TREE SPECIES IDENTIFICATION USING 3D SPECTRAL DATA AND 3D CONVOLUTIONAL NEURAL NETWORK

Ilkka Pölönen[1], *Leevi Annala*[1], *Samuli Rahkonen*[1], *Olli Nevalainen*[2], *Eija Honkavaara*[2],
Sakari Tuominen[3], *Niko Viljanen*[2], *Teemu Hakala*[2]

[1]Faculty of Information Technology, University of Jyväskylä, P.O. Box 35, FI-40014 Jyväskylä, Finland
[2]Finnish Geospatial Research Insititute, National Land Survey of Finland,
Geodeetinrinne 2, 02430 Masala, Finland
[3]Natural Resources Institute Finland, PL 2 00791 Helsinki, Finland

ABSTRACT

In this study we apply 3D convolutional neural network (CNN) for tree species identification. Study includes the three most common Finnish tree species. Study uses a relatively large high-resolution spectral data set, which contains also a digital surface model for the trees. Data has been gathered using an unmanned aerial vehicle, a framing hyperspectral imager and a regular RGB camera. Achieved classification results are promising by with overall accuracy of 96.2 % for the classification of the validation data set.

Index Terms— Tree species, spectral imaging, 3D, convolutional neural network, UAV

1. INTRODUCTION

This study is continuum for [1], where the individual tree detection and classification pipeline for the hyperspectral and point cloud data is clearly described. We are interested to see if deep learning methods could improve or simplify the data processing chain for identifying the species of individual trees.

There exists plenty of research concerning tree species identification, but it is mainly concentrated on large scale remote sensing, which uses forest stand and plot level data. For example in Scandinavia combination of airborne laser scanning and aerial images is used in forest inventory [2]. There are less studies and applications for the tree species identification from unmanned aerial vehicles (UAV) using hyperspectral sensors. If hyperspectral data has been used for tree species identification, the platform for data gathering has been manned aircraft or satellite.

As in [1], these remote sensing studies use quite traditional feature extraction and selection methods before classification. Deep learning methods have dramatically improved performance of pattern recognition [3]. Especially deep convolutional neural networks (CNN) have provided breakthroughs in image, video and audio processing. If we consider hyperspectral data, it seems that they should handle hyperspectral data combined with 3D data as well. There is currently increasing number of research, which applies CNN's and 3D CNN's to hyperspectral imager [4, 5].

In this paper we first test performance of 3D CNN for tree species classification. Neural networks has the nature of being a black box, that doesn't reveal how it has reasoned its results. However, while doing classification we can calculate saliency maps, which will give us hints on which parts of the input data are relevant for the CNN [6].

This paper has the following structure. First, in Section 2 we describe the used data set, its acquisition and preprocessing. Then the structure and functionality of the used 3D CNN is described. In Section 3, the results are presented and Section 4 includes the conclusion.

2. MATERIALS AND METHODS

2.1. Data gathering and preprocessing

The research data is the same as reported in [1]. The collected remote data was captured in Vesijako research forest area in the municipality of Padasjoki in southern Finland (approximately 61^{o}24'N and 25^{o}02'E). Area has been used for forestry research by Natural Resources Institute of Finland. The area contains experimental plots with different research setups. All the trees with the diameter of at least 50 mm at the breast-height were measured and estimated with various metrics, such as the tree species, diameter, height and volume. Locations of these trees were collected with GPS.

In total, 4142 trees were selected for further study. The data set contained three most common species of Finnish forests: scots pine (*Pinus sylvestris*, 2821 samples), norway spruce (*Picea abies*, 742 samples) and silver birch (*Betula bendula*, 579 samples). These selected trees were compared to aerial orthoimage mosaics to ensure that the GPS coordi-

This research has been co-financed by Finnish Funding Agency for Innovation Tekes (grants 2208/31/2013 and 1711/31/2016)

nates were in the centres of the treetops.

The used remote sensing data was a combination of two data modalities captured by the UAV remote sensing system, which belongs to Finnish Geospatial Research Institute. System consist of a Tarot 960 hexacopter and a Pixhawk autopilot. System is capable of carrying 3 kg payload at maximum. Average flying time of the system is 30 minutes. As a payload, we had a tunable Fabry-Pérot inteferometer based spectral imager (FPI) and an ordinary RGB camera, the Samsung NX1000 (RGB). Flying height from ground level varied between 83-94 meters.

