TECHNICAL UNIVERSITY OF MUNICH

TUM Data Innovation Lab

Deep Learning on High-Res Multispectral Aerial Imagery

Authors	Felix Buchert, Sarah Dörr, Filippo Galassi, Max Helleis, Kamilia Mullakaeva
Mentor(s)	M.Sc. David Dohmen, M.Sc. Felix Horvat (Ocell GmbH)
Co-Mentor	M.Sc. Oleh Melnyk (Department of Mathematics)
Project Lead	Dr. Ricardo Acevedo Cabra (Department of Mathematics)
Supervisor	Prof. Dr. Massimo Fornasier (Department of Mathematics)

Feb 2020

Abstract

To assess and monitor the condition of a forest for research or management purposes, it is crucial to have trustworthy information on the number of trees in a given area as well as the spatial distribution of different tree species or even dead trees over the terrain. Classically, these forest inventories are carried out manually by ground surveys from which the locally sampled results are extrapolated to the entire region of interest. Recent advances in image analysis using Fully Convolutional Neural Networks (FCNN) combined with efficient large scale data acquisition using aircrafts render an automation of this inventory process feasible. In this study, tree detection models trained on high-resolution multispectral imagery are developed by testing a variety of U-Net-based architectures. We provide a comprehensive analysis with respect to their performance on both tree localization and tree species classification. Furthermore, different definitions of ground truth label masks are explored and evaluated with respect to their eligibility for the given task. **[...]** The best performing approach, which was developed based on the combination of these findings, achieves an improvement in F1 Score of 9 to 14 percent.

Table of contents

1 Introduction

Note: This is the censored version of this report. Abridged content has been marked with [...].

1.1 Motivation

In times when the whole world is entering the age of automation and machine learning, it seems astonishing that a considerable part of forest management still depends on manual labour. OCELL is a technology platform that strives to change this by delivering an entirely new access to high-resolution aerial imagery and powerful analysis tools which finally enables data-driven process optimization in the industrial forest and farming sector. An example for a process that can be automated entirely is the forest stock taking, which is used to determine the overall stock volume of a forest estate. This generally takes 50 employees and more than six weeks of manual labour in the forest. The traditional approach is based on manually counting trees and identifying their species from which then estimations for entire regions are extrapolated based on the limited sample set of ground measurements. Therefore, in addition to saving costs and time, geo-information captured from high-resolution multi-spectral aerial imagery promises to yield significantly more accurate estimates of tree and species count compared to results obtained with the traditional, manual approach. This would give forest managers or public authorities timely access to in-depth analysis of valuable geo-information allowing them to precisely monitor the forests' condition and optimize forest management processes.

Apart from forest stock taking, models analyzing high-resolution multispectral imagery of forests could also be used to provide regular, automated health monitoring of the forests. Tools like these could prove to be critical in detecting and preventing tree diseases and infestations such as bark beetles at an early stage and therefore help to prevent big ecological and financial damages.

As outlined in the previous paragraphs, applying modern image-processing techniques to high-resolution mutlispectral imagery shows promise for a variety of applications in forestry. To further improve the capabilities of their existing machine learning systems for tree detection and species classification, OCELL teamed up with the Data Innovation Lab of the Technical University of Munich to form this project.

1.2 Problem Statement and Goals

The main goal of this project was to improve upon OCELL's current approach to tree localization and species classification on high-resolution multispectral imagery. On a basic level, OCELL approaches this task as an image segmentation problem. A Fully Convolutional Neural Network (FCNN) taking the high-resolution images as input is trained on fully annotated segmentation maps encoding the different tree species. Tree centers and species can then be extracted from predicted segmentation maps of a trained model.

In corporation with OCELL, three components of the current pipeline were identified as potential points of improvement over the course of the project. **[...]**

The present document is structured in the following way: In Section 2, we provide a comprehensive overview of the data sets and pre-processing steps as well as the training and evaluation pipeline. In the following sections, we then go on to describe the methodology and present results for each of the three previously mentioned approaches. In Section 6 we report and discuss further results concerning tree species specific performance, the effectiveness of the chosen data augmentation and model performance on an entirely new data set. Finally, we briefly summarize and discuss our results in Section 7 before concluding with a brief discussion of possible future directions of work in Section 8.

