SEMANTIC SEGMENTATION OF MANMADE LANDSCAPE STRUCTURES IN DIGITAL TERRAIN MODELS

B. Kazimi*, F. Thiemann, M. Sester

Institute of Cartography and Geoinformatics, Leibniz Universität Hannover, Germany
- (kazimi, thiemann, sester)@ikg.uni-hannover.de

KEY WORDS: Object Detection, Semantic Segmentation, Digital Terrain Models, Laser Scanning, Deep Learning

ABSTRACT:

We explore the use of semantic segmentation in Digital Terrain Models (DTMS) for detecting manmade landscape structures in archaeological sites. DTM data are stored and processed as large matrices of depth 1 as opposed to depth 3 in RGB images. The matrices usually contain continuous real-valued information upper bound of which is not fixed, such as distance or height from a reference surface. This is different from RGB images that contain integer values in a fixed range of 0 to 255. Additionally, RGB images are usually stored in smaller multidimensional matrices, and are more suitable as inputs for a neural network while the large DTMs are necessary to be split into smaller sub-matrices to be used by neural networks. Thus, while the spatial information of pixels in RGB images are important only locally within a single image, for DTM data, they are important locally, within a single sub-matrix processed for neural network, and also globally, in relation to the neighboring sub-matrices. To cope with the two differences, we apply min-max normalization to each input matrix fed to the neural network, and use a slightly modified version of DeepLabv3+ model for semantic segmentation. We show that with the architecture change, and the preprocessing, better results are achieved.

1. INTRODUCTION

Airborne Laser Scanning (ALS) is an efficient remote sensing technique used to collect data on large areas by measuring the range and reflectance of objects on their surface. Raw ALS data are stored as point clouds, and have information on non-terrain objects such as buildings and trees while a rasterized product of ALS data, Digital Terrain Model (DTM), is a filtered version in which only the information on the terrain points are preserved. DTM data could be leveraged to identify structures on the terrain. Previously, classical machine learning algorithms were used to detect structures and identify objects in products of ALS data. Random Forest (RF) is used to to predict the probability of palustrine wetland in digital elevation data (Maxwell et al., 2016). Tree-based algorithms are leveraged for ground water mapping in digital elevation data (Naghibi et al., 2016). RF and Support Vector Machines (SVMs) are exploited for forested landslide detection in DTM data (Li et al., 2015, Pawłuszek , Borkowski, 2016). However, in classical machine learning algorithms, the features need to be hand-engineered and selected from the raw data by experts, and only the meaningful features are used to train a model. For example, for models processing DTM data, meaningful features to extract are surface curvature, slope, aspect, distance from water bodies, and roughness of the terrain, among others.

Recently, deep learning has come into play and shown great success in many applications. In deep learning models, features are automatically learned and extracted from the data and no feature extraction is required beforehand. However, these models rely on a huge amount of data to learn properly and not overfit. For image/video data, there are many benchmark datasets with image and label pairs for hundreds to millions of examples such as LabelMe (Russell et al., 2008), ImageNet (Deng et al., 2009), Cityscapes (Cordts et al., 2016), and more. In this research, we use deep learning for semantic segmentation in a derivative of ALS data, namely DTM. We show that with minimal changes to input data preprocessing techniques, and a slight modification in the architecture of the model that proves to work best with RGB data, we can get reasonable results with DTM data. The preprocessing approach is applying min-max normalization on each input, and the modification to DeepLabv3+ architecture is changing the output size to be smaller than that of the input. The rest of this paper is designed as follows. Section 2 outlines previous research. First, it includes famous deep learning methodologies for image data, which can be adapted to work with DTM data. Then, applications of these models on ALS or specifically DTM data are discussed. Section 3 outlines the contributions of this research work including the deep learning model used for semantic segmentation, and the type of preprocessing on the DTM data required by the model. Section 4 discusses the architecture and the hyperparameters of the model in addition to the properties of the dataset used in this research. Section 5 shows the results of the experiment, and the qualitative and quantitative analysis of the results. Finally, section 6 concludes the paper and lists future research directions on the topic.

2. RELATED WORK

There are varieties of tasks using deep learning in computer vision research such as image classification, object localization/detection, semantic segmentation, and instance segmentation. Image classification is the task of providing a category/label/class for a given image. Object detection refers to giving labels and bounding boxes for all possible objects (of similar or different categories) in a given image. Semantic segmentation is the task of labelling each pixel in the image with the object category. Finally, instance segmentation is providing an outline of each object (of similar or different category) appearing in the given image, along with their class