The FPI imager captures raw data, which is processed to radiance based on the radiometric laboratory calibration [7]. The geometric imaging model was then determinated. The model includes both the interior and exterior orientations of the images. The digital surface model is calculated by dense image matching. Because of the slight variations between bands in the FPI camera, we had to apply registration of the spectral bands of FPI images. To make data cubes and further mosaics radiometrically homogenous, the radiometric imaging model has to be determined [8, 9]. The hyperspectral image mosaic is calculated after the radiometric model is applied to each cube. The detailed radiometric and geometric processing of the data set is explained in [1]. Finally, the spectral mosaics with 33 bands and digital surface model (DSM) both with 10 cm GSD are created.

For the tree species identification, 4×4 meter windows surrounding each treetop were extracted. The windows contained both DSMs as rasters and spectral cubes. For each treetop, the extracted DSMs were scaled by the minimum value of the whole DSM. The DSM and spectral cube for each treetop were concatenated in spectral axis to unified data cubes ($41 \times 41 \times 34$). In Finland there exist laser scanned nation wide ground surface elevation model, which is freely available. Thus, canopy surface model could have been calculated, but it isn't actually needed, because we are only using height of the treetops.

Figure 1 illustrates average treetops for each species. We can see that there are slight differences between the shapes. Pine's treetops are quite symmetric. Spruce's treetops are more of ellipses and aligned on north-west to south-east axis. Birch is more irregular, but its leaves and branches are towards south where the Sun shines.

Figure 2 represents how spectral distribution diverges to different wavelengths for each tree species. The line in the figure represents the average spectra for each treetop. Quite obvious differences can be found between birches and Nordic coniferous trees. Birches have stronger reflection in green and infrared regions. Birches have steeper spectrum at red edge area.

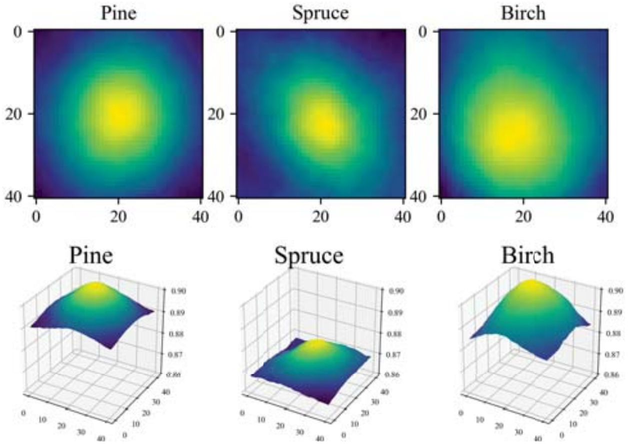

Fig. 1. Avarage shape of treetop for each tree species.

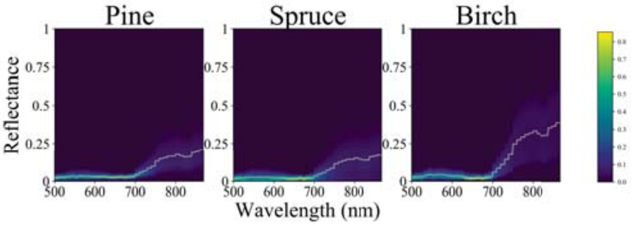

Fig. 2. Histogram spectra of each tree species. Black line is average spectrum.

2.2. Convolutional Neural Network

Originally CNN's were presented by LeCun and Bengio [10]. The idea was to tackle feature extraction and selection problem in fully connected feed-forward networks. The network uses a convolution matrices. Traditional neural network layers are usually based on consecutive dense (fully connected) neurons. In convolutional neural networks, there exists at least one convolution operation in the network. We applied quite simple structure to our CNN, using four types of layers: 3D convolutional, pooling, dropout and fully connected layers. Our network's structure is presented in Table 1.

In general, convolutional layers have trainable filters, which use convolution operations to extract features. In our implementation, the convolution layer uses activation called rectified linear unit (ReLU). ReLU has advantages of being efficient with non-linear relations and having less vanishing gradient problems during the network optimisation compared to other popular activation functions [11]. Pooling layers, which usually follow convolutional layers, are non-linear downsampling functions, which reduce dimensions of input data. Dropout layer is a regularization method for reducing overfitting in the neural network by introducing noise to the network. Flatten layer translates data to one dimensional stack.

