PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science (2024) 92:131–161
https://doi.org/10.1007/s41064-024-00281-3

ORIGINAL ARTICLE

Building a Fully-Automatized Active Learning Framework for the Semantic Segmentation of Geospatial 3D Point Clouds

Michael Kölle[1] · **Volker Walter**[1] · **Uwe Sörgel**[1]

Received: 25 September 2023 / Accepted: 25 February 2024 / Published online: 3 April 2024

Abstract

In recent years, significant progress has been made in developing supervised Machine Learning (ML) systems like Convolutional Neural Networks. However, it's crucial to recognize that the performance of these systems heavily relies on the quality of labeled training data. To address this, we propose a shift in focus towards developing sustainable methods of acquiring such data instead of solely building new classifiers in the ever-evolving ML field. Specifically, in the geospatial domain, the process of generating training data for ML systems has been largely neglected in research. Traditionally, experts have been burdened with the laborious task of labeling, which is not only time-consuming but also inefficient. In our system for the semantic interpretation of Airborne Laser Scanning point clouds, we break with this convention and completely remove labeling obligations from domain experts who have completed special training in geosciences and instead adopt a hybrid intelligence approach. This involves active and iterative collaboration between the ML model and humans through Active Learning, which identifies the most critical samples justifying manual inspection. Only these samples (typically $\ll 1\%$ of Passive Learning training points) are subject to human annotation. To carry out this annotation, we choose to outsource the task to a large group of non-specialists, referred to as the crowd, which comes with the inherent challenge of guiding those inexperienced annotators (i.e., "short-term employees") to still produce labels of sufficient quality. However, we acknowledge that attracting enough volunteers for crowdsourcing campaigns can be challenging due to the tedious nature of labeling tasks. To address this, we propose employing paid crowdsourcing and providing monetary incentives to crowdworkers. This approach ensures access to a vast pool of prospective workers through respective platforms, ensuring timely completion of jobs. Effectively, crowdworkers become *human processing units* in our hybrid intelligence system mirroring the functionality of *electronic processing units*.

Keywords Active Learning · Paid Crowdsourcing · Hybrid Intelligence System · 3D Point Clouds · Semantic Segmentation

1 Introduction

Over the past thirty years, there has been a significant surge of interest in Machine Learning (ML) systems, encompassing both feature-driven and data-driven methods, with Convolutional Neural Networks (CNNs) emerging as the most prominent representative. Notably, these systems have reached a remarkable level of performance, even surpassing human capabilities (Russakovsky et al. 2015a). Consequently, ML has become an integral part of our daily lives, from autonomous driving and speech recognition to personalized recommendations on streaming platforms and shopping suggestions. However, the success of these ML models hinges heavily on diverse and well-annotated training data they learn from. Simply fine-tuning the architecture of ML models as tools to extract patterns from data would have limited success if the underlying training data is of suboptimal quality. For instance, training sets that are too specific or poorly labeled can hinder proper generalization. Therefore, recent advice from Ng (2021) emphasizes the importance of data-centric approaches, rather than solely focusing on model-centric methodologies. This approach gives companies specializing in data collection, such as *Google*, a distinct advantage in developing successful ML systems. However, since such companies often do not allow

✉ Michael Kölle
michael.koelle@ifp.uni-stuttgart.de

[1] Institute for Photogrammetry and Geoinformatics, University of Stuttgart, Geschwister-Scholl-Str. 24D, 70174 Stuttgart, Germany

public access to their data pools, there have been efforts to build large-scale and freely available training data sets, like *ImageNet* (Deng et al. 2009; Russakovsky et al. 2015a).

Typically, the process of generating training data and teaching an ML model based on that data are considered separate and isolated steps. The focus in research has predominantly been on developing new ML models, often overlooking the supposedly straightforward but tedious annotation process involved in generating training data. However, there exists a significant untapped potential in treating these two processes as interconnected and mutually beneficial. This envisioned synergy arises from the machine's ability to analyze vast amounts of data and identify crucial samples that play a pivotal role in establishing a robust separation between different classes. By allowing the machine to communicate with the labeling engine, the annotation effort can be concentrated on those highly informative samples that genuinely enhance the machine's learning process. As a result, this approach holds the promise of significantly reducing the number of required labels. The essential link between the ML model and the labeling engine, the so-called oracle, is facilitated through Active Learning (AL) (Settles 2009). This iterative learning scheme precisely entails an *active* interaction between the two parties, contrasting the traditional Passive Learning (PL) concept, where a static training set is assumed to be sufficient for the task at hand, possibly lacking critical examples while containing redundant information.

Although AL is a valuable approach to enhance the efficiency of ML models, it still relies on human operators acting as oracles to provide labels to the machine. Typically, these operators are domain experts with knowledge of the data under consideration (Waldhauser et al. 2014). Despite AL's ability to reduce their labeling burden, a substantial amount of effort is still required from them. To create a scalable solution, we propose breaking down the labeling task into numerous subtasks that can be quickly completed in parallel by a diverse group of annotators. Each of these individual annotators becomes a *human processing unit*, functioning similarly to the *electronic processing units* used for the ML component (Gingold et al. 2012). By connecting these *electronic* and *human processing units* through AL, we establish a hybrid intelligence system (Vaughan 2018). In this system, both parties play to their strengths—the unparalleled interpretation capabilities of humans and the machine's proficiency in rapidly processing well-defined, repetitive tasks for exploring extensive data sets (Waldhauser et al. 2014; Russakovsky et al. 2015b).

However, implementing such an expert-driven system practically becomes unfeasible due to the unlikely participation of enough experts in the designated field. As the ultimate objective is to relieve experts from the burden of data annotation entirely, an alternative approach involving a potentially larger audience becomes necessary. In this regard, we turn to the crowd of internet users, already engaged in non-expert data annotation through activities like *reCAPTCHA*s (von Ahn et al. 2008). Apart from unconscious participation in such tasks, we can explicitly offer labeling tasks to the crowd on a voluntary basis. However, in volunteered crowdsourcing scenarios, a lack of participants is likely, especially for tedious labeling tasks that do not have the appeal of citizen science projects such as the *Galaxy Zoo* project (Lintott et al. 2008) where plenty of astronomy enthusiasts participate driven by their desire to contribute to the scientific advances in astronomy. But for more special use cases without a society-wide desire to solve the respective problem (such as the one discussed in our work) it is unlikely to find a pool of contributors that is large enough to complete all required tasks, also in a timely fashion. To address this challenge, paid crowdsourcing is a viable solution, where tasks are outsourced to potential contributors through an open call, offering them payment as an extrinsic incentive. By leveraging respective online platforms, we can reach millions of workers acting as short-term employees for the duration of a particular campaign, so that costs can be saved both through cheap crowdwork and by avoiding idle time when only a few projects need to be worked on that do not justify hiring full-time staff. By building the AL oracle on such a paid crowdsourcing system, we can establish a practical hybrid intelligence system that operates in a fully automated manner from the system operator's viewpoint. Despite human beings, i.e., crowdworkers, being integral to this system, they behave akin to *electronic processing units*, delivering prompt results due to the vast pool of available workers on the internet, ensuring ample participants to reliably complete even extensive campaigns that would otherwise occupy a large number of experts.

One of the primary challenges in such hybrid intelligence systems is effectively integrating non-deterministic components, namely *human processing units*. This requires special attention to the human aspect regarding automated quality control (Waldhauser et al. 2014; Ye et al. 2017). The complexity of this issue is further increased when dealing with non-experts, represented by the crowd, who have never completed any training in geosciences and also will not be trained in this regard, as individual workers will most likely only participate in one specific labeling campaign due to the dynamic nature of paid crowdsourcing platforms with millions of registered users. Essentially, we aim to hire and guide our crowdworkers to generate a set of labels with a quality level *sufficient* (but most likely worse compared to an expert's labels) for training ML models requiring special measures in this regard as, to them, specialized geospatial data typically presents an unfamiliar perspective of actually known environments. Moreover, in the case of (colorized) 3D Airborne Laser Scanning (ALS) point clouds,

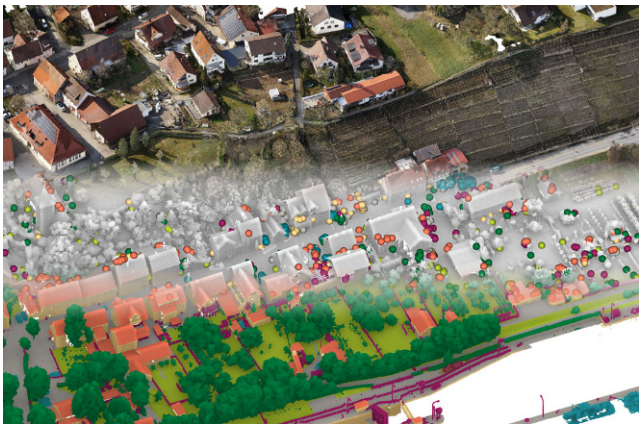

Fig. 1 From any given point cloud (top), we aim to derive a semantic segmentation (bottom) with a limited number of point-wise training samples (middle), which are generated without the involvement of an expert in labeling. This is achieved in context of a hybrid intelligence system constituted by linking the crowd with an ML model by means of AL

crowdworkers are additionally expected to possess an interpretation capability of 3D data, which may not be the case. But given the scarcity of open, large-scale, and fine-grained labeled data sets in this domain with only few available representatives (Niemeyer et al. 2014; Actueel Hoogtebestand Nederland 2021; Zolanvari et al. 2019; Varney et al. 2020; Ye et al. 2020; Kölle et al. 2021a), geospatial 3D data presents an ideal application domain for a hybrid intelligence system to fully unleash the potential of modern ML systems.

1.1 Objectives

Within this work, our primary goal is to establish a hybrid intelligence system built for the automated semantic segmentation of ALS point clouds, as illustrated in Fig. 1. To achieve this, we develop a comprehensive framework that can effectively enrich any given point cloud (cf. Fig. 1, *top*) with valuable semantic information (cf. Fig. 1, *bottom*) using ML models. The framework's key objectives are twofold: first, to significantly reduce the demand for labeled data, thereby requiring only a small fraction of training points (cf. Fig. 1, *middle*). Second, to completely eliminate the need for expert involvement in any labeling task. To achieve this, we combine two research fields, in which we identify our contributions as follows:

1.1.1 Active Learning for 3D Point Cloud Semantic Segmentation

- We aim to explore the potential of AL in the context of 3D point cloud semantic segmentation to substantially diminish the amount of labeled training data required and seek to uncover insights not only on the effectiveness of AL, but also as to *why* it works (Kölle et al. 2021a).
- For the key component of our AL system, i.e., identifying most informative points, we attempt to exploit the nature of our geospatial data to develop sampling methods that not only optimize the machine's performance but also ease the labeling process for human operators, specifically the crowd. Evaluation of resulting AL loops is done with both a feature-driven and data-driven model. This allows us to give a recommendation on the more efficient classifier in our AL context (Kölle et al. 2021a).
- In contrast to most research in AL, we challenge the assumption of an error-free oracle always providing accurate labels for the machine's queries, as this is a completely unrealistic assumption for real-world scenarios where humans, regardless of their expertise, are responsible for labeling (Marcus and Parameswaran 2015). Thus, on one hand, we simulate different error behaviors of the oracle to gain a realistic estimation of theoretically reachable accuracies. On the other hand, we refrain from any simulated offline labeling engine and instead employ a real human crowd as a viable alternative, leading us to our second research branch (Kölle et al. 2021a,b).

1.1.2 Paid Crowdsourcing for the Interpretation of 3D Geospatial Data

- In AL scenarios involving (paid) crowdworkers as oracles, an essential yet often neglected challenge is the creation of user-friendly labeling tools for non-experts, as highlighted by Kittur et al. (2008). Dealing with 3D data further complicates this issue since a significant number of workers may lack experience in this domain. Consequently, when developing appropriate tools for crowdsourced 3D data annotation, we venture into relatively unexplored territory.
- Given the inherent issue of data quality inhomogeneity in crowdsourced data collection, addressing this concern is of utmost importance. While one option could involve manual inspection of crowdworkers' results by an operator, this approach would significantly reduce the appeal of crowdsourcing. Therefore, an effective quality control system must operate automatically, eliminating the need for manual intervention. To achieve this, we adopt a dual strategy for quality control, both on task designing and in post-processing, with the latter leveraging the concept of *the Wisdom of Crowds* (Howe 2006) to obtain high-quality labels (Kölle et al. 2021b).

Please note that the work presented here is a significantly enhanced version of our previous contribution (Koelle et al. 2023) differing mainly in a more comprehensive methodol-

ogy section and adding further experiments incorporating real crowdworkers instead of just simulating such a crowd oracle.

2 Related Work

Before actually setting-up the envisioned hybrid intelligence system, we start with reviewing related literature with respect to its individual components. Firstly, this involves building a robust *human processing unit* that should operate with minimal errors as it is intended for teaching the machine (Sect. 2.1). Subsequently, the second part of the system is implementing an ML scheme capable of accessing the crowd engine *selectively* (Sect. 2.2), ultimately leading to the development of a powerful model for 3D point cloud classification.

2.1 Crowdsourcing Geospatial Information

The advent of Web 2.0 brought significant changes in the way we work, including the introduction of innovative concepts like crowdsourcing. This entails delegating tasks originally handled by individuals or single institutions to a large group of people, known as the crowd (Howe 2006). Within this context, solutions are typically built by a vast number of crowdworkers collaborating, each contributing to the overall objective by completing assigned subtasks (i.e., microtasks). A notable application of this approach is found in citizen science, where individuals interested in research offer their assistance (e.g., see the work of Korpela et al. (2001) and Okolloh (2009)). The generation of geodata through such a scheme has been termed Volunteered Geographic Information (VGI) by Goodchild (2007), with *Open Street Map (OSM)* being its most prominent representative.

2.1.1 Quality Control in Crowdsourcing

However, the primary concern surrounding crowdsourced (geospatial) data revolves around its quality (Goodchild and Li 2012; Fan et al. 2014; Senaratne et al. 2016). The crowd is a diverse mix of individuals from various cultural backgrounds, age groups, and educational histories (Kittur et al. 2008), consequently leading to heterogeneous data quality (Howe 2006; Goodchild 2007; Haklay and Weber 2008; Shaw et al. 2011; Dorn et al. 2015; Fonte et al. 2017; Chandler and Paolacci 2017; Chandler and Kapelner 2013). This is primarily because crowdworkers often lack familiarity with specific data concepts, such as geodata (Hashemi and Abbaspour 2015), and acquisition standards are often neglected to avoid discouraging enthusiastic participants (Antoniou and Skopeliti 2015). Furthermore, there is a risk of having inattentive crowdworkers (Fleischer et al. 2015) or even malicious ones deliberately providing false information (Hirth et al. 2013; Welinder et al. 2010; Whitehill et al. 2009). Zhang et al. (2016) propose distinguishing between two aspects to ensure the quality of crowdsourced data: *quality control on task designing* and *quality improvement after data collection*. An overview of various methods addressing these issues can also be found in the work of Chandler et al. (2013).

Quality control on task designing can be effectively achieved by presenting tasks in a clear and understandable manner for non-experts. This includes providing concise information about the data they will be working with and specifying the required actions (Sorokin and Forsyth 2008). Sorokin and Forsyth (2008) and Allahbakhsh et al. (2013) advocate for cheat-proof task design and propose filtering participating groups based on employer criteria. Various measures are commonly employed to guarantee quality. These include implementing qualification tests (Patterson et al. 2014; Estes et al. 2016; Endres et al. 2010), incorporating a pre-task training stage, and including tasks where true answers are already known (Sorokin and Forsyth 2008; Estes et al. 2016; Gebru et al. 2017; See et al. 2013; Hirth et al. 2013; Zhou et al. 2014; Salk et al. 2015; Marcus and Parameswaran 2015; Vondrick et al. 2012). Moreover, an intrinsic quality measure is presenting certain data points multiple times to the same crowdworker to assess result consistency (See et al. 2013).

A simple realization of the second principle, *quality improvement after data collection*, is reviewing already submitted tasks either by an expert (e.g., the employer) or other crowdworkers (Liu et al. 2018; Russell et al. 2007). More advanced methods draw inspiration from the concept known as the *Wisdom of the Crowds* (Galton 1907; Surowiecki 2004). This idea suggests that a group of independent individuals can produce results of similar, if not superior, quality compared to any single expert within that group. Translating the *Wisdom of the Crowds* paradigm into a rule of action for crowdsourced data acquisition entails duplicating a given task, assigning it to multiple crowdworkers, and aggregating their contributions, which can lead to high-quality results (Sorokin and Forsyth 2008). For labeling tasks, the simplest aggregation approach is majority voting (Parhami 1994). However, a drawback of this approach is the increased cost associated with assigning the same task to multiple crowdworkers. Recent research is focused on determining the optimal number of repetitions needed for *Wisdom of the Crowds* to take effect (van Dijk et al. 2020; Walter et al. 2022).

