

# **Correcting global elevation models for canopy and infrastructure using machine learning**

Haoyang Dong  
student #5302501

1st supervisor: Maarten Pronk  
2nd supervisor: Hugo Ledoux

July 1, 2022

# 1 Introduction

A digital elevation model (DEM) is a digital representation of the surface morphology and contains a wealth of topographic and geomorphological information necessary for analyzing geological applications. It has become one of the most important and basic geographic information data, and there is an increasing abundance of publicly released local and global DEM datasets. (Polidori and El Hage, 2020)

Accurate elevation data in DEMs are essential for many geoscience applications and other fields, such as soil erosion, reservoir planning, and flooding prediction. Notice that DEM is a generic term that concludes two surfaces model, the digital terrain model (DTM) and the digital surface model (DSM). Since it is widespread to use photogrammetry and LiDAR to produce DEMs, at the global scale, most DEMs are more like DSMs rather than DTMs. (Polidori and El Hage, 2020; Hawker et al., 2022)

Directly using DSMs in some applications will cause errors; for example, the threats might be underestimated in flooding prediction when using DSMs in urban or forested areas. (Kulp and Strauss, 2016) Therefore, post-processing strategies must be used to remove height bias from trees and buildings so that the DEM can theoretically be considered a DTM.

Although this issue has been researched in some studies like Hawker et al. (2022); Kulp and Strauss (2018), which convert global DSMs to DTMs, there are some drawbacks in their methods: 1) Besides DSM pixels, they need extra inputs from other sources like population density and vegetation density; 2) There are artifacts created in the results. Some deep learning methods that are successful in computer vision have been used in DTM extracting recently; however, most studies (Meadows and Wilson, 2021; Kazimi et al., 2020; Gevaert et al., 2018) rely on multiple bands of imagery or point clouds as inputs.

This project uses a fully convolutional network (FCN) for semantic segmentation and aims to:

1. Use only DSM as input;
2. Apply this model on DSMs with higher resolution.

The project will benefit applications that need the correct terrain representation and further research on correcting DEMs with machine learning.

## 2 Related work

### 2.1 DEM, DSM, and DTM

A digital elevation model (DEM) is "a regular gridded matrix representation of the continuous variation of relief over space" (Pratibha et al.). DEMs are widely used in geoscience. The methods for elevation mapping have developed rapidly over the last century. Nowadays, aerial photography and LiDAR are major methods of producing DEMs.

Based on the different definitions of the nominal surfaces, which are the physical surfaces to be modeled, two models, i.e., the digital terrain model and the digital surface model (Figure 1), are often considered. The nominal surface used in a DTM is the ground surface, i.e., terrain, while the upper surface above the trees, buildings, and other natural or artificial objects is used in a DSM. DSMs are provided by most DEM production techniques such as photogrammetry and short-wavelength radar technologies. (Polidori and El Hage, 2020)

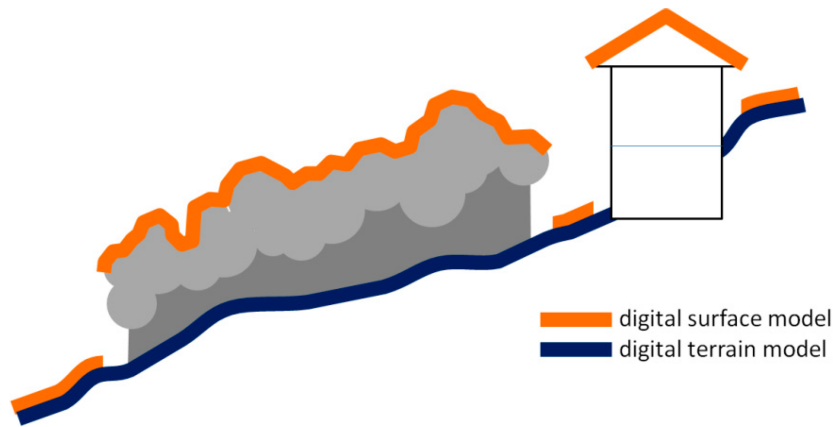


Figure 1: DTM vs DSM in the presence of trees and buildings. (Polidori and El Hage, 2020)

## 2.2 Regression analysis

The use of regression analysis for correcting the elevation errors is based on a small number of variables. The approaches use vegetation cover indices and apply mostly to removing the bias of canopies and are not capable for buildings. (Baugh et al., 2013; Su et al., 2015; Kulp and Strauss, 2018)

Incorporating additional variables that correlate with elevation error into the model could further improve correction results. However, as the number of variables increases, the utility of traditional parametric regression techniques will be limited due to the curse of dimensionality (Köppen, 2000) and the highly nonlinear relationship between variables; for example, Kulp and Strauss (2018) uses population density, vegetation density, and slope information as inputs. The term curse of dimensionality used there refers to the intractability of systematically searching through a high-dimensional space, the apparent intractability of accurately approximating a general high-dimensional function, and the intractability of integrating a high-dimensional function. (Donoho and others, 2000)

As an effective empirical method for regression and classification of nonlinear systems (Lary et al., 2016), deep learning may avoid the curse of dimensionality. (Poggio et al., 2017)

## 2.3 ANN: Artificial neural networks

Artificial neural networks are the most common models to develop nonparametric and nonlinear classification/regression (Rodriguez-Galiano et al., 2015). There are many different types of ANN. One of the most used ANNs: the feed-forward propagation neural network (Rumelhart et al., 1986) is described briefly in this section.

