suburban scene featuring single family houses and building blocks is represented by the V3D dataset [25] (visualized in Fig. 1). This point cloud captured in August 2008 incorporates a total of 9 classes (see Table 2). The point density is about $4-8$ pts/m^2. In order to also derive color features the point cloud is colorized by orthogonal projection of corresponding CIR images. As second dataset we rely on an UAV LiDAR point cloud colorized by simultaneously acquired imagery and

captured in March 2018 using the same flight mission parameters as in Haala et al. [12], henceforth referred to as Hessigheim 3D (H3D)[1]. The point density is about 800 pts/m^2, but for efficiency reasons, spatial subsampling to a minimum point distance of 0.3 m was applied. The point cloud representing a rural village was manually annotated by the authors using a fine-grained class scheme consisting of 12 classes (see Table 3). For both datasets the initial training set is provided by the crowd as outlined in Kölle et al. [17].

3 Results

3.1 Comparison of Selection Strategies

For evaluating, which strategy for selecting most informative points works best, we apply those presented in Sect. 2.2 on the V3D dataset in combination with the RF classifier using 30 iteration steps and a batch size of 300. We rely on an ensemble of 100 binary decision trees having a maximum depth of 18. The performance throughout the iteration loop is depicted in Fig. 5 (*left*). We want to stress that all our results are obtained after only labeling a small fraction of 1.15% from U. Accuracies within this work are evaluated for a distinct test dataset disjoint to the respective training dataset (i.e. samples are only drawn from U).

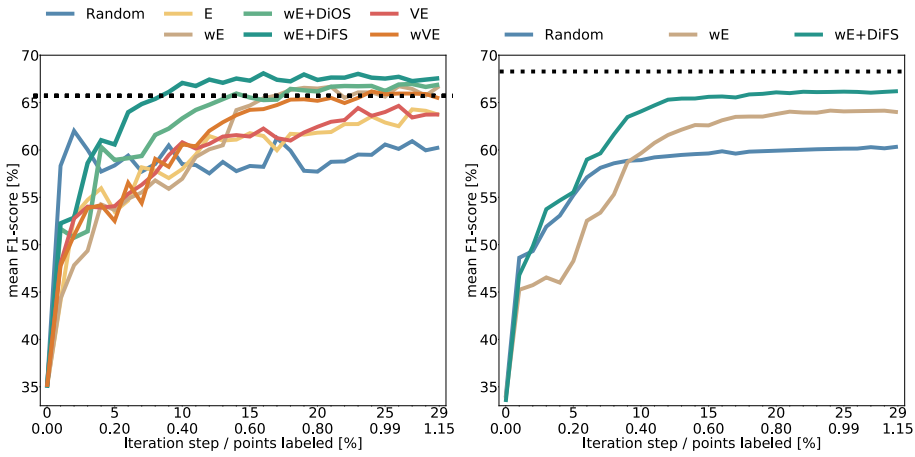

Fig. 5. Comparison of different selection strategies in combination with our RF (*left*) and our SCNN (*right*) classifier applied to the V3D dataset and evaluated according to F1-scores. For reference, the dotted black line depicts the mean F1-score for PL using the completely labeled dataset. (Color figure online)

Although the performance of random sampling rises steeply at first, it soon settles at a mean F1-score of about 60%, since less frequent classes are not

[1] Dataset will be made publicly available in early 2021.

selected sufficiently. Later it is outperformed by sampling strategies such as E or VE. These in turn are exceeded by the enhanced strategies by applying individual class weights in each iteration step (*wE* and *wVE*), since for every class a sufficient almost equally distributed number of labeled points is obtained. Nevertheless, one disadvantage of these weighting strategies is the resulting comparatively slow (but steady) increase in performance. Motivated by the strong performance gain of random sampling in early iteration steps due to selecting a greater bandwidth of points, we apply both *DiOS* using an empirically determined value for d_{DiOS} of 5 m and *DiFS*, where we set n_{DiFS} to 10.000 and form 300 clusters (see also Sect. 2.2). We analyze the effect of these two strategies for *wE*, which has proven to be an efficient sampling strategy regarding the reachable accuracy in the later course of the iteration. Figure 5 (*left*) outlines that both strategies of increasing diversity positively impact the performance of the AL loop.

