Automated Semantic Segmentation of Aerial Imagery using Synthetic Data

Camilo Cáceres
student #5362210

1st supervisor: Shenglan Du
2nd supervisor: Jantien Stoter
external supervisor: Sven Briels

January 26, 2022

1 Introduction

Image semantic segmentation is one of the fundamental tasks in remote sensing. Semantic segmentation is the ability to assign labels to all pixels of an image (Garcia-Garcia et al., 2017). Semantic segmentation proves to be an essential prerequisite for various applications such as urban planning, agriculture, and real-state (Kampffmeyer et al., 2016; Zhu et al., 2016). For instance, classification between roads and buildings, from aerial images, is important for change detection and updated cartography in the built environment (Saito et al., 2016). Traditionally, unsupervised and supervised methods are used to tackle semantic segmentation problems with the use of statistical properties on the feature space of the image (Rosenberger et al., 2006). Recently, automated segmentation of aerial imagery has been a problem addressed over the past years with deep learning techniques that account for satisfactory results as it derived the labels to targeted classification tasks (Yuan et al., 2021).

Most deep learning models consist of four phases. The first one is to obtain the data for the problem to solve. Second, to label the data to Third, train the model. Finally, is to inference the trained model into a real-image (Nikolenko, 2021; Liu et al., 2017). During the whole process of developing a deep learning model, with not enough open datasets, 80% of the time goes to the annotation phase since it has to be done manually or manually checked after an automatic process (Nikolenko, 2021). In addition, the acquisition of labelled data is expensive for vast geographic regions (Kong et al., 2019). Several approaches have been made to tackle this problem. For instance, Maggiori et al. (2017) created an extensive dataset with labelled data for five cities across the world, which helped scientists to lower the time of the annotation phase. Nevertheless, despite being large datasets, it lacks variability in real-world scenarios (Kong et al., 2019).

Another way to tackle the problem of annotation is to create synthetic data for training deep learning models. Synthetic data refers to imagery from a virtual world that simulates the real-world (Nikolenko, 2021; Kong et al., 2019; Ros et al., 2016). Compared to real imagery data, synthetic images have several significant advantages, such as simulating different conditions (e.g., lighting, camera positions), lowering production costs and producing unlimited possibilities of images with pixel-wise annotations (Nikolenko, 2021). Nevertheless, the use of synthetic data has led to new challenges such as the difference of domains between the real-world and the virtual-world (Nikolenko, 2021; Kong et al., 2019). For this problem, various domain adaptation techniques have been developed to adapt the domain of synthetic imagery to the real imagery domain.

Synthetic training data gives a opportunity to improve models, lower the production time and costs (Nikolenko, 2021; Kong et al., 2019). For these reasons, the current thesis aims to create a benchmark to produce synthetic data for automated aerial image segmentation.

For this research, ESRI city engine is used to edit and use default virtual cities with procedural architecture techniques following the renderization of images from a simulated camera. In this process, the annotations are made. The next step to create different 3D features in the synthetic city in order to train existing deep learning models to evaluate their performance. Finally, a domain adaptation technique exploration is performed to the synthetic images to further improve the results. The images in which the model will be performed are given by READAR and are in the Dutch context.

This proposal is organized as follows. In the second chapter, the related work is presented with relevant literature. In the third chapter, the research problem is defined with the scope of the project. Furthermore, chapter 4 presents the methodology and the proposed benchmark to create synthetic data for aerial images. In chapter 5, the time

planning and tools and datasets to be used are presented.

2 Related work

2.1 Deep Learning for automated semantic classification

Semantic segmentation is one of the main tasks for remote sensing, giving each pixel a meaningful class to the image (Zhu et al., 2016). Due to the variability, complexity, heterogeneity of the remote sensing data, it is a complex problem to do semantic segmentation in these images (Kampffmeyer et al., 2016). Nevertheless, deep learning models have shown outstanding performance for semantic segmentation (Yuan et al., 2021) making an impact in remote sensing.

Deep Learning is a machine learning technique that consists of methods that learns complex representations from raw data input (Goodfellow et al., 2016). The deep learning model consist of a set of layers, called neural networks, which are non-linear functions that compute mappings between the input and the output layer (Lecun et al., 2015). A neural network is composed of an input layer that contains the observable data, one or multiple hidden layers that extract features from the input data, and an output layer that contains the requested information of the input data (Goodfellow et al., 2016).

