EFFICIENT TRAINING OF SEMANTIC POINT CLOUD SEGMENTATION VIA ACTIVE LEARNING

Y. Lin[1, *], G. Vosselman[1], Y. Cao[2], M. Y. Yang[1]

[1] Dept. of Earth Observation Science, Faculty ITC, University of Twente, Enschede, The Netherlands - (y.lin, george.vosselman, michael.yang)@utwente.nl
[2] State Key Laboratory of Fluid Power and Mechatronic Systems, School of Mechanical Engineering, Zhejiang University, Hangzhou, China - caoyp@zju.edu.cn

Commission II, WG II/3

KEY WORDS: Point Clouds, Active Learning, Deep Learning, Semantic Segmentation

ABSTRACT:

With the development of LiDAR and photogrammetric techniques, more and more point clouds are available with high density and in large areas. Point cloud interpretation is an important step before many real applications like 3D city modelling. Many supervised machine learning techniques have been adapted to semantic point cloud segmentation, aiming to automatically label point clouds. Current deep learning methods have shown their potentials to produce high accuracy in semantic point cloud segmentation tasks. However, these supervised methods require a large amount of labelled data for proper model performance and good generalization. In practice, manual labelling of point clouds is very expensive and time-consuming. Active learning can iteratively select unlabelled samples for manual annotation based on current statistical models and then update the labelled data pool for next model training. In order to effectively label point clouds, we proposed a segment based active learning strategy to assess the informativeness of samples. Here, the proposed strategy uses 40% of the whole training dataset to achieve a mean IoU of 75.2% which is 99.1% of the accuracy in mIoU obtained from the model trained on the full dataset, while the baseline method using same amount of data only reaches 69.6% in mIoU corresponding to 90.9% of the accuracy in mIoU obtained from the model trained on the full dataset.

1. INTRODUCTION

Nowadays, detailed 3D city models are required in many disciplines, like land administration (Lemmen et al., 2015), urban planning (Murgante et al., 2009) and tourism (Cooper et al., 2013). They are supposed to give various information in complex urban environments, like the number of buildings and trees and the size of buildings, roads and vegetation coverage. Point clouds are an essential type of data to generate detailed 3D city models. However, manual labelling of point clouds in urban areas requires huge efforts. Therefore, machine learning techniques have been investigated to solve this semantic segmentation problem automatically.

In machine learning based approaches, labelled datasets are required to train statistical models. Examples of supervised learning models are random forest (Breiman, 2001), support vector machine (Cortes & Vapnik, 1995) and Adaboost (Hastie et al., 2009). More recent techniques are deep neural networks that have outstanding performance in many 2D classification and recognition tasks like AlexNet (Krizhevsky et al., 2012) and ResNet (He et al., 2015). Then deep learning based methods extend to point cloud processing like Kd-Networks (Klokov and Lempitsky, 2017), PointNet (Qi et al., 2017a) and PointCNN (Li et al., 2018). In order to get accurate predictions, datasets should be large enough to avoid overfitting, especially when using deep learning based algorithms that learn features from training data. Larger labelled datasets for training help statistical models generalize to more data. However, annotation of 3D point clouds is tedious and time-consuming. It takes over 2500 hours to manually label 260 million points into 8 classes in urban areas (Zolanvari et al., 2019). Therefore, it is necessary to develop methods to reduce this manual [1] work. In most of current researches, all samples in training datasets are treated equally and are fed into classifiers with random shuffling. However, the informativeness of these training samples differs. Some bring more information and give more contributions to the model performance, while some are less informative and even introduce noisy information to models (Settles, 2009). Thus, a more efficient learning strategy is required to optimize models with most informative samples and labelling efforts only need to be put on these informative samples.

Active learning strategies are developed to minimize manually labelling efforts while maximizing the model performance in a supervised learning process. The strategy is to evaluate the informativeness of unlabelled data by a model, label those informative samples by human annotators and then add the newly labelled samples to the current training data for the next training. Active learning has been applied to many disciplines like object detection (Sivaraman & Trivedi, 2014), semantic segmentation (Vezhnevets et al., 2012), image classification (Wang et al., 2017) and natural language processing (Wang et al., 2019). However, very few studies investigate how to apply active learning strategies to point cloud labelling tasks (Feng et al., 2019; Luo et al., 2018).

In this paper, we propose an active learning strategy for semantic segmentation of large-scale ALS point clouds. The main objective is to effectively select point cloud samples for network training and therefore, reduce the manual labelling work but maintain the model performance. Figure 1 demonstrates the framework in this paper. We estimate data informativeness by uncertainty. Experiments show that the active learning strategy using entropy within segments as the criteria can select the most informative data that improve model performance. The major contributions of this paper are as follows: 1) We propose an active learning framework to efficiently label point clouds based on a deep learning network,

* Corresponding author

PointNet++. To the best of our knowledge, our paper is the first one to combine active learning with deep learning for semantic segmentation of 3D point clouds. 2) Instead of simply assessing pointwise uncertainty using entropy, we consider interactions among points within segments. 3) The proposed strategy uses 40% of the whole dataset to achieve 99.1% of the accuracy in mIoU obtained from the model trained on the full dataset.