We want to stress that increasing diversity especially boosts the convergence of the AL loop, which means that less iteration steps are necessary for reaching the same performance of the trained model as if more iteration steps are conducted. For instance, applying $wE + DiFS$ achieves convergence after only 10 iteration steps. At this time, basic *wE* reaches a mean F1-score of about 10% points less. For reaching an accuracy similar to $wE + DiFS$, *wE* requires 10 iteration steps more and therefore additional labeling of 3000 points (10 iteration steps and batch size of 300). Relative to *DiFS*, the *DiOS* strategy performs slightly worse especially in the course of the first few iteration steps, but still outperforms the baseline of pure *wE*.

3.2 Comparison of Employed Classifiers

For comparing our SCNN classifier to the RF we focus on the selection strategies that have proven to be most effective (*wE*, $wE + DiFS$), visualized in Fig. 5 (*right*), which is to be interpreted relative to Fig. 5 (*left*). Regarding these two strategies, for both classifiers roughly the same number of iteration steps is necessary for convergence. The performance of SCNN increases more steadily and especially high-frequency oscillations do not occur because in contrast to the RF, each model is only retrained in each iteration step and not trained from scratch again. Although for our best strategy ($wE + DiFS$) both classifiers reach a similar accuracy, that of SCNN rises not as fast as for the RF (after 10 iteration steps mean F1-score for RF: 67% vs. SCNN: 64%). Furthermore, SCNN fails to exceed the accuracy of PL on the completely labeled dataset, which might be due to overfitting regarding the sparsely labeled training dataset. Nevertheless, the difference in Overall Accuracy (OA) between PL and AL is less than 3% points (see Table 2).

3.3 Comparison of Different Oracle Types

All aforementioned results assume an oracle behaving like $\mathcal{O}_O$, which can hardly be observed when working with real crowdworkers. For the more justified assumption of a noisy or a confused oracle we simulate 10%, 30%, 50%

Table 1. Behavior of our confused oracle regarding the V3D dataset.

True label	Powerl	L. Veg	I. Surf	Car	Fence	Roof	Façade	Shrub	Tree
Confused with	Roof	Fence	Façade	I. Surf	Shrub	Façade	Roof	Tree	Shrub

and 100% erroneous labels received in both cases. For the noisy oracle $\mathcal{O}_N$ we randomly use any label (excluding the true one). Regarding the systematically confused mapping of $\mathcal{O}_C$, we apply most observed confusions when employing real crowdworkers as presented in Kölle et al. [17], which are summarized in Table 1.

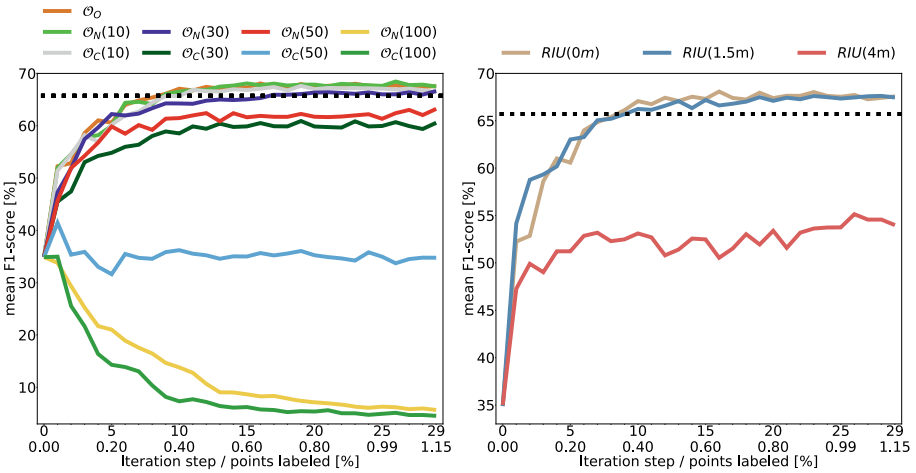

Fig. 6. Comparison of different AL-scenarios when relying on real crowdworkers for the V3D dataset using RF and $wE + DiFS$ (black line represents PL): simulated crowd errors (*left*) and impact of increasing distance to the class border via *RIU* (*right*). (Color figure online)