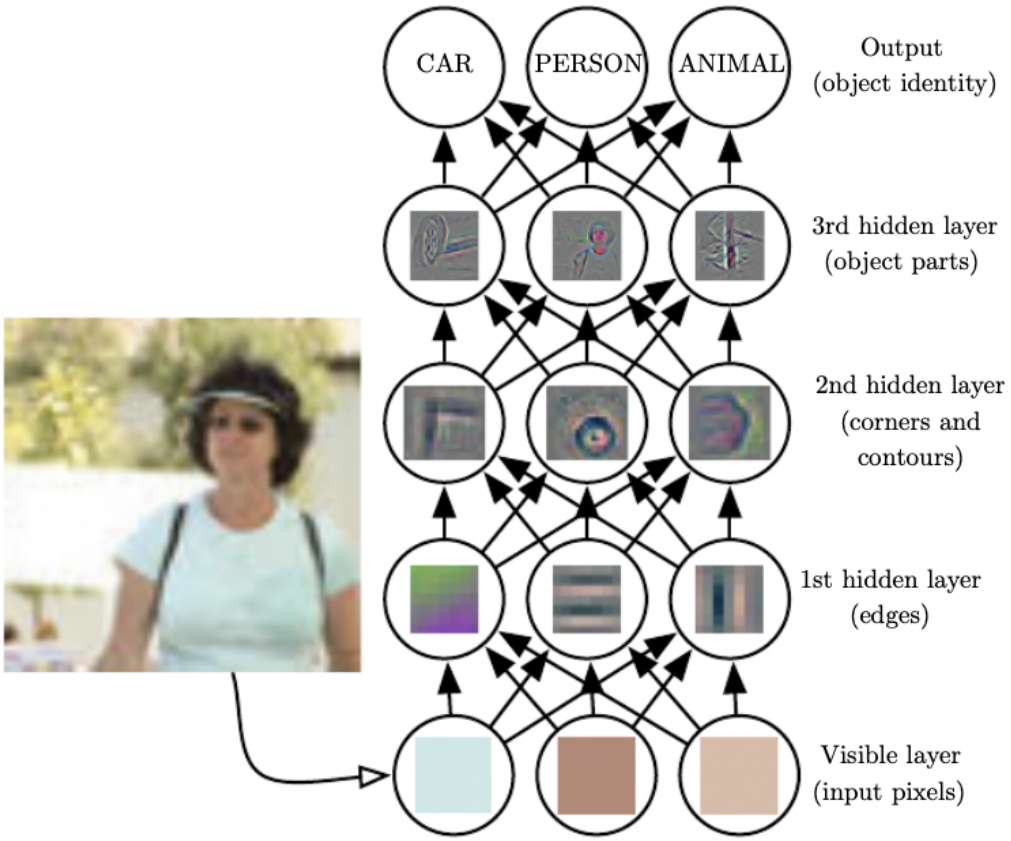

Figure 1: Example of a Deep Learning Architecture with the neurons as the kernels inside the circles. The hidden layers extracts different features of the image to the final layer that identifies the object in the input layer. Image taken from: (Goodfellow et al., 2016)

Images are represented as 2D arrays in a computer. Thus, Convolutional Neural Net-

works (CNN) are designed to deal with data in the form of arrays. CNNs is a neural network that uses convolutional operations with four main types of layers: Convolutional layers, which are filter layers; transposed convolutional layers that are for upsampling operations; non-linear function layer, which activates the non-linearity into the network and spatial pooling layers to reduce the size of the input volume (Liu et al., 2017)

Different architectures have been made to improve CNNs for semantic segmentation in remote sensing. For instance, Kampffmeyer et al. (2016) and Maggiori et al. (2017) used a Fully Convolutional Network (FCN) for semantic image segmentation. This study obtained 87% accuracy in both ISPRS datasets, Postdam and Vaihingen (ISPRS, 2020). FCNs are composed of three phases: multi-layer convolution, deconvolution and fusion. Specifically, FCNs use convolutional layers to get a score for each class. As pooling is used for the convolutional processes, the output size is smaller than the original. Thus, the deconvolution step returns the size back but loses spatial detail in the class score. To get back the spatial details, an unsampled deep layer is extracted and fused with a shallow layer by additional element-sum (Yuan et al., 2021).

Another architecture is U-Net which consist of convolutional and deconvolutional layers and aims to use little training data. The U-Net is made of a contracting path to get the context of an image and a symmetric expanding path to get precise localization of features (Ronneberger et al., 2015). This architecture is used by Xu et al. (2018) using very high-resolution aerial images, and performing a fusion with a DSM. For the Vaihingen dataset the U-Net has an accuracy of 96% and for Postdam dataset, 98%.

