Semantic Segmentation of RGB-Z Aerial Imagery Using Convolutional Neural Networks

MSc Geomatics Thesis Proposal

Amber Mulder

First supervisor: Balázs Dukai
Second supervisor: Ravi Peters
READAR supervisors: Sven Briels & Jean-Michel Renders

January 3, 2020

1 Introduction

Semantic segmentation can be described as the process in which every pixel of an image is associated with a class label. This process allows for the division of an image into meaningful, non-overlapping parts (Zhu et al., 2016). Automatic semantic segmentation of aerial imagery can be useful for many different types of applications that currently require extensive manual work by experts, which is time-consuming and costly. Applications are present in fields as mapping of land cover, object detection, land-use analysis and change detection (Saito et al., 2016; Kampffmeyer et al., 2016).

In example, ensuring high quality for topographic maps, such as the large scale dataset Basisregistratie Grootschalige Topografie (BGT) [1] of The Netherlands, requires regular updating of mutations. Current change detection techniques are often based on visual comparison of aerial images by experts. Automatic semantic segmentation can automate part of this process by semantically segmenting images of the same location of two different years. The segmentation results can be subject to simple overlay operations in order to detect mutations.

Even though semantic segmentation is researched by many, the topic remains challenging. The constantly increasing spatial and spectral resolution of remotely sensed imagery can be considered as one of the main difficulties. This high resolution has the benefit of being able to capture small details such as small objects. However, it also complicates the semantic segmentation process by introducing higher imbalances in class-distributions, large variance between classes and small differences within each class (Wang et al., 2016; Yuan et al., 2016).

In the last couple of years, the deep learning revolution has stimulated the use of deep architectures, usually Convolutional Neural Networks (CNNs), to successfully tackle general semantic segmentation problems (i.e. Long et al. (2014); Hariharan et al. (2014); Feng Ning et al. (2005)), including remote sensing related ones (i.e. Kampffmeyer et al. (2016); Paisitkriangkrai et al. (2015); Saito et al. (2016). When well trained, these algorithms act as non-linear functions that have the ability to take an image as an input, and provide a segmented version of this image as an output. CNNs have shown to outperform traditional computer vision and machine learning approaches in terms of accuracy and in some cases efficiency (Garcia-Garcia et al., 2017).

The focus in semantic segmentation in the last couple of years has been on two-dimensional imagery. However, the fast growing technological development in acquiring and analyzing 2.5D or 3D data, allows for the introduction of a new dimension next to RGB information (Garcia-Garcia et al., 2017; Qin et al., 2016). It is believed that the added 2.5D or 3D information, often in the form of depth maps, digital elevation model (DEM) or point cloud, has the ability to improve semantic segmentation results (Qi et al., 2017; Qin et al., 2016). Even though inclusion of three dimensional information in semantic segmentation problems has been researched for regular imagery (i.e. Qi et al. (2017); Gupta et al. (2014); Couprie et al. (2013)), the inclusion of pixel level height information to improve the quality of semantic segmentation is less represented in the current literature. Therefore, this proposed research aims to examine the added value of height information, Z, to semantic segmentation of aerial imagery. The following research question is addressed; to what extent can Convolutional Neural Networks be used for automatic semantic segmentation of RGB-Z aerial imagery?

[1] https://www.pdok.nl/introductie/-/article/basisregistratie-grootschalige-topografie-bgt-

For this research true ortho imagery with a resolution of 10 cm will be used, which is developed by the company READAR [2]. True ortho imagery is aerial imagery that is corrected for relief displacement (Sheng et al., 2003). Furthermore, the fourth input band provided to the networks, containing height information, will be derived from a digital surface model (DSM) which is also generated by READAR.

This proposal is structured as follows; first, a short discussion on related research will be provided. Hereafter, the research questions will be given, followed by the methodology, preliminary results and a time planning for the project. Finally, an overview will be provided on the to be used tools and datasets.

2 Related work

Due to the high amount of successful initiatives of the use of CNNs for semantic segmentation, together with the far from matured state of these techniques, this field of research is growing rapidly (Garcia-Garcia et al., 2017). This multitude of newly produced literature makes the task of keeping up with the most recent developments challenging. Nevertheless, after giving a brief description on deep learning and CNNs in general, this section aims to provide a short overview of relevant related studies to this research proposal.

2.1 Deep learning

Deep learning comprises a class of techniques in the field of machine learning in which computational models, called neural networks, consisting of multiple processing layers, learn to represent data with different levels of abstraction (LeCun et al., 2015). A neural network consist of an input layer, an output layer and one to many hidden layers. Each layer of a neural network consists of multiple neurons (figures 1 and 2). Each neuron can be seen as a feature, and is a mathematical operation which has its own learnable weights and biases. Input is provided to each neuron, which is multiplied by it's corresponding weight and then summed. Hereafter, the function corresponding to the neuron, called the activation function, is applied to the result. The output of this function is passed on to neurons in consecutive layers (Hush and Horne, 1993). Key to deep learning is that these layers of features used in the network are not provided, but are decided on by the network itself, using general-purpose learning (LeCun et al., 2015).

At each layer in the network, the input data is transformed to a higher level of abstraction. Very complex functions can be learned when enough of these transformations are executed. In the case of classification, representations of higher levels of abstraction allow for the elimination of unimportant variations and amplification of parts of the input that are important for distinction (LeCun et al., 2015).

2.2 CNNs

CNNs are a specific type of neural networks that are often used for semantic segmentation problems. When compared to ordinary neural networks, CNNs contain special types of layers allowing for specific functionality related to the computation of convolution, up-sampling

