

GENERATING SYNTHETIC 3D POINT SEGMENTS FOR IMPROVED CLASSIFICATION OF MOBILE LIDAR POINT CLOUDS

Sagar A. Chitnis, Zexian Huang, Kourosh Khoshelham

University of Melbourne, Parkville, Victoria 3010 – {sagarc@student, zexianh@student, k.khoshelham@}.unimelb.edu.au

KEY WORDS: Point Clouds, Synthetic Point Segments, Mobile Lidar, Adversarial Autoencoder, Classification.

ABSTRACT:

Mobile lidar point clouds are commonly used for 3d mapping of road environments as they provide a rich, highly detailed geometric representation of objects on and around the road. However, raw lidar point clouds lack semantic information about the type of objects, which is necessary for various applications. Existing methods for the classification of objects in mobile lidar data, including state of the art deep learning methods, achieve relatively low accuracies, and a primary reason for this under-performance is the inadequacy of available 3d training samples to sufficiently train deep networks. In this paper, we propose a generative model for creating synthetic 3d point segments that can aid in improving the classification performance of mobile lidar point clouds. We train a 3d Adversarial Autoencoder (3dAAE) to generate synthetic point segments that exhibit a high resemblance to and share similar geometric features with real point segments. We evaluate the performance of a PointNet-like classifier trained with and without the synthetic point segments. The evaluation results support our hypothesis that training a classifier with training data augmented with synthetic samples leads to significant improvement in the classification performance. Specifically, our model achieves an F1 score of 0.94 for vehicles and pedestrians and 1.00 for traffic signs.

1. INTRODUCTION

Mobile Lidar is the primary technology for capturing detailed 3d spatial data of road environments. Point clouds captured by mobile lidar systems provide an accurate 3d representations of real-world objects. Such highly detailed 3d representations are very useful in a variety of applications such as urban planning, traffic asset inventory, construction and autonomous driving (Guan et al., 2016). Although mobile lidar point cloud data provide an accurate geometric representation of the real world, they lack semantic information that is necessary for most applications. The common approach to efficient generation of semantic information from point cloud data is segmentation and classification using supervised machine learning.

The application of supervised machine learning to mobile lidar point clouds faces a few critical challenges. Mobile lidar data are characterised by varying point density and gaps in the point cloud. In addition, most objects appear incomplete in mobile lidar point cloud data due to the line-of-sight nature of lidar as well as occlusion (Xia et al., 2021). For example, a tree might appear in the point cloud without its trunk and only with leaves and branches, and vehicles are often scanned from one side only. But, arguably the most critical challenge in the supervised classification of mobile lidar data is the lack of adequate training samples for every object. This is particularly relevant for the state-of-the-art deep learning methods, which require a large number of training samples for training. Without adequate training data, classification is a challenging task for deep networks (He et al., 2020). While in other applications of deep learning with different forms of inputs, such as images and text, training data is often available in large quantities, the availability of 3d training data is a significant problem in the case of point clouds. Manual annotation of point clouds to generate training data is a challenging task due to the discrete nature of the point clouds as well as varying point density and incompleteness of point segments. Therefore, techniques that can aid in the generation of training data become vital for improving

the classification accuracy of point clouds.

This research is the first attempt to explore the application of synthetic point segments for improving the performance of deep networks in the classification of mobile lidar data of road environments. We propose a semi-supervised approach based on variational autoencoder (VAE) and generative adversarial network (GAN) to generate synthetic point segments from real point segments obtained from a mobile lidar dataset. We evaluate the classification performance of a PointNet-like classification network trained with and without synthetic samples to test the effectiveness of synthetic samples for accurate classification of mobile lidar point clouds. Our results indicate that significant improvement in the classification performance can be achieved by using synthetic training samples.

2. RELATED WORK

There is a rich body of literature on segmentation and classification of point clouds. In this section, we review the most relevant works on point cloud segmentation and classification as well as recent works on synthetic data generation.