The SegNet architecture (Badrinarayanan et al., 2017) consist of two sub-networks. An encoder and decoder. The encoder is a structure of convolutional and pooling layers to extract features. With this network, more meaningful classes are extracted, but spatial information loses detail. In the next network, the decoder is used to recuperate the lost spatial information using an upsampling process (Yuan et al., 2021). This architecture is used by Audebert et al. (2018) for Vaihingen dataset with an accuracy of 89%. From SegNet it was created the FuseNet, which uses three sub-networks, two encoders, the RGB values and Depth values and the decoder that, with a fusion layer, process the RGB-D values (Hazirbas et al., 2015). FuseNet was used by (Audebert et al., 2018) taking the DSM from point clouds and with very high-resolution images, getting an accuracy for both Postdam and Vaihingen datasets of 90%.

For this research the architecture used is the FuseNet because it enables the use of 4-band image (RGB-Z), it is all implemented on python and according to Mulder (2020) the mIoU was 0.87 on experiments in Harleem in the Netherlands. These results were better than using SegNet architecture.

2.2 Synthetic Imagery for training Deep Learning Models

Synthetic imagery is defined as ”imagery that has been captured from a simulated camera operating over a virtual world” (Kong et al., 2019). Instead of training deep learning models with costly aerial or satellite images, a synthetic imagery approach can have several benefits, such as free labelled data as classes are defined by design, simulation of different seasons or lighting conditions is adjustable, and the variability of the images can be set in the design process of the virtual world. (Nikolenko, 2021; Kong et al., 2019).

One of the first approaches for synthetic imagery for training deep learning models was the SYNTHIA dataset (Ros et al., 2016). This dataset was created to have pixel-perfect semantic segmentation for automated car navigation. SYNTHIA is created in a virtual environment using Unity, and it simulates a virtual array of cameras throughout the city that simulates a real car. The research provided two sets, one for training data for

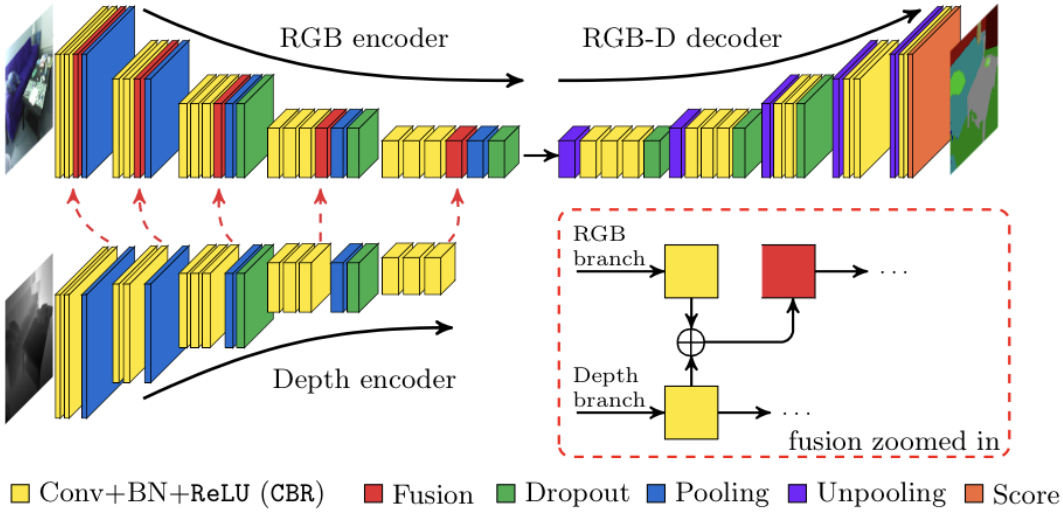

Figure 2: FuseNET architecture. Two networks are used, one for the RGB and other for Depth. These are used as an encoders. Later, a fused RGB-D decoder is used. Image taken from Hazirbas et al. (2015)

deep learning and another to analyse spatio-temporal constraints of objects (Ros et al., 2016). The first set consists of 13400 images trained in a FCN model and evaluated in CamVid dataset, which are images from Cambridge, UK. The results showed that only with CamVid dataset an accuracy of 78% and, adding the SYNTHIA dataset, the accuracy increased to 84% (Ros et al., 2016). Nevertheless, for the CBCL dataset from Chicago, USA, an initial 79% of accuracy with only its own dataset and with SYNTHIA it showed 75% of accuracy. They believed the decremented is due to the combination of early and late layers during upsampling, but no further evaluation was made (Ros et al., 2016).