As in the brain, the basic processing elements of an artificial neural network are neurons (units or nodes). In a neural network, the units are connected in layers, and information flows in one direction, from the input units - through the units in the hidden layer/layers - to the units in the output layer. The input unit assigns the signal to the hidden unit in the second layer. A neuron performs essentially a linear regression followed by a nonlinear function. Neurons in different layers are interconnected with corresponding links (weights). The output signal is generated by algebraically summing all the weighted inputs. The structure is shown in Figure 2 The goal of the algorithm is to find a set of weights to ensure that for each input vector, the resulting network vector is equal to or close enough to the desired output vector. Suppose there is a defined and finite set of input-output cases (patterns). In that case, the overall error of the network operation with a given set of weights can be calculated by comparing the actual output vector of each pattern with the desired output vector, for example, by

least squares. To train an ANN, it is necessary to choose a structure (hidden layers and number of nodes per layer), appropriate initialization of weights, learning rate, and regularization parameters to avoid overfitting. (Rodriguez-Galiano et al., 2015)

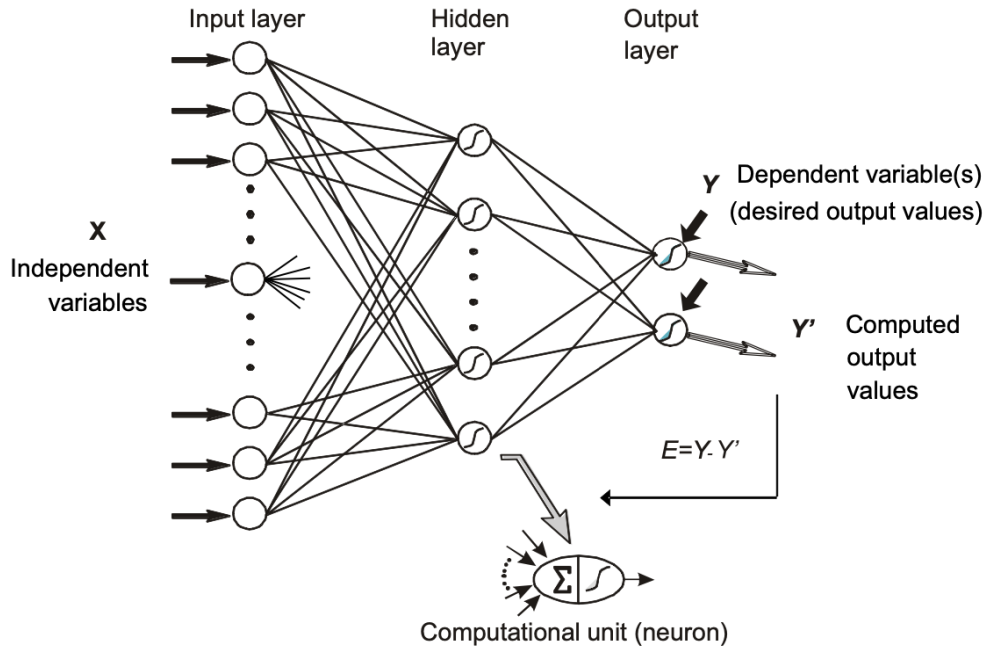


Figure 2: The structure of a multilayer perceptrons. (Park and Lek, 2016)

## 2.4 RF: Random forest

RF is a regression technique that combines the performance of numerous DT (Decision Tree) algorithms.

A DT (Figure 3) represents a set of hierarchically organized restrictions or conditions that are successively applied from a root to a terminal node or leaf of the tree. To induce the DT, recursive partitioning and multiple regressions are carried out from the dataset. From the root node, the data splitting process in each internal node of a rule of the tree is repeated until a specified stop condition is reached. Each of the terminal nodes, or leaves, has attached to it a simple regression model which applies in that node only.

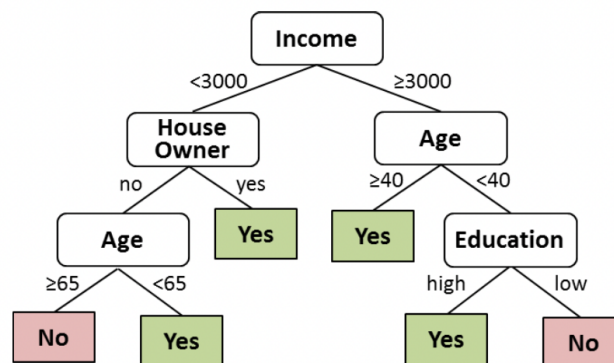


Figure 3: Architecture of a Decision Tree.



When RF receives an (x) input vector made up of the values of the different evidential features analyzed for a given training area, RF builds a number K of decision trees and averages the results (Figure 4).

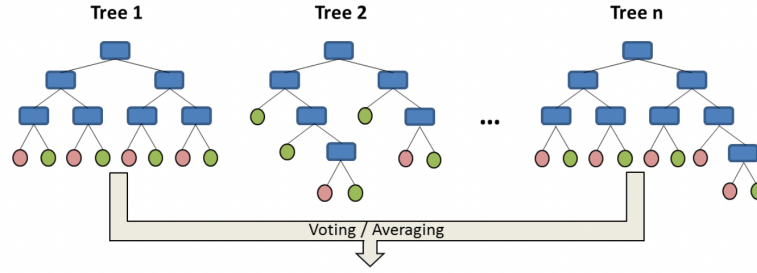


Figure 4: Random Forest as Tree Ensemble.

To avoid the correlation between the different trees, RF increases the diversity of the trees by making them grow from different training data subsets created through a procedure called bagging. Bagging is a technique used for training data creation by resampling the original dataset randomly with replacement, i.e., with no deletion of the data selected from the input sample for generating the next subset.

It is worth mentioning that this conclusion can only be applied to the best classification methods obtained from a complex optimization process since, in general terms, the performance of RF for all the parameter combinations was better than that of the rest in terms of stability and accuracy. (Rodriguez-Galiano et al., 2015)

Recently, Meadows and Wilson (2021) introduced the usage of fully convolutional networks to generate DTMs from multiple layers of remote-sensed input data and found that the FCN outperforms the other models.

## 2.5 FCN: Fully Convolutional Networks

Before we discuss the concept of FCN, the knowledge of convolutional neural networks is essential. CNN (Figure 5) is a class of deep learning techniques, most commonly applied to analyzing visual imagery. The structure of a CNN contains one input layer, one output layer, and multiple hidden layers. The hidden layers include convolutional, pooling, and fully connected layers. Convolutional layers extract critical information from images to a feature map; pooling layers reduce data dimension; fully connected layers connect every neuron in one layer to every neuron in another layer and output classifications.