2.1 Point Cloud Segmentation and Classification

Segmentation and classification are two primary tasks for understanding 3D point clouds. Segmentation and classification can be performed separately to partition the point cloud into point segments representing objects and assigning a class label to each segment to identify its type. This approach is usually referred to as segment-based (or instance) classification. Alternatively, the two tasks of segmentation and classification can be combined into one process where every point in the point cloud is assigned a specific segment and class label. This is commonly referred to as semantic segmentation. In the following we review the related literature on these two approaches.

2.1.1 Segment-based Classification Segment-based classification methods involve an initial segmentation of the point cloud followed by the classification of individual segments (Khoshelham et al., 2013, He et al., 2020). Compared to images with a regular structure of pixels, applying deep learning methods to point segments is more challenging due to the irregular distribution of the points (Bello et al., 2020). To overcome this challenge, some deep learning methods use an indirect approach which involves processing the raw point clouds into regular structures for feature extraction. These transformations aid the convolutional operations to process the point clouds, like regular input data formats, and segment them into different clusters. Indirect methods can be broadly categorised as voxel-based and multi-view based. Voxel-based methods, such as VoxNet (Maturana and Scherer, 2015), pre-process point clouds to 3D voxel structures that perform feature extraction using 3d kernels. Whereas multi-view based methods, such as VMCNN (Qi et al., 2016) and MVCNN (Su et al., 2015), convert point segments to a collection of 2D image representations. Indirect methods offer good classification accuracy; however, they have several limitations. The pre-processing of lidar data into a regular structure involves additional computation. Point cloud voxelisation appends additional information to the raw data, which increases the computational complexity of the network. The transformations can increase the number of voxel grids in point segments, thereby increasing memory consumption (Maturana and Scherer, 2015). To improve the computation time, one can generate fewer voxel grids but this could lead to critical information loss. Due to the limitations of indirect methods, we explore the direct 3D deep learning methods.

2.1.2 Semantic Segmentation The direct semantic segmentation techniques process the raw point clouds without any intermediate transformations. PointNet (Qi et al., 2017) is the first popular network architecture that directly processes the raw point clouds (Bello et al., 2020). Several other networks (Luo et al., 2020a, Luo et al., 2020b, Luo et al., 2021, Li et al., 2018, Wu et al., 2019) use the PointNet architecture as a base network and propose improvements. The PointNet network has specific modules that make the processing invariant to point permutation and geometric transformations. The network learns to assign a class label either to every individual point in the point cloud, for semantic segmentation, or an entire segment, for segment-based classification (Qi et al., 2017).

PointNet has several advantages over the indirect deep learning methods with respect to space and time complexity. As PointNet directly processes every point, the time complexity is $O(n)$. Other methods based on multi-view representation have a time complexity of $O(n^2)$, whereas those involving volumetric representations have a time complexity of $O(n^3)$ (Qi et al., 2017).

2.2 Synthetic Data Generation

Synthetic data is artificial data that possess characteristics or properties similar to original data. The idea of synthetic data in machine learning is to generate synthetic training samples that resemble real samples and share similar features with real data. Synthetic samples are particularly useful for 3D deep neural networks which require an adequate amount of data to train and classify objects effectively. Since manual labelling of mobile lidar data is time-consuming, synthetic samples can result in faster and cheaper data generation and more accurate classification of point clouds. Generative networks are a popular choice for creating artificial data with a feature distribution similar to that of the original data. We explore generative

models, such as the Variational Autoencoders (VAE) and Generative Adversarial Networks (GANs), for producing synthetic point segments.

2.2.1 Generative Adversarial Networks (GANs) GANs are deep neural networks that comprise two modules, generator and discriminator, which compete against one another to generate an output that satisfies certain conditions (Goodfellow et al., 2014). These internal neural network components ensure that the generated samples have a distribution close to the original real samples. The generator is often an autoencoder which encodes the original data into a lower-dimensional representation and reconstructs the original data from the compressed low-dimensional representation. The low-dimensional representation in the latent space captures the significant features of the original data. However, an autoencoder generates reconstructed copies of the input real samples, whereas for our task we require synthetic variations of the real samples. This requires learning the feature distribution of real samples.

