ETH Zurich
Image Interpretation, HS 2021

# Lab Report 1

Classification

Valérie Hellmüller, Pascal Imhof, Christine Rösch

October $27^{th}$, 2021

Professorship ........ Prof. Dr. Konrad Schindler
Supervision ........ Torben Peters

## 1 Introduction

The goal of the first lab as part of the course Image Interpretation is the classification of palm oil trees in satellite images. In total 64 Sentinel-2 images from the regions Indonesia, Malaysia, and Philippines are used as training data. These input images have a size of 10980x10980x4 and include the channels RGB (red, green, blue) and NIR (near infrared). In the process of classification, every pixel of an image gets assigned to one of the following classes: background, palm oil trees, or clouds. Our group's (group 3) task is to test the classification process using two steps. The first step is to use a pretrained network to extract pixel wise features. In a second step, a classifier is trained on these extracted features.

## 2 Methodology

The process of image classification is divided into three main parts. Firstly there is the pretrained neural network, secondly the classification, and thirdly the verification of the classification. To extract features of the input images, a pretrained neural network was chosen. Torch vision offers different model architectures which are already pretrained. According to the website of Torch vision, the neuronal networks for image segmentation are trained on a dataset that contains the categories which are presented in the Pascal VOC dataset. This includes for example airplanes, birds, cars, cats, etc.. In total there are 21 different classes. (PyTorch, URL: https://pytorch.org/vision/stable/models.html, 26.10.2021)
Since the aim is to classify satellite images in this project, these categories might seem unsuitable. On the other hand, the readily available networks are not trained on satellite imagery. For a first and easy approach, it is reasonable to use e network which is not trained on satellite images. However, we only use the neuronal network to get more information for the classifier in the second step of the pipeline. By using a neuronal network it is

possible to collect information from the surroundings of each pixel. For this project, the model FCN ResNet50 is used. ResNet50, which stands for Residual Network 50, is a convolutional neural network that is 50 layers deep. This model is widely used and easy to apply thanks to Torch vision. To process the input images with these models, they have to be normalized in a certain way. According to Torch Vision, all pretrained networks need the input images with the same normalization and standard deviation. Therefore these calculations are done in a first step. The approach of our group is not to use the entire available amount of data, because it should be kept as simple as possible. The implementation is done in a way that it is possible to run it on a computer with 32GB memory. To achieve this, patches with the size of 256 X 256 pixels are cut out with a regular grid. This patch size corresponds to the patch size with which the model ResNet50 is trained. As mentioned above 4 different channels (R,G,B,NIR) of input data are available. Since ResNet50 accepts only 3 input layers, the blue channel of RGB is replaced with the NIR channel. This choice seems to be reasonable because we want to detect plants and clouds. The green of the plants is well detectable in NIR, and for the clouds it does mot matter which color channel gets used because white includes all colors. The NIR channel has to be normalized due to the fact that ResNet50 expects the inputs to be images of the type Image. In the next step, the chosen pretrained network is applied to the dataset. The output of the neuronal network is a vector with a length 21 for each pixel. This vector describes the probability of this pixel belonging to one of the 21 classes. In the implemented approach only the highest float number is taken as a result of the neuronal network. To use the result of the neuronal network in the second step of the pipeline these numbers are stored in NumPy array files. As classifier, the Decision tree is selected. It is a rather simple and widely used algorithm and therefore serves nicely for this approach. It has to be mentioned that it is not possible to train the Decision tree iterative. So the amount of data that can be used to train the classifier is limited by the size of the memory of the computer. To train the classifier, the ground truth has to be generated, so that the image classification can be evaluated. This is done according to the example which is given by the supervision of the exercise. The training data consists of the following layers: R,G,B, NIR, and the output vector of ResNet50. The individual layers can be combined with each other as desired. In this approach, 70% of the available data is assigned to training data and 30% to validation data. This separation is done with the function *train_test_split* of scikit-learn.

To evaluate the performance of the classification algorithm, a confusion matrix is generated. The values of the confusion matrix enable to calculate the precision, the recall, and the f-score of the classification. The precision gives information about how often a prediction of a certain class is actually correct. The recall or true positive rate describes how often an actual class is predicted correctly. The f-score is the harmonic mean of the precision and the recall and serves as an additional accuracy measure. (Data School, URL: https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/, 26.10.2021)

# 3 Results and Discussion

In table 1 an overview of the numeric results is shown. In total eight experiments are presented in this report. To generate a baseline for the other experiments, only the RGB and the NIR channels are used to train the Decision tree. The output from the neuronal network is not used. The only hyperparameter which is not left at the default value is max depth. This parameter is randomly set to 5. The gridsize is set to 8x8. The result shows that no palm oil trees were detected there.

The first two experiments are about the amount of input data that is given to the classifier. The aim of these experiments is to get an idea of how much input data is needed to get a reasonable result. The following grid sizes are used: 4x4, 8x8. It has to be mentioned, that it is not possible to run the training dataset with a grid of 16x16 due to the fact, that the amount of data is too big for the memory of the used computer. So the 16x16