[2] https://readar.com/

[3] Image retrieved from: https://www.rsipvision.com/exploring-deep-learning/

[4] Image retrieved from: https://summer-story.tistory.com/6

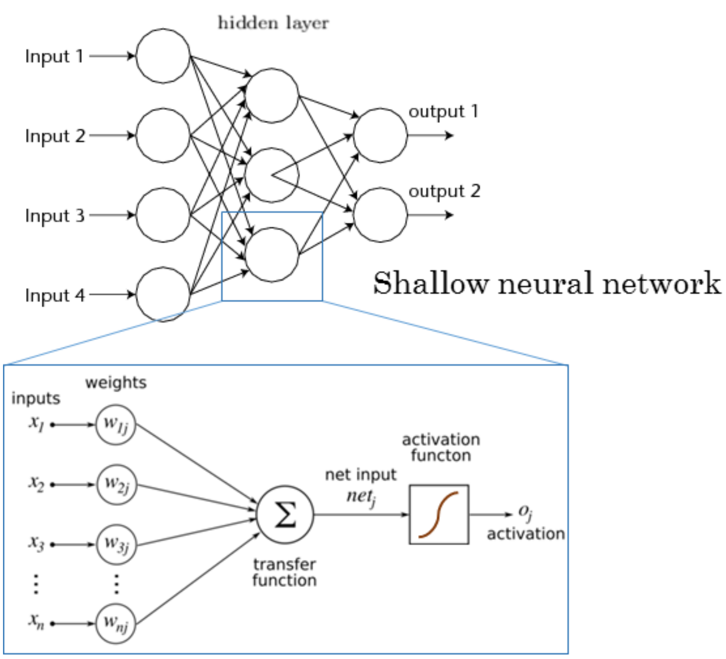

Figure 1: A diagram of a simple (shallow) neural network, containing only one hidden layer[3].

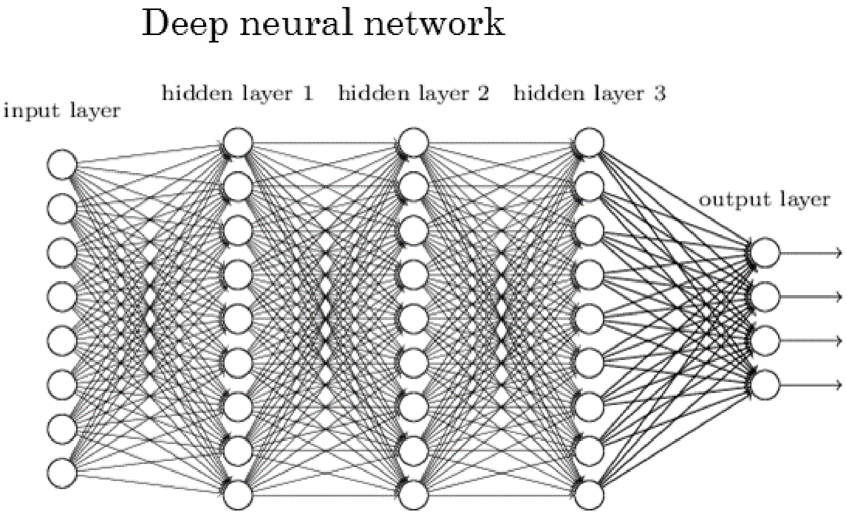

Figure 2: A diagram of a deep neural network, containing several hidden layers[4].

and down-sampling (Liu et al., 2017). This allows the algorithms to generate useful low-dimensional representations, while preserving spatial properties (Shorten and Khoshgoftaar, 2019).

When the input is an RGB image, the input data to a CNN is provided as three 2D arrays, giving pixel intensities of the three channels (LeCun et al., 2015). In semantic segmentation, one is not only interested in classification, but also in the projection of the classification onto pixel space. A general network architecture for semantic segmentation consists of an encoder network, connected to a decoder network. The encoder fulfills the role of classification, while the decoder ensures dense classification by projecting the classification onto pixel space (Shah, 2017). Four basic types of layers can be distinguished that are used in CNNs for semantic segmentation, namely;

- **Convolutional Layer**: Exists of simple filters which contain learnable parameters. Each neuron in the layer searches for a specific pattern. As the aim is to search for the same patterns throughout the whole input, the learnable weights and biases of the neurons of the output are shared (Liu et al., 2017).
- **Transposed Convolutional Layer**: Also referred to as deconvolution layer. These layers allow for upsampling of the input: the dimensions of the input are increased. The parameters can be based on simple bilinear interpolation or they can be learned (Long et al., 2014).
- **Non-linear Function Layer**: Often present after a convolutional layer. This type of layer adds non-linearity to the network, by introducing in example the Sigmoid function or the rectified linear unit (ReLU) (Glorot et al., 2011).
- **Spatial Pooling Layer**: Uses a filter to reduces the size of the input. Functions commonly used are max, sum and mean (Saxe et al., 2011).

Two types of approaches are distinguishable in current researches on semantic segmentation; patch-based methods and pixel-based methods. Patch-based methods use a small window to construct a label for each pixel independently. Consequently, the labels assigned to each pixel are only based on its near surrounding pixels (Sermanet et al., 2013). Pixel-based methods have a different approach by inferring the labels for all of the pixels at the same time. When semantically segmenting remote sensing imagery, this type of methods have outperformed patch-based methods (see Kampffmeyer et al. (2016) and Volpi and Tuia (2016)). Different CNNs differ in their architecture. Figure 3 displays the architecture of three well-known pixel-based methods.

2.3 Semantic segmentation of aerial imagery using CNNs

In their research, Saito et al. (2016) used a five-layered CNN to automatically detect objects from aerial imagery. Their goal was to generate a multi-channel label output from the input image, with one channel per class. This research showed that predicting classes simultaneously can lead to a higher accuracy than when each class is predicted separately. In addition, they proposed a new output function called 'channel-wise inhibited softmax' (CIS) which further improved the results. The recall, which can be described as the ratio of the detected pixels to the true pixels, were for the building class 0.9418, 0.9539 and 0.9686 for the single-channel model, the multi-channel model and the multi-channel model with CIS respectively. For the road class these values were 0.8507, 0.8701 and 0.9020. Even though promising results were obtained, only the classes roads and buildings were included, the ground truth pixel size differed from the dimensions of the input imagery, and no height information was incorporated.