*Corresponding author

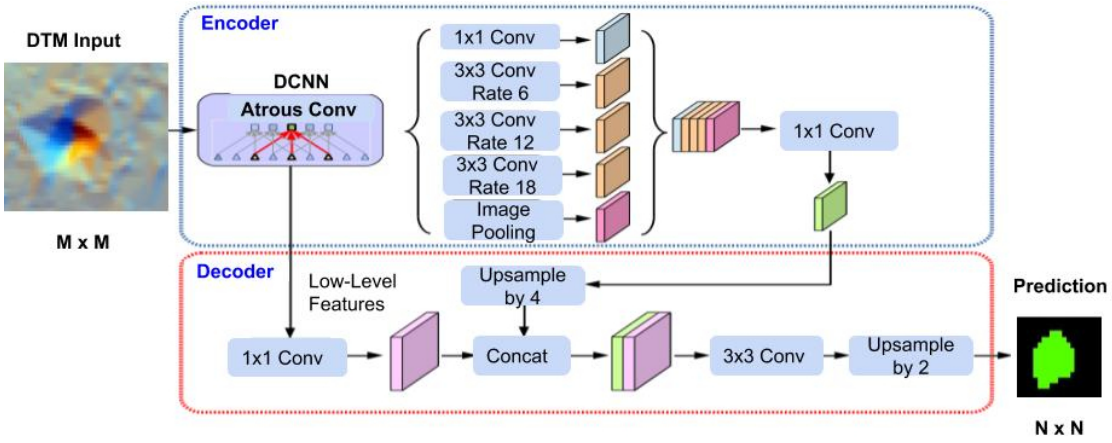


Figure 1. Architecture of DL4DTM. As opposed to DeepLabv3+, the output dimension is, $N = \frac{M}{2}$ and the second upsampling is done with a factor 2 rather 4. The rest of the model is exactly the same as DeepLabv3+

labels.

Convolutional Neural Networks (CNNs) are the main building blocks of computer vision tasks using deep learning. Even though CNNs were used for digit recognition in 1998 (LeCun et al., 1998), their usage became popular after AlexNet (Krizhevsky et al., 2012), a deep CNN architecture for image classification task, won the 2012 ImageNet Large-Scale Visual Recognition Challenge (Russakovsky et al., 2015). After that, many researchers designed models such as ZF-Net (Zeiler , Fergus, 2014), GoogLeNet (Szegedy et al., 2015), VGGNet (Simonyan , Zisserman, 2015), and ResNet (He et al., 2016) to improve image classification tasks. CNNs are also used in other computer vision tasks. Region based CNNs (RCNN) (Girshick et al., 2014) are proposed for object detection. Fast RCNN (Girshick, 2015), and Faster RCNN (Ren et al., 2015) are improvements to the RCNN model. YOLO-V1 (Redmon et al., 2016), YOLO-V2 (Redmon , Farhadi, 2017), and YOLO-V3 (Redmon , Farhadi, 2018) are incremental improvements in object detection tasks and are computationally efficient compared to versions of RCNN. In classification and bounding box regression tasks, the final layer of a deep CNN model is a dense layer with the same number of outputs as the number of classes or categories (in classification), and number of coordinates (in bounding box regression). A deep CNN model for image classification could be transformed to perform semantic segmentation by removing the final dense layer (all together referred to as the *encoder*), and adding a few layers of transposed convolutions and upsampling methods to get the same spatial dimensions as the input image. The second part of the network is called the *decoder* and it makes it possible to produce a label for each pixel. A Fully Convolutional Neural Network (FCN) (Long et al., 2015) is used for semantic segmentation. It uses VGGNet (Simonyan , Zisserman, 2015) as an encoder, and a series of transposed convolutions and bilinear upsampling as a decoder. Other examples of CNN for segmentation include U-Net (Ronneberger et al., 2015), DeepLabv1 (Chen et al., 2015), DeepLabv2 (Chen et al., 2018a), DeepLabv3 (Chen et al., 2017), DeepLabv3+ (Chen et al., 2018b), and the works in (Drozdzal et al., 2016, Jégou et al., 2017, Yu , Koltun, 2015). Another task leveraging CNNs is instance segmentation producing labels and outlines for every object present in a given image. Some examples of instance segmentation models are DeepMask (Pinheiro et al., 2015), SharpMask (Pinheiro et al., 2016), FCIS (Li et al., 2017), (He et al., 2017), and PANet (Liu et al., 2018).

The use of CNNs is studied for remote sensing data and specifically ALS as well. They are used for landuse classification in remote sensing images (Castelluccio et al., 2015), feature extraction and classification of hyperspectral images (Chen et al., 2016, Chen et al., 2014), classification and segmentation of ALS data (Yang et al., 2018), ground and multi-class land cover classification (Rizaldy et al., 2018), tree classification (Hamraz et al., 2018), ground point extraction and classification (Hu , Yuan, 2016), pointwise and object-based classification (Politz et al., 2018), classification of archaeological objects (Kazimi et al., 2018), forest tree detection and segmentation in ALS data (Windrim , Bryson, 2018), and semantic segmentation in ALS data (Politz , Sester, 2018), among others. While there are many applications of deep learning and CNNs for ALS data, there is still room for improvements. With an intent to contribute, we explore the use of semantic segmentation inspired by DeepLabv3+ (Chen et al., 2018b) for identifying objects in ALS data acquired from archaeological sites. Details of the network and our contribution are included in Section 3.

3. CONTRIBUTIONS

The contribution of this research is two-fold. First, we explore semantic segmentation for identifying objects in DTM data acquired from archaeological sites. Second, we show how a different data normalization technique is required for DTM data prior to training a neural network. We take DeepLabv3+ (Chen et al., 2018b) as the basis for our model and adapt it to our specific problem of semantic segmentation in DTM data. We first give detailed information on the original DeepLabv3+ model, and then describe the modifications we made to create a version for our needs.