The critical element of a convolutional layer is the convolution kernels, which are several matrices containing specific target patterns within the input image. After multiplying the original image by the convolution kernels and pooling, we can get three outputs indicating information in different parts of the original picture, respectively. By doing so, CNN targets the critical features within the input picture and makes decisions according to these extracted features. CNN has been applied to image classification in many areas and usually has a much higher classification accuracy than humans and other algorithms.

FCN (Figure 6) is an end-to-end, pixel-to-pixel network mainly used for image semantic segmentation (Long et al., 2015). The FCN has been applied in multiple domains due to its outstanding accuracy in image segmentation.

Unlike CNN, FCN replaces the fully connected layers with convolutional layers. Convolutional layers extract features from the original image and compress the information in multiple convolution outputs. Several rounds of convolution and pooling create a heatmap with abstracted features. The last step in FCN is to unsample the heatmap through deconvolution

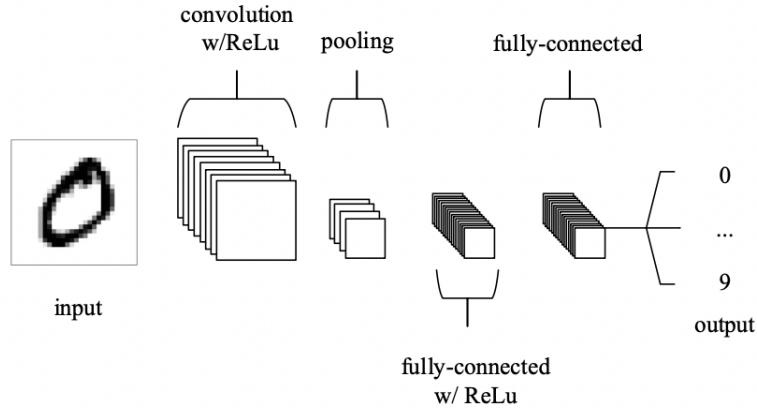


Figure 5: An simple CNN architecture, comprised of just five layers by O’Shea and Nash (2015)

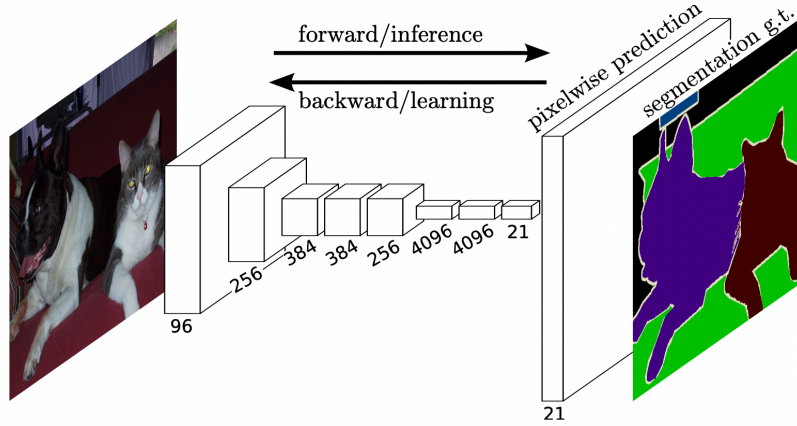


Figure 6: Fully convolutional networks can efficiently learn to make dense predictions for per-pixel tasks like semantic segmentation. (Long et al., 2015)

layers. Deconvolution is a reverse process of convolution that decompresses the information and fills the image with additional pixels. After convolution and deconvolution, the item’s border within the original image will be presented clearly in the output image.

### 3 Datasets and tools

#### 3.1 Dataset

##### 3.1.1 AHN4

The Actueel Hoogtebestand Nederland (AHN) is the digital height map for the whole of the Netherlands. The AHN4 is the latest dataset collected in the years 2020 and 2021.

The points in AHN4 have been classified by automatic classification and manual correction into several classes, such as ground level, buildings, water, artworks, and others (AHN, 2020). The DSM and the DTM of AHN4 are delivered in 2 different resolutions (0.5 m and 5 m) taking the use of the classified point clouds. The DSM and the DTM will be resampled to 30m and the process will be discussed more in the Methodology section.

## **3.2 Tools**

### **3.2.1 QGIS**

A free and open-source cross-platform desktop geographic information system (GIS) application.

### **3.2.2 Python and Pytorch**

Pytorch is an open-source machine learning framework based on Python.

## **4 Research questions**

### **4.1 Main question**

How to produce a reliable DTM from a DSM by removing the pixels of trees and buildings by semantic segmentation using FCN?

### **4.2 Sub-questions**

1. How to evaluate the results of the model trained with local DEMs in the project with the results of other models trained with global DEMs?
2. How to tune the parameters in the FCN model? How to adjust the window size for better performance?
3. Will the method be capable for the DEMs with higher resolution? How to apply the changes?
4. Will the use of morphological filter or other techniques benefits the model?

## **5 Methodology**

In this project, one FCN model will be built. The model takes the samples in the AHN4 DSM as inputs. After trained, the pixels in the DSM marked as trees or buildings by the models will be removed and then interpolated with surroundings. The differences between the produced DTM and the original one are the errors of this method. Similar procedures can be repeated on other published models to compare the performance differences.

### **5.1 From regression task to semantic segmentation task**

Correcting the elevation errors caused by canopy and infrastructure from DSMs to produce DTMs is a typical regression task aiming to map the input values with the continuous output variables. However, as mentioned above, it needs additional input sources and will create artifacts.

This project uses a method that first associates a label (canopy, infrastructure, or ground) with each pixel, called semantic segmentation (in computer science); then removes the pixels marked as canopies or infrastructures; finally, interpolates these pixels with their surrounding ground pixels.

By doing so, this method converts the regression task to a semantic segmentation task.

