JOINT CLASSIFICATION OF ALS AND DIM POINT CLOUDS

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

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

Commission II, WG II/3

KEY WORDS: Airborne Laser Scanning, Dense Image Matching, encoder-decoder Network, transfer learning, point cloud

ABSTRACT:

National mapping agencies (NMAs) have to acquire nation-wide Digital Terrain Models on a regular basis as part of their obligations to provide up-to-date data. Point clouds from Airborne Laser Scanning (ALS) are an important data source for this task; recently, NMAs also started deriving Dense Image Matching (DIM) point clouds from aerial images. As a result, NMAs have both point cloud data sources available, which they can exploit for their purposes. In this study, we investigate the potential of transfer learning from ALS to DIM data, so the time consuming step of data labelling can be reduced. Due to their specific individual measurement techniques, both point clouds have various distinct properties such as RGB or intensity values, which are often exploited for classification of either ALS or DIM point clouds. However, those features also hinder transfer learning between these two point cloud types, since they do not exist in the other point cloud type. As the mere 3D point is available in both point cloud types, we focus on transfer learning from an ALS to a DIM point cloud using exclusively the point coordinates. We are tackling the issue of different point densities by rasterizing the point cloud into a 2D grid and take important height features as input for classification. We train an encoder-decoder convolutional neural network with labelled ALS data as a baseline and then fine-tune this baseline with an increasing amount of labelled DIM data. We also train the same network exclusively on all available DIM data as reference to compare our results. We show that only 10% of labelled DIM data increase the classification results notably, which is especially relevant for practical applications.

1. INTRODUCTION

For remote sensing products such as digital terrain models (DTMs), digital surface models (DSMs) or 3D-city models, classifying a point clouds is a crucial step in the processing chain. Classification is often achieved using supervised learning. To this end, training data with ground truth information has to be provided. NMAs often acquire ALS and DIM in regular update cycles, but due to limited capacities, training a classifier from scratch is often not feasible, as it requires a huge amount of training samples. A possible solution to this problem is transfer learning. The core idea of transfer learning is utilizing an already existing classification model by adapting the weights to new and unknown datasets.

ALS as well as DIM are two typical methods to acquire point cloud data. In ALS, the runtime of a beam is used to measure the distance between a sensor and the earth's surface. With the distance and the plane's rotation and position, point coordinates are calculated. Point cloud densities of around 8-10 points/m² and more are common for nation-wide acquisitions (AHN3, 2019). A semi-global matching algorithm serves to create DIM point clouds from aerial images. Every pixel in these aerial images creates a point in the point cloud resulting in a point density similar to the ground sample distance. Aerial images for NMA's purpose often have a resolution of approximately 5 to 20cm, which equals to 25 to 100 points/m². DIM point clouds are usually a secondary product conducted by orthophoto flight missions or by smaller sensors such as unmanned aerial vehicles (UAV). Recently, there are also developments to integrate image data while laser scanning (Toschi et al., 2018).

As already pointed out by Mandlburger et al. (2017), ALS and DIM point clouds have several different characteristics. First, DIM point clouds have very smooth surfaces, so low vegetation often blends in ground and building edges are bevelled due to the smoothing constraint. Unless there are visible terrain points between trees on the images, there are hardly any ground points within forest regions in the DIM point clouds. In ALS, the laser beam penetrates vegetation and returns multiple signals back to the sensor leading to high volatile points in forest regions. Consequently, DIM only contains smooth tree canopies, while points in ALS reflected from the trees as well as the ground below. Second, regions with no texture or with shadows often have matching errors resulting in random heights in the DIM data. Finally, ALS and DIM have various distinct properties concerning the point density, where DIM exceeds ALS, the point accuracy, where ALS has a higher reliability and less occlusion than DIM, and radiometric information, where DIM returns RGB values, while ALS only returns the intensity. For classification, the latter are often used, which hinders transfer learning from one point cloud type to another, since those features are not available. All those different characteristics of both point clouds must be considered for transfer learning.

Since acquiring newly labelled data is very expensive due to extensive manual work, this study focuses on the potential of CNNs to transfer learn from ALS to DIM point clouds. Due to their different characteristics, we can safely assume that a network trained on ALS data will have issues when being applied to DIM data and thus will not reach the quality of a network trained with DIM data. Consequently, a compromise

* Corresponding author

https://doi.org/10.5194/isprs-archives-XLII-2-W13-1113-2019 | © Authors 2019. CC BY 4.0 License.

between the amount of new label data and loss in accuracy must be found. For this reason, we conduct the following experiments: we systematically increase the amount of newly added and labelled DIM data to see when this compromise is fulfilled. The scientific contributions of this paper can be summarized as follows:

- We tackle the problem of different point densities by rasterizing the point cloud into a 2D grid. The input for the network is entirely based on geometrical features and thus avoids any source dependent features, which are not available for another point cloud type.
- We train an encoder-decoder Convolutional Neural Network (CNN) exclusively on labelled ALS data as a baseline and fine-tune its weights in several setups using an increasing amount of labelled DIM data. We compare those setups with a network, which was trained from scratch using only DIM data. As for now, the network distinguishes ground, non-ground, building, water and an additional no data class for empty cells.
- We compare and analyse all trained networks on a separate DIM test set and evaluate the benefits from introducing DIM data to the classification. In addition, we show and discuss remaining problems of the proposed methodology as well as possible solutions.