3.1 DeepLabv3+

DeepLabv3+ (Chen et al., 2018b) is a an encoder decoder model for semantic segmentation, and an improvement on top of its previous versions, namely DeepLabv1 (Chen et al., 2015), DeepLabv2 (Chen et al., 2018a), and DeepLabv3 (Chen et al., 2017). In deep classifiers, in addition to convolution operations, there are additional striding and pooling layers.

Striding and pooling learn higher level abstract information and help avoid overfitting, decrease the output size, apply fewer filters, gain invariance to translations in the input, and gain computational efficiency. However, the abstraction causes loss of detailed information, and the translation invariance in the input harms pixel-perfect accuracy in semantic segmentation. DeepLabv3+ and its predecessors solve this problem by removing downsampling layers, e.g., pooling and striding and using atrous convolutions also known as dilated convolutions. It refers to enlarging the window size, without increasing the number of parameters, by inserting zeros in the convolution kernels. Thus, the output feature maps have a larger size, and fewer upsampling steps are needed. Moreover, objects in the input image can appear at different scales, and to capture this information, atrous spatial pyramid pooling (ASPP) is used. ASPP refers to applying multiple atrous convolution to the input feature with different rates, and concatenating the results to capture multiscale contextual information. Finally, rather than directly upsampling the output feature map to the original input size, it helps to gradually upsample it through a decoder with a convolutional layer in between.

In regular convolution, a filter is applied to the image in the spatial dimensions, i.e., width and height, in a sliding window manner considering all the input channels thus producing an output feature map with a single channel. If an output feature map of more than one channel is desired, it is necessary to apply the same number of filters to the image and stack the output feature maps together to get a multi-channel feature map. This results into having a high number of parameters to learn and many multiplication operations to perform, requiring more memory and more computation time. In DeepLabv3+, the convolutions in the encoder and decoder are operated with atrous separable convolutions. Depthwise separable convolutions, as used in the Xception model (Chollet, 2017), are done in two steps: depthwise convolution and pointwise convolution. In depthwise convolution, the convolutional filters have the same number of channels or *depth* as the input feature map. Each channel of the filter is convolved separately with the corresponding channel of the input feature map to produce a single-channel output feature map. The output feature maps are then stacked together, and have the same depth as the input feature map and the filter. Thereafter, pointwise convolution is applied along the depth of the resulting feature map. it is a 1x1 filter applied across channels. Multiple pointwise convolution is required to get the desired output depth. This combination of depthwise and pointwise convolutions, named depthwise separable convolution, requires fewer parameters to learn with fewer multiplications, thus resulting in significant gains in terms of memory and computation time. In DeepLabv3+, the depthwise convolutions are operated with atrous convolutions.

3.2 DeepLabv3+ for DTM

To correctly label a pixel in RGB images, it is important to capture the context or local information for the pixel in consideration using the surrounding pixels. Thus, pixels in the center of an image are better identified compared to those on the image boundaries. This is tolerable as usually the target objects are the focus of the acquisition device, and each image is independent of other images in the data in the spatial sense. Semantic segmentation models in general take an input image, and produce an output of the same spatial dimension as the input image containing label predictions for each pixel.

DTM data for a big area stored as a large matrix need to be divided into smaller sub-matrices to be fed to deep learning models. Each sub-matrix is spatially bound to its surrounding ones, and unlike RGB images, it is important to include the local information for pixels in the boundaries of each sub-matrix using their neighboring pixels that lie in other sub-matrices. To this end, we propose to design a model that takes an input (a sub-matrix of DTM) with a spatial dimension of MxM, and outputs a pixel-level prediction with spatial dimensions of NxN, where $N \geq \frac{M}{2}$, and the predictions belong to the central NxN pixels of the MxM input. Hence, the predictions for each pixel leverage the context in the surrounding pixels. Once trained, the model could segment the original big DTM matrix in a sliding window manner cropping inputs of MxM with a stride $S = M - N$.

3.3 Data preprocessing

Pixels in DTM data contain continuous information, generally height from a reference surface. Even though the difference in height, hence the differences in pixel values, are important to immediate neighboring pixels, it is less important or even irrelevant to distant pixels. For example a small grid of a DTM containing information for a bomb crater in a surface slightly above sea level differ immensely from another grid of the same spatial dimension with information for a similar bomb crater on top of a high hill. Feeding both grids as raw inputs to a deep learning model could confuse it, and cause a decrease in its performance. Therefore, a proper preprocessing of the input is necessary. There are many preprocessing techniques for image data prior to training a neural network. Some of the common ones are listed below:

- **Zero Centering**

$$newX_i = X_i - \bar{\mathbf{X}} \tag{1}$$

- **Standardization**

$$newX_i = \frac{X_i - \bar{\mathbf{X}}}{\sigma} \tag{2}$$

- **Min-Max Normalization**

$$newX_i = \frac{X_i - \bar{\mathbf{X}}}{max(\mathbf{X}) - min(\mathbf{X})} \tag{3}$$

In all the three equations above, $newX_i$ is the transformed example, X_i is the original raw example, min($\mathbf{X}$), max($\mathbf{X}$), $\bar{\mathbf{X}}$ and σ indicate the minimum, maximum, mean, and standard deviation of the whole dataset, respectively.