Figure 3: Examples of SYNTHIA images plus the depth image. SYNTHIA dataset is for unmanned vehicles purposes. Image taken from: Ros et al. (2016)

ProcSy (Khan et al., 2019) is a synthetic dataset from ESRI City Engine; it is used for automatic driving semantic segmentation. They used different weather conditions like cloud and rain. They created 8000 images for experiments. The training data is input to a DeepLab v3+ model into the real-world dataset CityScapes, obtaining good results with mIoU (mean intersection over union) between 70 and 75% (Khan et al., 2019). The virtual city was made from a real city in Canada, and they used the dataset to study the effect of different conditions in the current deep learning algorithms. This study did not show how the dataset will perform in another location and how it adapt from the virtual to the real domain.

In the case of synthetic data for aerial or satellite images, Kong et al. (2019) created a dataset called Synthinel. Using ESRI City engine, a virtual city was created from the program's default settings. It used 1640 images to train a DeepLab v3+ and a U-net model for building classification. The study used real and synthetic training data to evaluate it on the INRIA (Maggiori et al., 2017) and ISPRS (ISPRS, 2020) datasets. For U-net using INRIA dataset, the improvement of mIoU was 0.3%, from 69.0 to 69.3%. For DeepLab v3+ the improvement was greater (1.1%) from 72.2 to 73.3%. Nevertheless, the research also performed blind segmentation, which evaluates the ISPRS dataset with training data from both the Synthinel and INRIA datasets. In this case, the domain is changed. Thus the results decreased, but the impact of the synthetic imagery increased. Using a U-net without the synthinel the mIoU was 45.0%, and with it increased to 47.7%. On the other hand, for DeepLab v3+ the increment is greater from 58.1% to 63.5% (Kong et al., 2019)

Figure 4: Synthetic city from ESRI City Engine used in (Kong et al., 2019). Default settings from City Engine creates this city. Image taken from: Kong et al. (2019)

2.3 Domain Adaptation for Synthetic Imagery

Real and synthetic imagery have different distributions, which leads to a shift on its domains. This shift decreases the performance of the models. To reduce this difference, Domain Adaptation (DA) techniques come handily (Wang and Deng, 2018; Sankaranarayanan et al., 2017). Three different groups of DA have been classified, discrepancy-based, adversarial-based and reconstruction-based (Wang and Deng, 2018).

In discrepancy-based DA, the first method is Class Criterion that consists of labelling a small portion of the target (real) domain to train the synthetic domain (Wang and Deng, 2018). The second method is the Statistics Criterion. Rozantsev et al. (2016) uses the concept of Maximum Mean Discrepancy (MMD), that takes two distributions and through a kernel-based sample, tests if the domains distributions are different (Gretton et al., 2008). This technique is used in the DL model as a loss function to minimize this distribution difference. Another method of the statistics criterion is the Correlation Alignment (CORAL). This method used a second-order statistic, the correlation, to align the distribution between the training and the target set (Sun et al., 2016). Even though these methods have been used to train DL models, most are for object recognition tasks where clear borders and edges are present, different from segmentation.

The adversarial-based methods of DA consist of a generative model and an adversarial model. The generative models take the training images to feature real images. The

adversarial model creates a binary label from the training and the real images (Wang and Deng, 2018; Liu and Tuzel, 2016). An approach to this model is the Coupled Generative Adversarial Networks (CoGAN) (Liu and Tuzel, 2016). This method uses two GAN sharing parameters to force the models to learn the same domain. Then, the training data is used to create new synthetic data that is further used as training data to the DL model.

Furthermore, the reconstruction-based methods are reconstructed real images from the synthetic one, which facilitates DL models to perform in similar domains. Methods such as Cycle-Consistent GAN (CyCADA) mapped the trained images to the real images while removing the difference between the domains. First, an image-space adaptation is performed to map the synthetic data to the real imagery domain using GAN. Then, the model learns to operate in real imagery with adapted data and the labels from the synthetic data. Finally, another round of adaptation in feature-space between the adapted synthetic data and the real data is performed (Hoffman et al., 2018).

Figure 5: CyCada Domain Adaptation takes a synthetic image and outputs the same image with similar domain to a real world image. Image taken from: Hoffman et al. (2018)

3 Research questions

The main objective of this thesis is:

To what extent synthetic data can improve the current Deep Learning based models for automated semantic segmentation for aerial images?

The following sub-questions are listed to help the previous question be solved:

- How to create an automatic virtual city to take synthetic imagery for training data?
- To what extent does 3D feature models (trees, buildings or roads) of a virtual city affects the results of semantic segmentation of aerial images?
- To what extent the distribution of classes can be changed in virtual cities to improve semantic segmentation of aerial images?
- Which is the most suitable ratio between real and virtual training data for semantic segmentation of aerial images?
- Which domain adaptation technique is more effective for adapting from synthetic imagery to real imagery domain?