In large, potentially nationwide applications, we typically have to deal with varying ground heights. This often causes misclassifications between flat ground and roofs, when they share the same global height. For this study, we reduce the ground influence by creating a normalised Digital Surface Model (DSM) by calculating the height above ground using an existing DTM. Such an additional data source is typically available for NMAs, e.g. the DTM from the previous update cycle. It has been shown that for this purpose a coarse DTM is also already sufficient as long as it removes the ground influence, so that building points are above ground points (Rizaldy et al., 2018; Gavaert et al., 2018).

2. RELATED WORK

Point cloud classification in respect to Deep Learning approaches can be distinguished into 3D-based and 2D-based methods.

In 3D-based methods, the point cloud is processed as points, voxels or graphs. Qi et al. (2017a) proposed a method to process points directly using a Multilayer Perceptron architecture (MLP) to classify points within a 1m³ space using the point coordinates as well as colour information. Advancements in PointNet introduced deep hierarchical feature learning (Qi et al., 2017b), increased the spatial receptive field on input- and output-level for 3D outdoor scenes (Engelmann et al., 2017) or integrated a multi-scale classification (Yousefhussien et al., 2018). Nonetheless, Landrieu and Simonovsky (2017) condensed points with similar geometry into super points, which are the nodes for a graph convolution network. Likewise, Te et al. (2018) redefined convolution over graphs by applying a Chebyshev polynomial approximation and made their classification more robust by deploying a graph-signal smoothness prior into their loss function. In contrast, Huang and You (2016) proposed a 3D CNN with a voxel grid and classified points according to their neighbouring voxels. Similarly, Tchapmi et al. (2017) voxelized a scene and obtained class score probabilities using a 3D CNN as well. In addition, they transferred those class scores back to the original point cloud by introducing a trilinear interpolation step and globally optimized their classification results by implementing a Conditional Random Field as Recurrent Neural Network.

In 2D-based methods, the points are projected into a 2D image plane. Hu and Yuan (2016) rasterized point clouds into image space with normalized minimal, average and maximal point heights around each point as input for a CNN. They especially focused on ground and non-ground points for DTM generation. Similarly, Politz and Sester (2018) extended their idea, but used an encoder-decoder network to fasten up the classification process. Yang et al. (2017) and Xu and Yang (2018) applied a combination of intensity, eigenvalue-based features, normal vector based features and the height above ground as a three channel raster image for their classification. Zhao et al. (2018) interpolated height, intensity and roughness values for each point and its environment using natural neighbour interpolation and finally trained a multi-scale convolutional neural network for classification. Similarly, Rizaldy et al. (2018) converted an ALS point cloud into an image containing the height, return numbers, intensity and relative height above ground as features and classified those images in a multi-scale hierarchical network. Finally, Gevaert et al. (2018) selected rule-based ground and non-ground samples using a top hat filter from a point cloud and then applied a bicubic interpolation to approximate a DTM. They subtract the heights of the DTM from a DSM then and trained a fully convolutional neural network using those normalised heights as well as colour information for point cloud classification.

3. METHODOLOGY

In this section, we present the workflow to create height images, the encoder-decoder network and the segmentation setup. The workflow is shown in Figure 1.

3.1 Height images

3.1.1 Reducing ground influence: When dealing with uneven terrain in point clouds, it is beneficial to remove the influence of different terrain heights prior to processing. For that reason, we transform the point clouds into normalised digital surface models (nDSM). The Euclidean distance between each point and a DTM is calculated and this distance replaces the original height as normalised height. Using nDSM simplifies the segmentation task as points with the same class are sharing a similar height.

3.1.2 Calculating height images: ALS and DIM point clouds are irregular, but encoder-decoder networks require regular data. In order to create regular input for the classifier and deal with different point densities at the same time, we create 2D height images from the point clouds. For that reason, the point cloud is rasterised into cells with a length of 1m. We chose such a coarse resolution to ensure that there is a sufficient amount of points within each raster cell (see section 4.1.).

