Data Science in Earth Observation

Tree Species Classification

Project Report

Group members:

Manap Shymr
Kirill Volter
Jayendra Chorapalli
Sirui Wang
Vincent Orth

Under the supervision of:

Prof. Dr. -Ing. habil. Xiaoxiang Zhu
Prof. Dr. -Ing. Muhammad Shahzad
Mr. Yang Mu

M.Sc ESPACE
Technical University of Munich
February 09, 2023

Contents

Abstract

Tree species classification plays an important role in the fields of ecosystem monitoring and biomass prediction. Machine learning (ML) for Remote Sensing (RS) method provides an efficient way to classify tree species with low cost and high accuracy. To not only get information about location and extent but also about the species distribution high accuracy labeled tree species data is needed. In this project work the possibilities of the EU-Forest dataset [Mauri et al., 2017] were tested for tree species classification using a machine learning approach. ESA's Sentinel-2 data was used as the main dataset for the analysis. After cloud masking, three season-stacks have been generated. Vegetation indices have been calculated as extra features and were added to the stacks. 5 tree species were classified by Random Forest (RF), Artificial Neural Network (ANN), Convolutional Neural Network (CNN), Residual Neural Network (ResNet) [He et al., 2015] and Recurrent Neural Network (RNN) [Mou et al., 2017]. Based on the high-performing models, a decision fusion model was built. The Recurrent Residual Convolutional Neural Network (RRCNN) with CNN, Resnet, and RNN. The final overall accuracy varies from 49.352% (ANN) to 73.33% (Resnet 50) for the validation but lies only at 20.40% for the testing (RNN).

Keywords: tree species classification; Sentinel-2; multi-temporal; deep learning

1. Introduction

Earth observation (EO) has a high potential for biodiversity assessments, mainly for the description of vegetation habitats [Kuenzer, et. al., 2014]. Tree species diversity is very important for the ecosystem. For tree species classification there is a need for large annotated high-resolution tree species data sets. The EU-Forest data set [Marui et al. 2017] provides harmonized species occurrence data in a 1km x 1km INSPIRE grid for Europe. In this project work only Germany was considered. Within the EU-Forest data set, Germany contains a total of 84 tree species and 67,412 sample plots.

In our approach, the EU-Forest occurrence points were first filtered and then harmonized by sample size using data augmentation. The Sentinel-2 reference scenes were cloud masked and aggregated in 3 seasonal stacks. 5 vegetation indices were calculated and added to the seasonal stacks. The five tree species with the highest amount of occurrence data were selected for the classification (1-Picea Abies, 2- Fagus sylvatica, 3-Pinus sylvestris, 4-Quercus robur, and 5-Betula pendula).

The data were classified using different deep-learning approaches. Selected were: Random Forest (RF), Artificial Neural Network (ANN), Convolutional Neural Network (CNN), Residual Neural Network (ResNet) [He, 2015], and Recurrent Neural Network (RNN). For these five methods, fine-tuning was conducted, and based on the performance of the models, a decision fusion model Recurrent Residual Convolutional Neural Network (RRCNN) was built.

The model performance was tested on a 20% data split (validation) and on an unknown test data set provided by the supervisors. The workflow can be seen in Figure 1.

