EXPLORING ALS AND DIM DATA FOR SEMANTIC SEGMENTATION USING CNNS

F. Politz [1,*], M. Sester[1]

[1] Institute of Cartography and Geoinformatics, Leibniz University Hannover, Germany - (florian.politz, monika.sester)@ikg.uni-hannover.de

KEY WORDS: Airborne Laser Scanning, Dense Image Matching, CNN, encoder-decoder Network, semantic segmentation, point cloud

ABSTRACT:

Over the past years, the algorithms for dense image matching (DIM) to obtain point clouds from aerial images improved significantly. Consequently, DIM point clouds are now a good alternative to the established Airborne Laser Scanning (ALS) point clouds for remote sensing applications. In order to derive high-level applications such as digital terrain models or city models, each point within a point cloud must be assigned a class label. Usually, ALS and DIM are labelled with different classifiers due to their varying characteristics. In this work, we explore both point cloud types in a fully convolutional encoder-decoder network, which learns to classify ALS as well as DIM point clouds. As input, we project the point clouds onto a 2D image raster plane and calculate the minimal, average and maximal height values for each raster cell. The network then differentiates between the classes ground, non-ground, building and no data. We test our network in six training setups using only one point cloud type, both point clouds as well as several transfer-learning approaches. We quantitatively and qualitatively compare all results and discuss the advantages and disadvantages of all setups. The best network achieves an overall accuracy of 96% in an ALS and 83% in a DIM test set.

1. INTRODUCTION

Semantic classification is an essential step in processing point cloud data. A classified point cloud is the starting point for many high-level remote sensing products such as digital terrain models (DTMs) or city models. In contrast to small indoor scenes or point clouds measured by a terrestrial laser scanner, the point clouds used for remote sensing applications have a much larger extent of several kilometres compared to dozens of meters. There are two different types of point cloud data available. In Airborne Laser Scanning (ALS), the distance between a plane and the earth's surface is measured using the runtime of a laser beam. With the measured distance and the plane's rotation and position, the point coordinates are calculated. ALS point clouds have a quite sparse overall point density of only several points/m². Dense Image Matching (DIM) point clouds are the second point cloud type. A semi-global matching algorithm matches aerial image pixels to create DIM point clouds. Due to the algorithm, every pixel in those aerial images creates a point in the point cloud resulting in a high point density.

Due to their different origin, ALS and DIM have several different characteristics. The surface of a DIM point cloud is much smoother compared to the ALS point cloud. As a result, ground and low vegetation have similar characteristics in the DIM point cloud and edges on buildings are bevelled. While neighbouring points in ALS can be either ground or vegetation points because of several reflections from the laser beam, there are no ground points below dense foliage in DIM, since the optical sensors cannot reach to the ground. This being the case, DIM only describes the highest points within a scene, which form the surface. Due to these different characteristics, finding one classifier for both point cloud types is very challenging.

In the last couple of years, Convolutional Neural Networks (CNNs) won several benchmark tests for 2D image segmentation such as ImageNet and PASCAL VOC (Krizhevsky et al., 2012; Girshick et al., 2013). In this work, we use both point cloud types in different setups for training a CNN and explore the abilities of the network to deal with these characteristically different point clouds as input. The setups include training with only one point cloud type, training jointly with both point cloud types and training with a transfer-learning approach from an ALS network towards a DIM network. Rather than searching for the most optimized solution, we focus on comparing these different setups on an ALS and DIM test set.

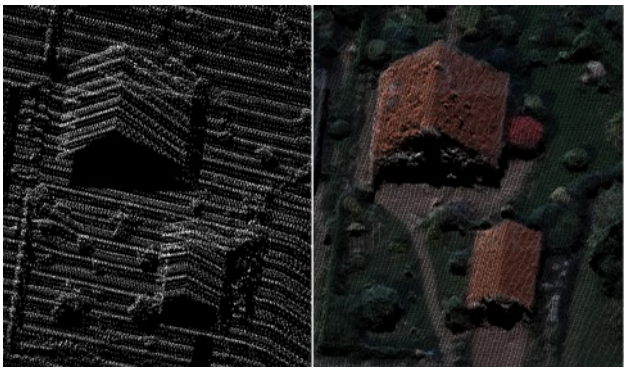

Figure 1. Example of an ALS point cloud (left) and a DIM point cloud (right)

2. RELATED WORK

In this section, we present related work in respect to Deep Learning approaches in context of point cloud classification. These approaches are can be distinguished into two groups: 3D-based methods and 2D-based methods.