2 Setup

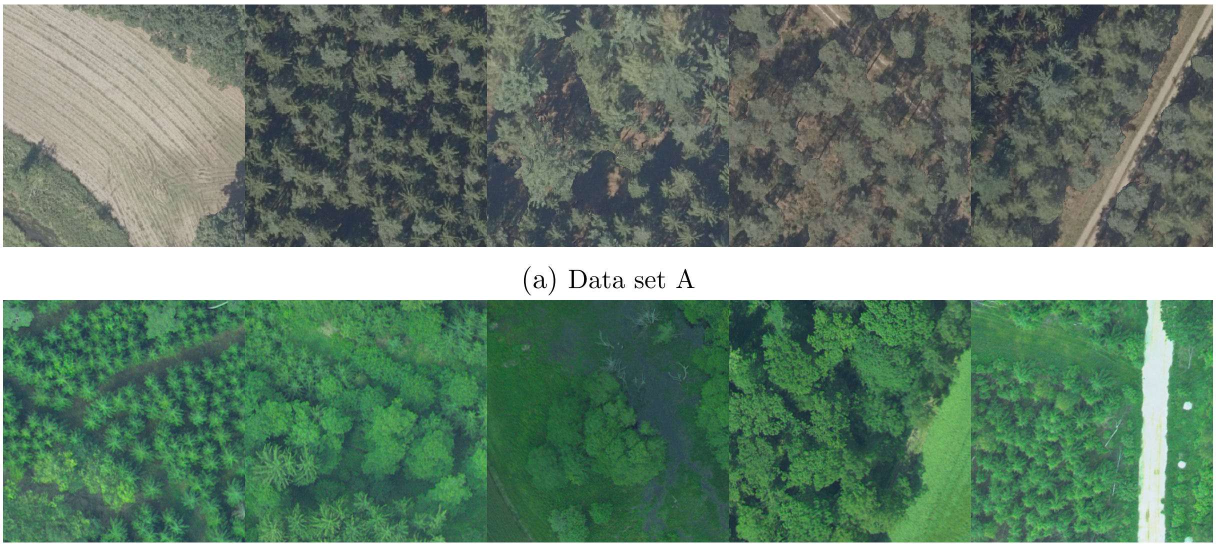
The setup that was used in terms of the data sets, data pre-processing, data augmentation and the training and evaluation pipeline will be described in the following.

2.1 Data Sets

OCELL provided us with two different data sets, herein referred to as data set A and B. Both of them contain orthorectified aerial images acquired over different regions in Germany using a sensor module developed by the company. An orthorectified image is a geometrically corrected image in which every pixel is labeled with its real world coordinates in a chosen coordinate system. The sensor module was mounted on an ultra light aircraft. The key parameters for both data sets are provided in Table 1. All results presented within the scope of Approaches 1 to 3 have been obtained on data set A. Data set B was only used for evaluation in Section 6.2 and Section 6.3.

Images of data set A consists of 3 bands in the optical domain of the electromagnetic spectrum (red, green, blue) and an additional band in the near-infrared (NIR) domain, as well as a separate digital surface model (DSM) which contains height information for every pixel, relative to the ground level.

[...] In Figure 1 some samples of both data sets are provided.



(a) Data set A

(b) Data set B

Figure 1: **Data sets:** 5 samples from data set A and B, only RGB channels

Table 1: Parameters of the two data sets provided by OCELL

Dataset	**Bands**	**Wavelength** $[\mu m]$	**GSD** $[m]$	**Width** $[km]$	**Height** $[km]$
A	Red	0.62 - 0.68	0.1 x 0.1	8.4	9.0
	Green	0.52 - 0.60	0.1 x 0.1		
	Blue	0.45 - 0.50	0.1 x 0.1		
	NIR	0.75 - 0.90	0.1 x 0.1		
	DSM	-	0.2 x 0.2		
B	Red	0.62 - 0.68	0.1 x 0.1	5.6	4.6
	Green	0.52 - 0.60	0.1 x 0.1		
	Blue	0.45 - 0.50	0.1 x 0.1		
	DSM	-	0.36 x 0.36		

An example for the DSM can be seen in Figure 2. The lower part shows the 2D image whereas the upper part visualizes the height information encoded in each pixel as a 3D-model. The height is color-coded ranging from green (0 m) to white to brown (33 m).

