Large-scale mapping of small roads in lidar images using deep convolutional neural networks

Arnt-Børre Salberg[1], Øivind Due Trier[1], and Michael Kampffmeyer[2]

[1] Norwegian Computing Center, PO-Box 114 Blindern, NO-0314 Oslo
salberg@nr.no,trier@nr.no
[2] UiT - The Arctic University of Norway, NO-9037 Tromsø
michael.c.kampffmeyer@uit.no

Abstract. Detailed and complete mapping of forest roads is important for the forest industry since they are used for timber transport by trucks with long trailers. This paper proposes a new automatic method for large-scale mapping forest roads from airborne laser scanning data. The method is based on a fully convolutional neural network that performs end-to-end segmentation. To train the the network, a large set of image patches with corresponding road label information are applied. The final network is then applied to detect and map forest roads from lidar data covering the Etnedal municipality in Norway. The results show that we are able to map the forrest roads with an overall accuracy of 97.2%. We conclude that the method has a strong potential for large-scale operational mapping of forest roads.

Keywords: Deep learning, convolutional neural networks, lidar, remote sensing

1 Introduction

In 2015, Norway officially decided to collect airborne laser scanning (ALS) data for the entire land area below the timber line. The point density will be at least two first returns per square metre, with the main purpose to obtain a very detailed digital terrain model (DTM) of the entire country. For open areas above the tree line, i.e., in the mountains, the DTM will be based on automatic image matching from aerial photography.

The national coverage of ALS data provides large opportunities for new mapping products, e.g. maps of small roads like forest roads that are difficult to observe in optical remote sensing images. Forest roads are used for timber transport by trucks with long trailers, and due to the forest industry's demands for profitable management, accurate, detailed and complete mapping of forest roads is important.

Remote sensing imagery is often characterized by complex data properties in the form of heterogeneity and class imbalance, as well as overlapping class-conditional distributions [3]. Together, these aspects constitute severe challenges for creating land cover maps or detecting and localizing objects, producing a

high degree of uncertainty in obtained results, even for the best performing model [13, 16].

Automatic detection and mapping of road networks from remote sensing data has previously been studied by several authors [7, 22], however, most of the work focus on optical data [22], and current state-of-the art algorithms fail to extract roads in optical images for cases where surrounding objects like water, buildings, trees, grass and cars occlude the road or cast shadows, especially with influence of spatial structures such as overpasses [22].

In recent years, deep convolutional neural networks (deep CNNs) have emerged as the leading modelling tools for image pixel classification and segmentation in general [8, 14], and have had an increasing impact also in remote sensing [12, 13, 16, 17, 19]. This increasing interest is reflected for example in the ISPRS semantic segmentation challenge [11], where deep CNNs are dominating and are shown to provide the best performing models.

In practice, there are currently two main approaches to performing image segmentation using CNNs. The first one, which we refer to as patch-based, relies on predicting every pixel in the image by looking at the enclosing region of the pixel. This is commonly done by training a classifier on small image patches and then either classify all pixels using a sliding window approach, or more efficiently, converting the fully connected layers to convolutional layers [20], thereby avoiding overlapping computations. Further improvements may be achieved by using multi-scale approaches or by iteratively improving the results in a recurrent CNN [6, 18].

The second approach is based on the idea of pixel-to-pixel semantic segmentation using end-to-end learning [14]. It uses the idea of a fully convolutional network (FCN), consisting of an encoder and a decoder. The encoder is responsible for mapping the image to a low resolution representation, whereas the decoder provides a mapping from the low resolution representation to the pixel-wise predictions. Up-sampling is achieved using fractional-strided convolutions [14]. This approach has recently improved the state-of-the art performance on many image tasks and, due to the lack of fully-connected layers, allows pixel-wise predictions for arbitrary image sizes.