2.2.2 Learning Data Distributions Variational Autoencoders (Kingma and Welling, 2013) are generative models that generate synthetic data by learning the distribution of features using a loss function. The VAE architecture also consists of two components, namely the encoder and decoder. The encoder is a neural network that compresses the input to a lower dimensional vector. On the other hand, the decoder network tries to reconstruct copies of the input from the compressed vector by sampling from learned feature distribution. Therefore the decoder network can generate instances from a distribution similar to the original data (Kingma and Welling, 2013).

VAEs are suitable for generating variational samples; however, they do have limitations. Firstly, VAEs use the normal distribution as prior with a zero mean and unit variance. The normal distribution helps the regularisation parameter, KL divergence, in the loss function to be tractable (Zamorski et al., 2019). Therefore we cannot use any other distribution, such as Bernoulli. Secondly, there is a possibility that the exploding latent space problem due to a restrictive model causes bad sampling ability (Braithwaite and Kleijn, 2018).

Due to the shortcomings of the VAE, Adversarial Autoencoders (AAE) (Makhzani et al., 2015) that use the Jensen-Shannon divergence as the regularisation term have been preferred for the generation of synthetic samples. AAEs can be trained with any distribution as the prior. The AAE architecture consists of a variational autoencoder as the generator and a binary classifier as the discriminator. The generator learns a probabilistic latent representation of real samples, which can be sampled to generate synthetic samples. The training aims to minimise the reconstruction error and at the same time enforce the latent variables to form a prior distribution. This is achieved by defining a training loss function that combines a reconstruction error term with a regularisation term that represents the distance between the learned distribution and the prior (Makhzani et al., 2015).

3. METHODOLOGY

3.1 Combining Geometrically Similar Categories

Our objective is to study and improve the classification of mobile lidar objects in a roadside environment. Therefore, we use the Sydney Urban Dataset, a mobile lidar dataset captured in Sydney inner suburbs, as the original real dataset. We club

the objects with similar geometric representations, such as cars, buses, vans clubbed together into vehicles. Traffic lights and traffic signs are clubbed together, and pedestrians are separated. Therefore, we have three classes in our Sydney dataset, namely vehicles, pedestrians, traffic symbols as seen in Table 1.

3.2 Synthetic Point Segment Generation using 3dAAE

To generate synthetic samples from the original point segments, we propose a workflow illustrated in Figure 1. We train a 3d adversarial autoencoder (3dAAE) to create synthetic point segments similar to the real data distribution based on the 3dAAE of (Zamorski et al., 2019). To adequately train the 3dAAE, we apply data augmentation techniques to increase our dataset by a factor of 30. Data augmentation gives the classification network a new perception of the same points permuted in the 3d space. Data augmentation helps the model to generalise the training data and improve the performance (Yang et al., 2018). We apply rotation around the z-axis to the point segments in 10-degree intervals and add jitter to slightly displace the points.

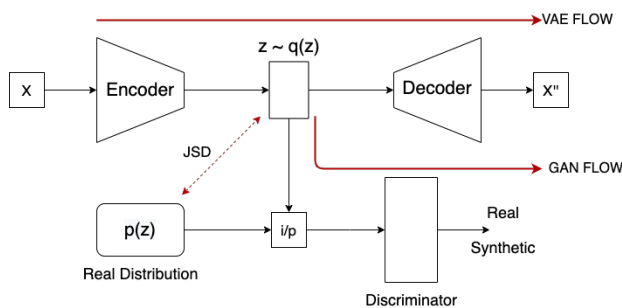


Figure 1. Architecture of 3dAAE

The encoder of our 3dAAE network has two roles during the training: First, it encodes the original point segments into a lower dimension embedding in the training phase. Second, it acts as a generator during the GAN stage of the training. The 3dAAE is trained in two phases: In phase one, the reconstruction phase, the AAE works as a simple autoencoder, focusing on generating highly-reliable reconstructed point segments that are as close to the original point segments, using the compressed vector embeddings. In the second phase, the regularisation phase, the GAN comes into effect and imposes the Jensen-Shannon divergence (JSD) as shown in the GAN flow of Figure 1. The JSD ensures that the distribution of the latent embeddings is close to the distribution of the original dataset (Goodfellow et al., 2014). Following the training procedure described in (Zamorski et al., 2019), the parameters of encoder and discriminator are updated alternately.