2.1.2 Motivating the Crowd to Contribute

In the previously mentioned crowdsourcing projects, individuals are incentivized to participate by intrinsic factors,

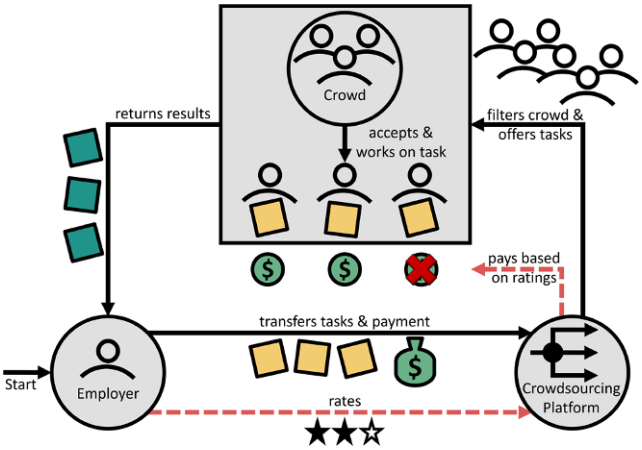

Fig. 2 General concept of paid crowdsourcing

such as the opportunity to create tools that are useful in everyday life, as seen in the case of *Open Street Map (OSM)*. However, this raises the question of how to effectively harness *human computing capacity* of crowdworkers for tasks that may not be inherently appealing, like data labeling. To address this issue, the concept of paid crowdsourcing (see Fig. 2) was introduced, offering easy access and motivation of crowdworkers through monetary compensation. To simplify the technical overhead of the employer, dedicated crowdsourcing platforms like *Amazon Mechanical Turk (MTurk)* (Chandler et al. 2013) and *microWorkers (MW)* (Hirth et al. 2011) were established, acting as mediators between employers and the crowd.

Relying on such platforms, the employer posts campaigns as an open invitation, allowing crowdworkers to decide whether they wish to participate. The employer sets essential parameters, such as the payment amount and may also apply certain criteria to limit the targeted group of participants (e.g., requiring a specific reputation score). Once crowdworkers submit their completed tasks, the employer can approve the work and prompt payment through the platform if satisfied (as illustrated by the *dashed red arrows* in Fig. 2). Additionally, the employer has the option to award boni to crowdworkers (Geiger et al. 2011). Typically, crowdworkers receive a remuneration amounting to a few cents per task, with the median payment on the *MTurk* platform being approximately 1.38 $/h (Haralabopoulos et al. 2019).

Compared to voluntary crowdsourcing projects, the importance of quality control is even greater when using paid crowdworking. While paid crowdworkers may strive to attain or maintain a reasonable reputation score (Mao et al. 2013), they are often found to be less reliable than their volunteered counterparts (Redi and Povoa 2014). In the paid crowdsourcing domain, these crowdworkers are commonly referred to as *satisficers*, as they tend to exert minimal effort (Chandler et al. 2013; Marcus and Parameswaran 2015) while aiming to maximize their income (Gingold et al. 2012). For instance, research conducted by Kittur et al. (2008) revealed that up to 30% of submissions in paid crowdsourcing projects might suffer from low quality.

2.1.3 Crowdsourcing as Part of Hybrid Intelligence Systems

Data labeling itself plays a crucial role in advancing modern ML methods and provides ample justification for the existence of paid crowdsourcing platforms (Vaughan 2018). However, the significance of such platforms extends beyond this. As emphasized by Jeff Bezos, these platforms are designed for a wide range of tasks that are simple for humans to solve but exceedingly challenging for machines: *"Normally, a human makes a request of a computer, and the computer does the computation of the task. But **artificial** artificial intelligences like Mechanical Turk invert all that. The computer has a task that is easy for a human but extraordinarily hard for the computer. So instead of calling a computer service to perform the function, it calls a human."* (Bezos 2007).

The vision presented here centers around collaborative work between machines and humans (Allahbakhsh et al. 2013), leading to the concept of *hybrid intelligence systems* (Vaughan 2018). These systems encompass algorithms where certain subroutines are not handled by machines but are delegated to *human processors*, allowing each party to perform the parts of the processing chain in which they perform best. For example, human interpretation capabilities may complement the automatic performance of repetitive, easily formulated tasks. To realize such systems, paid crowdsourcing platforms are crucial to ensure the availability of a sufficient number of crowdworkers (processors) on demand, enabling scalability. However, a potential challenge arises when non-deterministic humans (or "semi-qualified workers") who possibly lack a comprehensive understanding of errors harmful for ML systems (Endres et al. 2010), become integrated into such an algorithm (Ye et al. 2017; Endres et al. 2010).

Although paid crowdsourcing systems and ML models fueled by or operating in conjunction with crowdsourced data are commonplace in the Computer Vision community, these techniques are still in their infancy in the geospatial domain where commonly fully annotated training sets are expected. Nonetheless, ongoing research is already directed towards developing a robust human component that could potentially serve as a basis for a corresponding machine component to learn from (Li and Zipf 2022). However, dedicated campaigns to build data for geospatial scene interpretation are only scarcely conducted. Most studies in this area deal with employing a paid (Walter and Soergel 2018) or unpaid crowd for annotating and interpreting aerial images (Salk et al. 2015; Estes et al. 2016; Juni and Eck-

stein 2017) or street view scenes (Hara et al. 2015; Hecht et al. 2018; Maddalena et al. 2020). But 3D point clouds as the data of interest are very rarely considered, and, to the best of our knowledge, are limited to the contributions of Herfort et al. (2018); Walter et al. (2020) and Walter et al. (2021).

2.2 Active Learning (AL)

Hybrid intelligence systems, as discussed in the preceding section, often leverage the concept of AL (Settles 2009; Kovashka et al. 2016). Besides connecting *human* and *electronic processing units*, the fundamental idea of AL, particularly in the context of semantic segmentation tasks, is to concentrate labeling efforts on only a subset of the available instances. In contrast to the conventional ML approach of PL, where a fixed training data set is used, AL *actively* involves the classifier in setting up the training data set. Starting from an initial (suboptimal) training set, the objective is to iteratively enhance it (and thus the ML model) by labeling those samples from the unlabeled data set the classifier is currently the most uncertain about in its prediction, which ultimately reduces the epistemic uncertainty of the ML model (Gal et al. 2017). In essence, the goal of AL is to enable a given classifier to achieve top performance with minimal labeling effort.

AL can be considered as a system of four main components: (i) the base classifier for the task at hand, (ii) the query function for the selection of instances to be labeled (Sect. 2.2.1), (iii) the so-called oracle, e.g., a human operator (Sect. 2.2.2) and (iv) an abortion criterion for the iteration (Sect. 2.2.3).

2.2.1 Querying Samples in AL

As the performance of an AL system is heavily influenced by the design of its query function, naturally, this is also the main focus in literature. Solutions stem from various methodologies, such as *uncertainty sampling*, *query-by-committee* strategies, and *representativeness sampling*.

Initially, *uncertainty sampling* was closely linked to Support Vector Machines (SVMs) (Cortes and Vapnik 1995), where decisions are solely based on support vectors, regardless the size of the labeled data set. As a result, labeling the remaining instances is actually superfluous, also providing insight *why* AL is so effective. In this regard, Ertekin et al. (2007) proposed sampling only those points situated closest to the current decision border of the classifier. The primary aim of *uncertainty sampling* is to select samples that the current classifier considers most uncertain in terms of inter-class similarity (Settles 2009). To measure this uncertainty, typical query functions, such as *least certainty* sampling (Lewis and Gale 1994), *breaking ties* sampling (Scheffer et al. 2001), or *entropy* sampling (Shannon 1948), operate directly on the posterior probability.

On the other hand, *Query-by-committee* is specifically designed to accurately estimate epistemic uncertainty, but it requires the presence of a committee or an ensemble of classifiers. The sampling strategies employed in this approach aim to select instances from regions of the feature space that have not been well-represented so far, where the ensemble members, all trained using the same data set, disagree. This disagreement is typically assessed by *vote entropy* (Argamon-Engelson and Dagan 1999), *Kullback-Leibler divergence* (McCallum and Nigam 1998), or *mutual information* between model predictions and model parameters, as employed in context of *Bayesian Active Learning by Disagreement (BALD)* (Houlsby et al. 2011).

Representativeness-based sampling strategies aim to query a subset of points that is as representative as possible for the entire data set. For instance, this can be accomplished by solving a *core-set-selection-problem* (Sener and Savarese 2018) or by computing a hierarchical clustering of all data to enable gradual sampling of points following this hierarchical structure (Dasgupta and Hsu 2008). In the Deep Learning (DL) era, *representativeness*-based sampling can, for instance, be realized with *VAAL* (Sinha et al. 2019). The core idea is to jointly learn a latent feature space from both labeled and unlabeled instances using a variational autoencoder, while an adversarial discriminator is trained to distinguish whether a sample is from the labeled or unlabeled data set, so that those new points can be sampled that, according to the discriminator, are likely to come from the unlabeled set.

Although the aforementioned sampling heuristics were developed in the pre-DL era, theoretically, CNNs could be employed out-of-the-box as they also output (pseudo) posterior probabilities. However, CNNs are notorious for overestimating their confidence when extrapolating in regions of the feature space not represented in training (Gal and Ghahramani 2016), i.e., they lack awareness of epistemic uncertainty. To obtain more sensitive uncertainty measures, the authors propose approximating *Bayesian* CNNs using *Monte Carlo dropout ensembles* (Gal et al. 2017). An alternative approximation for *Bayesian* CNNs are *deep ensembles*, which, according to Beluch et al. (2018) and Feng et al. (2019), often outperform *Monte Carlo dropout ensembles* due to higher capacity and greater independence among ensemble members. However, this advantage comes with the drawback of higher computational cost, as it involves training multiple networks. For a comprehensive review of both the challenges and solutions of Deep AL, refer to the work of Ren et al. (2022).

Independent of the chosen heuristic, all sampling strategies are meant to query points based on the current state of a classifier. As the inclusion of any additional training

samples may alter the model's current beliefs, these methods perform best when selecting one instance per iteration step. However, this approach leads to re-training the ML model after adding only one single sample, which can be considered statistically unreasonable and computationally infeasible, especially when employing CNNs as base classifiers (Sener and Savarese 2018; Kirsch et al. 2019). Consequently, a common practice is to sample a larger batch of points in each iteration step. However, when multiple points with the highest scores are selected, often respective instances are too similar to one another, essentially leading to labeling quasi-duplicates. As a result, labeling resources may be wasted on non-most-informative instances. To address this issue, methods that yield a diverse batch of informative samples are often employed. Achieving such diversity can be accomplished through score-weighted *k-means* clustering of feature space, where k corresponds to the number of points to be sampled (Zhdanov 2019; Ash et al. 2019; Prabhu et al. 2021).

2.2.2 Oracles in AL

In addition to determining the most suitable data points for the ML process, careful attention must also be given to the oracle when creating an AL scheme. In many cases, a Ground Truth (GT) oracle is assumed, i.e., an oracle that consistently provides perfectly accurate answers to label queries. While this assumption can be useful for evaluating the theoretical performance of simulated AL runs, it is unrealistic to expect such results when working with human annotators, especially crowdworkers, as the obtained results might contain both random and systematic labeling errors (Chandler et al. 2013; Lockhart et al. 2020).

Moreover, it is essential to acknowledge the presence of samples that are inherently more challenging and error-prone to label compared to others, thus also incurring higher costs due to the need of assigning them to multiple crowdworkers to achieve a consensus (Deng et al. 2009). An alternative approach is adapting the sampling strategy to balance informativeness for the machine with feasibility for human labeling, i.e., selecting samples that remain as informative as possible while being suited for human oracles to label. For instance, Mackowiak et al. (2018) propose a method that combines a common AL informativeness measure with a learned estimate of the required labeling effort, creating a final score function that balances both factors. Similarly, Vijayanarasimhan and Grauman (2009) train an SVM classifier to predict labeling costs based on image features along with annotation times collected from real *MTurk* crowdworkers.

2.2.3 Terminating AL Loops

To ensure effective termination of an AL iteration, a suitable stopping criterion must be defined striking a balance between running the AL loop long enough to achieve stable performance and minimizing unnecessary iteration steps to reduce costs (Settles 2009). While the easiest way is to measure performance on a large representative test set, this approach defeats the purpose of AL, which aims to restrict labeling to only a few samples. An alternative solution is to assess the similarity between newly queried samples and those already included in the training set (Vlachos 2008). In the same spirit, a SVM-specific stopping criterion is to terminate the iteration as soon as no support vectors are available anymore (Ertekin et al. 2007). Other approaches measure the stability of the predictions of the current classifier on a large unlabeled data set, focusing on the classifier's confidence in its predictions (Vlachos 2008), or the agreement between the current and previous predictions (Bloodgood and Vijay-Shanker 2009). If an ensemble classifier is used, stopping can be based on comparing the classification disagreement of the ensemble members for the remaining unlabeled pool with that of an independent and unlabeled validation set (Olsson and Tomanek 2009).

2.2.4 AL in Remote Sensing and Semantic 3D Point Cloud Segmentation

In the remote sensing community, AL has also been explored, with a primary focus on minimizing labeling effort, such as reducing visual inspection of data or conducting field surveys. The main application lies in semantic segmentation of aerial imagery, where well-known AL concepts, as discussed previously, are applied (Tuia et al. 2011; Crawford et al. 2013). Dealing with spatially meaningful data introduces both challenges and opportunities, as summarized by Crawford et al. (2013). One challenge arises in modifying conventional sample selection strategies when field surveys are necessary for labeling. In such cases, the selection of samples (i.e., locations) should consider the traveling distance between queried locations. On the other hand, spatial information can be leveraged to diversify the selection of the most informative samples. Thoreau et al. (2022) recently conducted a comparison of sampling strategies for a state-of-the-art CNN classifier in remote sensing. Among this research, the work of Tuia and Munoz-Mari (2013) is noteworthy as, to the best of our knowledge, it is the only one that acknowledges an imperfect oracle, consequently tailoring the query of points to its needs, i.e., avoiding samples that most likely are too complex/ambiguous for labeling.

While the application of AL for semantic segmentation of 2D images has been extensively studied, its utilization

for the semantic segmentation of 3D point clouds, particularly ALS point clouds, has been explored only to a limited extent.

To the best of our knowledge, the first work addressing this issue is the pipeline presented by Luo et al. (2018) for the semantic segmentation of high-resolution Mobile Laser Scanning (MLS) point clouds based on a pair-wise Conditional Random Field model built upon a Random Forest (RF) classifier operating on supervoxels. Mainly for the classification of terrestrial point clouds, also CNN-based approaches were proposed (Wu et al. 2021; Shi et al. 2021; Shao et al. 2022). They follow a similar pattern of first computing superpoint regions as AL units instead of single point, each with its distinctive selection strategy. For the sake of completeness, another AL approach operating on MLS scans worth to be mentioned, is proposed by Feng et al. (2019). Although designed for object detection, this method also exploits *Monte Carlo dropout ensembles* and *deep ensembles* for *entropy*-based uncertainty estimation.

As previously mentioned, AL approaches for semantic segmentation of ALS point clouds are rare. In one such study, Hui et al. (2019) handle the generation of a Digital Terrain Model (DTM), i.e., the filtering of ground points, as a classification problem. The authors employ a SVM and use AL to iteratively refine the filtering process and thus the resulting terrain model. In each iteration step, a sigmoid function scores the distance of points to the current DTM level which, together with the SVM prediction, is used to automatically assign a point to the *Ground*/*Non-Ground* class. A similar understanding of an oracle can be found in the work of Li and Pfeifer (2019), who design a semi-supervised AL pipeline for 3D point cloud classification where labels of an initially provided (suboptimal) training data set are propagated each to the point in an optimal neighborhood that incorporates the highest *breaking ties* score to gradually improve an RF classifier.

Supervised AL for large-scale point cloud classification is conducted by Lin et al. (2020b,a). The authors work with a regularly tiled point cloud as input and employ AL to minimize the required training data by selecting only the most informative tiles. However, the authors expect that the selected tiles receive full point-wise labeling, which, from an economic point of view, does not fully exploit the potential of AL to reduce costs by minimizing required labels. For sampling, classic point-wise *entropy*, segment-wise *entropy* and *mutual information* based on a *Monte Carlo dropout ensemble* are compared to each other.

3 Methodology

As explained earlier, the integration of the individual components of our hybrid intelligence system is achieved by means of AL, which is the primary focus of our methodology and is described in detail in Sects. 3.1–3.4. This is followed by a discussion of considerations for both our *human processing units* (Sect. 3.5) and *electronic* ones (Sect. 3.6), to allow the machine to learn from annotations supplied by the crowd.

3.1 An Outline of the Proposed Hybrid Intelligence System

To initiate the process, as shown schematically in Fig. 3, an unlabeled ALS point cloud U is given to an annotation engine $\mathcal{O}$, that can be either a simulated GT oracle or a realistic crowd oracle (cf. Sect. 3.5). In an ideal scenario, we would be capable of attracting and motivating a substantial pool of crowdworkers to willingly engage in our labeling campaigns. However, promoting such a campaign and then recruiting crowdworkers is a time-intensive and often fruitless venture, particularly when dealing with tedious data annotation tasks. To overcome these recruitment challenges, we turn to paid crowdsourcing. By leveraging appropriate platforms (in our case *MW*), the campaigns can be published as open calls to all crowdworkers that are registered on the particular platform, which greatly facilitates the recruitment aspect.

Our crowd's first task is generating an initialization data set of samples for each class defined by an operator, that can then be leveraged to train an ML model for semantic segmentation of 3D point clouds (cf. Fig. 3). Subsequently, we can derive a prediction for all remaining unlabeled data points (cf. Fig. 3). As this already yields a complete annotation of the originally given point cloud U, the pipeline could theoretically already be stopped at this point. But since it is unlikely that a model built solely on such a sparse initialization data set would produce satisfactory results, a refinement of the model and the derived annotation is now sought.