The preprocessing methods listed above are usually performed to help the machine learning algorithm converge faster. In our experiments of semantic segmentation on DTM data, we found that using the preprocessing steps mentioned earlier confuses the model and the model fails to converge. This is due to the fact that any two sub-matrices of DTM used as input examples have different values if they are sampled from two different regions, e.g., one from a hilltop and the other from a surface slightly above sea level, even if they represent the same structure, e.g., bomb craters or charcoal kilns. Due to the nature of DTM data, it is not the difference in raw values among two sub-matrices that help distinguish different structures, but the difference in their values when they are both separately rescaled in the range $[0, 1]$ using their correspondent minimum

and maximum values. Thus, rather than applying *min-max normalization* using the maximum and minimum of the whole dataset to calculate rescaled examples, we rescale each example using the maximum and minimum of that example only. The equation is given below:

$$newX_i = \frac{X_i - \bar{X}_i}{max(X_i) - min(X_i)} \quad (4)$$

Where $\bar{X}_i$, $max(X_i)$, and $min(X_i)$ represent the mean, maximum, and minimum for each example i, which is different to those of the whole dataset as used in equation 3 This approach transforms two sub-matrices representing the same structures to have similar distribution, even if they were originally sampled from two different grids of different height. The original min-max normalization would map them to different distributions again, and it would be of no help to the learning algorithm.

With the slight change in the architecture of the DeepLabv3+ model to create a semantic segmentation framework for DTM data, hereafter referred to as DeepLab for DTM or in short DL4DTM, and applying the data preprocessing, we performed some experiments, details of which are given in the next section.

4. EXPERIMENTS

In this research, we aim to use a slightly modified version of DeepLabv3+, named DL4DTM, to do semantic segmentation on DTM data. The following sections give details of the model architecture, the dataset used, and the experiment setup.

4.1 DL4DTM architecture

DL4DTM basically has the same architecture as DeepLabv3+ with the difference that instead of taking RGB input images with 3 channels, it accepts DTM sub-matrices which have a single channel. Another difference between DL4DTM and DeepLabv3+ lies in the output size. Instead of outputting a prediction matrix of the same spatial dimension as the input, DL4DTM outputs a prediction matrix with the spatial dimensions equal to half of the input, resulting into predictions only for the pixels in the mid-region of the input. Consequently, the output of the final convolutional layer is upsampled with a factor of 2 rather than 4. The architecture for DL4DTM is depicted in Figure 1.

4.2 Dataset

The dataset used in this research is DTM derived from ALS data acquired from the Harz Mountains in Lower Saxony, Germany which is home to early mining regions, water management systems, and hosts the UNESCO World Heritage Site, Goslar. The goal of the project is to identify and localize manmade landscape structures such as bomb craters, charcoal kilns, dammed ponds, mine shafts, heaps, and ditches, among others. The challenge with DTM data is creating ground truth labels. It is time consuming to manually create labels, and additionally, it is hard to identify structures and objects in DTM data visually without expert knowledge of the area. In this experiment, we have labelled data for bomb craters, and charcoal kilns. The model learns to categorize input data to bomb craters, charcoal kilns, and background, where background contains everything else in the dataset.