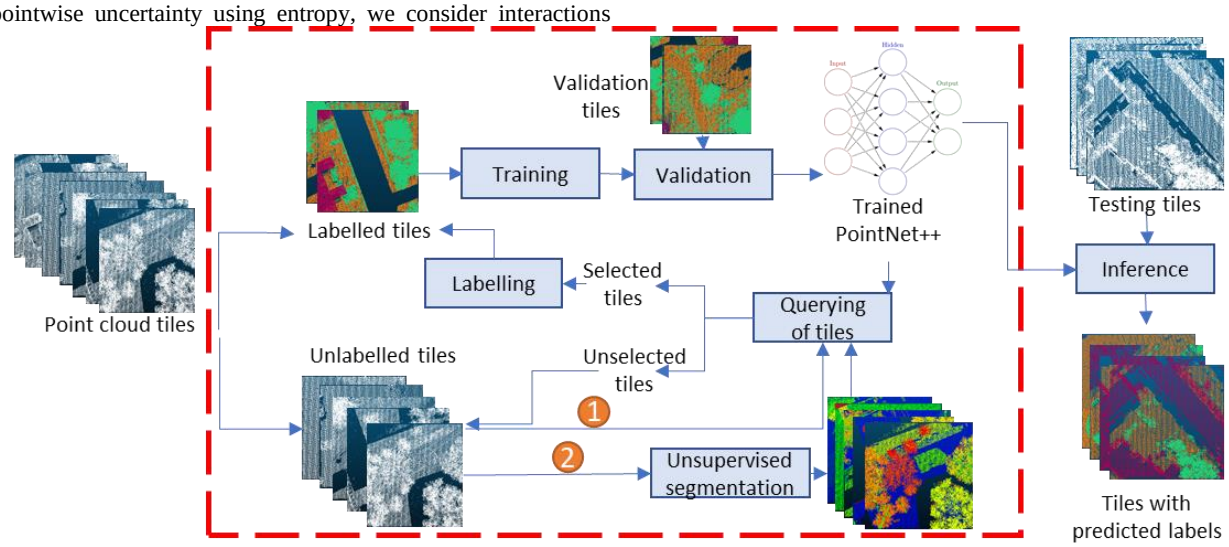

Figure 1. The proposed workflow for active learning strategy for semantic segmentation point cloud. Firstly, point clouds are split into tiles and separated into two groups: labelled (minority) and unlabelled (majority). Then, the network is trained on labelled tiles and the trained network selects segmented unlabelled tiles. Here two queries are tried. One query 1 directly consumes unlabelled tiles and the other one (query 2) relies on the unsupervised segmentation. Selected tiles are labelled before the next training. The trained network is evaluated on testing tiles in each iteration.

2. RELATED WORK

A lot of efforts have been spent in semantic point cloud segmentation tasks, starting from machine learning techniques with predefined features to deep learning methods achieving state-of-the-art performance. The following paragraphs give a brief review on recent deep learning approaches and related active learning strategies.

2.1 Deep learning approaches

Recently, more and more researchers put efforts on addressing semantic segmentation of point clouds via deep learning techniques. Current methods are divided into two categories. They are 2D based methods and 3D based methods. In the 2D category, 3D point clouds are converted into 2D images and then fed into 2D convolutional networks. For example, Su et al. (2015) propose image-based networks called Multi-view CNNs which take rendered views of 3D shapes as inputs. Similarly, Kalogerakis et al. (2017) propose fully convolutional networks (FCNs) to achieve the semantic segmentation at object levels. FCN gives a confidence map to every single view and then a surface based CRF (conditional random field) layer aggregates all these maps with geometric consistency cues to produce coherent part segmentation. Attempts have also been made to apply image based CNNs to large scale ALS data. To separate ground and non-ground points, Rizaldy et al. (2018) convert ALS point clouds into height images which are the input of FCN.

Semantic point cloud segmentation is also solved by many 3D deep learning networks. Some of them split the 3D space into small grid voxels to adapt to 3D convolution filters (Maturana and Scherer, 2015; Wu et al., 2015). However, this voxelization always introduces artefacts that not only impede the learning of effective 3D features but also hinder the network generalization to semantic segmentation tasks. To mitigate the side-effects of voxelization, networks are designed to directly consume points. Klokov and Lempitsky (2017) propose Kd-Networks to handle unstructured point clouds, which is free from conventional uniform 2D and 3D grids. They use Kd-trees as underlying structures where point clouds are recursively split by binary spatial partition. Then, point clouds are constructed in a hierarchical way and this encodes shape information that is valuable for recognition and segmentation tasks. Qi et al. (2017a) propose PointNet which can also directly consume point clouds. The network is robust to variance in geometric transformations and can process unordered point sets. Then, PointNet++ is proposed to learn features in multiple scales (Qi et al., 2017b). Taking PointNet and PointNet++ architecture as backbones, Wang et al. (2019) design an associatively segmenting instances and semantics framework, aiming to perform instance and semantic segmentation simultaneously. The framework takes advantages of two tasks and contributes to a win-win situation. Some attempts explore the potential of graph convolutional networks (GCNs) in point cloud labelling tasks. SuperPoint Graph (SPG) implements GCNs on segments, aiming to deal with large scale data (Landrieu and Simonovsky, 2018). The graph is constructed by superpoints and edges. Superpoints are geometrically homogeneous elements and edges describe the adjacency relationship between superpoints. GCNs take the graph as the input to exploit contextual information between shapes and objects, and achieve top accuracy in semantic segmentation of point clouds in large scales. Wang et al. (2018) propose a GCN based module Edge-Conv that dynamically computes graphs in order to capture geometrical structure at different scales. Griffiths and Boehm (2019) and Xie et al. (2019) review most recent deep learning techniques for 3D data classification.