Continuing the workflow is based upon the assumption that the classifier is able to detect the samples in the remaining pool of unlabeled data U with the highest predictive uncertainty, and that inclusion of these points in the training data set would result in a more powerful model (i.e., we consider a pool-based AL setting). In order to identify the few most informative points (cf. Fig. 3), an adequate query function is used to compute individual sampling scores. The selected n^+ points are then fed to the oracle $\mathcal{O}$, which is again responsible for labeling. Relying on the training pool augmented with the current batch of training samples, the next training cycle of this iterative procedure can commence and the loop is repeated until the labeling budget is depleted or, preferably, until convergence, where a more extensive labeling effort will only lead to marginal performance improvements.

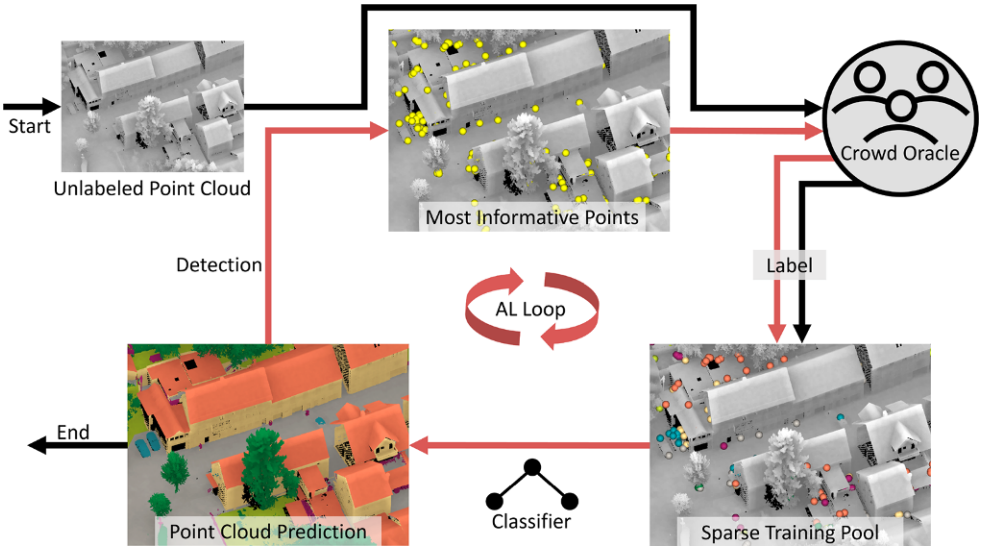

Fig. 3 An overview of our crowd-based pipeline for semantic segmentation of 3D point clouds

We reason that the entire AL loop, involving the crowd, can be regarded as a completely automated workflow. Although humans are part of such a hybrid intelligence system, they can be considered as *human processing units* that cooperate with *electronic processing units* and behave in the same way as the *electronic* units.

3.2 Exploring *why* AL is a Viable Alternative to PL

Prior to discussing the details of the AL working principle and the employment of crowdworkers in this approach, we give some insight as to why learning with just a few samples is a well-founded procedure. Besides being an achievement of AL, the idea of building a classifier on the basis of only a few samples is also implemented in the well-known SVM classifier. To train such a classifier, a fully labeled training set is often utilized, but eventually to train an SVM is to identify the so-called support vectors. In the end, the partitioning of the feature space using hyperplanes is based solely on these support vectors. We also illustrate this concept in Fig. 4, where we trained an SVM classifier on the basis of both geometric and radiometric features for the V3D benchmark data set (cf. Sects. 3.6 and 4). We can conclude that the training procedure is equivalent to a filtering step, where only points describing object boundaries are preserved, as the point cloud of support vectors in Fig. 4b is akin to a map showing the demarcations or outlines of individual buildings and properties. Therefore, we anticipate that such points will be the most informative for the purpose of classification.

It seems that if one of the top classifiers of the pre-DL era, namely an SVM, can get by with so few samples, other classifiers ought to be able to do likewise. The significance of this finding lies in the tremendous potential to save labeling effort. For example, our SVM classifier retains only 21.68% of the training points provided and infers its predictions only from these, so labeling the remaining points is needless, actually.

So if we can figure out a way to directly and automatically identify the points that have the strongest impact on the final model *without* knowing their labels, we could really save the cost of labeling the rest of the points, which

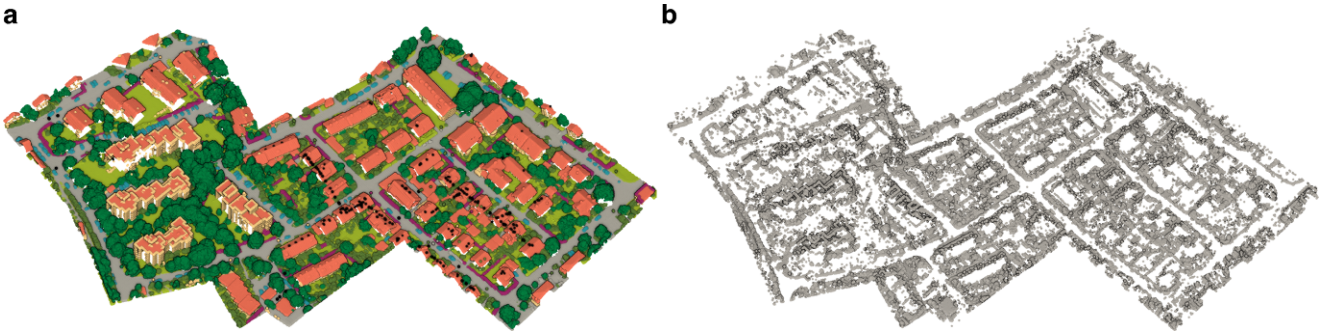

Fig. 4 Comparison of the labeled V3D training data set **(a)** and the derived support vectors **(b)**. The support vectors are mainly located at the class borders, so that outlines can be clearly identified. Class color coding is depicted in Fig. 5

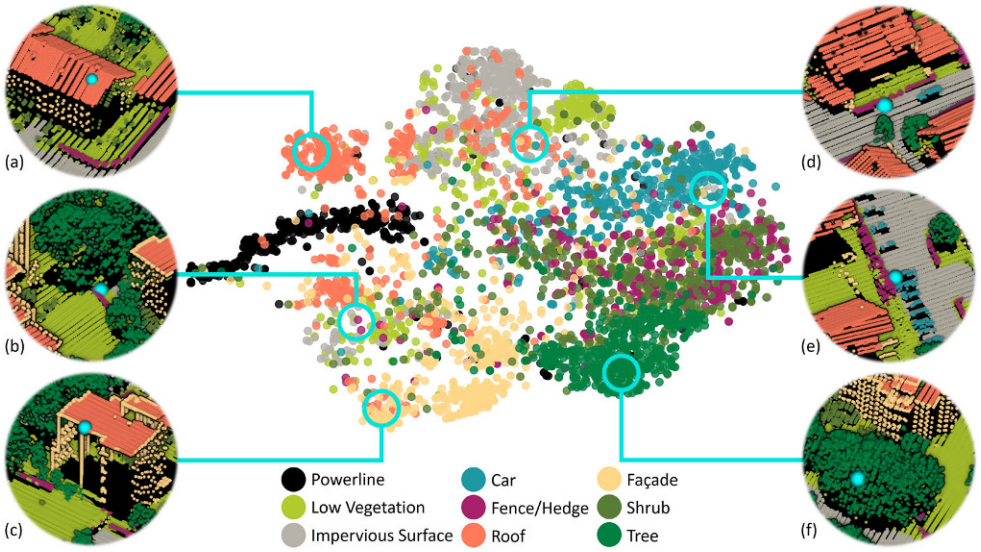

Fig. 5 Embedding feature vectors from V3D into 2D space through *t-SNE* along with exemplary regions from which AL points are drawn (blue circles). For each region shown, a representative is traced into object space and displayed in blue. While points **(a)** and **(f)** were selected by crowdworkers in the initialization step, the remaining examples were actively queried during the AL loop

is where AL comes in. Given an initial training data set, we assess the uncertainty of the model at predicting unlabeled data points and retrieve those for which the model is most uncertain. If this sampling is achieved by an efficient query strategy, we often need to label even a smaller number of points than an SVM would find support vectors (Mackowiak et al. 2018; Kellenberger et al. 2019). The reason for this is the SVM's utilization of the entire vicinity of the decision border. Thus, many informative but often alike points, quasi-duplicates, are involved. AL seeks to circumvent this by using specially tailored query functions (cf. Sect. 3.3). In our later experiments, up to 81.21% of the sampled points are in fact support vectors. The implication is that there are also non-support vectors in the training set, sampled primarily in early iteration steps in which the most challenging regions cannot yet be identified due to suboptimal estimation of the separation hypothesis at this stage (Tuia et al. 2011). However, as the iteration progresses, the separating hyperplanes approach an optimal position as the training set gradually increases, allowing for more accurate queries (of support vectors).

For further insight into how AL works, we aim to visualize the regions that AL focuses on in both feature space and object space. While classifiers operate in a high-dimensional feature space, humans tend to analyze the distribution of selected points in object space. However, the selection of AL points is only a result of effects in feature space, so we opt to focus on that as well. In the interest of interpretability, though, we map the high-dimensional feature space to 2D using *t-SNE* (van der Maaten and Hinton 2008). This is done for the training set of V3D, using the same features as in the SVM classification.

Fig. 5 exemplarily depicts regions of the training set from which AL draws its points in the (reduced) feature space, and also traces these selections into object space. We differentiate between those regions from which samples are drawn in the initialization step (examples a and f) and regions visited during the iteration (examples b, c, d and e). In the initialization step, crowdworkers, of course operating in object space, can freely select representatives of all classes. They naturally tend to selecting points that are as easy to label as possible, i.e., they would simply pick a point in the middle of the object in question, far away from the class borders (see Fig. 5 object space snippets). Indeed, this agrees well with the illustration in the *t-SNE* plot, where we can see rather homogeneous regions. However, active sampling within the AL loop happens in feature space and primarily favors heterogeneous regions where there is a mixture of different classes. Tracing exemplary points of such regions back into object space, we notice that such points are not only close to the decision borders in feature space, but also close to the class borders in object space. This leaves only the question of how to identify these most informative instances in the high-dimensional feature space, which is the focus of the next section.

3.3 Querying Points in the AL Loop

Effective AL sampling strategies usually take into account the predictive uncertainty of the classifier given its current state. Uncertainty-based sampling strategies directly rely on the posterior probability $p(c|\boldsymbol{x})$ that the point $\boldsymbol{x}$ belongs to class c. Because we deal with multi-class settings where we wish to reflect the predicted score for all n_Ω classes, we use *entropy* E (Shannon 1948) to assess this uncertainty. By theory, this metric evaluates aleatoric uncertainty, i.e., samples are drawn from near the current decision border when the posterior probability is given by a single classification model (though in early iteration steps epistemic uncertainty

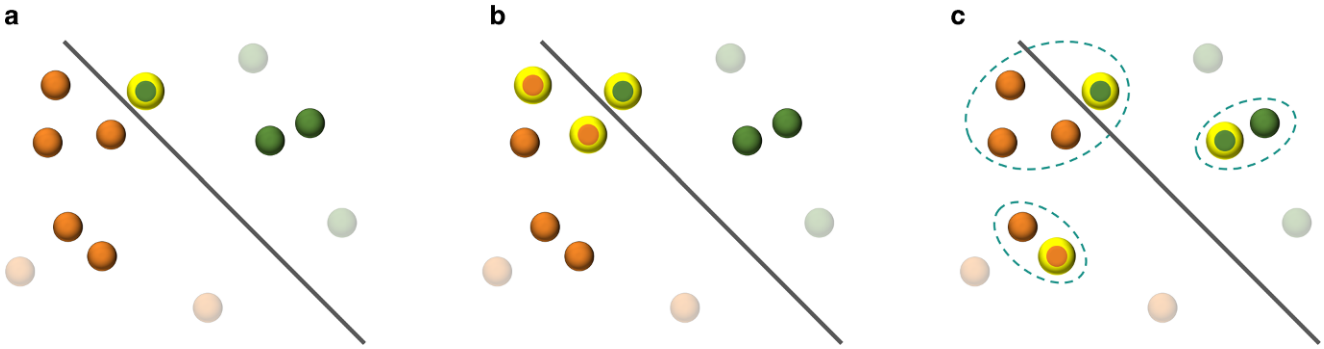

Fig. 6 Comparison of different sampling strategies to select most informative points. Transparent points represent the current training data defining the decision border. Yellow border lines indicate samples with highest scores. **a** *Entropy* sampling—emphasizing aleatoric uncertainty, but also epistemic 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

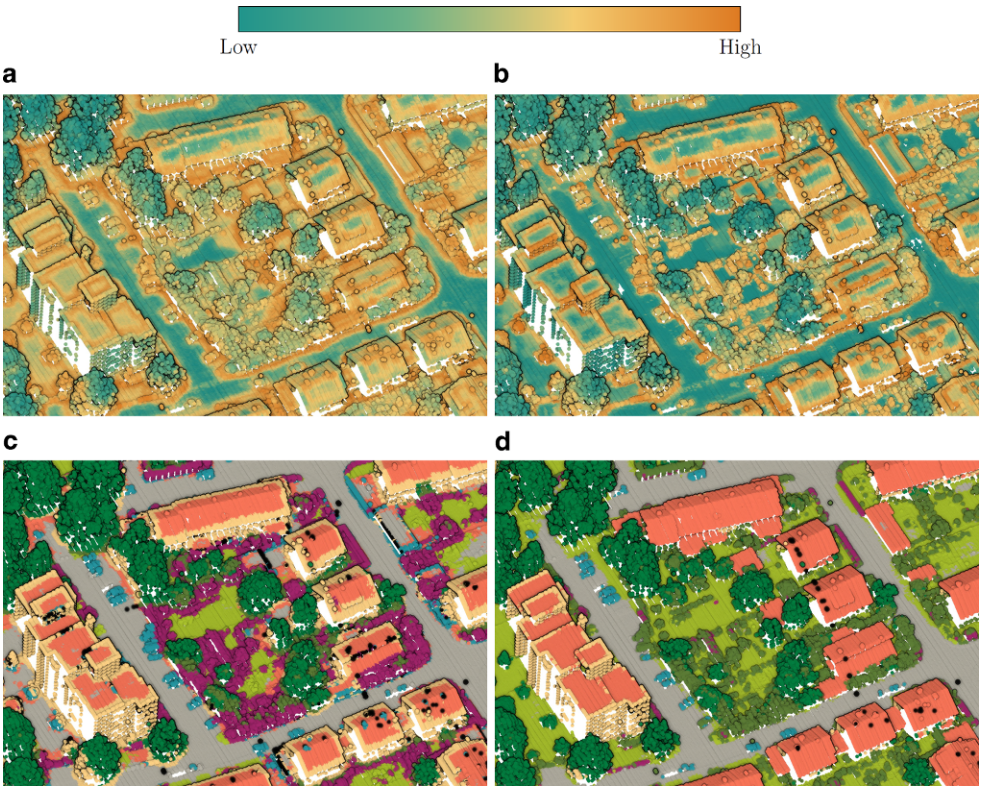

Fig. 7 *Entropy* evaluated in the course of the iteration for the first **(a)** and the last **(b)** iteration step for the V3D data set. Overall, the classifier gets more confident in its predictions, well corresponding to an improvement of predicted class labels (**c** vs. **d**), but class borders remain most challenging. Class color coding is depicted in Fig. 5

may dominate the scores). Yet, we can also measure epistemic uncertainty if, for example, the averaged posterior probability of several models, e.g., from a *deep ensemble*, is used. Either way, samples are drawn according to:

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

Fig. 6a provides a visual interpretation of *entropy* sampling. For a two-class setting, the *entropy*-based sampling strategy corresponds to selecting the point closest to the current decision border of the model as learned from training points that are already available. Fig. 7a and b displays an exemplary evaluation of the *entropy* scores for the V3D data set for different stages of the iteration. While the classifier becomes more confident as the iteration progresses, we can note that throughout the loop, points located on or near the class borders are the most uncertain for the classifier to categorize, as they may reveal ambiguous feature vectors, but also provide the most helpful hints for refining the decision borders. Ultimately, we prioritize sampling of support vectors (cf. Sect. 3.2) to incrementally improve the current separation hypothesis.

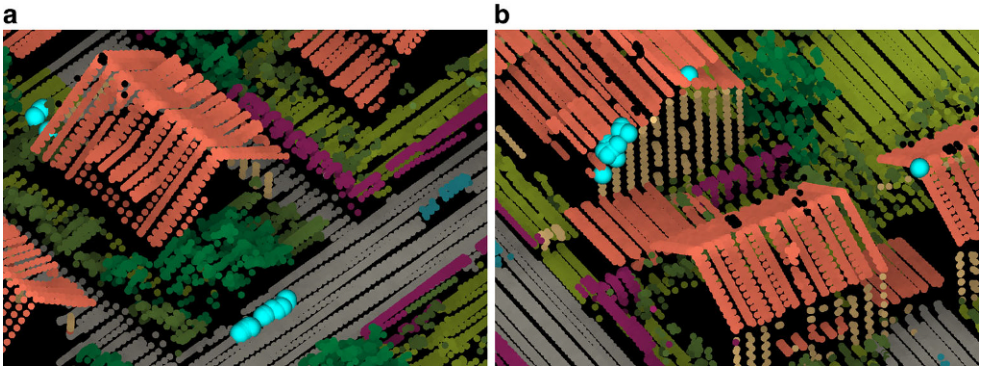

Fig. 8 *Entropy* sampled points (depicted in cyan) of two exemplary iteration steps of a batch-mode AL loop. Class color coding is depicted in Fig. 5

3.3.1 Fostering an Equally Class-Distributed Training Set

In the case point selection is based exclusively on *entropy* scores, it is likely that classes that are underrepresented in the data set will also be underrepresented in our point queries, assuming that the relative number of points near the class borders is equal to that of the entire data set. Thus, if points of smaller classes remain undrawn, refinement of the decision border(s) to effectively separate such underrepresented classes is unlikely to be prioritized and could only be done implicitly by improving other class borders. Therefore, moving the decision border(s) in feature space to a cluster representing such an underrepresented class is also not prioritized, and if points of such a class are not adjacent to the class border, they may never be sampled.