Kampffmeyer et al. (2016) and Liu et al. (2017) did include height information in their researches on semantic segmentation of aerial imagery. In their research, Kampffmeyer and his colleagues used six classes, namely 'impervious surfaces', 'building', 'low vegetation', 'tree', 'car', 'clutter/ background'. Kampffmeyer showed that their implementation in which three architectures were combined (PB, FCN and FCN-MFB), performed best when compared to the individual architecture performances, when considering accuracy for small objects while still maintaining a high overall accuracy. This implementation gave an overall accuracy of 87.03% (with 93.92% accuracy for the class building). For assessing the quality, ground truth data for which the class boundaries were eroded to lower the effect of class boundaries, was used. Liu and his colleagues also designed their own network architecture which integrates

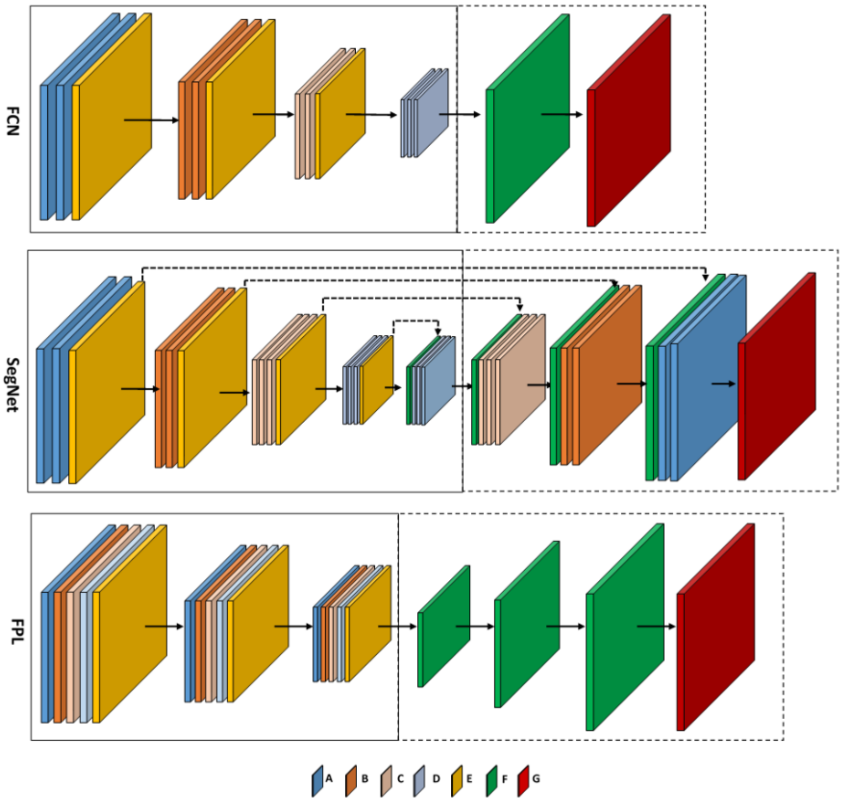

Figure 3: The design of the fully-convolutional network (FCN)(Kampffmeyer et al., 2016), SegNet (Badrinarayanan et al., 2017) and full patch labeling (FPL) (Volpi and Tuia, 2016). A, B, C and D are convolutional layers; E is a pooling layer; F is a transposed convolutional layer or unpooling layer (in SegNet); G is a loss layer. Figure extracted from Liu et al. (2017).

an ’inception’ and ’residual’ module in the conventional encoder-decoder paradigm. The inception module comprises the collection of filters of different sizes into one layer, enabling the gathering of information from receptive areas of different scales. The residual module allows to directly feed forward information from the encoder to the decoder. Using the same dataset and classes as in the research of Kampffmeyer, Liu and his colleagues retrieved an overall accuracy of 88.82% (with 94.67% accuracy for the building class). When using the non-eroded ground truth data, the overall accuracy was 85.39% (92.34% accuracy for the building class).

Both these researches did not examine the added value of the included height information, they solely focused on optimizing the semantic segmentation while using the extra band. Even though the objects of interests in both these researches did not completely overlap with the objects of interests in this proposed research, examination of their implementation, or even using it as a starting point would be interesting. Unfortunately, the corresponding codes of the implementations could not be discovered. However, findings of these papers will be taken into consideration while executing this research. In example, one of the conclusions from Kampffmeyer and his colleagues that pixel-based approaches outperformed patch-based approached for semantic segmentation of aerial imagery, will be taken into account when selecting architectures for this research.

Even though deviating classes are used, achieved accuracies for RGB-Z imagery in the research corresponding to this proposal are expected to be similar to the ones achieved in the researches described in this section. Furthermore, when considering the conclusion of Qi et al. (2017) that 3D information has the ability to improve semantic segmentation results for regular imagery, it is expected that the accuracy retrieved when height information is included to the aerial imagery, will be higher than the accuracy gained when height information is not included.

3 Research questions

The proposed research aims at answering the following question;

To what extent can convolutional neural networks be used for automatic semantic segmentation of RGB-Z aerial imagery?

In order to answer this question, the following sub-questions are specified;

- Which neural network is a suitable starting point for semantic segmentation of aerial RGB-Z imagery?
- To what extent does the addition of the Z dimension improve semantic segmentation results?
- How does resolution influence the quality of the semantic segmentation?
- For which classes is the segmentation most successful; for ’building’, ’road’, ’water’ or ’other’?

4 Methodology

Figure 4 displays the methodology proposed for this research. In the following paragraphs, the steps will be described in more detail.

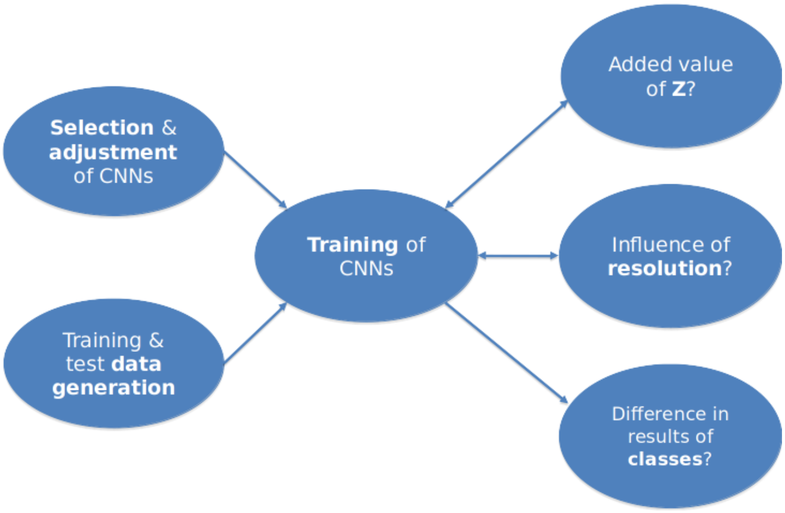
Selection & adjustment of CNNs
Training & test data generation
Training of CNNs
Added value of Z?
Influence of resolution?
Difference in results of classes?

Figure 4: Overview of methodology.