In this paper we build upon the work by Mnih and Hinton [15], who applied a neural network to detect roads in very high resolution optical remote sensing data, and hypothesize that we can train a deep convolutional neural network and apply it to perform automatic mapping of small roads in lidar data. To address the hypothesis we rely on state-of-the-art fully convolutional neural networks, tailored to perform semantic mapping [14], also with good results on remote sensing images [12].

2 Data

ALS data of the majority of Etnedal municipality, Oppland County, Norway, have been captured with an average of 6.5 ground hits per m^2. However, this varies from 0 (below dense canopies of deciduous trees) to $20/m^2$ at strip over-

laps. For the entire Etnedal municipality, vector data in the form of ESRI shape files, containing the current official mapping of road centre lines and road area, are available.

2.1 Pre-processing

The ALS data consisted of point measurements (x, y, z) in UTM zone 32. Each point had a number of attributes, including:

1. Class (one of: ground, vegetation, building, other)
2. Return number, a number between 1 and 4 where 1 denotes the first return and 4 the last return.
3. Return intensity (uncalibrated radiance)

From these attributes, the following images were generated:

1. Digital terrain model (DTM) from all ALS points labelled as ground
2. Elevation gradient of DTM, measured in degrees
3. Digital surface model (DSM) from all ALS points labelled as first returns

The pilot study [21] and further investigations indicated that the gradient image had the best potential for automatic detection of roads, compared with alternative representations of the ALS data. The other representations included: laser return intensity image, hill-shaded relief image, local relief image, aspect direction image, and the ALS point cloud of ground returns.

Two resolutions for the DTM, and thus, the gradient image, were evaluated. Although 0.2 m gives slightly better detail than 0.5 m whenever there are multiple ALS ground returns within a 0.5 m pixel, the increased data volume has a negative impact on the deep learning method. Smaller areas, measured in m^2, are input to the vision methods of the deep neural network, described below, meaning that the context of a road may be lost in the vision task. Also, with $6.5/m^2$ ALS ground point density on average, not much detail is lost (on average) when reducing the resolution from 25 pixels per m^2 to 4 pixels per m^2, corresponding to 0.2 m and 0.5 m pixel sizes, respectively. Another benefit is that the number of pixels is reduced by a factor of 6.25.

The gradient image of Etnedal municipality was divided into two sets, one for training and one for testing. Form each data set 256×256 image patches with 50% overlap were extracted. For the training and test dataset, only image patches that contain road segments were used. A total of 59004 images of size 256×256 pixels were available. This was divided into two equal sized datasets, one for training and one for test. 10% of the training images were used as validation data.

3 Automatic detection of roads

We applied the same FCN architecture as Kampffmeyer et al. [12], which allowed end-to-end learning of pixel-to-pixel semantic segmentation. The network

was implemented on a graphical processing unit (GPU), in order to speed up computations. The network was trained in mini-batches on patches of 256×256 pixels. The patch size was chosen due to GPU memory considerations.

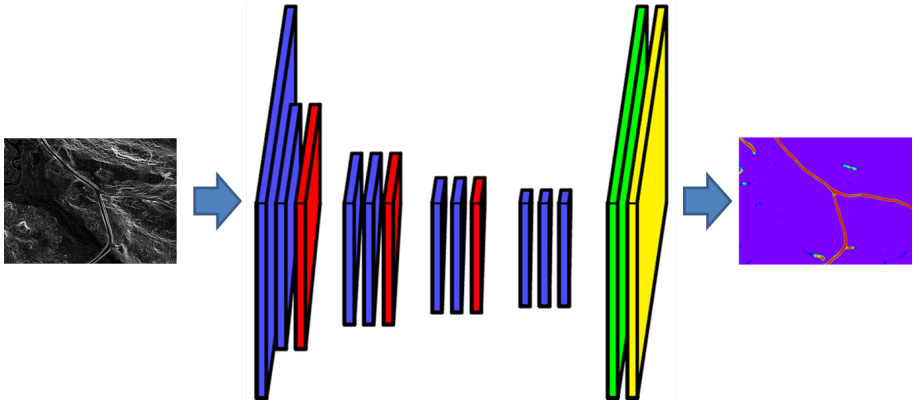

Fig. 1. Pixel-to-pixel architecture. Blue layers represent convolutional layers (including ReLU and batch-normalization layer), red layers represent pooling layers, the green layer represents the fractional-strided convolution layer and the yellow layer the softmax layer.

