Acquisition of Urban Trees using Artificial Neural Networks and Remote Sensing Data

Amelie Haas
Technische Universität Dresden / Chair of Geoinformatics
Helmholtzstr. 10
01069 Dresden, Germany
amelie.haas@tu-dresden.de

Pierre Karrasch
Technische Universität Dresden / Chair of Geoinformatics
Helmholtzstr. 10
01069 Dresden, Germany
pierre.karrasch@tu-dresden.de

Lars Bernard
Technische Universität Dresden / Chair of Geoinformatics
Helmholtzstr. 10
01069 Dresden, Germany
lars.bernard@tu-dresden.de

Abstract

Green-spaces, especially trees, influence climate in urban areas in a number of ways and can contribute to climate change mitigation as well as adaptation. Information about this resource is therefore an important basis for decision-making in spatial planning and urban management. While a plethora of remote sensing data is available, in many cases it cannot be used efficiently due to the lack of suitable analysis tools. With regard to image data processing, artificial neural networks, especially convolutional neural networks (CNNs), have become a widely established method during the last decade, boosted by the increased availability of training data and computing power. This work investigates their suitability for the derivation of information on trees in urban areas from remote sensing data.

Multiple CNNs are trained for three different input formats (24x24, 50x50, 100x100 pixels) in order to derive a variety of information, namely tree location, genus, height, age and crown diameter. Digital orthophotos (DOPs) as well as digital surface and elevation models (DSM, DEM) are used as input data. Example data is created using the street tree inventory of the city of Leipzig (Saxony, Germany). The trained models are applied to new data using a sliding window.

The results of this work confirm the great potential of CNNs as generic tools for the analysis of image or raster data shown in previous studies. Upon application to test data, the detection of input images containing visually distinguishable tree crowns is performed with an accuracy of up to 99%. For the classification of tree genera, an overall accuracy of up to 72% is reached, whereas confusion matrices show differences in accuracies for single genera. The remaining target variables are predicted with minimal error values (RMSE) of 9 a for the tree age, 1.8 m for the tree height and 1 m for the crown diameter. As the amount of example data is limited, a strong influence of its composition and quality can be observed.

Keywords: CNNs, Remote Sensing Data, Tree Acquisition, Urban Forestry.

1 Introduction

Maintaining the quality of life in urban areas presents an increasing challenge, considering the backdrop of advancing climate change and urbanization. In this context, urban trees play an important role - amongst other aspects - due to their manifold effects on the local climate (FAO, 2016). For spatial planning and urban management strategies, information about this resource is therefore an important basis for decision-making.

While on-site acquisition is time-consuming, labor-intensive and costly, a large amount of (optical) remote sensing data is readily available. Therefore, the application of suitable methods or rather the development of appropriate tools to analyze the data and derive relevant information is crucial.

Machine learning methods provide an efficient way to analyze large amounts of data. During the last decades, a fast development has taken place in this field, especially with respect to artificial neural networks (ANNs) (LeCun, Bengio & Hinton, 2015; Schmidhuber, 2015). Considering image or raster data, *convolutional neural networks* (CNNs or ConvNets) present an efficient and generic tool for information retrieval. They have therefore become one of the most widely used methods for both image and pattern recognition (Krizhevsky, Sutskever & Hinton, 2017; Simonyan & Zisserman, 2014) as well as for object detection and localization (Erhan *et al.*, 2014; Szegedy, Toshev & Erhan, 2013). Originally developed in the domain of computer vision, this approach has already been adapted to various tasks in remote sensing, such as land cover and land use (LCLU) classifications (Castelluccio *et al.*, 2015; Hu *et al.*, 2015).

This work investigates to which extent CNNs help to derive different types of information on urban trees from remote sensing data. For this purpose, three different models are trained (using three different input formats for each of them):

- CNN1: detection of tree crowns (classification of input data as tree or background),
- CNN2: classification of tree genus,
- CNN3: determination of tree age, height and crown diameter (regression).