Layer	Kernel / pool size or Activation	Output Shape	Parameters
Conv3D	(3,3,1) ReLU	(39, 39, 33, 64)	640
Conv3D	(3,3,3) ReLU	(37, 37, 31, 64)	110656
MaxPooling3D	(2,2,1)	(18, 18, 31, 64)	0
Conv3D	(3,3,3) ReLU	(16, 16, 29, 128)	221312
MaxPooling3D	(2,2,3)	(8, 8, 9, 128)	0
Conv3D	(3,3,3) ReLU	(6, 6, 7, 256)	884992
MaxPooling3D	(2,2,3)	(3, 3, 2, 256)	0
Flatten		(4608)	0
Dense	ReLU	(128)	589952
Dropout (0.25)		(128)	0
Dense	SoftMax	(3)	387
Total params:	1,807,939		
Trainable params:	1,807,939		
Non-trainable params:	0		

Table 1. Structure of our experimental CNN.

A dense layer is a fully connected layer, which consists of parallel neurons which are connected to all previous layer's outputs. Weights of the connections and activation functions determine which features are correlating with different tree species. The last dense layer is activated with *softmax* function, whose output is the final classification.

If the amount of data is limited, meaning that the number of training samples is low, then there is option to apply data augmentation. Basically this means that we will generate new training data from existing ones. In this study we fivefold our training data by using simple rotation and flipping operations. Selected training data was flipped both horizontally and vertically. Data was also rotated 90 degrees to left and right.

In machine learning structures like neural networks are so called "black box" solutions. We don't have clear vision how data is classified. It is reasonable to ask, is the classification based on real feature of wanted object or something secondary such as ground type in tree species recognition. Luckily there are methods to see where network is putting weight in classified data. It is possible to calculate gradient over layers from output to input. This way to get actually image, where areas with higher values contributes most to classification result. These maps are called *saliency maps*.

Stochastic gradient decent was used to tune weights between layers. We used categorical cross entropy as a loss function, which basically calculates cross entropy between categories probability distributions. Primary metric for model evaluation was accuracy

$$acc = \frac{TP + TN}{TP + FP + FN + TN},$$

where TP is true positive, TN is true negative, FP is false positive and FN is false negative classification result.

CNN's were trained by using IBM PowerAI platform which includes two Tesla V100-SXM2 16 GB GPU units. Tensorflow was used as a computational backend [12]. All machine learning phase coding was done using Python 3.6 and Keras library [13]. Saliency maps were calculated using Keras-vis library [14].

3. RESULTS

Altogether 3311 trees were randomly selected for the training of the 3D CNN. After data augmentation there was 16555 samples. Training was performed with batch size 128 and with 100 epocs. Training took two and half hours (approx. 88 seconds/epoch). Results were validated with 831 samples, which weren't included in training set.

Figure 3 shows that accuracy of trained model is relatively high. It seems that we can with quite large confidence identify tree species from each other. Overall accuracy for classification of validation set was 96.2%, which is higher with earlier results achieved in [1]. Producer accuracies were for each tree species were 96.2% (Pine), 86.6 % (Spruce) and 98.2 % (Birch). Respectively users accuracies were 96.3 %, 83.8 % and 95.7 %.

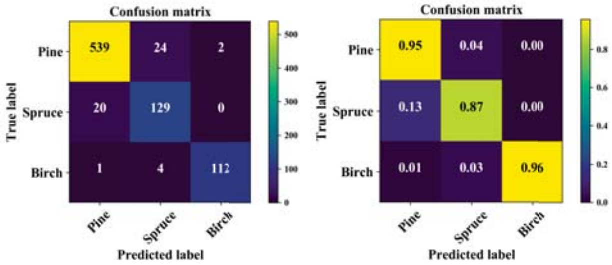

Fig. 3. Confusion matrices show good separation between tree species.

Figure 4 is presenting average saliency maps in spatial domain over all input bands of validation data. It seems that most of the important features are handling data surrounding tree top. This is shown more clear in the figure 5, where figure's 4 maps are rendered over validation sets average 3D treetops. Thus, we can be quite confident that, at least in spatial domain, tree top's shape is relevant feature in classification.

In spectral domain most characterising features seems to be located between wavelengths from 600 to 720 nm. Figure 6 presents average salience in each spectral band. It can seen that there is differences between tree species. For example birch has lower saliency in 560 nm and higher in 700 nm than coniferous trees.