Model	mIOU
DeepLabv3+	**0.77**
DL4DTM	0.75

Table 1. mIOU scores for both models on 220 test examples

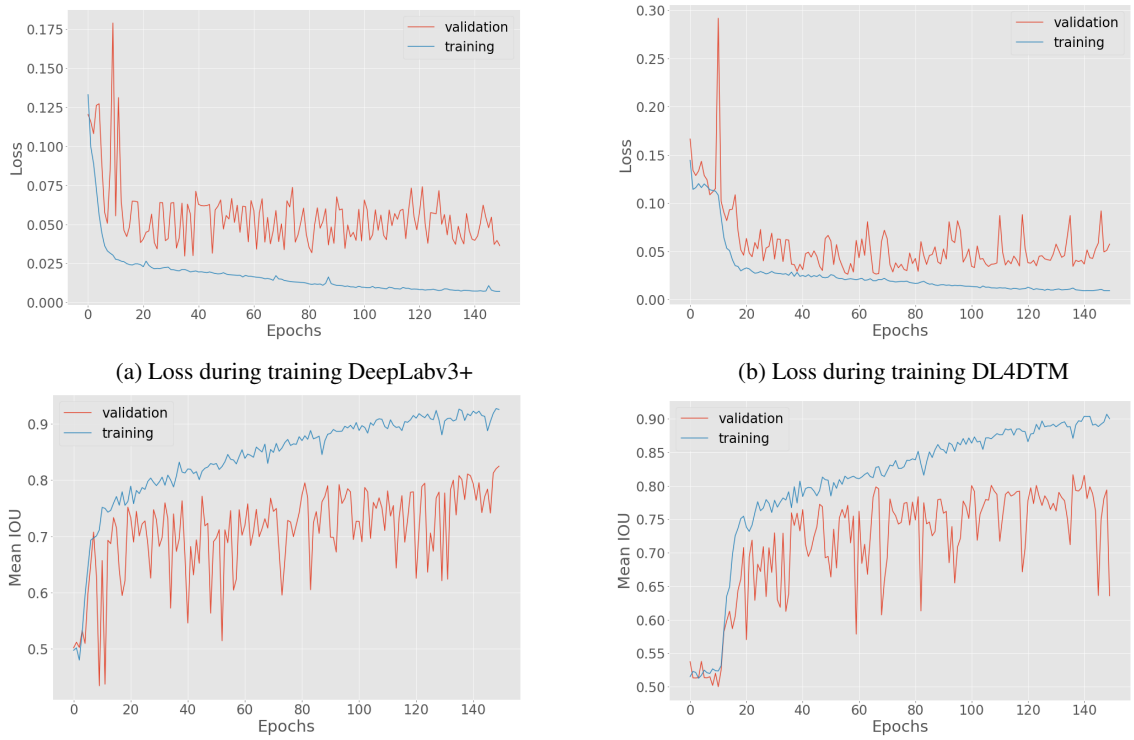
The DTM has a resolution of 0.5 meters per pixel, and covers a total area of approximately 3000 kilometers squared with the highest elevation of approximately 971 meters. We have labelled some bomb craters, and charcoal kilns in this area. There are 157 labelled bomb craters with a diameter in the range 2 to 10 meters. For charcoal kilns, we have 1044 labelled examples, with diameters in the range 3 to 19 meters. The maximum diameters for bomb craters and charcoal kilns are 10 and 19 meters, respectively, which means that in a DTM with a resolution of 0.5 meters per pixel, a grid of 20x20 and 38x38 pixels in size are required to include the whole object. To include context, we created input grids of 128x128 pixels for each example, and to ensure diversity, the 128x128 input grids are cropped randomly from another grid of size 256x256 centered at the object in consideration. Thus, not every input of size 128x128 contains objects at the center. In addition to random cropping, which also serves as data augmentation technique, we performed random rotations to the input during training. Finally, the data preprocessing technique explained in Section 3.3 is done for each input grid prior to being fed to the network.

4.3 Experiment setup

The input data is prepared as 128x128 pixel grids, thus the DL4DTM model is designed to take inputs of this dimensionality. The output of the model however, as explained in Section 3.2, has a size equal to half of the input size, i.e., 64x64 pixels. The output pixels are label predictions for the central pixels of the input grid. DL4DTM inspired by DeepLabv3+ is implemented in Python using Keras (Chollet, 2015). The model is trained for 150 epochs with a batch size of 20, randomly generated from the dataset using augmentation techniques explained in Section 4.2. It is trained to minimize sparse categorical cross entropy using Adam (Kingma , Ba, 2014) optimization algorithm with a default learning rate of 0.001. The metric for evaluation of its performance is mean intersection over union (mIOU). The dataset (before augmentation) is split to two, around 80 percent (127 bomb craters, and 854 charcoal kilns) for training and 20 percent (30 bomb craters, and 190 charcoal kilns) for validation. In order to compare and evaluate our approach, we use the original DeepLabv3+ architecture without any change, and train it on our dataset with the same specifications (except the output size, which is equal to input size in the original model). At each epoch, the model is saved to disk, and at the end of training, the best model is used to scan the test region in a sliding window fashion as explained in Section 3.2, and produce pixel-level predictions. The qualitative and quantitative analysis of the experiments are detailed in the next section.

5. RESULTS

We plot the results of training the two models in the experiments from Section 4 in Figure 2. The final versions of the two models are tested on the same validation data, and the mIOU scores are shown in Table 1.



(a) Loss during training DeepLabv3+

(b) Loss during training DL4DTM

(c) Mean IOU for DeepLabv3+

(d) Mean IOU for DL4DTM

Figure 2. Performance of the DeepLabv3+ and DL4DTM models on our dataset.

Model	mIOU at region 1	mIOU at region2
DeepLabv3+	0.35	**0.37**
DL4DTM	0.37	**0.47**

Table 2. mIOU scores for both models on two test regions

Object Type	DeepLabv3+	DL4DTM
Bomb craters	44/51 (86%)	**49/51 (96%)**
Charcoal kilns	188/233 (81%)	**228/233 (98%)**

Table 3. Detection rates for DeepLabv3+ and DL4DTM for each object category.

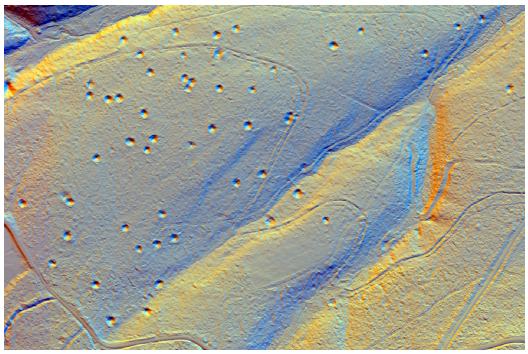
The trained models are used to perform segmentation on large DTMs of test region using a sliding window approach. The test region is scanned using a window of size MxM, with a stride of M for DeepLabv3+, and that of size $M - N$ for DL4DTM where M, and N are the dimensions of the outputs by DeepLabv3+ and DL4DTM, respectively, and $N = \frac{M}{2}$. Example test regions, and the predictions by the two models are shown in Figures 3 and 4 . Mean IOU for the test region predictions are shown in Table 2.