Architecture The CNN architecture of the FCN network (Fig. 1) consisted of four sets of two 3×3 convolutions (blue layers), each set separated by a 2×2 max pooling layer with stride 2 (red layers).

All convolution layers have a stride of 1, except the first one, which has a stride of 2. The change in the first convolution layer was a design choice, which was mainly made due to limits in GPU memory during test phase when considering large images. All convolutional layers were followed by a ReLU nonlinearity and a batch normalization layer [10]. Weights were initialized according to He et al. [9]. The final 3×3 convolution was followed by a 1×1 convolution, which consisted of one kernel for each class to produce class scores. The convolutional layers were followed by a fractional-strided convolution layer [14] (green layer in Fig. 1, sometimes also referred to as deconvolution layer), which learned to upsample the prediction back to the original image size, and a softmax layer (yellow layer in Fig. 1). The network was trained end-to-end using backpropagation

Data augmentation The image patches were extracted from the input image with 50% overlap and were flipped (left to right) and rotated at 90 degree intervals, yielding 8 augmentations per image patch.

Median frequency balancing Training of the FCN network was done using the cross-entropy loss function. However, as this loss was computed by summing over all the pixels, it did not account well for imbalanced classes. To take the imbalanced classes into account, the loss of the classes was weighted using median frequency balancing [1, 5, 12]. Median frequency balancing weights the class loss by the ratio of the median class frequency in the training set and the actual class frequency. The modified cross-entropy function is

$$L = -\frac{1}{N}\sum_{n=1}^{N}\sum_{c\in\mathcal{C}} \ell_c^{(n)} \log\left(\hat{p}_c^{(n)}\right) w_c, \tag{1}$$

where N is the number of samples in a mini-batch,

$$w_c = \frac{\text{median}\,(f_c | c \in \mathcal{C})}{f_c} \tag{2}$$

is the class weight for class c, f_c the frequency of pixels in class c, $\hat{p}_c^{(n)}$) is the softmax probability of sample n being in class c, $\ell_c^{(n)}$ corresponds to the label of sample n for class c when the label is given in one-hot encoding and $\mathcal{C}$ is the set of all classes.

3.1 Pre-processing and post-processing

Merging output probabilities and class image In test mode each 256 × 256 image was augmented by 90 degree rotation and flipping as described in Section 3, and sent through the CNN. The output of the CNN for a given image was a score map for each class. The score maps for each rotation and flip is rotated backwards, and merged by averaging. From the averaged score image, the class image was computed by, for each pixel, selecting the class with the largest score.

Merging of classification result The neural network outputs classified images of size 256 × 256, based on input images of the same size. In order to avoid edge effects in the merged classification result, the input images are generated with 50% overlap between any two neighbouring images vertically or horizontally. In other words, subimages of size 256 × 256 pixels are generated with 128 pixels step size from the gradient image.

The classified images of size 256 × 256 pixels contain the values 1 (road) and 0 (background). These images are then cropped to size 128 × 128 pixels by removing pixels that are less than 64 pixels from the edge. These cropped images of size 128 × 128 pixels are then merged edge-to-edge to form the full classification map.

4 Results

The average classification accuracies of using the FCN approach to classify the validation dataset were

- Non-road: 97.2%
- Road: 95.3%
- Overall: 97.2%

Please note that the overall accuracy is equal to the non-road accuracy. This is due to the high class imbalance between the road and non-road classes.

The automatic road detection method produced results that are not perfect. However, when comparing with the existing road centre lines, the automatic mapping often produced more accurate centre lines. For example, a gap in the existing tractor road centre line was closed by the automatic method (Fig. 2), the existing tractor road centre line ran outside of the tractor road at some curves, whereas the automatically generated centre lines stay inside the tractor road (Fig. 3), and there was no existing tractor road (or path) centre line at the detected location (Fig. 4).