If we consider individual trees, it seems that classifying is working quite efficiently. In figure 7 there is one tree of each species from the validation set. It can be seen that for example pine in this case doesn't have very clear treetop, but classifier is able to find one and saliency map seems to confirm the result.

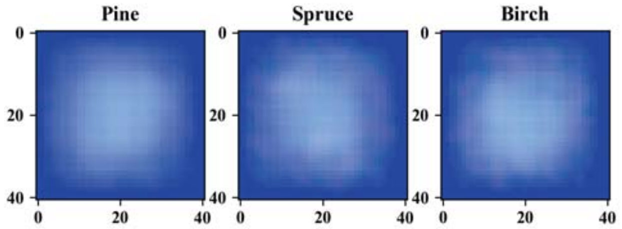

Fig. 4. Average saliency maps in spatial domain over all input bands of validation data for each tree species. Brighter pixel indicates that band is probably more meaningful in classification.

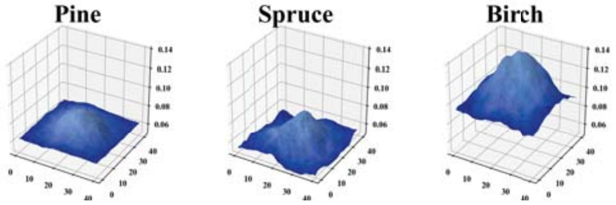

Fig. 5. Here figure's 4 saliency maps are rendered over validation set's average 3D treetops. It can be seen that maps surround quite well treetops.

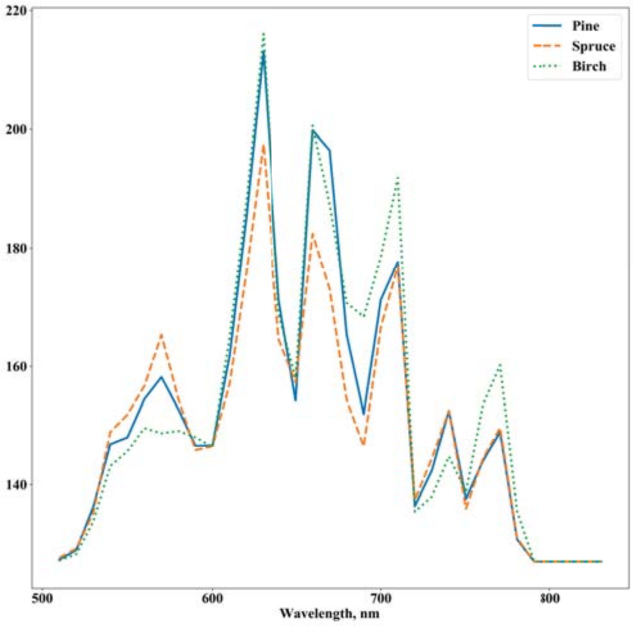

Fig. 6. Average saliencies of spectral domain for each tree species. Higher value indicates that band is probably more meaningful in classification.

4. CONCLUSIONS

In this paper we demonstrate how 3D hyperspectral data can be analysed using 3D convolutional neural networks. As a concluded result we can see that even with quite simple 3D CNN, it is possible to create network, which has good capability to classify single trees based on their shape and spectral features.

In classical machine learning one of the most time consuming thing for data analysist has been feature extraction and selection. In case of convolutional neural network this phase is now automated. After preprocessing there is quite limited amount of things to do, if you want to utilize trained network. Network training itself is time consuming, but before hand trained network can deliver results almost in realtime. In our case training took two and half hours.

Compared to earlier work [1] we actually used all captured test areas. In original paper one area was left behind, because of the poor quality of image block. Based on that, our results seems to show that trained 3D CNN is actually more robust as a classifier than methods used in previous study.

It is obvious that more studies is needed. Used network structure is one of the most simple ones. With more sophisticated structures it might be possible to improve learning results. One of the tested things in the future is, how general trained model actually is. If we have another data set, can we have similar classification results? We used quite limited amount of data augmentation. Even tough overfitting wasn't observed based loss and accuracy curves during the training, it would be useful to do more cross-validation within data.

One potential research question is that how many bands and what GSD is needed, if we want to gain similar results. Our next steps include more augmented data to training such as scaling, adding noise, chancing lightness and adding more rotation to see if we could detect trees also with lower resolution.