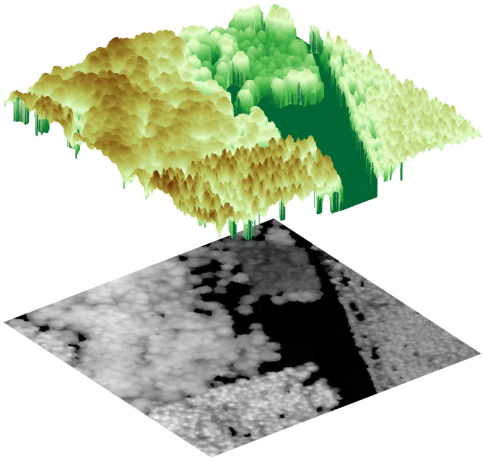


Figure 2: Digital Surface Model (DSM). The 3D view (upper part) shows the modelled tree canopy. The 2D view (lower part) shows the plain image data.

Figure 3 shows examples for the used optical channels red, green, blue and NIR. The high reflectance of the NIR band can be seen, as well as the comparatively high reflection in the green band, which is typical for vegetation.

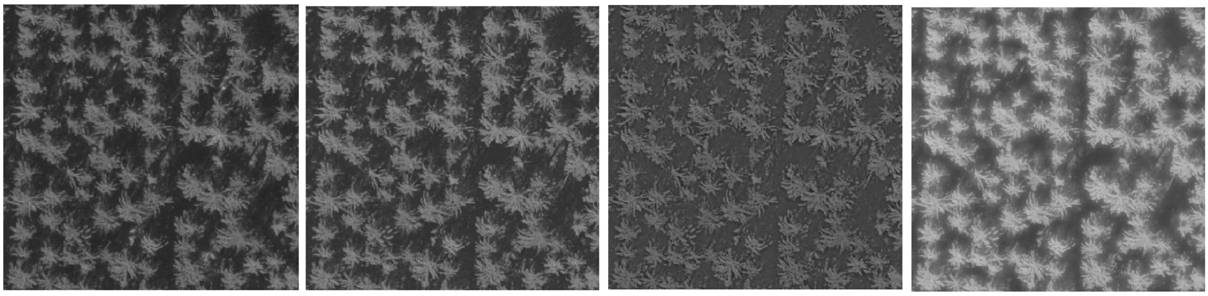


Figure 3: Examples for the red, green, blue and near-infrared channels, ordered from left to right.

2.2 Data Preprocessing

Neural networks are highly sensitive to noise or light conditions in image data. As recording data during different seasons or illumination conditions might change the color spectrum significantly, this poses a serious problem. We can see in Figure 1 that data set A is visually quite different from data set B. [...]

2.2.1 Data augmentation for multispectral images

[...]

2.2.2 Data split

Both data sets were randomly split in a train, validation and test data set containing 70%, 20% and 10% of the data respectively. In order to ensure comparability of results, the same split was used in every training process. Data set A has a total count of 161 images, whereas data set B has a total count of images. Table 2 presents the number of samples of each tree species in each subset.

The training and validation data set is used during our training pipeline, whereas the test set is only used for comparing the scores of different training setups. During the training process the training data is used for optimizing the model. Based on its performance on validation data, we extract the best performing model.

Table 2: Distribution of tree species in the data sets

Data Set	Data Split	Dead Tree	Douglas fir	Larch	Leaved Tree	Pine	Spruce
A	all	93	925	1380	1034	1305	3582
	training	78	654	850	771	880	2600
	validation	14	161	212	226	267	675
	test	0	82	224	36	158	158
	test (filtered)	0	82	224	33	146	158
B	all	117	0	0	3119	774	3923

2.2.3 [...]