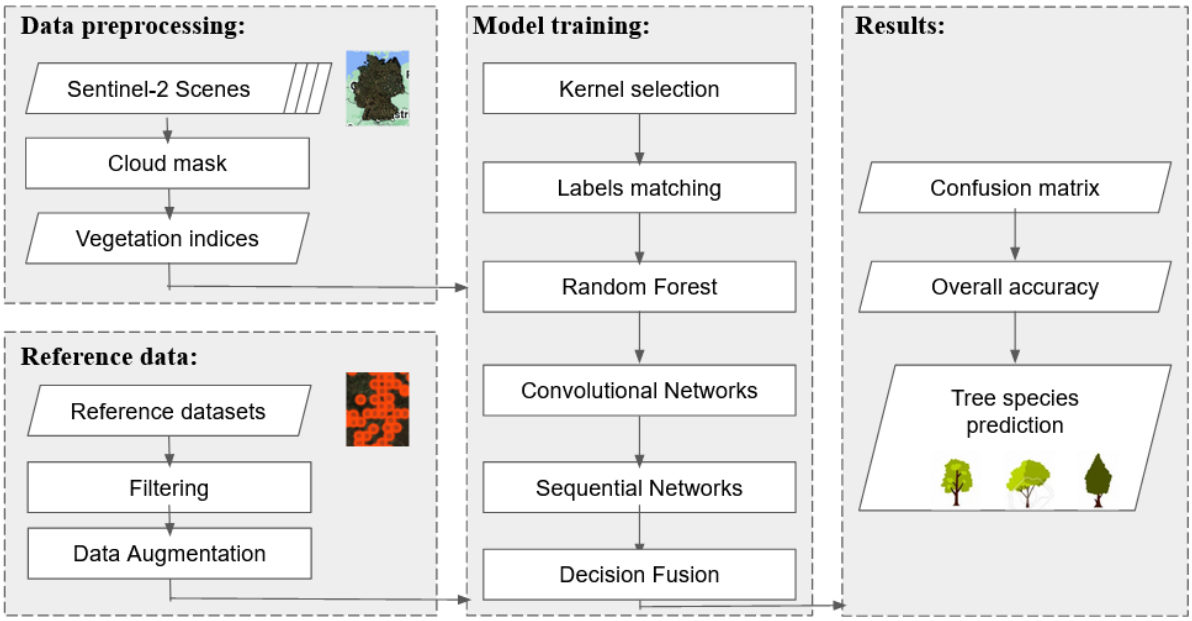

Figure 1 Workflow of tree species classification

2. Data Curation

2.1 Data Acquisition

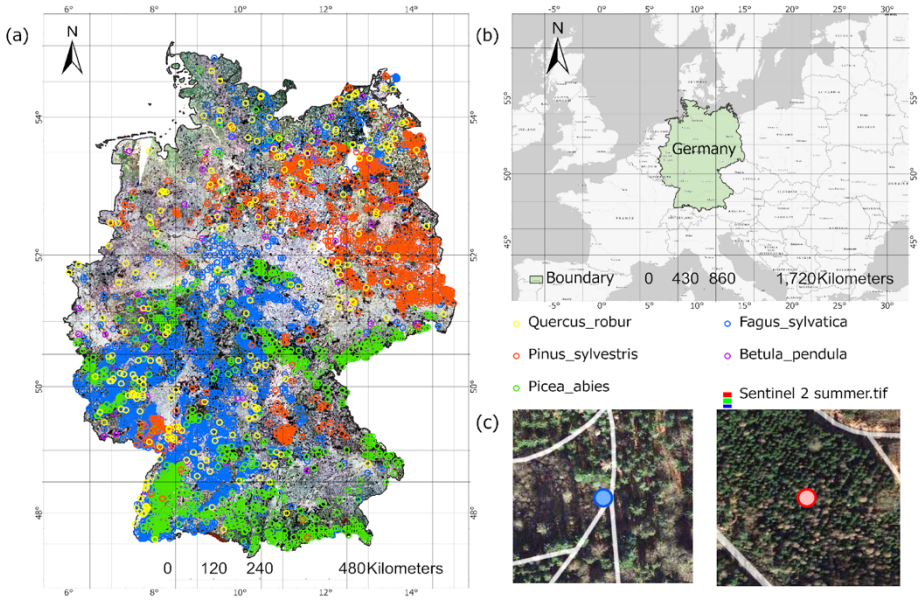

Figure 2 (a) Regular grid of reference data feature collection and Sentinel-2 RGB scene. (b) Location of ROI in Europe. (c) Examples of samples on Google Earth map (left is label of broad leaf tree; right is label of needle leaf tree).

In Figure 2 the distribution of the five species over Germany (a), the location of Germany within Europe (b) and two examples of EU-Forest points (c) can be seen. In Table 1 all the datasets that were used in this project are presented. The Sentinel-2 imagery and the reference EU-Forest species data set are the core of our approach. While the very high-resolution CNES/Airbus images and the CGLS Landcover layers were used as auxiliary data for our filtering.

Dataset Name	Time	Resolution (m)	Description
Sentinel-2	2021	10	10 Multispectral bands with 3 seasons
Reference data	2017	1000	Labels of tree species of EU-Forest
CNES/ Airbus	2023	0.5	Very high-resolution images on Google Earth from The French Space Agency (CNES) for interpretation
CGLS-LC100	2019	100	Copernicus Global Land Cover Layers Collection 3 for forest mask

Table 1 Overview of Datasets

2.2 Data Preparation

Labelled reference data preparation: The German tree species data of EU-Forest [Mauri et al., 2017] is used as the reference data, which contains a total of 84 tree species and 67,412 sample plots. The data was provided in two '.csv'-Files. The first contains the (X, Y) coordinates and the species information. The second is the statistical result sorted for the different species. The data was converted to WGS84 projection and uploaded to Google Earth Engine (GEE).

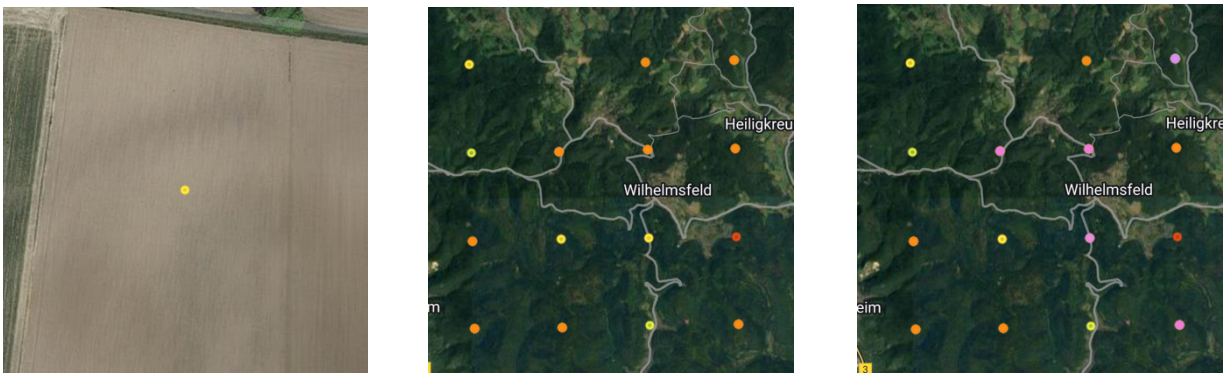

Figure 3 100x100 meters grid example for the non-forest point (left). and 5x5km example for location overlapping (middle and right)