The resulting initial synthetic samples may not necessarily resemble the actual samples. Therefore, they are fed to a discriminator, which is a binary classifier that tries to discern synthetic samples from the real ones. The training aims to minimise the reconstruction error while maximising the confusion between the real and the synthetic point cloud instances. Consequently, the 3dAAE network learns to generate variational synthetic samples that resemble actual point segments of the Sydney dataset.

3.3 Point Segment Classification using PointNet

PointNet consists of multiple fully-connected layers that extract the global features of a point cloud segment and two trans-

formation networks for input and feature transformations. As point clouds do not have a structured grid-like structure, convolutional operations are challenging. PointNet uses the multi-layered perceptron (MLP) for feature extraction from raw point clouds. The MLPs are connected to a Max pooling layer that aggregates the global and local features from all the segment points. The Max pooling layer is a symmetric function that accepts several vector inputs and outputs a new vector invariant to the order of the inputs. The aggregated features of the point cloud are then passed to an MLP for classification (Qi et al., 2017).

We apply the mean squared error (MSE) loss function, which gives better classification results than multi-class cross-entropy loss, to measure the classification error between the predicted and ground-truth labels of input point segments. Hence, we formulate the classification loss as:

$$MSE = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n} \quad (1)$$

where y_i denotes the ground-truth one-hot encoded vector, \hat{y}_i denotes the predicted one-hot encoded vector and n denotes the number of point segment samples.

4. EXPERIMENTS

4.1 Training Setup

We use the Sydney Urban Dataset, which contains labelled point segments captured by mobile lidar, for training and testing. We club the categories that are geometrically similar to obtain three main categories: vehicles, pedestrians, and traffic signs. We split the data for every point cloud category in an 80%-20% split for training and testing respectively. From the training data, we split another 5% as validation samples, to monitor the training phase. Most of our point segments have 1024 points, but some instances may have fewer. Therefore, we implement zero padding for instances with fewer points, to ensure all segments have the same size.

4.2 Baseline Classification

We perform a baseline classification on the Sydney dataset without performing any preprocessing or data augmentation techniques on the raw point cloud segments. Table 1 summarises the number of real training and test samples obtained from Sydney dataset. Our classifier is a PointNet-like network trained by minimising the MSE loss (Eq. 1). We use the rectified linear unit (ReLU) as our activation function during the convolution and feature transformation.

Class Labels	Training Samples	Test Samples
Veh.	119	34
Ped.	118	31
Tra.	70	21
Total	307	86

Table 1. Overview of real training and test samples obtained from Sydney Dataset

The baseline classification results are shown in the confusion matrix in Figure 2. The baseline classification achieves an accuracy of 83.72% on test data from Sydney Urban dataset. We observe that there are eight vehicle point segments that are

predicted as pedestrians. Also, five traffic signs are classified as pedestrians and one pedestrian is identified as a vehicle. Visualising samples of pedestrians and traffic signs reveals that the linear shape of some of the pedestrian point segments can be confusing with traffic signs as shown in Figure 3.

	Veh.	26	8	0
Actual	Ped.	1	30	0
	Tra.	0	5	16
		Veh.	Ped.	Tra.
		Predicted		

Figure 2. Confusion Matrix for the baseline model trained and tested on real samples from Sydney dataset



Figure 3. Misclassification due geometrical similarities between Traffic Signs (Left) and Pedestrians (Right)

Figure 4 shows the training curves for the baseline classification experiment. The relatively large validation loss means that the baseline classifier is not able to generalise to samples that it has not seen during training. This is caused by the inadequacy of the training samples which results in the inability to distinguish vehicles from pedestrians and pedestrians from traffic signs.

4.3 Synthetic Segment Generation

4.3.1 Training 3dAAE without data augmentation We train the 3dAAE network on each individual category separately using real samples from Sydney Urban dataset to generate synthetic samples. Since the size of our dataset is very small, as shown in Table 1, the 3dAAE network is unable to learn the global features of the different categories effectively. Consequently, the generated synthetic segments do not resemble real point segments captured by mobile lidar. Figure 5 shows an example synthetic synthetic point segment generated from real vehicle samples. We observe that the points are unevenly scattered with a dense cluster in the centre, and do not form the geometric representation of a vehicle point segment.