2.2 Active learning

Active learning aims at querying informative samples to maximize model performance. The main challenge in active learning is to estimate the informativeness of samples, which has been researched for a long history in the machine learning

community. There are various approaches to selecting unlabelled data. Uncertainty sampling is the most commonly used method, which preferential selects the samples that models are least confident about. Density weighted methods select samples that are not only uncertain but also representative of the underlying data distribution (Settles and Craven, 2008). Expected change based methods choose data that cause the largest change in the current model (Vezhnevets et al., 2012). A comprehensive summary of active learning techniques is given by Settles (2009).

Recently, few studies are applying active learning techniques to point cloud processing. Luo et al. (2018) design a framework to combine active learning and higher order MRF for the semantic segmentation of mobile LiDAR point clouds. During the sampling, they consider neighbour-consistency in a way that two spatially adjacent supervoxels are likely to have the same label. That means, for an unlabelled supervoxel, if its predicted label is different from the label of its nearby manually labelled supervoxel, it is considered as a misclassified sample and is supposed to be selected and manually labelled to improve the statistical model in the next iteration. Although the work selects optimal training data and saves some manual labelling work, the classifier MRF still requires pre-defined features which are not enough representative compared to deep learning features. Feng et al. (2019) integrate active learning with a state-of-the-art deep learning method for 3D object detection in LiDAR data. They use deep ensembles and Monte-Carlo dropout techniques to estimate both aleatoric (data dependent) and epistemic (model dependent) uncertainty. By active learning, the labelling efforts are reduced by 60%. To the best of our knowledge, there is no research which combines active learning with deep learning of semantic segmentation of point clouds.

3. METHOD

Figure 1 demonstrates the workflow in this paper. The red dash line box demonstrates the active learning strategy proposed in this paper. There are three components, namely, training, querying tiles with unsupervised segmentation information and labelling. The following sections firstly give an overview of the active learning framework and then explain the network structure and the training method. Finally, point entropy and segment entropy are introduced to query tiles.

3.1 Active learning

Algorithm 1: Active Learning Algorithm

Input: a pool of unlabelled point cloud tiles S
Output: the manually labelled point cloud tiles D_L, and a neural network W.
1: initialize D_L by manually annotating some tiles
2: repeat:
3: W = neural_network(D_L)
4: x_s = AL criterion(w, S)
5: $D_L = D_L \setminus x_s$
6: $S = S \cup x_s$
7: until the stopping condition is met
8: return D_L and W

Algorithm 1 demonstrates active learning strategy in steps. The first step in active learning is to initialize the network with several labelled point cloud tiles. After the training, the trained model evaluates the informativeness of each tile in the unlabelled pool. Here, we select unlabelled tiles (x_s) by query functions introduced in section 3.3 instead of selecting points or super voxels (Luo et al., 2018). Luo et al. (2018) extract predefined pointwise features, like linearity and planarity which can only be calculated from neighbouring points. MRF classifiers can assign every point a label according to its predefined features. However, in deep learning based methods, geometrical features are learned from data, so networks have to consume point cloud tiles that preserve geometrical information. If only several sparsely distributed points within a tile are labelled, the whole tile is still required in the network to learn geometrical features. The computational cost is the same no matter whether fully labelled tiles or partially labelled tiles are used. The only difference lies in the loss function where unlabelled points have no contribution. If we select points from all unlabelled data, we have to put all training points (all tiles) into the network for every training. The training time is then the same as for using fully labelled training data, which is quite time-consuming. If we query tiles, the less tiles we select, the less time is required for training. Considering time efficiency, we label point clouds by tiles. Then selected tiles (x_s) are used to update the training data D_L and then are removed from unlabelled pool S. This training and selecting process is iterated until the stopping criterion is satisfied, like there is no significant improvement in network performance for several iterations or the network performance is sufficient.

3.2 Semantic segmentation by PointNet++

3.2.1 Network structure

PointNet++ (Qi et al., 2017b) is a hierarchical neural network that recursively implements PointNet (Qi et al., 2017a). It encodes point cloud features by set abstraction modules at multiple scales in order to capture point cloud structures in a larger context. A set abstraction module consists of three layers, namely, sampling, grouping, and PointNet. In the sampling layer, a set of points are selected by iterative farthest point sampling (FPS). This helps receptive fields to adapt to the data distribution and avoids sampled point clustering within a small region. Then, neighbours around centroid points are grouped within a given radius. Next, selected points are fed into a PointNet layer. Here, each input point corresponds to a small group of points in a small local region and each group member has its own features like XYZ coordinates or features extracted from last set abstraction module. The PointNet encodes this neighbouring information into a 1D vector.

3.2.2 Network training

Networks in this paper are trained from scratch and all weights in PointNet++ are randomly initialized. However, we only initialize model weights once and all models start from the same initialization. This is because this paper aims to compare how different sampling strategies influence the model performance, while different starting points could lead to different network performances. This side-effect should be mitigated because we are only interested in the influence of different training samples on the model performance. During the training, networks are optimized end to end by stochastic gradient descent, aiming to minimize a weighted cross entropy loss. The loss function puts more weights on the loss caused by less frequent classes, dealing well with imbalanced data. Networks are trained with dropout to avoid over-fitting. In order to find optimal network weights, networks are assessed on validation data for every epoch. Weights are saved if validation loss improves and the training stops when validation loss has no improvement for several epochs.