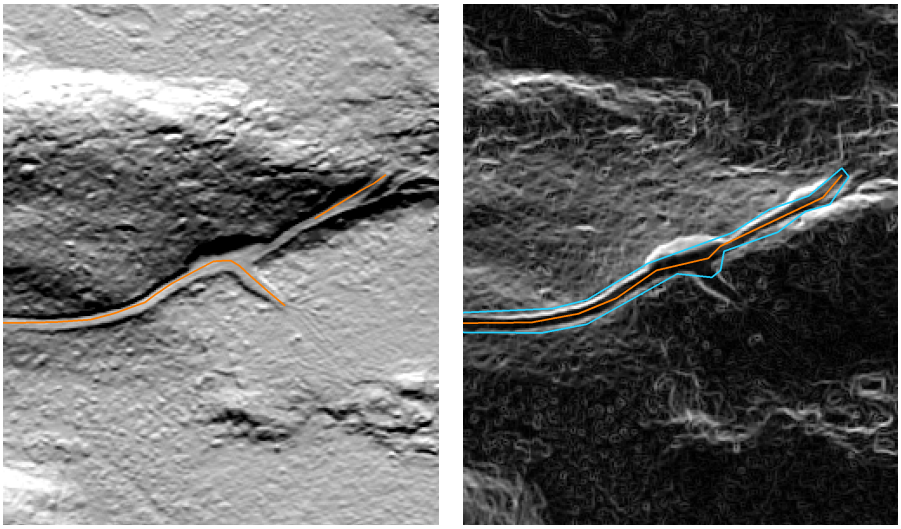

Fig. 2. There is a gap in the original centre line (left, orange) that is closed in the automatically detected centre line (right,orange) The cyan outline indicates the detected road area.

There are also examples of situations where the automatic method had problems. For example, a road that is difficult to see in the gradient image (Fig. 5) may be missed. A road crossing a field (Fig. 6), may result in fragmented mapping. Some terrain features, e.g. two parallel ditches (Fig. 7) may result in a false road.