3.1 Scope of research

The scope of this research is to create a pipeline to generate a virtual city with class distribution parameters and take images with its semantic segmentation labels to be used in existing deep learning models. This thesis will not evaluate the behaviour of a DL architecture. Instead, it will use a current, well-known model to evaluate the quality of the synthetic imagery. Furthermore, this thesis will focus on creating synthetic training data and how it fits to train real-world imagery for semantic segmentation.

4 Methodology

In this section the methodology to perform a automated semantic segmentation of the aerial images from synthetic imagery is explained. This approach will detect three classes; Buildings, roads and other that consist in vegetation and other features in the real-world. In the following steps is explained from the synthetic city generation to the evaluation of DL in real imagery.

4.1 Synthetic City Creation

The first objective of this thesis is to create a virtual city. For this purpose I will used procedural building modeling from Esri CityEngine. This program is used mainly because the default rules that enables to create random virtual worlds which could be easily customizated either manually and automatically through a python API. In addition, the software focus mainly in urban planning and building modelling which makes the best option to build a virtual city with different building models.

First, The CityEngine pipeline consists in input layers. A DTM and a land-use map are optionally input in the model. The DTM is to align the city to the terrain and land-use to restrict areas of the city for specific use. Furthermore, the street network (graph) is created followed by lots and streets shapes (blocks). Finally, a 3D model is created by applying a set of procedural rules from the 2D shapes.

In the current thesis the CityEngine pipeline will be built with the help of the python API.

4.2 Parameters for 3D models

Having the network with the streets and lots created, the next step is to use the computer generated architechture (CGA) to assign rules to the 2D shape in order to create 3D models (Figure 7). It consist of a series of rules that performs instructions to the initial shape selected. In this case the initial shape is either the lot or the street shape. The instructions perform geometry operations such as extrusion, transformation, division, adding textures or even changing the level of detail.

In this work, the *InternationalCity.cga (IC)* rule that comes from default in the basic CityEngine licence will be used and edited. In CGA the procedural modeling starts always with the "start rule". In the case of the IC rule, the starting rule is to classify the lot in a normal, inner or corner lot. Then depending on the area of the lot, different types of block are assign to the lot. The types of block can be "residential", "apartment buildings", "office buildings" or "open space". Now, for every type of blocks, there are some rules to form the 3D model of the lot. For instance, there is a rule that controls the angle of the roof, being 0 a flat roof and 45 a gable roof.

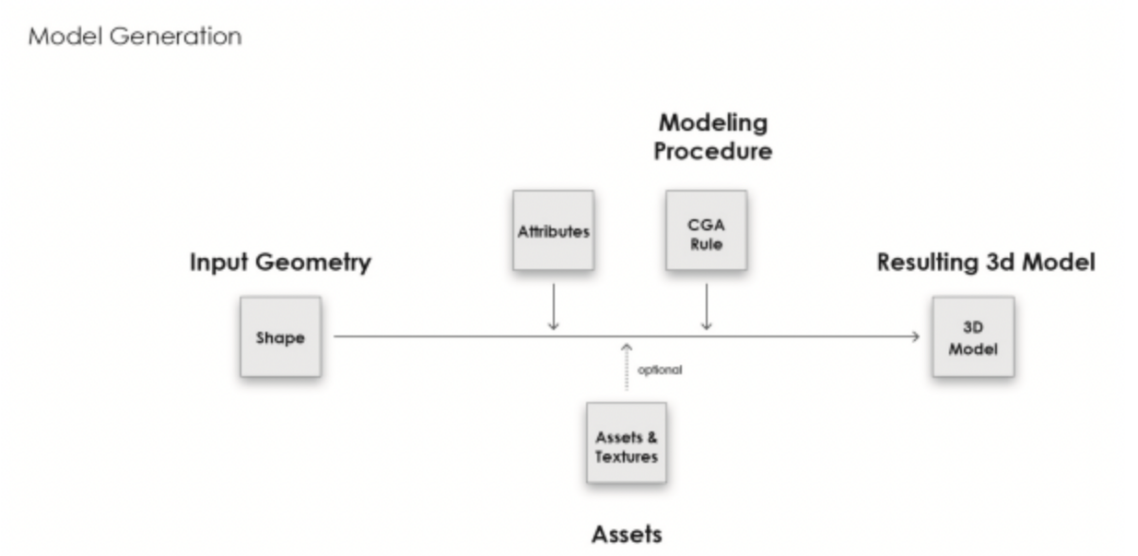

Figure 6: City Engine model generation. The box of attributes refers to the graph network consisting in roads and lots. Assets and textures are images from real buildings or roads to make realistic looks of the models. CGA rules are for the modeling of the attributes. Image taken from: https://doc.arcgis.com/en/cityengine