Consequently, dedicated measures to give more priority to underrepresented classes are advisable. This can be accomplished by introducing a dynamic weighting factor $w_c(i)$ for each class c at each iteration step i, by taking into account the ratio between the total number of samples currently contained in the training dataset n_L and the number of samples for a given class n_c:

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

Those weights are then multiplied with the predicted score of each class, normalized and inserted into the *entropy* formula. That way, the class score of the rarest class is always increased compared to the other classes. The operating principle of the weighting scheme is twofold. While we aim to increase the sampling scores of points from rare classes, at the same time we aim to decrease the scores of well-represented classes in order to "free up" sampling capacity for samples from such rare classes.

3.3.2 Guaranteeing Diversity in Sampled Batches

In terms of optimal point selection, the classification rule should be updated with each new training sample (selected using the above procedure) to obtain the best possible estimate of the next point to be labeled. However, one sample per iteration step is statistically questionable and simply not efficient for most classifiers. Therefore, a batch of points is usually selected at once. The simplest approach to this would be to pick the n^+ points with the highest sampling score. When this is done based on *entropy*, the selected samples can be fairly similar to one another (in terms of their representation in both feature and object space), as can be seen in Fig. 6b and Fig. 8. In other words, with the exception of the sample closest to the decision border, quasi-duplicates are picked that may not provide significant value to refine the separation hypothesis. Statistical significance is thus addressed, but efficiency remains an issue. To boost the convergence of the AL loop, the most informative and diverse set of points should be selected. To achieve the latter, a feature space clustering algorithm can be used to obtain n^+ clusters, where we would like to select only one point per cluster so that all points in the set are sufficiently diverse. This can be realized by the *k-means* algorithm (Lloyd 1982), which sets k cluster centers (in our case $k = n^+$) such that the Euclidean distance between each data point in U and the distributed cluster centers $\boldsymbol{\mu}$ is minimal.

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

Note that we also introduce an individual weighting for each data point, given by the sampling score s (i.e., E or its weighted variant wE) of each point (Zhdanov 2019). In a pure *k-means* sampling, the focus would be solely on the diversity criterion, but clusters would likely populate high-density class concentrations with low score values. Consequently, only an insignificant improvement in current classification beliefs can be expected when picking points from these clusters. Including the sampling score in the optimization process, can ensure that most clusters are close to the decision borders, but still we group most similar points into the same clusters, thus avoiding sampling duplicates.

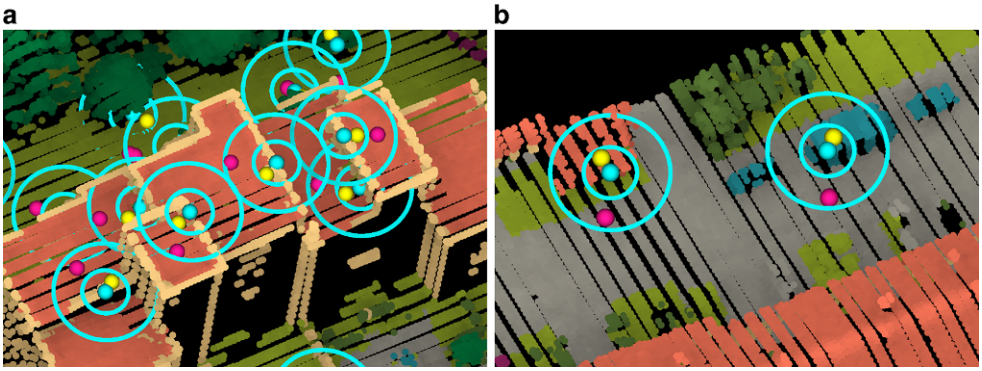

Fig. 9 Operating principle of our *RIU* technique for reducing label ambiguities for the oracle, shown for two different scenes. Instead of the actual queried point (cyan), an alternative point within a certain radius (1.5 m for yellow and 4 m for magenta), which is intended to be easier for human interpretation, is sent to the oracle for labeling. The color coding of the classes is shown in Fig. 5

Fig. 6c demonstrates the improvement in terms of batch diversity when relying on feature space clustering (pure *k-means* clustering is shown for simplicity). After clustering, we select the point with the highest sampling score from each cluster and refer to this method as *Diversity in Feature Space (DiFS)*.

3.3.3 Addressing Imperfect Oracles in AL

While *entropy* sampling together with its addition to ensure diversity (*DiFS*) aimed at the best selection of points from the machine's point of view, we conclude this section by also addressing the fact that, unlike many AL-related publications (Marcus and Parameswaran 2015), we are dealing with a realistic, imperfect crowd oracle. Thus, to obtain correct answers from crowdworkers, we may have to sacrifice informativeness. The idea is to consider the choice, i.e., the score of the machine, only as a prior for the final point selection.

As discussed in Sect. 3.2 and evident from Fig. 7, classifiers tend to select points located at the class borders in object space. There we have the highest informativeness values because points contain features of multiple classes. But we assume that interpretability for the machine is also related to interpretability for humans, so the points where the ML algorithm is not confident in its decision are also challenging for crowdworkers. Therefore, we argue that increasing the distance to class borders is related to *Reducing Interpretation Uncertainty (RIU)*, and thus refer to this method as such. More specifically, we consider a point chosen by the machine as the seed point, but instead query a certain other point within a certain radius d_{RIU} around this point. In the end, the point with the lowest score in this region, and thus hopefully with the highest interpretability, is selected.

Fig. 9 visualizes this modification in point querying. Like expected, the classifier selects points at class borders where it might be difficult to decide which class to choose. For example, the points at the borders between the classes *Façade* and *Roof* in Fig. 9a may very well be assigned to either class, depending on personal understanding of class membership. When we move away from the class borders, such ambiguities disappear, and a human operator can safely and quickly decide on a particular class. However, if the distance of the point from the class border becomes excessive, the label may end up being of no use for training the ML model (cf. Fig. 9). Thus, we are faced with a trade-off between human interpretability and information content for the ML model.

3.4 Stopping the Loop

As highlighted earlier, our objective is to establish an automated hybrid intelligence system with the primary goal of minimizing human involvement and associated expenses to the greatest extent possible. In the pursuit of both automation and cost reduction, the significance of an efficient stopping criterion for the AL loop becomes evident. One relatively straightforward approach would involve a scenario wherein a limited labeling budget dictates the permissible number of labels, subsequently determining the number of iteration steps (with a known batch size n^+). A more sophisticated termination criterion, on the other hand, proves valuable in situations where the labeling budget is substantial and the aim is to continue the loop for a sufficient number of iteration steps to achieve stable performance while avoiding unnecessary expenses. In essence, we aspire to determine the optimal point where further iteration steps would only slightly impact results.

The inherent difficulty in defining a stopping criterion is that we cannot assume to have a representative labeled test data set on which we could evaluate our classifier—otherwise, the AL idea of avoiding such a necessity would be violated. Nonetheless, it is feasible to generate predictions for a sufficiently extensive and representative point cloud. This could encompass either the remaining *unlabeled* training data set denoted as U or an independent *unlabeled* test data set. By this, we can assess the agreement between predictions from two successive iterations. If a notable disparity exists between the present predic-

tion and that from d_{stop} iteration steps earlier, it's likely beneficial to continue the iteration. Drawing inspiration from the methodology proposed by Bloodgood and Vijay-Shanker (2009), we compute the overall congruence C_o through a straightforward comparison of predicted labels. Furthermore, we derive an additional measure, C_{ac}, which is sensitive towards underrepresented classes. To this end, for each class, a unique congruence value is computed by evaluating whether all points currently designated as that specific class were assigned to the same class in a specific prior iteration step. Subsequently, an average value across all class-specific congruence values is computed to yield a class-aware indicator. Terminating the loop is based on comparing the standard deviation of congruence values (either C_o or C_{ac}) computed over the most recent n_{stop} iteration steps (which can be interpreted as the derivative of congruence values) against a user-defined threshold t_{stop}. This avoids the necessity of specifying an absolute congruence value, a parameter prone to considerable variation depending on the task and data specifics. The threshold's definition governs the stringency of the stopping criterion, thereby determining whether the loop should stop at the earliest iteration step where minimal change is anticipated or continue until convergence can be confidently presumed.

3.5 The Crowd as AL Oracle

While the previous sections discussed AL as backbone of our hybrid intelligence system, we will now focus on the human component, in our case the crowd. Before doing so, we briefly review different oracle types in an AL setting for semantic segmentation (Sect. 3.5.1) and afterwards focus on means to minimize labeling errors, which can be achieved either by easy-to-use tools (Sect. 3.5.2) or measures for automated quality control (Sect. 3.5.3).

3.5.1 Oracle Types in AL

In the end, the performance of our AL pipeline is determined by the quality of labels provided by the oracle for the selected points. While in literature, often an all-knowing GT oracle $\mathcal{O}_O$ is assumed, this idealization falls short in real-world scenarios where human annotators are entrusted with point labeling. Consequently, labeling errors should also be incorporated into oracle simulations. These errors can manifest either as entirely random or can exhibit systematic tendencies (Lockhart et al. 2020). In the case of a noisy oracle $\mathcal{O}_N$, a fraction of points is consistently assigned to classes other than the correct one. However, more significantly, a confused oracle adheres to distinct mapping functions (e.g., systematically labeling façades as class *Roof*), a phenomenon that can have a significant negative impact on classifiers (Kölle et al. 2021a). This issue is particularly pronounced in AL, as point sampling occurs at class borders (both in feature and object space, as depicted in Fig. 5), where selected points are often ambiguous. Consequently, systematic errors can arise due to differing class interpretations. However, with *RIU* (cf. Sect. 3.3.3), we hope to avoid occurence of the latter effect.

3.5.2 Designing Labeling Tools for the Crowd

Prior to discussing techniques aimed at enhancing the accuracy of labels generated by the crowd, we provide a concise introduction to our web tools necessary for executing the crowd-related tasks within our AL-driven framework. In essence, these tools should be fashioned to be effortlessly accessible, avoiding the need for extensive intros. This approach aligns with the primary objective of the majority of crowdworkers in a paid crowdsourcing scenario, which is quickly earning money. Consequently, extensive instructions are likely to be ignored either way (Endres et al. 2010).

As mentioned in Sect. 3.1, the crowd is expected to provide both an initialization data set and labels for points requested during the AL loop. The workflow begins by showing the RGB-colored point cloud to crowdworkers. They are instructed to mark one point for each class defined by the system operator at the beginning of the AL loop. A snapshot of the tool used for this purpose, referred to as crowd task *Type A*, is displayed in Fig. 10a. Since crowdworkers can freely choose points, ensuring quality control becomes challenging because there is no real opportunity to include checks in the task. Moreover, using a method like majority voting after data collection is not practical since it is unlikely that multiple crowdworkers will pick the same point.

To ensure generation of a high-quality initialization data set for the classifier, we have developed a second tool aimed at identifying errors originating from the prior campaign with our *Type A* tool. As evident from Fig. 10b, the overall design of our *Type B* tool closely resembles that of *Type A*. Nonetheless, its complexity is considerably streamlined as crowdworkers in fact (and in contrast to *Type A*) are not required (but allowed) to interact with the data. Instead, they only need to determine whether a presented point is correctly assigned to its designated class. This task is expected to be straightforward in terms of both 3D navigation and interpretability. Points identified as incorrect can subsequently be discarded, effectively rendering crowdworkers employing the *Type B* tool as "filters" for refining the outcomes generated by *Type A* campaigns. The third tool, denoted as *Type C*, is tailored for labeling points selected during the AL iteration (cf. Sect. 3.1), and will thus be employed most frequently. *Type C* is structurally akin to *Type B*, with the primary distinction being that crowdworkers are tasked not with a binary decision, but with solving a multi-class classification problem. The significant advan-

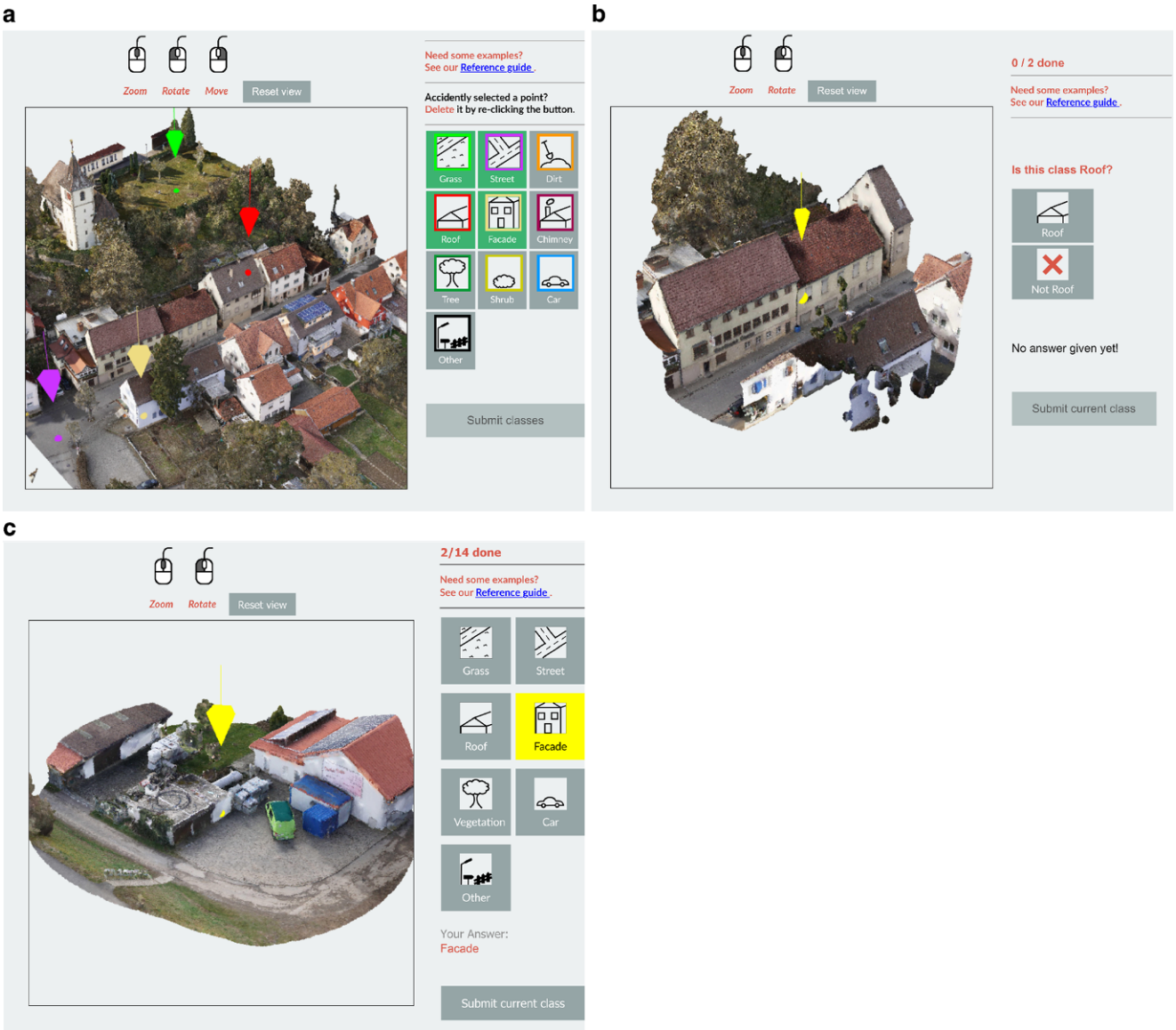

Fig. 10 Compilation of our tools required for the human crowd component within our AL-based framework. Each tool also offers a short introduction video, a task description, and a feedback option, not visualized here. **a** *Type A*: Selecting one point for each class, **b** *Type B*: Checking labels of selected points, **c** *Type C*: Labeling selected AL points

tage of these latter two tools lies in their inherent ability to allow for quality control measures, as outlined in the next section.

3.5.3 Automated Quality Control

In order to mitigate crowd labeling errors, we draw upon strategies outlined by Zhang et al. (2016), encompassing: (i) *quality control during task design* and (ii) *quality enhancement after data collection*. As previously mentioned, these measures are only valid for our crowd tasks of *Type B* and *Type C* (cf. Sect. 3.5.2). Crucially, all quality control approaches must be inherently automatable, devoid of manual checks or operator interventions, as our ultimate objective is to achieve a fully automated pipeline.

To address *quality control during task design*, a straightforward approach is presenting a specific point for labeling multiple times. This redundancy serves to assess the crowdworkers' consistency in their labeling, without necessarily guaranteeing correctness of the label. As this tactic might not effectively identify crowdworkers deliberately assigning the same class label to all points, irrespective of the data, a more robust strategy is combining consistency tests with check points for which the true label is known (Kittur et al. 2008). These check points can be randomly mixed into real payload tasks. Consequently, only results from crowdworkers who successfully pass all validation checks

are kept and justify payment of the respective worker. This strategy can lead to crowdworkers stopping contributing to our campaigns out of frustration of not passing respective checks, which however can be considered a desired property as our campaigns will not benefit from said worker's contributions and we would rather have respective points labeled by another worker out of the millions of registered crowdworkers.