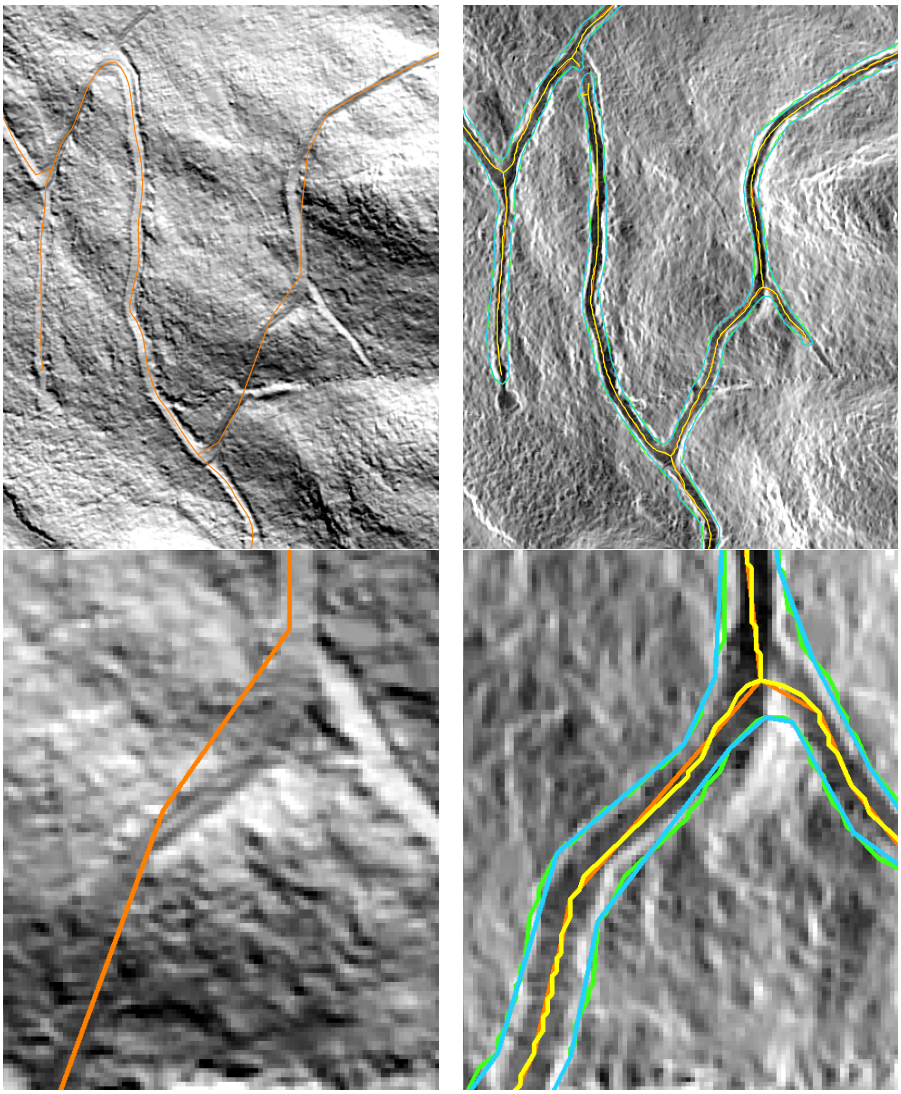

Fig. 3. Left: Existing tractor road centre line, with hill-shaded DTM. Right: road centre line and outline from automatic method, with gradient of DTM. Yellow/green: 0.25 m maximum displacement in point reduction. Orange/cyan: 1.0 m maximum displacement.

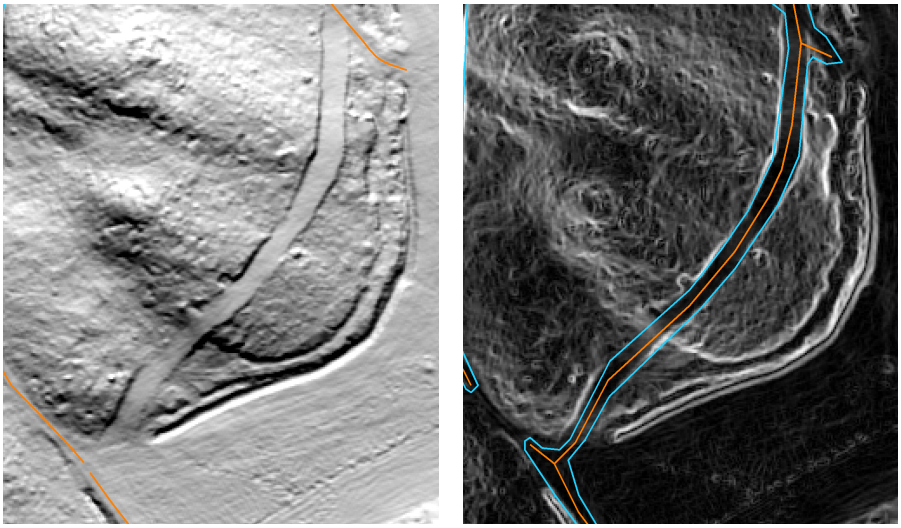

Fig. 4. A road that was mapped by the automatic method (right), but missing in the existing vector data (left).