## 5.2 Data preprocessing

To clarify the description, an area near Wassenaar is used as an example (Figure 7)

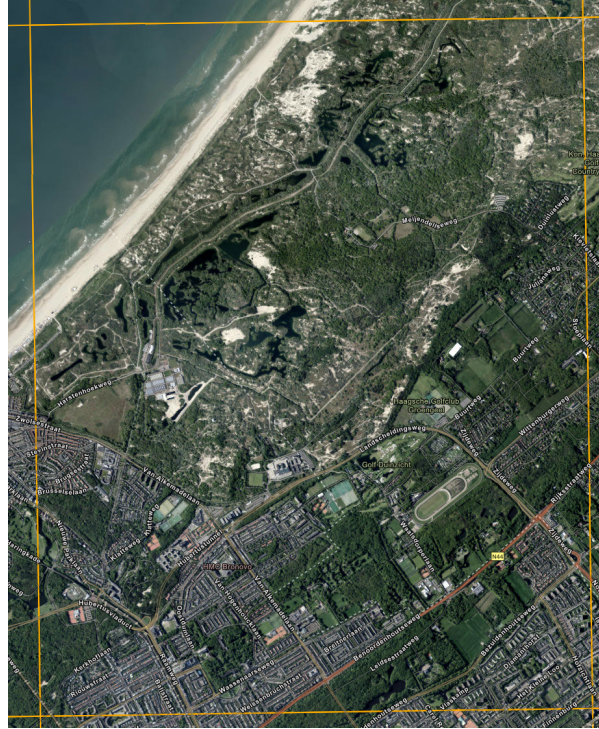


Figure 7: Patch 30GN1 of AHN4 on ArcGIS (Esri, 2021) map view.

### 5.2.1 No-data filling

There are no-value pixels in DSMs due to building blocking and other errors when processing LiDAR data. Therefore, a no-data filling method is needed for preprocessing to avoid errors. Figure 8 shows the results made in QGIS.

### 5.2.2 Resample AHN4 to 30m

In order to evaluate the method by comparing the results with FABDEM and CoastalDEM, the DTM and DSM of AHN4 will be first resampled to the resolution of 30m.

Although this can be done in QGIS (Figure 9), the standard method will remove the no-data cells in DTM, which are essential when labeling and making ground truth maps; new methods are needed for this process. A possible solution is to keep the resampled cell as a no-data cell by voting, i.e., by counting how many cells are no-data cells before downsampling.

### 5.2.3 Segmentation and step

The samples are segments of the DSM after resampling and no-data filling with a window size of 9 by 9. In order to avoid ignoring the pixels at the edges of each segment, a minor step is used. For a 9 by 9 window, the step for sliding the window is 7 (Figure 10). The labels of the pixels at the overlapping boundaries should also be determined by voting.



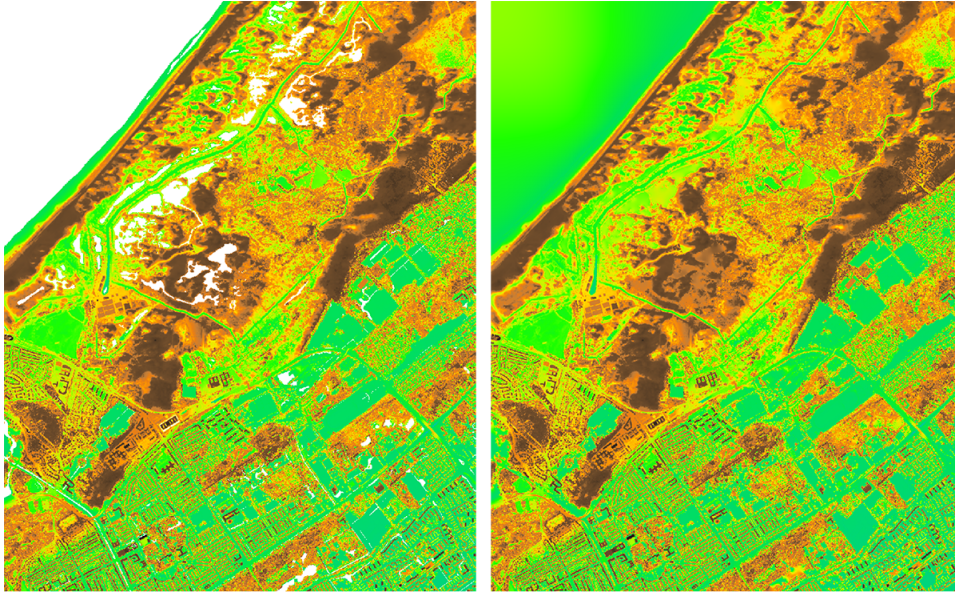


Figure 8: The DSM of selected area (left) and the result after filling (reight).

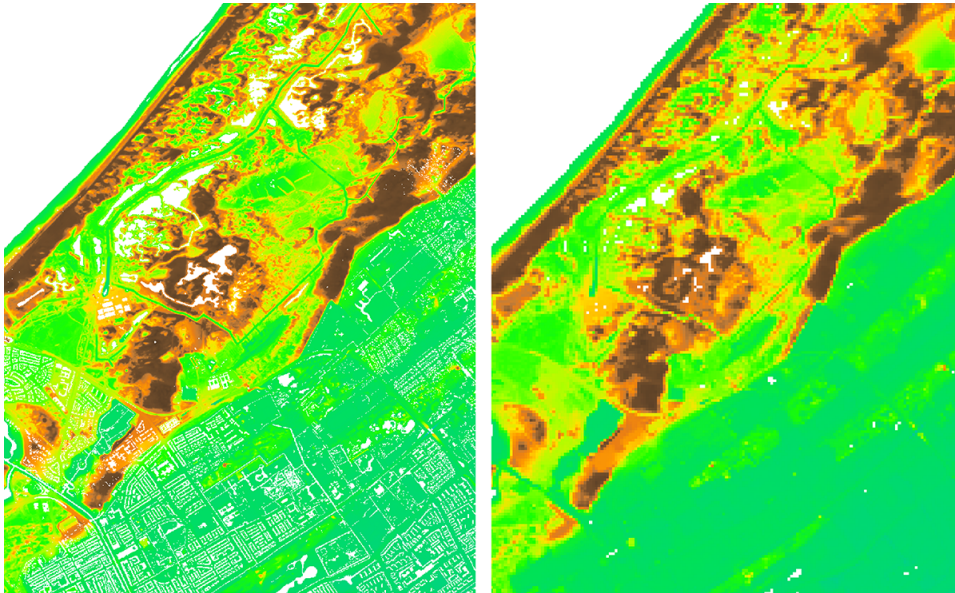


Figure 9: The resample method in QGIS will remove the no-data cells in DTM (left).