4.3.2 Training 3dAAE with data augmentation To generate more real-looking synthetic samples, we apply data augmentation to increase the number of real samples for training 3dAAE. We rotate every segment in 10-degree intervals and apply jitter to increase the training data by a factor of 30. We then

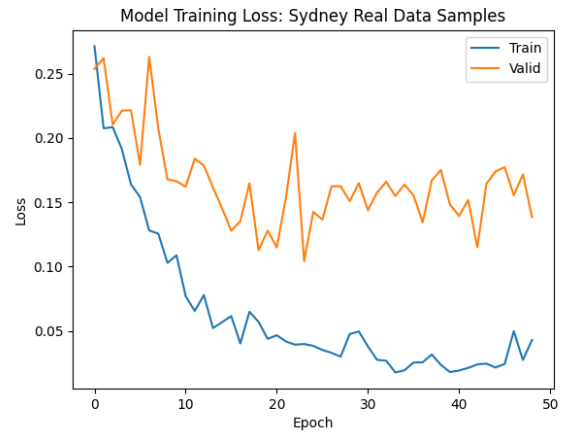


Figure 4. Training curves for the baseline model trained with real samples from Sydney dataset

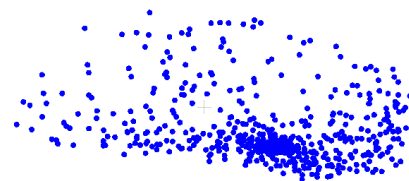


Figure 5. Synthetic vehicle sample generated by 3dAAE trained without data augmentation

use these samples to train our 3dAAE. The generated synthetic samples are then visualised to examine their similarity to real samples.

Figure 6 shows a real traffic sign point segment in blue, with a grid-like structure on top, a pole-like structure in the middle and three leg-like stands at the bottom. Our variational synthetic segments, shown in black, exhibit similar properties, such as the pole-like structure of varying heights. Figure 7 shows a real vehicle segment in blue, where occlusion results in lack of points on one side of the segment. The synthetic vehicle segment, shown in black, is a variation of the original giving a more complete car like structure. The occlusion is observed at certain angles for the synthetic segment, however the geometrical features resemble the real vehicle segment.

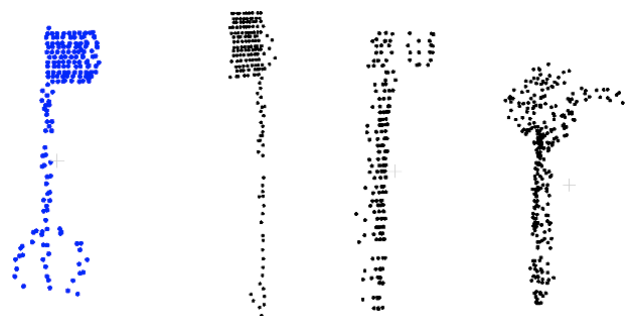


Figure 6. Real traffic sign sample (Blue) and variational synthetic samples (Black)

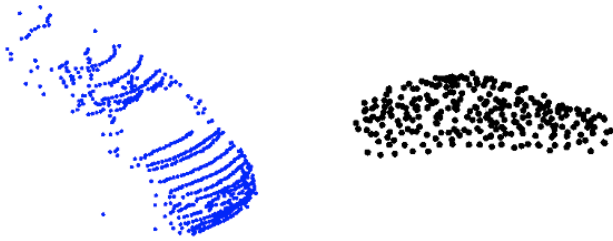


Figure 7. Real vehicle sample (Blue) and variational synthetic sample (Black)

4.4 Classification with PointNet Trained by Both Real and Synthetic Point Segments

Using 3dAAE trained by augmented real samples, we generate synthetic samples for each category. Table 2 shows the number of real and synthetic training samples per category. With a combination of 50%-50% original and synthetic samples, we train our PointNet with the same parameters as we did for the baseline.