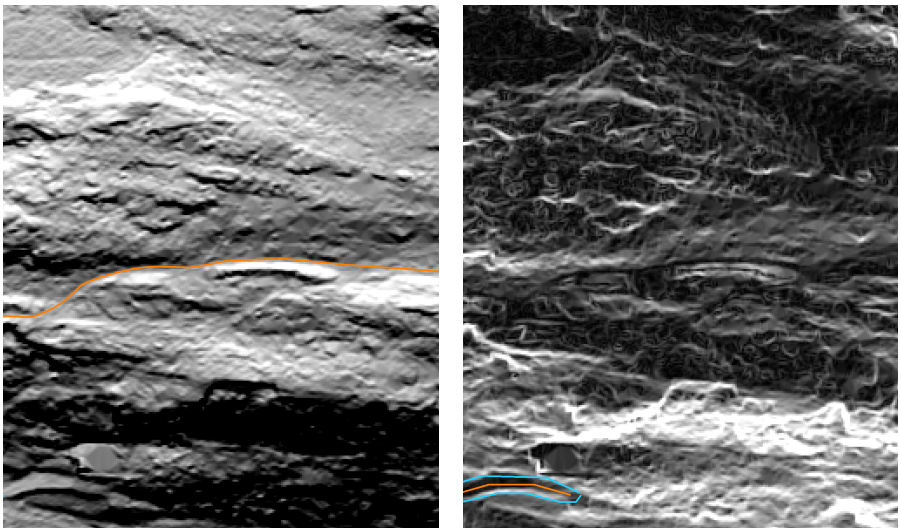

Fig. 5. Tractor road that was missed by the automatic method. Left: existing road centre line. Right: the road is difficult to see in the gradient image.

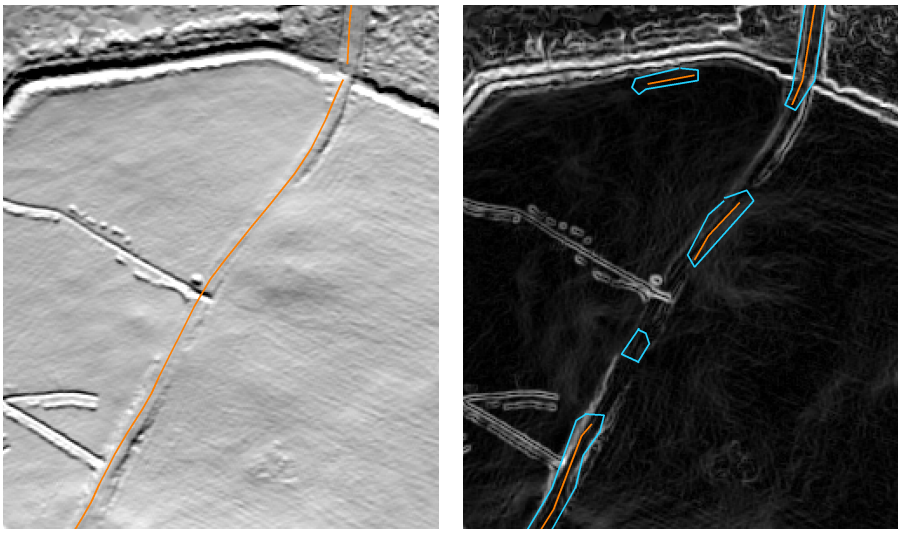

Fig. 6. Tractor road crossing a field. Left: existing centre line. Right: result of automatic detection.