But even crowdworkers passing the check points might not provide optimal labeling for the actual payload points. This concern is particularly pronounced in our *Type C* tasks, where labeling selected AL points can be considerably complex for interpretation (especially in later iteration steps) so that a single crowdworker might fail in deriving the true label. To counteract this issue by means of *quality improvement after data collection*, we leverage the concept of the *Wisdom of the Crowds*. In our case, this can be realized by assigning a task to multiple crowdworkers and afterwards aggregating the results automatically through majority voting, again avoiding engagement of an expert. However, in paid crowdsourcing, repetitively labeling points translates to a multiplication of costs. Thus, the number of multiple acquisitions should be kept limited, which raises the question of *how many is enough*, as recently answered by Kölle et al. (2021b) for this specific task setting.

3.6 Classifiers for 3D Semantic Segmentation

After presenting the human part of our hybrid intelligence system, for the machine part, the only component left to discuss is the classifier that is to be employed for semantic segmentation of 3D point clouds. To demonstrate generalizability of results, we 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 for 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 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 Fig. 11, 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 Fig. 11 often resemble typical features utilized by feature-driven classifiers. For instance, Fig. 11a is reminiscent of a verticality measure and Fig. 11c 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 Fig. 11e resembling height above ground, which can only be inferred from a wider spatial context. Contrary to the encoding branch,

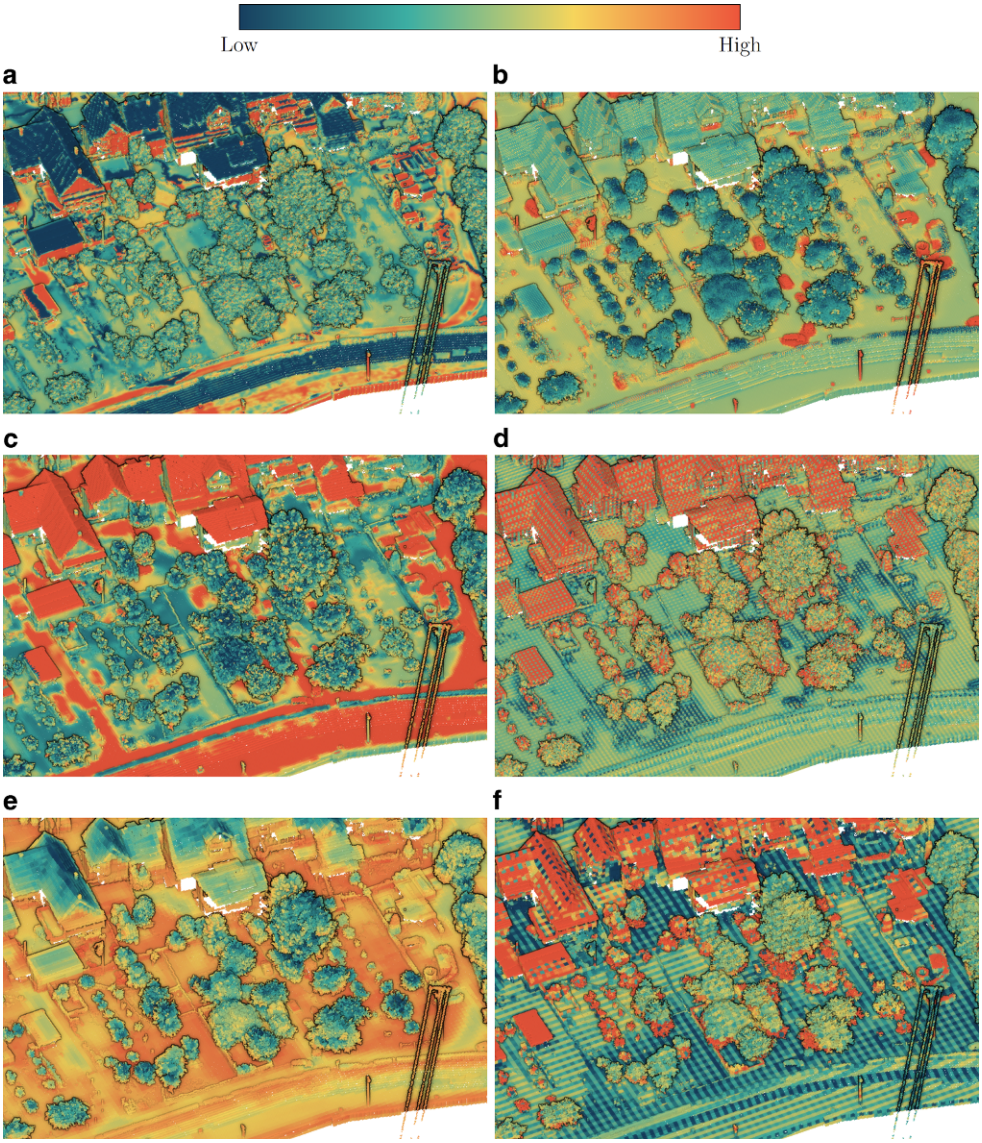

Fig. 11 Filter responses from selected filters at different levels of our SCN. Subfigures are arranged in an order to match the *U*-shape of the SCN. **a** Encoding branch level 1, **b** Decoding branch level 1, **c** Encoding branch level 2, **d** Decoding branch level 2, **e** Encoding branch level 3, **f** Decoding branch level 3

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, Fig. 11f 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 Fig. 11d. Eventually, Fig. 11b is clearly suited to extract points of a specific class, in this case class *Car*.

4 Data

To test the presented methodology, we draw on the current benchmark data sets for semantic segmentation of geospatial ALS point clouds provided by ISPRS. These data sets include the Vaihingen 3D Semantic Labeling Contest (V3D), which serves as a representative ALS point cloud example (Niemeyer et al. 2014), and epoch march 2018 of the high-resolution Hessigheim 3D Benchmark (H3D) captured from an UAV (Kölle et al. 2021a). While both data sets offer diverse and demanding class categories, they are limited in spatial coverage. To supplement this limitation, we incorporate a third data set — a National Mapping Agency ALS point cloud depicting Stuttgart's city center (S3D). This data set spans an area approximately 30 times larger than the V3D data set, yet it features a relatively small class catalog, as indicated in Table 3. Nonetheless, this data set is suitable for evaluating the scalability of AL.

Since one of the goals of this work is to minimize labeling effort by experts, we aim to give an estimate of the labeling costs for such data sets, exemplarily for H3D's epoch November 2018. Annotating the complete point cloud from

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

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

scratch and checking each point two times (checks were conducted by different student assistants each) took about 1490 hours. That is for an area of about 0.207 km^2 resulting in an average time effort of 0.431 min/m^2 and makes with a salary of \$14.69 /h an amount of \$0.106 $/m^2$, respectively. Please note that, on one hand, labeling costs highly depend on the complexity of the scene, the defined class catalog and the desired accuracy level, defining the number of multiple acquisitions or checks. On the other hand they vary with the skill and salary of the workers. Nevertheless, this calculation is supposed to give an impression of effort required to obtain data for real-world projects. For com-

parison, Zolanvari et al. (2019) report a similar workload of 0.194 min/m^2 for labeling the DublinCity data set (13 classes).

5 Results

Following the outline of our methodology (Sect. 3) and the presentation of the data sets of interest (Sect. 4), we can finally run our hybrid intelligence system for a variety of experiments. At first, we will focus on finding an optimal configuration and parametrization of the machine part utilizing a simulated crowd oracle (cf. Sect. 5.1), which is then subsequently replaced by a real crowd oracle (cf. Sect. 5.2).

5.1 Simulation of the Hybrid Intelligence System

To assess the (theoretical) capabilities of AL, we now employ the different sampling strategies and classifiers described in Sects. 3.3 and 3.6 to our three different data sets. 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 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 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.

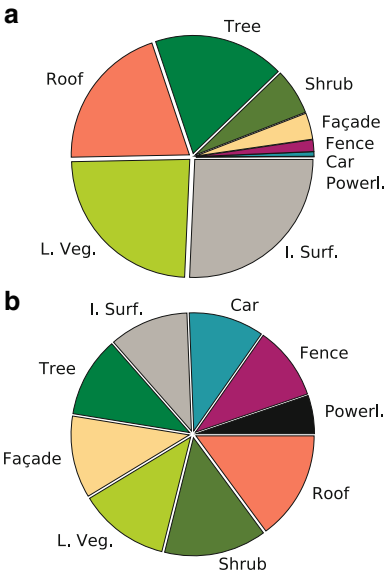

Fig. 12 Comparison between the class distributions in the original V3D training data set **(a)** vs. the one obtained by AL after 30 iteration steps **(b)**

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. Sect. 3.3.1), has a distribution that is close to that of an equal distribution, as clearly visible from Fig. 12.

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 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).

5.2 Running the Real-World Hybrid Intelligence System

Recalling the overall aim of enabling ML by an efficient and fast generation of training data through paid crowdsourcing, this section is dedicated to evaluating the performance of a hybrid intelligence system formed by combination of an ML algorithm and the crowd, merged by means of AL. Thus, this section can be considered as reiteration of parts of the experiments conducted in the previous section, differing, however, by the fact that we no longer rely on simulated oracles, but on real crowd oracles $\mathcal{O}_C$. Again, we test the presented framework for all three test sites with different characteristics (cf. Sect. 4), especially focusing on the individual and joint performance of the crowd in interplay with the machine learning from *the human processing units*. All the experiments discussed in the following are run utilizing the *CATEGORISE* framework (Kölle et al. 2021b).

5.2.1 Initializing the Loop

Due to our goal of transferring all labeling tasks from an expert operator to crowdworkers, their first responsibility is the generation of an initialization data set necessary to kick off the AL run, i.e., the interactive communication between crowd and machine in the first place. Thus, by utilizing our web tool of *Type A* (cf. Sect. 3.5.2), for each data set, our crowdworkers are asked to identify one point for each class. Precisely, 100 workers are tasked at a payment rate of \$0.10. The respective confusion matrices for each data set can be found in Fig. 13a for the V3D data set. Unsurprisingly, a lot of crowdworkers deliver insufficient labels, causing a rather low OA of about 53%, which is due to no quality control mechanisms being in place.

To refine generated labels, the next step is to filter the results obtained in the first place by a second group of crowdworkers. This is accomplished by means of our *Type B* labeling tool (cf. Sect. 3.5.2). Based on the findings of Kölle et al. (2021b), here we ask a total of 3 crowdworkers to check each point at a payment of \$0.10 + \$0.05 (base payment + bonus) in the sense of majority voting. Table 4 gives an overview of the results from these binary categorization campaigns (*Correct*/*False*) after aggregation of answers from our crowdworkers by means of majority voting. Since we consider these campaigns as *filters*, recall values are of special interest for detecting both false and correct labels. While almost no correctly labeled points are marked as incorrect by the crowd, crowdworkers struggle to identify false labels. In other words, crowdworkers seem to hesitate to flag points as incorrect if they are not completely sure about the correct answer. In this regard, the crowd performs worst on the H3D data set, which is actually supposed to give the easiest representation of data.

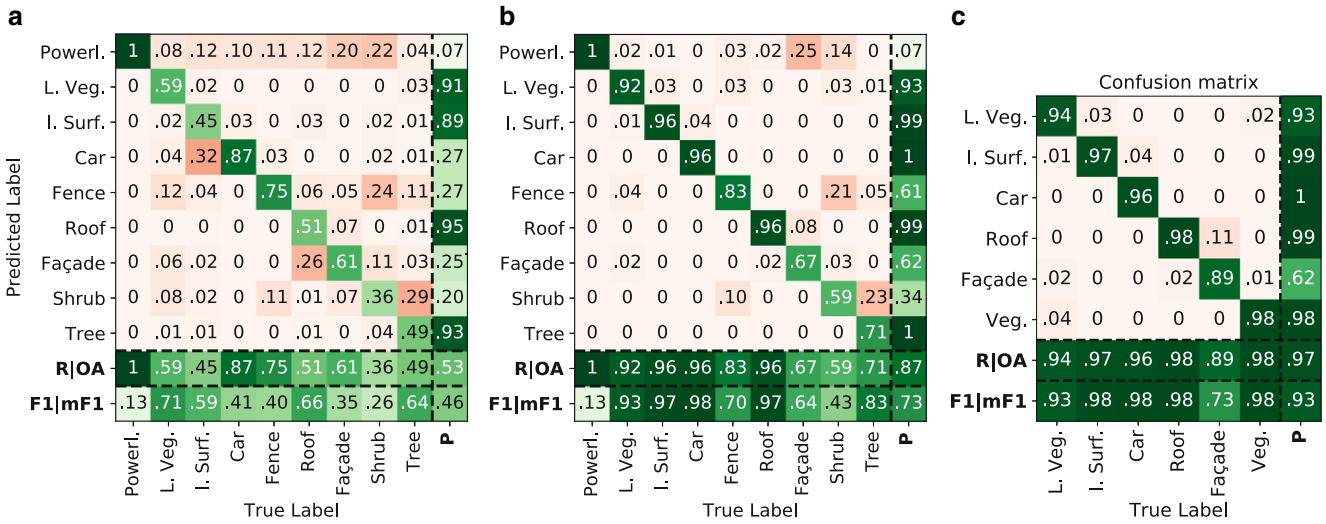

Fig. 13 Labeling accuracy of the crowd in context of the initialization of an AL loop on the V3D data set. The accuracy of the raw and unchecked crowd labels **(a)** are controlled by a second group of crowdworkers in a crowd campaign of *Type B* to refine results **(b)**. To further ease the labeling complexity, we restrict the class catalog and merge classes accordingly **(c)**

Table 4 Statistics of our *Type B* crowd campaigns for improving the initially assigned labels by crowdworkers. The relative amount of remaining samples refers to all points that are marked by the crowd to be labeled correctly and thus remain for the initialization data set. All measures are given in [%]

Data set	Recall		OA	Rel. amount of Remaining samples
	False	*Correct*		
V3D	87.94	98.05	91.66	57.60
H3D	60.31	97.31	84.44	78.80
S3D	85.71	98.96	95.25	75.25

However, the poorer performance is mainly due to more obvious errors in the other data sets. V3D and S3D rely on a suboptimal colorization by orthogonal projection of a temporally disjoint orthophoto. Among other issues such as roof-like colorization of façades, this is especially problematic for dynamic objects such as cars. In our campaigns, often, points geometrically depicting streets but radiometrically representing cars (due to mapping of car color values to streets) are mistaken for cars in the *Type A* campaigns. With a closer look at the point cloud and with the simplified 3D point cloud navigation requirements in our *Type B* campaigns, such errors can be easily detected. Thus, data sets with non-optimal colorization perform better in this evaluation of *Type B* campaign results (cf. Table 4) since the errors are more obvious. After checking results, all points that are marked as *False* are eliminated, which is why a high recall value for *Correct* is particularly desirable (i.e., dropping true labels should be minimized). On average about 30% of initially collected labels are discarded in the process (cf. Table 4) and *filtered* confusion matrices can be built (cf. Fig. 13b). By this, we can improve OA of our crowdsourced labels by about 24 pp on average (over all data sets), underlining the impact of such control tools, but this improvement in accuracy comes at the expense of the size of the initial training set.

Nevertheless, some errors remain, especially for the V3D and H3D data sets, containing rich class catalogs. This means that some classes are hard to comprehend for our crowdworkers, being in accordance to the findings of Bayas et al. (2016), stressing a limited feasible class catalog for crowdsourced data acquisition. However, these errors are more due to rather ambiguous classes and not hard label errors, e.g., in case of V3D, *Shrub* vs. *Tree* and *Fence* vs. *Shrub*, but also due to insufficiently and sparsely depicted objects such as powerlines. Since distinguishing between such classes is hard to communicate to crowdworkers and even experts might argue on those, for V3D, we decide to merge classes *Fence*, *Shrub* and *Tree* into class *Vegetation* and merge class *Powerline* into class *Roof*. Similarly, for H3D, classes *Shrub* and *Tree* are merged into *Vegetation*, *Chimney* is added to *Roof* and *Soil/Gravel* to *Low Vegetation* (class *Vertical Surface* was merged directly with *Façade* in the first place following a similar arguing). Naturally, this step further improves accuracies, where remaining confusion for V3D mainly happens between classes *Roof* vs. *Façade*, which is also due to the aforementioned merging of classes *Powerline* and *Roof*, as crowdworkers originally sometimes tend to label sparsely discretized linear point agglomerates as *Powerline* instead of *Façade*. Nevertheless, training data sets with an average OA over all our test sites of 95% can be utilized for initializing the AL loop.