* Corresponding author

In the 3D-based methods, the point cloud is processed either as points or as a voxel structure. Qi et al. (2017) proposed a Multilayer Perceptron architecture (MLP), which classifies each point within a 1m³ space using point specific attributes and a global feature vector. It achieves superb results on indoor data; however, classifying one 1m³ block at a time is not feasible for remote sensing data with a large extent. Engelmann et al. (2017) extended this idea by introducing several mechanisms on the input- and output-level to increase the spatial receptive field for 3D outdoor scenes. Instead of classifying single points, Landrieu and Simonovsky (2017) condensed points with similar geometry into superpoints. A graph convolution network classifies those superpoints as separate graph nodes. In contrast to point-based methods, Huang and You (2016) proposed a 3D CNN with a voxel grid, which classified points according to their neighbouring voxels. Similarly, Tchapmi et al. (2017) also used a 3D CNN in a voxelized scene to obtain class score probabilities. Additionally, they introduced a trilinear interpolation step to transfer those class scores back to the original point cloud. Finally, they used a fully connected Conditional Random Field implemented as a Recurrent Neural Network for global optimization. Despite achieving excellent results in several benchmark tests on 3D terrestrial laser scanning data, the 3D-based methods are hardly applicable for airborne data. Usually, ALS and DIM data have a much larger extent, which increases the inference time for those methods quite significantly. Additionally, although DIM data contains up to 100 points/m² compared to several points/m² for ALS data, they are also much sparser than terrestrial point clouds with hundreds of points/m², which is required for most of the 3D-based applications. Equally, ALS and DIM data are only 2.5D and not 3D, making classifying unnecessarily complicated.

In 2D-based methods, a point cloud is projected into a 2D image plane. Hu and Yuan (2016) characterized an ALS point cloud as a raster image with normalized minimal, average and maximal point heights as input values for a CNN. In their work, they focused on differentiating ground and non-ground points for DTM generation. Politz et al. (2018) extended their idea by introducing a car class as well as using transfer learning on a network, which has been previously learnt on the ImageNet dataset (Krizhevsky et al., 2012). Instead of normalized minimal, average and maximal height, Yang et al. (2017) and Xu and Yang (2018) applied a combination of intensity, eigenvalue-based features, normal vector based features and height above DTM as a three channel raster image for their classification. In contrast, Boulch et al. (2017) collect 2D snapshots of a point cloud containing RGB and depth information as well as the normal deviation to the vertical and a noise estimation for classifying a point cloud using SegNet and U-Net as their CNNs (Badrinarayanan et al., 2017; Ronneberger et al., 2015).

In this work, we will follow the idea of Hu and Yuan (2016) for height image generation. In contrast to the work of Hu and Yuan (2016), we classify whole height images instead of individual points. For the semantic segmentation, we use a simpler version of U-Net by Ronneberger et al. (2015), since it proved to achieve outstanding results in semantic segmentation for image data. Furthermore, we focus on an additional building class besides ground and non-ground as it is necessary for city modelling.

3. INPUT DATA

The National Mapping Agency of the German State of Mecklenburg-Vorpommern (Landesamt für innere Verwaltung Mecklenburg-Vorpommern – LAiV-MV) provided the point cloud data used in this work. The data covers an area of around 19 km² in the southeast of Rostock, Germany. This region includes an urban area with residential and industrial buildings as well as garden plots with several cottages. Additionally, the region has huge agricultural areas, grassland, forests, a river and several small lakes. The ALS and the DIM point cloud cover the same area in 2016, but were derived from different flights. A gyrocopter measured the ALS point cloud, which has a point density of around 19 points/m². The DIM point cloud was derived from aerial imagery by LAiV-MV using the SURE software (Rothermel et al., 2012). A Vexcel camera acquired the images for the DIM point cloud. The original images have an overlap of 80% along and 30% across flight direction. The provided DIM point cloud has a point density of around 97 points/m² on average. Both point cloud types have a 2.5D structure. Besides the raw point clouds, a classification from a semi-automatic process were provided for the ALS data. The points are classified into the classes ground and non-ground. As additional data source, building ground plans from cadastre were available, which enable the additional class building for our experiments as described in 3.2.

3.1 Creation of height images

ALS and DIM point clouds are both highly irregular. In order to create a regular input for a CNN, we exploit the 2.5D property of these point clouds by creating 2D height images. Since the ALS and DIM point clouds originates from an airborne viewpoint, they do not contain any 3D shape information of the observed objects such as building facades. For that reason, we can focus on the surface information on top of an object as well as on the bottom of it rather than on the complete 3D shape. An example of this would be the first and last pulse reflection in ALS, where the point cloud contains information about the treetop as well as about the ground below that tree. In this case, we can split the tree information into two layers, which describe the ground and the tree crown accordingly. This is similar to a digital terrain model (DTM) and a digital surface model (DSM), where the DTM only describes the ground, while the DSM also contains information about objects above ground. Depending on the object, the distance between these two layers varies. For instance, the height above ground is different for a tree and a car, but is quite similar for a tree and a building. For this reason, we need more information about the distribution and the context of the 3D points. In the case of buildings, we can assume a constant slope on most rooftops, which yields points that are very close in height. In opposite to that, the points on a tree crown are noisier, which can cause larger height differences between neighbouring points. We can utilize the distance between the average height and the maximal height in a certain neighbourhood as an indicator to smooth and rough surfaces.