Class Labels	Original Samples	Synthetic Samples
Veh.	119	120
Ped.	118	120
Tra.	70	70
Total	307	310

Table 2. Number of real and synthetic training samples

Figure 8 shows the confusion matrix for PointNet classifier trained by both original and synthetic samples. The classifier achieves an accuracy of 95.34%. From the confusion matrix, we see a notable improvement in detecting traffic lights and signs, which are particularly challenging categories in the classification of mobile lidar point clouds. We also see a significant improvement in the classification of vehicles, with 5 out of 8 vehicles that were previously misclassified, being correctly identified. The training curves for PointNet trained by the combination of real and synthetic samples are shown in Figure 9. We observe that the addition of synthetic samples to the training results in smaller validation loss for the model.

Actual	Predicted		
	Veh.	Ped.	Tra.
Veh.	31	3	0
Ped.	1	30	0
Tra.	0	0	21

Figure 8. Confusion Matrix for PointNet trained with original + synthetic segments

Table 3 provides a comparison of classification results for the baseline model and the model trained by both real and synthetic samples. As it can be seen, the introduction of synthetic

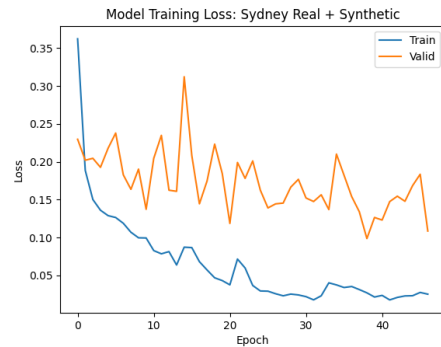


Figure 9. Training curves for the model trained with real+synthetic samples

samples to the training leads to a significant improvement in the classification results. In particular, traffic signs and traffic lights are identified with an accuracy of 100%.

To observe the effect of synthetic training samples on the weights of the trained classifier, we plot the distribution of the weights for the global feature aggregation layer of PointNet. Figure 10 shows the distribution of the weights of the PointNet model trained with and without synthetic samples. We observe that the weights in the two networks have a similar distribution, although the baseline model has a wider distribution with a peak that is slightly shifted towards larger weights.

	Model trained by real samples			Model trained by real+synthetic samples		
	Precision	Recall	F1	Precision	Recall	F1
Veh.	0.96	0.76	0.85	0.97	0.91	0.94
Ped.	0.70	0.97	0.81	0.91	0.97	0.94
Tra.	1.00	0.76	0.86	1.00	1.00	1.00

Table 3. Comparison of classification results for models trained with and without synthetic samples

5. CONCLUSION

In this paper, we proposed a novel method for the generation of synthetic 3D samples to improve the classification accuracy of mobile lidar point clouds. We demonstrated that the use of synthetic point segments for training a deep classification network leads to significant improvements in the classification results. Contrary to strategies that utilise point clouds of CAD models for synthetic data generation, we use a mobile lidar point cloud dataset.

Our results show that data augmentation is important for generating synthetic samples. Using augmented data, we successfully train a 3dAAE to generate multiple synthetic point segments with variations and increase the volume of the training data. Our results and analysis show significant improvement in the accuracy of a PointNet classifier trained by a combination of real and synthetic samples. We achieve an F1 score of 1.00 for traffic signs and traffic lights, which are some of the difficult categories to classify.