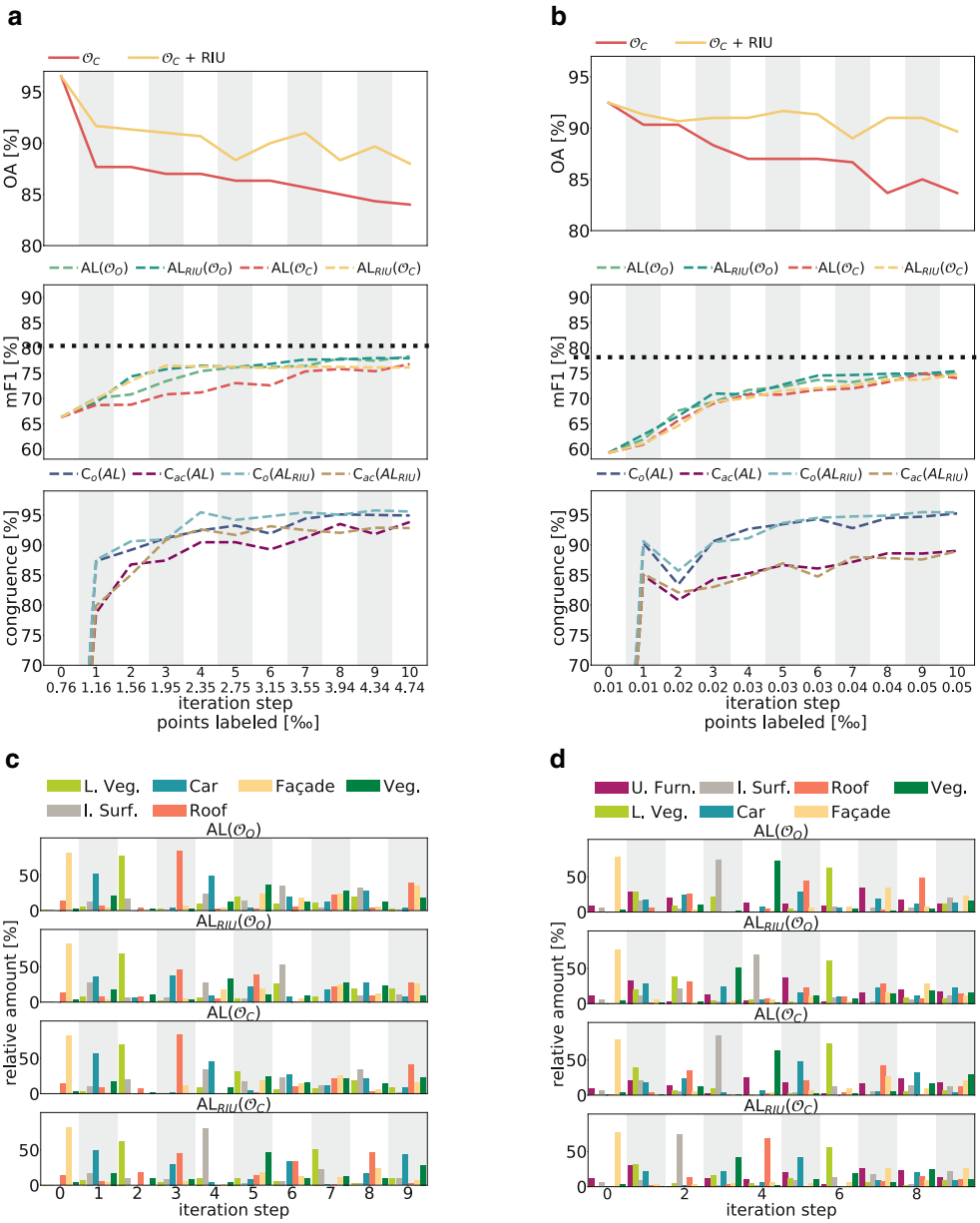

Fig. 14 Overview of the achieved performance for our hybrid intelligence system for both the V3D (first column) and H3D data set (second column). In **a** and **b** we report the achieved labeling accuracy of the crowd (first row) as well as the accuracy of the machine learning from the crowd (second row) along with our congruence measures $\boldsymbol{C}_o$ and $\boldsymbol{C}_{ac}$ evaluated for runs relying on real crowd oracles $\mathcal{O}_C$. Dotted black lines represent the result of PL. Depictions in **c** and **d** represent the class-wise relative amount of points sampled in each iteration step evaluated based on the GT labels

5.2.2 Parametrization of the Hybrid Intelligence System

Based on these initial training sets, we launch respective AL loops for all our test sites applying the *weighted entropy* sampling strategy with *DiFS* sampling add-on and a batch size $n^+ = 300$. In each case, we conduct a total of 10 iteration steps and allocate 10 payload points per crowd job (of *Type C*), which are posted to 3 different crowdworkers for the purpose of majority voting at a payment rate of \$0.10 + \$0.05 (Kölle et al. 2021b). To compare the runs relying on a crowd oracle $\mathcal{O}_C$ to the theoretically maximum achievable performance, in each case, we also simulate corresponding AL loops by utilizing GT oracles $\mathcal{O}_O$ and contrast all runs to the respective PL solutions using the completely labeled training set. Regarding this, Fig. 14 and 15 give an impression of the learning process of the hybrid intelligence system over all iteration steps. Table 5 summarizes the final accuracies achieved by the ML model for the semantic segmentation task.

5.2.3 The Performance of the Crowd

With respect to the labeling accuracy of the crowd, we can observe that the OA values yielded in the respective iteration steps are significantly worse compared to those of the initial training set (i.e., iteration step 0) (cf. Figs. 14a and b and 15a). But in contrast to the initial labeling step, where crowdworkers themselves select the points to be labeled, in the AL iteration steps, they are determined by the machine. Since the ML model aims to resolve ambiguities between

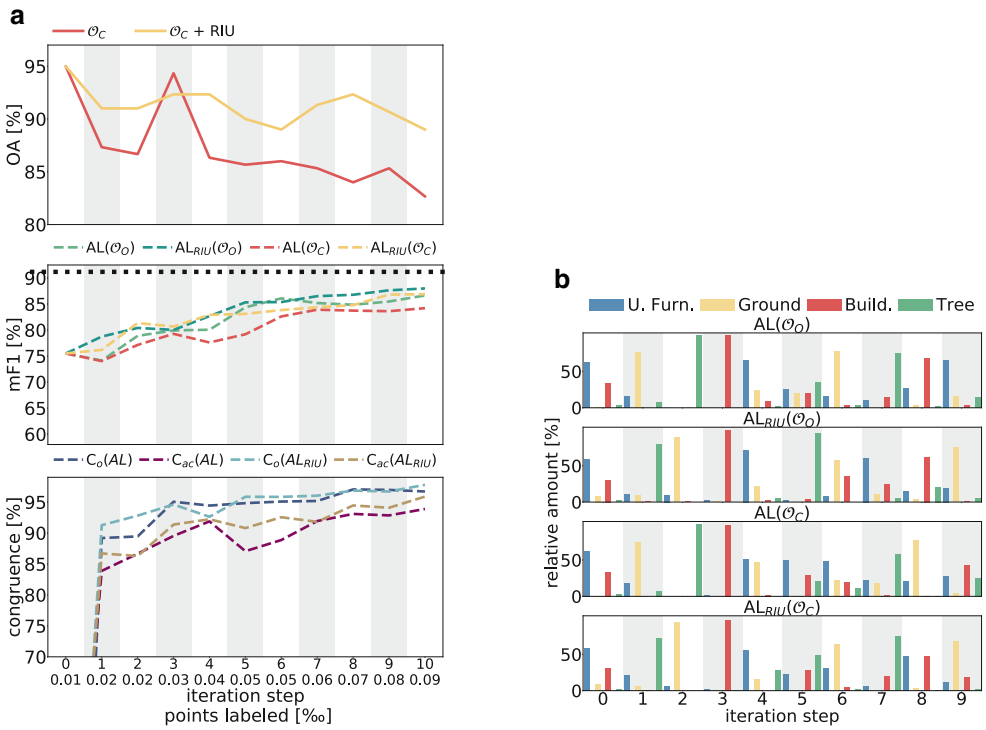

Fig. 15 Overview of the achieved performance for our hybrid intelligence system for the S3D data set. In **a** we report the achieved labeling accuracy of the crowd (first row) as well as the accuracy of the machine learning from the crowd (second row), along with our congruence measures $\boldsymbol{C}_o$ and $\boldsymbol{C}_{ac}$ evaluated for runs relying on real crowd oracles $\mathcal{O}_C$. The dotted black line represents the result of PL. Depictions in **b** represent the class-wise relative amount of points sampled in each iteration step evaluated based on the GT labels

classes, it will tend to choose points it is currently most uncertain about near the decision borders, which often correspond to class borders as well (cf. Sect. 3.2). Thus, the reduced accuracy in AL iteration steps is due to dealing with points that are more complex for crowdworkers to label. Apart from this observation, all accuracy curves follow a decreasing trend. But at the same time, the more advanced the iteration, the better the performance of our model, as border cases of previous iteration steps can now be solved successfully. However, this also means that cases where the machine is uncertain become gradually more demanding, i.e., it focuses on more and more special edge cases which are complex for interpretation not only for the machine but also for the crowd.

As hoped for, the *RIU* technique (with d_{RIU} = 1.5 m) effectively helps to ease labeling for crowdworkers by focusing on points that are related to, but easier to interpret than the point originally selected by the machine. A more detailed analysis of the crowd performance both with and without support from *RIU* is given in Fig. 16 for the H3D data set. *RIU* is indeed capable of improving the OA of a training set generated within the AL iteration by about 4 pp, which is caused by presenting points to the crowd that are easier to label. This technique allows to resolve label ambiguities between adjacent classes, such as *Low Vegetation* vs. *Impervious Surface*, *Roof* vs. *Façade*, *Impervious Surface* vs. *Façade*, as can be seen from Fig. 16. Although the rate of confusion of class *Urban Furniture* with the other classes can be reduced, the highest frequency of confusion can still be observed with respect to this class due to the high intra-class variability. This issue cannot be solved by simply increasing the distance to the class border when the class affiliation of the object under consideration is questionable as a whole.

Apart from the accuracy of the human component of the system, also the corresponding time effort should be considered, especially as it ultimately determines the required time of a complete run of the hybrid intelligence system. This is due to a negligible processing time of the machine part, i.e., the classifier, provided that we rely on the *RF* model (cf. Sect. 5.1). As for the time required for the labeling step of a crowd-driven training cycle/iteration step of our system, this can be specified as less than 11 hours. Thus, a whole AL run including initialization is completed in about 5 days (10 iteration steps · approx. 11 h + approx. 16 h for initialization = 126 h).

5.2.4 The Performance of the Machine

Considering the machine part of the intelligence system, i.e., the ML model, its performance is depicted in Figs. 14a and b and 15a for our different data sets. As for the runs with a simulated omniscient oracle $\mathcal{O}_O$ (as baseline solution), AL runs of all data sets show the typical convergence behavior and approximate the PL baseline solution. When adding the *RIU* technique (i.e., $AL_{RIU}(\mathcal{O}_O)$), respective runs tend to perform even better. This is because increasing distance from the class border can correspond to selecting a more generalized batch rather than only focusing on most informative and thus most demanding instances. Although we

Table 5 Comparison of accuracies reached for the V3D, H3D and S3D data set for PL and various AL approaches after 10 iteration steps using different oracle types and sampling functions

Method	Oracle	F1-score [%]							[%]	
		U. Furn.	Car	L. Veg.	I. Surf.	Roof	Façade	Veg.	**mF1**	**OA**
V3D										
PL	–	–	65.64	82.25	91.28	94.81	62.30	86.24	80.42	88.11
AL	$\mathcal{O}_O$	–	66.00	79.35	90.70	**93.14**	**57.86**	**83.06**	**78.35**	**85.89**
	$\mathcal{O}_O + RIU$	–	67.80	80.67	91.13	91.12	55.22	81.83	77.96	85.29
	$\mathcal{O}_C$	–	66.05	80.11	90.71	91.25	49.87	82.91	76.82	85.03
	$\mathcal{O}_C + RIU$	–	**68.26**	**81.10**	**91.24**	88.76	45.39	82.14	76.15	84.19
H3D										
PL	–	41.14	51.63	90.68	85.26	92.96	83.77	93.05	76.92	88.16
AL	$\mathcal{O}_O$	33.93	56.34	**90.31**	82.70	88.33	79.73	**92.66**	74.86	**86.65**
	$\mathcal{O}_O + RIU$	**36.97**	55.42	89.91	**83.84**	**90.05**	79.61	91.91	**75.39**	86.60
	$\mathcal{O}_C$	33.37	**57.40**	88.34	78.14	88.89	**79.83**	92.07	74.01	85.15
	$\mathcal{O}_C + RIU$	34.56	56.20	89.59	81.81	89.18	79.35	92.56	74.75	86.26
S3D										
PL	–	75.30		98.63		96.82		93.97	91.18	95.51
AL	$\mathcal{O}_O$	62.06		98.18		94.66		91.69	86.65	93.56
	$\mathcal{O}_O + RIU$	**66.03**		**98.32**		95.22		92.40	87.99	**94.09**
	$\mathcal{O}_C$	54.68		97.62		92.34		92.15	84.20	92.18
	$\mathcal{O}_C + RIU$	61.48		97.67		**95.53**		**92.84**	86.88	93.86

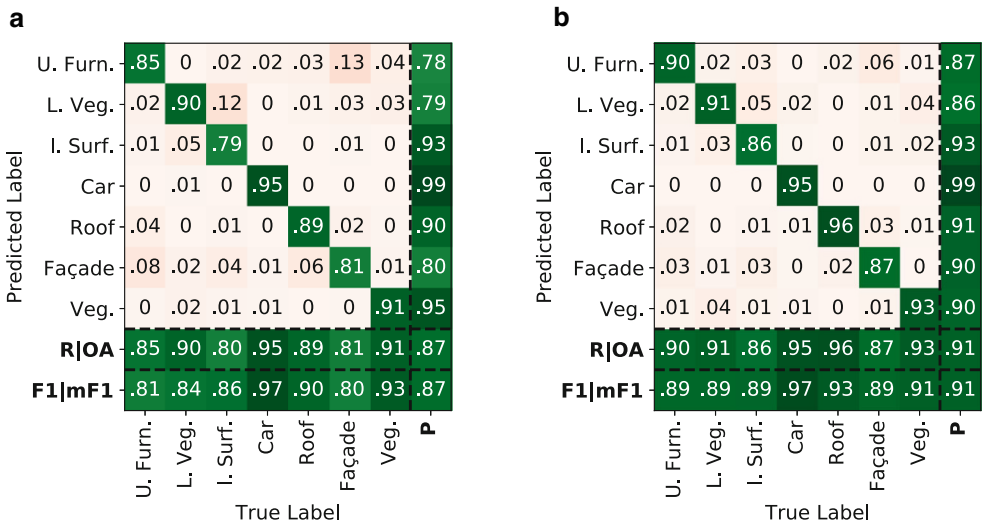

Fig. 16 Obtained confusion matrices over all iteration steps providing the crowd either with the pure choice of the classifier or applying the *RIU* technique (cf. Sect. 3.3.3). **a** $\mathcal{O}_{CM}$, **b** $\mathcal{O}_{CM} + RIU$

need to resolve exactly such occurrences, in early iteration steps, performance can often be boosted when relying on more typical samples for different classes. In addition, in the first and second iteration steps, the selection of points in $AL_{RIU}(\mathcal{O}_O)$ is slightly more related to an equal class distribution than in $AL(\mathcal{O}_O)$ (cf. Fig. 14c) possibly being beneficial for the ML model. While mF1-scores of runs both with and without *RIU* differ only marginally (especially towards the end of the iteration), the gap between increases when replacing the simulated oracle $\mathcal{O}_O$ with the real crowd oracle $\mathcal{O}_C$ in case of all data sets due to the higher error level of labeled data in the variants without *RIU*, i.e. $AL(\mathcal{O}_C)$ (cf. Figs. 14a and b and 15a). In this respect, the *RIU* technique not only increases the labeling accuracy of the crowd, but also the performance of the RF classifier learning from the crowd, and is thus capable of positively impacting the course of the iteration. However, in case of the H3D data set the influence is lowest thanks to the general high interpretability of this high resolution data set.

From the selection of samples in each iteration step in Figs. 14c and d and 15b, especially from the corresponding loops with omniscient oracles $\mathcal{O}_O$, we learn that *RIU* can have a real effect on the course of the iteration. Initially,

RF models of each run learn from the same initialization data set, so early selections differ only slightly, but deviate gradually more with the number of iteration steps due to the altered sample selection. Furthermore, an overall trend of sampling batches of points more related to an equal class distribution becomes obvious for the V3D and H3D data set. Please note that this is not necessarily a desired property. For instance, if the model is generally confused about a specific class with high intra-class variability, it would be desirable to focus sampling on only this specific class, but where selected samples cover the whole bandwidth of this class in feature space, i.e., samples are drawn from different *DiFS* clusters (cf. Sect. 3.3.2). This is also the reason why samples in batches of the large-scale S3D data set are not approximately equally distributed (at least not in the first 10 iteration steps), as we are confronted with a plethora of different representatives for each of the rather generalized classes in a data set spanning such a vast area. In the long run, however, it is for sure beneficial to provide the classifier with a training data set that evenly covers all classes, as our *weighted* sampling strategies intend to accomplish (cf. Sect. 3.3.1).

However, the question remains how these crowd-powered runs fare in comparison to our baseline solutions of PL, which is depicted in Figs. 14a and b and 15a, respectively, and is also contrasted numerically to our AL runs after $n_i = 10$ iteration steps in Table 5. With respect to achievable OAs, the method we advocate for, AL with the *RIU* technique relying on a real crowd oracle ($AL_{RIU}(\mathcal{O}_C)$) completely excluding an expert annotator, is only 3.92 pp (V3D), 1.90 pp (H3D) and 1.65 pp (S3D) below the performance of the respective PL baselines. As discussed in Sect. 5.1, classes with great intra-class variety such as *Urban Furniture* and *Façade* (including façade furniture) suffer the most from focusing only on a small AL training data set, but at the same time underrepresented classes (such as class *Car*) profit from AL sampling. It is worth emphasizing that these results are achieved by only labeling a small fraction of available training points of 4.7 ‰ (V3D), 0.1 ‰ (H3D) and 0.1 ‰ (S3D) (the absolute amount of labeled training points for each data set is 300 points · 10 it. steps + initialization points). Thus, these results come at a labeling cost of \$190 (100 jobs · \$0.10 + 100 jobs · 3 *rep.* · \$0.15 + $[n^+/10$ pts per job$]$ · 3 *rep.* · 10 it. steps · \$0.15) plus a 10% *MW* fee.