3.3 Query functions

This paper compares three strategies of sampling point cloud tiles. They are random sampling, point entropy sampling and segment entropy sampling. The first one randomly draws samples and we take it as a baseline to see whether the other two methods can outperform it. The other two methods are introduced in the following sections.

3.3.1 Point entropy

Shannon Entropy (SE) estimates the amount of information is required to 'encode' a distribution.

$$E = -\sum_{c=1}^{C} p(y=c|\boldsymbol{x}) \log p(y=c|\boldsymbol{x}) \quad (1)$$

Where p(y=c|**x**) is the predictive probability for class c coming after the softmax function at the end of the network. If the model is very confident about a certain class label by assigning high predictive probability to that class and assigning very low values to other classes, the entropy for that point is low. On the contrary, the entropy is high when the model gives similar probabilities to all possible classes. Here, we query data which current trained networks are uncertain about. Therefore data with high entropy are preferred. As mentioned in section 3.1, this research selects point cloud tiles to save the time in training networks. To find the most informative tiles, pointwise entropies are averaged within all unlabelled tiles. Tiles with high averaged entropy are selected, labelled and combined with other labelled tiles.

3.3.2 Segment entropy

The uncertainty cannot only be estimated at the point level but also the segment level. Point cloud segmentation aims to separate points into geometrical homogenous units. Here we use the segmentation algorithm proposed by Vosselman et al., (2017). The unsupervised segmentation algorithm takes the advantages of both planar surface extraction and point feature based segmentation methods. The first step is to segment point clouds into planar objects by Hough transform and surface growing algorithm. However, this planar surface extraction over-segments non-planar objects like vegetation into small pieces. Therefore, only very large segments are kept and the rest of points are re-segmented by a segment growing algorithm relying on planarity and normal vector directions. This groups points on vegetation, cars and chimneys. To solve the over-segmentation on slightly non-planar ground points, adjacent large segments are merged if they share borders, their normal vectors are parallel and points in one segment can also fit the plane of the other segment and vice versa. Unlabelled points are assigned to segment labels based on neighbours' majority voting. Isolated points remain unsegmented and do not count for segment entropy calculation.

Here, we assume that points within a segment are supposed to share the same label. Therefore, if different labels are predicted for points within a segment (Figure 2 middle), the model is likely to give the wrong predictions on those segments and therefore those difficult segments should be selected for training in the next iteration. The distribution of predicted labels within segments is estimated by segment entropy:

$$E_{seg} = -\sum_{c=1}^{C} q(c) \log q(c) \quad (2)$$

$$\hat{y} = argmax_y P(y|\boldsymbol{x}) \quad (3)$$

$$\mathrm{q(c)} = \frac{\sum_{n=1}^{N} f(\hat{y}_n, c)}{N} \quad (4)$$

$$f(\hat{y}_n, c) = \begin{cases} 1, & if\ \hat{y}_n = c \\ 0, & else \end{cases} \quad (5)$$

Where E_{seg} is the entropy of a segments and $q(c)$ represents the percentage of points that are predicted as class c. $\hat{y}$ is the class label that has the highest predictive probability. $q(c)$ is calculated by equation 4, where N is the number points within a segment. Figure 2 demonstrates low and high segment entropies on a roof segment. As samples are selected by tiles, points within a segment share the same segment entropy and pointwise segment entropies are averaged within all unlabelled tiles.

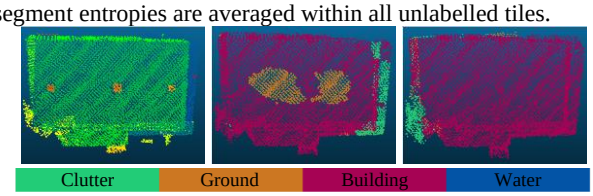

Figure 2. Segment entropy. Left: unsupervised segmentation results. Middle: high entropy within the roof segment. Right: low entropy within the roof segment.

4. EXPERIMENTS

To verify the efficiency of active learning in semantic point cloud segmentation tasks, airborne Lidar point clouds are taken as the source data in our experiments. More details about the dataset, the specific structure of PointNet++, training parameters and how the proposed query functions are implemented are explained in the following paragraphs.

4.1 Dataset and implementation

4.1.1 Dataset

In this paper, we use a subset of AHN3 dataset (which can be downloaded from https://www.pdok.nl/nl/ahn3-downloads) which is a 2km*2km area in the centre of Rotterdam for the experiment, as shown in Figure 3. The selected point cloud was captured on 4th December 2016 by an IGI LM6800 system with a 60° field of view. The mean strip overlap is 30% and the point density is about 30 points/m^2. Points are classified into 4 classes, namely building, water, ground, and clutter (including vegetation, bridge, and car). The whole area is split into 3 parts for training, validation and testing.