2.3 Training Pipeline

During the project we developed a custom training pipeline which automatizes all steps and decisions of the training process. It was designed with two main requirements in mind. It needed to be flexible enough to be easily usable for all different approaches. It also had to be efficient as the in-depth analysis of all approaches required training many architectures in different setups (input data, ground truth label masks). Therefore, the training of different models is run automatically in a sequential manner. Apart from that, the training process comprises of the following setup:

- **Choice of best model:** During the training process the current model is evaluated on the validation data set. The best model is then chosen by comparing the validation score.
- [...]
- [...]

In the context of the training process, we refer to one iteration as the processing of a single batch. One epoch is defined as the pass over all samples contained in the training data set.

2.3.1 Label Definition

[...]

2.3.2 Loss function

[...]

2.4 Evaluation Pipeline

In the context of OCELL's goal of tree localization and species classification, a models' performance cannot be reasonably evaluated based on pixel-wise metrics. **[...]**

2.4.1 [...]

2.4.2 [...]

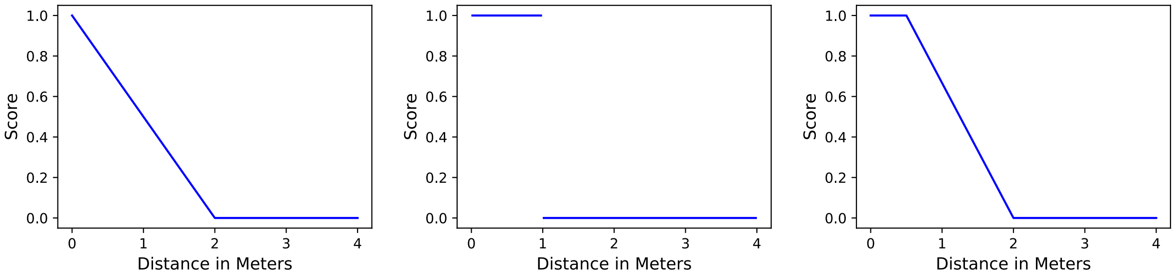
2.4.3 Score Definition

Finally, the most commonly used metrics in classification tasks, Recall, Precision, F1 Score and Accuracy, can be computed. In order to gain a more differentiated understanding of a model's performance, scores for tree center localization and species classification are calculated separately. Moreover, we introduce an aggregated score allowing us to compare overall performance, i.e. on both tree localization and species classification. With respect to the task of tree localization not only a discrete score, i.e. binary encoding whether a tree was detected or not, is of interest. Therefore, also continuous scores are introduced which are defined as functions of the distance between the ground truth and the predicted tree center.

Metric Definition Performance measures of classification tasks are generally calculated from *true positives* (TP), *false positives* (FP), *true negatives* (TN) and *false negatives* (FN). Based on these definitions, the most common measures, Recall, Precision, F1 and Accuracy, can be calculated. **[...]**

We can then define the metrics for a class c as

$$\begin{aligned}\text{Precision}_c &= \frac{TP}{TP+FP}\\ \text{Recall}_c &= \frac{TP}{TP+FN}\\ \text{F1-Score}_c &= 2\cdot\frac{\text{Precision}_c\cdot\text{Recall}_c}{\text{Precision}_c+\text{Recall}_c}\\ \text{Accuracy}_c &= \frac{TP+TN}{TP+FN+FP+TN}\end{aligned}$$



(a) Linear Score (2)

(b) Discrete Score (1)

(c) Piecewise Linear Score (0.5, 2)

Figure 4: **Score functions**

and the total F1-score as

$$\text{F1-Score}_w = 2 \cdot \frac{\text{Precision}_w \cdot \text{Recall}_w}{\text{Precision}_w + \text{Recall}_w}.$$

[...]

Figure 4 illustrates possible configurations of these score functions. **[...]**

Class Scores A models' performance on the species classification task is assessed using two different score types:

- **Class-only:** Classification metrics are only computed with respect to matched label-prediction pairs
- **Aggregate Center-Class:** Classification metrics are computed with respect to all labels and predictions

[...]

3 Approach 1 – Architectures

The first approach to improve upon OCELL's current state was to directly improve the employed neural network architecture. **[...]**

3.1 Introduction of Network Architectures

[...]

3.2 Results

As introduced in Section 2.4.3 the performance of a model is separately evaluated on center localization and tree species classification. Moreover, we assessed the aggregated class-center scores to obtain a combined metric for class and center prediction. **[...]**. Throughout the tables the names of the score functions are shortened, such that PL, D and L refer to piecewise-linear, discrete and linear score function.