5.2.5 Terminating the Loop

While the accuracy values discussed before refer to the final classification performance after a fixed number of $n_i = 10$ iteration steps, we anticipate that a similar level of accuracy can be achieved with less iteration steps and thus less label effort. In other words, we aim to achieve a model that performs well with a minimum number of required iteration steps, i.e., for efficiency reasons, an accuracy curve reaching a stable, high-level fast is preferred to one performing slightly better after a larger number of iteration steps (assuming the same batch size n^+). We strive to identify the state of iteration where more label effort would only marginally improve model performance by means of our congruence values (cf. Sect. 3.4) evaluated for our crowd-based runs. We compute both the overall congruence C_o and the class-wise congruence C_{ac} between the current and the previous iteration step, i.e., $d_{stop} = 1$ (cf. Sect. 3.4). We succeed in describing the progress of the iteration if we are capable of computing curves that behave as similar as possible to the accuracy graphs, but without relying on GT data not available in real-world applications.

In case of V3D (cf. Fig. 14a), this seems to be true. After a congruence value of 0 in the first iteration step (due to the lack of a model prediction from a previous step), congruence values correspond well to accuracy curves. Whenever there is an upward trend in accuracy, the congruence values also increase (e.g., consider congruence measures of $AL(\mathcal{O}_C)$; *dark blue* and *violet* curve in Fig. 14a). Vice versa, flattening of accuracy curves also corresponds to flattening of congruence graphs towards the end of the iteration. In this regard $AL(\mathcal{O}_C)$ and $AL_{RIU}(\mathcal{O}_C)$ are suitable examples. While $AL(\mathcal{O}_C)$ shows a linearly increasing behavior, the latter flattens from the third iteration step on. In the congruence curves, we observe the same effects as well both with an inherent delay of one iteration step due to the required comparison with the prediction of the previous step. When the standard deviation of C_{ac} congruence values is computed, e.g., over the last $n_{stop} = 5$ iteration steps, we achieve a value of 1.4% for $AL(\mathcal{O}_C)$ vs. 0.5% for $AL_{RIU}(\mathcal{O}_C)$ at the tenth iteration step. Thus, depending on the stopping threshold, and as desired, $AL_{RIU}(\mathcal{O}_C)$ would be terminated earlier than $AL(\mathcal{O}_C)$. But of course, the termination of the loop also depends on the number of iteration steps n_{stop} which support the calculation of the standard deviation. This was chosen gently, i.e. in such a way that stopping too early is avoided, to ensure that a stable plateau has been identified.

For the H3D data set (cf. Fig. 14b), again congruence curves of $AL(\mathcal{O}_C)$ and $AL_{RIU}(\mathcal{O}_C)$ follow the steady linear increase of mF1 curves and resemble congruence values of $AL(\mathcal{O}_C)$ in V3D. But in contrast to V3D, a clear drop in congruence is observable for H3D at the second iteration step. This is not necessarily a bad omen for the progress of the iteration, as it only indicates a significant change in predicted labels at an unstable stage of the training. This is due to adding new samples that have significantly altered the current belief about decision borders. Fig. 14d underlines this hypothesis. In the first iteration step, mainly samples of class *Façade* are queried, which supposedly leads to an

improvement in the recognition of representatives of this class and probably also implicitly improves accuracies for adjacent classes, which might now be better distinguishable. However, the second iteration step is the first time a great variety of classes within the batch are sampled. Therefore, the RF model is able to improve its overall classification capabilities with respect to a greater bandwidth of classes, thus predicting significantly different. Remember that this effect becomes visible in congruence curves with a delay of one iteration step, explaining the strong discrepancy between the two successive predictions.

A similar behavior, showing the fidelity of the congruence curves with respect to accuracy curves (and thus suitability for defining a stopping criterion) can be spotted for the S3D data set (cf. Fig. 15a). Please note that the characteristic up-and-down bending of the $AL(\mathcal{O}_C)$ mF1 curve between iteration steps 2 and 4 is also translated to the corresponding $C_{ac}(AL)$ curve (again with a delay of one iteration step), where the first upward trend is triggered by the inclusion of a considerable number of high intra-class variability *Building* samples in the third iteration step (cf. Fig. 15b).

6 Conclusion & Outlook

The motivation driving this work was to generate a framework that is capable of teaching a supervised ML system to automatically enrich an arbitrary point cloud with semantic information by only providing it with the respective cloud as well as access to a (fixed) labeling budget. This was achieved by setting up a hybrid intelligence system with an AL backbone in which both an ML model and human beings work together, with the latter being considered as *human processing units* in reference to *electronic processing units* typically encountered in automated systems. When paid crowdworkers take this role, there are still humans working in the AL loop, but from an operator's perspective, we are dealing with a fully automated system because the work of crowdworkers can be considered a non-deterministic subroutine of our program for timely returning labels, but behaving just like other routines that are accomplished by *electronic processing units* (such as training our RF or SCN classifier). This concept was applied to a variety of three ALS point clouds representing spatially distinct areas (with two of them being state-of-the-art benchmark data sets) with different characteristics. For those, we achieved accuracies at a quality level similar to that of PL by annotating only few most-informative training points out of the complete set of potentially available training data (typically $\ll 1\%$), thus causing minimal monetary expenses. Timewise, with an average annotation time of about 40 s per crowd annotated point, we oppose an overall time effort of about 140 h (40 s·[9 classes·100 points+9 classes·100 points·3 crowdworkers + 10 it. steps·300 points · 3 crowdworkers]) for our crowd-based pipeline to an overall expert labeling time of 1490 h for a full annotation (numbers are valid for the H3D data set as this is the only one the authors are aware of the required labeling effort).

To summarize, we consider our approach to be efficient in situations with a limited labeling budget. Naturally, by restricting the training set to only few most-informative points, our models trained on these samples achieve slightly worse accuracy scores compared to the PL approach due to not being able to represent each and every aspect, especially of diverse classes, to their full extent. However, we believe that in times of rapid data acquisition cycles, labeling effort has to be focused on most-decisive points only, in favor of reasonable processing throughput for semantic data interpretation. Nevertheless, our pipeline is flexible enough to allow an operator to individually decide how much resources to spend on a specific data set, while still providing a recommendation of cost vs. benefit by means of our stopping criterion. Hence, the limiting factor with respect to accuracy of semantic segmentation (given a feasible class catalog) is available resources rather than the crowd's interpretation capability as our experiments have shown that the crowd oracle behaves very similar to an omniscient oracle. Still, with a better performing crowd a desired accuracy level can be reached earlier in the iteration (with less money spent), which can, for instance, be achieved by providing a data modality that is easier to comprehend for crowdworkers (Kölle et al. 2021b).

With respect to further optimizing the proposed methodology, we suggest to follow a promising approach from the weakly supervised domain that was recently adapted by Lin et al. (2022) for the automatic interpretation of ALS point clouds. In this concept, an ML model is enabled to derive point-wise predictions although only learning from so-called scene-level weak labels for point cloud subsets indicating *that* there are representatives of a specific class included but not *which* point(s) actually belong to this class. Thus, when our crowdworkers are required to only provide such scene-level labels, from our point of view, this offers the potential to drastically ease label complexity.

However, the main drawback we see in crowdsourced labeling is that the crowd is unable to annotate points according to an arbitrarily complex and fine-grained class catalog (also identified by Bayas et al. (2016)), why we actually had to simplify V3D's and H3D's class catalog. According to our findings, the crowd performs well for classes that are easy and straightforward to understand from the pure class names, but fails whenever classes are non-intuitive or are rather concerned with a detailed and maybe even subjective description (e.g., *Shrub* vs. *Hedge*). Furthermore, unspecific classes such as *Urban Furniture* are often misused as quasi-

class *Other* whenever crowdworkers are unsure about the class affiliation of a specific point. Thus, it remains an open question *if* and *how* complex class catalogs could be taught to the crowd.

Despite these limitations, we have succeeded in formulating a fully automated hybrid intelligence system based on AL with subprocesses being carried out by inherently non-deterministic *human processing units* (i.e., crowdworkers) and that does not involve an expert in any labeling task. By conducting this work, we hope to have paved the way for a wider acceptance and dissemination of hybrid intelligence systems relying on (paid) crowdsourcing in the field of geospatial data analysis.

Funding Open Access funding enabled and organized by Projekt DEAL.

Conflict of interest The authors declare that they have no conflict of interest.

References

Actueel Hoogtebestand Nederland (2021) Dataset: Actueel Hoogtebestand Nederland (AHN3) [WWW Document]. URL: https://www.pdok.nl/introductie/-/article/actueel-hoogtebestand-nederland-ahn3- (accessed February 2, 2021)

von Ahn L, Maurer B, McMillen C, Abraham D, Blum M (2008) reCAPTCHA: Human-based character recognition via web security measures. Science 321(5895):1465–1468, https://doi.org/10.1126/science.1160379

Allahbakhsh M, Benatallah B, Ignjatovic A, Motahari-Nezhad HR, Bertino E, Dustdar S (2013) Quality control in crowdsourcing systems: Issues and directions. IEEE Internet Computing 17(2):76–81, https://doi.org/10.1109/mic.2013.20

Antoniou V, Skopeliti A (2015) Measures and indicators of VGI quality: An overview. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences II-3/W5:345–351, https://doi.org/10.5194/isprsannals-II-3-W5-345-2015, https://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/II-3-W5/345/2015/

Argamon-Engelson S, Dagan I (1999) Committee-Based Sample Selection For Probabilistic Classifiers. Journal of Artificial Intelligence Research 11:335–360

Ash JT, Zhang C, Krishnamurthy A, Langford J, Agarwal A (2019) Deep batch active learning by diverse, uncertain gradient lower bounds. CoRR abs/1906.03671, https://doi.org/10.48550/ARXIV.1906.03671

Bayas JL, See L, Fritz S, Sturn T, Perger C, Dürauer M, Karner M, Moorthy I, Schepaschenko D, Domian D, McCallum I (2016) Crowdsourcing in-situ data on land cover and land use using gamification and mobile technology. Remote Sensing 8(11):905, https://doi.org/10.3390/rs8110905

Beluch WH, Genewein T, Nurnberger A, Kohler JM (2018) The power of ensembles for active learning in image classification. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, https://doi.org/10.1109/cvpr.2018.00976

Bezos J (2007) Artificial Intelligence, With Help From the Humans [WWW Document]. https://www.nytimes.com/2007/03/25/business/yourmoney/25Stream.html, (accessed February 18, 2022)

Bloodgood M, Vijay-Shanker K (2009) A method for stopping active learning based on stabilizing predictions and the need for user-adjustable stopping. In: Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009), Association for Computational Linguistics, Boulder, Colorado, pp 39–47, https://www.aclweb.org/anthology/W09-1107

Chandler D, Kapelner A (2013) Breaking monotony with meaning: Motivation in crowdsourcing markets. Journal of Economic Behavior & Organization 90:123–133, https://doi.org/10.1016/j.jebo.2013.03.003

Chandler JJ, Paolacci G (2017) Lie for a dime. Social Psychological and Personality Science 8(5):500–508, https://doi.org/10.1177/1948550617698203

Chandler J, Paolacci G, Mueller P (2013) Risks and rewards of crowdsourcing marketplaces. In: Handbook of Human Computation, Springer New York, pp 377–392, https://doi.org/10.1007/978-1-4614-8806-4_30

Cortes C, Vapnik V (1995) Support-vector networks. Machine Learning 20(3):273–297

Crawford MM, Tuia D, Yang HL (2013) Active learning: Any value for classification of remotely sensed data? Proceedings of the IEEE 101(3):593–608, https://doi.org/10.1109/jproc.2012.2231951

Dasgupta S, Hsu D (2008) Hierarchical sampling for active learning. In: Proceedings of the 25th international conference on Machine learning – ICML '08, ACM Press, https://doi.org/10.1145/1390156.1390183

Deng J, Dong W, Socher R, Li LJ, Li K, Li FF (2009) ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR 2009, pp 248–255, https://doi.org/10.1109/CVPR.2009.5206848

van Dijk TC, Fischer N, Häussner B (2020) Algorithmic improvement of crowdsourced data. In: Proceedings of the 28th International Conference on Advances in Geographic Information Systems, ACM, https://doi.org/10.1145/3397536.3422260

Dorn H, Törnros T, Zipf A (2015) Quality evaluation of VGI using authoritative data—a comparison with land use data in southern germany. ISPRS International Journal of Geo-Information 4(3):1657–1671, https://doi.org/10.3390/ijgi4031657

Endres I, Farhadi A, Hoiem D, Forsyth DA (2010) The benefits and challenges of collecting richer object annotations. 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition – Workshops pp 1–8

Ertekin S, Huang J, Bottou L, Giles L (2007) Learning on the Border: Active Learning in Imbalanced Data Classification. In: CIKM 2007, ACM, New York, NY, USA, pp 127–136, https://doi.org/10.1145/1321440.1321461, http://doi.acm.org/10.1145/1321440.1321461

Estes L, McRitchie D, Choi J, Debats S, Evans T, Guthe W, Luo D, Ragazzo G, Zempleni R, Caylor K (2016) A platform for crowdsourcing the creation of representative, accurate landcover maps. Environmental Modelling & Software 80:41–53, https://doi.org/10.1016/j.envsoft.2016.01.011

Fan H, Zipf A, Fu Q, Neis P (2014) Quality assessment for building footprints data on OpenStreetMap. International Journal of Geo-

graphical Information Science 28(4):700–719, https://doi.org/10.1080/13658816.2013.867495

Feng D, Wei X, Rosenbaum L, Maki A, Dietmayer K (2019) Deep active learning for efficient training of a LiDAR 3d object detector. In: 2019 IEEE Intelligent Vehicles Symposium (IV), IEEE, https://doi.org/10.1109/ivs.2019.8814236

Fleischer A, Mead AD, Huang J (2015) Inattentive responding in MTurk and other online samples. Industrial and Organizational Psychology 8(2):196–202, https://doi.org/10.1017/iop.2015.25

Fonte C, Antoniou V, Bastin L, Estima J, Jokar Arsanjani J, Laso Bayas J, See L, Vatseva R (2017) Assessing VGI Data Quality, Ubiquity Press, pp 137–163. https://doi.org/10.5334/bbf.g

Gal Y, Ghahramani Z (2016) Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning. In: ICML 2016, PMLR, New York, NY, USA, vol 48, pp 1050–1059, http://proceedings.mlr.press/v48/gal16.html

Gal Y, Islam R, Ghahramani Z (2017) Deep bayesian active learning with image data. In: Proceedings of the 34th International Conference on Machine Learning – Volume 70, JMLR.org, ICML'17, p 1183–1192

Galton F (1907) Vox populi. Nature 75(1949):450–451, https://doi.org/10.1038/075450a0

Gebru T, Krause J, Deng J, Fei-Fei L (2017) Scalable annotation of fine-grained categories without experts. Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems

Geiger D, Seedorf S, Schulze T, Nickerson RC, Schader M (2011) Managing the crowd: Towards a taxonomy of crowdsourcing processes. In: AMCIS

Gingold Y, Shamir A, Cohen-Or D (2012) Micro perceptual human computation for visual tasks. ACM Transactions on Graphics 31(5):1–12, https://doi.org/10.1145/2231816.2231817

Goodchild MF (2007) Citizens as sensors: the world of volunteered geography. GeoJournal 69(4):211–221, https://doi.org/10.1007/s10708-007-9111-y

Goodchild MF, Li L (2012) Assuring the quality of volunteered geographic information. Spatial Statistics 1:110–120, https://doi.org/10.1016/j.spasta.2012.03.002

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, https://doi.org/10.5194/isprs-annals-V-2-2020-727-2020

Haklay M, Weber P (2008) OpenStreetMap: User-generated street maps. IEEE Pervasive Computing 7(4):12–18, https://doi.org/10.1109/mprv.2008.80

Hara K, Azenkot S, Campbell M, Bennett CL, Le V, Pannella S, Moore R, Minckler K, Ng RH, Froehlich JE (2015) Improving public transit accessibility for blind riders by crowdsourcing bus stop landmark locations with google street view: An extended analysis. ACM Transactions on Accessible Computing 6(2):1–23, https://doi.org/10.1145/2717513

Haralabopoulos G, Wagner C, McAuley D, Anagnostopoulos I (2019) Paid crowdsourcing, low income contributors, and subjectivity. In: IFIP Advances in Information and Communication Technology, Springer International Publishing, pp 225–231, https://doi.org/10.1007/978-3-030-19909-8_20

Hashemi P, Abbaspour RA (2015) Assessment of logical consistency in OpenStreetMap based on the spatial similarity concept. In: Lecture Notes in Geoinformation and Cartography, Springer International Publishing, pp 19–36, https://doi.org/10.1007/978-3-319-14280-7_2

Hecht R, Kalla M, Krüger T (2018) Crowd-sourced data collection to support automatic classification of building footprint data. Proceedings of the ICA 1:1–7, https://doi.org/10.5194/ica-proc-1-54-2018