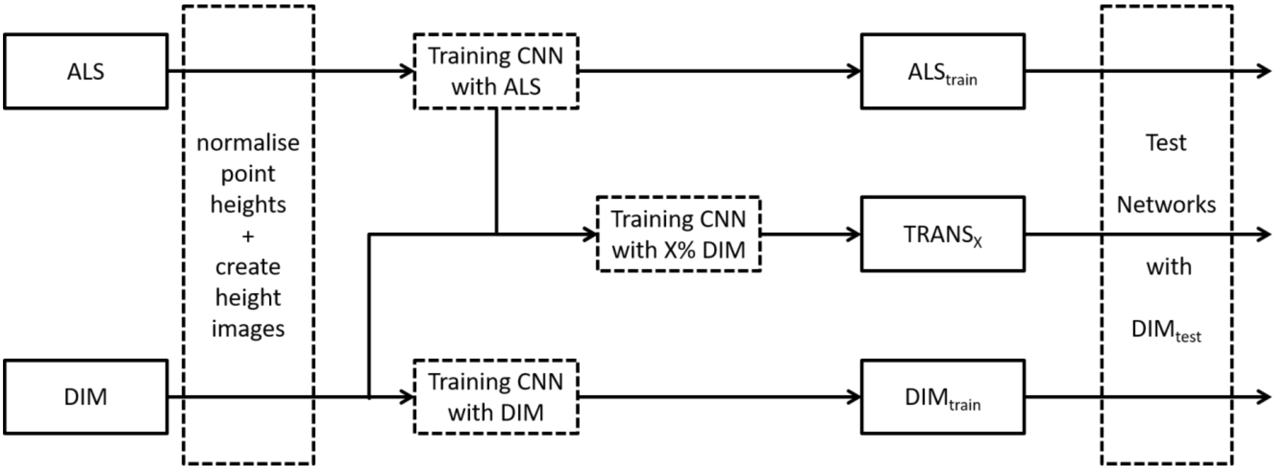

Figure 1: Workflow of our method. The heights of ALS and DIM point clouds are normalised and height images are created. ALS_{train}, and DIM_{train} are trained based exclusively on ALS or DIM point cloud as input data. $TRANS_x$ takes ALS_{train} as baseline and fine-tunes the classification results using X% of DIM data. All setups are tested on a DIM test set.

Additionally, the following features are calculated from all points within a raster cell:

$$z_{min} = min(z_i) \tag{1}$$

$$z_{mean} = \frac{1}{n}\sum_{i=1}^{n} z_i \tag{2}$$

$$z_{max} = max(z_i) \tag{3}$$

where z_i = normalised height of point i
n = amounts of points within a raster cell

Finally, we crop the data into non-overlapping images, where every feature from equation (1 - 3) represents one channel of the final height image respectively. We set the image size to 100 x 100 pixels in order to keep context information. In case of industrial building, this size will not ensure images with ground pixel, but due to the height reduction as described in 3.1.1., the height of the pixels will indicate the network, if the points are on or above ground level.

3.1.3 Reference Data: In order to obtain reference class labels, the point clouds are semi-automatically labelled into four classes: ground, non-ground, building and water. Depending on the normalised height values from 3.1.1, the point cloud is automatically labelled as non-ground, if the normalized point height is above a given threshold, and as ground class in any other case. We set the threshold to 0.3m for the ALS and DIM point cloud to get a common 'ground' for ALS and DIM, which also includes near-ground vegetation due to the properties of DIM of only containing the surface. Furthermore, we project manually labelled building and water shapes generated from orthophotos onto the point cloud plane. Whenever a point is within such a shape, it will receive the respective class label. If it is outside of any shape, their original ground or non-ground label remains.

After rasterising the point cloud as described in 3.1.2, there are multiple points with different reference classes within a raster cell. As we are aiming at a strategy to classify DIM point clouds without learning the network from scratch and since DIM only contains surface points, we chose the highest point within each cell to determine the reference class for this respective cell. A less noisy alternative to the maximum height class would be picking the majority class within the cell. However, in vegetation areas, this would lead to random class decisions in the ALS point cloud, where also ground could be picked as a raster label, which would not be picked in a DIM point cloud at the same place. If there are no points within a cell, this cell will be given default height values and is assigned to a 'no data' class. The default values for z_{min}, z_{mean} and z_{max} are set to -10.0 m in order to simplify the classification of these pixels, since raster cells with real values will mostly avoid the negative range.

3.2 Encoder-Decoder Network

As encoder-decoder network for the segmentation, we use a similar network as proposed by Politz and Sester (2018). This network consists of an encoder part, which codes the height image data into latent variables, and a decoder part, which decodes those latent variables back to the original height size. At the end, the network transforms those decoded features into posteriori probabilities using a softmax classifier. The network includes convolutional blocks, which consist of convolutional layer, batch normalization (Ioffe and Szegedy, 2015) and a rectified linear unit (ReLU). In the encoder, a max-pooling layer follows two of those convolutional blocks and decreases the image size. In the decoder, the latent variables from the encoder are upsampled by a factor of two, concatenated with the encoder of similar size using skip connections (Mao et al., 2016) and finally convolved using two convolutional blocks. Skip connections throughout the network prevent vanishing gradients and support the network restoring the original object shape. In addition, there is a dropout layer (Srivastava et al., 2014) in the middle of the network to reduce overfitting. All convolutional layers have a kernel size of 3x3. The output layer has the same image resolution as the input with one channel for each possible class label. The final amount of training parameters are comparably low with only around 1.87 million, since the network does not contain any dense layers. For backpropagation, we use Adam (Kingma and Ba, 2015) as optimizer and the categorical cross entropy as loss function. An overview of the network structure is shown in Figure 2.

3.3 Training Setup

Since ALS and DIM have different characteristics, transfer learning from ALS to DIM point clouds will always be a compromise between the amount of available label data and loss in accuracy. For the training setup, we test how much the classification results benefit given an increasing amount of labelled DIM data. First, we train the proposed encoder-decoder network exclusively with ALS data (ALS$_{\text{train}}$) as the baseline for our transfer learning approach. Second, we freeze the weights of the encoder part and fine-tune only the weights of the decoder by introducing an increasing amount of labelled DIM data to the network (TRANS$_{\text{x}}$ with X% of added DIM data). In this study, X is set to 10 to 50% of the labelled DIM data. Third, we train the network exclusively on labelled DIM data (DIM$_{\text{train}}$), which represents the optimal configuration. Finally, we will evaluate all setups using DIM test data (DIM$_{\text{test}}$).