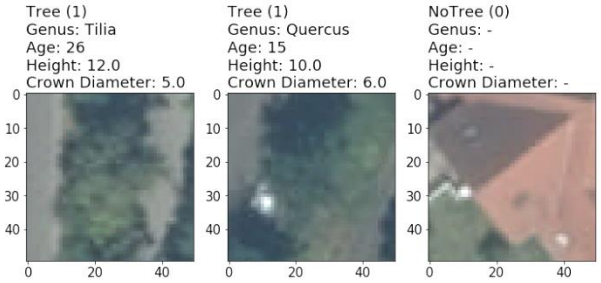
2 Data

The data basis used for this work comprises digital orthophotos (containing red, green, blue and infra-red

channel) with a spatial resolution of 0.2 m as well as digital surface and elevation models with a resolution of 2 m each, covering a selected region of 40 km² in the Greater Leipzig area (Saxony, Germany). Furthermore, the street tree inventory of the city of Leipzig (Stadt Leipzig, 2018) is used, containing coordinates and further information (tree genus, age, height and crown diameter, amongst others) of 16,406 street trees located in this area.

For the development, more precisely the training and evaluation of ANNs, supervised learning is the most frequently used method (LeCun, Bengio & Hinton, 2015:p.436), requiring classified data, i.e. a set of examples consisting of pairs of input data (images) and the related outputs (labels). For this work, the information of the street tree inventory is used to semi-automatically extract squared patches of different sizes from the above mentioned raster data, each containing a centered tree crown. Then, the same number of image patches containing other LCLU classes (background) is created. Each of these patches is labeled according to the target variables of the CNNs (Figure 1).

Figure 1: Example data (DOP shown in true color) consisting of input data (image patches, e.g. 50x50 pixels) and labels



3 Methods

3.1 Data Preprocessing

For data preprocessing, ArcGIS with the site package ArcPy as well as the Python libraries *numpy* (Oliphant, 2006) and *keras* (Chollet & others, 2015) are used.

To speed up computations and avoid the 'curse of dimensionality' arising from the definition of too many predictors, redundant information is excluded. Analysis of the grey values at the center points of the extracted image patches shows high correlations (0.81 to 0.86 for Kendall's τ) between the three color channels. The number of bands is therefore reduced from four (R, G, B, IR) to two (NDVI, G). DSM and DEM are aggregated as nDSM (normalized digital surface model) and resampled to the same resolution as the DOPs (0.2 m). As a result, the input image or raster data (predictors) used for the CNNs comprises three bands (NDVI, G, nDSM) which are normalized and converted to *numpy arrays* for further processing using Python.

The tree locations as recorded in the street tree inventory differ from the positions of the tree crowns observed in the DOPs due to tilting effects. Therefore, a manual shift of the coordinates is necessary so that they are located at the center of the tree crowns in order to extract patches containing one entire, centered tree each. This process is very time-consuming. Thus, the number of examples (32,812 in total) derived from the available data is rather small compared to studies carried out in the domain of computer vision (e.g. Krizhevsky, Sutskever & Hinton, 2017). However, within the field of remote sensing, the amount of example data used for this study is remarkable (cf. Penatti, Nogueira & dos Santos, 2015).

To account for different scales or rather crown diameters, three input formats are used for the CNNs: 24x24, 50x50 and 100x100 pixels (Table 1). According to the distribution of the crown diameter in the original data, the number of extracted examples differs for each format. In addition, the same number of image patches containing a background class is created. For this purpose, gray values at the tree locations are analyzed and image parts having similar values ($\geq \mu - 0.5*\sigma$) are excluded while random sample points are distributed in the remaining area. Then, image patches centered at these points are extracted, according to the procedure described for the tree locations.