4.1.2 Preprocessing

Due to the limited GPU memory, the network cannot directly process the entire study area. Therefore, the point cloud is subdivided into 50m*50m tiles. Within each tile, only position information for each point is taken as the input of the network. X, and Y coordinates are normalized by the starting location of the tiles and Z coordinates remain unchanged. In our experiments, for each tile, 20000 points are randomly picked during the training without replacement. As divided tiles vary in point density, for tiles with less than 20000 points, points are randomly and repeatedly selected. Tiles with less than 2000 points are excluded from the training. In order to improve the model robustness to noise and orientation, training point clouds are randomly rotated around the Z-axis. Also, XYZ coordinates are jittered by adding Gaussian noise which is centred at zero with σ=4cm and the values are clipped to a maximum jitter of

15cm. These values are set empirically to add noises while maintaining the geometrical features for target objects.

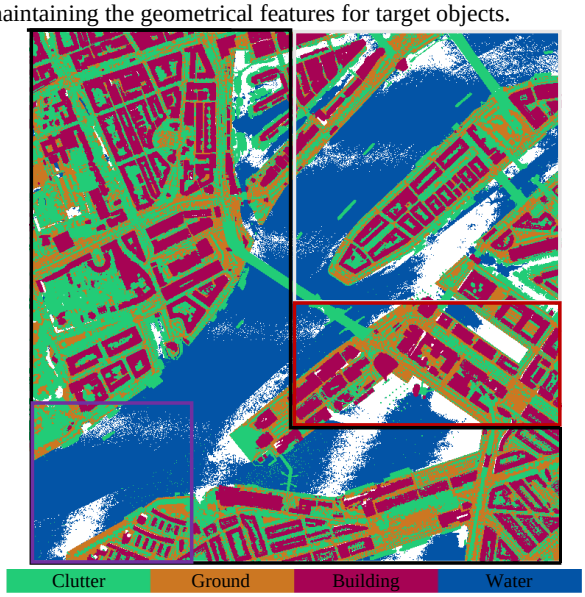

Figure 3. An overview of the study area. The training area is in the black box. The validation area is in the red box and the testing area is in the grey box. The area to initialize the model is in the purple box.

4.1.3 PointNet++ implementation

According to section 3.2, PointNet++ is a sequence of sampling and grouping layers. Table 1 demonstrates the spatial scales of four set abstraction modules. At the first level, in one tile, 4096 points are sampled from 20000 points according to iterative farthest point sampling strategy. Then, neighbouring points are grouped at two scales. 16 neighbours are searched within 2 meters and 32 neighbours are searched within 4 meters. For the second level, 1024 points are sampled from the 4096 points at the first level and then grouped at two larger scales. At higher levels, fewer points are available, and this inevitably causes the loss in information, but this sampling allows the network to capture a wider range of contextual information.

Level	Number of Points	Search radius (m)	Number of neighbours
0	20000		
1	4096	[2, 4]	[16, 32]
2	1024	[4, 8]	[16, 32]
3	256	[8, 16]	[16, 32]
4	64	[16, 32]	[16, 32]

Table 1. Parameter setting of multiple grouping modules in PointNet++

During the training, the learning rate starts from 0.005 with a decay rate of 0.7 at every 5 epochs. The learning rate stops decreasing when it is smaller than 0.0001 and its value remains at 0.0001. The training stops when there is no improvement in model performance on the validation dataset for 30 epochs.

PointNet++ is only able to take a fixed number of points in each point cloud tile. As a result of sampling, some points are still unlabelled in the original dataset. However, some points are duplicated because of repetition. Thus, we use nearest neighbour interpolation to propagate the probability distribution of predicted points to the whole original tiles. The class with the highest probability is assigned to each unclassified point.

4.1.4 Accuracy assessment

Network performances are evaluated by Intersection over Union (IoU) (Everingham et al., 2010). IoU is calculated from true positives (TP), false negatives (FN) and false positives (FP) in confusion matrices as TP / (TP + FN + FP).

4.1.5 Active learning

An area of 600m*600m located at the southeast of the study area is picked for the first training (purple box in Figure 3) because it includes all four classes (ground, building, water, and clutter). Excluding very sparse tiles, 107 tiles are selected to initialize the model. There are 783 tiles in the unlabelled pool. Considering the time efforts, it is not feasible to select very few samples in each iteration and to run the training and selecting process for too many times. However, it also does not make sense to select a large portion of the data like half or a quarter of the data for labelling in one iteration as all point cloud tiles would then be sampled in two or four iterations. This cannot demonstrate how informative samples progressively improve the model performance. To keep the balance between time and model performance, 35 tiles are sampled at each iteration, corresponding to 5% of the initially unlabelled tiles. Instead of manually labelling the selected tiles, the 4-class labels are taken from the original AHN3 dataset. To test the efficiency of active learning strategies, we run the querying and training for 10 times and this selects about half of the whole area. In this research, the active learning strategies based on point entropy and segment entropy are compared with the baseline method where unlabelled tiles are randomly queried. Each strategy runs for 3 times.