4.1 Selection and adjustment of CNNs

Firstly, a selection is made of existing neural networks that are considered to be suitable for this research. This will be done by examining code implementations corresponding to promising research papers. Encountered networks are considered to be suitable when adherent to the following criteria;

- Shown successful performance of semantic segmentation of any type of imagery (performance mostly based on research papers: promising values for performance measures when compared to other researches)
- Source code available online, without any license restrictions
- Implementation not too complex; adjustments in network structure are (easily) implementable
- Implementation is done in python

As a high number of networks exists, it is aimed to select a few most promising ones. These will be adjusted to fit to the RGB imagery of this research. Depending on the used network this could mean in example incorporating spatial reference information in the segmented output, or allowing to feed multiple images to a trained network instead of only one.

4.2 Training and test data generation

In order to train a network, example data is needed. For this research, data of the city of Haarlem in The Netherlands is used. Each training example is a combination of an RGB-(Z) image plus a mask layer. The mask layer shows the ground truth; the correct semantic segmentation of the images. In order to retrieve this information a cleaned version of the Dutch national topographic dataset Basisregistratie Grootschalige Topografie (BGT) is used. In this cleaned version, terminated objects are removed and no overlapping objects are present by keeping only objects visible from the air. In example, at locations where a bridge is going over water, the bridge is kept and the water is removed. This cleaned version of the BGT is provided by a colleague from READAR. This cleaned BGT is reclassified into 'building', 'road', 'water' and 'other' (table 1 & table 2). Vegetation is not considered as a separate class as after exploring the BGT dataset it was concluded that both the classes 'begroeidterreindeel' as 'onbegroeidterreindeel' contained areas with vegetation and areas without vegetation. It is believed that this inconsistency strongly limits the ability of networks to properly learn to distinct vegetation from non-vegetation.

For the training/validation data an area of 4km^2 (2X2 km) is selected, containing both urban and more rural areas. This area was split into 25000 images, each containing 400X400 pixels. Every pixel is 10X10 cm. The mask images are of the same dimensions and contain one class label per pixel (figure 5). The test data is a different location of Haarlem of 4km^2, also containing both urban and more rural areas and showing a similar frequency of the classes as the test data (figure 6).

Class code	Class name
0	Other
1	Building
2	Road
3	Water

Table 1: Segmentation classes

Class code	Class BGT
0	BegroeidTerreindeel
0	GebouwInstallatie
0	Kunstwerkdeel
0	OnbegroeidTerreindeel
0	OndersteunendWaterdeel
0	OndersteunendWegdeel
0	OpenbareRuimte
2	Overbruggingsdeel
0	OverigBouwwerk
1	Pand
-	Tunneldeel
3	Waterdeel
2	Wegdeel

Table 2: Mapping of BGT classes to segmentation classes.

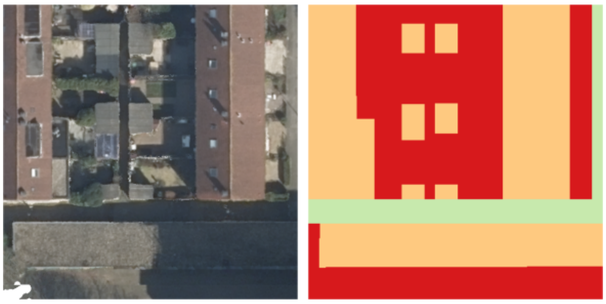

Figure 5: A training image with its mask layer. Orange = Building, Green = Road, Red = Other.

The DSM has the exact same resolution and pixel locations as the true ortho imagery, and will be cut in the exact same pieces as the training and test data. Consequently, when the networks are trained with the additional height information, every pixel contains one value for the red band, one for the green band, one for the green band, one for the absolute height relative to NAP and one class label derived from the BGT.

After generation of the training data, data augmentation will be applied. Data augmentation comprises different techniques that augment the size and the quality of the training data to allow for the generation of deep learning models of a higher quality (Shorten and Khoshgoftaar, 2019). Examples of image augmentation techniques are geometric transformations such as rotation, zooming and flipping but also adjustment of brightness and hue. Research will be done to decide on the most suitable augmentation techniques. In example, for this research rotation is an unsuitable augmentation technique as it will result in unrealistic shadows which

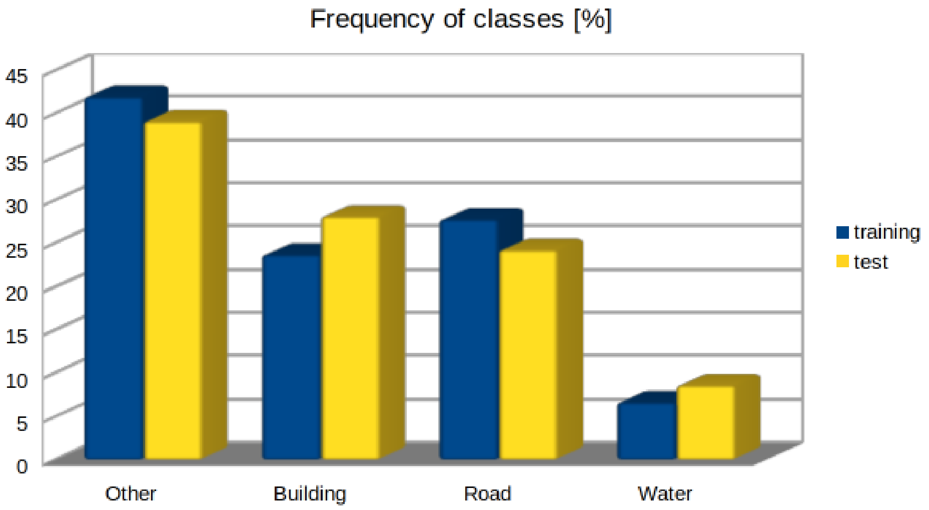

Figure 6: The frequency of the different classes occurring in the training area and in the test area.

can provide a limitation on successful training of the network.

4.3 Training of CNNs

Due to a limited availability of computational power and storage on an ordinary laptop, a server is used for training and testing of the neural networks. The selected and adjusted networks will first be trained and tested using only the true ortho imagery, without the height information. It is aimed to first achieve the highest possible performance of the network on the aerial images before including the height information, in order to ensure a valid assessment of the added value of the height information to the segmentation results. Models can be either pre-trained on a different dataset, or not pre-trained. Working with a pre-trained model, also when trained on an unrelated dataset, can save training time (Azizpour et al., 2015). However, a problem occurs when working with RGB-Z data as these pre-trained networks generally lack the support of an extra band next to RGB (Kampffmeyer et al., 2016).