Table 1: Extraction of example data at three different scales/formats, according to the crown diameters recorded in the street tree inventory

Input format [pixels]	Crown diameter [m]	Number of examples		
		Trees	Back-ground	Total
24x24	≤ 4.8	7,981	7,981	15,962
50x50	> 4.8 and ≤ 10	3,958	3,958	7,916
100x100	> 10	4,467	4,467	8,934

The resulting raster data is labeled according to the target variables of the models to be used as examples during supervised learning. For this purpose, the three datasets are again divided into three parts to obtain training data (80%; used for adjustment of model parameters, i.e. weights), validation data (10%; used for adjustment of hyperparameters during training) and test data (10%; used for model evaluation after training).

3.2 Design and Training of CNNs

Three different CNNs (CNN1, CNN2, CNN3) are trained for each of the three input formats (24x24, 50x50 and 100x100 pixels), which gives nine models in total. All of them basically share the same architecture, but differ in the first and last layer due to the different inputs (image formats) and outputs (target variables). Both the development and the application of the CNNs is carried out using Python, namely the libraries *numpy* (Chollet & others, 2015), *matplotlib* (Hunter, 2007) and *keras* (Chollet & others, 2015).

The basic architecture (Figure 2) comprises two convolutional layers (*Conv2D* in *keras*), each having 40 filters with a size of 5x5 and 10x10 pixels and followed by max-pooling layers (*MaxPooling2D*) with a size of 2x2 and 4x4 pixels and a stride of 2 and 4 pixels, respectively. The output of the last pooling layer can be interpreted as a feature vector which serves as input for two fully connected layers (*Dense*) with rectifier activation function (Nogueira, Penatti & dos Santos, 2017), the first having 100 neurons (also called

rectified linear units or ReLUs). The number of neurons in the last layer as well as the type of activation function depends on the output (target variable) of the CNN.

The first model (CNN1) is specialized in the detection of tree crowns or rather input images containing trees. Therefore, the example data used for development comprises all of the extracted image patches (Table 1), divided into three parts for training and evaluation (Table 2). All images are labeled as tree (1) or background (0). The last layer of CNN1 contains one single neuron with sigmoid activation function, so the output value ($p \in [0, 1]$) corresponds to the predicted probability that the input data contains a tree.

Figure 2: Basic architecture of the CNNs

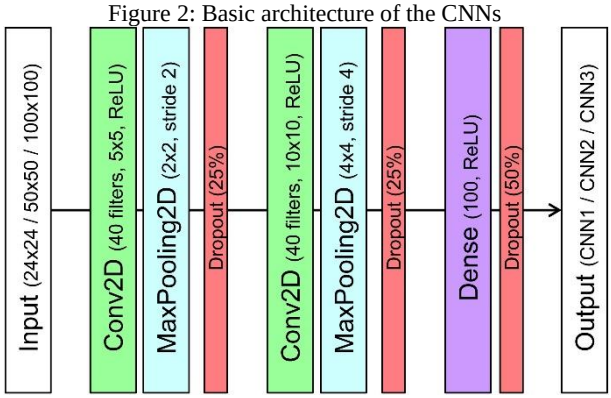


Table 2: Number of examples in training, evaluation and test datasets for all CNNs

Model	Format [pixels]	Number of examples		
		Training	Validation	Test
CNN1	24x24	12,770	1,596	1,596
	50x50	6,332	792	792
	100x100	7,148	893	893
CNN2	24x24	5,094	645	638
	50x50	2,572	326	310
	100x100	3,298	410	430
CNN3	24x24	6,366	810	805
	50x50	3,167	399	392
	100x100	3,564	444	459

For the development of the remaining two CNNs, only the example data created from tree locations are used (cf. Table 1).

CNN2 is specialized in the classification of tree genera. While a total of 33 different genera can be found in the original data used from the tree inventory, many of them occur not more than once or twice. Therefore, only the five most frequent genera for each input format are used (Figure 3). Even though this reduces the amount of example data (cf. Table 2), this approach is necessary for a successful network training. The distribution of the selected genera in the training, validation and test datasets is similar. However, as shown in Figure 3, some genera (e.g. *Tilia*) are very dominant while others are scarce (e.g. *Robinia*).