Filtering: A 24 by 24 pixel-kernel based on Sentinel-2 images around the reference points were obtained as training data. Since the resolution of the gridded INSPIRE points is 1 km × 1 km, the location of the point corresponds to the center of the grid cell in which the species was found and not to the true location of the species like in Figure 3. So it could happen that the point was not in a forest area Figure 3a or that different species shared the same location Figure 3b and 3c. Therefore the reference points need to be filtered by several properties to make sure the kernel areas are meaningful and interpretable. The following filtering steps were used:

1. Semantic Filtering:
   The Copernicus Global Land Cover Layers (CGLS) was used as an auxiliary forest filter. All points kept, that lie within their respective CGLS-forest type (70% canopy density). The filtering rule was:
   a) **evergreen needle leaf** for species 1 (*Picea abies*) and 3 (*Pinus Sylvestris*)
   b) **deciduous broadleaf** for species 2 (*Fagus sylvatica*), 4 (*Quercus Robur*) and 5 (*Betula pendula*)

2. Unique Location:
   Locations corresponding to multiple species had to be deleted and only truly unique combinations of species and locations were kept. For this filtering step, the coordinate distinguishing method on Google Earth Engine was used.

3. Visual Check:
   To make sure the training data points are located in the dense forest, a visual comparison for a random subset of the data points was done. The INSPIRE data points were manually compared with CNES/Airbus very high resolution (VHR) images of 0.5m spatial resolution available on Google Earth Engine. The 5-6 years timestep, between the acquisition of the INSPIRE plot points in 2017 and the CNES/Airbus VHR images in 2023, was considered to be neglectable since forests grow and change over longer time intervals.

Satellite image data preparation: Since the test data provided was based on spring, summer and autumn, only these 3 seasons were considered in this approach. All images are grouped per season (Spring: March 1st - May 31; Summer: June 1st - August 31; Autumn: September 1st - November 30). The 10 spectral bands providing valuable information for vegetation are selected (visible bands 2, 3 and 4; red-edge bands 5, 6 and 7; NIR bands 8 and 8A; as well as the SWIR bands 11 and 12). 24 by 24 pixel-kernel patches around sample points were obtained from Google Earth Engine. Seasonal composites are stacked, which contain 30 bands (3 seasons with 10 bands each).
In Figure 4 the spectral lines of the 5 tree species can be seen after the filtering for the three seasons. With the filtering the discriminability of the species clearly improved. Also for summer and autumn season a clear difference in the behavior of the the evergreen needle leaf trees (*1-Picea Abies*, *3-Pinus sylvestris*) can be seen compared to the deciduous broadleaf trees (*2-Fagus slyvatica*, *4-Quercus robur* and *5-Betula pendula*).

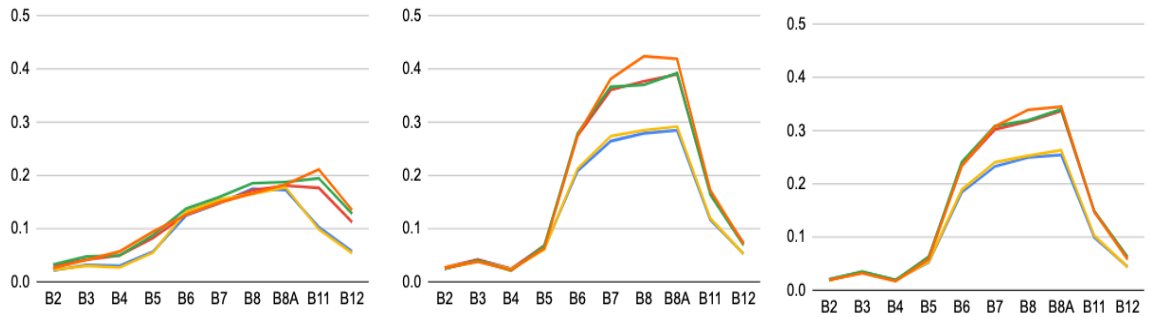

Figure 4 Spectral lines of tree species for Spring (left), Summer (middle) and Autumn (right)

Training and testing file preparation: In order to improve the speed of neural network training, all GeoJson files are converted to '*.npy*' files. Training and testing samples are separated by a ratio of 80% and 20%.