The mIOU scores by DeepLabv3+ as shown in Table 1 are slightly better than that of DL4DTM, but that is expected because for the same input example, the output for DeepLabv3+ is bigger (128x128) than that of DL4DTM (64x64). In the ground truth masks, the objects are mostly located in the center, and pixels far from the center are labeled as zeros. Thus, for the bigger outputs (as in DeepLabv3+), the proportion of background labels (zeros) are higher than that of smaller outputs (as in DL4DTM), and the mean IOU for bigger outputs is expected to be higher than smaller outputs just by getting the background labels correct. The superiority of our method is clearly visible in Table 2, and Figures 3 and 4 since both the models are evaluated on the same large areas.

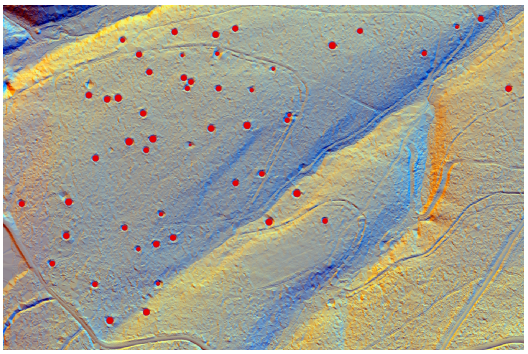
Bomb craters and charcoal kilns do not come in circular shapes, but it is hard to label each of them in their actual shape while creating training examples. Therefore, the ground truth bomb craters and charcoal kilns are represented with a circular shape. Consequently, the calculated mIOU score for the detections may not be a very accurate measure of performance. To resolve this, we also count the number of instances for each class that were detected by the models. There are 51 bomb craters in region 1, and 233 charcoal kilns in region 2. The detections are first converted to polygons using ArcGIS software. Using the *Spatial Join* operation in ArcGIS, we count the number of intersections between the ground truth polygons and the detections by each model. As listed in Table 3, we can see that our approach captured more examples than the original DeepLabv3+ directly applied. Even though we are interested in the exact outline of each object, we are equally interested in the detection accuracy of the model. Even though the mIOU scores for region 1 in Table 2 are not significantly different for both models, the detection rate listed in Table 3 is quite significant for our approach.

The quantitative analysis show that even though for individual input data, the original DeepLabv3+ performs slightly better in terms of mIOU score (as listed in Table 1), in detection rates, and segmenting larger areas, our approach gives better results, as recorded in Tables 2 and 3. It is also clear in Figures 3 and 4 that many examples are missed by DeepLabv3+ while our approach, DL4DTM, correctly identifies them. DL4DTM also detects more examples that were not labeled manually.

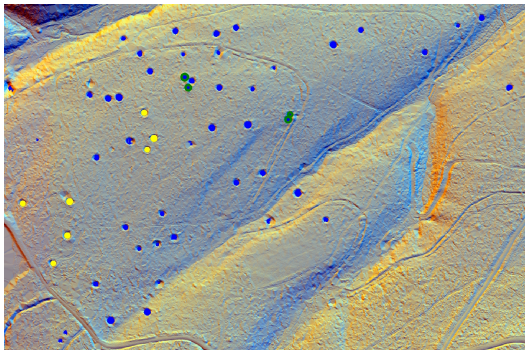
This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-IV-2-W7-87-2019 | © Authors 2019. CC BY 4.0 License.