The regression model (CNN3) has three different output layers, each containing a single neuron with linear activation function to compute the target values tree age, height and crown diameter.

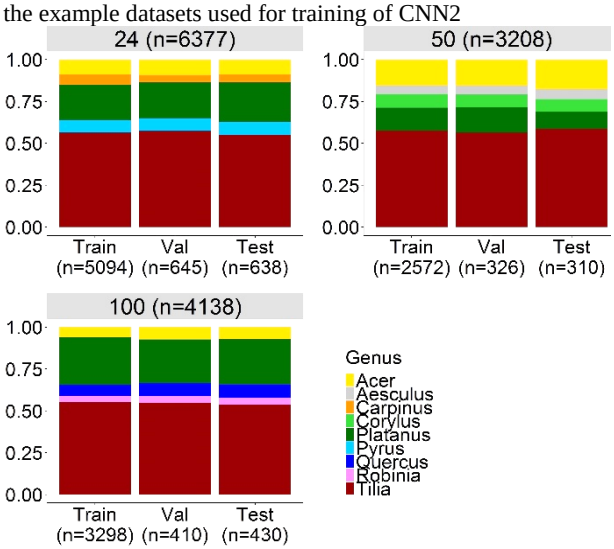
During training, stochastic gradient descent (SGD) and a batch size of 50 (Ruder, 2017) is used for adjustment of the model parameters, also called weights. Except for one case, all models are trained over 10 epochs. The learning rate is adapted automatically using *RMSProp optimizer* (Ruder, 2017:p.7) to accelerate the training. In addition, dropout (Srivastava *et al.*, 2014) is applied to the pooling layers and the first fully connected layer (cf. Figure 2), using only a random choice of 25% or 50% of the neurons during training.

For each of the nine models, 10 different training alternatives are tested:

- *original*: the CNNs are created and trained as described above
- *data augmentation*: training data is modified using a random shearing (max. 10°) and rotation (max. 360°); due to the higher variation in the example data, the number of epochs is increased to 20
- L_1- and L_2-*regularization*: a penalty term is added to the loss function of the second convolutional layer as well as the first fully connected layer to reduce the value of irrelevant weights using a regularization rate of 1%, 10% or 50% for each variant ($L_{1/2}$)
- *reduction*: the model complexity is decreased by reducing the number of filters in both convolutional layers as well as the number of neurons in the first fully connected layer by 25% or 50%

The selection of the best performing models is based on the metrics resulting from the test data, i.e. the maximum overall accuracy for CNN1 and CNN2 and the minimum RSME for CNN3. Given similar values, less complex models are favored.

Figure 3: Distribution of the five most frequent tree genera in the example datasets used for training of CNN2



3.3 Application to new Data

Remote sensing data usually consists of large-size raster data comprising several thousands of pixels, which cannot be used

as input for CNNs due to the enormous computing effort. Therefore, models are trained for smaller input images.

To apply these models to new data, it firstly needs to be preprocessed in the same way as the inputs of the example data (creation of NDVI-G-nDSM composite, resampling, normalization, conversion to *numpy array*). Then, a sliding window of the same size as the input format of the CNNs is used to scan the data and calculate model outputs for each part. Because *numpy arrays* do not contain any spatial reference, the (spatial) coordinates of the upper left corner of the scanned raster and the image coordinates of the window positions are used to locate the model outputs in the original geodata.

To reduce computing time, outputs of the models specialized in tree data (CNN2, CNN3) are only calculated for image patches classified accordingly by CNN1. To identify a tree, a threshold value of $p \geq 0.98$ is determined empirically. In addition, the sliding window is moved by the length of its edges (i.e. 24, 50 or 100 pixels) instead of only one pixel in case a tree has been detected at the previous window position. This avoids processing of overlapping image parts.