2.3 Data Pre-processing & Augmentation

2.3.1 Data Pre-processing

Indices Calculation: 5 indices have been selected and calculated to improve the performance of the model. These indices store key information regarding the growth, greenness, and water content in the trees [Immitzer, 2019]. The indices are presented in Table 2.

Vegetation Indices	Formulary
Normalized Difference Vegetation Index (NDVI)	(NIR-Red)/(NIR+Red)
Normalized Difference Water Index (NDWI)	(NIR-SWIR1)/(NIR+SWIR1)
Standard Ratio of NIR and RE1 (SRNIRRE1)	NIR/RE1
Normalized Difference Red-Edge and SWIR2 (NDRESWIR)	(RE2-SWIR2)/(RE2+SWIR2)
Greenness Index (GI)	Green/Red

Table 2 Vegetation Indices Table [Immitzer, 2019]

2.3.2 Data Augmentation

The below plot Figure 5. (a). shows the sample size for each species in the training + validation data set after the data pre-processing.

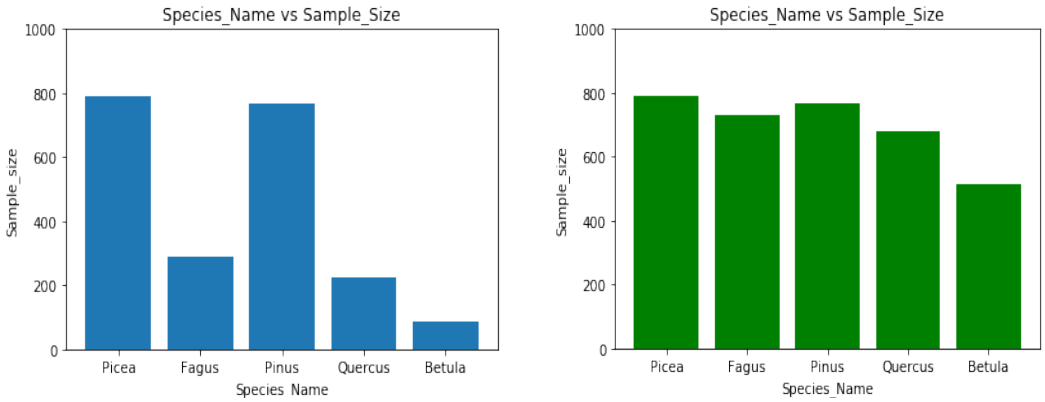

Figure 5 species number before augmentation (left); species number after augmentation (right)

It can be seen that the train+validation set is dominated by the *Picea Abies* and *Pinus Sylvestris*. *Betula Pendula* has the lowest number of 86 samples. Whereas the sample size for *Fagus Sylvatica* and *Quercus Robur* are around 300.

If we would train our model with this training data set, the model will be skewed towards the *Picea Abies* and *Pinus Sylvestris* as they make up around 75% of the overall data. Due to this reason, we improved the number of samples using data augmentation. The goal was to increase the size of the smaller species of the training dataset, so that the sample sizes are in the same range. Additionally, this step also increases the total number of training samples. This is important in Deep Learning (DL) models because larger datasets generally lead to better model performance. However, it may not always be feasible to acquire a large amount of annotated data. In these cases, data augmentation can be used to synthesize additional training examples from the available data.

Data augmentation was performed for *Fagus Sylvatica* and *Quercus Robur* by rotating the samples to 90 and 180 degrees and hence increasing the sample size. Due to the very small sample size of *Betula Pendula*, rotations of 90, 180 and 270 degrees and translation of pixels towards right and left by 4 columns were performed. Figure 6 shows an example of rotation performed on a *Quercus Robur* sample.

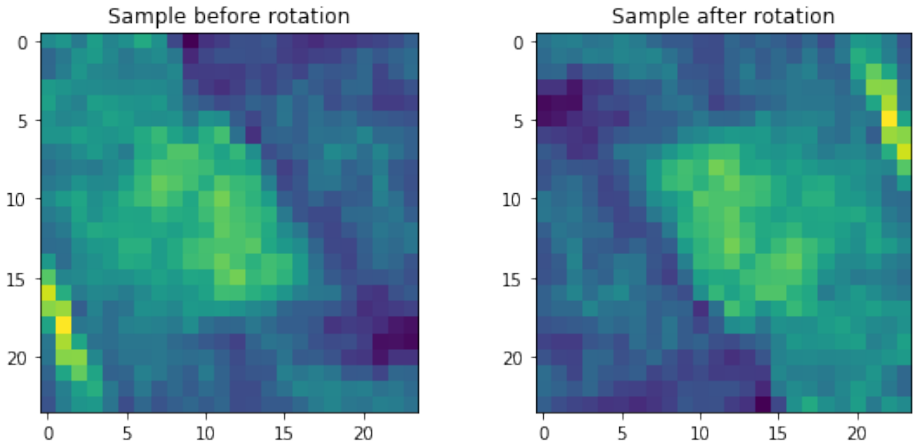

Figure 6 Rotating data augmentation samples

No	Species Name	Training + Validation data	Training + Validation data + augmentation	Testing data
1	Picea Abies	(790,24,24,45)	(790,24,24,45)	(100, 24,24,45)
2	Fagus Sylvatica	(291,24,24,45)	(732,24,24,45)	(100, 24,24,45)
3	Pinus Sylvestris	(767,24,24,45)	(767,24,24,45)	(100, 24,24,45)
4	Quercus Robur	(227,24,24,45)	(681,24,24,45)	(100, 24,24,45)
5	Betula Pendula	(86,24,24,45)	(516,24,24,45)	(100, 24,24,45)