An important part of the training process comprises the tweaking of the hyperparameters such as the learning-rate, number of iterations, the cost-function and the optimizer. Models will be assessed based on the performance measure (mean) intersection over union (IoU), which is a standard performance measure in segmentation. In contrast to normal accuracy measures, this measure can overcome the problem of class imbalance. IoU represents the resemblance between the ground-truth and the predicted segmentation. It is calculated for an objects in an image by dividing the size of the intersection by the union of the ground-truth region with the predicted region (Rahman and Wang, 2016). IoU is computed for each class and then averaged, providing the Mean IoU (MIoU) (figure 7) (Garcia-Garcia et al., 2017). The model delivering the best Mean IoU is selected to execute the rest of the research. Even though it is considered as a very simple metric and it contains the problem of class imbalance, some related researches use overall accuracy as performance measure. This performance measure gives the total number of correctly-classified pixels divided by the total number of all pixels (Liu et al., 2017). In order to allow for comparison of achieved results with these researches, this performance measure will also be calculated.

$$MIoU = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k}p_{ij} + \sum_{j=0}^{k}p_{ji} - p_{ii}}$$

Figure 7: Formula for calculation MIoU. k = number of classes, i = actual class of pixel , j = predicted class of pixel, p_{ii} = number of true positives, p_{ij} = number of false positives, p_{ji} = number of false negatives
(Garcia-Garcia et al., 2017).

4.4 Height information, class differences and resolution

The best performing network will be adjusted to support an extra band containing the height information per pixel. The rest of the network's architecture, such as the amount of layers and the number of nodes in each layer, is kept exactly the same. The segmentation results and the corresponding IoUs derived when height information is included, will be compared to the results without the inclusion of height information. In addition, to further assess the value of the height information, the difference in segmentation results per class will be examined and compared to the results without inclusion of height information. This comparison can be made by examining the IoU per class. Furthermore, several tests will be executed with the goal to assess the influence of resolution, by altering the pixel size of the training and test images. A comparison will then again be made between results retrieved from the 3-band network and the 4-band network.

5 Preliminary results

5.1 Analysis of the BGT

The BGT is regulated by law and freely available for any user. The BGT is developed through a collaboration between municipalities, provinces, waterboards, the Ministry of Economic Affairs, the Ministry of Defence, Rijkswaterstaat and ProRail. Every so called 'bronhouder' is responsible for delivering a specific piece of the dataset. Rules are set on the minimal required quality that should be delivered, as an attempt to guarantee high quality topographic information [5]. To ensure that the quality of the dataset is sufficient for this research and to assess the suitability of the dataset for being a mask layer, the BGT was assessed manually before preparing the training data. The metadata of the BGT was examined and the data covering Haarlem was visually compared to corresponding aerial imagery. The following paragraphs summarise the most important findings.

The BGT covers many different classes, which suggests that this dataset has the potential to train a network to distinguish a broad amount of objects. In addition, the large size and extent of the dataset ensures a high number of instances per class, leading to a high amount of training data when combined with aerial imagery. Furthermore, the borders of the geometries of the objects in the BGT generally match with the borders of the objects visible in the aerial imagery. Nevertheless, in some rare occasions the boundaries of the polygons show some deviation from the true ortho imagery (figure 8). However, it is believed that this will not have a large impact on the quality of the results as the amount of detected occurrences of this problem is small and the algorithms take surrounding pixels into account during classification.

[5] https://zakelijk.kadaster.nl/bgt

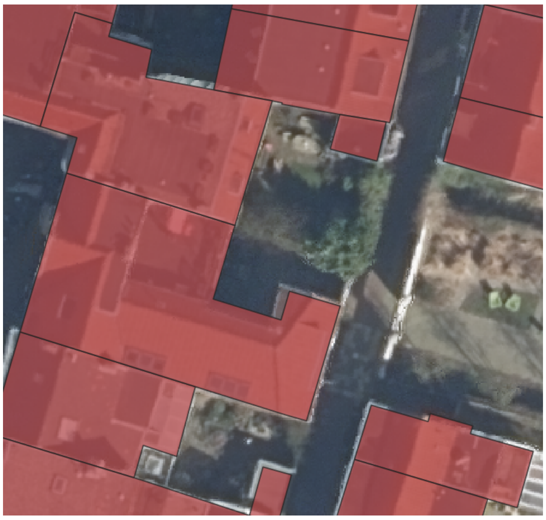

Figure 8: In some situations the boundaries of the BGT show some deviation from the true ortho imagery.

A limitation detected is the content of the classes 'begroeid terreindeel' and 'onbegroeid terreindeel' which indicate continuous vegetation and terrain without continuous vegetation respectively. It was discovered that in practice both these classes have polygons containing vegetation and no vegetation. In addition, the classification was incomplete and inconsistent. In example, patches containing a dense forest were classifies as 'onbegroeid terreindeel' and patches full of garden sheds were classified as 'begroeid terreindeel' (figure 9 and 10). Consequently, when no manual pre-selection is executed of the used training data, the BGT is considered as unsuitable for training a network to classify vegetation. Therefore, this class is not included in this research.

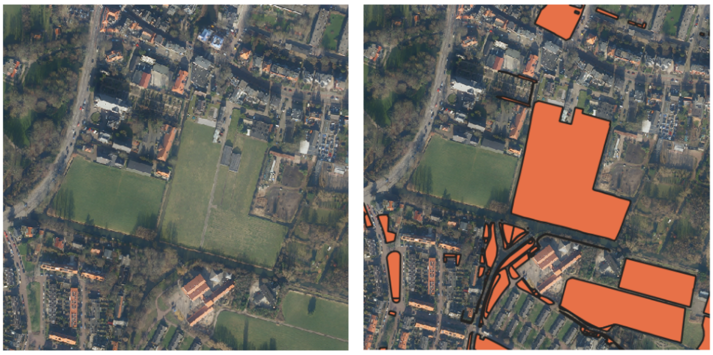

Figure 9: Incomplete and incorrect classification of the BGT. Orange is 'begroeid terreindeel'. The grass field should also be 'begroeid terreindeel' or both the grass fields should be 'onbegroeid terreindeel'. In addition, the tree block at the top left is classified as 'onbegroeid terreindeel' while it should be 'begroeid terreindeel'.

