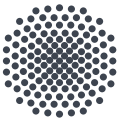


Universität Stuttgart

masterarbeit am ifp

Semantic Segmentation with Remote Sensing Data and Reference Labels Based on Simulation Methods

Shu Suo

Supervisor: Dr. Stefan Auer (DLR)

Examiner: Prof. Dr. Uwe Sörgel

Declaration

Hereby, I **Shu Suo** confirm that the submitted Master thesis on the topic

Semantic Segmentation with Remote Sensing Data and Reference Labels Based on Simulation Methods

1. has been composed by myself fully independently,
2. that I have used no sources other than those indicated, and that I have appropriately declared all citations,
3. that the submitted Master thesis was neither completely nor in substantial parts subject of another examination,
4. that the work was neither completely nor in parts published before, and
5. that the electronic copy is consistent with the other copies.

_____ _____

place, date, signature

Contents

Table of figures

Abstract

Deep learning provides more opportunities for image segmentation. Meanwhile, neural network becomes a popular method for ground surface classification. Manually selected label in training data needs investment in both cost and time. However, this problem could be well solved by the program, named SimGeoI[1]. SimGeoI simulates optical image, SAR (Synthetic Aperture Radar) image and ground-surface labels from DSM (digital surface model) data.

In this thesis, a batch processing part of SimGeoI and the semantic segmentation based on data set generated by SimGeoI are implemented. Through SimGeoI batch processing, the satellite dataset for semantic segmentation can be significantly expanded. Semantic segmentation based on simulation-methods generated dataset is also implemented. The case studies in Munich area, with WorldView-2 imagery and TerraSAR-X data, confirms the opportunity of semantic segmentation using dataset generated by SimGeoI.

Index Terms: SimGeoI, DSM, satellite optical image, SAR data, training data set, urban classification, image segmentation.

1 Introduction, Motivation and Objectives

The interpretation of earth surface imagery is a fundamental topic in remote sensing. Ground classification with pixelwise resolution is a general and important task in remote sensing data interpretation. There is also variety of method to classify remote sensing image, including supervised and unsupervised learning. Supervised learning is a machine learning task that learns the function of mapping input to output based on example input-output pairs[2]. It infers a function from labeled training data consisting of a set of training examples[3]. On the opposite, unsupervised classification is kinds of self-organization, which allows for modeling of probability densities over input[4]. Unsupervised classification is applied when human-labeled data is not available. Both learning methods and the combination of them, named semi-supervised learning, are widely used.

Semantic segmentation is a popular topic in in the field of computer science. It worth to mention that comparing to the definition of classification in remote sensing, classification in computer science means to mark a label to an object, which is usually in an area. In the field of remote sensing, classification for each pixel is called semantic segmentation. To recognize the label of each pixel, convolutional network is modified to Fully Convolutional Network (FCN) by Jonathan Long, Evan Shelhamer and Trevor Darrell in 2014[5]. FCN becomes one of the most popular supervised learning methods.

Requirement of remote sensing dataset rises with the rapid development of deep learning. Meanwhile, the amount of remote sensing satellite increases greatly as the optical sensors and SAR sensors prosper. High-resolution optical imaging satellite and more SAR satellite contribute to multiple data set prominently. There are some benchmark datasets for remote sensing, like ISPRS benchmark dataset of Potsdam. However, the samples in training dataset influences the result of the neural network, while buildings and nature sceneries vary greatly in different countries and areas. For most areas, training data set for a specific area is still limited because of the high price of manually labeling land categories on a large number of images in that specific area. Hence, an automatic process to generate benchmark would facilitate the application of remote sensing data in convolutional neural network (CNN) training.

To solve the dataset limitation problem, in this thesis, a tool is used to generate “ground truth” automatically and to manage the data into a dataset which could be used as training data. Batch processing of SimGeoI marks label of pixel for either optical image or SAR data.

The reference label images generated by SimGeoI are used in fully convolutional network. The tool for preprocessing input data of neural network and program for FCN are finished. In order to investigate the performance of the proposed tool, I selected optical images from WorldView2 and SAR intensity images from TerraSAR-X over the area of Munich.

The algorithms and process will be introduced in the following way:

Chapter 2 introduces the background and related algorithms for this thesis, including the principle of SimGeoI and semantic segmentation algorithms. In SimGeoI part, DSM is used to

generate images based on rendering (raytracing) and derive interpretation layers. In deep learning part, the fully convolutional network is introduced, which is used in test and analyses.

Chapter 3 explains the process of the whole project. Data examples are given and data preprocessing steps for both SimGeoI and FCN training are explained. SimGeoI tool is extended with a batch processing to produce a group of data automatically.

Chapter 4 gives examples of case study, with optical image dataset and SAR intensity image dataset in Munich Area.

Chapter 5 summaries the overall conclusion of the project. Firstly, the advantages and achievements are mentioned to emphasis the meaning of the thesis. Secondly, the disadvantages like limitation and accuracy are also listed. An outlook in this direction and further possible disposals are also discussed in the end.

2 Background and Related Works

2.1 Simulation Data Generation by SimGeoI

SimGeoI is a simulation framework for the ground object interpretation of original optical and SAR images. This frame takes the effect of sensor-specific geometric distortion into consideration, with the input data as both Digital Surface Model (DSM) and meta-information of corresponding real selected data. This tool offers elevation related information into the urban classification. DSM means digital surface model. The value on each pixel means the altitude on the corresponding area.

In the following, the main process will be introduced in A) and the advantages and limitations will be given in B).

A). Main process of SimGeoI

The main process of SimGeoI contains scene definition, ray tracing, image generation, geocoding, interpretation layer generation and image part extraction. Only interpretation layers are used in this thesis. Interpretation layers are layers indicate some basic class label, like ground, vegetation, and building. Those interpreted mask layers are superimposed in one image, which is also called label image.

Figure 2-1 shows the overall processing flow of SimGeoI.

In the first row as input, satellite image offers meta data like the nadir angle of sensor. DSM is the main input element, which provide the 3D information of the ground. In the preprocessing part, inputted DSM is processed with filters.

The preprocessing steps are as following:

First, SimGeoI generates normalized DSM (nDSM), and digital terrain model (DTM). Only with the absolute elevation data from DSM, it is hard to define the ground and building area. To solve this problem, DSM data should be processed to nDSM, which indicates the related elevation. DTM is calculated from DSM, with algorithms to define the horizontal plane like scanline extent, height threshold, slope threshold.

In this step, some “holes” in the DSM will be filled, which come from errors in DSM data, where some area gets no elevation data or wrong elevation. Also, there are areas without overlapped pixels on stereo images, causing no height information in DSM data, especially in urban areas with dense high building. For those abnormal values, SimGeoI sets “0” in DTM to mark those pixels as ground.

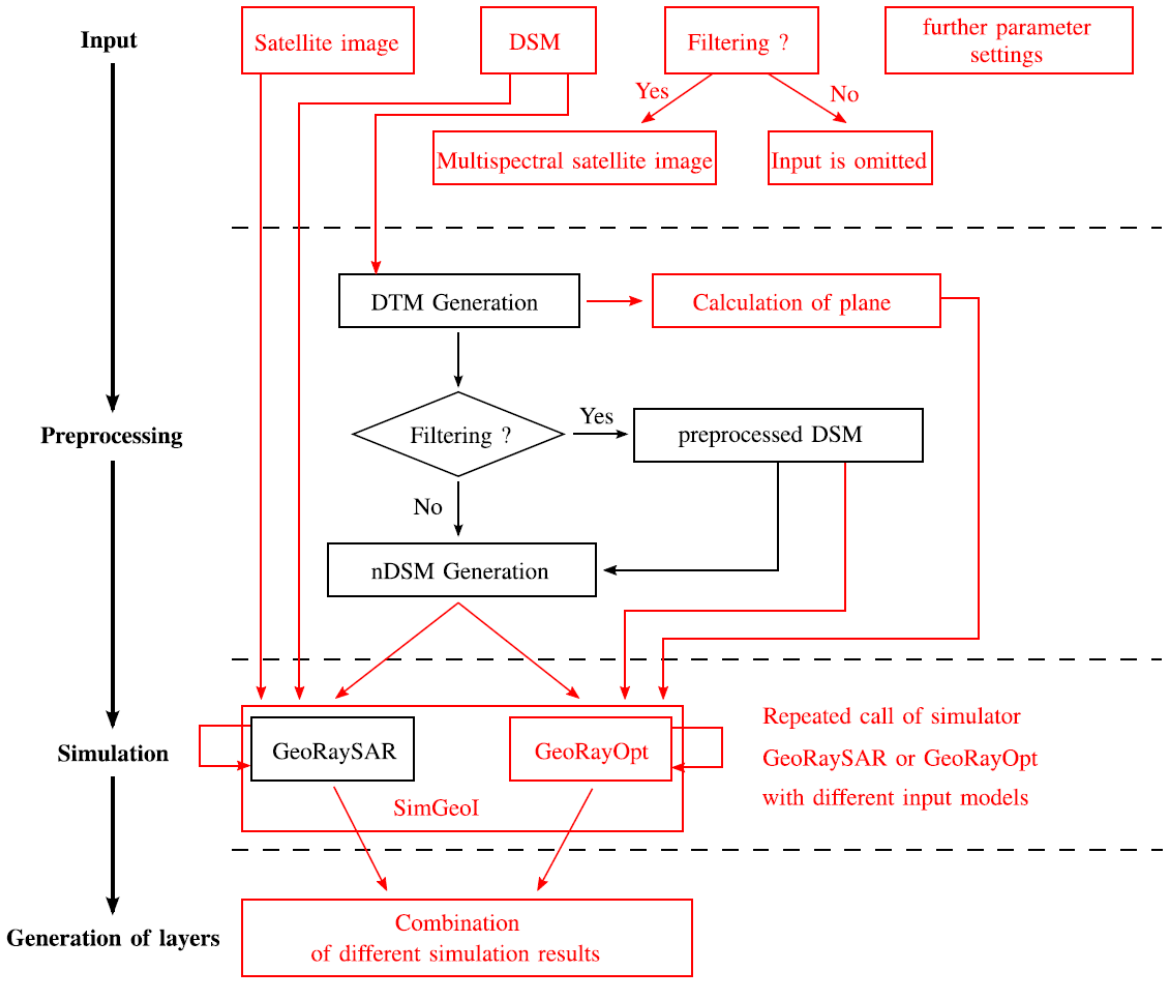


Figure 2-1. Flow chart of SimGeoI[1]

In case if the "filter" is selected, a median filter will be applied to smooth the DSM data, to get rid of noise in original DSM. Meanwhile, meadow will be filtered to be flatter field.

Vegetation filtering is an optional process, depending on whether there is an ortho image with Red (R) and Near Infrared (NIR) bands. If such an ortho image is given, R and NIR bands will be used to do the vegetation estimation with concept Normalized Difference Vegetation Index (NDVI). The corresponding calculation of NDVI is shown in formula (2.1). When the index is larger than a threshold, this area is regarded to contain green vegetation.

$$\mathrm{NDVI} = \frac{\mathrm{NIR} - \mathrm{R}}{\mathrm{NIR} + \mathrm{R}} \qquad \text{formula (2.1)}$$

The filtered DSM and DTM data are used to generate nDSM. This data keeps the values where difference value between DSM and DTM larger than 0.1 m. After this step, the elevation model is without trees, so that the elevation information could be used to distinguish buildings without noise from elevated vegetation.

Then the most important part, simulation, will be realized with nDSM data. Here SimGeoI offers two modes: optical simulation and SAR simulation. In our application, "*.xdibias", a self-defined image format from DLR, is used to store the input and output data. In this format, the image data and metadata information are saved in one folder. Image part is saved in a file and metadata is saved in a "*.XML" file in the same folder. In this format, the metadata is more convenient for the manual check.

The simulation principle of optical scenario is shown in Figure 2-2. Among these parameters, view angle and illuminating source angle are read from metadata file. H_{ref} should be provided in the parameter file. When view angle and illuminating angle are the same, there will not be a shadow class. Otherwise, the area, which could be observed from view direction but not illuminated by source, will be labeled as shadow class.