Table 3 Samples of different tree species after data augmentation

Table 3 represents the shape of the training + validation data before the augmentation (column 3), after the augmentation (column 4) and the test data (column 5). The first value in the shape represents the number of samples, followed by the pixel size of the samples and the number of bands.

3. Model Selection

In this project, the five machine learning methods: Random Forest (RF), Artificial neural network (ANN), Convolutional neural network (CNN), Residual neural network (ResNet) [He, et al, 2015], Recurrent neural network (RNN) have been selected.

Random Forest [Breiman 2001] is a popular ensemble machine learning algorithm that is used for both regression and classification tasks. The algorithm works by building multiple decision trees, each trained on a bootstrapped sample of the training data, and aggregating the predictions of these decision trees to make a final prediction. It's a 1D approach, just one line vector with all the features. This approach gives fast results and helps to check if the dataset is ready.

Convolutional Neural Networks [Sharma et al., 2018] are well suited for tree classification while detecting the texture of patches and spectral features at the same time. They use convolutional layers to scan the image and extract local features, and pooling layers to reduce the spatial dimensions of the features. This allows CNNs to learn hierarchical representations of the image that are invariant to translations, rotations, and scaling's. We firstly use the simple neural network ANN with a flatten layer as 1D CNN. We have implemented a CNN method by adding 3 stages of 'Conv2D - MaxPool' layers and 2 stages of 'Flattening - Dense' layers.

ResNet-50 [He, et al, 2015] is a 50-layer convolutional neural network (48 convolutional layers, one MaxPool layer, and one average pool layer). Residual neural networks are a type of artificial neural network (ANN) that forms networks by stacking residual blocks. The architecture is named after the residual connections that are used to improve training of very deep networks, which were previously difficult to train effectively. In ResNet50, the residual connections allow information to flow more easily through the network, reducing the vanishing gradients problem and enabling the network to learn more complex representations of the data. The architecture is shown in Figure 7

Recurrent Neural Networks [Mou et al., 2017] are temporal sequential neural networks. Unlike traditional neural networks, which process information in a feedforward manner, RNNs have a feedback loop that allows them to process information in a recurrent manner, where the output of a neuron depends not only on its inputs, but also on its previous state. This makes RNNs well-suited for processing sequences, where the current output depends on the previous elements in the sequence. Spectral lines with 3 seasons can generate sequences containing spectral and temporal information.

Recurrent Residual Convolutional Neural Network: The fusion of high performance model features can improve the accuracy. The result after decision fusion [Hoffmann, 2019] will be robust to outliers. The features of different models were normalized to have the same weight for each network. We then chose the three models with good performance: Resnet, CNN and RNN and made a fusion. The result is our Recurrent Residual Convolutional Neural Network (RRCNN).