Overall it can be concluded that both the quantity and quality of the BGT are sufficient for the dataset to serve as a mask layer for the classes in this research, namely, 'building', 'road', 'water' and 'other'.

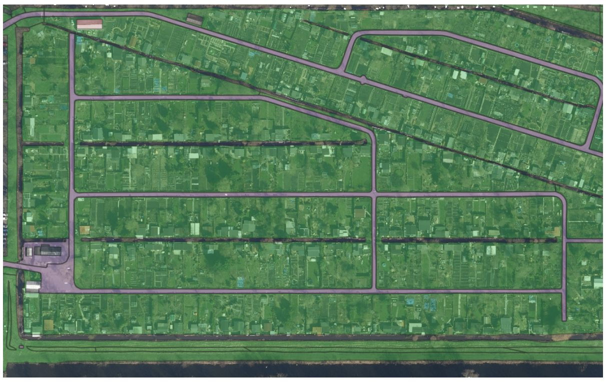

Figure 10: Green is 'begroeidterreindeel' and purple is 'ongebroeid terreindeel'. The terrain contains many garden sheds which will negatively influence the training of a network in detecting vegetation.

5.2 Preliminary tests

The Pytorch-Semseg repository on github aims at mirroring successful and popular semantic segmentation architectures in the open source machine learning framework PyTorch [6]. In this repository currently 11 different architectures are implemented and it is used as a starting point for this research. It is selected as it is well documented, relatively easily modifiable to your own needs and models are directly linked to the corresponding research papers. The repository was set up in 2017, but it is still maintained and most of the models are less than one year old.

As the architectures have not been developed with spatial data in mind, some adjustments had to be made in order to execute preliminary tests, such as the inclusion of a coordinate reference system (CRS) to the output imagery. In addition, a dataloader needed to be created to allow for the architecture to use this specific training data. Two tests were executed. The goal of these first tests was to see if the pipeline worked, and especially if the data loader generated passed on the training data correctly. In addition, it was aimed to compare segmentation results on regular ortho imagery and true ortho imagery. The first tests were executed using the FCN-8s architecture, which was pre-trained (Long et al., 2014). This architecture can be considered as a baseline architecture on which improved architectures have been built (Shah, 2017). The architecture consists of an encoder that lowers the resolution of the image and a decoder that is responsible for generating pixel-wise predictions out of the low resolution representations. A test is executed on both the regular ortho imagery and the true ortho imagery. The regular ortho imagery is retrieved from Beeldmateriaal[7]. Height information was not included as the architectures did not yet allow for the addition of an extra band. For both the regular ortho imagery and the true ortho imagery the data corresponding to the training area was randomly subdivided into three parts, a training set (70%), validation set (15%) and a test set (15%). The hyperparameters used were based on default settings and on the capacity of the server used (table 3).

The MIoUs and the overall accuracies achieved on the validation set for both the model based on regular ortho imagery and the one based on true ortho imagery are presented in table 4. These results show that using true ortho imagery delivered better semantic segmentation results than when regular ortho imagery was used. Four results of applying the trained

[6] https://github.com/meetshah1995/pytorch-semseg

[7] https://www.beeldmateriaal.nl/

networks are presented in figure 11 and 12. In these tests, the number of iterations was fixed, instead of based on the performance on the validation data, height information was not included and augmentation of the training data was not yet applied.

Hyperparameter	**Value**
Training iterations	20 000
Loss function	Cross entropy
Optimizer	Adam
Learning rate	1.0 e-3
Number of workers	5
Batch size	5

Table 3: Hyperparameters used with the FCN8s architecture and both the regular ortho imagery and the true ortho imagery.

	MIoU	**OAcc**
FCN-8s (Ortho)	68.51	80.14
FCN-8s (True ortho)	75.74	84.89

Table 4: Performance of the two models. Mean intersection over union (MIoU) and overall accuracy (OAcc) are provided in percentages.

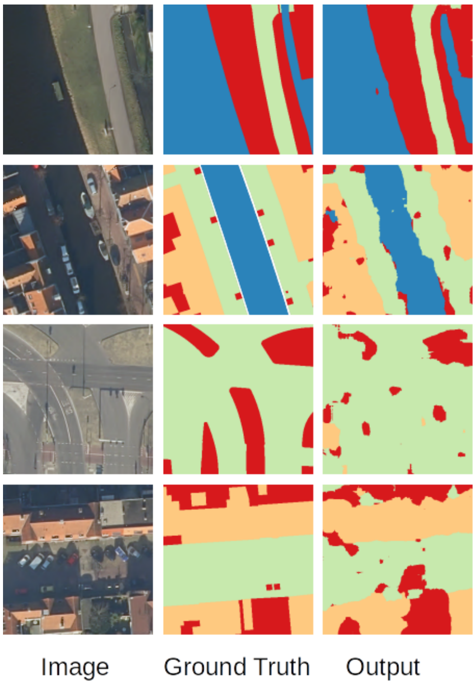

Figure 11: Example results of preliminary tests using the FCN-8s architecture and regular ortho imagery.

It can be noted that predicted boundaries resulting from the regular ortho imagery test are more fuzzy when compared to the boundaries predicted using the true ortho imagery. This

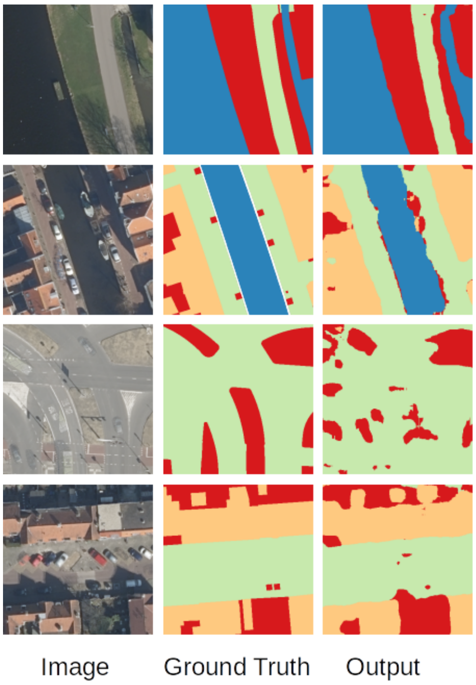

Figure 12: Example results of preliminary tests using the FCN-8s architecture and true ortho imagery.

was expected as the true ortho imagery is corrected for relief displacement. These results support the decision of this proposed research to use true ortho imagery rather than regular ortho imagery.