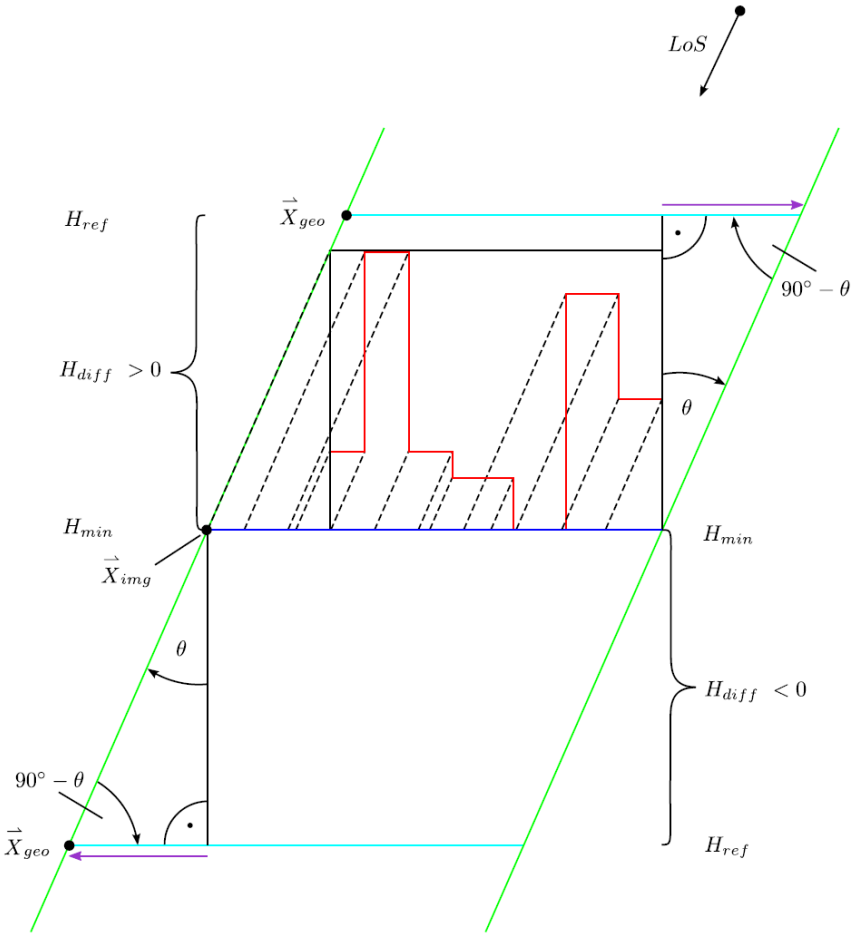


Figure 2-2. Principle of optical image simulation[1]

Figure 2-2 indicates the basic principle of optical projection, with main parameters of illuminating angle, view angle (θ) and reference height (H_{ref}) of the projective plane. The coverage of the image plane is marked with a green line. The red line represents the DSM

geometry, and its simulated optical image is projected onto the blue-marked plane (at the minimum height of the DSM). The cyan line indicates projected optical image on a projection plane with reference height of H_{ref}. The purple arrow marks the offset to be compensated (in UTM coordinates).

Considering the large altitude of sensor, like 770 km for WorldView-2, and the relatively small frame of image, parallel incident angle is used to simplify the calculation. The parallel incident angle is computed for the center pixel of the image, as "average" incident angle in that area.

Since the incident angles of optical images are usually small, and view is closed to nadir view, the displacement caused by height difference is usually not obvious as that in SAR.

In SAR case, the illuminating angle is same as view angle, due to activate sensor. The reference height is also given in the orbit meta file.

Figure 2-3 shows the effect of different projection planes on SAR simulation. The colored lines and symbols have same meaning as in Figure 2-2.

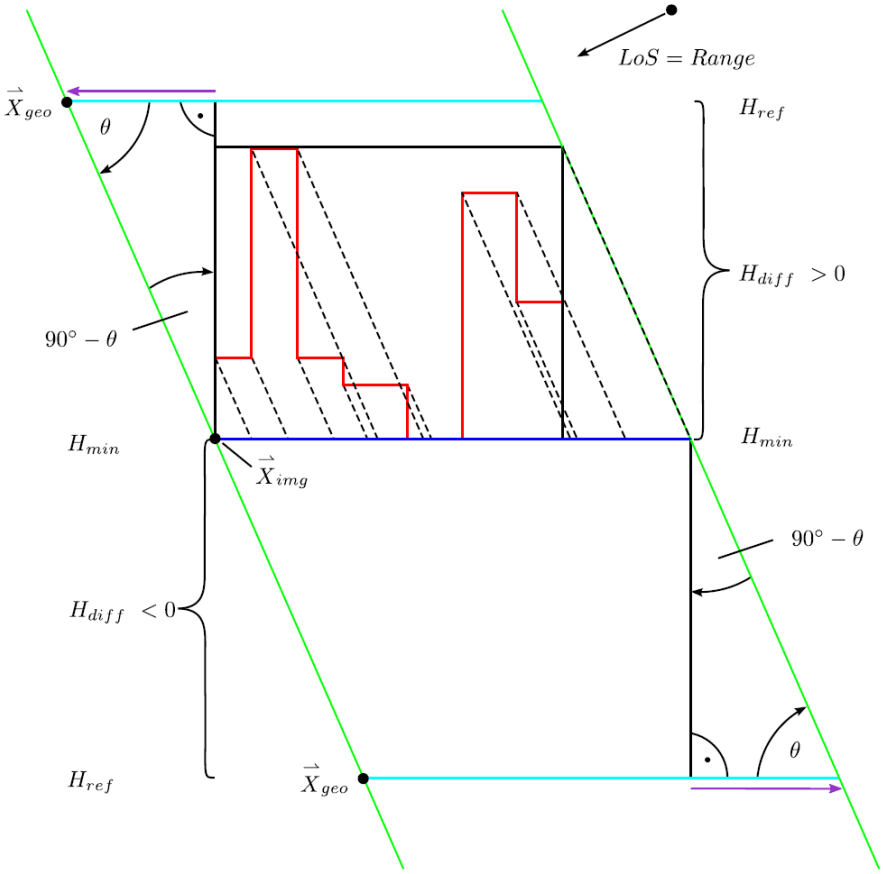


Figure 2-3. Principle of SAR image simulation[1]

SAR image simulation part calculates the first and second reflection only. The range of first reflection should be the distance between sensor and object. But, in SimGeoI, a parallel perspective is used to simplify the range calculation. Because satellite altitude of TerraSAR-X is 500 km[6], the object size is very small for the distance to sensor. So, the SAR signal is considered as parallel signal in a small image.

The projected image depends on view angle (θ) and distance between objects and sensor, which is also called range. Since view angle to collect SAR data is significantly larger than the angle to get optical image, in SAR simulation part, the height difference usually causes dramatical displacement in the image.

The output data used in this thesis is the overlapped layers indicating different classes. The output values of optical layers are: "0" for background, "1" for no value in DSM, "2" for shadow, "3" for ground, "4" for man-made building and "5" for vegetation.

For SAR case output, values in output means: "0" for background, "1" for no value in DSM, "2" for layover shadow (caused by building), "3" for ground, "4" for building, "5" for vegetation.

B). Advantages and Limitations of SimGeoI

Advantages:

1) Using geometry information to separate ground and building.

2) Use elevation information from DSM to identify buildings, separate from ground class which usually has similar feature in RGB feature space.

3) Give chance to do the segmentation for off-nadir SAR data, which is very difficult for manual interpretation.

4) Make image segmentation with varying sensor perspectives possible.

Limitations:

1) DSM accuracy restricts the accuracy of simulation scenario.

2) The classified result is with accuracy on object level. Some blurs in the edge of different classes.

3) Due to the perspective principle, the projected image may extend the boundary of area in inputted DSM, and labels in some area are not predicted.

2.2 Semantic Segmentation Algorithm

It is challenging to identify different object with classic methods, especially if the methods use pre-defined features. Deep learning is one of the most popular field in image classification. Image classification gives one label as an output, which gets the highest probability index. In classification task, there is only one label for an image or an area in image, like “tabby cat” mark in the upper graph in Figure 2-4. The shallower layers are the same as classification networks, like VGG16. The last few layers are fully connected layers (FC), which gives the final label for an image.

In my task, a label with acceptable accuracy should be given for each pixel of an image. Fully convolutional networks (FCNs) as proposed by Long et al[5], enable the prediction of pixelwise labels for the task of semantic segmentation. These methods will be introduced in detail in A). Afterwards, a short introduction about parameters which influence the training network will be given in B), and the evaluation method for a network will be introduced in C).

A). Fully Convolutional Networks

FCNs substitute the fully connected layers in CNN frame with convolutional layers, so as to enable an output as heatmap and make the dense learning possible.[5] These network output a heat map with predicted labels, instead of a single label for the whole image. An example of a heat map, shown at the bottom of Figure 2-4. The heatmap is smaller than the inputted image, because of down-sampling in max pooling.

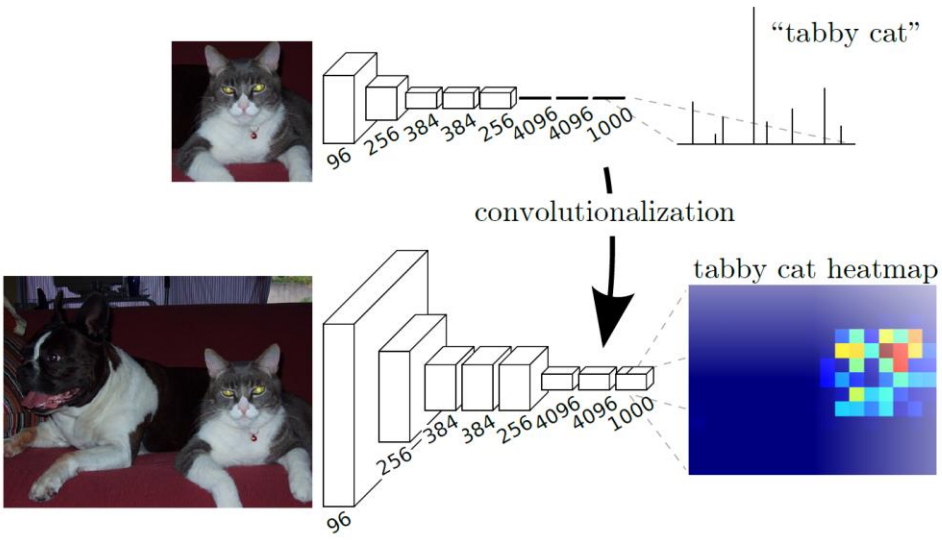


Figure 2-4. Transforming fully connected layers into convolution[5]

After the pooling layers and convolutional layer, upsampling layers are added to give predicted marks on each inputted pixel. The overall idea of FCN is shown in Figure 2-5.

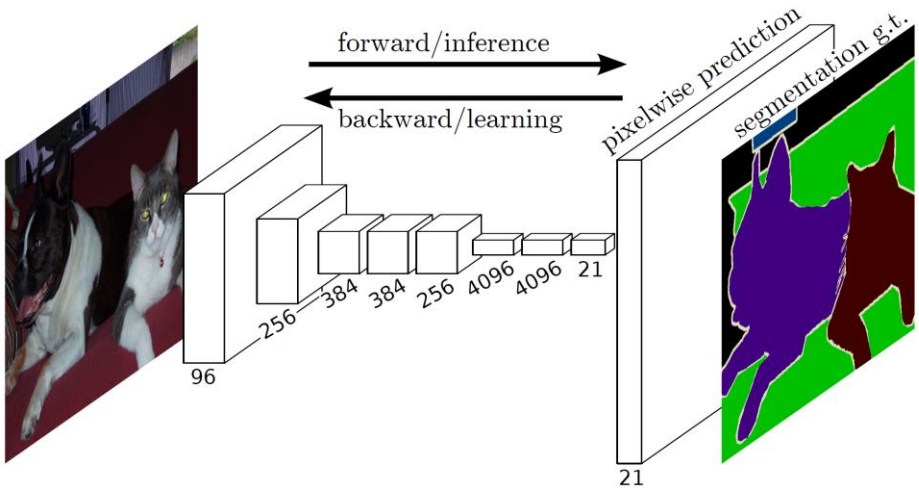


Figure 2-5. Fully convolutional network[5]

The prediction of image is firstly with a smaller size and would be upsampled to match the expected size. There are 3 upsampling levels: FCN-32s, FCN-16s and FCN-8s, as in Figure 2-6. In the upsampling step, a bridge is built between finer, lower level layers (left side in Figure 2-6) and the rougher, higher level layers. with this bridge, a higher-resolution segmentation will be obtained.

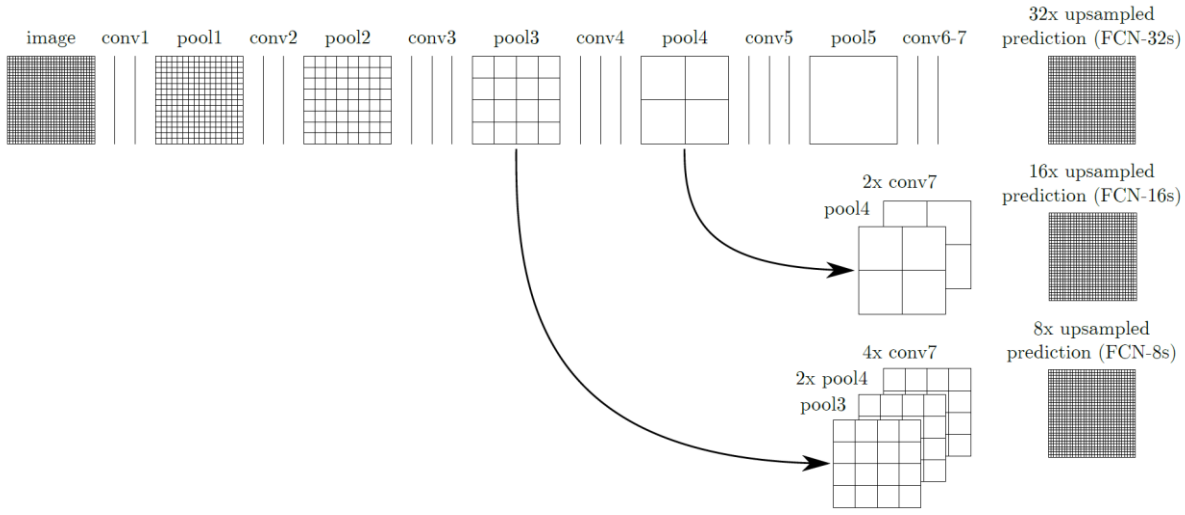


Figure 2-6. Structure of FCN [5]

In the process to upsample the predicted result, a backwards convolution (deconvolution) is applied. The deconvolution filter combines "what" and "where", and gives local prediction respecting global structure.[5] This step makes the whole process an end-to-end, pixel-to-pixel segmentation.