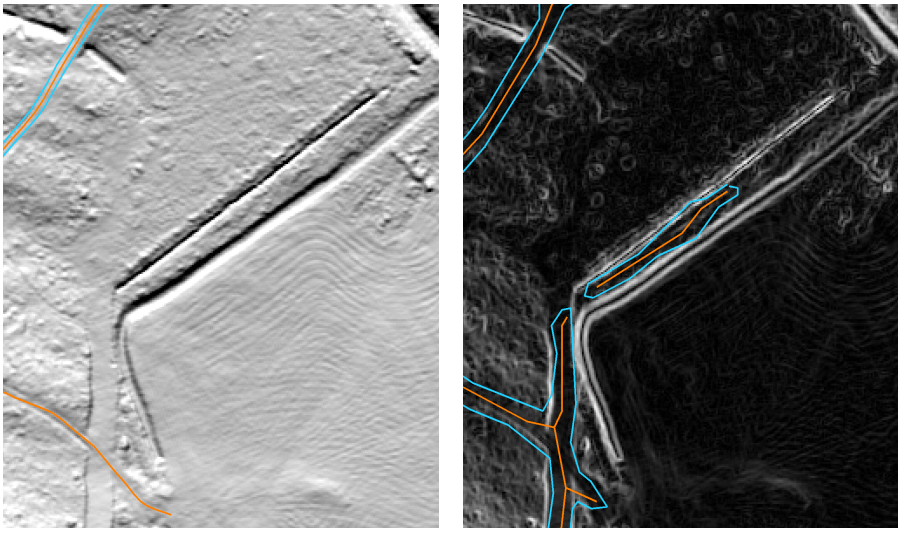

Fig. 7. False automatic detection of road.

5 Discussion

Even though the deep fully convolutional neural network provides very good results for mapping forest roads in lidar data, there is a potential to improve the approach. Adding more training data often helps to improve the performance of deep neural networks. More data also provides you with the opportunity to increase the network size and thereby its modelling capabilities. In terms of network architectures, the topology aware FCN proposed by BenTaieb and Hamarneh [2] is one promising method that should be investigated. Another approach is to use a conditional random field (CRF) based post-processing.

Pre-trained networks, e.g. Alexnet, VGGnet or GoogleNet, in combination with fine-tuning, could also be applied as part of the FCN [14]. The use of pre-trained networks has become a standard technique in computer vision and may provide a performance gain, in particular if we have a limited number of training images.

As a post-processing step for the automatic road detection method, a point reduction method [4], or a method that is good in replicating the curvatures of actual roads, may be used. Clearly, there is a lower limit on the radius of a turn. This radius may be measured at any vertex by finding the circle arc that passes through that vertex, the preceding vertex and the succeeding vertex.

Another alternative could be to grow the skeleton image (by creating a distance map with a maximum distance limit) and then to re-create the skeleton image by thinning a thresholded distance map. This may produce a smoother skeleton image. However, the skeleton image will always result in a vectorised result with only eight possible directions (multiples of 45 degrees), so a smoothing or point reduction of the vector data is always needed.

Training of the detection method was done on a subset of the Etnedal dataset. There is always a trade-off between training and testing. If the training data set is too small, then the method may be over-fitted on the training data and may produce bad results on other areas. E.g., if the training data only includes steep hillsides with roads with many turns, then the method may be bad at detecting straight roads in flat terrain, and vice versa. However, if the method is trained on representative parts of all of Norway, then the method may be bad at making local adaptions. So, a solution may be to run training or fine-tuning with existing road centre line data for each dataset, and then run automatic detection on the same dataset, or combine the results of a model for whole Norway with the results of from a local model. In both cases, the result may be improved centre lines in those parts of the terrain where the original centre lines were inaccurate or missing. Further, it could be interesting to compute quality metrics by comparing the new centre lines with the existing:

1. For all roads where there is an old and a new centre line, what is the average deviation between the vertices of the new centre line and the corresponding closest locations on the old centre line?
2. How many metres of new road centre lines do not match an existing road centre line?

3. How many metres of existing road centre line do not have a corresponding new road centre line?

6 Conclusions

In this paper, we have demonstrated that end-to-end segmentation using a fully convolutional neural networks provides very good results in terms of mapping forests roads in lidar data. Do to these promising results, we conclude that deep neural network methods provides a good basis for designing algorithms for large scale mapping of roads, but also other objects like e.g. cultural heritages, in lidar data.

Acknowledgments Part of this research was financed by the Norwegian Mapping Authority, Hamar regional office, which also provided vector data. Airborne laser scanning data was provided by Oppland County Administration.