Herfort B, Höfle B, Klonner C (2018) 3D micro-mapping: Towards assessing the quality of crowdsourcing to support 3D point cloud analysis. ISPRS Journal of Photogrammetry and Remote Sensing 137:73–83, https://doi.org/10.1016/j.isprsjprs.2018.01.009

Hirth M, Hoßfeld T, Tran-Gia P (2011) Anatomy of a Crowdsourcing Platform – Using the Example of Microworkers.com. In: IMIS 2011, IEEE Computer Society, Washington, DC, USA, pp 322–329, https://doi.org/10.1109/IMIS.2011.89, https://doi.org/10.1109/IMIS.2011.89

Hirth M, Hoßfeld T, Tran-Gia P (2013) Analyzing costs and accuracy of validation mechanisms for crowdsourcing platforms. Mathematical and Computer Modelling 57(11-12):2918–2932, https://doi.org/10.1016/j.mcm.2012.01.006

Houlsby N, Huszár F, Ghahramani Z, Lengyel M (2011) Bayesian active learning for classification and preference learning. https://doi.org/10.48550/ARXIV.1112.5745

Howe J (2006) The rise of crowdsourcing. Wired Magazine 6(14):1–4

Hui Z, Jin S, Cheng P, Ziggah YY, 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 LV, 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, https://doi.org/10.1109/mci.2022.3155327

Juni MZ, Eckstein MP (2017) The wisdom of crowds for visual search. Proceedings of the National Academy of Sciences 114(21):E4306–E4315, https://doi.org/10.1073/pnas.1610732114

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. TRGS 57(12):9524–9533, https://doi.org/10.1109/TGRS.2019.2927393, https://doi.org/10.1109/TGRS.2019.2927393

Kirsch A, van Amersfoort J, Gal Y (2019) BatchBALD: Efficient and Diverse Batch Acquisition for Deep Bayesian Active Learning. In: NIPS 2019, Curran Associates, Inc., pp 7026–7037

Kittur A, Chi EH, Suh B (2008) Crowdsourcing user studies with mechanical turk. In: Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems – CHI '08, ACM Press, https://doi.org/10.1145/1357054.1357127

Koelle M, Walter V, Schmohl S, Soergel U (2023) Learning on the edge: Benchmarking active learning for the semantic segmentation of als point clouds. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences X-1/W1-2023:945–952, https://doi.org/10.5194/isprs-annals-X-1-W1-2023-945-2023, https://isprs-annals.copernicus.org/articles/X-1-W1-2023/945/2023/

Kölle M, Laupheimer D, Schmohl S, Haala N, Rottensteiner F, Wegner JD, 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, https://doi.org/10.1016/j.ophoto.2021.100001

Kölle M, Laupheimer D, Walter V, Haala N, Soergel U (2021b) Which 3D data representation does the crowd like best? crowd-based active learning for coupled semantic segmentation of point clouds and textured meshes. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences V-2-2021:93–100, https://doi.org/10.5194/isprs-annals-V-2-2021-93-2021, https://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/V-2-2021/93/2021/

Kölle M, Walter V, Schmohl S, Soergel U (2021a) Remembering both the machine and the crowd when sampling points: Active learning for semantic segmentation of ALS point clouds. In: ICPR Inter-

national Workshops and Challenges, Springer International Publishing, Cham, pp 505–520

Kölle M, Walter V, Shiller I, Soergel U (2021b) Categorise: An automated framework for utilizing the workforce of the crowd for semantic segmentation of 3D point clouds. In: ICPR International Workshops and Challenges, Springer International Publishing, Cham, pp 505–520

Korpela E, Werthimer D, Anderson D, Cobb J, Leboisky M (2001) Seti@home-massively distributed computing for seti. Computing in Science Engineering 3(1):78–83, https://doi.org/10.1109/5992.895191

Kovashka A, Russakovsky O, Fei-Fei L, Grauman K (2016) Crowdsourcing in Computer Vision. Foundations and Trends in Computer Graphics and Vision 10(3):177–243, https://doi.org/10.1561/0600000071

Lewis DD, Gale WA (1994) A sequential algorithm for training text classifiers. In: SIGIR '94, Springer London, pp 3–12, https://doi.org/10.1007/978-1-4471-2099-5_1

Li H, Zipf A (2022) A conceptual model for converting openstreetmap contribution to geospatial machine learning training data. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLIII-B4-2022:253–259, https://doi.org/10.5194/isprs-archives-xliii-b4-2022-253-2022

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 MY (2020a) Active and incremental learning for semantic ALS point cloud segmentation. ISPRS Journal of Photogrammetry and Remote Sensing 169:73–92, https://doi.org/10.1016/j.isprsjprs.2020.09.003

Lin Y, Vosselman G, Cao Y, Yang MY (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, https://doi.org/10.5194/isprs-annals-V-2-2020-243-2020, https://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/V-2-2020/243/2020/

Lin Y, Vosselman G, Yang MY (2022) Weakly supervised semantic segmentation of airborne laser scanning point clouds. ISPRS Journal of Photogrammetry and Remote Sensing 187:79–100, https://doi.org/10.1016/j.isprsjprs.2022.03.001

Lintott CJ, Schawinski K, Slosar A, Land K, Bamford S, Thomas D, Raddick MJ, Nichol RC, Szalay A, Andreescu D, Murray P, Vandenberg J (2008) Galaxy zoo: morphologies derived from visual inspection of galaxies from the sloan digital sky survey. Monthly Notices of the Royal Astronomical Society 389(3):1179–1189, https://doi.org/10.1111/j.1365-2966.2008.13689.x, https://doi.org/10.1111/j.1365-2966.2008.13689.x

Liu Z, Shabani S, Balet NG, Sokhn M, Cretton F (2018) How to motivate participation and improve quality of crowdsourcing when building accessibility maps. In: 2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC), IEEE, https://doi.org/10.1109/ccnc.2018.8319237

Lloyd SP (1982) Least squares quantization in PCM. IEEE Trans Inf Theory 28(2):129–137

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, 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. TGRS 56(7):3631–3644

van der Maaten L, Hinton G (2008) Visualizing data using t-SNE. Journal of Machine Learning Research 9:2579–2605, http://www.jmlr.org/papers/v9/vandermaaten08a.html

Mackowiak R, Lenz P, Ghori O, Diego F, Lange O, Rother C (2018) CEREALS - Cost-Effective REgion-based Active Learning for Semantic Segmentation. BMVC 2018 http://arxiv.org/abs/1810.09726, 1810.09726

Maddalena E, Ibáñez LD, Simperl E (2020) Mapping points of interest through street view imagery and paid crowdsourcing. ACM Transactions on Intelligent Systems and Technology 11(5):1–28, https://doi.org/10.1145/3403931

Mao A, Kamar E, Chen Y, Horvitz E, Schwamb ME, Lintott CJ, Smith AM (2013) Volunteering versus work for pay: Incentives and tradeoffs in crowdsourcing. In: In Proceedings of the First AAAI Conference on Human Computation and Crowdsourcing (HCOMP '13

Marcus A, Parameswaran A (2015) Crowdsourced data management: Industry and academic perspectives. Foundations and Trends in Databases 6(1-2):1–161, https://doi.org/10.1561/1900000044

McCallum A, Nigam K (1998) Employing EM and pool-based active learning for text classification. In: Proceedings of the Fifteenth International Conference on Machine Learning, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, ICML '98, p 350–358

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, https://doi.org/10.1016/j.isprsjprs.2013.11.001

Okolloh O (2009) Ushahidi, or 'testimony': Web 2.0 tools for crowdsourcing crisis information. Participatory Learning and Action 59:65–70

Olsson F, Tomanek K (2009) An intrinsic stopping criterion for committee-based active learning. In: Proceedings of the Thirteenth Conference on Computational Natural Language Learning, Association for Computational Linguistics, USA, CoNLL '09, p 138–146

Parhami B (1994) Voting algorithms. IEEE Transactions on Reliability 43(4):617–629, https://doi.org/10.1109/24.370218

Patterson G, Xu C, Su H, Hays J (2014) The SUN attribute database: Beyond categories for deeper scene understanding. International Journal of Computer Vision 108(1-2):59–81, https://doi.org/10.1007/s11263-013-0695-z

Prabhu V, Chandrasekaran A, Saenko K, Hoffman J (2021) Active domain adaptation via clustering uncertainty-weighted embeddings. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp 8485–8494, https://doi.org/10.1109/ICCV48922.2021.00839

Redi J, Povoa I (2014) Crowdsourcing for rating image aesthetic appeal. In: Proceedings of the 2014 International ACM Workshop on Crowdsourcing for Multimedia – CrowdMM '14, ACM Press, https://doi.org/10.1145/2660114.2660118

Ren P, Xiao Y, Chang X, Huang PY, Li Z, Gupta BB, Chen X, Wang X (2022) A survey of deep active learning. ACM Computing Surveys 54(9):1–40, https://doi.org/10.1145/3472291

Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg AC, Fei-Fei L (2015a) ImageNet large scale visual recognition challenge. International Journal of Computer Vision 115(3):211–252, https://doi.org/10.1007/s11263-015-0816-y

Russakovsky O, Li LJ, Fei-Fei L (2015b) Best of both worlds: Human-machine collaboration for object annotation. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, https://doi.org/10.1109/cvpr.2015.7298824

Russell BC, Torralba A, Murphy KP, Freeman WT (2007) LabelMe: A database and web-based tool for image annotation. International Journal of Computer Vision 77(1-3):157–173, https://doi.org/10.1007/s11263-007-0090-8

Salk CF, Sturn T, See L, Fritz S, Perger C (2015) Assessing quality of volunteer crowdsourcing contributions: lessons from the cropland capture game. International Journal of Digital Earth 9(4):410–426, https://doi.org/10.1080/17538947.2015.1039609

Scheffer T, Decomain C, Wrobel S (2001) Active hidden markov models for information extraction. In: Advances in Intelligent Data Analysis, Springer Berlin Heidelberg, pp 309–318, https://doi.org/10.1007/3-540-44816-0_31

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, https://doi.org/10.5194/isprs-annals-IV-2-W5-77-2019, https://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/IV-2-W5/77/2019/

See L, Comber A, Salk C, Fritz S, van der Velde M, Perger C, Schill C, McCallum I, Kraxner F, Obersteiner M (2013) Comparing the quality of crowdsourced data contributed by expert and non-experts. PLoS ONE 8(7):e69958, https://doi.org/10.1371/journal.pone.0069958

Senaratne H, Mobasheri A, Ali AL, Capineri C, Haklay MM (2016) A review of volunteered geographic information quality assessment methods. International Journal of Geographical Information Science 31(1):139–167, https://doi.org/10.1080/13658816.2016.1189556

Sener O, Savarese S (2018) Active learning for convolutional neural networks: A core-set approach. In: International Conference on Learning Representations, https://openreview.net/forum?id=H1aIuk-RW

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

Shannon CE (1948) A mathematical theory of communication. Bell System Technical Journal 27(3):379–423, https://doi.org/10.1002/j.1538-7305.1948.tb01338.x

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

Shaw AD, Horton JJ, Chen DL (2011) Designing incentives for inexpert human raters. In: Proceedings of the ACM 2011 conference on Computer supported cooperative work – CSCW '11, ACM Press, https://doi.org/10.1145/1958824.1958865

Shi X, Xu X, Chen K, Cai L, Foo CS, Jia K (2021) Label-efficient point cloud semantic segmentation: An active learning approach. CoRR abs/2101.06931, https://doi.org/10.48550/ARXIV.2101.06931

Sinha S, Ebrahimi S, Darrell T (2019) Variational adversarial active learning. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), IEEE Computer Society, Los Alamitos, CA, USA, pp 5971–5980, https://doi.org/10.1109/ICCV.2019.00607, https://doi.ieeecomputersociety.org/10.1109/ICCV.2019.00607

Sorokin A, Forsyth D (2008) Utility data annotation with amazon mechanical turk. In: 2008 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, IEEE, https://doi.org/10.1109/cvprw.2008.4562953

Surowiecki J (2004) The Wisdom of Crowds. Anchor

Thoreau R, Achard V, Risser L, Berthelot B, Briottet X (2022) Active learning for hyperspectral image classification: A comparative review. IEEE Geoscience and Remote Sensing Magazine pp 2–24, https://doi.org/10.1109/mgrs.2022.3169947

Tuia D, Munoz-Mari J (2013) Learning user's confidence for active learning. IEEE Transactions on Geoscience and Remote Sensing 51(2):872–880, https://doi.org/10.1109/tgrs.2012.2203605

Tuia D, Volpi M, Copa L, Kanevski M, Munoz-Mari J (2011) A survey of active learning algorithms for supervised remote sensing image classification. IEEE Journal of Selected Topics in Signal Processing 5(3):606–617, https://doi.org/10.1109/jstsp.2011.2139193

Varney N, Asari VK, Graehling Q (2020) Dales: A large-scale aerial lidar data set for semantic segmentation. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp 717–726, https://doi.org/10.1109/CVPRW50498.2020.00101

Vaughan JW (2018) Making better use of the crowd: How crowdsourcing can advance machine learning research. Journal of Machine Learning Research 18(193):1–46, http://jmlr.org/papers/v18/17-234.html

Vijayanarasimhan S, Grauman K (2009) What's it going to cost you?: Predicting effort vs. informativeness for multi-label image annotations. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, https://doi.org/10.1109/cvpr.2009.5206705

Vlachos A (2008) A stopping criterion for active learning. Computer Speech & Language 22(3):295–312, https://doi.org/10.1016/j.csl.2007.12.001

Vondrick C, Patterson D, Ramanan D (2012) Efficiently scaling up crowdsourced video annotation. International Journal of Computer Vision 101(1):184–204, https://doi.org/10.1007/s11263-012-0564-1

Waldhauser C, Hochreiter R, Otepka J, Pfeifer N, Ghuffar S, Korzeniowska K, Wagner G (2014) Automated classification of airborne laser scanning point clouds. In: Solving Computationally Expensive Engineering Problems, Springer International Publishing, pp 269–292, https://doi.org/10.1007/978-3-319-08985-0_12

Walter V, Soergel U (2018) Implementation, Results, and Problems of Paid Crowd-Based Geospatial Data Collection. PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science 86:187–197

Walter V, Kölle M, Yin Y (2020) Evaluation and Optimisation of Crowd-Based Collection of Trees from 3D Point Clouds. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences V-4-2020:49–56, https://doi.org/10.5194/isprs-annals-V-4-2020-49-2020, https://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/V-4-2020/49/2020/

Walter V, Kölle M, Collmar D, Zhang Y (2021) A two-level approach for the crowd-based collection of vehicles from 3D point clouds. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences V-4-2021:97–104, https://doi.org/10.5194/isprs-annals-V-4-2021-97-2021, https://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/V-4-2021/97/2021/

Walter V, Kölle M, Collmar D (2022) Measuring the wisdom of the crowd: How many is enough? PFG – Journal of Photogrammetry, Remote Sensing and Geoinformation Science 90:269–291, https://doi.org/10.1007/s41064-022-00202-2

Welinder P, Branson S, Perona P, Belongie S (2010) The multidimensional wisdom of crowds. In: Lafferty J, Williams C, Shawe-Taylor J, Zemel R, Culotta A (eds) Advances in Neural Information Processing Systems, Curran Associates, Inc., vol 23, https://proceedings.neurips.cc/paper/2010/file/0f9cafd014db7a619ddb4276af0d692c-Paper.pdf

Whitehill J, Wu Tf, Bergsma J, Movellan J, Ruvolo P (2009) Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In: Bengio Y, Schuurmans D, Lafferty J, Williams C, Culotta A (eds) Advances in Neural Information Processing Systems, Curran Associates, Inc., vol 22, https://proceedings.neurips.cc/paper/2009/file/f899139df5e1059396431415e770c6dd-Paper.pdf

Wu TH, Liu YC, Huang YK, Lee HY, Su HT, Huang PC, Hsu WH (2021) Redal: Region-based and diversity-aware active learning for point cloud semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp 15510–15519

Ye T, You S, Robert Jr L (2017) When does more money work? examining the role of perceived fairness in pay on the performance quality of crowdworkers. Proceedings of the International AAAI Conference on Web and Social Media 11(1):327–336, https://ojs.aaai.org/index.php/ICWSM/article/view/14876

Ye Z, Xu Y, Huang R, Tong X, Li X, Liu X, Luan K, Hoegner L, Stilla U (2020) LASDU: A large-scale aerial LiDAR dataset for semantic labeling in dense urban areas. ISPRS International Journal of Geo-Information 9(7):450, https://doi.org/10.3390/ijgi9070450

Zhang J, Wu X, Sheng VS (2016) Learning from crowdsourced labeled data: a survey. Artificial Intelligence Review 46(4):543–576, https://doi.org/10.1007/s10462-016-9491-9, https://doi.org/10.1007/s10462-016-9491-9

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

Zhou B, Lapedriza A, Xiao J, Torralba A, Oliva A (2014) Learning deep features for scene recognition using places database. In: Ghahramani Z, Welling M, Cortes C, Lawrence N, Weinberger KQ (eds) Advances in Neural Information Processing Systems, Curran Associates, Inc., vol 27, https://proceedings.neurips.cc/paper/2014/file/3fe94a002317b5f9259f82690aeea4cd-Paper.pdf

Zolanvari SMI, Ruano S, Rana A, Cummins A, da Silva RE, Rahbar M, Smolic A (2019) Dublincity: Annotated lidar point cloud and its applications. CoRR abs/1909.03613, https://doi.org/10.48550/ARXIV.1909.03613, https://arxiv.org/abs/1909.03613