### 5.3 Create ground truth

Since canopies or infrastructures performs differently on DEMs, this project designs three labels for canopy, infrastructure, and ground pixels. Notice that only the pixels with label 0 will remain.

#### 5.3.1 0-1-2 label

As we attempt to remove the pixels of the canopy and the infrastructure, we can use label 1 to respect the pixels of infrastructures, 2 for canopies, and 0 for the ones to remain. The labeling (Figure 11) is processed by comparing the values of selected pixels in the DSM between the values of the corresponding pixels in the DTM.

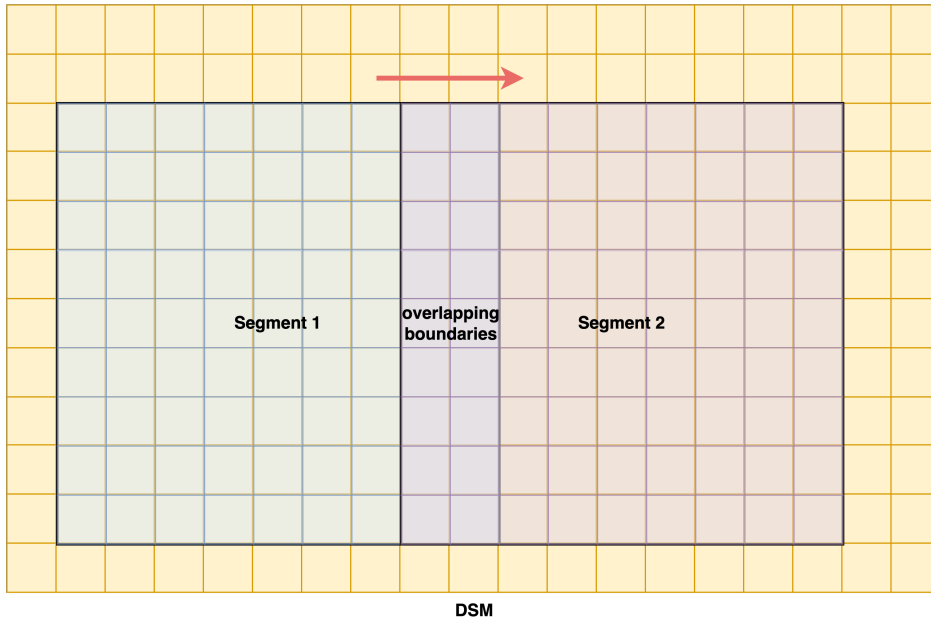


Figure 10: An illustration showing the window of segments and a step.

1. If there are no values in the corresponding pixels, the label of the selected pixels is 1;
2. or the difference between values is larger than a threshold, then the label of the selected pixels is 2;
3. otherwise, it is 0.

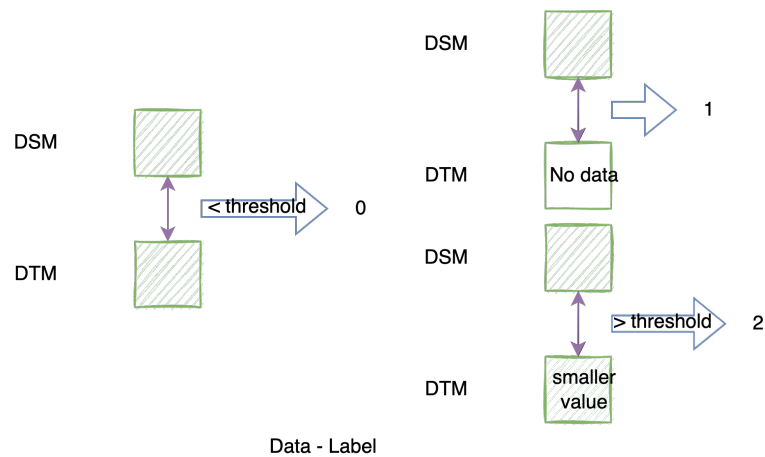


Figure 11: Labeling based on the value difference between the DTM and the DSM.

### 5.3.2 Ground truth map

A ground truth map (Figure 12) for each sample can then be created based on the labeling rules.

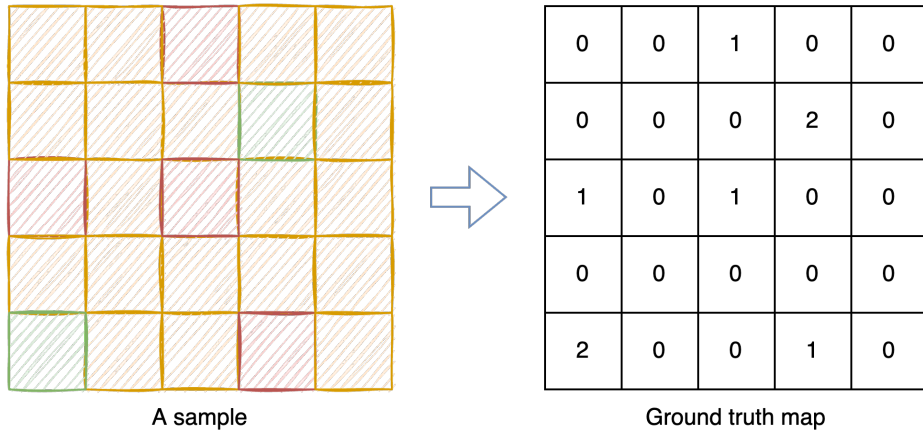


Figure 12: The corresponding relationship between a sample and its ground truth map.

## 5.4 FCN model

A V-Net (Milletari et al., 2016) is designed for this project, the input is a 9 by 9 sample of pixels, and the expected output is a segmentation with label prediction for each pixel (Figure 13).