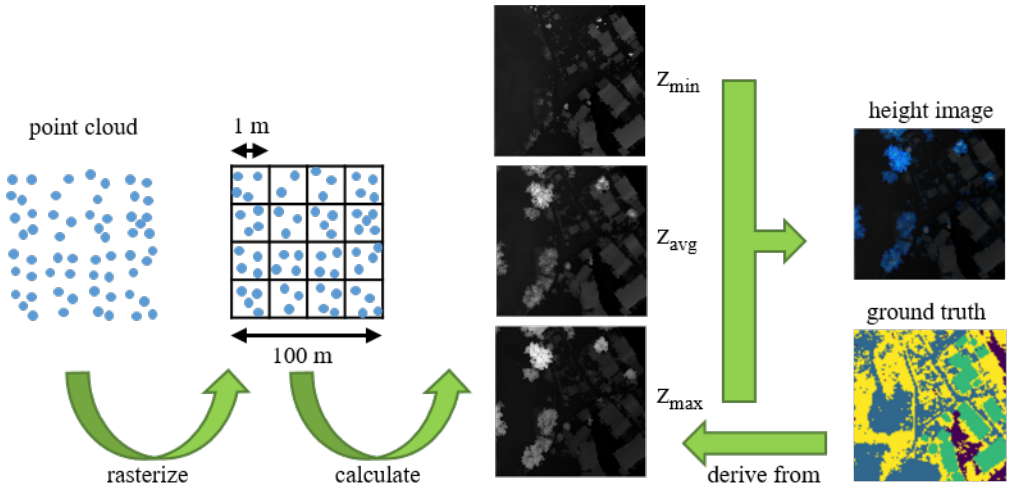

Figure 2. Process chain to create height images. For this figure, the final artificial 3-channel height image is rendered as normalized RGB image. The ground truth is divided into four classes: no data (purple), ground (blue), building (green) and non-ground (yellow).

Consequently, the distance between average and maximal height should be smaller for buildings than for trees. In our approach, we utilize the minimal, average and maximal height as three feature channels in an artificial 2D height image in order to provide indicators to distinguish between ground, building and vegetation.

To create regular height images from irregular point clouds, three steps are necessary as shown in figure 2. Firstly, we rasterize all points into raster cells with a size of 1m. By choosing such a coarse resolution, we have a decent amount of points within each raster cell. Secondly, we calculate the minimal, average and maximal point height within each raster cell. Finally, we crop our data into non-overlapping images with a size of 100 x 100 pixels. This patch size is necessary for keeping context information such as ground or vegetation even for large industrial buildings, which can easily exceed more than half of the image. In addition, these image patches with the three feature channels are the input for the CNN to learn its segmentation task. In the end, the ground is represented rather smooth and has the smallest pixel values due to their lowest height (see figure 2). Buildings and vegetation have higher pixel values. While vegetation has different height values in all three channels, the height values of rooftops are quite similar. In summary, we create 1889 image patches for each point cloud, which cover the complete area of our datasets.

Using height images instead of the original point cloud has several advantages. In contrast to irregular point clouds, images have a regular structure, which is necessary for CNNs. Additionally, by modelling the minimal, average and maximal height of the area, we indirectly give the network additional information such as the ground height of our dataset, the mean and the absolute difference between the mean and its extrema. However, since DIM only includes surface points, the height differences between minimal and maximal height are much smaller than for ALS. For that reason, our trained CNNs behave differently when testing ALS and DIM data as discussed in section 5. Another advantage is that by cropping the dataset into tiles, we allow the network to gather information about the topology between several objects. Finally, we reduce the amount of data to process significantly by eliminating redundant information in neighbouring points, which describe the same object and have similar coordinate and height values.

3.2 Ground truth

The pre-classified ALS point clouds and the additional building ground plans serve as ground truth for the semantic segmentation. The provided ALS point cloud is pre-classified into the classes ground and non-ground. The class ground includes the terrain as well as roads and agricultural land. The class non-ground contains objects, which are not part of a DTM such as buildings, low and high vegetation, bridges and vehicles. In order to separate buildings from other non-ground classes, we project the ground plans from the buildings onto the ALS point clouds so that every point within these ground plans receives a building class label.

However, by projecting the ground plan information onto the ALS point cloud, we also transfer its errors as shown in figure 3. The first two problems occur because of our naïve approach. Since the rooftops of typical residential buildings in Germany are larger than the ground plan of that building, an encircling ring of non-ground points remain around each building (a). The trained networks actually adapt to this characteristics as discussed in section 5. Secondly, if there are points above a building such as a tree, then those points will also receive a building class label (b). Having stated that, this problem only occurs on a couple of cottages close to a forest, which is an insignificant small error for our experiments. What is more problematic, are some coarse errors such as missing buildings (a) or not up to date and partially wrong ground plans (c, d). These errors remain in the ALS point cloud completely. However, our experiments revealed, that our networks can actually fix these major problems in most cases by labelling them with their appropriate label. Since this negatively affects the confusion matrix in all our experiments, we evaluate our results quantitatively as well as qualitatively. The quantitative results allow a coarse estimation, if a network learns, how to distinguish between the classes, while the qualitative results exhibit, what the network has actually learned about the classes.

After rasterizing the point cloud as described in 3.1, the highest point within each cell decides the ground truth for this respective cell. The idea behind this decision follows the characteristics of the DIM point cloud, which only describes the highest points within a scene. If there are no points within a raster cell, then the cell will be given default height values of

0.0 m for all three channels and will be assigned to a ‘no data’ class. The final class distribution is highly imbalanced with 64.91% pixels belonging to ground, 26.98% to non-ground and only 3.29% and 4.83% to building and to no data respectively.