In order to find the optimal hyperparameter values, we use a 5-fold cross validation. The height images of a given point cloud are randomly split into non-overlapping training and test sets. The training set is further split into five sets, where four sets are for training and one set is for evaluation at a time. In order to increase the training's examples, we randomly flip the height images horizontally and vertically while training. We choose the best hyperparameter set depending on the averaged validation results, train the networks again on all five training sets and evaluate the final network on DIM$_{\text{test}}$.

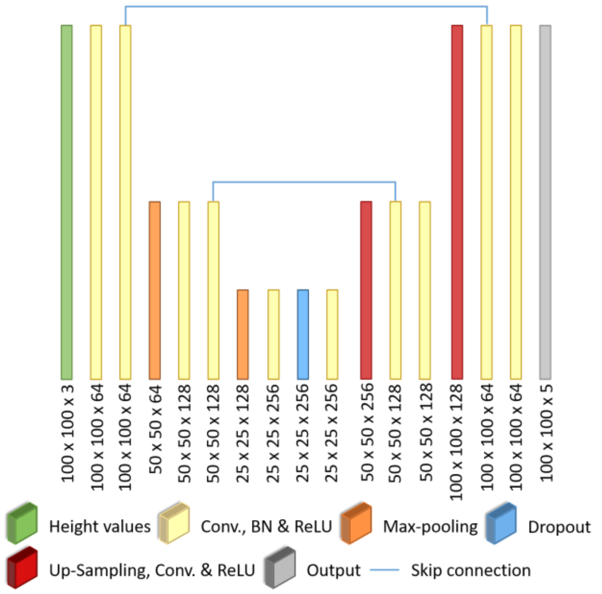

Figure 2. Network structure of the proposed approach

4. EXPERIMENTS

4.1 Input data

The state survey department of Mecklenburg-Vorpommern, Germany (Landesamt für innere Verwaltung Mecklenburg-Vorpommern – LAiV-MV) provided ALS and DIM point cloud data used in this work. The data covers an area in southern Rostock, Germany. In two different flight missions, the point cloud data and the original image data for DIM were captured in 2016 and cover the same area of around 19km². The ALS point cloud has a point density of approximately 19 points/m^2 with a horizontal and vertical accuracy of 15 cm and 30 cm, respectively. The DIM point cloud has a point density of around 96 points/m² with an accuracy of 20 cm horizontally and 30 cm vertically. Urban areas with residential and industrial buildings, garden plots with small cottages, huge agricultural areas, grassland, forests, a river and several small lakes characterise the region.

The ALS and DIM point clouds are pre-processed as described in section 3.1 and each point cloud generates 1889 images in total. These images are then randomly split into 300 test images and 1589 training images, which are further split into five sets of around 318 images for training the 5-fold cross validation as stated in section 3.3. The images are split the same way for ALS and DIM, so the training, validation and test sets cover the same areas. For the transfer learning setups, X% of samples are randomly picked from the 1589 training images and then used for fine-tuning the already trained ALS$_{\text{train}}$. The final class distribution of all training and testing examples is shown in Table 1. Although the point clouds cover the same area, the different classes are highly unbalanced within a point cloud type, but also between both point cloud types. There are two principle differences in the ALS and DIM class distributions: the amount of water pixels for each point cloud type and the relation between ground and non-ground class in both point cloud types.

When hitting water, the laser pulse in ALS only returns in nadir direction and is reflected away with increasing incidence angle, thus in general, only a few water points are present in ALS. In DIM data, water is present, however it is characterised by apparently random heights due to the low structure on the water surface. A height threshold is used to split the normalised point cloud into ground and non-ground. In order to generate a common 'ground' surface in both point clouds, we set the threshold to 0.3m in height. Except for regions with low texture and consequently high noise, the real ground surface of the DIM point cloud lies within this limit of 0.3m. In ALS on the other hand, the ground class will contain all ground points as well as near-ground shrub and grass. As a result and although they are covering the same area, the ALS point cloud will have fewer non-ground pixel and more ground pixel than the DIM data set (Table 1 ALS$_{\text{train}}$, DIM$_{\text{train}}$).

Point cloud	No data	Ground	Building	Water	Non-ground
ALS$_{\text{train}}$	4.82	64.91	4.19	0.73	25.36
DIM$_{\text{train}}$	0.27	50.59	4.73	2.50	41.90
TRANS$_{10}$	0.27	48.26	4.28	2.91	44.28
TRANS$_{20}$	0.23	50.46	5.05	2.21	42.05
TRANS$_{30}$	0.23	49.91	5.20	2.73	41.94
TRANS$_{40}$	0.25	50.28	4.96	2.74	41.77
TRANS$_{50}$	0.27	50.16	4.79	2.79	41.98
DIM$_{\text{test}}$	0.28	49.24	5.16	3.23	42.10

Table 1. Class distribution in the height image data [%]. ALS$_{\text{train}}$ and DIM$_{\text{train}}$ include the images for training and validation set and DIM$_{\text{test}}$ includes images for testing. TRANS$_{\text{X}}$ with X between 10, …, 50 includes a percentage of DIM data randomly picked from DIM$_{\text{train}}$ for transfer learning.