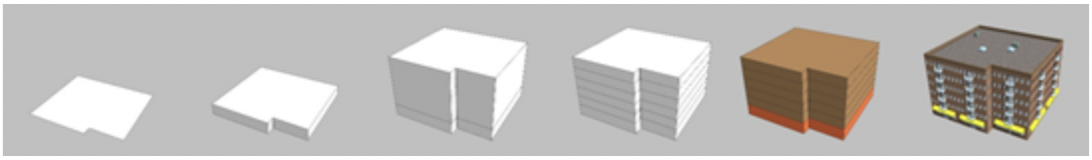

Figure 7: CGA rules are applied to a simple 2D shape to build complex building 3D models. Image taken from: https://doc.arcgis.com/en/cityengine

In this work, different distributions of blocks will be adjusted in the IC rule, this distribution will be according the classes to detect. For instance, a distribution of a city can be 30% of buildings, 20% roads and 50% other.

In addition, I will build rules to create more variety of buildings to focus on a specific area. For example, for the Dutch case in which the synthetic images will be tested, common building types will be added to be further evaluate.

Furthermore, three different types of tree models will be tested. It is important that the model has a balance between storage space and realism (Figure 8).

For modeling roads, the *Streets_Advanced.cga (SA)* will be used. SA is a set of rules made from ESRI to create realistic roads. It is composed of streets layouts, sidewalks and green space. For this rule different attributes can be added such as bike lines, lights, cars, or pedestrians. This rule package will be edited to evaluate the importance of different 3D models that are present in the real world.

Finally, the city model will be imported as an OBJ file to be further processed.

4.3 Rendering Images

Once the city model is in OBJ format, a python-based pipeline called BlenderProc will be used. BlenderProc is a procedural pipeline to create, from virtual scenes, realistic images to be used in DL models. This pipeline provides quality rendering, perfect se-

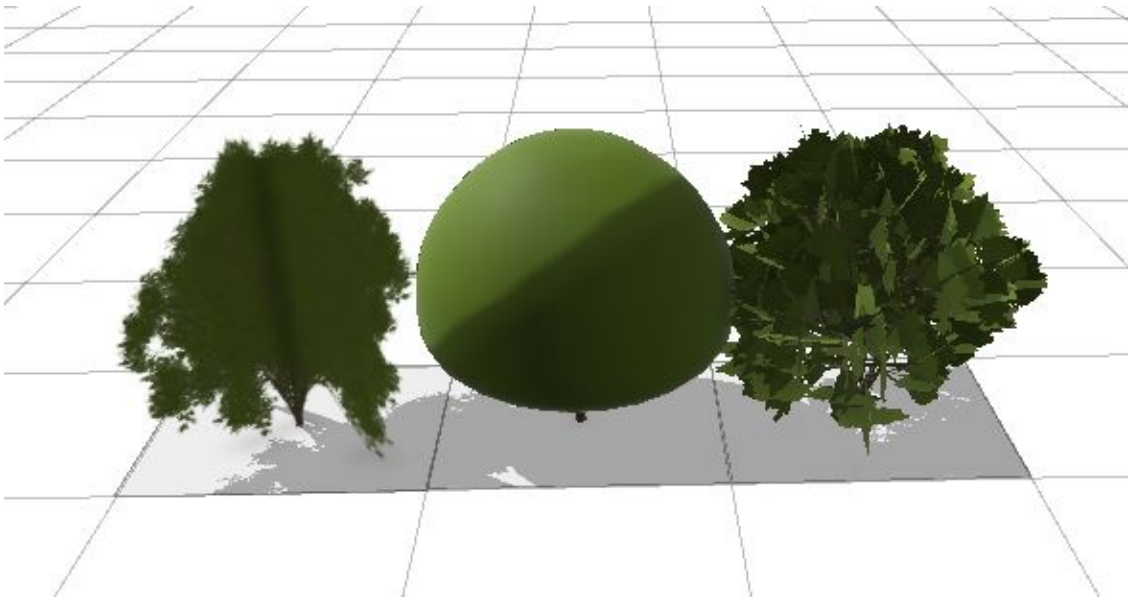

Figure 8: Different types of tree models available in City Engine. Image made in City Engine

mantic segmentation and depth images in a open source 3D model software, blender. (Denninger et al., 2019). In this thesis, first the OBJ model is imported and then saved the whole scene. Furthermore, a path is created with different positions of the camera to take the imagery. Additionally, every object in the scene has to be categorized for the semantic segmentation of the image. For this purpose, every texture is categorized with the predefined classes. (Buildings, roads and other). With this attributes, the lighting parameters are set with the light angle and light intensity. Finally, the scene goes through BlenderProc to get three different images for each location of the camera. The color image, the semantic segmentation image (labelled image) and the depth image (DSM) (Figure 9).