The DIM point cloud data does not contain any pre-classified labels, but as it covers the same area in the same year as the ALS point cloud, we use the same ground truth for both datasets. However, there are some minor differences between DIM and ALS. For instance, the position of dynamical objects such as vehicles differs between both point clouds. In addition, ALS does not cover water surfaces, which leads to pixels having the no data class. In contrast, DIM does have height information for these pixels, but those are very noisy. These minor differences influence the results of each setup as discussed in section 5.

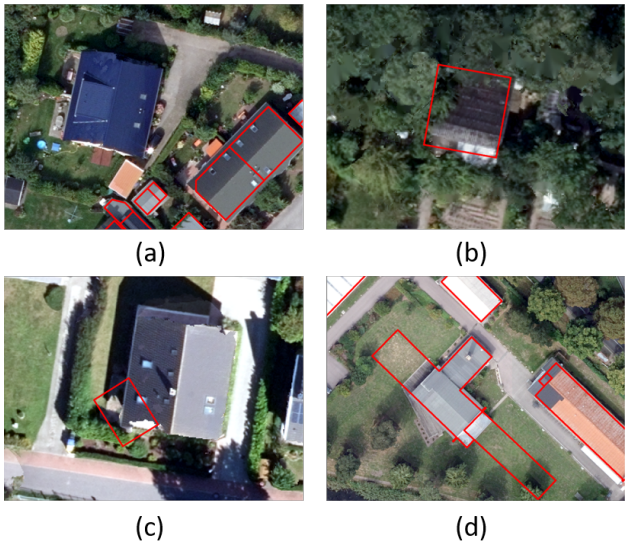

Figure 3. Incorrect building ground plans shown in red lines. (a) Building ground plan is missing. (b) Building is occluded. (c, d) Ground plan is not up to date or partially wrong.

4. METHODOLOGY

In this paragraph, we will present our fully convolutional encoder-decoder network and describe our segmentation setup for our experiments using ALS and DIM point clouds.

4.1 Encoder-decoder network

For the segmentation, we use a similar version to U-Net proposed by Ronneberger et al. (2015) to classify the height images into the four classes as described in 3.2. The utilized CNN is shown in figure 4. The network possesses an encoder and a decoder part. The encoder part codes the height information into aggregated features, where the amount of features for each layer doubles whenever the image resolution decreases. The encoder consists of several convolution blocks, which include a convolution followed by batch normalization (Ioffe and Szegedy, 2015) and a rectified linear unit (ReLU). After two convolution blocks, the image size is reduced by a max-pooling layer. Between the encoder and the decoder part, there is a dropout layer to prevent overfitting (Srivastava et al., 2014). The decoder part then decodes the aggregated features back into the original image resolution and finishes with a class estimation for each pixel using a softmax function. All convolutional layers have a kernel size of 3x3. The output layer has the same spatial resolution as the input and one channel for each class respectively. In addition, symmetrical skip connections connect the encoder with the decoder in order to prevent vanishing gradients and to restore the original object shape (Mao et al., 2016). Due to being fully convolutional, the network only contains around 1.87 million parameter to train. We use cross entropy as loss function and the Adam optimizer (Kingma and Ba, 2015).

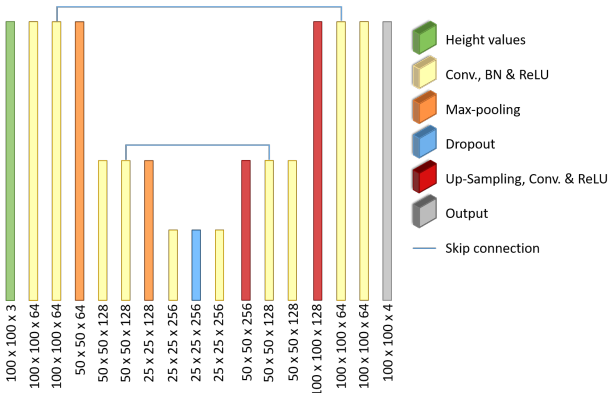

Figure 4. Scheme of our network.

4.2 Training setup

For this work, we train the proposed CNN on several different setups by exploring ALS and DIM point clouds for the segmentation task. In ALS_{train} and DIM_{train}, we train the network exclusively on one point cloud type. In $BOTH_{train}$, we combine both training sets, which cover the same area. Additionally, we also explore a transfer learning approach. Transfer learning considers the real world scenario, where both datasets are not available all the time, but only some segmentation model from another period of time, place or sensor system. Similarly, there might not be enough data to train a new segmentation model. Consequently, transfer learning adapts one domain to another. The starting point for the transfer learning approach is the model trained for ALS_{train} using the best parameter set. Afterwards, the weights of the decoder part are frozen and the training continues using a fraction of the DIM data for transfer learning. In this second training period, the network adapts the encoder part to the DIM data and consequently accomplishes a domain adaption. We explore transfer learning for 10%, 50% and 100% of the DIM data to simulate real world scenarios, where only a fraction of newly labelled data is available. We call those setups $TRANS_{10}$, $TRANS_{50}$ and $TRANS_{100}$ respectively. After training, we test all networks on ALS and DIM test data to see how far they can generalize between both point cloud types.