4.2 Hyperparameter of the network

The proposed network from section 3.2 also requires setting several hyperparameters. The batch size describes the amount of samples in each training step. The optimizing function requires a given learning rate, which is necessary for gradient descent. The dropout rate decides how many neurons randomly drop out of the network for each sample. Picking a higher dropout rate supports the network against overfitting. Finally, an epoch parameter controls the maximal amount of epochs to train. We used Latin Hypercube Sampling (McKay et al., 1979) to choose different hyperparameter combinations for cross validation, since it explores the complete feature space. After analysing the results from the 5-fold cross validation, we set the batch size to 128, the learning rate to 0.0005, the dropout rate to 0.85 and the maximal amount of epochs to 100 for all training setups.

4.2.1 Quantitative results: We evaluate our results using the overall accuracy as well as the F1-score (eq. 4 - 7):

$$precision = \frac{T_p}{T_p+F_p} \tag{4}$$

$$recall = \frac{T_p}{T_p+F_n} \tag{5}$$

$$F_1 = 2 * \frac{precision*recall}{precision+recall} \tag{6}$$

$$accuracy = \frac{T_p}{N} \tag{7}$$

where T_p = True positive
F_p = False positive
F_n = False negative
N = number of all pixel

The F1-score and the overall accuracy for DIM_{test} in all seven setups is shown in Table 2 and 3, respectively. The overall F1-score increases when introducing DIM data in the learning process: from 78.7% to 87.1% with 10% DIM data up to 90.2% when including 50% of DIM data. As expected, the best classification is only reached when the network is exclusively trained with DIM data (96.8%).

In the following, the quality of the different experiments will be analysed in detail. It can be observed that the increase in the overall F1-score is different for each class and fluctuates due to inter class relationships. The water and building class benefit the most from incorporating DIM data. As water pixels hardly exist in the ALS training set (see Table 1), giving the network additional DIM data increases the F1-score of water quite notably from 1.6% in ALS_{train} to 65.1% in $TRANS_{10}$. By increasing the amount of available DIM data, the F1-score fluctuates between 60% and 70% for all TRANS setups. However, these scores are still below the F1-score of DIM_{train} of 89.3%, where water is represented well during training. Whereas tree points in ALS are very volatile in structure, the points in tree canopy in DIM point clouds are rather stable. When testing ALS_{train} on DIM_{test}, the network often recognizes these smooth tree crowns as buildings (see Figure 3d, 3h) leading to a low precision of only 24.3% and a poor F1-score of only 38.3% (Table 2).

Incorporating DIM data into the learning process increases the F1-score of the building class notably by 20% to 30%; however, it does not achieve the 86.5% of DIM_{train}. In contrast, the F1-score for the ground class decreases from 92.1% in ALS_{train} to 85.5% in $TRANS_{40}$ and then increases again to 98.6% in DIM_{train}. The F1-scores for the non-ground class increases for $TRANS_{10}$, but then slowly decreases when introducing more and more data for fine-tuning. Still, the F1-scores of all TRANS methods remain above the score for ALS_{train}. The overall F1-score and accuracy in Table 3 is also affected and decreases with higher ratios of DIM data due to its correlation with the ground and non-ground class, which contribute around 91% of all pixels in DIM_{test} (see Table 1). Consequently, the overall accuracy decreases by 6% from $TRANS_{10}$ to $TRANS_{40}$.

By comparing the confusion matrix of ALS_{train} and $TRANS_{10}$, the consequences when introducing DIM data for transfer learning are shown in Table 4 and 5. Since ALS_{train} only contains 0.73% water pixels (Table 1), introducing DIM data especially boosts the accuracy of water from 2.38% in ALS_{train} to 49.61% in $TRANS_{10}$ in the confusion matrix (Table 4, 5). However, there are still some misclassifications of water pixels left, which are classified as ground or non-ground instead. In addition, the accuracy of non-ground pixels increases from 62.91% in ALS_{train} to 97.75% in $TRANS_{10}$. Despite these improvements, $TRANS_{10}$ falsely classifies buildings as non-ground, which decreases the building accuracy by 40% notably. Still, the overall accuracy of $TRANS_{10}$ increases due to the imbalance between building and non-ground class in the training sets.

Point cloud	No data	Ground	Building	Water	Non-ground
ALS_{train}	98.6	92.1	38.3	1.6	73.8
$TRANS_{10}$	64.6	91.2	60.7	65.1	87.3
$TRANS_{20}$	58.5	87.7	68.1	59.3	84.8
$TRANS_{30}$	65.8	85.9	71.4	63.7	83.9
$TRANS_{40}$	75.1	85.5	60.7	71.7	80.9
$TRANS_{50}$	82.6	96.1	58.9	66.3	89.0
DIM_{train}	99.3	98.6	86.5	89.3	96.6

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

Point cloud	Overall accuracy	Overall F1
ALS_{train}	74.7	78.7
$TRANS_{10}$	87.3	87.1
$TRANS_{20}$	84.6	84.4
$TRANS_{30}$	83.6	83.5
$TRANS_{40}$	81.4	81.8
$TRANS_{50}$	90.0	90.2
DIM_{train}	96.8	96.8

