ETH Zurich
Image Interpretation, HS 2021

# Lab Report 2

Regression

Marie-Josianne Fandré, Valérie Hellmüller, Pascal Imhof, Yatao Zhang

Novembre $17^{th}$, 2021

Professorship ..... Prof. Dr. Konrad Schindler
Supervision ..... Mikhail Usvyatsov

# 1 Introduction

For the second Lab within the course Image Interpretation the goal is to build a neuronal network regression model to predict a height value for each pixel. The given data set contains Sentinel-2 images with RGB and NIR channels and a ground truth for vegetation height above the ground. The task of our group is to implement and train a neuronal network to predict the canopy height for each pixel.

# 2 Methodology

## 2.1 Pre-Prossessing

The received data consists of satellite images which are showing six different regions in Switzerland. There is a small overlap between some images. This overlap is neglected in this work. The satellite images are taken from different overflights. Depending on the region, 20 or 30 images are available from different overflights. The main task of the pre-processing is to compute a complete images from the different overflights as depending on the images a large part can be missing. Firstly, a quality check of the input images is done. For that a histogram of each input image is computed and the RGB image, NIR image and cloud coverage are analysed. When the distribution of the RGB values matched the plotted image, the image was taken as input data, this was done manually. Afterwards, the images were added up and then the mean was taken using the number of pixels with information. However, it must be taken into account that clouds cover partially the image. To filter out the clouds there is a cloud mask available for each image. At the end of the pre-processing there are 6 images left, of which four are used for the training and 2 for the testing of the network.

After further analysis of the results, several problems with pixel values were found. This is due to some errors in the input data, which were found to late for this project. In detail there was a high cloud coverage indicated on pixels, which had no information in them. This resulted in a faulty mean for a small amount of pixels.

## 2.2 Neural Network

Within this lab two different neuronal networks are implemented and tested. Both are already presented in lab 1. These are the neural networks UNET and FCN-ResNet50.

### 2.2.1 Data Loader

The data loader of lab 1 is used in this lab with some adaptations. The first one is the data normalization. Analyses of the input data have shown that the input data is distributed over a wide range, but most of the RGB values are below 3000. For the NIR input data the same pattern can be observed. Due to this distribution the data input RGB data is clipped to a range of 0-3000 and the NIR data to 0-8000. Clipping means that all values greater than the limit value (RGB: 3000, NIR: 8000) are set to this limit value. By dividing through the limit value it is possible to ensure that all values are between 0 and 1. This is an important step. Otherwise the neuronal networks are not able to learn something.

The second adaptation of the data loader is the treatment of the NaN values in the input data. The final solution sets all NaN values in the input data to 0 (RGB and NIR). The labels of these pixels are set to -1. This is done due to the fact that the pixel with label -1 are not use to compute the loss function of the neuronal

network.

The epoch number was 200 during the training (75% of the data set) and validation (25% of the data set) process. The epoch with the least loss was regarded as the best model, and its model weights would be recorded. Also, the loss function was set as the MSE (mean-square-error) loss to calculate the error between the ground truth and the prediction output. According to this setting, the canopy height was the 1-D output feature in a 2-D image.

#### 2.2.2 UNET

The network used in this lab is already presented by Group 2 in Lab 1. In detail, UNET used a symmetric and complete encoder-decoder architecture to achieve pixel-wise segmentation, including a contracting path and a similar expanding path [1]. In this lab, we adjusted it to make sure it can be applied in the regression task.

1. It is adapted in a way that the output is a regression layer instead of a classification layer.
2. As optimizer the stochastic gradient descent is used, the learning rate is set to 0.000001 and the momentum to 0.9.

#### 2.2.3 FCN-ResNet50

The FCN-ResNet50 refers to a deep residual network with a fully convolutional network, which is initially designed for pixel-wise image segmentation [2]. The main architecture remains the same as the FCN-ResNet50, including the backbone part, the classifier part and the aux_classifier part. In this lab, we revised it to adapt the task of pixel wise regression, and the details are listed as follows.

1. Different from the original version of FCN-ResNet50 that possesses three input channels, we changed the inputs into four channels, i.e. R, G, B and NIR. To more efficiently train the network, we copied the weight of the R channel to the NIR channel before training. In the transform part, we also set the same weight for the R and NIR channel.
2. To adapt the network to the regression work, we adjusted the output layer of FCN-ResNet50 as the 1-D output. In addition, we selected stochastic gradient descent as the optimizer with the learning rate of 0.001 and momentum of 0.9.
3. In the inference process, the best model recorded was directly used to predict the canopy height. Also, we used RMSE (root-mean-square-error) and MAE (mean-absolute-error) to evaluate the performance of the proposed network.