The approach employs two symmetric contracting (downward) and expanding (upward) paths. It exploits a fully-convolutional structure, with the presence of convolution operations exclusively and the absence of pooling layers.

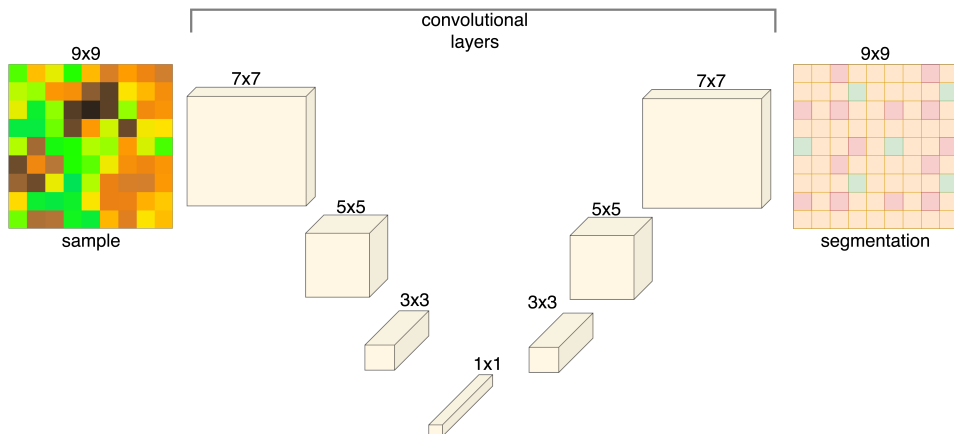


Figure 13: The structure of the V-Net for this project.

## 6 Evaluation and Further work

### 6.1 Evaluation

#### 6.1.1 Within the dataset

1. segmentation accuracy;
2. RMSE between the original DTM and DTM produced after interpolation

### 6.1.2 Comparison with other products using other datasets

1. COPDEM30 used in Hawker et al. (2022)
2. SRTM used in Kulp and Strauss (2018)

## 6.2 Further work

### 6.2.1 Local DEM: higher resolution

Possible solutions:

1. bigger window size
2. deeper V-Net

## 7 Current progress

### 7.1 Dataset builds

A small dataset that contains 5,000 samples has been built. Each sample has 9 features, which are the normalized values in a 3 by 3 window. The data is labeled and split. Figure 14 shows the structure of the dataset.

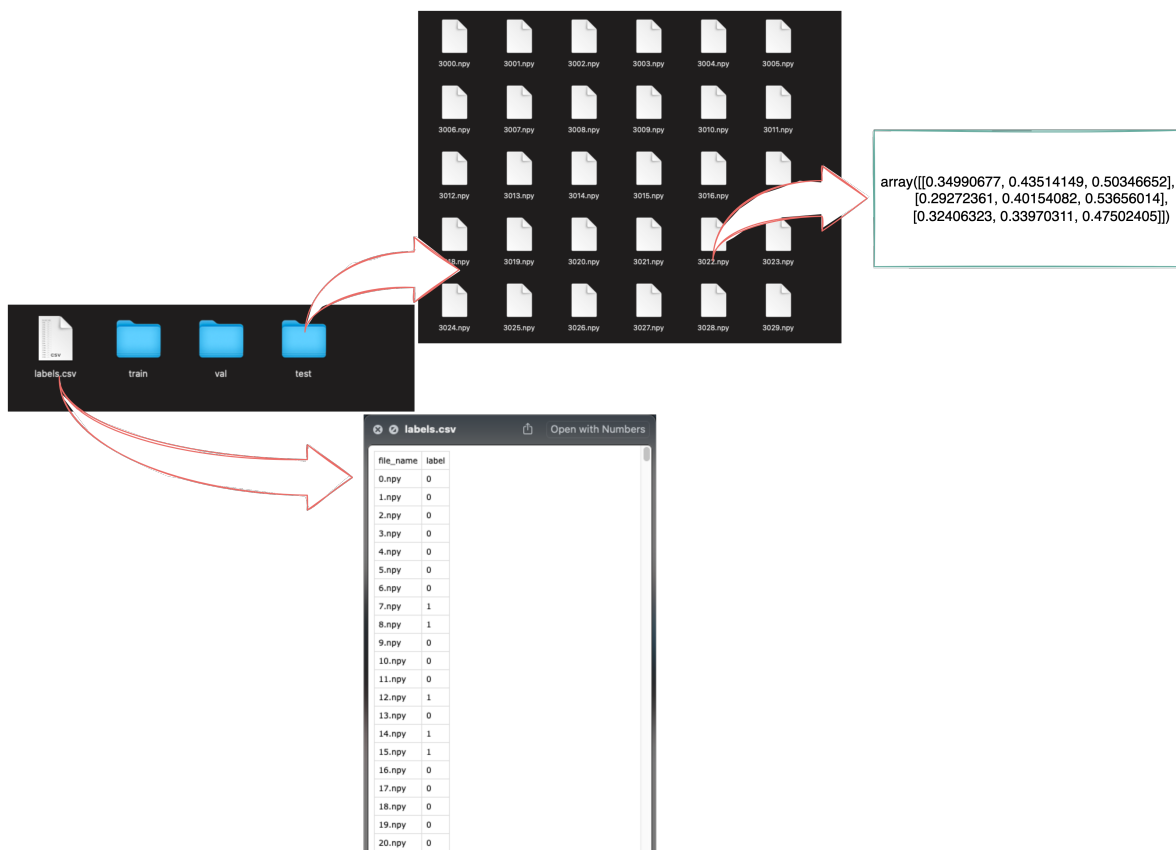
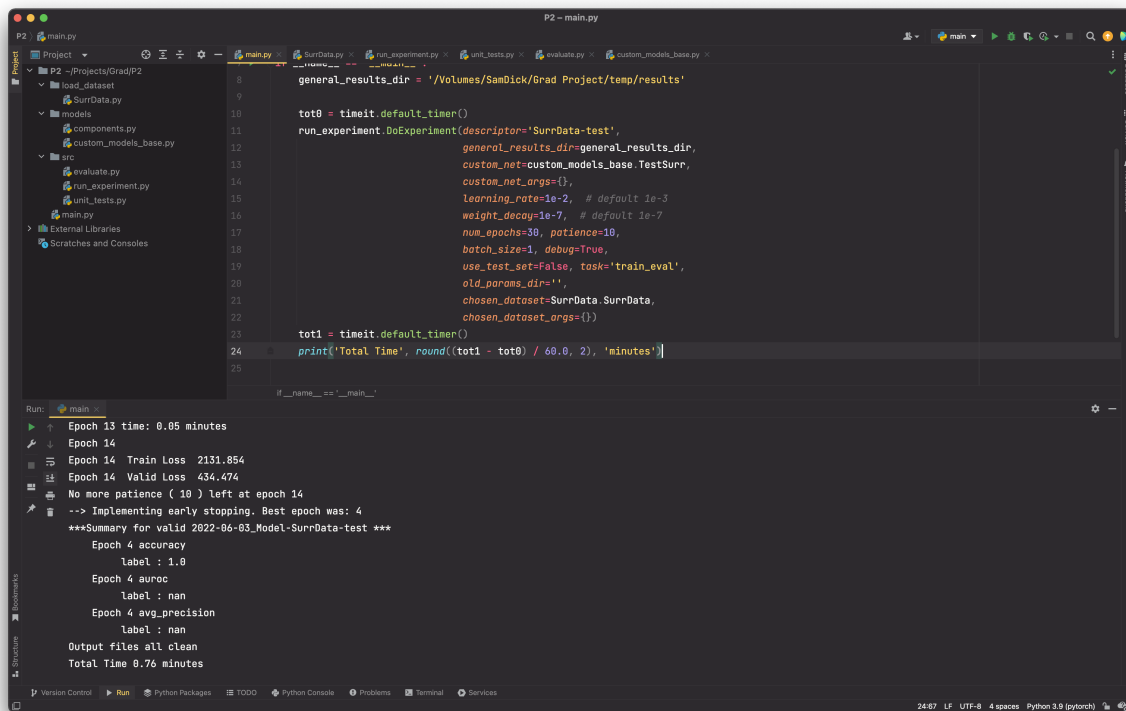