For the training part, images and corresponding same-sized label images are used as input data. The image should be colored image with 3 channels, and benchmark image is one-channel

matrix with number that indicating the class number. Each number in benchmark map indicates a specified object, numbering from “0”.

In test part, the output is also a one-channel matrix.

B). Training parameters

Input parameters influence the training, like epoch or iteration number, batch size, loss function, and optimization methods.

An epoch refers to a single cycle through the whole training dataset. Batch size is the image number used in one calculation. Calculation with the defined batch size for one time, is one iteration. Iteration are used as parameters in this thesis, instead of epoch as the training time. When the batch size is not “1”, people usually add a “drop_last” mode to avoid a remainder.

A widely used loss function is cross entropy. The formula is in formula (2.2).

$$L = -\sum_{i=1}^{c}\left(y^{(i)} \cdot \log(\hat{y}^{(i)})\right) \qquad \text{formula (2.2)}$$

Optimization is often applied via gradient descent (GDS) in deep learning. The parameters in this part include learning rate, weight decay and momentum.

The learning rate is an adjustment parameter in an optimization algorithm that determines the step size of each iteration while moving towards the minimum value of the loss function.[7] A large learning rate achieve quick convergence, but may cause overshooting. On the contrary, a small learning rate will make the network convergent slower but could avoid strong overshooting. The problem is that a too small learning rate may lead to a local optimum but not a global one. So a suitable learning rate for a specified dataset should be chosen.[8] To avoid oscillation and local optimum, weight decay and momentum are added for optimization.

Decay is used to stabilize learning in a good place and avoid oscillations. When the constant learning rate is too high, learning may jump back and forth at the minimum value.[8]

Momentum is like a ball on a rolling hill. In our expectation, the ball should fall at the lowest point of the mountain (corresponding to the lowest error). When the error cost gradient goes in the same direction for a long time, the momentum can both speed up the learning speed, and avoid local minima by "tumbling" small bumps.[8]

C). Evaluation of a trained model

The dataset is divided into three parts: training data, validation data and test data. Training data is used to training the parameters in the network. Validation dataset is sed to calculate some index for evaluation. Because the validation is independent from validation dataset, so the evaluation is more meaningful. Even if the model overfits the training dataset, the validation loss IoU and loss will show some difference with training evaluation index.

The authors of FCNs give some index like matrices to judge the training situation. The evaluation matrices are pixel accuracy, mean accuracy and mean Intersection over Union (IoU)[6].

Formulas are as formula (2.3), formula (2.4) and formula (2.5)

Pixel accuracy:	$\sum_i n_{ii} / \sum_i t_i$	formula (2.3)
Mean accuracy:	$(1/n_{cl}) \sum_i n_{ii}/t_i$	formula (2.4)
Mean IoU:	$(1/n_{cl}) \sum_i n_{ii} /(t_i + \sum_j n_{ji} - n_{ii})$	formula (2.5)

Where n_{ij} indicates the number of pixels marked as class i and predicted as class j, n_{cl} means the number of classes applied in the model, and $t_i$means the total number of pixels in class j. t_i is calculate as $t_i = \sum_j n_{ij}$.

IoU is the overlapped ratio of benchmark defined pixels and predicted pixels in one class, as shown in Figure 2-7. This value indicates somehow the accuracy of a model. If the IoU is larger, the result is usually considered better.

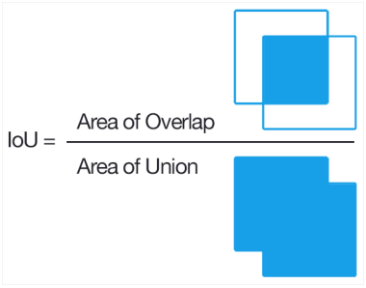


Figure 2-7. Meaning and formula of IoU[9]

One exception is the so-called overfitting. In overfitting case, the IoU will be nice but the model is not the best one. An example of this situation is shown in Figure 2-8, with black straight line being good fitting and blue curve line being overfitting. This situation appears when the network is with over deep layers or with too many iterations of training.

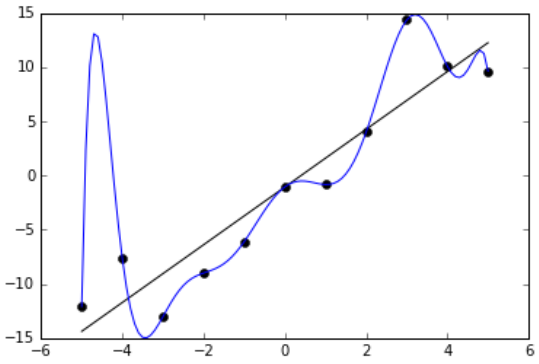


Figure 2-8. Overfitting[10]

3 Methods of Semantic Segmentation Based on Simulation Driven Data

Both optical image and SAR data can be simulated in SimGeoI. The DSM is used to apply the simulation in SimGeoI, to generate simulated image as well as the mask layers. The layers indicating different classes will be stored in the DSM data folder. Then the data needed in semantic segmentation will be collected and preprocessed as image dataset.

The overall procedures of two kinds of data are similar, as shown in Figure 3-1. The whole project is separated into two steps: preprocessing (1) and batch process of SimGeoI, preprocessing (2) and FCN training. The main procedure is shown in the flow chart in Figure 3-1. Difference mode and different input data (optical image or SAR data) will give different simulated data and mask layers.

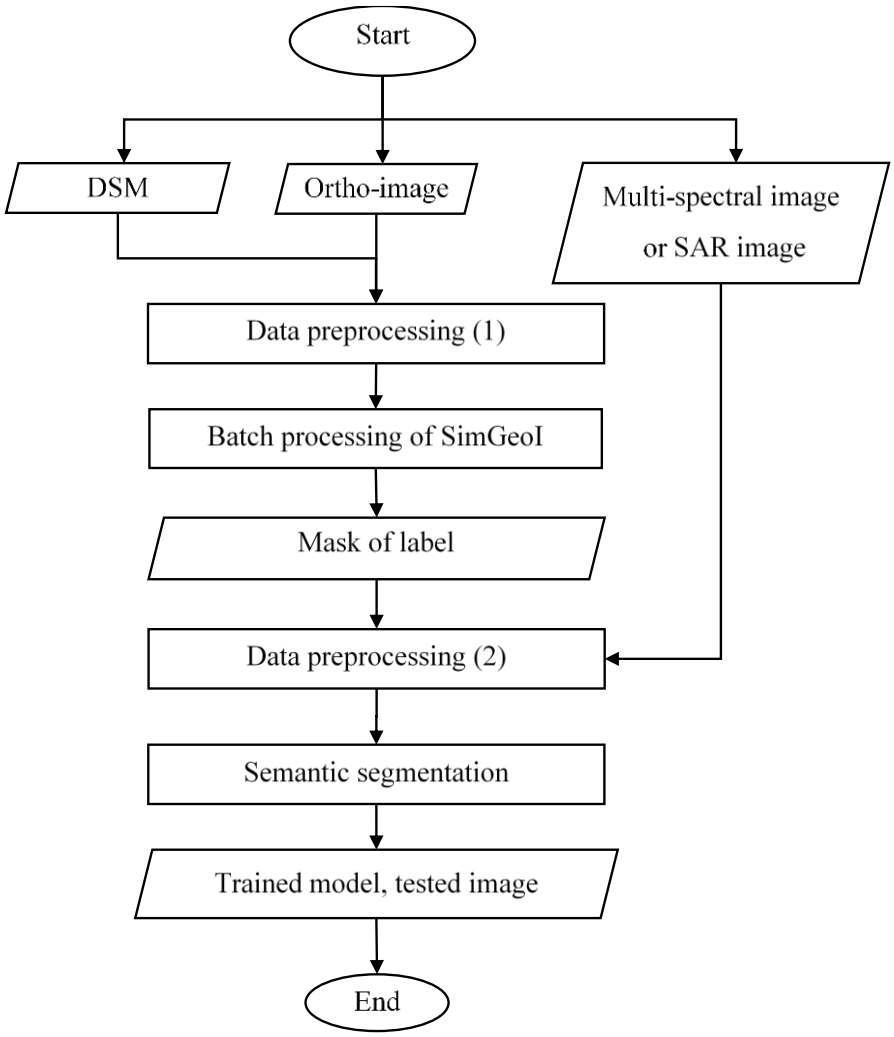


Figure 3-1. Flow chart for data process

In the following section, the data source introduction, data preprocessing and batch processing of SimGeoI, as well as data preprocessing and semantic segmentation will be explained in detail.

3.1 Data Sources Introduction

There are several data sources for our work. To get label mask layers, SimGeoI requires input data like DSM data and orthoimages. For the semantic segmentation training, optical remote sensing images and SAR intensity images are used input data. Examples for those data sources will be given, and some nature features will be introduced.

The inputted data could be one image or more than one. Data format is "*.xdibias", which is developed by German Aerospace Center (DLR). Data in this format is constructed with image data and "*.XML" file storing metadata for the image. The metadata includes georeferenced information like coordinates, the spatial resolution, signal channels, capturing time and sensor angle.

An example is shown in Figure 3-2, where the lighter pixel indicates higher area and the dark area could be ground.

Figure 3-2. Example of DSM as input of SimGeoI

The DSMs used in this work are produced from overlapped panchromatic images, which offer higher spatial resolution than multi-spectral images. The DSM data can be merged into a big frame and stored in a single file.

Orthoimage is produced from DSM data and original multi-spectral image. This data is only used to identify vegetations in SimGeoI, with NDVI value. The orthoimage is in orthographic

projective view while original image from satellite is in perspective view. Figure 3-3 illustrates the difference. Geometrically corrected ("orthorectified") by elevation information from DSM, the orthoimage contains same channel information as original multi-spectral image. An example of orthoimage is given in Figure 3-4.

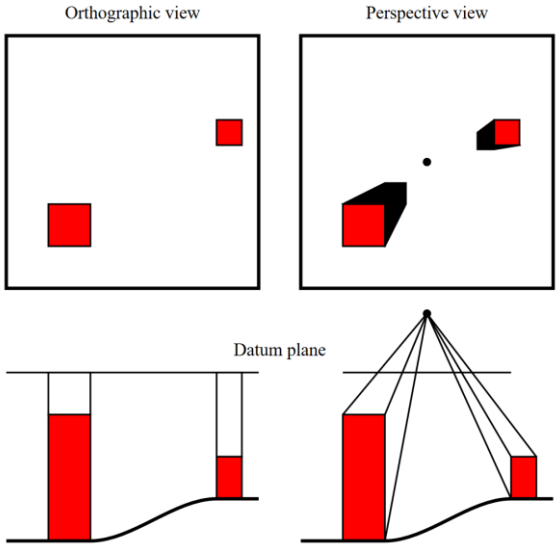


Figure 3-3. Orthographic projection and perspective projection[11]

Figure 3-4. Example of orthoimage

A comparison of an optical image and its orthoimage is shown in Figure 3-5. There is displacement of elevated buildings in the original optical image, shown in (a). In the orthoimage

(b), the pixels of building are rectified to the position of orthographic view. This phenomenon is obvious for high building, like the church in center area. Some problems may occur because of the accuracy of DSM.

(a) original optical image

(b) orthoimage

Figure 3-5. Comparison of optical image and orthoimage

For the segmentation part, input data includes optical remote sensing images or SAR images, and the corresponding label images. Label images will be introduced in chapter 3.2, as output data of SimGeoI.

Optical remote sensing images used in this thesis are from satellite WourldView-2. The data is captured by push broom optical sensor, so that the level 2 image has perspective view in cross-track direction. There are 8 channels in WorldView-2 multispectral image. Pixel grey value could reach 11bits and is coded in 16-bits integer number. An example of color image with RGB channels extracted from8 channels are given in Figure 3-14.

SAR image is input image for semantic segmentation. The data is with only one channel and coded in 16-bits integer number. Examples of optical image and SAR intensity image are shown in Figure 3-15. Building edge can make strong reflection in SAR image.

3.2 Image Preprocessing (1) and Batch Processing of SimGeoI

The input data of the proposed workflow includes DSM data, ortho-image (with multi-spectral) for overlapped area and corresponding original multi-spectral image.

Considering that the application situation is usually with non-orthorectified image, original perspective view images are used as input in the later training part. Optical image with

perspective view also keeps the façade information to be used in further application, like 3D reconstruction.

Because the number, size, coverage area and spatial resolution of input data are different, a preprocessing should be performed before the batch processing of SimGeoI. The following section A) and B) will introduce these two steps separately.

A). Data preprocess (1)

Data preprocess for SimGeoI part is to cut the DSM data into a nice size and organize the data in a folder for the convenience of the following step.