Table 3. Overall accuracy and overall F1-Score for DIM_{test} [%]

		Predicted				
		No data	Ground	Building	Water	Non-ground
Reference	No data	99.72	0.27	0.01	0.00	0.00
	Ground	0.00	87.69	0.00	10.71	1.61
	Building	0.00	0.01	91.23	0.00	8.76
	Water	0.21	38.02	0.47	2.38	58.92
	Non-ground	0.00	0.38	34.85	1.86	62.91

Table 4. Confusion matrix of ALS_{train} [%]

		Predicted				
		No data	Ground	Building	Water	Non-ground
Reference	No data	47.78	34.67	0.00	6.03	11.52
	Ground	0.00	84.92	0.00	0.04	15.04
	Building	0.00	0.00	51.24	0.00	48.76
	Water	0.02	17.69	0.98	49.61	31.70
	Non-ground	0.00	0.04	2.07	0.14	97.75

Table 5. Confusion matrix of $TRANS_{10}$ [%]

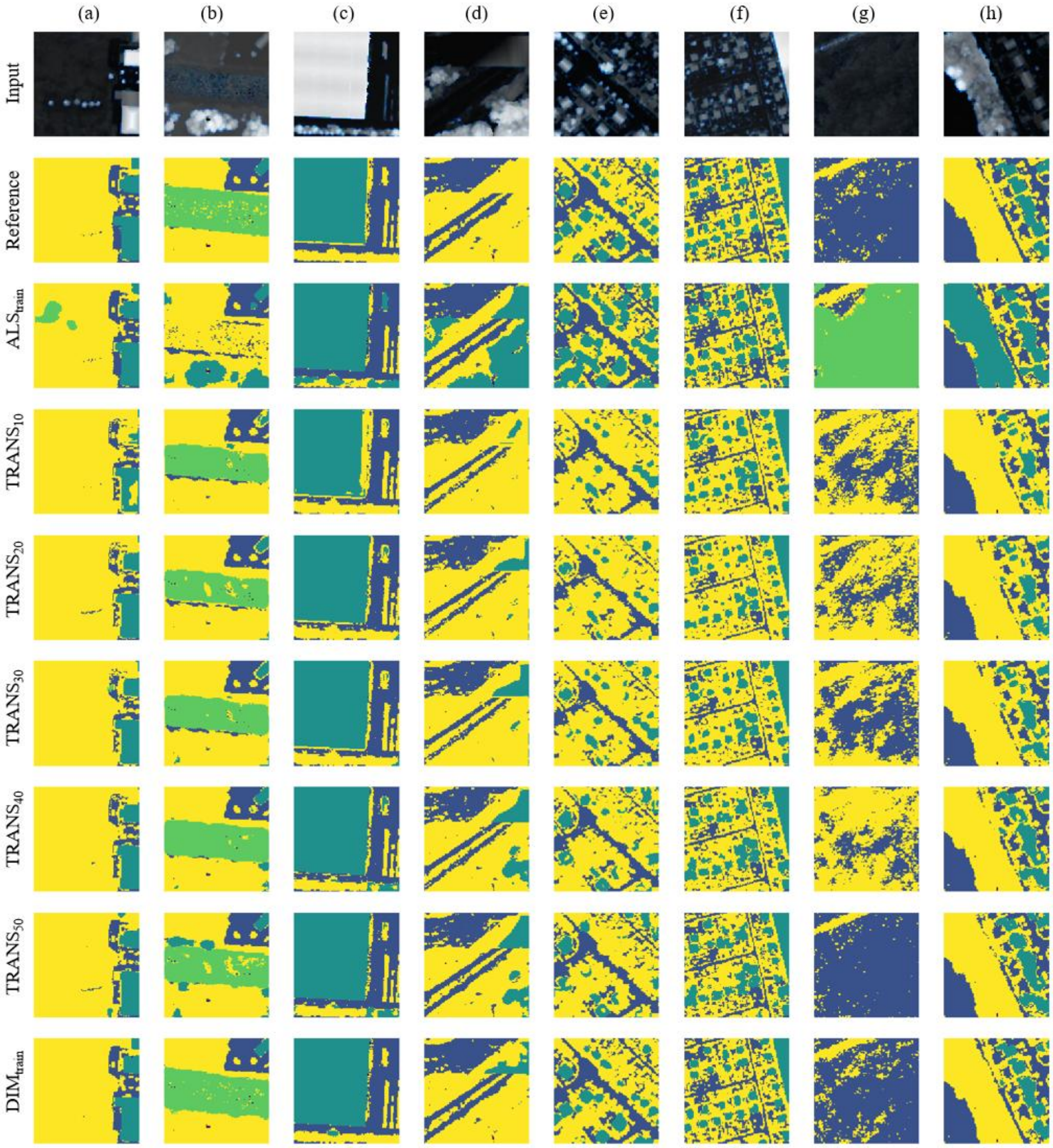

Figure 3. Qualitative results of the trained network setups in comparison for DIM test data. The input height images are rendered as normalized RGB image. The printed classes are ground (dark blue), building (petrol), water (light green) and non-ground (yellow), respectively.