Figure 14: The file structure of the datasets. The labels are stored in a csv file.



## 7.2 A demo Neural Network

A demo neural network with 2 hidden layers has been built. High loss and code errors are shown in the results (Figure 15).



The screenshot shows a Python IDE with a project named 'P2'. The main.py file contains the following code:

```
8 general_results_dir = '/Volumes/SamDick/Grad Project/temp/results'
9
10 tot0 = timeit.default_timer()
11 run_experiment.DoExperiment(descriptor='SurrData-test',
12                             general_results_dir=general_results_dir,
13                             custom_net=custom_models_base.TestSurr,
14                             custom_net_args={},
15                             learning_rate=1e-2, # default 1e-3
16                             weight_decay=1e-7, # default 1e-7
17                             num_epochs=10, patience=10,
18                             batch_size=1, debug=True,
19                             use_test_set=False, task='train_eval',
20                             old_params_dir='',
21                             chosen_dataset=SurrData.SurrData,
22                             chosen_dataset_args={})
23
24 tot1 = timeit.default_timer()
25 print('Total Time', round((tot1 - tot0) / 60.0, 2), 'minutes')
26
27 if __name__ == '__main__':
```

The Run console shows the following output:

```
Epoch 13 time: 0.05 minutes
Epoch 14
Epoch 14 Train Loss 2131.854
Epoch 14 Valid Loss 434.474
No more patience ( 10 ) left at epoch 14
--> Implementing early stopping. Best epoch was: 4
***Summary for valid 2022-06-03_SurrData-test ***
Epoch 4 accuracy
Label : 1.0
Epoch 4 auroc
Label : nan
Epoch 4 avg_precision
Label : nan
Output files all clean
Total Time 0.76 minutes
```

Figure 15: A screenshot of the Python project.

## 8 Time planning

Further work includes methodology and model improvement, testing, and evaluation process. The time planning is shown in a Gantt Chart 16. The expected P4 presentation is in November.

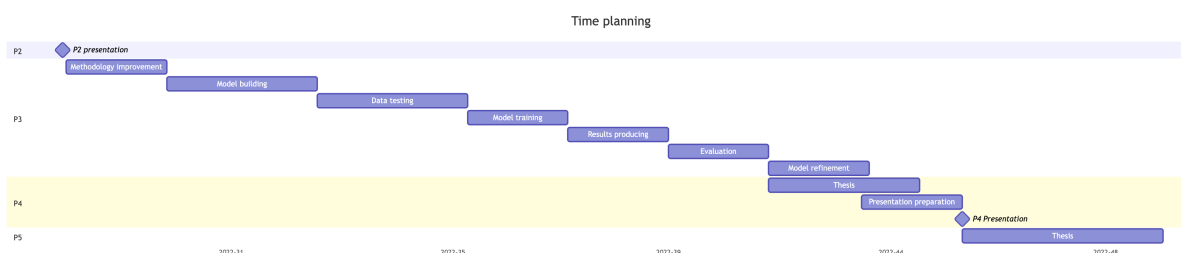


Figure 16: A Gantt chart of the time planning.