A large image causes some troubles in SimGeoI processing. On one hand, larger data need more computer memory when running SimGeoI. Meanwhile, the large simulated model will also make the calculation time much longer. On the other hand, due to the perspective principle as shown in Figure 2-2 and Figure 2-3, the difference of view angle on edge of large image cannot be neglected. Moreover, the original input DSM data and satellite image are in large frame and cover different area. So, it is better to cut it into small pieces and process them one by one.

The large DSM data is cut into small pieces in same size. The georeferencing data of small DSM will be used to extract small orthographic image as well as original optical image covering the same area. Then three folders with small DSM data, small orthoimages and small optical images are generated. To compromise the image for SimGeoI and the image for neural network training, 522*522 pixel is selected as the optical size. A tool package named “GDAL” is used to clip the image in within same coordinates. This tool will save the new smaller images with their metadata.

We used the following strategy:

Firstly, check whether the covering area of small DSM is with optical or SAR image. Only the overlapped area should be kept in dataset.

Secondly, check whether the optical or SAR image contains “0” in this area. If “0” exist in image, then there are pixels with no value in image, so this small piece should not be kept in dataset.

The output data is three folders with small DSM data, orthoimages and optical images. Each image is named with city and number indicating the related position. The cutting method is shown in Figure 3-6.

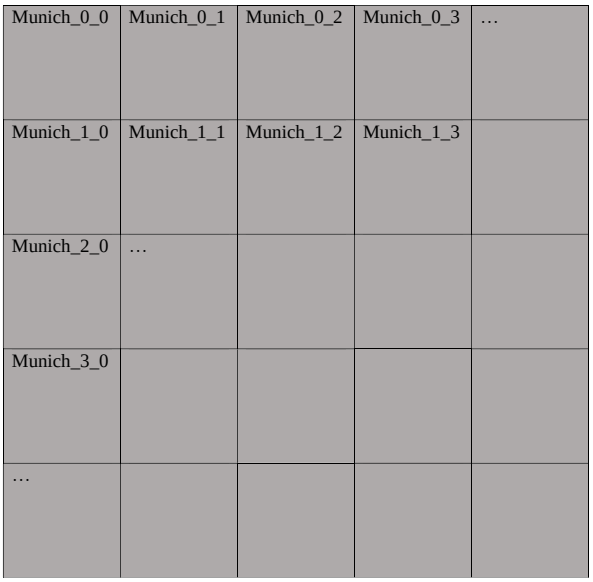


Figure 3-6. Cutting of large input image

B). Batch processing of SimGeoI

Based on the old SimGeoI program, the batch processing function is extended. The old SimGeoI program generates layers of urban class for one pair of DSM data and orthoimage. The new batch processing of SimGeoI allow a sequence processing for a group of data.

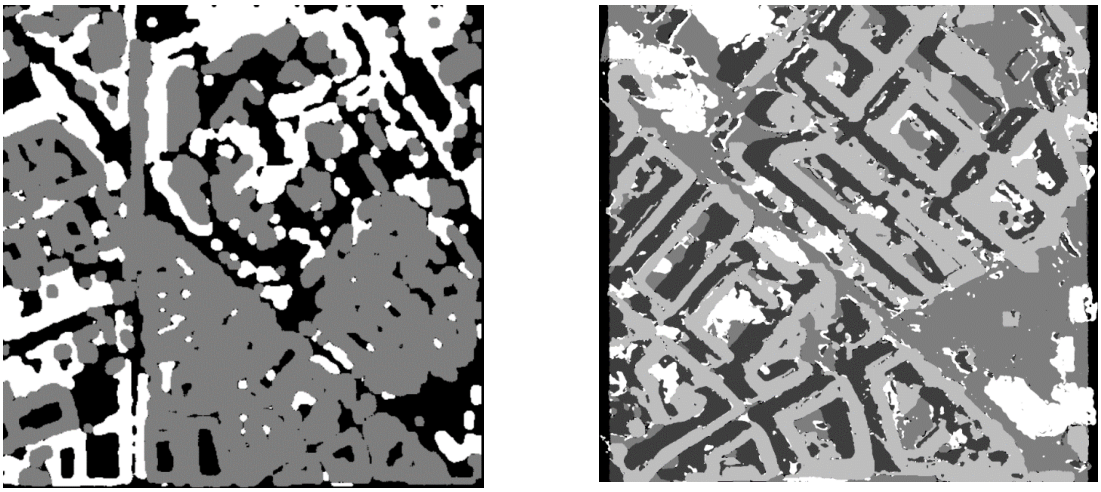
When the DSM and the corresponding orthoimage have different spatial resolution, the coarse data (with lower resolution) defines the resolution of output data. The original optical image defines the metadata of output data.

Output data includes mask layers and simulated image. Those data with be store in the same folder as DSM data. To be noticed, the output label image usually covers larger area due to perspective view, which brings some difficulty in data organization for deep learning, where the image is not directly linked to coordinates.

This part is meaningful because manual interpretation of SAR image is very difficult. SimGeoI use 3D model to produce a simulated SAR image and the corresponding label image, which gives possibility to relate the real SAR image and interpolation information.

C). Samples of label mask from simulation driven data

In this project, images made of overlapped mask layers are selected to be used as reference label in next chapter. Samples are given in Figure 3-7. In Figure 3-7, (a) is the mask layers of optical image, (b) is the mask layers of SAR image(covering southeast quarter of optical example). The comparison indicates that SAR mask images are different with optical mask images, due to the different perspective principles. Because of a large incident angle for SAR data, the SAR image and label contain a lot of pixels showing façade of building.



(a) Mask layers of optical image

(b) Mask layers of SAR data

Figure 3-7. Example of overlapped mask layers

The link between the mask layers and the real images is the simulated image. An example of simulated optical image is given in Figure 3-8. The 3D model is mapped to a certain plane and is consistent with real optical image. Simulated images like it link real image and the simulated mask layers.

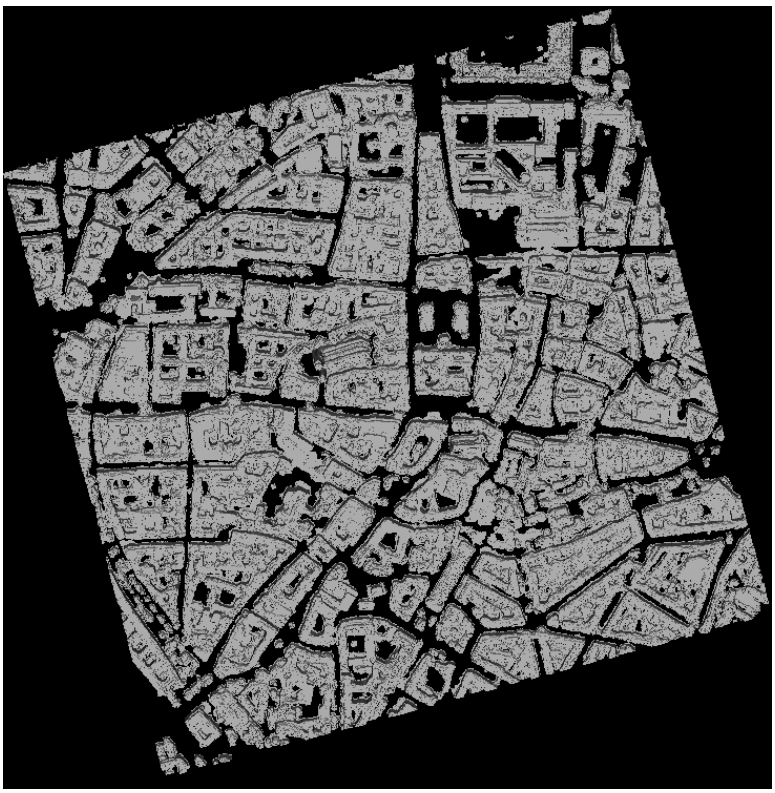


Figure 3-8. Example of simulated optical image.

3.3 Data Preprocessing (2) and Fully Convolutional Network Application

This section includes data preprocessing in A), parameter calculation in B) and the fully convolutional network implement in the other parts.

A). Data preprocessing for FCN

Data preprocessing (2) contains four parts: water label generation (for optical case only), incorrectly labeled data removing, image cutting and dataset organization.

Water label is added for optical image with a classic method, NDWI. Like NDVI, NDWI means normalized differential water index. The calculation method is shown in formula (3.1).

$$\mathrm{NDWI} = \frac{\mathrm{G} - \mathrm{NIR}}{\mathrm{G} + \mathrm{NIR}} \qquad \text{formula (3.1)}$$

Bad data removing is finished by manual selecting. This is because errors from DSM leads wrong labels for image. There are two main error sources: homogeneous areas and clouds. In homogeneous area, the dense stereo matching may fall. For example, the DSM data used in this work is produced with semi-global matching (SGM)[12]. SGM algorithm is based on radiometric differences between stereo image pairs, so that not sensitive in homogeneous areas. Those mismatching points cause elevation anomaly in DSM. For clouds on image, the elevation will be very high and looks like a high light point on DSM image. Those abnormal parts in the dataset should be manually removed. Some examples of errors are shown in Figure 3-9.

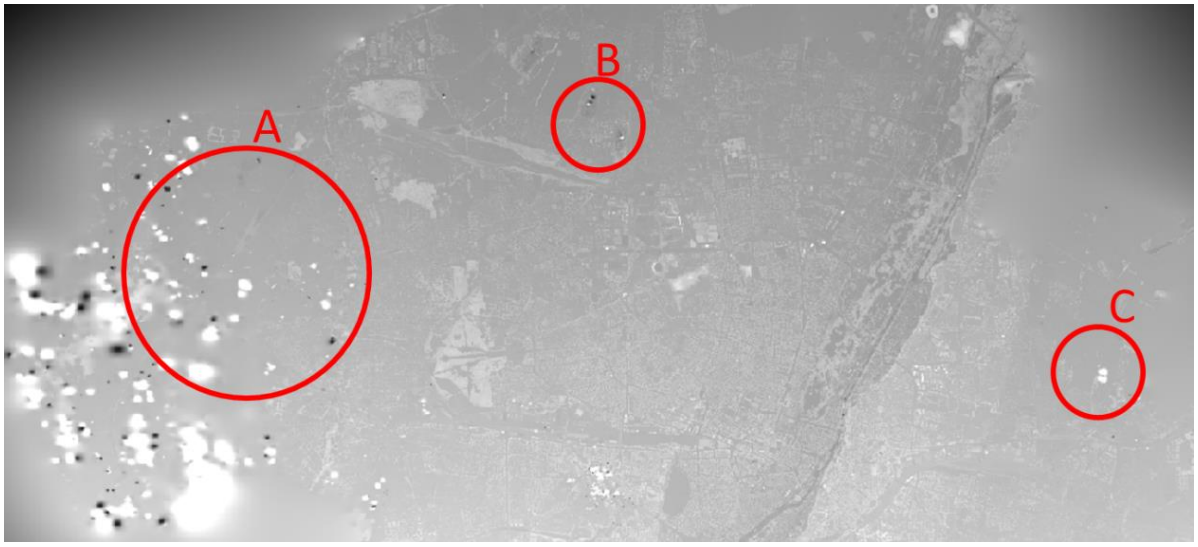


Figure 3-9. DSM examples

For areas like A, B and C in Figure 3-9, data should be removed. In A and C, the elevation is influenced by cloud. In B the homogeneous area like lake may cause error in dense matching and cause anomaly in elevation.

Image cutting is to remove the edge of the marked image data according to the coverage area of the DSM. There are two reasons: one reason is to avoid invalid areas without elevation information or elevation errors, another reason is the coordinates of pixels on mask images may be different up to half of ground sampling distance (GSD).

The simulated image area (same area as labelled image), DSM area and the effective cut-out area are shown in Figure 3-10. In both optical and SAR case, with perspective view, the simulated image and the mask layers cover a larger area than DSM. Some pixels on the edge of simulated image exceeds DSM area. But not all pixels on this part are labelled, because the information for some edge pixels may be on another DSM patch. So, the edge area should be deleted to avoid invalid data. To simplify the selection of overlapping areas, the inside 512 * 512-pixel part of DSM are chosen as valid area. “GDAL” is used to cut image with area for same coordinates. This size will fit the neural network too.

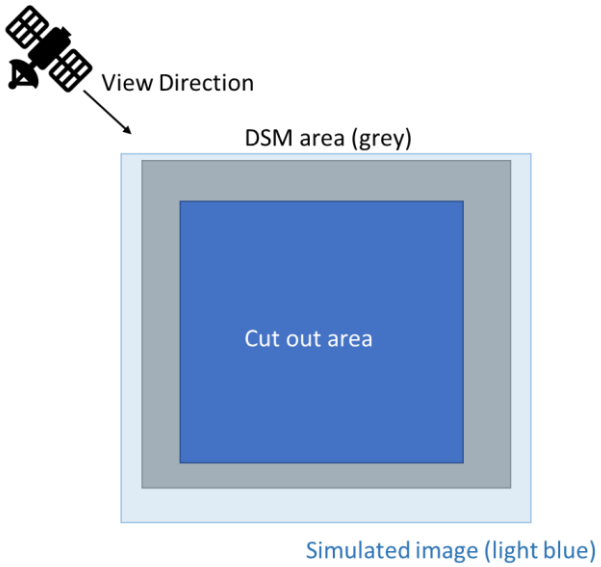


Figure 3-10. Cutting out area of labels and imagery

In the last step, original images and label images should be extracted from SimGeoI’s working folders and stored in 2 folders for the data loading in neural network training. The examples of optical and SAR labels are shown in Figure 3-11 and Figure 3-12.