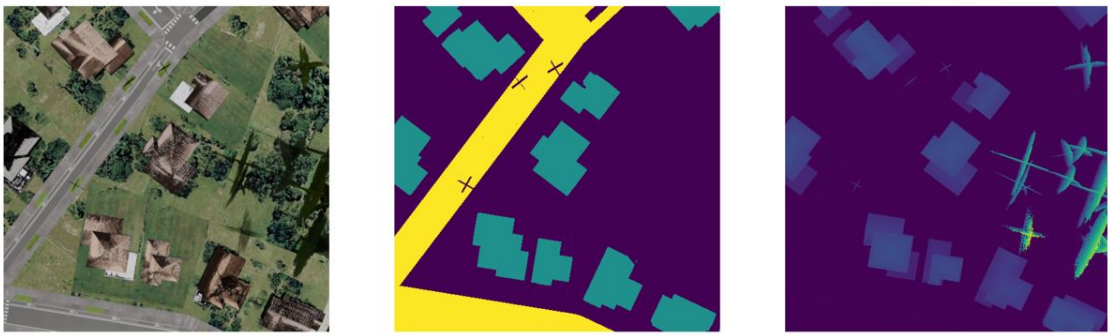

Figure 9: Left: Color image from synthetic city. Centre: Semantic segmentation of the image. Right: DSM of the synthetic city.

4.4 FuseNet implementation Details

After having the training data ready, a FuseNet model will be performed. The parameters will be constant as the focus is to evaluate the quality of the training data rather than the model.The parameters will be taken as the average of the top five best epochs

in the model. The number of epochs will be until a defined threshold for the networks convergence. The loss function is the cross entropy function for the imbalance of the classes and for classification purposes (Fleuret, 2021). Adam optimizer, that consists of a gradient descent moving average (Fleuret, 2021) is used with a initial learning rate of 1e-4. These parameters are also used by Mulder (2020) in her model for the Netherlands.

4.5 Evaluation

For the evaluation of the model, a 1X1 km of imagery from The Hoorn, Netherlands is used. The ground truth is made from the *Basisregistratie Grootschalige Topografie (BGT)* (PDOK, 2021). A cleaned version is taken from different layers and then performed raster operations to get the ground truth. The main idea of this project is to evaluate all different training datasets with the same model parameters to study the importance of the design of the synthetic city. Thus, the models will be assessed with precision, recall, accuracy (Tempfli et al., 2009) and mIoU (Garcia-Garcia et al., 2017).

- Precision: It reflects on how the model classifies a class as positive. It is the ratio between true positives to true positives plus false positives.

$$Precision = TP/(TP + FP) \tag{1}$$

- Recall: It reflects how the model identifies positives samples. It is the ratio between true positives to true positives plus false negatives.

$$Recall = TP/(TP + FN) \tag{2}$$

- Accuracy: It refers to the samples that were correctly predicted over the total samples. It is the ratio between true positives and false negatives over the total number of samples. It has one big problem and is the class imbalance, Very important in this research as the real world is not balance. Nevertheless it is important to compute as most of the related work use this metric to evaluate their results.

$$Accuracy = TP + TN/(TP + TN + FN + FP) \tag{3}$$

- mean Intersection over Union: It is the average of ratio between the intersection and the union of the predicted samples and the ground truth between different classes. It can also be the mean ratio between true positives between the sum of true positives, false negatives and false positives (Garcia-Garcia et al., 2017).

$$mIoU = \frac{1}{Classes} \sum (TP/(TP + FN + FP)) \tag{4}$$

4.6 Domain Adaptation

Finally, two methods of domain adaptation will be assessed. The first one is the Statistics Criterion. The MMD (Gretton et al., 2008) and the CORAL (Sun et al., 2016) techniques will be used to statistically align the synthetic domain to the real domain. On the other hand, CyCADA (Hoffman et al., 2018) model will be used to adapt the synthetic domain to the real world domain.

5 Time planning

The Following figure 10 shows the time planning for the current research. Most of the weeks will consist of at least one day of writing and one day of coding.