4.2 Results and discussion

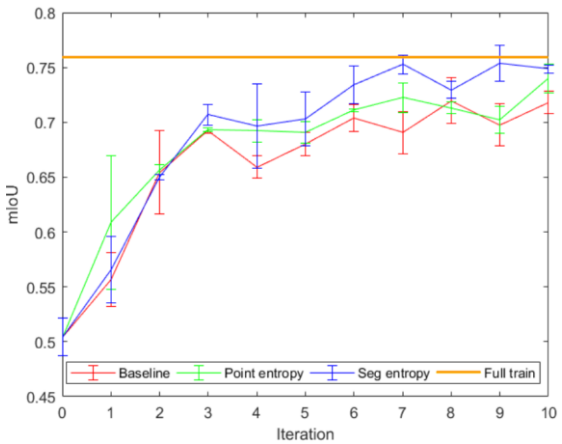

Figure 4. Mean IoU scores of baseline and active learning strategies with different query functions. The horizontal axis represents the iteration. Error bars represent standard deviation.

demonstrates the performance of different active learning strategies. It can be seen that with the increasing size of labelled data, the mIoU keeps increasing with some fluctuations for all three approaches. Before the fourth iteration, mIoUs for all three methods sharply increase and the performances of point based entropy and segment based entropy are not significantly better than the baseline. This is probably because, at the beginning stage, the tiles used for training are quite similar. For example, at the third iteration, 105 tiles are supposed to be labelled in addition to the 107 tiles with which we initialize models in all three methods. Therefore, we have 212 training tiles in total and at least half of the training tiles are the same. Also, for the first several runs, networks are scarce of data and give very low mIoU scores. Adding more tiles is less likely to provide redundant information even if point cloud tiles are randomly selected. At the third iteration, the mIoU reaches 70.17%, 69.31% and 69.25% for segment entropy, point

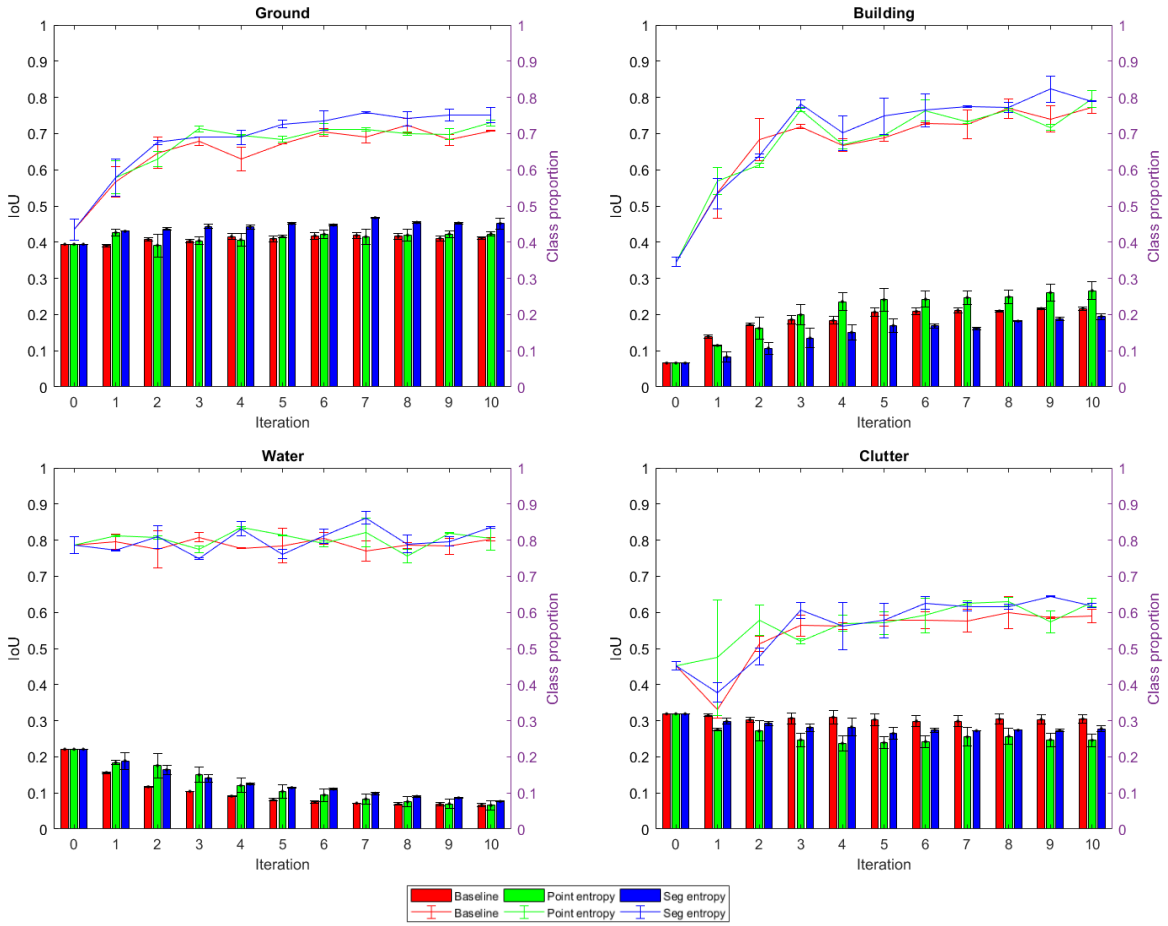

Figure 5. IoU (lines) and data distribution (columns) for different classes. The horizontal axis represents the iteration. Error bars represent standard deviation.