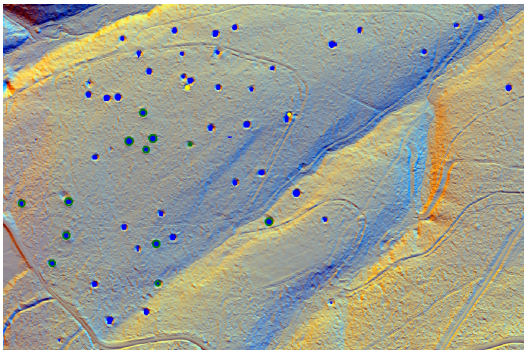
(a) Hillshade of region 1



(b) Ground truth labels for bomb craters

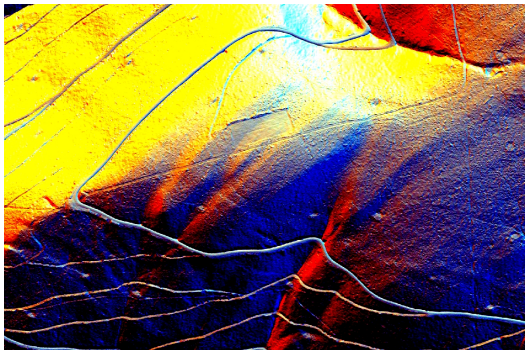


(c) Predictions by DeepLabv3+

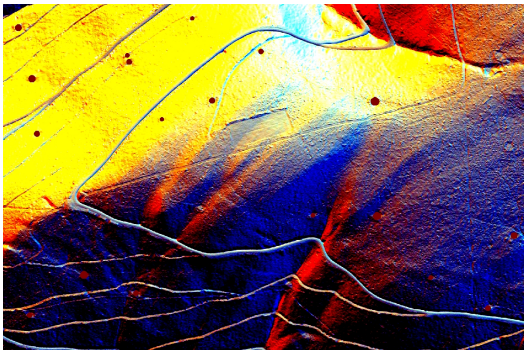


(d) Predictions by DL4DTM

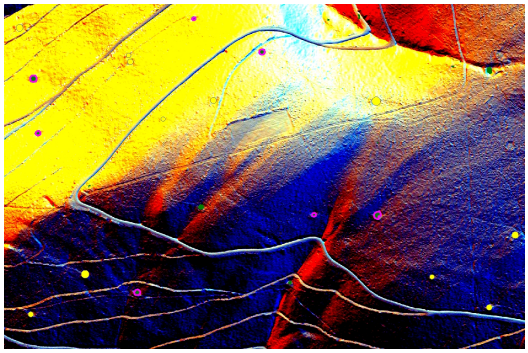
Figure 3. Region 1 containing bomb craters. In 3c and 3d, blue indicate correct predictions, yellow indicate bomb craters that were not detected by the models, the green circle indicate that the model made a better prediction than the other.



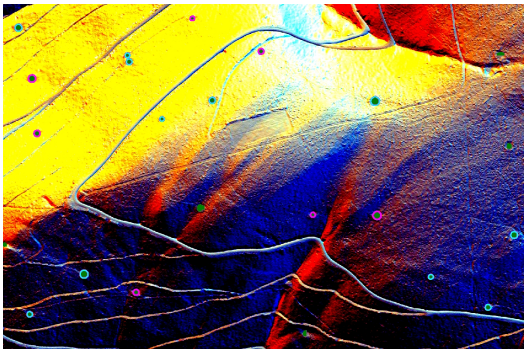
(a) Hillshade of region 2



(b) Ground truth labels for charcoal kilns



(c) Predictions by DeepLabv3+



(d) Predictions by DL4DTM

Figure 4. Region 2 containing charcoal kilns. In 4c and 4d, green indicate correct predictions, the cyan circle indicate that the model made a better prediction than the other, green shapes without the magenta circle around indicate correct predictions that were not captured during manual labeling, and yellow indicate examples not detected by the model.

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-IV-2-W7-87-2019 | © Authors 2019. CC BY 4.0 License.

6. CONCLUSION

In this work, we demonstrated that deep convolutional networks can be used to perform semantic segmentation and identify objects in DTM, a derivative of airborne laser scanning data. Due to the inherent differences in a spatial sense, and the difference in values between RGB images and DTM data, a different data preprocessing technique is required prior to training a neural network. The neural network for semantic segmentation in this research is a modified version of DeepLabv3+ (Chen et al., 2018b). To evaluate the performance, our approach is compared with the original DeepLabv3+ model using the same dataset and the same hyperparameters. Results of the experiment indicate successful application of our approach, and prove that the modification in DeepLabv3+ enhances the performance in the case of DTM data. The focus of our study is detection of manmade landscape structures in archaeological sites, and the current approach is used to detect charcoal kilns, and bomb craters. Due to labeling the ground truth as circular shapes, even though the models learn to segment the objects in their actual structure, they are still constrained, and trained to produce somewhat circular outputs. Additional processing of the input data, and other post processing steps could improve segmentation performances.

Future research directions include application of this technique in detection of other interesting objects such as dammed ponds, mine shafts, heaps, grave hills, and ditches, among others in the region of study. Moreover, other computer vision techniques such as instance segmentation, bounding box regression and object localization could be explored for the same purpose.

REFERENCES

Castelluccio, M., Poggi, G., Sansone, C., Verdoliva, L., 2015. Land use classification in remote sensing images by convolutional neural networks. *ArXiv*, abs/1508.00092. http://arxiv.org/abs/1508.00092.

Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L., 2015. Semantic image segmentation with deep convolutional nets and fully connected crfs. *3rd International Conference on Learning Representations*.

Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L., 2018a. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40, 834–848.

Chen, L.C., Papandreou, G., Schroff, F., Adam, H., 2017. Rethinking atrous convolution for semantic image segmentation. *ArXiv*, abs/1706.05587.

Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H., 2018b. Encoder-decoder with atrous separable convolution for semantic image segmentation. *Proceedings of the European Conference on Computer Vision*, 801–818.

Chen, Y., Jiang, H., Li, C., Jia, X., Ghamisi, P., 2016. Deep feature extraction and classification of hyperspectral images based on convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 54, 6232–6251.

Chen, Y., Lin, Z., Zhao, X., Wang, G., Gu, Y., 2014. Deep learning-based classification of hyperspectral data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7, 2094–2107.

Chollet, F., 2015. Keras. https://keras.io.

Chollet, F., 2017. Xception: Deep learning with depthwise separable convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 1800–1807.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B., 2016. The cityscapes dataset for semantic urban scene understanding. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3213–3223.

Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database. *IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 248–255.