For all different architectures proposed in Section 3.1 both metrics, F1 Score and Accuracy, are evaluated. The results of different score functions from Section 2.4.3 are shown in Table 3. It is noticeable that the Architectures 1-3 outperform all other models followed by Architecture 7. Especially, Architecture 1 yields the best scores for center prediction. Since Architecture 1 uses a **[...]** as encoder, it is more powerful and therefore performing better than the other Architectures 1-3. The same behavior can be seen between Architecture 2 and Architecture 3. Generally, by using a smaller network, such as Architecture 3, less computations are needed and thus training and evaluation are faster. Furthermore, a smaller number of trainable parameters serves as a "natural" form of regularization by limiting the expressiveness of a model. As a general heuristic, the smaller network should be used if there is no significant difference in performance between two models.

Another reason, why Architectures 1-3 outperform the basic Architecture 7, might be that no pre-trained encoder could be used for Architecture 7. The same holds for the Architecture 6. By using a pre-trained encoder, the network needs less epochs for training and fewer hyperparameter tuning which could in turn be another explanation for the superior performance of Architectures 1-3.

Architectures 4, 5 and 6 only predict 10 to 70 trees correctly of a total count of more then 600 trees. Therefore, class prediction results in the following paragraph have to be carefully considered for these architectures, since class metrics are computed on only a few correctly detected trees. Even though, **[...]** is a very deep neural network, it usually faces the vanishing gradient problem, which might be an explanation for their poor performance on the center prediction task. By applying shortcut connections as described **[...]** can solve this issue.

Table 3: Prediction Scores

		Architecture 1	Architecture 2	Architecture 3	Architecture 4	Architecture 5	Architecture 6	Architecture 7
PL(0.5, 2)	F1 Score	**0.69**	0.64	0.59	0.12	0.06	0.05	0.44
	Accuracy	**0.52**	0.48	0.42	0.07	0.03	0.03	0.28
D(1)	F1 Score	**0.67**	0.63	0.57	0.12	0.05	0.05	0.42
	Accuracy	**0.50**	0.46	0.40	0.07	0.03	0.03	0.26
L(2)	F1 Score	**0.54**	0.51	0.47	0.10	0.05	0.04	0.35
	Accuracy	**0.37**	0.34	0.30	0.05	0.02	0.02	0.21

Class Prediction Secondly, we evaluate the sample-weighted F1 Score and Accuracy, defined in Section 2.4.3, for the tree species prediction. Table 4 summarizes our findings with respect to all architectures proposed in Section 3.1. Generally, we can see that the classification results differ significantly from the center predictions. Most models perform better than 80 percent on both score functions. First it might be surprisingly that Architecture 4 performs mostly better than all other architectures, but with respect to the center predictions we have to treat these results carefully. Since only 70 trees are detected correctly, it might be that these are only from one species. Therefore, the comparison is questionable. Apart from that, we can observe that all Architectures 1-3 and Architecture 7 perform similarly well. Taking into account the considerably larger variations in performance with respect to tree center predictions, this indicates the that a key aspect for a model's overall performance is its capacity to correctly identify tree centers.

Table 4: Sample-Weighted Class Scores

		Architecture 1	Architecture 2	Architecture 3	Architecture 4	Architecture 5	Architecture 6	Architecture 7
D(1)	F1 Score	0.94	0.94	0.94	**0.95**	0.78	0.80	0.94
	Accuracy	**0.88**	0.86	0.86	0.87	0.49	0.59	0.85
D(2)	F1 Score	0.94	0.94	0.94	**0.95**	0.76	0.80	0.93
	Accuracy	**0.88**	0.86	0.87	**0.88**	0.48	0.58	0.84

Aggregate Class-Center Scores Lastly, the aggregated results for both measurements, tree localization and species prediction, are reported in Table 5. It appears, that the overall result is influenced more by the center prediction than by the class prediction. In general, Architectures 1-3 again outperform the other network architectures.