4 Results

4.1 Evaluation of trained Models

The performance of all CNNs is assessed using the test datasets as input for the trained models. According to the results (metrics), the best models are selected and applied to new data (Table 3, Figure 4).

Table 3 Selected CNNs with best performance on test data

Model	Format [pixels]	Model Type [1]	Overall accuracy [%]
CNN1	24x24	50% reduction	97
	50x50	25% reduction	98
	100x100	original	99
CNN2	24x24	25% reduction	69
	50x50	50% reduction	65
	100x100	25% reduction	72

			RMSE		
Model	Format [pixels]	Model Type [1]	Crown Ø [m]	Height [m]	Age [a]
CNN3	24x24	50% reduction	1	1.8	9
	50x50	25% reduction	1.5	3	28
	100x100	50% reduction	2.2	3.6	31

[1] see explanations in 3.2

The best overall accuracy for CNN1 is achieved with the largest input format (100x100 pixels). For the remaining formats, a maximum overall accuracy of 98% (50x50 pixels, 25% reduction) and 97% (24x24 pixels, 50% reduction) is obtained.

Concerning CNN2, the best results are also reached using the less complex model variants, namely 50% reduction (50x50 pixels) and 25% reduction (24x24, 100x100 pixels), showing an overall accuracy of 65 to 72%. However, this value is not sufficient to describe model performance in this case, as it provides no information about class specific classification accuracy. Analysis of the confusion matrices shows, that dominant genera, especially *Tilia*, are predicted with a much higher accuracy than less frequent genera, e.g. *Acer* (Table 4; cf. 3.2, Figure 3). In general, misclassifications occur in favor of these dominant classes, while others may not be predicted at all for the test data.

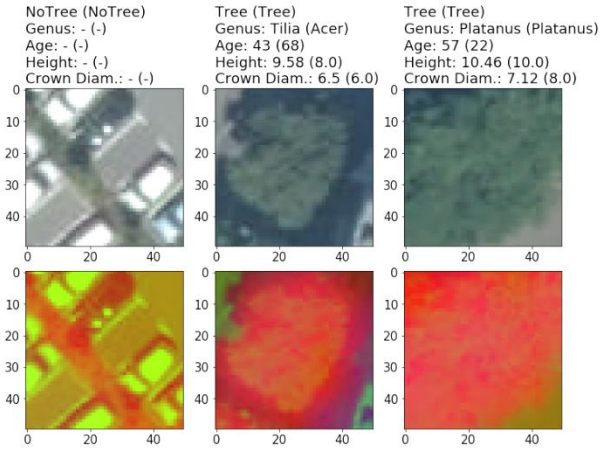
Table 4 Confusion matrix of CNN2 (for application on test data); exemplarily shown for best performing model of input format 24x24 (25% reduction)

	A.	C.	Pl.	Py.	T.	∑	UA [1] [%]
Acer	0	0	14	1	41	56	0
Carpinus	0	2	5	1	22	30	7
Platanus	0	0	106	0	45	151	70
Pyrus	0	0	5	13	31	49	27
Tilia	0	0	28	4	320	352	91
∑	0	2	158	19	459		
PA [1] [%]	-	100	67	68	70	overall acc.: 69%	

[1] PA: producer's accuracy, UA: user's accuracy

In the case of CNN3, the selected models present a compromise, as minimal RMSE values for the three outputs (tree age, height and crown diameter) are sometimes achieved by different models. Again, the less complex models - 25% reduction (50x50 pixels) and 50% reduction (24x24, 100x100 pixels) - show the best results yielding at errors (RMSE) of 9 to 31 a for the tree age, 1.8 to 3.6 m for the height and 1 to 2.2 m for the crown diameter.