4. Model Training

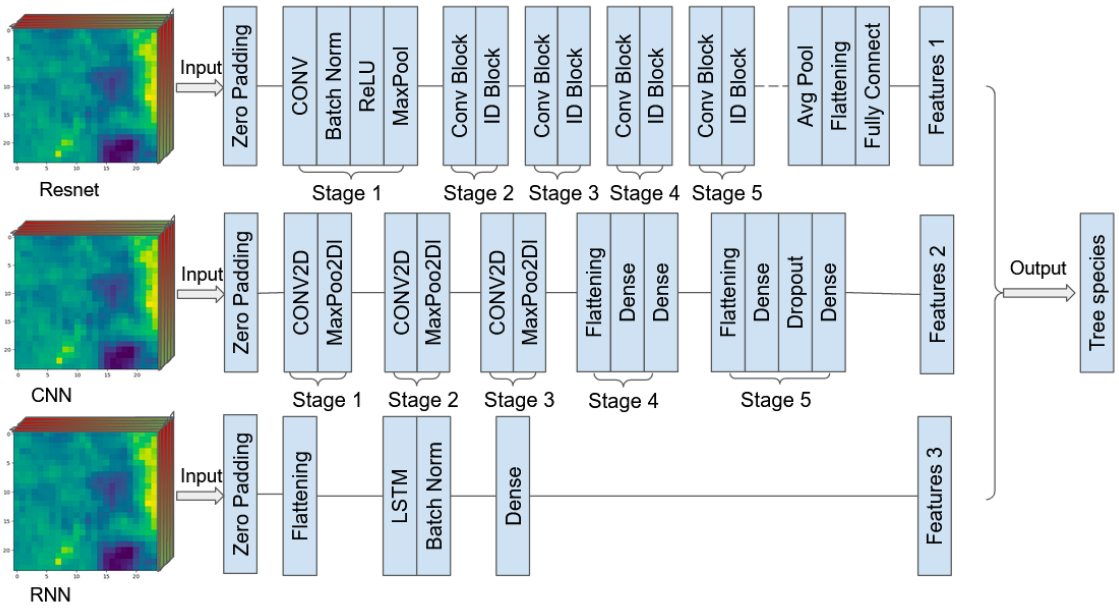

Figure 7Architecture of RRCNN Model

Adjustment of epoch number: For deep learning neural networks, it's easy to be overfitting after several epochs. In order to avoid this problem, we defined three callbacks to perform an early stop, they can be seen in Table 5. These three callbacks are suitable for different models and the *'early stop 3'* usually is used for fine tuning. Patience represents the number of epochs with no improvement after which training will be stopped.

Options	Early stop 1	Early stop 2	Early stop 3
Monitor	Validation loss	Validation loss	Validation accuracy
Min delta	0.005	0.005	0.005
Patience	20	10	5
Mode	Min	Min	Min

Table 4 Options of Early stop callback

5. Network Tuning

During initial training, only a few layers are trained on top of the base model. The weights of the pretrained network are not updated during training. To further improve the performance, the weights of the top layer of the pre-trained model are trained simultaneously. The training process will force the weights to be adjusted from a generic feature map to features associated specifically with the dataset. Also, fine-tuning the few top layers is very important since the higher a layer is the more specialized it is. The first few layers learn very simple and general features. As the number of layers

increases, the features become more and more specific to the dataset. The goal of fine-tuning is to increase the performance. These are the three main steps for fine tuning, that we conducted:

1. Unfreeze the top layer of the model: Unfreeze the base_model and set the bottom layer as non-trainable. Then recompile the model and resume training.

2. Compile model: Use a lower learning rate, otherwise the model may overfit quickly.

3. Continue to train the model: Using early stopping callbacks to detect training convergence improves accuracy by a few percentage points. The loss will decrease and the model will become more stable.

Taking the example of Resnet in Figure 8, the accuracy is higher and more stable after starting fine-tuning from the 50th epoch.

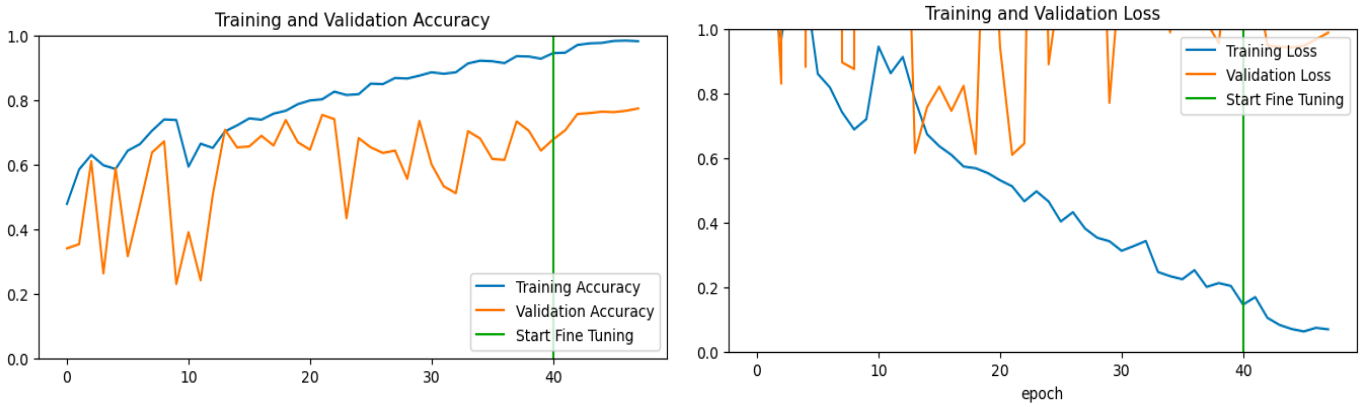

Figure 8 Accuracy and loss plot with epochs as x axis

6. Experimental Results

Model	Validation data (S/S)		Test data (S/S)	Validation data (S/S/A)		Test data (S/S/A)
	OA	Fine-tuning	OA	OA	Fine-tuning	OA
Random Forest	0.7259	– –	– –	**0.7704**	– –	**0.2387**
ANN	0.4993	0.5681	0.1510	0.4878	0.4935	0.1939
CNN	0.6801	0.6843	0.2020	0.6643	0.6714	0.1857
Resnet 50	0.7044	**0.7331**	0.1979	0.7059	**0.7733**	0.2000
RNN	0.2396	0.6499	0.1979	0.4648	0.6513	0.2040
RRCNN	– –	**0.7288**	**0.1979**	– –	**0.7733**	**0.2000**

Table 5 curacy table (S/S= spring and summer; S/S/A = spring, summer and autumn)

Based on this approach we got the final results presented in Table 6. The overall accuracy increased remarkably after fine tuning. Among all the single models, Resnet 50 has the highest accuracy for the validation data. The fusion model RRCNN also has very high accuracy. When we use two seasons, the fusion model RRCNN will be worse because of the performance of RNN. This happened due to the sequence length for RNN being reduced to 30. We also tried to predict the test data but only got an accuracy of 0.2 at the end.