5. EXPERIMENTS

For our experiments, we use a 5-fold cross validation for parameter tuning. For that reason, we split our data in a non-overlapping training and test set. We further divide the training set in five parts, where four of them are for training and one for evaluating the network. From each point cloud, we derive 1889 height images and randomly take 300 images as test set (ALS_{test}

and DIM_{test}), leaving the remaining 1589 for training and validation. To further increase the amount of training data, we randomly flip the images horizontally and vertically while training, which multiplies our available training data by a factor of 4. We choose the best parameter set depending on the averaged validation results, train the network again using all 1589 image patches and test them on ALS_{test} and DIM_{test}. To choose different parameter sets for the cross validation, we use random hypercube sampling (McKay et al., 1979). According to the validation results, we set the batch size to 32, the learning rate to 0.0001, the decay value to 0.0, the dropout rate to 0.75 and the amount of training epochs to 150. For the transfer learning approach, we only sample values for the learning rate and the decay value, which are 0.00005 and 0.0 respectively. The batch size, dropout rate and the amount of training epochs remain the same.

Quantitative results over all classes on both test sets are shown in tables 1-3. In tables 1 and 2, we compare the F1-Scores of all classes and for all methods on ALS_{test} and DIM_{test}. Table 3 shows the overall accuracy on both test sets for all methods. All values are in %. In most cases, the F1-score and the overall accuracy for ALS_{test} are generally better than for DIM_{test}. This is contingent on training the network solely on ALS ground truth. Likewise, the classes ground and non-ground yield the highest F1-Scores. As explained in 3.2, those classes dominate the overall class distribution by representing nearly 92% of all pixels. Consequently, the networks can optimize those classes better than the uncommon classes building and no data. In addition, the results of building and no data influence the results for the non-ground class, as there are wrong segmentations between those classes. The combined methods $BOTH_{train}$ and the several TRANS variations show very diverse results. On the one hand, $BOTH_{train}$ achieves nearly the same F1-Scores and overall accuracy as ALS_{train} on the ALS_{test} set. In case of DIM_{test}, the F1-score of $BOTH_{train}$ even surpasses the score of ALS_{train} by a large margin up to 50% for the no data class and over 30% for the building class. Consequently, $BOTH_{train}$ achieves a 10% higher overall accuracy on DIM_{test} than ALS_{train}. While training, the network of $BOTH_{train}$ gathers the specialized information about both point clouds and consequently is able to distinguish between both point cloud types. On the other hand, the F1-scores of all TRANS methods are significantly lower compared to the other methods. Especially, the F1-score for buildings for $TRANS_{10}$ is very low with a score of only 5.37% on ALS_{test} and 5.96% on DIM_{test}. However, the results improve with increasing training data as shown in the scores of $TRANS_{50}$ or $TRANS_{100}$ compared to $TRANS_{10}$. Since we picked the fraction of DIM data for the TRANS methods randomly, one possible reason for the poor results of $TRANS_{10}$ could lie in the class distribution of the DIM data, where building pixels might be significantly under-represented.

To compare the results qualitatively, five examples of class predictions, their respective height images and ALS ground truth are shown in figure 5 for all setups. In the figure, no data is marked in purple, ground in blue, building in green and non-ground in yellow. Every two rows belong to one test example having results for ALS_{test} in the first row and for DIM_{test} in the second. In the first column, the input is plotted as normalized RGB values for minimal, average and maximal height respectively. The second column shows the ALS ground truth, which was the basis for training in all setups. The remaining columns show the predictions of every method.

Method	no data	ground	building	non-ground
ALS_{train}	99.88	98.12	79.04	94.56
DIM_{train}	33.68	88.68	70.06	46.42
$BOTH_{train}$	99.07	97.84	79.47	94.06
$TRANS_{10}$	23.12	89.03	5.37	71.31
$TRANS_{50}$	49.47	86.45	45.29	48.51
$TRANS_{100}$	72.49	85.72	53.80	50.72

Table 1. F1-Score for ALS_{test} [%]

Method	no data	ground	building	non-ground
ALS_{train}	24.52	86.01	40.00	58.66
DIM_{train}	68.97	88.20	70.90	75.46
$BOTH_{train}$	72.41	88.38	72.85	75.98
$TRANS_{10}$	51.01	83.50	5.96	60.38
$TRANS_{50}$	59.32	85.43	45.80	61.85
$TRANS_{100}$	60.50	85.65	54.29	62.22

Table 2. F1-Score for DIM_{test} [%]

Method	ALS_{test}	DIM_{test}
ALS_{train}	96.48	73.25
DIM_{train}	72.82	82.85
$BOTH_{train}$	96.13	83.26
$TRANS_{10}$	79.79	74.23
$TRANS_{50}$	74.83	77.28
$TRANS_{100}$	76.66	77.74

Table 3. Overall accuracy for ALS_{test} and DIM_{test} [%]

In general, all predictions confirm the results from the F1-Scores and overall accuracies. ALS_{train} and $BOTH_{train}$ share the most similarities with their according ground truth. While catching the overall context, ALS_{train} also retains a highly detailed prediction. DIM_{train} on the other hand smooths object edges, reduces ground noise and creates distinct segmentations. $BOTH_{train}$ utilizes the advantages of ALS_{train} and DIM_{train} by predicting detailed results for ALS_{test} and smoothed results for DIM_{train}. On the contrary, $TRANS_{10}$, $TRANS_{50}$ and $TRANS_{100}$ have some issues with classifying the smaller classes building and no data correctly. In these cases, the network for $TRANS_{10}$ predicts pixel as non-ground instead of building and as ground instead of no data. The overall segmentation however is comparable with the other methods and improves with more DIM data available. When it comes to incorrect ground truth, nearly all networks are able to classify the pixels with the most logical class (5c).