Figure 4: Test examples and predictions of selected models (50x50 pixels); test labels specifying the correct values are given in brackets; true color images are shown in the first row for visualization only, the second row contains the composite raster data used as input (R=NDVI, G=Green, B=nDSM)



4.2 Application of best Models to new Data

The selected models are applied to new data, i.e. input data (DOP, DSM, DEM) that has not previously been used for the extraction of examples. For this purpose, the data is preprocessed and sliding windows are applied according to the procedure described in 3.3. Due to the lack of reference data, a validation of the results is not possible in this case. Instead, a visual and statistical analysis of the network outputs is given in this section.

Trees are located according to the outputs of CNN1 by determining the spatial coordinates at the center of all window positions for which a value of $p \geq 0.98$ is predicted. Visualization of the predicted crown area is possible by combining this information with the crown diameter of CNN3 (Figure 6). Regarding solitary street trees - as contained in the example data - the resulting delineation of the crowns seems quite correct, while in areas where tree crowns are connected and overlapping, e.g. in parks or forests, a regular pattern of predicted tree locations is distinguishable, resulting from the positive classification of multiple windows in neighboring positions and the determined stride according to the window sizes (Figure 7).

Concerning the tree genera, statistical analysis shows that the dominant class in the example data, *Tilia*, is predicted in more than 60% of the cases for all formats (Figure 5, cf. Figure 3). Besides, the predictions for the two smaller formats are limited to four genera, despite five different classes have been used for training. Considering the results for the test data described above (cf. 4.1, Table 4), this has been expected. However, misclassifications are inevitable in this case, as the models are trained for a limited set of genera. As the tree genus is not visually distinguishable in the DOPs, the model predictions cannot be verified.

The predicted values of CNN3 for all three target variables are also clustered around single values (Figure 5). Again, those values are similar to those occurring most frequently in the example data used for model training.

Figure 6: Crown area (blue circles, created from predicted tree locations and crown diameters) resulting from application of CNNs with input format 24x24 pixels

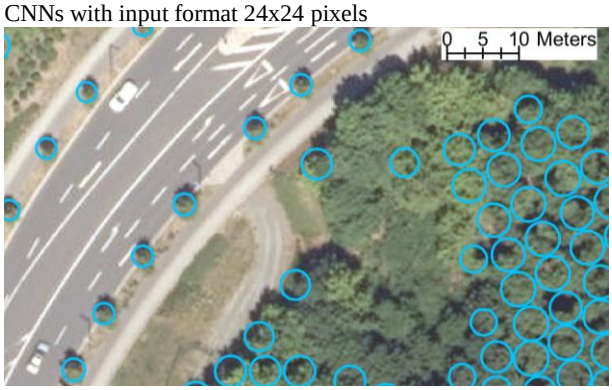


Figure 7: Predicted tree locations resulting from application of CNN1 with input formats 24x24 pixels (blue), 50x50 pixels (green) and 100x100 pixels (red)