The used data set has more parameters for single trees (height, estimated volume, etc..) and there is also 300 fixed radius (9 m) sample plots, which have been used for area based forest inventory. In near future we will also test how well 3D CNN approach is able to estimate these parameters.

Our consortium has ongoing research project where our aim is to produce real time processing for the DSM and hyperspectral mosaics. This combined with pre-trained CNN classifier, could be significant tool to provide forest tree identification and parameter estimation without wasting time on massive preprocessing.

5. REFERENCES

[1] Olli Nevalainen, Eija Honkavaara, Sakari Tuominen, Niko Viljanen, Teemu Hakala, Xiaowei Yu, Juha Hyyppä, Heikki Saari, Ilkka Pölönen, Nilton N Imai, et al., "Individual tree detection and classification with uav-based photogrammetric point clouds and hy-

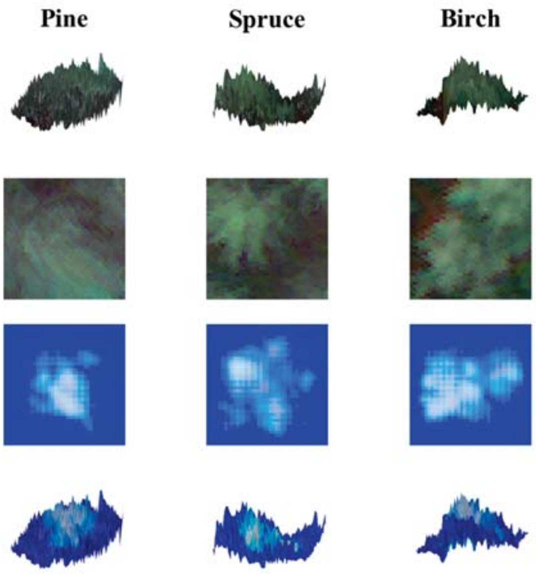

Fig. 7. Comparison individual trees. CNN is capable to detect trees surprisingly well.

perspectral imaging," *Remote Sensing*, vol. 9, no. 3, pp. 185, 2017.

[2] Erik Næsset, "Predicting forest stand characteristics with airborne scanning laser using a practical two-stage procedure and field data," *Remote sensing of environment*, vol. 80, no. 1, pp. 88–99, 2002.

[3] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[4] Sungbin Choi, "Plant identification with deep convolutional neural network: Snumedinfo at lifeclef plant identification task 2015.," in *CLEF (Working Notes)*, 2015.

[5] Luiz G Hafemann, Luiz S Oliveira, and Paulo Cavalin, "Forest species recognition using deep convolutional neural networks," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*. IEEE, 2014, pp. 1103–1107.

[6] Mariusz Bojarski, Philip Yeres, Anna Choromanska, Krzysztof Choromanski, Bernhard Firner, Lawrence Jackel, and Urs Muller, "Explaining how a deep neural network trained with end-to-end learning steers a car," *arXiv preprint arXiv:1704.07911*, 2017.

[7] Heikki Saari, Ilkka Pölönen, Heikki Salo, Eija Honkavaara, Teemu Hakala, Christer Holmlund, Jussi Mäkynen, Rami Mannila, Tapani Antila, and Altti Akujärvi, "Miniaturized hyperspectral imager calibration and uav flight campaigns," 2013, vol. 8889, pp. 88891O–88891O–12.

[8] L. Markelin, E. Honkavaara, R. Näsi, K. Nurminen, and T. Hakala, "Geometric processing workflow for vertical and oblique hyperspectral frame images collected using uav," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XL-3, pp. 205–210, 2014.

[9] Eija Honkavaara, Heikki Saari, Jere Kaivosoja, Ilkka Pölönen, Teemu Hakala, Paula Litkey, Jussi Mäkynen, and Liisa Pesonen, "Processing and assessment of spectrometric, stereoscopic imagery collected using a lightweight uav spectral camera for precision agriculture," *Remote Sensing*, vol. 5, no. 10, pp. 5006–5039, 2013.

[10] Yann LeCun and Yoshua Bengio, "Convolutional networks for images, speech, and time-series," 1995.

[11] Xavier Glorot, Antoine Bordes, and Yoshua Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011, pp. 315–323.

[12] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, Software available from tensorflow.org.

[13] François Chollet et al., "Keras," 2015.

[14] Raghavendra Kotikalapudi and contributors, "keras-vis," https://github.com/raghakot/keras-vis, 2017.