4.2.2 Qualitative results: In order to compare our results qualitatively, we randomly picked eight samples from DIM$_{test}$ and present their input, reference data as well as the predictions of all setups in Figure 3. Each column shows the results for one sample. The height images of DIM$_{test}$ are the input of the trained networks and are plotted as RGB images, which are normalized to the interval [0, 1]. As reference, the class of the highest point within a raster cell is selected as described in section 3.1.3. The remaining rows show the predictions for all setups.

In general, all predictions visually confirm the results in their respective F1-scores and the overall accuracy. In most cases, all setups classify ground and non-ground pixels correctly. However, if the ground surface is rather rough (g), the networks of ALS$_{train}$ and TRANS$_{10}$ to TRANS$_{40}$ mistake ground for non-ground. Since ALS data contains only a small amount of water pixels, the network ALS$_{train}$ hardly classifies water in real water bodies (b), but on randomly located spots on ground level (a, g). This issue is fixed when introducing DIM data in all TRANS setups as well as in DIM$_{train}$. Similarly, introducing DIM data into the training process improves another issue in ALS$_{train}$. In both point clouds, z_{min}, z_{mean} and z_{max} values are quite similar for ground and building points. The normalised height value separates building from ground points in this case. Nevertheless, these three values differ a lot for vegetation in ALS such as trees, since z_{min} still captures ground information, while z_{max} is based on points in the treetop or on branches. In DIM data however, the difference between all three values for vegetation is much smaller than in ALS data as it mainly represents the tree canopy. Consequently, ALS_{train} mistakes non-ground for building whenever a tree has a smooth treetop (b, d, e, h).

However, there are still some unsolved issues within the predictions. In contrast to the flat huge building in (c), which is classified correctly by all network setups, the underpass in (d) causes trouble for all setups. Due to its flat surface, it is often recognized as building class instead of the correct non-ground class (d). In addition, all transfer learning setups have problems classifying small buildings at all or the complete shape of normal size buildings.

5. DISCUSSION

In this section, we critically discuss our proposed method as well as possible improvements for future work.

Testing a network, which was trained on ALS data, on DIM data achieved an F1-score of 92% for the ground, 74% for the non-ground, 38% for the building and only 2% for the water class (Table 3). Incorporating only 10% of newly labelled DIM data in the training process improved the classification results of non-ground, water and building class notably.

As the water class was hardly represented in ALS_{train} with only 0.7% in the class distribution (see Table 1), introducing more water pixels in $TRANS_{10}$ reduced the misclassifications as ground and non-ground by more than 20% in the confusion matrix (Table 4 and 5). Similarly, ALS_{train} often classifies smooth tree canopy in DIM_{test} as building instead of non-ground due to the different characteristics in both point cloud types (Figure 3). Introducing DIM data reduced this misclassification by 30% in $TRANS_{10}$ (Table 4 and 5). Consequently, incorporating 10% of DIM data into the training already results in an increase of the overall F1-score from 79% to 87% (Table 3). However and as expected, none of the networks, which applied transfer learning, achieved classification results close to DIM_{train}. There are several options to further improve our transfer learning approach.

Possible solutions for the misclassifications, which origin in the different class distributions, are either balancing the class distribution in the input data or by weighting the classes differently in the loss function, e.g. using the focal loss (Lin et al., 2017). In addition, weighting the loss value depending on each class distribution also could resolve the need for the no data class. As no data pixels could receive a weight of zero, the neurons, which are dedicated to the no data class, could be utilized for other classes.

The usage of minimal and maximal values may support classifying noise rather than real objects. This may not be an issue with a filtered point cloud, but can potentially cause some unexpected behavior of the network and its classification results. An alternative to the minimal and maximal value could be some other statistics for points below and above the mean height within a raster cell or by just taking e.g. the 10% highest and lowest point instead of the extreme values (Gevaert et al. 2018). Decreasing the raster cell size will also reduce the amount of raster cells with mixed objects and thus improve the overall classification. In addition, the classification could be split into two parts: the first part uses a 2D raster to gather global information as described in this paper and the second part aggregates the points with this global information for a point based classification similar to the idea of Qi et al. (2017a).

Finally, instead of requiring a DTM in order to achieve height above ground, we would like to find a replacement, which only requires the point cloud itself. This could be accomplished using a hierarchical classification, where the point cloud is first classified into ground and non-ground and then further specified into more classes similar to Rizaldy et al. (2018). In this case, the ground height could be integrated into the classification of non-ground points. Alternatively, the ground surface could be approximated using a local minimum within a certain radius or by some rules (Rizaldy et al., 2018; Gavaert et al., 2018).

The results of this study can lead to adapted workflows in the NMAs to adjust the amount of training data for their classifications, as now the degradations in quality when using less information have been quantified.

6. CONCLUSION

In this work, we focused on transfer learning from ALS to DIM point cloud data. We restricted the approach to exclusively using the geometry of the points, since they are part of both point cloud types, and we projected the point clouds into a 2D grid to deal with different point densities. As input for an encoder-decoder CNN, we calculate the minimal, mean and maximal point height within a raster cell. Since labelling training data is expensive and time-consuming, we fine-tuned an encoder-decoder CNN, which was trained on ALS data, in different setups using an increasing amount of newly added and labelled data. These setups are compared to the initial ALS based network as well as to a network, which was trained only on DIM data. When tested on DIM data, our results show that the classification result improves notable for a transfer learned network compared to a model, which was only trained on ALS data. As expected, none of our transfer learned models could accomplish the classification quality from the network, which was completely trained on DIM point cloud data. However, we show that already 10% of labelled DIM data increase the classification results notably, which is especially relevant for practical applications.