For optical labels, like in Figure 3-11, brown indicates building, green means vegetation, dark green marks ground and blue stands for water area.

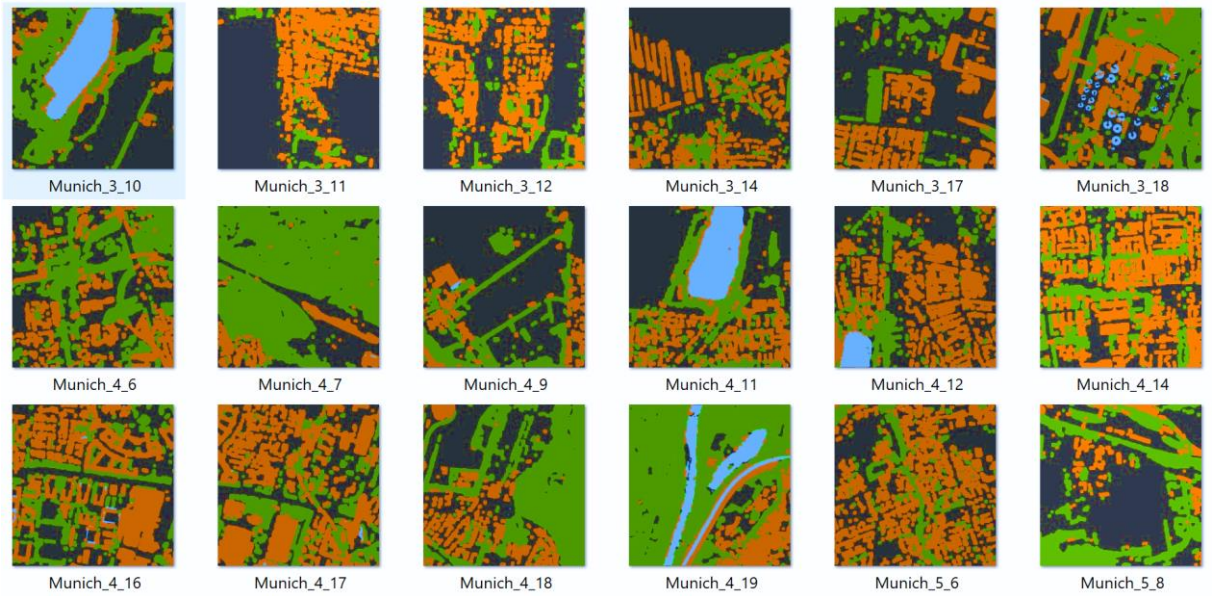


Figure 3-11. Examples of optical image label dataset

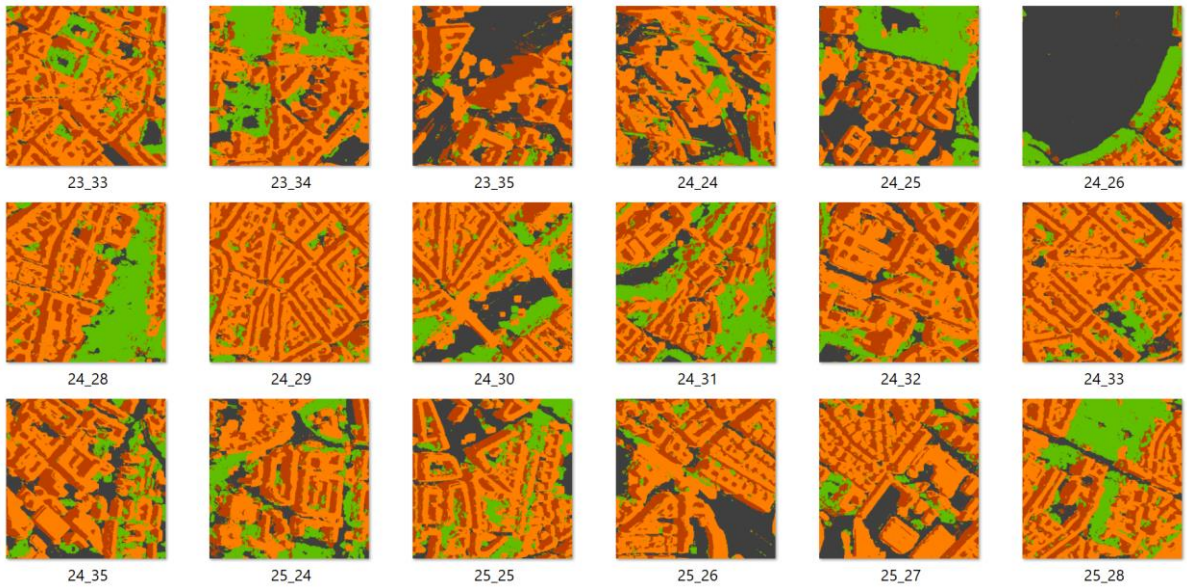


Figure 3-12. Examples of SAR label dataset

For SAR labels, like in Figure 3-12, light brown indicates layover caused by buildings, dark brown indicates shadow on SAR image (shadow caused by elevation of buildings), green indicates vegetations and grey stands for ground and other classes.

Lists of images for training, verification and testing should be defined. Around 75% of the images are used as training data, 20% of the images are used as validation data, and 5% of the images are used as testing and visualization.

An example of images is given in Figure 3-13 to show the different data sources and labels of optical and SAR data. Optical images used here have a GSD of 2m and an example of optical

image is displayed as RGB mode and shown in Figure 3-14. The corresponding label image for optical image is shown in Figure 3-15. SAR data is with GSD as 1m and the covering area is the south-east part of optical image area. An example of SAR image is shown in Figure 3-16. The corresponding label is shown in Figure 3-17.

Figure 3-13. Example of DSM as input of batch processing of SimGeoI

Figure 3-14. Example of optical image in dataset

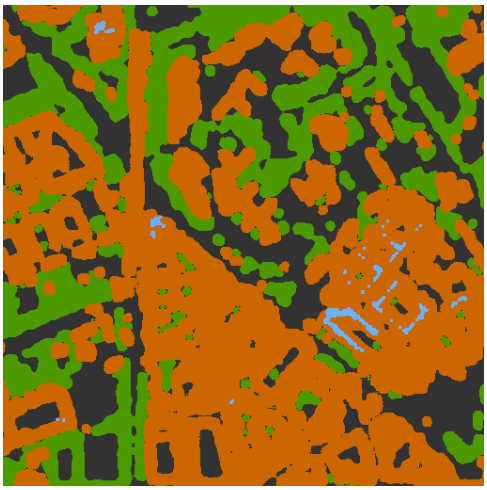


Figure 3-15. Example of label image for optical image

Figure 3-16. Example of SAR image

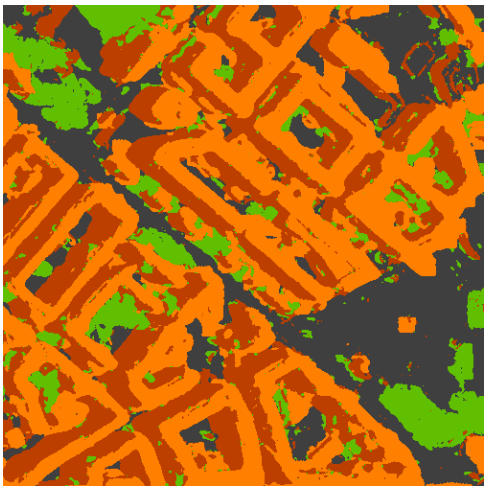


Figure 3-17. Example of label image for SAR image

B). Normalization parameters calculation

In the training and testing of neural networks, data must be normalized before input into the networks.

There are some methods for normalization, like maximum-minimum normalization and standard score normalization. Max-Min normalization compress the maximum value in the data set as "1", the minimum value as "0", and other values are in the range of [0,1]. Considering that the remote sensing image values are usually close to normal distribution, it is more appropriate to normalize data by the standard score.

The average value is calculated by averaging all pixel values in the data set. The average and original values in the image are used to calculate the standard deviation. For optical images with 3 channels, there are 3 mean values and 3 deviations. For SAR image, there are only one mean value and one deviation.

C). Parameter input with YAML file

YAML means Human-readable data serialization language. It is usually used for configuration files and applications for storing or transferring data. In this work, Python-style indentation is used to indicate nesting and makes the parameter input easier when running a Python program with a lot of input parameters.[13]

D). Data loader for our dataset

A data loader is built to fit some characteristics of remote sensing data.

The normalization in data loader is done with standard score normalization mentioned in B). Formula (3.2) illustrates the calculation of standard score normalization with mean value and standard deviation.

$$I_{norm} = (I_{orig} - mean)/Std \qquad \text{Formula (3.2)}$$

Where I_{norm} is normalized grey value in one channel, $I_{orig}$is original.

In image loading part, besides the general setup like batch size, input and output size, our loader can select 3 channels from multi-spectral image. For SAR data, the intensity value is copied to the other 2 channels to fit the network structure.

If input images are in different size, the parameter "drop_last" should be added. If the input images are in different and parameter named "image size" should be a given size but not "same".

In label image loading part, some input classes could be merged. Some classes contain too few samples, like unclassified class and shadow class for near nadir optical image. The lack of training sample of some classes will lead low accuracy and makes no sense.

E). Train FCN

There are several basic frames for FCN. In this thesis, VGG16 is used as the basic network[14]. The structure of VGG16 is shown in Figure 3-18. VGG was introduced by Simonyan and Zisserman in their 2014 paper[14], and means Very Deep Convolutional Networks for Large Scale Image Recognition. VGG16 has 16 weight layers (convolutional layers) in the network, which are colored in blue in Figure 3-18.

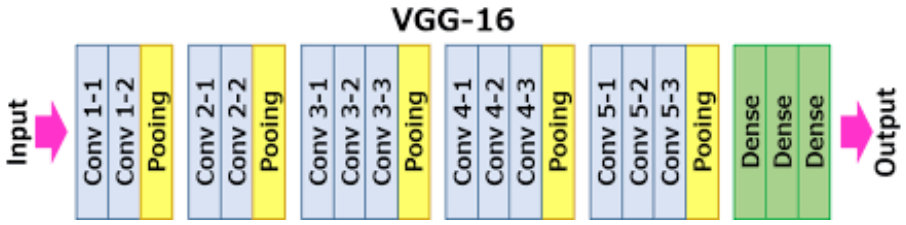


Figure 3-18.VGG16[14]

Then the fully connected (FC) layers in the deeper part, marked as green color, should be modified to fully convolutional layers.

To mark each pixel on input images, deconvolution should be done by combining the feature map from fully convolutional layers with shallower layers. The test result of FCN with different deconvolution levels is shown in Table 1, based on PASCAL VOC dataset, which is a general image dataset marked with 20 classes.

Table 1. accuracy of FCN with PASCAL VOC dataset[5]

	pixel acc.	mean acc.	mean IU	f.w. IU
FCN-32s-fixed	83.0	59.7	45.4	72.0
FCN-32s	89.1	73.3	59.4	81.4
FCN-16s	90.0	75.7	62.4	83.0
FCN-8s	**90.3**	**75.9**	**62.7**	**83.2**

Concerned the test result with general dataset, FCN-8s is chosen as the deconvolution frame to train our own dataset. FCN-16s is also tested with optical data.

To sum up, the program trains a fine network for satellite image segmentation with data set generated in out project. The output includes a model from training and some sample images.

4 Case Study and Analyze

Optical data and SAR data are tested. Chapter 4.1 gives results of optical data and chapter 4.2 gives results of SAR data. The detail of the test and analysis will be explained in the following part.

The environments used in this project are:

Computer system: Ubuntu 16.04.2 LTS (GNU/Linux 4.4.0-66-generic x86_64)

GPU: NVIDIA TITAN X (PASCAL)

Configuration: python2.7

4.1 Case Study 1: Optical Data in Munich Area

Original input data includes DSM of whole Munich area, 2 orthoimages, and 2 multispectral images.

The area of DSM is shown in Figure 4-1.

The areas of optical and ortho images are shown in Figure 4-2. The whole background in image is the DSM data. Blue and green squares mark the location of two used optical images. Orthoimages and original multi-spectral image cover the same area.