This research demonstrates the potential of synthetic data for

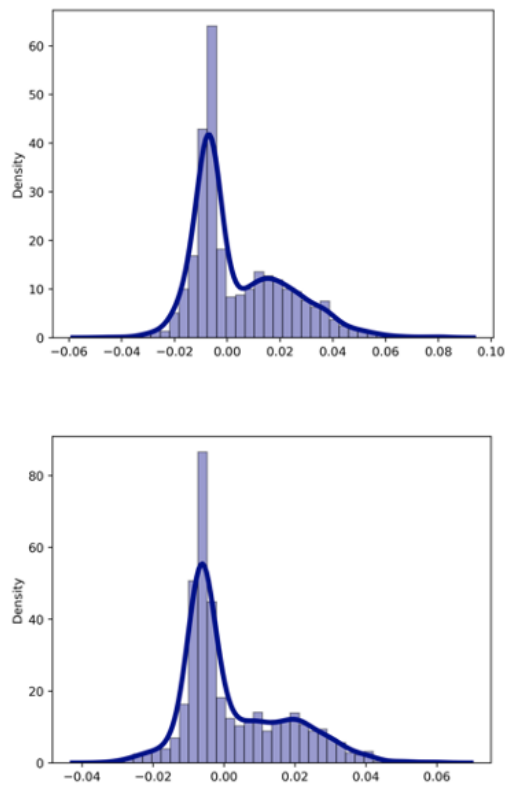


Figure 10. Distribution of weights of the baseline model (Top) and the model trained by real+synthetic samples (Bottom)

improving the performance of machine learning methods applied to point clouds. For future research, we seek to include more categories from the Sydney Urban dataset. Classes such as poles, pillars and buildings are common to road environments and will be included in future research. We will also experiment with other mobile lidar point cloud datasets to further evaluate the effectiveness of using synthetic training samples for point cloud classification. Lastly, we will experiment with other classifiers in addition to PointNet to observe the effectiveness of synthetic samples in training different classification networks.

REFERENCES

- Bello, S. A., Yu, S., Wang, C., Adam, J. M., Li, J., 2020. deep learning on 3D point clouds. *Remote Sensing*, 12(11), 1729.
- Braithwaite, D. T., Kleijn, W. B., 2018. Bounded information rate variational autoencoders. *arXiv preprint arXiv:1807.07306*.
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y., 2014. Generative adversarial nets. Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, K. Q. Weinberger (eds), *Advances in Neural Information Processing Systems*, 27, Curran Associates, Inc.
- Guan, H., Li, J., Cao, S., Yu, Y., 2016. Use of mobile LiDAR in road information inventory: A review. *International Journal of Image and Data Fusion*, 7(3), 219–242.
- He, H., Khoshelham, K., Fraser, C., 2020. A multiclass TrAdaBoost transfer learning algorithm for the classification of mobile lidar data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 166, 118–127.
- Khoshelham, K., Elberink, S. O., Xu, S., 2013. Segment-based classification of damaged building roofs in aerial laser scanning data. *IEEE Geoscience and Remote Sensing Letters*, 10(5), 1258–1262.
- Kingma, D. P., Welling, M., 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
- 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*, 31, 820–830.
- Luo, H., Chen, C., Fang, L., Khoshelham, K., Shen, G., 2020a. Ms-rrfsegnet: Multiscale regional relation feature segmentation network for semantic segmentation of urban scene point clouds. *IEEE Transactions on Geoscience and Remote Sensing*, 58(12), 8301–8315.
- Luo, H., Khoshelham, K., Chen, C., He, H., 2021. Individual tree extraction from urban mobile laser scanning point clouds using deep pointwise direction embedding. *ISPRS Journal of Photogrammetry and Remote Sensing*, 175, 326–339.
- Luo, H., Khoshelham, K., Fang, L., Chen, C., 2020b. Unsupervised scene adaptation for semantic segmentation of urban mobile laser scanning point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 169, 253–267.
- Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., Frey, B., 2015. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*.
- 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)*, IEEE, 922–928.
- 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*, 652–660.
- Qi, C. R., Su, H., Nießner, M., Dai, A., Yan, M., Guibas, L. J., 2016. Volumetric and multi-view cnns for object classification on 3d data. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5648–5656.
- 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.
- Wu, W., Qi, Z., Fuxin, L., 2019. Pointconv: Deep convolutional networks on 3d point clouds. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9621–9630.
- Xia, Y., Xu, Y., Wang, C., Stilla, U., 2021. VPC-Net: Completion of 3D vehicles from MLS point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 174, 166–181.
- Yang, B., Luo, W., Urtasun, R., 2018. Pixor: Real-time 3d object detection from point clouds. *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 7652–7660.
- Zamorski, M., Zięba, M., Klukowski, P., Nowak, R., Kurach, K., Stokowiec, W., Trzciniński, T., 2019. Adversarial autoencoders for compact representations of 3d point clouds.