There are also a few issues in all segmentations. As mentioned in 3.2, the real rooftops are typically larger than the ground plans. As a result, the networks learn that each building has a one pixel wide border around each roof (5b, 5c and 5d). In addition, there are only a few industrial buildings in the whole dataset, which have a larger area than normal residential buildings and which usually have a flat roof. Due to the lack of larger buildings with a flat roof, the networks indirectly learn the width of a standard building and consequently classify the rest of the flat roof as ground (5b). Despite that, ALS_{train}, $BOTH_{train}$ and $TRANS_{10}$ manage to find chimneys on top of those roofs as non-ground. Another interesting issue only occurs when training with ALS data and testing it on DIM data. When training with ALS data, the buildings and the ground are the only smooth objects, while higher and lower vegetation has a rough surface. DIM data on the other hand has a higher point density, which even smoothes treetops.

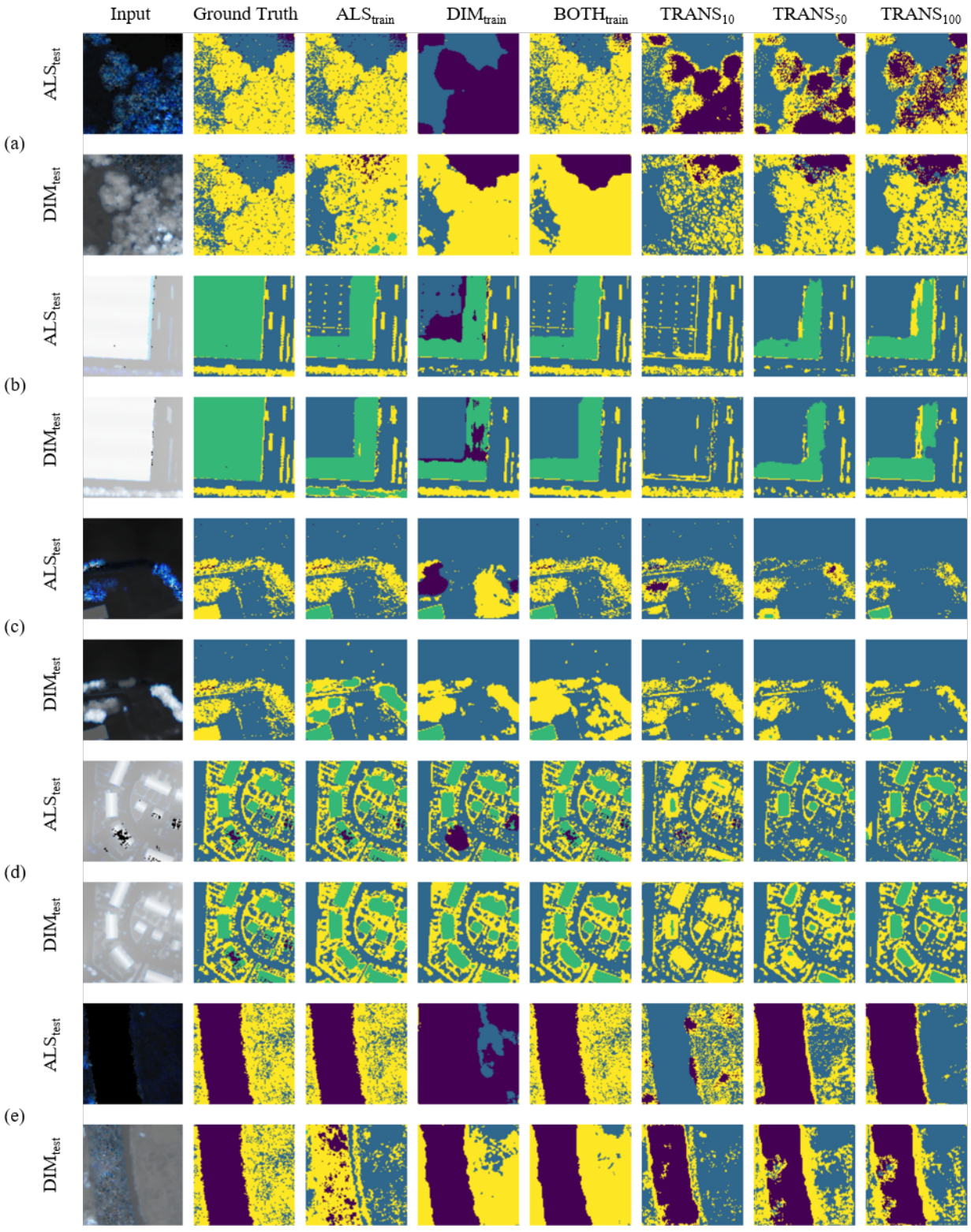

Figure 5. Qualitative results of our network setups. Each trained network is tested on ALS$_{test}$ and DIM$_{test}$ (rows). The input is rendered as normalized RGB image. The classes are no data (purple), ground (blue), building (green) and non-ground (yellow) respectively.