References

1. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. arXiv preprint arXiv:1511.00561 (2015)
2. BenTaieb, A., Hamarneh, G.: Topology aware fully convolutional networks for histology gland segmentation. In: Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II. pp. 460–468. Springer International Publishing, Cham (2016)
3. Camp-Valls, G., Bruzzone, L.: Kernel Methods for Remote Sensing Data Analysis / Edited by Gustavo Camps-Valls, Lorenzo Bruzzone. Wiley, Chichester, UK (2009)
4. Douglas, D.H., Peucker, T.K.: Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. Cartographica: The International Journal for Geographic Information and Geovisualization 10(2), 112–122 (1973)
5. Eigen, D., Fergus, R.: Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: Proc. IEEE Int. Conf. Computer Vision. pp. 2650–2658 (2015)
6. Farabet, C., Couprie, C., Najman, L., LeCun, Y.: Learning hierarchical features for scene labeling. IEEE Trans. Pattern Anal. Machine Intell. 35(8), 1915–1929 (2013)
7. Ferraz, A., Mallet, C., Chehata, N.: Large-scale road detection in forested mountainous areas using airborne topographic lidar data. ISPRS J. Photogramm. Remote Sensing 112, 23–36 (2016)
8. Hariharan, B., Arbelez, P., Girshick, R., Malik, J.: Hypercolumns for object segmentation and fine-grained localization. In: Proc. IEEE Conf. Computer Vision Pattern Recognition. pp. 447–456 (2015)
9. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proc. IEEE Int. Conf. Computer Vision. pp. 1026–1034 (2015)

10. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)
11. ISPRS: ISPRS 2D Semantic Labeling Contest. `http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html` (2015)
12. Kampffmeyer, M., Salberg, A.B., Jenssen, R.: Semantic segmentation of small objects and modeling of uncertainty in urban remote sensing images using deep convolutional neural networks. In: Proc. IEEE Conf. Computer Vision Pattern Recognition Workshops. pp. 1–9 (2016)
13. Lagrange, A., Saux, B.L., Beaupre, A., Boulch, A., Chan-Hon-Tong, A., Herbin, S., Randrianarivo, H., Ferecatu, M.: Benchmarking classification of earth-observation data: From learning explicit features to convolutional networks. In: 2015 IEEE Int. Geoscience Remote Sensing Symposium (IGARSS). pp. 4173–4176 (2015)
14. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proc. IEEE Conf. Computer Vision Pattern Recognition. pp. 3431–3440 (2015)
15. Mnih, V., Hinton, G.E.: Learning to detect roads in high-resolution aerial images. In: Computer Vision – ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part VI. pp. 210–223. Springer Berlin Heidelberg", Berlin, Heidelberg (2010)
16. Paisitkriangkrai, S., Sherrah, J., Janney, P., Hengel, A.: Effective semantic pixel labelling with convolutional networks and conditional random fields. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition Workshops. pp. 36–43 (2015)
17. Penatti, O.A.B., Nogueira, K., dos Santos, J.A.: Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? In: Proc. IEEE Conf. Computer Vision Pattern Recognition Workshops. pp. 44–51 (2015)
18. Pinheiro, P., Collobert, R.: Recurrent convolutional neural networks for scene parsing. arXiv preprint arXiv:1306.2795 (2013)
19. Salberg, A.B.: Detection of seals in remote sensing images using features extracted from deep convolutional neural networks. In: 2015 IEEE Int. Geoscience Remote Sensing Symposium (IGARSS). pp. 1893–1896 (2015)
20. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: OverFeat: Integrated recognition, localization and detection using convolutional neural networks. In: Int. Conf. Learning Representations (ICLR). CBLS, Banff, Canada (April 2014)
21. Trier, Ø.D.: Evaluation of methods for detection of roads in laser data - preliminary results. LasTrak pilot project (in norwegian). NR-Note SAMBA/09/15, Norwegian Computing Center, Oslo (2015)
22. Wang, W., Yang, N., Zhang, Y., Wang, F., Cao, T., Eklund, P.: A review of road extraction from remote sensing images. Journal of Traffic and Transportation Engineering (English Edition) 3(3), 271–282 (2016)