## 3 Results and Discussion

In this lab, we trained the UNET and FCN-ResNet50 in a Linux system with 340GB RAM, 96 threads and 1 TESLA V100 16GB GPU. The inference was also executed in the same environment based on the recorded model with the least validation error.

### 3.1 Running Time and Memory

In the table 1, we showed the information about training and inference time when using the UNET and FCN-ResNet50 to predict the canopy height. Compared with the FCN-ResNe50, the number of weights and parameters that need to be updated in the UNET is less. Thus, it's obvious to find that the UNET spends less time on training the model and inferring new samples than the FCN-ResNet50. Similarly, the accounted memory of UNET is also less, about 3.10 GB in the maximal usage of memory.

Table 1: Training and inference time for UNET and FCN-ResNet50

| | Training time | Inference time | Memory (before) | Memory (max) | Memory (after) |
|---|---|---|---|---|---|
| UNET | 24737.11s | 29.87s | 3.03GB | 3.10GB | 2.41GB |
| FCN-ResNet50 | 27689.97s | 54.40s | 3.05GB | 3.21GB | 2.59GB |

### 3.2 Training and Validation Loss

The Figure 1 presents the training and validation MSE loss curve when updating weights of the UNET and FCN-ResNet50. According to their curves, there are some characteristics of training these two models.

1. Convergence trend. The convergence trend of these two network both decline along the increase of epoch numbers. In detail, for the UNET, the change variations of the training and validation loss are quite close in terms of tendency and quantity. But for the FCN-ResNet50, its trend is a little different between the training and validation loss. In addition, the convergence trend of the UNET is quite smooth and stable, but there are lots of fluctuations for the FCN-ResNet50.

2. Convergence speed and quantity. The UNET has a faster convergence speed and almost keeps stable in the epoch number of 70. But the FCN-ResNet50 achieves a relatively convergence state when the epoch number is above 180. But at the convergence stage, the MSE loss value of FCN-ResNet50 is smaller than the UNET.

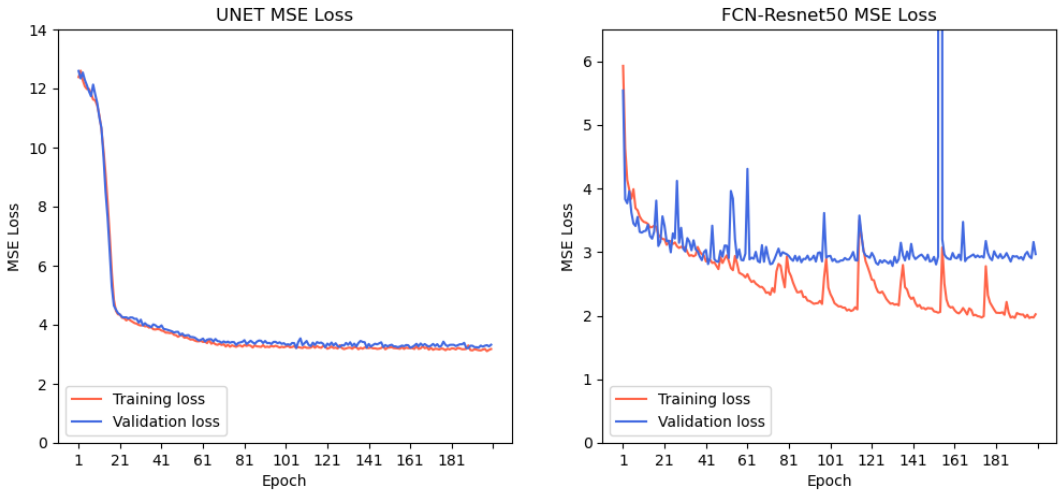


Figure 1: Training and validation MSE loss for UNET and FCN-ResNet50

### 3.3 Evaluation Metrics and Visualization

The evaluation metrics for the test data set are in Table 2. RMSE and MAE provides a quantifying measurement of residual between ground truth and prediction outputs. According to Table 2, the FCN-ResNet holds a better predictive ability to evaluate the canopy tree with a RMSE of 5.13m and a MAE of 2.91m. The UNET also holds a similar result with a RMSE of 5.21m and a MAE of 2.98m.