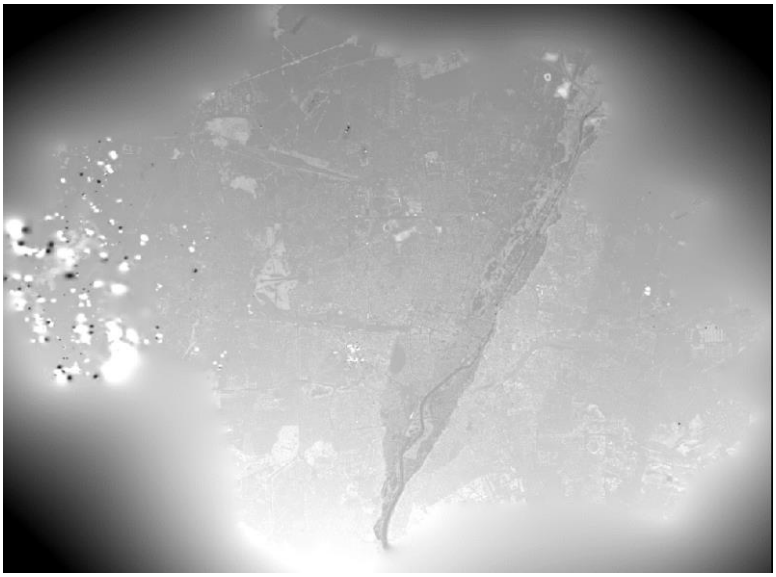


Figure 4-1. DSM in Munich area

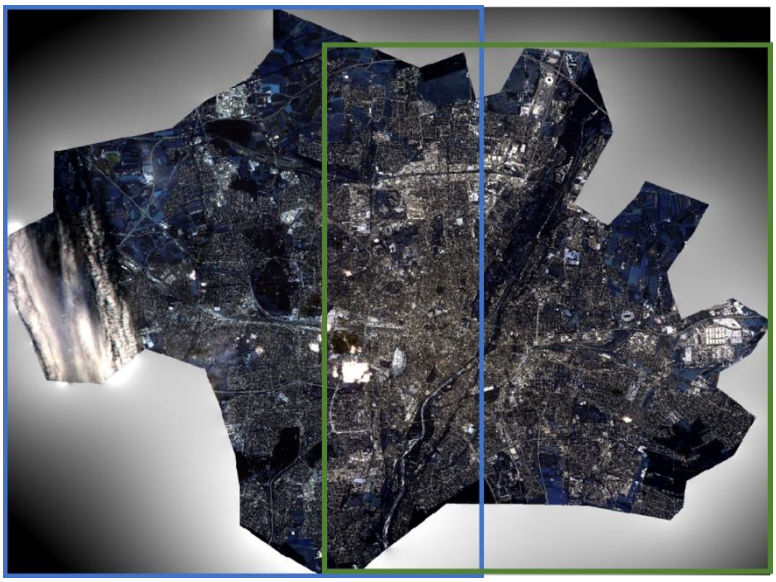


Figure 4-2. Optical images over DSM data

Parameters for SimGeoI are set as:

> Optical mode, with DSM filter, with orthoimage for NDVI, projective plane at height as 600m, illumination angle same as sensor angle.

Three training tests with parameters:

	Test 1	Test 2	Test 3
Model	FCN-16s	FCN-8s	FCN-8s
Learning rate	10^{-4}	10^{-4}	10^{-5}
Iteration	60 000	60 000	60 000

Optical dataset:

> Ground sampling distance: 2m.
>
> Image size: 512*512.
>
> Training dataset: 92 images.
>
> Validation dataset: 22 images.
>
> Test dataset: 6 images.
>
> Class number: 4. Ground, building, vegetation, and water.

Three examples of optical images in the dataset are shown in Figure 4-3, Figure 4-5 and Figure 4-7.

The corresponding reference label images, generated by batch processing of SimGeoI, are shown in Figure 4-4, Figure 4-6 and Figure 4-8.

Figure 4-3. Optical image for area Munich_4_11

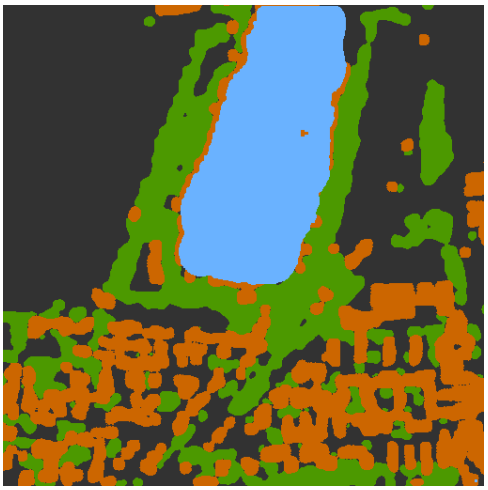


Figure 4-4. Ground truth of optical image for area Munich_4_11

Figure 4-5. Optical image for area Munich_8_17

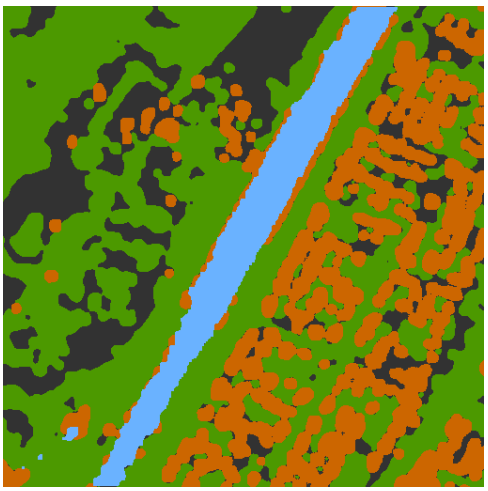


Figure 4-6. Ground truth of optical image for area Munich_8_17

Figure 4-7. Optical image for area Munich_15_13

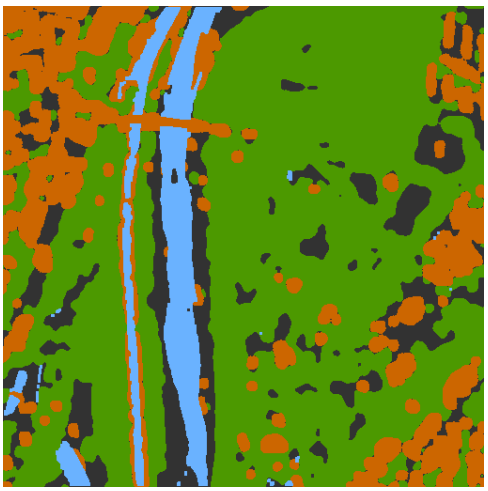


Figure 4-8. Ground truth of optical image for area Munich_15_13

Example test results for Test 1 are shown in Figure 4-9, Figure 4-12 and Figure 4-15.
Example test results for Test 2 are shown in Figure 4-10, Figure 4-13 and Figure 4-16.
Example test results for Test 3 are shown in Figure 4-11, Figure 4-14 and Figure 4-17.

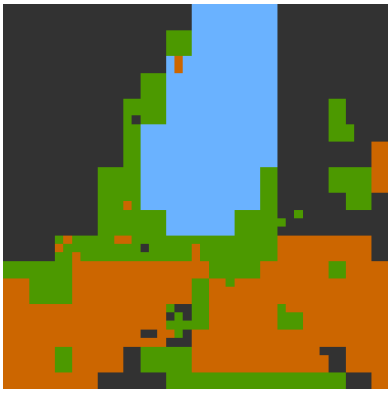


Figure 4-9. Result of Test 1 for area Munich_4_11

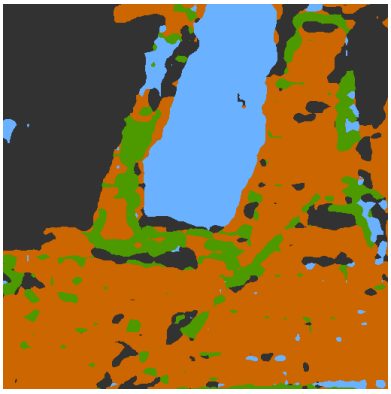


Figure 4-10. Result of Test 2 for area Munich_4_11

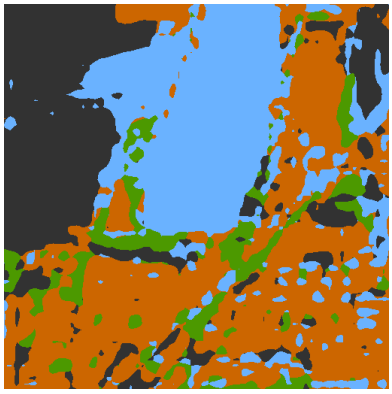


Figure 4-11. Result of Test 3 for area Munich_4_11

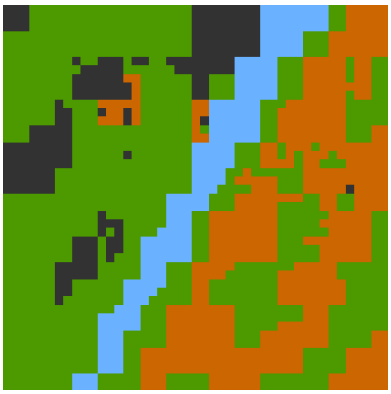


Figure 4-12. Result of Test 1 for area Munich_8_17

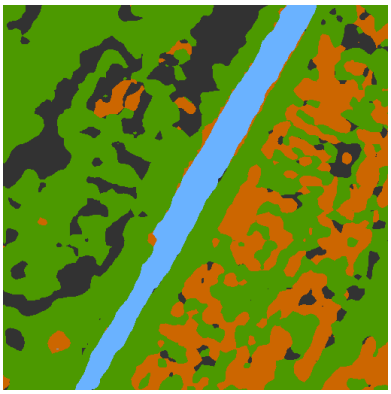


Figure 4-13. Result of Test 2 for area Munich_8_17

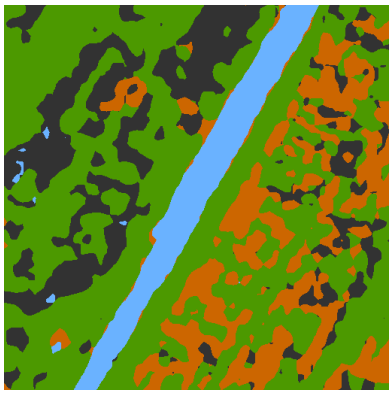


Figure 4-14.Result of Test 3 for area Munich_8_17

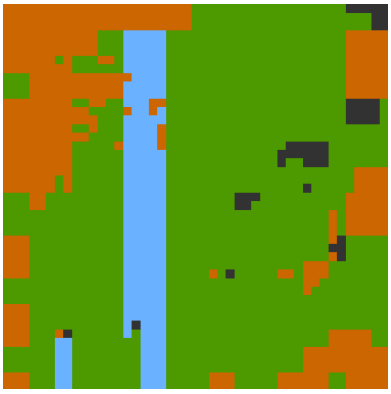


Figure 4-15. Result of Test 1 for area Munich_15_13

Figure 4-16. Result of Test 2 for area Munich_15_13

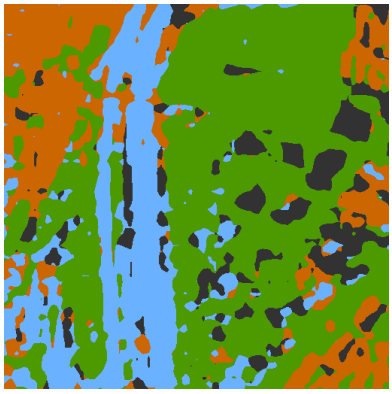


Figure 4-17. Result of Test 3 for area Munich_15_13

Analyze for case study 1:

A). FCN-8 has better results than FCN-16

Comparing the results of Test 1 (Figure 4-9, Figure 4-12 and Figure 4-15) with second and third columns (Figure 4-10, Figure 4-13 and Figure 4-16, Figure 4-11, Figure 4-14 and Figure 4-17), we can easily find that that FCN-8s (Test 2 and Test3) offers significantly better result than FCN-16s (Test 1). Because FCN-16 uses one less polling layer for upsampling than FCN-8, FCN-16 will contain less local information in the predicted image.

In Figure 4-18 and Figure 4-19, the lowest loss value of Test 1 is 0.76, while the lowest loss value of Test 2 is 0.49. The figure also shows the same phenomenon that FCN-8 has better results than FCN-16.

B). Small learning rate may lead to local optimum model.

Comparing then results of Test 2 (Figure 4-10, Figure 4-13 and Figure 4-16) and Test 3 (Figure 4-11, Figure 4-14 and Figure 4-17), we could see that learning rate at 10^{-4} achieves better results than 10^{-5}. The validation loss graphs in Figure 4-19 and Figure 4-20 tell us that a small learning rate could lead to a local minimum but not the global minimum, as the smallest validation loss stays at 0.52 for learning rate as 10^{-5}, but the best validation loss is 0.49 for learning rate as 10^{-4}. So, 10^{-4} is a more suitable learning rate for this dataset.

C). Noise can be compensated by amount of data.

The segmentation label from trained network could give be more reliable than the label generated in this project. Take ground truth from SimGeoI and test result from FCN-8 for area Munich_4_11 as examples, shown in Figure 4-4 and Figure 4-10, the small buildings on east on the lake are not marked in SimGeoI generated ground truth, but well recognized by neural network. Those buildings are not small so are not recognized by SimGeoI. But building roofs have similar features in color space of “buildings” and could be recognized by neural network. With large amount of correct data, some noise could be compensated.

D). Overfitting models have good training loss value but higher validation loss value.

Figure 4-18, Figure 4-19 and Figure 4-20 show the loss values in training with different parameters. The overall trend of training loss is going down continuously, but validation loss decreases firstly, and then, after reaching a dip, starts to increase. The models at lowest validation values are considered as the best model in training. As the training time increases, the model fits training dataset better, but the model could fall into an overfitting case. So, the loss value of validation indicates the truly best model.