Table 5: Aggregate Class-Center Scores

		Architecture 1	Architecture 2	Architecture 3	Architecture 4	Architecture 5	Architecture 6	Architecture 7
D(1)	F1 Score	**0.41**	0.37	0.32	0.06	0.01	0.01	0.21
	Accuracy	**0.59**	0.54	0.49	0.11	0.03	0.03	0.35
D(2)	F1 Score	**0.54**	0.48	0.43	0.07	0.02	0.02	0.28
	Accuracy	**0.70**	0.65	0.60	0.13	0.04	0.03	0.44

3.3 Conclusion

Within this approach we evaluated the performance of different network architectures. As the Architecture 1 performs well on all three evaluation methods, its choice is generally reasonable. However, using Architecture 2 or Architecture 3 instead, increases efficiency with respect to training time, GPU memory and inference time. Further improvements might be observed changing the skip connections between the encoder and the decoder. Currently, low-level feature maps are added to high-level feature maps, which might cause some discrepancy throughout the learning and thus adversely affect the prediction procedure. **[...]**

4 Approach 2 – Redefining Labels

The second approach to improving upon OCELL's current state explored to what extent the definition of label masks influences a models' performance. **[...]** Hence, in addition to evaluating the different neural network architectures described in Section 3, different labelling techniques were assessed. **[...]**

4.1 Label Definitions

All label definitions we considered for the evaluation are listed and comprehensively described in the following. **[...]**

4.2 Tree localization and Classification

The models are scored as outlined in Section 2.4. **[...]** The methods used to extract these points depend on the labeling technique and are described in the following. **[...]**

4.3 Results

The performance of the models trained on the different label masks introduced in Section 4.1 is evaluated with respect to the two main tasks of interest: **[...]** and tree species classification. Moreover, the aggregate class-center metric definition, described in Section 2.4.3, is used to compare overall performance on both tasks. In the following, we give both a quantitative evaluation as well as a qualitative analysis of the labeling techniques. While the former focuses on the plain comparison of scoring metrics, the latter investigates possible explanations for the observed results.

4.3.1 Quantitative Evaluation

[...]

Table 6: Prediction F1 Scores

		Architecture 1	Architecture 2	Architecture 3	Architecture 4	Architecture 5	Architecture 6	Architecture 7
PL(0.5, 2)	Def 1	**0.69**	0.64	0.59	0.12	0.06	0.05	0.44
	Def 2	**0.32**	0.30	0.26	0.07	0.04	0.11	0.18
	Def 3	**0.75**	0.70	0.71	0.61	0.45	0.71	0.65
	Def 4	0.72	**0.75**	0.74	0.24	0.37	0.69	0.47
D(1)	Def 1	**0.67**	0.63	0.57	0.12	0.05	0.05	0.42
	Def 2	**0.29**	0.27	0.21	0.06	0.03	0.09	0.16
	Def 3	**0.75**	0.69	0.70	0.61	0.45	0.69	0.63
	Def 4	0.72	**0.75**	0.74	0.22	0.35	0.66	0.45
L(2)	Def 1	**0.54**	0.51	0.47	0.10	0.05	0.04	0.35
	Def 2	**0.25**	0.23	0.20	0.05	0.03	0.09	0.14
	Def 3	**0.60**	0.56	0.57	0.48	0.36	0.56	0.51
	Def 4	0.58	**0.60**	**0.60**	0.19	0.29	0.54	0.37

In accordance with our findings in Section 3, it can be readily seen that Architectures 1, 2 and 3 outperform all other architectures on average. Moreover, an architecture trained on the **[...]**labelling techniques consistently outperforms the same architecture trained on the two other label definitions. For Architecture 1, an improvement between 9 to 12 percent can be observed, depending on the score function. **[...]**

Interestingly, the Architecture 1 consistently performs best on the generic-center label definition, while Architecture 2 outperforms all other networks in the two model approach. Although the difference in performance is not substantial, this pattern can be observed over all score functions. A possible explanation for this result might be given by the varying degree of complexity required in training on the label definitions. Accordingly, the deeper Architecture 1 might be better suited for the more complex labeling technique whereas the Architecture 2 exhibits an appropriate number of learnable parameters for the two model label definition consisting of two "simpler" models.