## References

- AHN. AHN: The making of, Feb. 2020. URL <https://www.ahn.nl/ahn-the-making-of>. Publisher: AHN.
- C. A. Baugh, P. D. Bates, G. Schumann, and M. A. Trigg. SRTM vegetation removal and hydrodynamic modeling accuracy. *Water Resources Research*, 49(9):5276–5289, 2013. ISSN 1944-7973. doi: 10.1002/wrcr.20412. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/wrcr.20412>. eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/wrcr.20412>.
- D. L. Donoho and others. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS math challenges lecture*, 1(2000):32, 2000. Publisher: Citeseer.
- Esri. AHN4 - Download kaartbladen - Overview, Sept. 2021. URL <https://www.arcgis.com/home/item.html?id=77da2e9eeea8427aab2ac83b79097b1a>.
- C. M. Gevaert, C. Persello, F. Nex, and G. Vosselman. A deep learning approach to DTM extraction from imagery using rule-based training labels. *ISPRS Journal of Photogrammetry and Remote Sensing*, 142:106–123, Aug. 2018. ISSN 0924-2716. doi: 10.1016/j.isprsjprs.2018.06.001. URL <https://www.sciencedirect.com/science/article/pii/S0924271618301643>.
- L. Hawker, P. Uhe, L. Paulo, J. Sosa, J. Savage, C. Sampson, and J. Neal. A 30 m global map of elevation with forests and buildings removed. *Environmental Research Letters*, 17(2):024016, Feb. 2022. ISSN 1748-9326. doi: 10.1088/1748-9326/ac4d4f. URL <https://doi.org/10.1088/1748-9326/ac4d4f>. Publisher: IOP Publishing.
- B. Kazimi, F. Thiemann, and M. Sester. DETECTION OF TERRAIN STRUCTURES IN AIRBORNE LASER SCANNING DATA USING DEEP LEARNING. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume V-2-2020, pages 493–500. Copernicus GmbH, Aug. 2020. doi: 10.5194/isprs-annals-V-2-2020-493-2020. URL <https://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/V-2-2020/493/2020/>. ISSN: 2194-9042.
- S. Kulp and B. H. Strauss. Global DEM Errors Underpredict Coastal Vulnerability to Sea Level Rise and Flooding. *Frontiers in Earth Science*, 4, 2016. ISSN 2296-6463. URL <https://www.frontiersin.org/article/10.3389/feart.2016.00036>.
- S. A. Kulp and B. H. Strauss. CoastalDEM: A global coastal digital elevation model improved from SRTM using a neural network. *Remote Sensing of Environment*, 206:231–239, Mar. 2018. ISSN 0034-4257. doi: 10.1016/j.rse.2017.12.026. URL <https://www.sciencedirect.com/science/article/pii/S0034425717306016>.
- M. Köppen. The curse of dimensionality. In *5th online world conference on soft computing in industrial applications (WSC5)*, volume 1, pages 4–8, 2000.
- D. J. Lary, A. H. Alavi, A. H. Gandomi, and A. L. Walker. Machine learning in geosciences and remote sensing. *Geoscience Frontiers*, 7(1):3–10, Jan. 2016. ISSN 1674-9871. doi: 10.1016/j.gsf.2015.07.003. URL <https://www.sciencedirect.com/science/article/pii/S1674987115000821>.
- J. Long, E. Shelhamer, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. pages 3431–3440, 2015. URL [https://openaccess.thecvf.com/content\\_cvpr\\_2015/html/Long\\_Fully\\_Convolutional\\_Networks\\_2015\\_CVPR\\_paper.html](https://openaccess.thecvf.com/content_cvpr_2015/html/Long_Fully_Convolutional_Networks_2015_CVPR_paper.html).

- M. Meadows and M. Wilson. A Comparison of Machine Learning Approaches to Improve Free Topography Data for Flood Modelling. *Remote Sensing*, 13(2):275, Jan. 2021. ISSN 2072-4292. doi: 10.3390/rs13020275. URL <https://www.mdpi.com/2072-4292/13/2/275>. Number: 2 Publisher: Multidisciplinary Digital Publishing Institute.
- F. Milletari, N. Navab, and S.-A. Ahmadi. V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571, Oct. 2016. doi: 10.1109/3DV.2016.79.
- K. O’Shea and R. Nash. An Introduction to Convolutional Neural Networks. Technical Report arXiv:1511.08458, arXiv, Dec. 2015. URL <http://arxiv.org/abs/1511.08458>. arXiv:1511.08458 [cs] type: article.
- Y. S. Park and S. Lek. Chapter 7 - Artificial Neural Networks: Multilayer Perceptron for Ecological Modeling. In S. E. Jørgensen, editor, *Developments in Environmental Modelling*, volume 28 of *Ecological Model Types*, pages 123–140. Elsevier, Jan. 2016. doi: 10.1016/B978-0-444-63623-2.00007-4. URL <https://www.sciencedirect.com/science/article/pii/B9780444636232000074>.
- T. Poggio, H. Mhaskar, L. Rosasco, B. Miranda, and Q. Liao. Why and when can deep-but not shallow-networks avoid the curse of dimensionality: A review. *International Journal of Automation and Computing*, 14(5):503–519, Oct. 2017. ISSN 1751-8520. doi: 10.1007/s11633-017-1054-2. URL <https://doi.org/10.1007/s11633-017-1054-2>.
- L. Polidori and M. El Hage. Digital Elevation Model Quality Assessment Methods: A Critical Review. *Remote Sensing*, 12(21):3522, Jan. 2020. ISSN 2072-4292. doi: 10.3390/rs12213522. URL <https://www.mdpi.com/2072-4292/12/21/3522>. Number: 21 Publisher: Multidisciplinary Digital Publishing Institute.
- D. Pratibha, P. Shingare, M. Sumit, and S. Kale. Review on Digital Elevation Model.
- V. Rodriguez-Galiano, M. Sanchez-Castillo, M. Chica-Olmo, and M. Chica-Rivas. Machine learning predictive models for mineral prospectivity: An evaluation of neural networks, random forest, regression trees and support vector machines. *Ore Geology Reviews*, 71:804–818, Dec. 2015. ISSN 0169-1368. doi: 10.1016/j.oregeorev.2015.01.001. URL <https://www.sciencedirect.com/science/article/pii/S0169136815000037>.
- D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, Oct. 1986. ISSN 1476-4687. doi: 10.1038/323533a0. URL <https://www.nature.com/articles/323533a0>. Number: 6088 Publisher: Nature Publishing Group.
- Y. Su, Q. Guo, Q. Ma, and W. Li. SRTM DEM Correction in Vegetated Mountain Areas through the Integration of Spaceborne LiDAR, Airborne LiDAR, and Optical Imagery. *Remote Sensing*, 7(9):11202–11225, Sept. 2015. ISSN 2072-4292. doi: 10.3390/rs70911202. URL <https://www.mdpi.com/2072-4292/7/9/11202>. Number: 9 Publisher: Multidisciplinary Digital Publishing Institute.