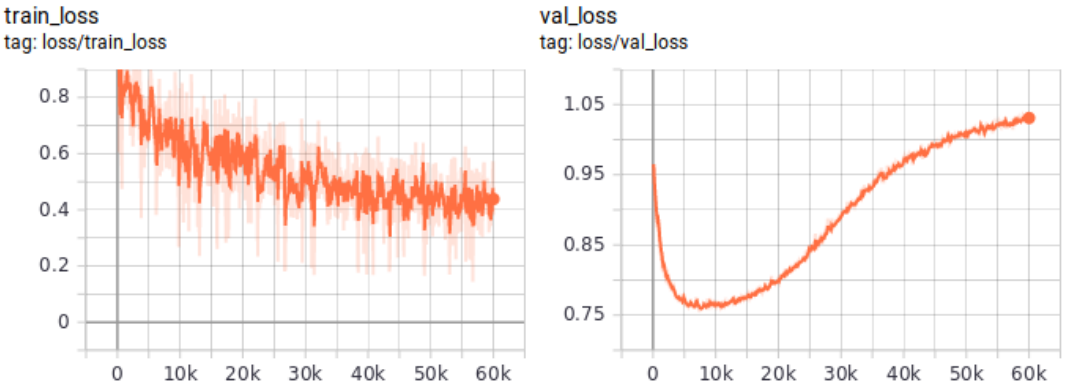


Figure 4-18. Loss value of optical data training: Test 1, fcn-16s, lr=10^{-4}, iter=60k

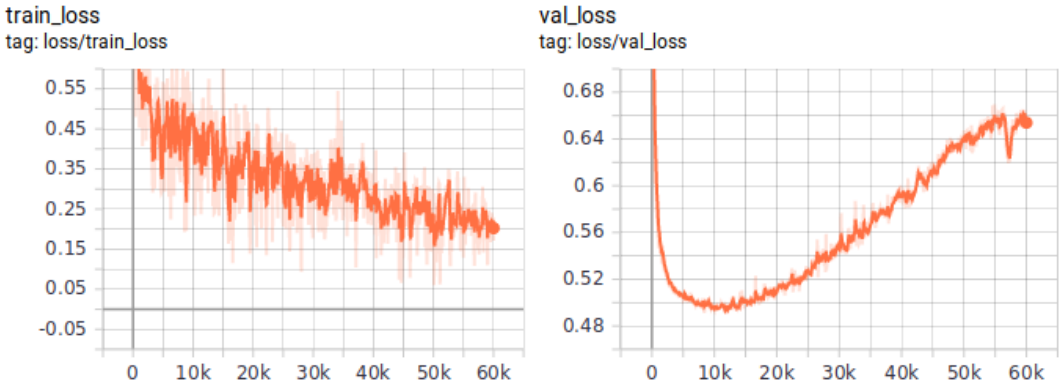


Figure 4-19. Loss value of optical data training: Test 2, fcn-8s, lr=10^{-4}, iter=60k

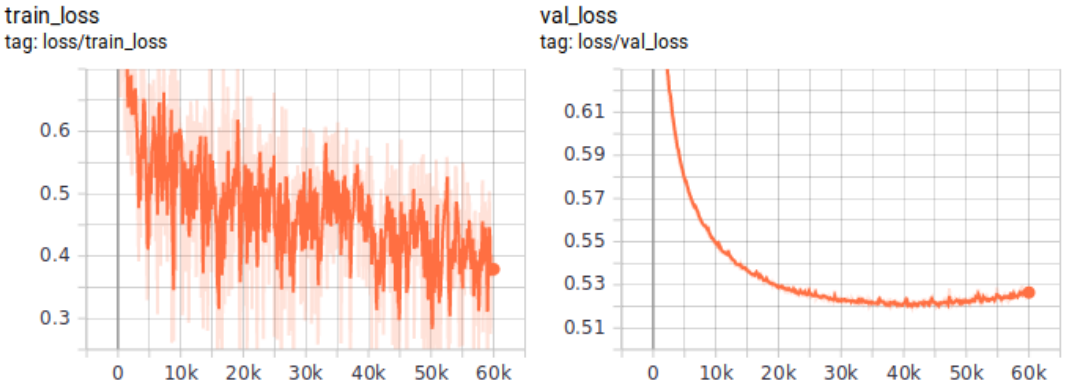


Figure 4-20. Loss value of optical data training: Test 3, fcn-8s, lr=10^{-5}, iter=60k

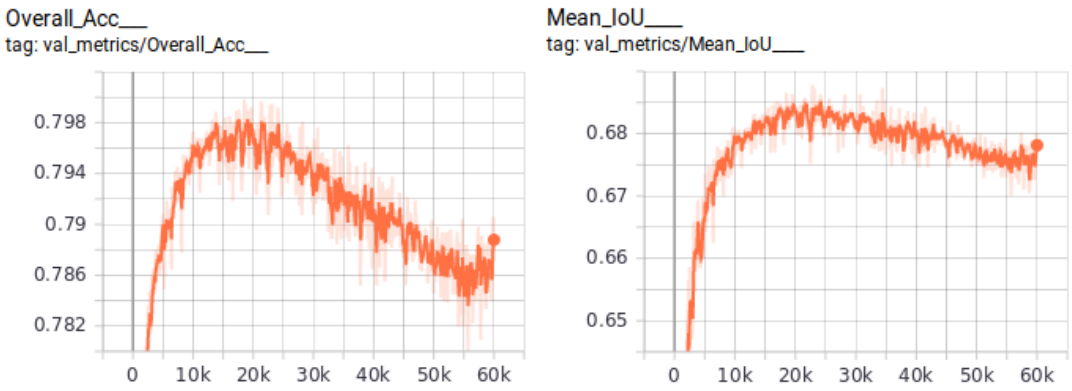


Figure 4-21. Accuracy and IoU in training: Test2, fcn8s, lr=10^{-4}, iter=60k

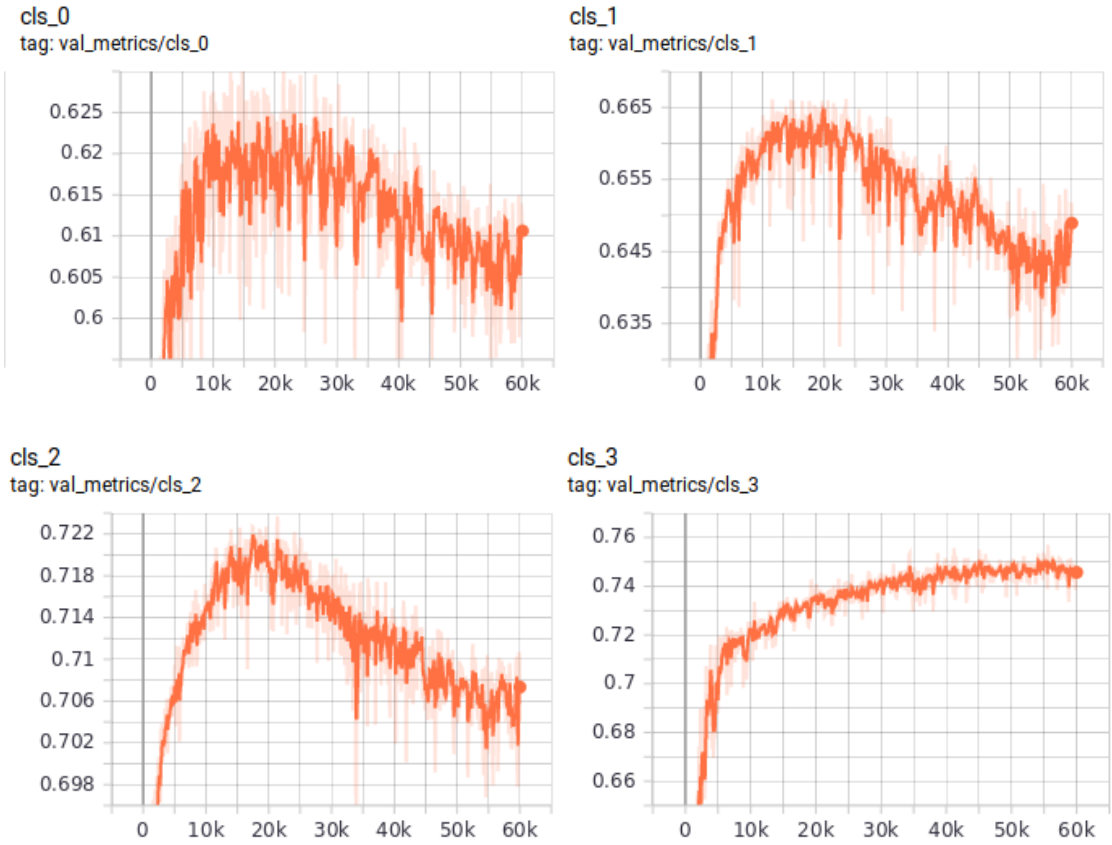


Figure 4-22. Accuracy of each class in training: Test2, fcn8s, lr=10^{-4}, iter=60k

E). The accuracy of each class in test 2 is given in Figure 4-22, where “0” is ground, “1” is building, “2” is vegetation and “3” is water. Accuracies of ground, building and vegetation have similar trend, while accuracy water class is abnormal. This may because of the sample amount and quality of water class.

4.2 Case Study 2: SAR Data in Munich Area

One example of the network training with SAR images and the mask images generated in this project.

Original input data:

DSM of whole Munich area, two SAR images. Their areas are shown in Figure 4-23. The whole background in image is the DSM data. Blue square and green square are area of these two SAR intensity images.

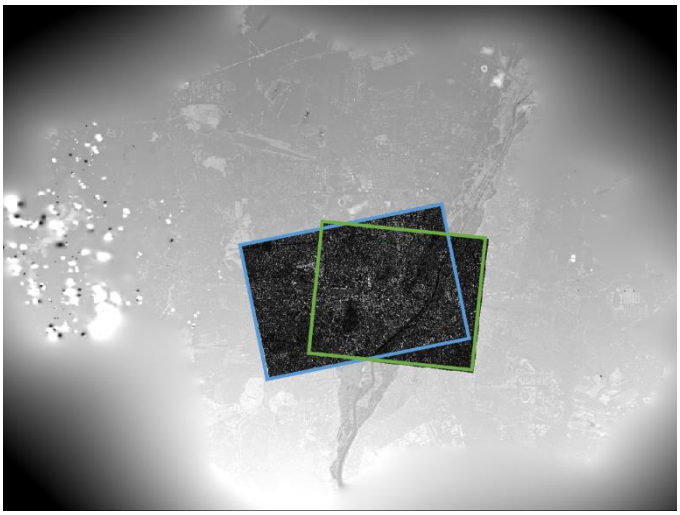


Figure 4-23. SAR intensity images over DSM

Parameters for SimGeoI:

SAR mode, with DSM filter, with orthoimage for NDVI, projective plane defined in metadata of original SAR image, illumination angle same as sensor angle.

Input parameters for FCN:

Model: fcn-8s. Learning rate: 10^{-4}. Iteration: 20 000

SAR dataset:

Ground sampling distance: 1m.

Image size: 512*512.

Training dataset: 84 images.

Validation dataset: 15 images.

Test dataset: 6 images.

Class number:4. Ground, building, layover (façade) and vegetation.

The example SAR images are shown in Figure 4-24, Figure 4-27 and Figure 4-30.

The corresponding reference label images, generated by batch processing of SimGeoI, are shown in Figure 4-25, Figure 4-28 and Figure 4-31.

The corresponding test results are shown in Figure 4-26, Figure 4-29 and Figure 4-32.