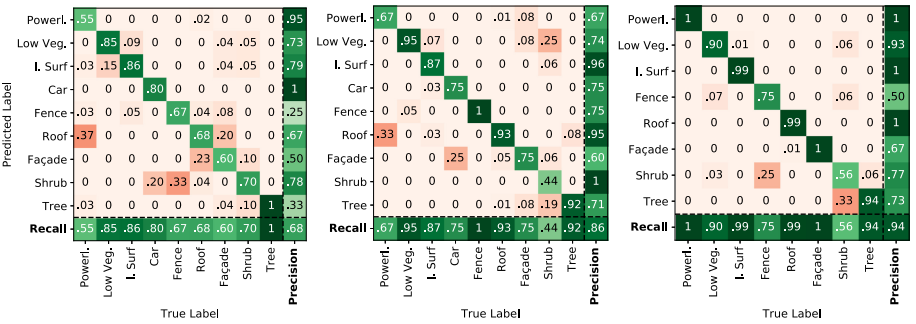

Fig. 7. Comparison of reachable accuracies (normalized confusion matrices) of the crowd when using different values for d_{RIU}. From left to right: $d_{RIU} = 0$ m/1.5 m and 4 m. Accuracies are aggregated via majority vote from 3 acquisitions per point.

This simulation is based on the RF classifier using $wE + DiFS$ with a batch size of 300 for sampling. As visualized in Fig. 6 (*left*), $\mathcal{O}_O$ leads as expected to one of the best performances of the AL loop together with $\mathcal{O}_N(10\%)$ and $\mathcal{O}_C(10\%)$ demonstrating the robustness of our pipeline to a moderate number of labeling errors. All other oracle behaviors naturally diminish accuracies to some extent. Generally it is observable that the confused oracle is especially harmful to the AL loop since systematic false labeling (e.g. according to Table 1) is induced. For instance, the RF performs significantly better when the oracle labels 50% of points randomly false compared to when only 30% of points are labeled systematically false. Other mapping functions of malicious crowdworkers (for instance, labeling all points same or according to any absurd function) are not considered, since such workers can be easily identified using control tasks.

Since our proposed method for minimizing confused labeling (RIU) is only reasonable when the performance of the AL loop can be uphold, we simulate respective AL runs in Fig. 6 (*right*). While increasing the distance to the class border by maximum 1.5 m has no significant impact on the performance of the AL iteration, $d_{RIU} = 4$ m causes the mean F1-score to drop significantly. This is due to selecting less informative samples (i.e. points further away from the class boundary) or that points belonging to a different class than the seed point are selected (see Fig. 4 (*right*) where with $d_{RIU} = 4$ m a street point is selected instead of a car point).

In order to evaluate whether this method helps crowdworkers labeling points, we conducted three crowdsourcing campaigns using the same parameters as for the simulation in Fig. 4 (*right*) and varied $d_{RIU} = 0$ m/1.5 m/4 m. We offered these jobs to the crowd using the *Microworkers* platform as discussed in Kölle et al. [17]. Figure 7 proves our hypothesis that increasing distance to class boundaries is closely tied to label accuracy of crowdworkers. OA was improved from 68% for $d_{RIU} = 0$ m to 86% for 1.5 m and to 94% for 4 m. For $d_{RIU} = 0$ m typical confusion is due to bivalent interpretation possibilities, for instance classes *Roof* vs. *Façade*, *Impervious Surface* vs. *Low Vegetation* and *Shrub* vs. *Fence/Hedge*. Confusion between *Roof* and *Powerline* is mainly caused by the sparsity of the V3D dataset where powerlines are just single points in air difficult for interpretation. Although the labeling accuracy of most classes improves when increasing distance to decision boundary, this does not hold for class *Shrub*, which is either confused with *Low Vegetation* in case of $d_{RIU} = 1.5$ m or *Tree* for $d_{RIU} = 4$ m. This might rather be a problem of misunderstanding of this class and can therefore not be resolved by this strategy.