Validation: In the confusion matrix (Figure 9a) it can clearly be seen that our model can distinguish and predict the 5 species with high accuracy. It performs best for both evergreen needle species (1-Picea Abies, 3-Pinus Sylvestris). For species 2-Fagus Sylvatica the prediction is still quite high even if there are some confusions with the other deciduous tree species (4-Quercus Robur, 5-Betula pendula). For species 4 and 5 the points wrongly classified as different deciduous tree species are the highest. But for all three deciduous forest species, the majority of points are correctly classified.

Test: The confusion matrix (Figure 9b) for the test case shows that all points get assigned into one class. Because of the uniform distribution of the samples in the test data set one fifth gets assigned correctly, therefore resulting in around 20% accuracy.

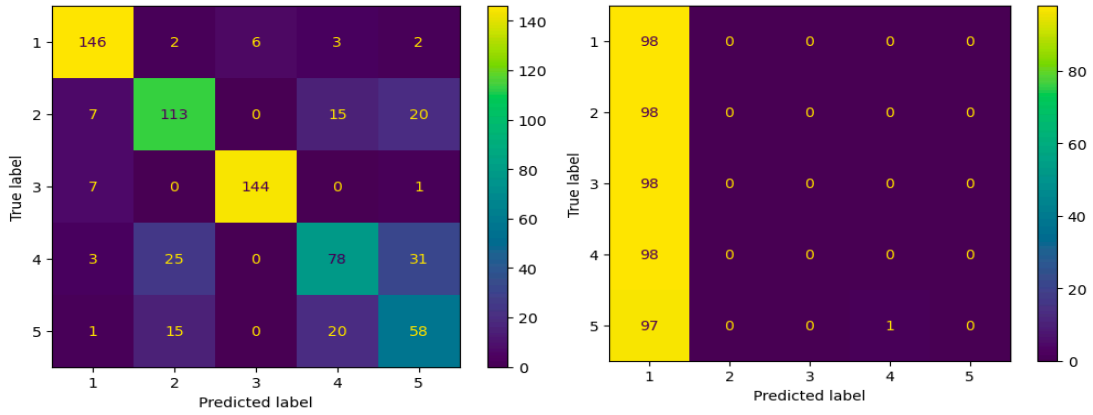

Figure 9 Confusion matrix for validation with 3 season (left); Confusion matrix for test with 3 seasons (right)

7. Discussion

Data pre-processing and Filtering: The semantic filtering process based on the auxiliary CGLS dataset was tested for different combinations. Next to the broadleaf/needle categories that were used in the end we tested a more general forest/no-forest split. The filtering steps described in 3.2. were conducted for both combinations equally. Both possible combinations were then tested on a 1-D Random Forest for comparison. The accuracy for the broadleaf/needle approach was 73.4%, while the forest/no-forest split only had 65.9%. Based on this accuracy difference we decided to continue working with the broadleaf/needle filtering approach.

Testing data accuracy problem: We filtered the training data set to increase the discriminability of different species, because of the coarse INSPIRE gridding of 1km x 1km. After filtering, the samples of each species have condensed spectral line characteristics. Three times four random but different point samples for Pinus Sylvestris are shown in Figure 10. It can be seen that our filtering results in a strong spectral line that performs well on a filtered validation set,

while the spectral line for the test data behaves similarly to the original data. Because of the different spectral information in the test data, we can't predict it with high accuracy.

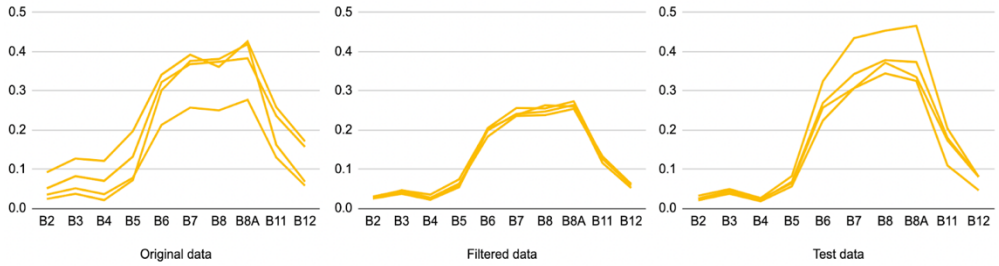

Figure 10 Spectral line of four samples in Pinus sylvestris from original, filtered and test data

The problem with discriminability of the test data set can also be seen in Fig 11. Here the spectral lines for the five different species can be seen for spring, summer and autumn. It can be seen that all five species are behaving in a similar fashion and no distinguishable feature can be seen. If compared to the spectral lines for our training data set after filtering in Figure 4, the clear difference in discriminability can be seen.