If the treetop has a unified height, ALS_{train} classifies it as building instead of non-ground as shown in 5a, 5b and 5c. By introducing DIM data for training as well, $BOTH_{train}$ eliminates those incorrect building pixels on top of trees. Furthermore, water surfaces are a problem. In ALS, there are no points representing water because of reflection. In images, water pixels exist, but they have no texture, which causes matching errors in DIM causing the height of water points to be quite noisy. As shown in 5e, when training with ALS data, the river is classified correctly as no data for ALS_{test} and as non-ground for DIM_{test}, because it has a similar noisy appearance as trees in ALS data. While training with DIM data, the opposite case occurs and the network learns that noisy data represents the no data class and consequently classifies higher vegetation in ALS_{test} as such (5a). We will discuss how to solve those issues in the next section.

6. DISCUSSION

In this section, we would like to discuss ideas how to improve our results and how to eliminate segmentation errors.

There are a few options to improve the results for buildings. Instead of classifying with the raw measured heights, the normalised heights above ground could be used. Even a low resolution DTM is sufficient, since it will already reduce the influence of the approximate ground height. Alternatively, the point cloud can be pre-processed using an established filtering algorithm such as Zhang et al. (2003) to gather ground information. By normalizing heights, buildings with flat roofs and the ground should have distinct height values and classifying flat roofs should be more precisely. Integrating a class balancing into the training process can improve the overall segmentation of buildings. Currently, the class distribution is highly imbalanced and for that reason, the network optimizes mostly on the ground and non-ground class, which is around 92% of all raster cells in the dataset (see 3.2).

Dealing with no data raster cells is another critical step to improve the quality of our result. For this work, we simply gave empty raster cells default values and added a no data class. In the future though, we would like to exclude those raster cells from the training process completely by introducing sparse convolutional layers into our network (Uhrig et al., 2017). In consideration of that, the network can focus on distinguishing between important classes.

The experiments with $TRANS_x$ with x being 10%, 50% and 100% did not satisfy our expectation. This being the case, we have to investigate possible reasons for the low quality results as well as options for improving those. Currently, we only picked the fraction of DIM data randomly without checking the class distributions, where building and no data class pixels could be highly under-represented. This might be the reason, why the network suddenly starts classifying no data as ground and building as non-ground as being the most similar classes. What is more, the parameters for transfer learning might not be suiting yet. We already reduced the size of possible learning rates and decay values while cross validating the $TRANS_x$ methods, but we might have to reduce them further. Considering that both datasets are not available all the time, we remain convinced, that transfer learning from one point cloud to another is a crucial setup for our segmentation task.

Another issue is the missing and incorrect ground truth for DIM data. As explained in section 3.2, both point clouds cover the same area in the same year. Consequently, most objects are identical in both point clouds and so we used the ALS ground truth for the DIM point cloud as well. However, dynamic objects such as cars or trains, which are in the non-ground class, occur at different spots within the point cloud. Likewise, no data raster cells in the ground truth do not match those in the DIM dataset. This is especially crucial when it comes to segmenting water surfaces, where the water in DIM data has a similar appearance as ALS trees. In order to improve that, a ground truth for the DIM dataset is necessary. Using a correct ground truth will support the network while training, since it eliminates contradictory labels between ALS and DIM data. Likewise, introducing water as an additional class could also be beneficial to distinguish water from no data in both datasets and all setups.

7. CONCLUSION

In this work, we focused on different setups for training a CNN, which is able to segment ALS and DIM point clouds. As CNN require regular input, we project the point clouds into a 2D image raster. Then, we calculate the minimal, average and maximal height of all points within a raster cell as image values. We train the network in six different setups using a cross validation for parameter tuning. Two settings only train on one point cloud type each, one setting on both at the same time and the remaining three setups are firstly trained on the ALS data and then the weights of the CNN are refined using different fractions from the DIM dataset. We achieved satisfying results when training only with ALS and training with both datasets. The results of training only with DIM data were limited due to missing and contradictory ground truth. However, the results from the transfer learning approaches did not satisfy our expectations. We discussed the specific issues with every setup and how to improve them.

For future work, we will improve our results by implementing the ideas discussed in section 6. In addition, we want to extend the network to work also with different time stamps. Furthermore, we would like to introduce more classes such as water, roads, farmland, vehicles and vegetation in order to divide the ground and non-ground class into more specific and meaningful classes. Since we focussed on comparing different setups with ALS and DIM data in general in order to classify both point cloud types with the same CNN, we currently only worked with our datasets, which included both point cloud types. However, we would like to apply our approach to benchmark datasets in order to compare it with other methods.

ACKNOWLEDGEMENTS

We thank the Landesamt für Geoinformation und Landesvermessung Niedersachsen (LGLN), the Landesamt für Vermessung und Geoinformation Schleswig-Holstein (LVermGeo) and the Landesamt für innere Verwaltung Mecklenburg-Vorpommern (LAiV-MV) for providing the point cloud data and for their support of this project.

REFERENCES

Badrinarayanan, V., Kendall, A. and Cipolla, R., 2017: SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. In IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 39 (12), pp. 2481-2495, doi: TPAMI.2016.2644615.