3.4 Estimation of Reachable Accuracies with Real Crowdworkers

Finally all previous findings are combined in order to estimate the performance of our proposed human-in-the-loop pipeline for our two datasets (Table 2 and 3) and classifiers (Table 2). In each table we compare the respective result of PL on the completely labeled training dataset to AL using $wE + DiFS$, stepwise adding RIU ($d_{RIU} = 1.5$ m, for avoiding $\mathcal{O}_C$) and a noisy oracle $\mathcal{O}_N(10\%)$ (noise assumed to be 10% following Kölle et al. [17] and Fig. 7). Table 2 outlines

Table 2. Comparison of reachable accuracies [%] for different training approaches and assumed oracles using RF and SCNN for the V3D dataset.

Method	F1-score									**OA**
	Powerl	L. Veg	I. Surf	Car	Fence	Roof	Façade	Shrub	Tree	
RF										
PL	48.39	**83.16**	**91.93**	72.68	14.94	**95.17**	**64.30**	40.60	**80.73**	**84.25**
$wE + DiFS$	61.90	80.53	90.24	**73.12**	**28.58**	94.14	57.08	**43.55**	78.99	82.43
$wE + DiFS + RIU$	67.35	79.37	89.50	70.32	28.53	92.77	60.45	39.62	79.24	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	81.17
SCNN										
PL	42.11	**81.40**	**91.11**	72.15	**41.22**	**94.10**	**59.65**	**48.87**	**83.88**	**83.86**
$wE + DiFS$	60.57	79.31	88.59	72.28	24.92	91.21	55.34	43.44	80.16	81.13
$wE + DiFS + RIU$	**63.02**	79.52	89.62	**75.03**	26.33	91.18	54.41	38.45	78.27	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	80.59

Table 3. Comparison of reachable accuracies [%] for different training approaches and assumed oracles using RF for the H3D dataset.

Method	F1-score												**OA**
	Powerl	L. Veg	I. Surf	Car	U. Fur	Roof	Façade	Shrub	Tree	Gravel	V. Surf	Chim	
PL	30.37	**93.59**	80.23	42.74	**36.71**	**93.80**	**83.03**	**71.11**	**97.84**	32.10	40.93	40.82	**84.85**
$wE + DiFS$	26.10	88.24	81.71	**65.31**	32.97	89.76	77.53	65.33	94.76	**48.65**	**64.06**	**76.22**	83.82
$wE + DiFS + RIU$	32.67	87.88	**85.29**	37.93	34.29	89.65	73.30	61.69	94.40	42.33	57.63	59.81	83.22
$wE + DiFS + RIU + \mathcal{O}_N$	**36.00**	86.70	82.74	38.73	26.90	90.08	73.85	60.96	93.54	48.48	56.14	58.75	82.22

that for the RF, $wE + DiFS$ allows to achieve a segmentation result, which only differs in OA by less than 2% points from PL while only requiring labeling of 1.15% of points from U (assuming unrealistic $\mathcal{O}_O$). When supporting the crowd by *RIU*, our results still differ less than 3% points from the baseline result of PL or only marginally worse when additionally adding $\mathcal{O}_N$. Compared to our RF classifier, the SCNN yields a slightly bigger loss in OA when applying AL, which is due to the aforementioned overfitting issue. Assuming real crowdworkers (i.e. with *RIU* and $\mathcal{O}_N$), respective accuracies are less diminished than for the RF.

For the H3D dataset (Table 3) except for n_{DiFS}, which was increased to 100.000 due to the higher point count, all parameters are same as before. Here, sampling and labeling of just 0.59% of U and assuming a realistic crowd oracle only diminishes the OA by less than 3% points. We further observed that underrepresented classes such as *Powerline*, *Gravel*, *Vertical Surface* and *Chimney* tend to perform better using AL strategies while the accuracies of overrepresented classes decrease marginally. Independent of the dataset, when using AL and considering a real crowd (last row in each table), the impact on classes *Façade*, *Shrub* and *Urban Furniture* (H3D) is greatest. This is mainly due to the great diversity within these classes. For example, with regard to *Façade* consider any type of façade furniture such as balconies, signs and lamps. Such structures might not be sufficiently sampled by AL and especially by *RIU*.