Figure 4-24. SAR image Munich_17_25

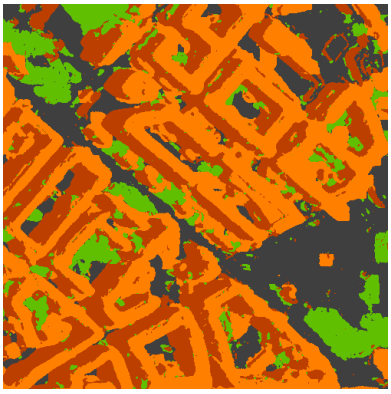


Figure 4-25. Ground truth of SAR image Munich_17_25

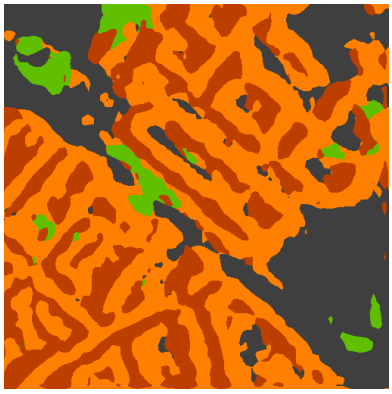


Figure 4-26. Test result: Munich_17_25

Figure 4-27. SAR image Munich_19_29

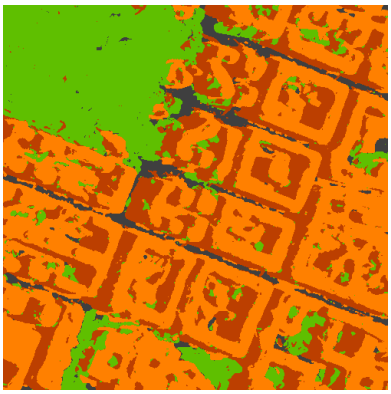


Figure 4-28. Ground truth of SAR image Munich_19_29

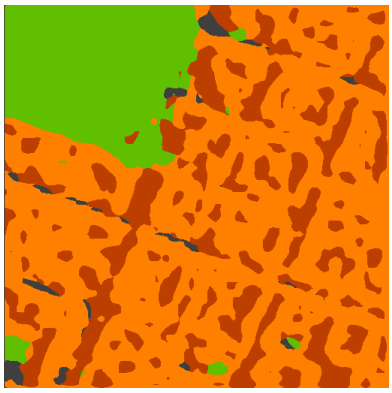


Figure 4-29. Test result: Munich_19_29

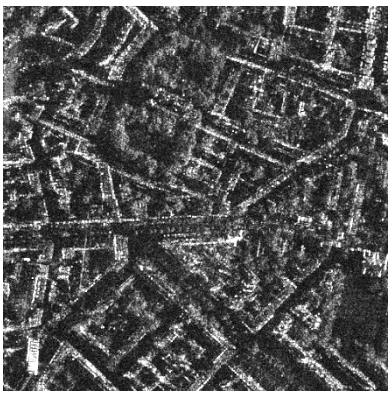


Figure 4-30. SAR image Munich_23_33

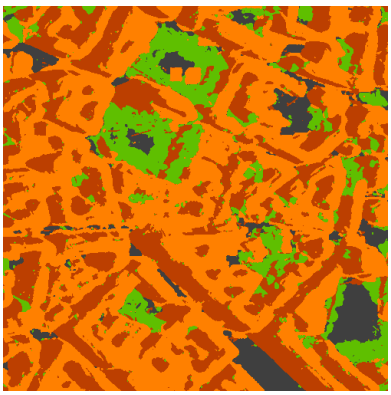


Figure 4-31. Ground truth of SAR image Munich_23_33

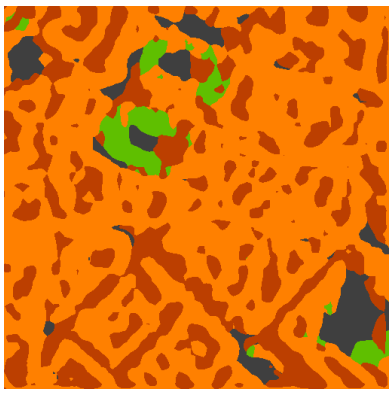


Figure 4-32. Test result: Munich_23_33

Analyze of case study 2:

A). the overall results are very good.

The segmentation of SAR data is regard as a trying because the single channel of intensity contains limited information and extremely challenging for manual labelling. However, the segmentation results are remarkable: vegetation, ground, building and layover (façade) class are segmented well, and very similar to the ground truth.

B). Noise can be compensated by amount of data too.

Although building class and layover (façade) class are a little bit mixed in reference label images, due to DSM errors, the best accuracy of SAR image segmentation in this test still reaches 0.666.

C). The loss values have stronger fluctuation than loss values of optical data. And the accuracy of SAR dataset is worse than optical dataset.

This could be caused by the feature of SAR data, like “noisy” data distribution. Or the limited information from data source: only intensity data is used.

The loss graphs are shown in Figure 4-33, and the accuracy graphs are shown in Figure 4-34.

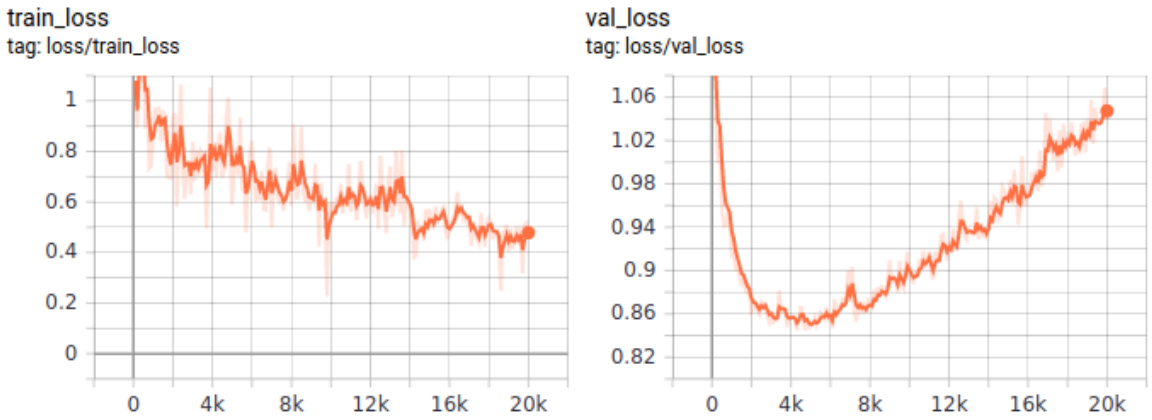


Figure 4-33. Loss of SAR image training

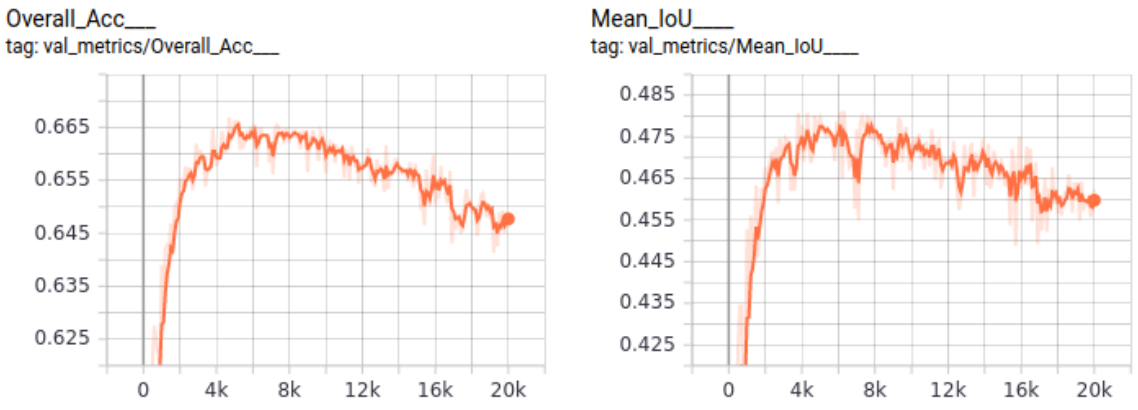


Figure 4-34. Accuracy and IoU of SAR training

D). The building class (cls_1 in Figure 4-35) achieves lowest accuracy, compared with ground (cls_0 in Figure 4-35), layover (cls_2 in Figure 4-35) and vegetation (cls_3 in Figure 4-35).

The accuracies of each class are shown in Figure 4-35.

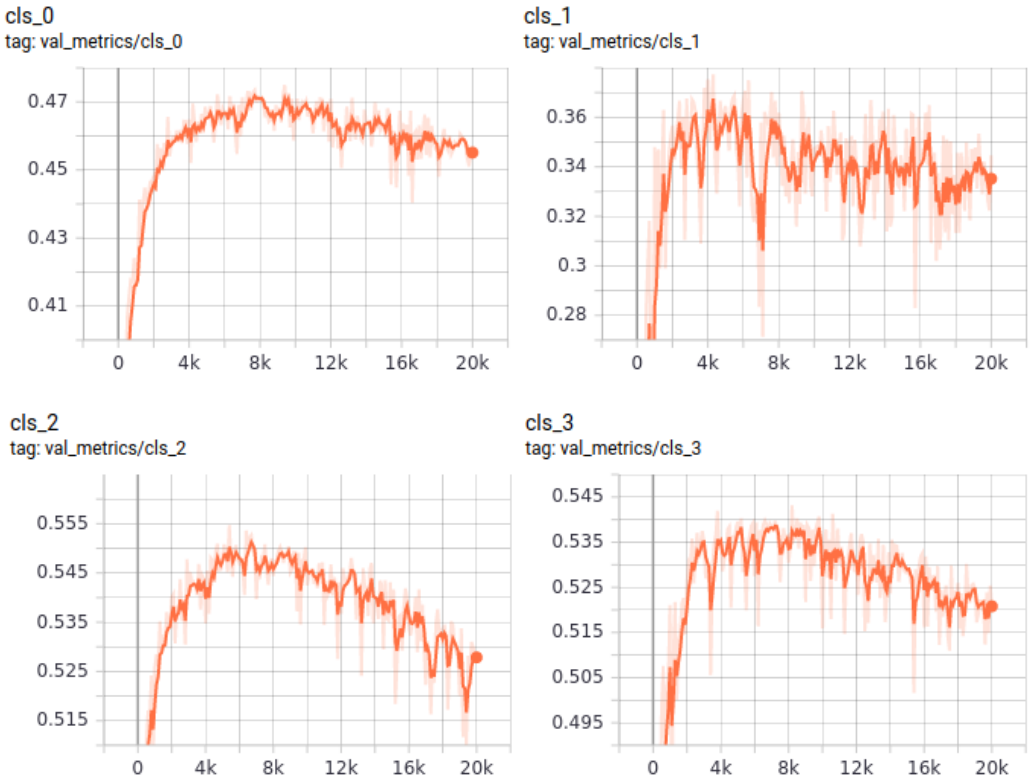


Figure 4-35. Accuracies of each class in SAR training

The overall conclusion is that the SimGeoI generated label images for both optical and SAR data are suitable for semantic segmentation. This batch processing of SimGeoI could be used to produce ground truth labels.

5 Conclusion and Outlook

This thesis finished a batch processing of SimGeoI, developed a tool to connect labels generated by SimGeoI with neural network frame and trained fully convolutional network models with those data. The test scene in Munich area achieves a good result for optical image, and the first try for SAR data.

Although result shows some curbs, the overall summary gives a positive attitude. The test results of both optical and SAR data indicate that the labels generated by SimGeoI could be used for semantic segmentation. The batch processing function and tools to preprocess data implement a processing chain that producing training dataset. The information could be transferred from elevation data to image labels, which is meaningful. Especially for SAR data, this label generator is remarkable, because of the difficulty of manual interpretation.

To summarize, this thesis gives possibility to generate ground truth labels from DSM data, with good recognition of elevated objects, and confirms that SimGeoI generated data could be used in semantic segmentation,

Due to the time limitation of a master thesis, and the Corona pandemic, this work is not perfectly done. Restrictions include:

1). The output label accuracy is limited by the accuracy of DSM.

2). The spatial resolution of DSM and image may be different, and the resolution of output data depends on the lowest resolution.

3). The reference coordinates of generated mask image are usually not same as reference coordinates of original optical image or SAR image. Resampling need to be done for same coordinates. Some pixels are not completely consistent.

4). The water class in only added in optical case. Due to the DSM data in water is always wrong, the water area is marked as building by SimGeoI. It is not practice to calculate NDWI with orthoimage and map it with DSM data, because the water area will be mapped to other area with wrong DSM. Water areas in optical case are recognized with original optical images, after SimGeoI processing.

5). Neural network result may be restricted by single network model used in this thesis, because this thesis aims to certify that labels generated by SimGeoI could be used for neural network for task of semantic segmentation.

For further improvement, there are some possible proposals:

1). Use more reliable DSM data.

2). Use data with higher spatial resolution.

3). Besides intensity of SAR data, add amplitude, phase, or coherence as data source.

4). Recognize water class within SimGeoI based on orthoimage, with some strategies, to both optical and SAR data.

5). Try more network models.

6). Reduce loss of local information in network, like use smaller image size.

7). Maybe dataset generated with elevation information could be used for height image training.

Reference

[1] Stefan Auer, Isabel Hornig, Michael Schmitt. Simulation-Based Interpretation and Alignment of High-Resolution Optical and SAR Images. Nov 2017

[2] Stuart J. Russell, Peter Norvig. Artificial Intelligence: A Modern Approach. 2010, Third Edition, Prentice Hall ISBN 9780136042594.

[3] Mehryar Mohri, Afshin Rostamizadeh, Ameet Talwalkar. Foundations of Machine Learning. 2012, The MIT Press ISBN 9780262018258.

[4] Hinton, Geoffrey, Sejnowski, Terrence. Unsupervised Learning: Foundations of Neural Computation. 1999, MIT Press ISBN 978-0262581684.

[5] J. Long, E. Shelhamer, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. Mar 2015.

[6] https://earth.esa.int/web/eoportal/satellite-missions/t/terrasar-x

[7] Murphy, P. Kevin. Machine Learning: A Probabilistic Perspective. (2012) Cambridge: MIT Press. p. 247.

[8] https://en.wikipedia.org/wiki/Learning_rate

[9] https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection

[10] https://en.wikipedia.org/wiki/Overfitting

[11] https://en.wikipedia.org/wiki/Orthophoto

[12] H. Hirschmüller. Semi-Global Matching. 2005.

[13] https://yaml.org/

[14] K. Simonyan, A. Zisserman. Very deep convolutional networks for large-scale image recognition. 2014.

[15] https://neurohive.io/en/popular-networks/vgg16/