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

AHN3, 2019: The Actueel Hoogtebestand Nederland (AHN) dataset, https://www.pdok.nl/nl/ahn3-downloads.

Engelmann, F., Kontogianni, T., Hermans, A. and Leibe, B., 2017: Exploring Spatial Context for 3D Semantic Segmentation of Point Clouds. *Workshop paper of ICCVW 2017.*

Gevaert, C.M., Persello, C., Nex, F. and Vosselman, G., 2018: A deep learning approach to DTM extraction from imagery using rule-based training labels. In: *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 142, pp. 106-123.

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

Huang, J & You, S, 2016: Point Cloud Labeling using 3D Convolutional Neural Network. *Proceedings of ICPR 2016*.

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

Kingma, D.P. and Ba, J.L., 2015: Adam: A method for stochastic optimization. *Proceedings of ICLR 2015*.

Landrieu, L. and Simonovsky, M., 2018: Large-scale Point Cloud Semantic Segmentation with Superpoint Graphs. *Proceedings of CVPR 2018*.

Lin, T., Goyal, P., Girshick, R-. He, K. and Dollár, P., 2017: Focal Loss for Dense Object Detection. *Proceedings of ICCV*.

Mandlburger, G., Wenzel, K., Spitzer, A., Haala, N., Glira, P. and Pfeifer, N., 2017: IMPROVED TOPOGRAPHIC MODELS VIA CONCURRENT AIRBORNE LIDAR AND DENSE IMAGE MATCHING. In: *Int. Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, Vol. IV-2/W4, pp. 259-266.

Mao, X.J., Shen, C. and Yang, Y.B., 2016: Image Restoration Using Very Deep Convolutional Encoder-Decoder Networks with Symmetric Skip Connections. *Proceedings of NIPS 2016*.

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. In: *Technometrics*, Vol. 21 (2), pp. 239-245.

Politz, F. and Sester, M., 2018: EXPLORING ALS AND DIM DATA FOR SEMANTIC SEGMENTATION USING CNNS. In: *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, Vol. XLII-1. pp. 347-354.

Qi, C., Su, H., Mo, K. and Guibas, L.J., 2017a: PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. *Proceedings of CVPR 2017.*

Qi, C., Yi, L., Su, H. and Guibas, J., 2017b: PointNet++: Deep Hiarchical Feature Learning on Point Sets in a Metric Space. *Proceedings of NIPS 2017*.

Rizaldy, A., Persello, C., Gevaert, C., Elberink, S.O. and Vosselmann, G., 2018: Ground and Multi-Class Classification of Airborne Laser Scanner Point clouds Using Fully Convolutional Networks. In: *Remote Sensing*, Vol. 10 (11), 1723.

Ronneberger, O., Fischer. P and Brox, T., 2015: U-Net: Convolutional Networks for Biomedical Image Segmentation. In: *Medical Image Computing and Computer-Assisted Intervention*, Springer, LNCS, Vol. 9351, pp. 234-241.

Rothermel, M., Wenzel, K., Fritsch D. and Haala, N., 2012. SURE: Photogrammetric Surface reconstruction from Imagery. *Proceedings of LC3D Workshop*, Berlin 2012, http://www.ifp.uni-stuttgart.de/publications/software/sure/index.en.html (December 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.

Tchapmi, L.P., Choy, C.B., Armeni, I., Gwak, J. and Savarese, S., 2017: SEGCloud: Semantic Segmentation of 3D Point Clouds. *Spotlight paper of the International Conference of 3D Vision (3DV) 2017*.

Te, G., Hu, W., Zheng, A. and Guo, Z., 2018: RGCNN: Regularized Graph CNN for Point Cloud Segmentation. *Proceedings of ACM*, pp. 746-754.

Toschi, I., Remondino, F., Rothe, R. and Klimek, K., 2018: COMBINING AIRBORNE OBLIQUE CAMERA AND LIDAR SENSORS: INVESTIGATION AND NEW PERSPECTIVES. In: *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, Vol. XLII-1 pp. 437-444.

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

Xu, Z. and Yang, Z., 2018: EIGENENTROPY BASED CONVOLUTIONAL NEURAL NETWORK BASED POINT CLOUDS CLASSIFICATION METHOD. In: *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.*, Vol. XLII-3, pp. 2017-2022.

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*, Vol 9 (9), 936.

Yousefhussien, M., Kelbe, D. J., Ientilucci, E. J. and Salvaggio, C., 2018: A multi-scale fully convolutional network for semantic labeling of 3D point clouds. In: *ISPRS Journal of Photogrammetry and Remote Sensing*, Vol. 143, pp. 191-204.

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.

Zhao, R., Pang, M. and Wang, J., 2018: Classifying airborne LiDAR point clouds via deep features learned by a multi-scale convolutional neural network. In: *International Journal of Geographical Information Science*, Vol. 32 (5), pp. 960-979.