4 Conclusion

Within this paper we have shown that AL is a well founded approach for crucially reducing labeling effort for semantic segmentation, since annotation is targeted to the most informative 3D points following a similar pattern as the SVM. Basic AL sampling strategies can be purposefully enhanced by means of increasing diversity within one batch when using pool-based AL, thereby further boosting convergence of the iteration. Furthermore, we have proven that even CNN approaches can efficiently work with minimum training datasets. Since our ultimate goal is to shift labeling effort to the crowd, we aim to ease labeling for non-experts using *RIU* in order to avoid systematic errors, for we have demonstrated that especially the confused oracle greatly diminishes the performance of AL. Although *RIU* allows to significantly improve accuracies achieved by the crowd, labeling errors, which are of subjective nature and mainly caused by individual class understanding (e.g. *Tree* vs. *Shrub*), can hardly be avoided.

This work provides an in-depth understanding of the AL part of our proposed hybrid intelligence system where the machine learns solely from the crowd. In order to fully integrate the crowd into the AL loop respective web tools as presented in Kölle et al. [17] are essential. Eventually, we estimate plausible segmentation results for our classifiers (the machine) working together with real human operators (the crowd). We demonstrate that when labeling 1.15% (V3D)/0.59% (H3D) of available training points we can achieve an OA of only about 3% points less compared to PL on the completely labeled training dataset.

References

1. Argamon-Engelson, S., Dagan, I.: Committee-based sample selection for probabilistic classifiers. J. Artif. Intell. Res. **11**, 335–360 (1999)
2. Becker, C., Häni, N., Rosinskaya, E., d'Angelo, E., Strecha, C.: Classification of aerial photogrammetric 3D point clouds. ISPRS Annals IV-1/W1, pp. 3–10 (2017)
3. Branson, S., et al.: Visual recognition with humans in the loop. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6314, pp. 438–451. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15561-1_32
4. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001). https://doi.org/10.1023/A:1010933404324
5. Chehata, N., Guo, L., Mallet, C.: Airborne LiDAR feature selection for urban classification using random forests. ISPRS Arch. **38** (2009)
6. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995). https://doi.org/10.1007/BF00994018
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Li, F.F.: ImageNet: a large-scale hierarchical image database. In: CVPR 2009, pp. 248–255 (2009)
8. Ertekin, S., Huang, J., Bottou, L., Giles, L.: Learning on the border: active learning in imbalanced data classification. In: CIKM 2007, pp. 127–136. ACM, New York (2007)
9. Gadiraju, U., Kawase, R., Siehndel, P., Fetahu, B.: Breaking bad: understanding behavior of crowd workers in categorization microtasks. In: HT 2015, pp. 33–38. ACM (2015)