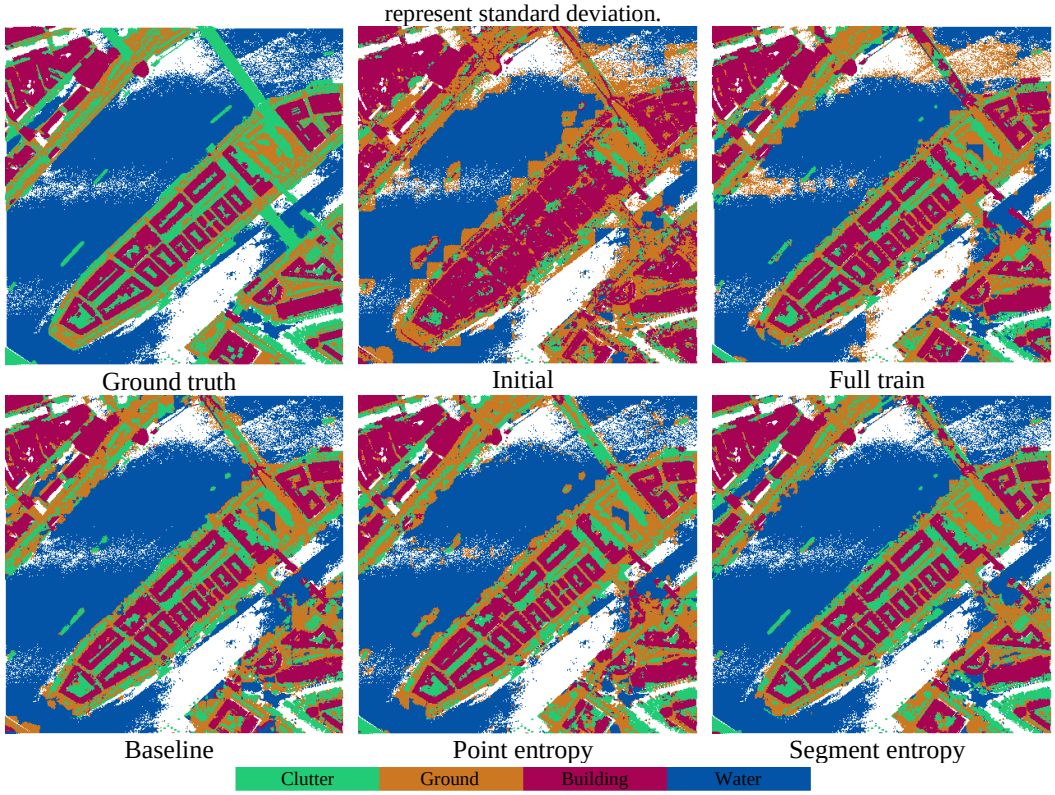

Figure 6. Qualitative comparison of different model performance. The first row shows the ground truth labels, model performances trained on initial data and full training data. The second row shows final model performances trained on data iteratively selected by baseline, point entropy and segment entropy strategies.

entropy and baseline respectively. Comparing to 75.9% mIoU obtained from a full training model where all points in the training area are labelled, all these three methods achieve over 90% of the accuracy in mIoU using only about a quarter of the training tiles.

Starting from the fourth iteration, the model performances are quite different. Although the model performance of the baseline is in a tendency to improve, the accuracy is always lower than the other two methods. In the segment entropy method, model performance firstly reaches 75.2% at seventh iteration which is 99.1% of the full train accuracy in mIoU. After that, the performance starts to be stable with some fluctuations.

Point entropy performs slightly better than the baseline between the fourth and seventh iteration. Then the baseline catches up with it at the eighth iteration. The performance of the point entropy is not as good as that of segment entropy. The highest mIoU (74.25%) is achieved at the tenth iteration which is 97.8% of the fully trained strategy accuracy.

The line plots in Figure 5 illustrate the change in IoU with increasing iteration for different classes. Column plots in Figure 5 demonstrate how the data distribution changes with different query functions. It can be seen that all classes experience an increase in IoU score except water. Figure 6 shows that water points are well predicted even by the initial model and their IoU keeps relatively high values which fluctuate around 80% (Figure 5). One possible reason is that the model is initialized by tiles with a large water area and the IoU is too high to be improved in the following iterations. Also, in our dataset, water is the easiest class which is featured by large flat areas. In the following selection, the water points take less and less percentage. Comparing to random selection, both point entropy and segment entropy keep higher proportion of water points which contributes to slightly better performances on small water areas (Figure 6).

Point entropy queries tiles with more building points. This is because the networks are very uncertain at large roof areas which are characterized by large flat areas. The confusion exists between the building and other classes and results in high entropy for building points. However, comparing to the performance of segment entropy, it seems simply selecting point cloud tiles with pointwise entropy without considering interactions between points cannot significantly improve the network performance.

It is interesting to note that segment entropy prefers tiles with more ground points compared to the other two methods. This preference gives rise to the higher IoU for ground. Although segment entropy selects fewer building points, the IoU score is relatively higher. This is because ground points are likely to be misclassified as building and vice versa. The model reduces the amount of false negatives for ground and at the same time decreases the number of false positives for building and is thereby contributing to better performance for the building class.

The change in the percentage of clutter demonstrates that both point and segment entropy query functions can ignore points that cannot make contributions to the model performance. By random sampling, clutter points always take about 30% of all selected points, while both designed query functions prefer smaller proportions of clutter points without impairing accuracy.

5. CONCLUSION

In this paper, we explore the application of active learning in semantic point cloud segmentation. Instead of simply assessing pointwise uncertainty, we proposed a segment based query function, considering interactions among points within segments, to assess the informativeness of samples. Here, the proposed strategy uses 40% of the whole training dataset to achieve 99.1% of the accuracy in mIoU obtained from the model trained on all full dataset. In the future, Bayesian networks will be explored to assess the uncertainty of samples to effectively reducing the labelling efforts for deep learning training.