Boulch, A., Guerry, J., Le Saux, B. and Audebert, N., 2017: SnapNet: 3D point cloud semantic labeling with 3D deep segmentation networks. In Computers & Graphics 2018, Vol. 71, pp. 189-198, doi: 10.1016/j.cag.2017.11.010.

Engelmann, F., Kontogianni, T., Hermans, A. and Leibe, B., 2017: Exploring Spatial Context for 3D Semantic Segmentation of Point Clouds. Published as a workshop paper for ICCVW 2017, doi:10.1109/ICCVW.2017.90.

Girshick, R., Donahue, J., Darrell, T. and Malik, J., 2013: Rich features hierarchies for accurate object detection and semantic segmentation. Extended version of the published conference paper at CVPR 2014. arXiv:1311.2524v5.

Hu, X. and Yuan, Y., 2016: Deep-Learning-Based Classification for DTM Extraction from ALS Point Cloud. In Remote Sensing 2016, Vol. 8 (9), 730, doi: /10.3390/rs8090730.

Huang, J & You, S, 2016: Point Cloud Labeling using 3D Convolutional Neural Network. Published as a conference paper at ICPR 2016, doi:10.1109/ICPR.2016.7900038.

Ioffe, S. and Szegedy, C., 2015: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. Published on arXiv 2015, arXiv:1502.03167.

Kingma, D.P. and Ba, J.L., 2015: Adam: A method for stochastic optimization. Published as a conference paper at ICLR 2015, arXiv:1412.6980v9.

Krizhevsky, A., Sutskever, I. and Hinton, G., 2012: ImageNet classification with deep convolutional neural networks. Proceedings of the 25th International Conference on Neural Information (NIPS) 2012, Vol. 1, pp. 1097-1105, doi:10.1145/3065386.

Landrieu, L. and Simonovsky, M., 2018: Large-scale Point Cloud Semantic Segmentation with Superpoint Graphs. Accepted as a conference paper at CVPR 2018, arXiv:1711.09869v2.

Mao, X.J., Shen, C. and Yang, Y.B., 2016: Image Restoration Using Very Deep Convolutional Encoder-Decoder Networks with Symmetric Skip Connections. Published as a conference paper at NIPS 2016, arXiv:1603.09056v2.

McKay, M.D., Beckman, R.J. and Conover, W.J., 1979: A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code. Published in Technometrics, Vol. 21 (2), pp. 239-245, doi: 10.2307/1268522.

Politz, F., Kazimi, B. and Sester, M., 2018: Classification of Laser Scanning Data Using Deep Learning. Published as conference paper at DGPF 2018, Vol. 27, pp. 597-610, https://www.dgpf.de/src/tagung/jt2018/proceedings/proceedings/papers/49_PFGK18_P07_Politz_et_al.pdf .

Ronneberger, O., Fischer. P and Brox, T., 2015: U-Net: Convolutional Networks for Biomedical Image Segmentation. In Medical Image Computing and Computer-Assisted Intervention (MICCAI), Springer, LNCS, Vol. 9351, pp. 234-241, doi:10.1007/978-3-319-24574-4_28.

Rothermel, M., Wenzel, K., Fritsch D. and Haala, N., 2012. SURE: Photogrammetric Surface reconstruction from Imagery. Proceedings at the LC3D Workshop, Berlin 2012, http://www.ifp.uni-stuttgart.de/publications/software/sure/index.en.html (June 19th, 2018).

Srivastava, N, Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R., 2014: Dropout: A Simple Way to Prevent Neural Networks from Overfitting. In Journal of Machine Learning Research, Vol. 15, pp. 1929-1958, http://jmlr.org/papers/v15/srivastava14a.html.

Tchapmi, L.P., Choy, C.B., Armeni, I., Gwak, J. and Savarese, S., 2017: SEGCloud: Semantic Segmentation of 3D Point Clouds. Accepted as a spotlight at the International Conference of 3D Vision (3DV) 2017, arXiv:1710.07563v1.

Uhrig, J., Schneider, N., Schneider, L., Franke, U., Brox, T. and Geiger, A., 2017: Sparsity Invariant CNNs. Proceedings at the International Conference of 3D Vision (3DV) 2017, arXiv: 1708.06500v2.

Qi, C., Su, H., Mo, K. and Guibas, L.J., 2017: PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. Published as a conference paper at CVPR 2017. arXiv: 1612.00593v2.

Yang, Z., Jiang, W., Xu, B., Zhu, Q., Jiang, S. and Huang, W., 2017: A Convolutional Neural Network-based 3D Semantic Labeling Method for ALS Point Clouds. In Remote Sensing 2017, Vol 9 (9), 936, doi:10.3390/rs9090936.

Xu, Z. and Yang, Z., 2018: EIGENENTROPY BASED CONVOLUTIONAL NEURAL NETWORK BASED POINT CLOUDS CLASSIFICATION METHOD. In The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Vol. XLII-3, pp. 2017-2022, doi: 10.5194/isprs-archives-XLII-3-2017-2018.

Zhang, K., Chen, S., Whitman, D., Shyu, M., Yan, J. and Zhang, C., 2003: A progressive morphological filter for removing nonground measurements from airborne LIDAR data. In IEEE Transactions on Geoscience and Remote Sensing, Vol. 41 (4), pp.872-882, doi:10.1109/TGRS.2003.810682.