10. Gal, Y., Ghahramani, Z.: Dropout as a Bayesian approximation: representing model uncertainty in deep learning. In: ICML 2016, vol. 48, pp. 1050–1059. PMLR, New York (2016)
11. Graham, B., Engelcke, M., van der Maaten, L.: 3D semantic segmentation with submanifold sparse convolutional networks. In: CVPR 2018, pp. 9224–9232 (2018)
12. Haala, N., Kölle, M., Cramer, M., Laupheimer, D., Mandlburger, G., Glira, P.: hybrid georeferencing, enhancement and classification of ultra-high resolution UAV LiDAR and image point clouds for monitoring applications. ISPRS Annals V-2-2020, pp. 727–734 (2020)
13. Hirth, M., Hoßfeld, T., Tran-Gia, P.: Anatomy of a crowdsourcing platform - using the example of Microworkers.com. In: IMIS 2011, pp. 322–329. IEEE Computer Society, Washington (2011)
14. Hui, Z., et al.: An active learning method for DEM extraction from airborne LiDAR point clouds. IEEE Access **7**, 89366–89378 (2019)
15. Kellenberger, B., Marcos, D., Lobry, S., Tuia, D.: Half a percent of labels is enough: efficient animal detection in UAV imagery using deep CNNs and active learning. TRGS **57**(12), 9524–9533 (2019)
16. Kirsch, A., van Amersfoort, J., Gal, Y.: BatchBALD: efficient and diverse batch acquisition for deep Bayesian active learning. In: NIPS 2019, pp. 7026–7037. Curran Associates, Inc. (2019)
17. Kölle, M., Walter, V., Schmohl, S., Soergel, U.: Hybrid acquisition of high quality training data for semantic segmentation of 3D point clouds using crowd-based active learning. ISPRS Annals V-2-2020, pp. 501–508 (2020)
18. Krizhevsky, A.: Learning multiple layers of features from tiny images. Technical Report TR-2009, University of Toronto, Toronto (2009)
19. Li, N., Pfeifer, N.: Active learning to extend training data for large area airborne LiDAR classification. ISPRS Archives XLII-2/W13, pp. 1033–1037 (2019)
20. Lin, Y., Vosselman, G., Cao, Y., Yang, M.Y.: Efficient training of semantic point cloud segmentation via active learning. ISPRS Annals V-2-2020, pp. 243–250 (2020)
21. Lloyd, S.P.: Least squares quantization in PCM. IEEE Trans. Inf. Theory **28**(2), 129–137 (1982)
22. Lockhart, J., Assefa, S., Balch, T., Veloso, M.: Some people aren't worth listening to: periodically retraining classifiers with feedback from a team of end users. CoRR abs/2004.13152 (2020)
23. Luo, H., et al.: Semantic labeling of mobile lidar point clouds via active learning and higher order MRF. TGRS **56**(7), 3631–3644 (2018)
24. Mackowiak, R., Lenz, P., Ghori, O., Diego, F., Lange, O., Rother, C.: CEREALS - cost-effective region-based active learning for semantic segmentation. In: BMVC 2018 (2018)
25. Niemeyer, J., Rottensteiner, F., Soergel, U.: Contextual classification of lidar data and building object detection in urban areas. ISPRS J. **87**, 152–165 (2014)
26. Patra, S., Bruzzone, L.: A cluster-assumption based batch mode active learning technique. Pattern Recogn. Lett. **33**(9), 1042–1048 (2012)
27. Schmohl, S., Sörgel, U.: Submanifold sparse convolutional networks for semantic segmentation of large-scale ALS point clouds. ISPRS Annals IV-2/W5, pp. 77–84 (2019)
28. Settles, B.: Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin-Madison (2009)
29. Tuia, D., Ratle, F., Pacifici, F., Kanevski, M.F., Emery, W.J.: Active learning methods for remote sensing image classification. TGRS **47**(7), 2218–2232 (2009)

30. Vaughan, J.W.: Making better use of the crowd: how crowdsourcing can advance machine learning research. J. Mach. Learn. Res. **18**(193), 1–46 (2018)
31. Walter, V., Kölle, M., Yin, Y.: Evaluation and optimisation of crowd-based collection of trees from 3D point clouds. ISPRS Annals V-4-2020, pp. 49–56 (2020)
32. Walter, V., Soergel, U.: Implementation, results, and problems of paid crowd-based geospatial data collection. PFG **86**, 187–197 (2018). https://doi.org/10.1007/s41064-018-0058-z
33. Weinmann, M., Jutzi, B., Hinz, S., Mallet, C.: Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. ISPRS J. **105**, 286–304 (2015)
34. Xu, Z., Akella, R., Zhang, Y.: Incorporating diversity and density in active learning for relevance feedback. In: Amati, G., Carpineto, C., Romano, G. (eds.) ECIR 2007. LNCS, vol. 4425, pp. 246–257. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-71496-5_24
35. Zhdanov, F.: Diverse mini-batch active learning. CoRR abs/1901.05954 (2019)