Class Prediction In a second step, F1 scores calculated with respect to the tree species classification task are evaluated for all model architectures. In Table 7 F1 scores are reported based on the weighted class score definition introduced in Section 2.4.3. Again, the Architectures 1, 2 and 3 architectures yield the best results on average, while generally achieving higher absolute F1 scores ranging from 0.75 to 0.89 compared to the F1 scores for center prediction. Interestingly, the observed improvement of F1 scores from 0.88 to 0.89 with respect to species classification is significantly smaller than the improvements observed with respect to center prediction.

Table 7: Sample-Weighted Class F1 Scores

		Architecture 1	Architecture 2	Architecture 3	Architecture 4	Architecture 5	Architecture 6	Architecture 7
D(1)	Def 1	**0.88**	0.86	0.86	0.87	0.49	0.59	0.85
	Def 2	**0.81**	0.79	0.80	0.48	0.52	0.45	0.65
	Def 3	0.87	0.88	**0.89**	0.86	0.83	0.83	0.87
	Def 4	0.88	**0.89**	0.88	0.85	0.88	0.81	0.85
D(2)	Def 1	**0.88**	0.86	0.87	**0.88**	0.48	0.58	0.84
	Def 2	**0.81**	0.79	0.75	0.40	0.49	0.40	0.62
	Def 3	0.88	**0.89**	**0.89**	0.84	0.81	0.83	0.87
	Def 4	0.88	**0.89**	**0.89**	0.87	0.87	0.82	0.85

[...]

Table 8: Aggregate Class-Center F1 Scores

		Architecture 1	Architecture 2	Architecture 3	Architecture 4	Architecture 5	Architecture 6	Architecture 7
D(1)	Def 1	**0.59**	0.54	0.49	0.11	0.03	0.03	0.35
	Def 2	**0.23**	0.21	0.17	0.02	0.01	0.04	0.10
	Def 3	**0.65**	0.60	0.63	0.52	0.37	0.57	0.54
	Def 4	0.63	**0.67**	0.65	0.19	0.31	0.53	0.38
D(2)	Def 1	**0.70**	0.65	0.60	0.13	0.04	0.03	0.44
	Def 2	**0.35**	0.32	0.29	0.04	0.03	0.06	0.15
	Def 3	**0.75**	0.71	0.73	0.59	0.41	0.69	0.65
	Def 4	0.73	**0.76**	**0.76**	0.25	0.40	0.66	0.47

4.3.2 Qualitative Evaluation

After having evaluated performance with respect to the different labelling techniques, we provide a qualitative analysis exploring possible explanations for the strong variation of performance. In order to illustrate our findings, we chose the architectures Architecture 1 and Architecture 6 as their results are best suited to explain advantages and disadvantages of label definitions in a clear and descriptive manner.

[...]

4.4 Conclusion

[...]

5 Approach 3

[...]

5.1 Architectures

The architectures used were the same as described in section 3. Since the provided framework had so far only been used with RGB data, the whole pipeline had to be adapted to be able to ingest the additional bands.

[...]

To investigate the impact of providing additional information to the models, all models were trained using the additional information.

[...]

5.2 Results

[...]

5.3 Conclusion

[...]

6 Further Results

In addition to the central results presented in the previous three sections, this section presents further notable observations and results.

First of all, it is important to note that not all tree species are equally important to OCELL and its customers. Therefore, a review of model performance by tree species is presented in Section 6.1.

[...]

6.1 Tree Species

In the previous sections, models have only been evaluated with respect to the overall score. [...]

6.2 Data Augmentation

[...]

6.3 Comparison of generic-center and two model

[...]

7 Conclusion

In this study we investigated several aspects of training deep neural networks on high resolution multispectral aerial imagery for tree detection. Firstly, several state-of-the-art Fully Convolutional Neural Network (FCNN) architectures for image segmentation were evaluated and compared. The results showed that Architectures 1, 2 and 3 consistently performed best in tree localization and species classification.

[...]

[...]

8 Future Research

Suggestions for future work based on the findings of this study are described in the following. Suggestion are roughly ordered according to our assessment of their potential impact on model performance from high to low. Of course, this ordering is subjective and might change depending on prioritization of the main goals.

[...]