Drozdzal, M., Vorontsov, E., Chartrand, G., Kadoury, S., Pal, C., 2016. The importance of skip connections in biomedical image segmentation. *Deep Learning and Data Labeling for Medical Applications*, Springer, 179–187.

Girshick, R., 2015. Fast r-cnn. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1440–1448.

Girshick, R., Donahue, J., Darrell, T., Malik, J., 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 580–587.

Hamraz, H., Jacobs, N.B., Contreras, M.A., Clark, C.H., 2018. Deep learning for conifer/deciduous classification of airborne LiDAR 3D point clouds representing individual trees. *arXiv*.

He, K., Gkioxari, G., Dollr, P., Girshick, R., 2017. Mask r-cnn. *Proceedings of the IEEE International Conference on Computer Vision*, IEEE, 2961–2969.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778.

Hu, X., Yuan, Y., 2016. Deep-learning-based classification for DTM extraction from ALS point cloud. *Remote Sensing*, 8, 730.

Jégou, S., Drozdzal, M., Vazquez, D., Romero, A., Bengio, Y., 2017. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 11–19.

Kazimi, B., Thiemann, F., Malek, K., Sester, M., Khoshelham, K., 2018. Deep learning for archaeological object detection in airborne laser scanning data. *Proceedings of the 2nd Workshop On Computing Techniques For Spatio-Temporal Data in Archaeology And Cultural Heritage co-located with 10th International Conference on Geographical Information Science*.

Kingma, D.P., Ba, J., 2014. Adam: Amethod for stochastic optimization. *Proceedings of the 3rd International Conference on Learning Representations*.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 1097–1105.

LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86, 2278–2324.

Li, X., Cheng, X., Chen, W., Chen, G., Liu, S., 2015. Identification of forested landslides using LiDar data, object-based image analysis, and machine learning algorithms. *Remote Sensing*, 7, 9705–9726. http://www.mdpi.com/2072-4292/7/8/9705.

Li, Y., Qi, H., Dai, J., Ji, X., Wei, Y., 2017. Fully convolutional instance-aware semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2359–2367.

Liu, S., Qi, L., Qin, H., Shi, J., Jia, J., 2018. Path aggregation network for instance segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8759–8768.

Long, J., Shelhamer, E., Darrell, T., 2015. Fully convolutional networks for semantic segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3431–3440.

Maxwell, A.E., Warner, T.A., Strager, M.P., 2016. Predicting palustrine wetland probability using random forest machine learning and digital elevation data-derived terrain variables. *Photogrammetric Engineering & Remote Sensing*, 82, 437–447.

Naghibi, S.A., Pourghasemi, H.R., Dixon, B., 2016. GIS-based groundwater potential mapping using boosted regression tree, classification and regression tree, and random forest machine learning models in Iran. *Environmental Monitoring and Assessment*, 188, 44. https://doi.org/10.1007/s10661-015-5049-6.

Pawłuszek, K., Borkowski, A., 2016. Landslides identification using airborne laser scanning data derived topographic terrain attributes and support vector machine classification. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 8, 145–149.

Pinheiro, P.O., Collobert, R., Dollár, P., 2015. Learning to segment object candidates. *Advances in Neural Information Processing Systems*, 1990–1998.

Pinheiro, P.O., Lin, T.Y., Collobert, R., Dollár, P., 2016. Learning to refine object segments. *European Conference on Computer Vision*, Springer, 75–91.

Politz, F., Kazimi, B., Sester, M., 2018. Classification of laser scanning data using deep learning. *38th Scientific Technical Annual Meeting of the German Society for Photogrammetry, Remote Sensing and Geoinformation*, 27.

Politz, F., Sester, M., 2018. Exploring ALS and DIM data for semantic segmentation using CNNs. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42, 347–354.

Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 779–788.

Redmon, J., Farhadi, A., 2017. Yolo9000: Better, faster, stronger. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Redmon, J., Farhadi, A., 2018. YOLOv3: An incremental improvement. *ArXiv*, abs/1804.02767.

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 91–99.

Rizaldy, A., Persello, C., Gevaert, C.M., Oude Elberink, S.J., 2018. Fully convolutional networks for ground classification from LiDAR point clouds. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 4.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. *International Conference on Medical Image Computing and Computer-assisted Intervention*, Springer, 234–241.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., 2015. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115, 211-252.

Russell, B.C., Torralba, A., Murphy, K.P., Freeman, W.T., 2008. LabelMe: a database and web-based tool for image annotation. *International Journal of Computer Vision*, 77, 157–173.

Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1–9.

Windrim, L., Bryson, M., 2018. Forest tree detection and segmentation using high resolution airborne LiDAR. *ArXiv*, abs/1810.12536. http://arxiv.org/abs/1810.12536.

Yang, Z., Tan, B., Pei, H., Jiang, W., 2018. Segmentation and multi-scale convolutional neural network-based classification of airborne laser scanner data. *Sensors*, 18, 3347.

Yu, F., Koltun, V., 2015. Multi-scale context aggregation by dilated convolutions. *ArXiv*.

Zeiler, M.D., Fergus, R., 2014. Visualizing and understanding convolutional networks. *European Conference on Computer Vision*, Springer, 818–833.