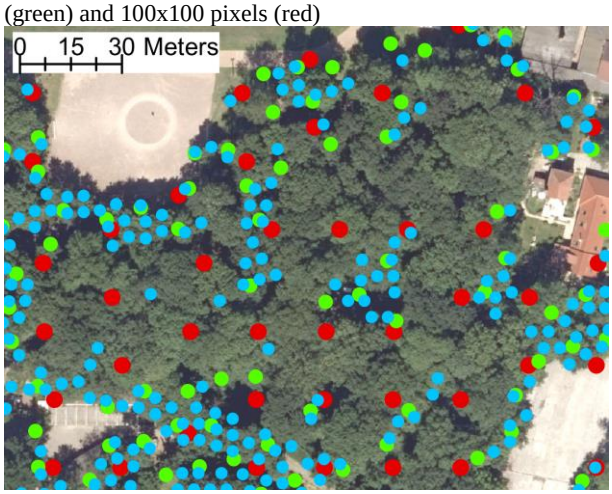
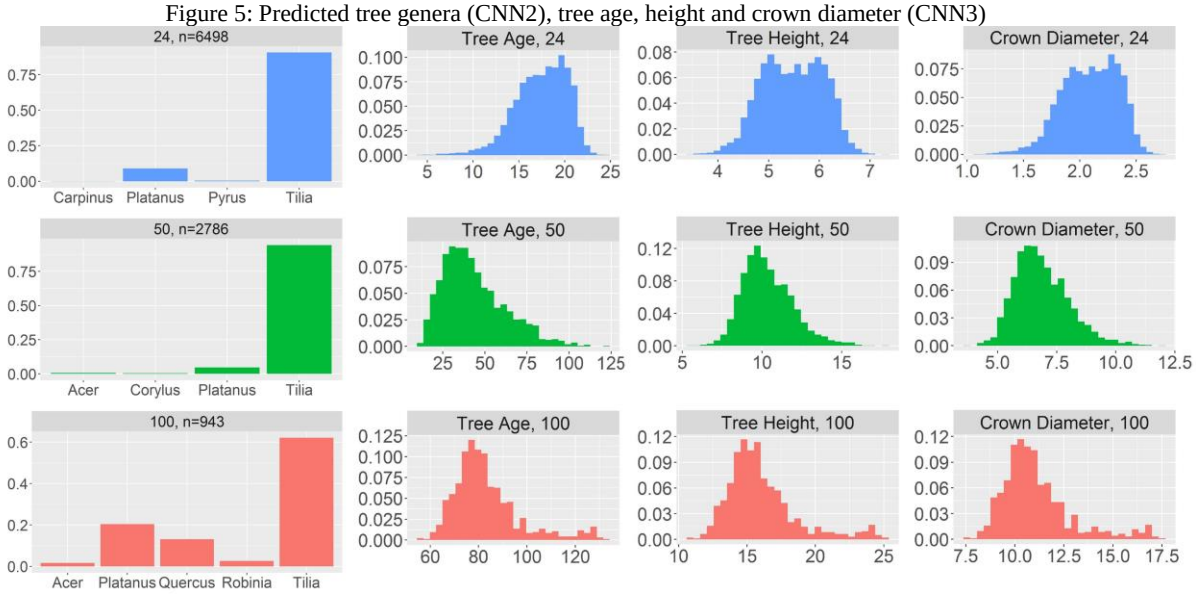


Figure 5: Predicted tree genera (CNN2), tree age, height and crown diameter (CNN3)



5 Conclusion

The results of this work confirm the great potential of CNNs concerning analysis of image or raster data, also in the domain of deriving and enriching urban tree inventories. The fact that all classification and regression models (CNN1, CNN2, CNN3) rely on the same architecture shows the high flexibility and genericity of this type of ANNs as tools to derive urban tree features.

However, quantity as well as quality of the available example data turns out to be crucial for model development. While a simple distinction between image patches containing trees and those containing background (CNN1) is possible with a very high accuracy (on test data), satisfying results for more complex tasks (CNN2, CNN3) cannot be achieved for all models. In these cases, predictions show a strong bias towards dominant target values present in the example data.

To better assess the suitability of CNNs for the issues investigated in this work, further studies are necessary using a larger data basis (more tree genera, different recording time of remote sensing data, etc.). Nevertheless, this work shows remarkable results, considering the limited amount of example data (3,208 to 15,962 examples for each CNN), which is rather small compared to studies within the domain of computer vision.

Moreover, a comparison to existing techniques, such as segmentation methods (eCognition) or maximum likelihood classification (ERDAS Imagine).

A more precise localization of single trees is possible by delineating the crown using a bounding box. This approach can also be realized using CNNs, but appropriate example data is required. Further improvements and enhanced implementations of the proposed approach could lead to an almost fully automatized workflow for the creation and maintenance of (urban) tree inventories.

References

Castelluccio, M., Poggi, G., Sansone, C. & Verdoliva, L. (2015) *Land Use Classification in Remote Sensing Images by Convolutional Neural Networks*. Available from: http://arxiv.org/pdf/1508.00092v1 [Accessed 25th June 2018].