Furthermore, the achieved overall accuracy on the true ortho imagery of 84.89% is already similar to the achieved overall accuracy described in the related research section of 85.39% from Liu et al. (2017) on non-eroded ground truth data. The accuracy of the preliminary results for the building class is 88.35%, which is a bit lower than the 92.23% achieved by Liu and his colleagues. Even though the used classes were not identical and the overall accuracy is not the most inclusive performance measure, the similar value in overall accuracy is promising. This statement can be made as, in contrast to these preliminary results, Liu and his colleagues did already include height information and the specific architecture used and the hyperparameters selected for these preliminary tests, where somewhat arbitrary.

6 Time planning

An overview of the time planning for this research is presented in figure 13. Writing of the thesis will be done parallel to the research. From February onwards, one day a week will be dedicated to writing.

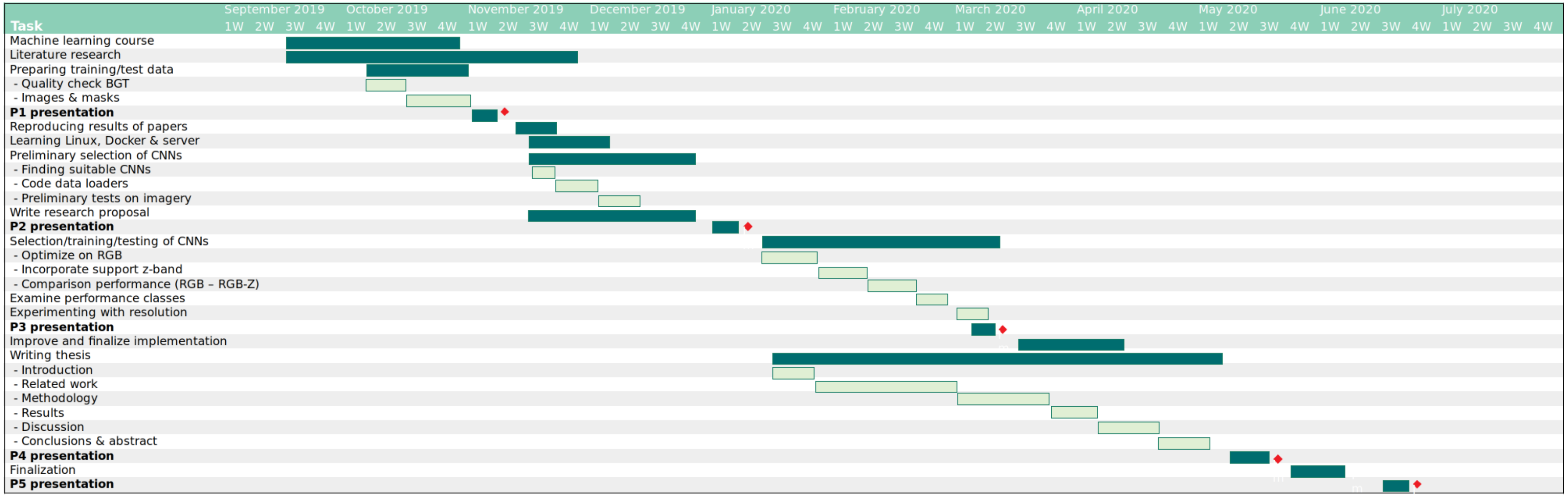

Figure 13: Gantt chart of the time planning for the research.

7 Tools and datasets used

7.1 Tools

In order to prepare the training and test data, QGIS software including the 'GridSplitter' tool is used [8]. The existing neural networks which will be tested and adjusted are mostly stored on github and coded in python, using the open source machine learning framework PyTorch[9]. This package uses tensors which are similar to NumPy's ndarrays. These tensors can use the power of GPUs to allow for fast computations. In addition, as the computational power of used laptops are not sufficient, Docker is used to generate containers[10]. These containers allow for training and testing of the networks on an external server.

7.2 Datasets

The aerial imagery used as input for the neural networks will be true ortho imagery generated by the company READAR. As described before, this is RGB aerial imagery with a 10 cm resolution and which is corrected for relief displacement. Relief displacement comprises the problem that due to deviating distances from the central perspective and vertical relief, too much information can be visible on one side of objects, while occlusions occur on other sides of the objects in images (Sheng et al., 2003; Lemmens, 2011). The true ortho imagery is generated using READAR's dense matching software based on deep learning techniques. In order to create this imagery, the software uses point clouds, which are obtained from the national high resolution stereo imagery, captured during spring 2018, from Beeldmateriaal Nederland[11]. In addition, interpolation techniques are used for estimating values where occlusion has occurred. The resulting true ortho imagery contains an extra band providing information on whether or not the present pixel values were interpolated. The obtained point clouds represent the absolute height relative to NAP per pixel and are converted to a digital surface model (DSM). This DSM fulfils the role of the fourth band, the Z-band, in this research.

Furthermore, the dataset Basisregistratie Grootschalige Topografie (BGT) is used as a mask layer for the training examples. This dataset has an accuracy of 20 centimeters and provides detailed topographic information of The Netherlands. It can be accessed from the PDOK portal[12], which is the national open dataset portal for geo-information of the Dutch government.

[8] https://plugins.qgis.org/plugins/gridSplitter/

[9] https://pytorch.org/

[10] https://www.docker.com/

[11] https://www.beeldmateriaal.nl/luchtfotografie-hoge-resolutie/

[12] http://pdok.nl

References

H. Azizpour, A. S. Razavian, J. Sullivan, A. Maki, and S. Carlsson. Factors of transferability for a generic convnet representation. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1790–1802, 2015.

V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

C. Couprie, C. Farabet, L. Najman, and Y. LeCun. Indoor semantic segmentation using depth information. *arXiv preprint arXiv:1301.3572*, 2013.

Feng Ning, D. Delhomme, Y. LeCun, F. Piano, L. Bottou, and P. Barbano. Toward automatic phenotyping of developing embryos from videos. *IEEE Transactions on Image Processing*, 14(9):1360–1371, Sept. 2005. ISSN 1057-7149. doi: 10.1109/TIP.2005.852470. URL `http://ieeexplore.ieee.org/document/1495508/`.

A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, and J. Garcia-Rodriguez. A Review on Deep Learning Techniques Applied to Semantic Segmentation. Apr. 2017. URL `https://arxiv.org/abs/1704.06857v1`.