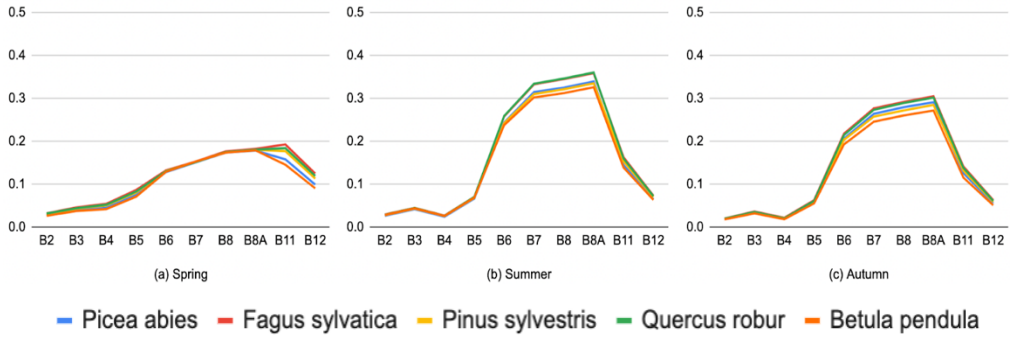

Figure 11 Average spectral line of five different tree species

8. Conclusion

In this project, the final deep learning model achieved more than 77% accuracy for 5 tree species classification in Germany on our training + validation data set. With the data preparation and augmentation, we got a scientific dataset with specific samples for each species. The Recurrent Residual Convolutional Neural Network (RRCNN) model can not only classify the needle and broadleaf forest but also can separate detailed tree species. With the early stopping callback and hyperparameter fine tuning we were able to get a stable and not overfitting model. The number of samples in the training dataset is influencing the accuracy of the network. Thus, if there were more samples in the training dataset after filtering, the accuracy of the network would have been better. Also, the test data and the training data have different spectral signatures (mainly because of the training data filtering) and this negatively affects the model accuracy when applied to the test dataset.

Link to script on GitHub:
https://github.com/SiruiWang0731/Tree-species-classification
Link to processed dataset:
https://drive.google.com/drive/folders/1kEuNuCG2-6UgkFQ9CP8JHVdu9bZrg-aD?usp=share_link

9. References

[1] Tree species introduction: https://www.tree-guide.com/
[2] Kuenzer, C.; Ottinger, M.; Wegmann, M.; Guo, H.; Wang, C.; Zhang, J.; Dech, S.; Wikelski, M. (2014). Earth observation satellite sensors for biodiversity monitoring: Potentials and bottlenecks.Int. J. *Remote Sensing, 35*, 6599–6647
[3] Immitzer, M., Atzberger, C., & Koukal, T. (2012). Tree species classification with Random forest using very high spatial resolution 8-band worldView-2 satellite data. *Remote Sensing*, *4*(9), 2661–2693. https://doi.org/10.3390/rs4092661
[4] Immitzer, M., Neuwirth, M., Böck, S., Brenner, H., Vuolo, F., & Atzberger, C. (2019). Optimal input features for tree species classification in Central Europe based on multi-temporal Sentinel-2 data. *Remote Sensing*, *11*(22). https://doi.org/10.3390/rs11222599
[5] Dalponte, M., Ørka, H. O., Gobakken, T., Gianelle, D., & Næsset, E. (2013). Tree species classification in boreal forests with hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, *51*(5), 2632–2645. https://doi.org/10.1109/TGRS.2012.2216272
[6] Mauri, A., Strona, G., & San-Miguel-Ayanz, J. (2017). EU-Forest, a high-resolution tree occurrence dataset for Europe. Scientific data, 4(1), 1-8.
[7] He, K., Zhang, X., Ren, S., & Sun, J. (2015). *Deep Residual Learning for Image Recognition*. http://image-net.org/challenges/LSVRC/2015/
[8] Hoffmann, E. J., Wang, Y., Werner, M., Kang, J., & Zhu, X. X. (2019). Model fusion for building type classification from aerial and street view images. *Remote Sensing*, *11*(11). https://doi.org/10.3390/rs11111259
[9] Corona, N., Mauricio Galeana-Pizaña, J., Manuel Núñez, J., Luis Silván Cárdenas, J., Corona Romero, N., Mauricio Galeana Pizaña, J., Manuel Nuñez Hernández, J., Manuel Madrigal Gómez, J., Silván Cárdenas, J. L., Corona Romero, N., & Galeana Pizaña, J. M. (2015). *Geospatial Technologies to Support Coniferous Forests Research and Conservation Efforts in Mexico.*
https://www.researchgate.net/publication/283503310
[10] Mou, L., Ghamisi, P., & Zhu, X. (2017). Deep recurrent neural networks for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, *55*(7), 3639–3655. https://doi.org/10.1109/TGRS.2016.2636241
[11] Sharma, N., Jain, V., Mishra, A. (2018). An Analysis Of Convolutional Neural Networks For Image Classification. *International Conference on Computational Intelligence and Data Science*, 132(1), 377-384.
[12] Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32.