However, when we visualized the prediction result of these two models in Figure 2, we can find the output of the UNET has a better interpretation comparing to the similarity of ground truth. The output of FCN-ResNet50 is slightly fuzzy on the whole, which is not a good choice when evaluating the landscape of the research area.

Table 2: Evaluation metrics of UNET and FCN-ResNet50 (m)

| Model | RMSE | MAE | Residual ave | Residual max | Residual min |
|---|---|---|---|---|---|
| UNET | 5.21 | 2.98 | 0.42 | 49.75 | -59.94 |
| FCN-ResNet50 | 5.13 | 2.91 | -0.07 | 42.09 | -62.1 |

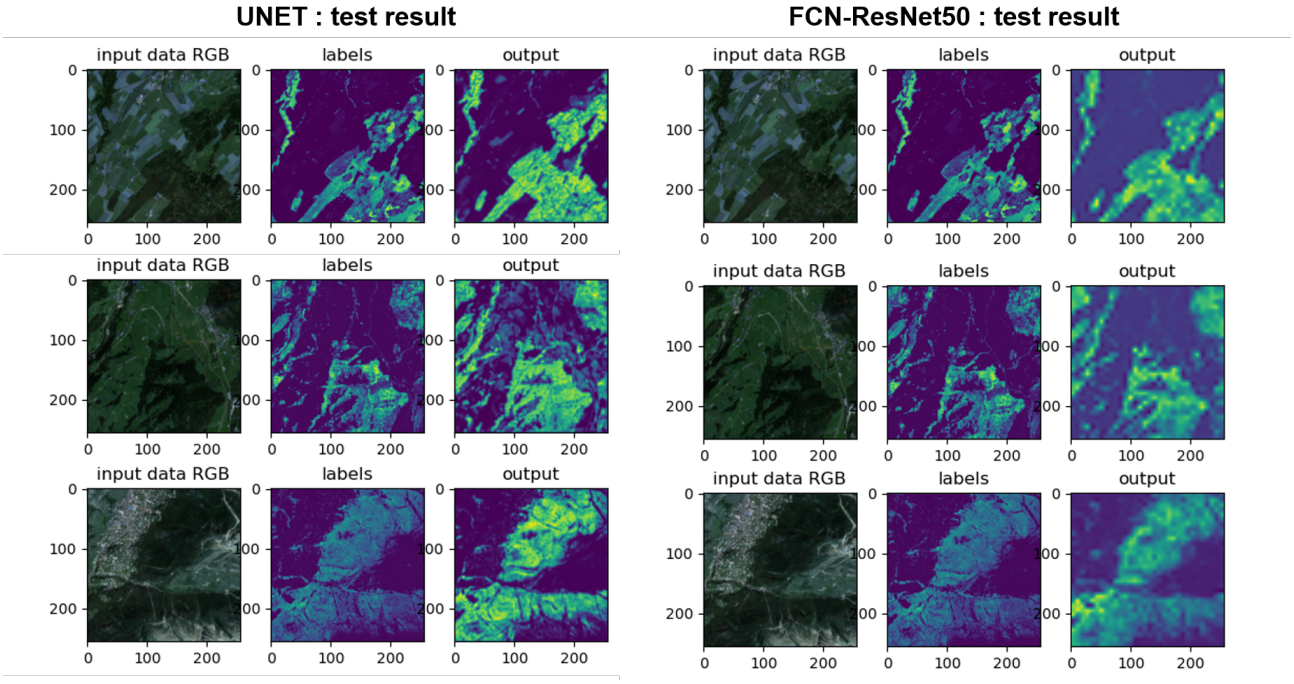


Figure 2: The comparison of prediction result between UNET and FCN-ResNet50

## 4 Conclusion and Outlook

In this lab, we presented two models (UNET and FCN-ResNet50) to implement the regression task and predict the canopy height in Switzerland. In the whole, the FCN-ResNet50 performs better in the evaluation metrics but failed to present the overall landscape of research areas, while the UNET has a better interpretation in depicting the landscape of research areas. To improve the performance of our models, it will be suggested to revise the architecture of these two models to better suit the regression task.

Furthermore, it was seen that the cloud coverage indicated a high coverage while the RGB and NIR image had no information. Further investigation has to take place, so that this problem will not occur in the next lab.

# References


[1] Binge Cui, Xin Chen, and Yan Lu. "Semantic segmentation of remote sensing images using transfer learning and deep convolutional neural network with dense connection". In: *Ieee Access* 8 (2020), pp. 116744–116755.

[2] Jonathan Long, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 3431–3440.