| | Overall accuracy testing | Precision background | Recall background | F1 background | Precision palm oil | Recall palm oil | F1 palm oil | Precision clouds | Recall clouds | F1 clouds |
|---|---|---|---|---|---|---|---|---|---|---|
| Basline 8x8 (Test data) | 0.667 | 0.774 | 0.844 | 0.807 | NaN | 0 | NaN | 0.156 | 0.224 | 0.184 |
| Experiment 1 | 0.852 | 0.837 | 0.989 | 0.846 | 0.511 | 0.003 | 0.005 | 0.940 | 0.933 | 0.937 |
| Experiment 2 | 0.835 | 0.821 | 0.978 | 0.893 | 0.579 | 0.061 | 0.111 | 0.936 | 0.934 | 0.935 |
| Experiment 3.1 depth=10 | 0.846 | 0.843 | 0.961 | 0.898 | 0.616 | 0.199 | 0.300 | 0.934 | 0.953 | 0.944 |
| Experiment 3.2 depth=20 | 0.861 | 0.858 | 0.962 | 0.907 | 0.681 | 0.287 | 0.404 | 0.952 | 0.960 | 0.956 |
| Experiment 3.3 depth=30 | 0.917 | 0.909 | 0.979 | 0.943 | 0.868 | 0.553 | 0.675 | 0.984 | 0.986 | 0.985 |
| Experiment 3.4 depth=40 | 0.972 | 0.966 | 0.995 | 0.981 | 0.977 | 0.839 | 0.903 | 0.996 | 0.998 | 0.997 |
| Experiment 3.5 depth=None | 0.999 | 0.999 | 1.00 | 0.999 | 1.000 | 0.993 | 0.996 | 1.000 | 1.000 | 1.000 |
| Experiment 4 validation | 0.835 | 0.821 | 0.978 | 0.893 | 0.579 | 0.062 | 0.111 | 0.936 | 0.934 | 0.935 |
| Experiment 4 test | 0.802 | 0.786 | 0.959 | 0.864 | 0.451 | 0.048 | 0.087 | 0.936 | 0.934 | 0.935 |
| Experiment 5 validation | 0.835 | 0.821 | 0.978 | 0.893 | 0.579 | 0.062 | 0.111 | 0.936 | 0.934 | 0.935 |
| Experiment 5 test | 0.800 | 0.786 | 0.959 | 0.864 | 0.460 | 0.051 | 0.093 | 0.936 | 0.934 | 0.935 |
| Experiment 6 validation | 0.999 | 0.999 | 1.000 | 0.999 | 1.000 | 0.993 | 0.996 | 1.000 | 1.000 | 1.000 |
| Experiment 6 test | 0.742 | 0.813 | 0.795 | 0.804 | 0.349 | 0.361 | 0.355 | 0.999 | 0.999 | 0.999 |
| Experiment 7 validation | 0.846 | 0.843 | 0.961 | 0.898 | 0.616 | 0.199 | 0.300 | 0.934 | 0.953 | 0.944 |
| Experiment 7 test | 0.810 | 0.809 | 0.932 | 0.867 | 0.556 | 0.194 | 0.287 | 0.934 | 0.953 | 0.944 |
| Experiment 8 validation | 0.999 | 0.999 | 1.000 | 0.999 | 1.000 | 0.994 | 0.997 | 1.000 | 1.000 | 1.000 |
| Experiment 8 test | 0.771 | 0.837 | 0.824 | 0.830 | 0.420 | 0.420 | 0.420 | 1.000 | 1.000 | 1.000 |

Table 1: Results of different experiments

grid is only used for the test dataset (see experiment 4). To reduce the computation time, the maximal depth of the decision tree equals 5, as for the baseline. In the first experiment, the images are divided in 4x4 patches and the ResNet50 model is applied before the decision tree, the results can be found in the table 1.
In the second experiment the images are devided in 8x8 patches and the classifier is fed with all the information of the input layers. Again the max depth of the Decision tree is set to 5.
By comparing experiments 1, 2, and the baseline, it is possible to state that with more images the precision, recall, and f1 value for palm oil increase. One possible reason is, that with the gridsize of 4x4 there are not enough pixels with the label palm oil in the training data. There is no further investigation done about that. But it is possible to state, that in comparison to the baseline all values increased. This indicates, that the information from the neural network is helpful for the classification task.
Another approach is experiment three, in which the maximal depth of the decision tree is changed to 10, 20, 30, 40, and None. When the depth of the tree is set to None, it means all leaves are pure, implying the leaf will only have labels if you choose default for the min leaf which is per default one. The result in the table shows that the deeper the tree, the better the result. However, when these decision trees are used with the test data, only an overall accuracy of about 70 % can be achieved. This strongly suggests overfitting if the tree is too deep.
The goal of experiments four, five, and six is to evaluate, if girdsize of the testing data has an influence on the final result. In experiment four, the grid size is set to 16x16, in experiments five and six the gridsize is set to 32x32. For all three experiments the gridsize of the training data is set to 8x8. In experiments four and five, the max depth of the decision tree is set to 5. For experiment 6 the default value for the max depth *None* is used. The table shows the results for the validation and the test dataset. It seems that there is no big difference between the results of experiments four and five. Due to the fact that more data cannot be worse, a gridsize of 32 x 32 is used for experiment seven in the test set. In experiment six, the same problem arises as in experiment three.
In experiment seven, the max depth of the tree is set equals to 10 and the gridsize of the training data to 8x8. As mentioned above the girdsize of the test data is set to 32x32 and the patches stayed like in five and six on 32x32. The results of this approach are looking reasonable.
For the last experiment done in the context of this assignment, the training data was again divided in 8x8, but the hyperparameter splitter is set equal to *random*. The max depth of the decision tree is again set on *None*. The idea behind this experiment is to reduce the overfitting by splitting the data randomly. In the table above it is visible that this experiment produces not the desired result.
In summary, it can be said that experiment seven has the best result.

## 4 Summary and Outlook

The different experiments showed, that it is possible to use a neuronal network in combination with a normal classifier to classify satellite images. It also showed, that not the complete amount of data is needed to achieve usable results. For future work, it would be interesting to enlarge the amount of data that is used to train the classifier. Additionally, the hyperparameter tuning has so far been done at a very basic level. In this approach, only one classifier is used. With more time, other classifiers could be tested and the hyperparameter tuning could be done on a higher level. Moreover, it would be possible to get more information from the neuronal network by not only using the highest probability.