X. Glorot, A. Bordes, and Y. Bengio. Deep Sparse Rectifier Neural Networks. In G. Gordon, D. Dunson, and M. Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 315–323, Fort Lauderdale, FL, USA, Apr. 2011. PMLR. URL `http://proceedings.mlr.press/v15/glorot11a.html`.

S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. Learning Rich Features from RGB-D Images for Object Detection and Segmentation. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, volume 8695, pages 345–360. Springer International Publishing, Cham, 2014. ISBN 978-3-319-10583-3 978-3-319-10584-0. doi: 10.1007/978-3-319-10584-0_23. URL `http://link.springer.com/10.1007/978-3-319-10584-0_23`.

B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Simultaneous Detection and Segmentation. July 2014. URL `https://arxiv.org/abs/1407.1808v1`.

D. R. Hush and B. G. Horne. Progress in supervised neural networks. *IEEE signal processing magazine*, 10(1):8–39, 1993.

M. Kampffmeyer, A.-B. Salberg, and R. Jenssen. Semantic Segmentation of Small Objects and Modeling of Uncertainty in Urban Remote Sensing Images Using Deep Convolutional Neural Networks. pages 1–9, 2016. URL `https://www.cv-foundation.org/openaccess/content_cvpr_2016_workshops/w19/html/Kampffmeyer_Semantic_Segmentation_of_CVPR_2016_paper.html`.

Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature14539. URL `http://www.nature.com/articles/nature14539`.

M. Lemmens. *Geo-information: technologies, applications and the environment*, volume 5. Springer Science & Business Media, 2011.

Y. Liu, D. Minh Nguyen, N. Deligiannis, W. Ding, and A. Munteanu. Hourglass-ShapeNetwork Based Semantic Segmentation for High Resolution Aerial Imagery. *Remote Sensing*, 9(6):522, May 2017. ISSN 2072-4292. doi: 10.3390/rs9060522. URL http://www.mdpi.com/2072-4292/9/6/522.

J. Long, E. Shelhamer, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. Nov. 2014. URL https://arxiv.org/abs/1411.4038v2.

S. Paisitkriangkrai, J. Sherrah, P. Janney, and A. Van-Den Hengel. Effective semantic pixel labelling with convolutional networks and Conditional Random Fields. In *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 36–43, Boston, MA, USA, June 2015. IEEE. ISBN 978-1-4673-6759-2. doi: 10.1109/CVPRW.2015.7301381. URL http://ieeexplore.ieee.org/document/7301381/.

X. Qi, R. Liao, J. Jia, S. Fidler, and R. Urtasun. 3d Graph Neural Networks for RGBD Semantic Segmentation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5209–5218, Oct. 2017. doi: 10.1109/ICCV.2017.556.

R. Qin, J. Tian, and P. Reinartz. 3d change detection – Approaches and applications. *ISPRS Journal of Photogrammetry and Remote Sensing*, 122:41–56, Dec. 2016. ISSN 09242716. doi: 10.1016/j.isprsjprs.2016.09.013. URL https://linkinghub.elsevier.com/retrieve/pii/S0924271616304026.

M. A. Rahman and Y. Wang. Optimizing Intersection-Over-Union in Deep Neural Networks for Image Segmentation. In G. Bebis, R. Boyle, B. Parvin, D. Koracin, F. Porikli, S. Skaff, A. Entezari, J. Min, D. Iwai, A. Sadagic, C. Scheidegger, and T. Isenberg, editors, *Advances in Visual Computing*, volume 10072, pages 234–244. Springer International Publishing, Cham, 2016. ISBN 978-3-319-50834-4 978-3-319-50835-1. doi: 10.1007/978-3-319-50835-1_22. URL http://link.springer.com/10.1007/978-3-319-50835-1_22.

S. Saito, T. Yamashita, and Y. Aoki. Multiple Object Extraction from Aerial Imagery with Convolutional Neural Networks. *Electronic Imaging*, 2016(10):1–9, Feb. 2016. ISSN 2470-1173. doi: 10.2352/ISSN.2470-1173.2016.10.ROBVIS-392. URL http://www.ingentaconnect.com/content/10.2352/ISSN.2470-1173.2016.10.ROBVIS-392.

A. M. Saxe, P. W. Koh, Z. Chen, M. Bhand, B. Suresh, and A. Y. Ng. On random weights and unsupervised feature learning. In *ICML*, volume 2, page 6, 2011.

P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*, 2013.

M. P. Shah. Semantic Segmentation Architectures Implemented in PyTorch. *https://github.com/meetshah1995/pytorch-semseg*, 2017.

Y. Sheng, P. Gong, and G. S. Biging. True Orthoimage Production for Forested Areas from Large-Scale Aerial Photographs. *Photogrammetric Engineering & Remote Sensing*, 69(3): 259–266, 2003. ISSN 0099-1112. doi: doi:10.14358/PERS.69.3.259. URL https://www.ingentaconnect.com/content/asprs/pers/2003/00000069/00000003/art00002.

C. Shorten and T. M. Khoshgoftaar. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1):60, Dec. 2019. ISSN 2196-1115. doi: 10.1186/s40537-019-0197-0. URL https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0197-0.

M. Volpi and D. Tuia. Dense semantic labeling of subdecimeter resolution images with convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 55(2):881–893, 2016.

Q. Wang, J. Lin, and Y. Yuan. Salient Band Selection for Hyperspectral Image Classification via Manifold Ranking. *IEEE Transactions on Neural Networks and Learning Systems*, 27(6): 1279–1289, June 2016. ISSN 2162-2388. doi: 10.1109/TNNLS.2015.2477537.

Y. Yuan, D. Ma, and Q. Wang. Hyperspectral Anomaly Detection by Graph Pixel Selection. *IEEE Transactions on Cybernetics*, 46(12):3123–3134, Dec. 2016. ISSN 2168-2275. doi: 10.1109/TCYB.2015.2497711.

H. Zhu, F. Meng, J. Cai, and S. Lu. Beyond pixels: A comprehensive survey from bottom-up to semantic image segmentation and cosegmentation. *Journal of Visual Communication and Image Representation*, 34:12–27, Jan. 2016. ISSN 10473203. doi: 10.1016/j.jvcir.2015.10.012. URL https://linkinghub.elsevier.com/retrieve/pii/S1047320315002035.