REFERENCES

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

Cooper, H. M., Chen, Q., Fletcher, C. H., & Barbee, M. M. 2013. Assessing Vulnerability Due to Sea-Level Rise in Maui, Hawai'i Using Lidar Remote Sensing and GIS. *Climatic Change*, *116*(3–4), 547–563.

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

Feng, D., Wei, X., Rosenbaum, L., Maki, A., & Dietmayer, K. 2019. Deep Active Learning for Efficient Training of a LiDAR 3D Object Detector. *ArXiv Preprint ArXiv:1901.10609*.

Griffiths, D., & Boehm, J. 2019. A Review on Deep Learning Techniques for 3D Sensed Data Classification. *Remote Sensing*, *11*(12).

Hastie, T., Rosset, S., Zhu, J., & Zou, H. 2009. Multi-class AdaBoost. *Statistics and Its Interface*, *2*(3), 349–360.

He, K., Zhang, X., Ren, S., & Sun, J. 2015. Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Kalogerakis, E., Averkiou, M., Maji, S., & Chaudhuri, S. 2017. 3D Shape Segmentation with Projective Convolutional Networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3779–3788.

Klokov, R., & Lempitsky, V. 2017. Escape from Cells: Deep Kd-Networks for the Recognition of 3D Point Cloud Models. *Proceedings of the IEEE International Conference on Computer Vision*, 863–872.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. 2012. ImageNet Classification with Deep Convolutional Neural Networks. *Proceedings of the 25th International Conference on Neural Information Processing Systems*, *1*, 1097–1105.

Lemmen, C., van Oosterom, P., & Bennett, R. 2015. The Land Administration Domain Model. *Land Use Policy*, *49*, 535–545.

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

Li, Y., Bu, R., Sun, M., Wu, W., Di, X., & Chen, B. 2018. PointCNN: Convolution On X-Transformed Points. *Advances in Neural Information Processing Systems*, 820–830.

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

Maturana, D., & Scherer, S. 2015. VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 922–928.

Murgante, B., Borruso, G., & Lapucci, A. 2009. Geocomputation and Urban Planning. In *Geocomputation and Urban Planning* (pp. 1–17).

Qi, C. R., Su, H., Mo, K., & Guibas, L. J. 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, *1*(2), 4.

Qi, C. R., Yi, L., Su, H., & Guibas, L. J. 2017. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. *Advances in Neural Information Processing Systems*, 5099–5108.

Rizaldy, A., Persello, C., Gevaert, C., Oude Elberink, S., & Vosselman, G. 2018. Ground and Multi-Class Classification of Airborne Laser Scanner Point Clouds Using Fully Convolutional Networks. *Remote Sensing*, *10*(11), 1723.

Settles, B. 2009. *Active Learning Literature Survey*. University of Wisconsin-Madison Department of Computer Sciences.

Settles, B., & Craven, M. 2008. An Analysis of Active Learning Strategies for Sequence Labeling Tasks. *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Sivaraman, S., & Trivedi, M. M. 2014. Active Learning for On-Road Vehicle Detection: A Comparative Study. *Machine Vision and Applications*, *25*(3), 599–611.

Su, H., Maji, S., Kalogerakis, E., & Learned-Miller, E. 2015. Multi-view Convolutional Neural Networks for 3D Shape Recognition. *Proceedings of the IEEE International Conference on Computer Vision*, 945–953.

Tang, M., Luo, X., & Roukos, S. 2001. Active Learning for Statistical Natural Language Parsing. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics - ACL '02*, 120.

Vezhnevets, A., Buhmann, J. M., & Ferrari, V. 2012. Active Learning for Semantic Segmentation with Expected Change. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 3162–3169.

Vosselman, G., Coenen, M., & Rottensteiner, F. 2017. Semantic point cloud interpretation based on optimal neighborhoods. *ISPRS Journal of Photogrammetry and Remote Sensing*, *128*, 354–371.

Wang, K., Zhang, D., Li, Y., Zhang, R., & Lin, L. 2017. Cost-Effective Active Learning for Deep Image Classification. *IEEE Transactions on Circuits and Systems for Video Technology*, *27*(12), 2591–2600.

Wang, X., Liu, S., Shen, X., Shen, C., & Jia, J. 2019. Associatively Segmenting Instances and Semantics in Point Clouds. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4096–4105.

Wang, Y., Mendez Mendez, A. E., Cartwright, M., & Bello, J. P. 2019. Active Learning for Efficient Audio Annotation and Classification with a Large Amount of Unlabeled Data. *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 880–884.

Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., & Solomon, J. M. 2019. Dynamic Graph CNN for Learning on Point Clouds. *ACM Transactions on Graphics (TOG)*, *38*(5), 1–12.

Wu, Z., Song, S., Khosla, A., Fisher, Y., Zhang, L., Tang, X., & Xiao, J. 2015. 3D ShapeNets: A Deep Representation for Volumetric Shapes. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1912–1920.

Xie, Y., Tian, J., & Zhu, X. X. 2019. A Review of Point Cloud Semantic Segmentation. *IEEE Geoscience and Remote Sensing Magazine*.

Zolanvari, S. M. I., Ruano, S., Rana, A., Cummins, A., da Silva, R. E., Rahbar, M., & Smolic, A. 2019. DublinCity: Annotated LiDAR Point Cloud and its Applications. *Proceedings of the 30th British Machine Vision Conference*.