Chollet, F. & others (2015) Keras. Available from: https://keras.io [Accessed 19th February 2019].

Erhan, D., Szegedy, C., Toshev, A. & Anguelov, D. (2014) Scalable Object Detection Using Deep Neural Networks. In: *2014 IEEE Conference on Computer Vision and Pattern Recognition, 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, OH, USA, 23-28 June, IEEE, pp. 2155–2162. Available from: doi:10.1109/CVPR.2014.276.

Food and Agriculture Organisation of the United Nations (FAO) (2016) *Building greener cities: nine benefits of urban trees: Find out why trees in cities matter*. Available from: http://www.fao.org/zhc/detail-events/en/c/454543/ [Accessed 9th December 2018].

Hu, F., Xia, G.-S., Hu, J. & Zhang, L. (2015) Transferring Deep Convolutional Neural Networks for the Scene Classification of High-Resolution Remote Sensing Imagery. *Remote Sensing*. 7 (11), pp. 14680–14707. Available from: doi:10.3390/rs71114680 [Accessed 27th June 2018].

Hunter, J. D. (2007) Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*. 9 (3), pp. 90–95. Available from: doi:10.1109/MCSE.2007.55 [Accessed 19th February 2019].

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2017) ImageNet classification with deep convolutional neural networks. *Communications of the ACM*. 60 (6), pp. 84–90. Available from: doi:10.1145/3065386 [Accessed 25th June 2018].

LeCun, Y., Bengio, Y. & Hinton, G. (2015) Deep learning. *Nature*. 521 (7553), pp. 436–444. Available from: doi:10.1038/nature14539 [Accessed 25th June 2018].

Nogueira, K., Penatti, O. A.B. & dos Santos, J. A. (2017) Towards better exploiting convolutional neural networks for remote sensing scene classification. *Pattern Recognition*. 61, pp. 539–556. Available from: doi:10.1016/j.patcog.2016.07.001 [Accessed 4th July 2018].

Oliphant, T. E. (2006) *Guide to Numpy*. opensource. Available from: https://archive.org/details/NumPyBook [Accessed 19th February 2019].

Penatti, O. A. B., Nogueira, K. & dos Santos, J. A. (2015) Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? In: *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Boston, MA, USA, 7-12 June, IEEE, pp. 44–51. Available from: doi:10.1109/CVPRW.2015.7301382 [Accessed 27th June 2018].

Ruder, S. (2017) *An overview of gradient descent optimization algorithms*. Available from: http://arxiv.org/pdf/1609.04747v2.

Schmidhuber, J. (2015) Deep learning in neural networks: an overview. *Neural Networks: the Official Journal of the International Neural Network Society*. 61, pp. 85–117. Available from: doi:10.1016/j.neunet.2014.09.003 [Accessed 25th June 2018].

Simonyan, K. & Zisserman, A. (2014) *Very Deep Convolutional Networks for Large-Scale Image Recognition*. Available from: http://arxiv.org/pdf/1409.1556v6 [Accessed 25th June 2018].

Srivastava, N., E. Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. (2014) Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*. 15, pp. 1929–1958. Available from: https://www.cs.toronto.edu/~hinton/absps/JMLRdropout.pdf [Accessed 4th July 2018].

Stadt Leipzig (2018) *Stadtbäume*. Available from: https://www.leipzig.de/umwelt-und-verkehr/umwelt-und-

naturschutz/baeume-und-baumschutz/stadtbaeume/ [Accessed 9th December 2018].

Szegedy, C., Toshev, A. & Erhan, D. (2013) Deep Neural Networks for Object Detection. In: C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K. Q. Weinberger (eds.). *2013 NIPS Neural Information Processing Systems, Neural Information Processing Systems*, Lake Tahoe, Nevada, USA, 5-10 December, Curran Associates, Inc, pp. 2553–2561. Available from: http://papers.nips.cc/paper/5207-deep-neural-networks-for-object-detection.pdf [Accessed 25th June 2018].