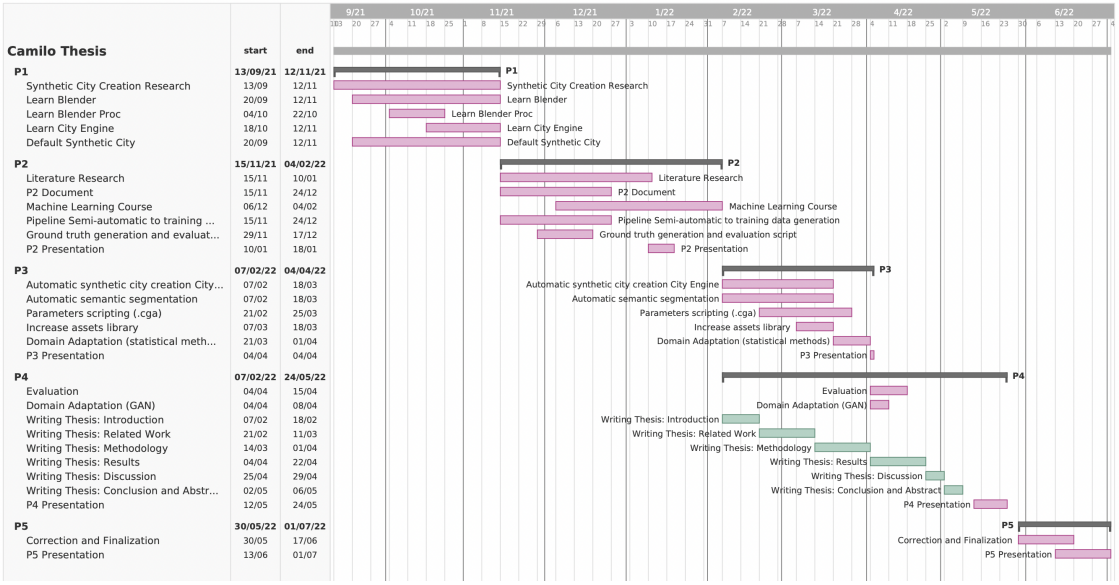

Figure 10: Gantt Chart for the time management of the thesis

6 Tools and datasets used

6.1 Tools

To create the synthetic city ESRI City Engine will be used. It is a 3D city model tool for different purposes such as urban planning and urban analysis. The student licence will be used. In addition, to prepare the city model for the semantic segmentation, the images and the DSM map will go through Blender. To recreate the images, the pipeline of Blender Proc will be used (Denninger et al., 2019). Python will be used for running the baseline approach Fuse-Net (Hazirbas et al., 2015; Mulder, 2020). For the evaluation and domain adaptation, python scripts are made. QGIS is used for the creation of the ground truth.

6.2 Datasets

For the creation of the the real training set and the test set, true orthophotos of 10cm of resolution will be given by READAR. These true orthophotos are composed by 4 bands; RGB and one additional band indicating if the pixel was interpolated in the creation of the true orthophoto. Additionally a DSM maps are also given by READAR. The ground truth will be made from the BGT of the Netherlands and the ISPRS (ISPRS, 2020) and INRIA (Maggiori et al., 2017) datasets.

Furthermore, ESRI library will be used for the creation of the synthetic city assets of textures.

References

N. Audebert, B. Le Saux, and S. Lefèvre. Beyond RGB: Very high resolution urban remote sensing with multimodal deep networks. *ISPRS Journal of Photogrammetry and Remote Sensing*, 140:20–32, 6 2018. ISSN 0924-2716. doi: 10.1016/J.ISPRSJPRS.2017.11.011.

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. doi: 10.1109/TPAMI.2016.2644615.

M. Denninger, M. Sundermeyer, D. Winkelbauer, Y. Zidan, D. Olefir, M. Elbadrawy, A. Lodhi, and H. Katam. BlenderProc. 10 2019. URL `https://arxiv.org/abs/1911.01911`.

F. Fleuret. Deep learning, 12 2021.

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. 4 2017. URL `http://arxiv.org/abs/1704.06857`.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

A. Gretton, K. Borgwardt, M. J. Rasch, B. Scholkopf, and A. J. Smola. A Kernel Method for the Two-Sample Problem. 5 2008. URL `http://arxiv.org/abs/0805.2368`.

C. Hazirbas, L. Ma, C. Domokos, and D. Cremers. FuseNet: Incorporating Depth into Semantic Segmentation via Fusion-based CNN Architecture. Technical report, 2015.

J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell. CyCADA: Cycle-Consistent Adversarial Domain Adaptation. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1989–1998. PMLR, 11 2018. URL `https://proceedings.mlr.press/v80/hoffman18a.html`.

ISPRS. ISPRS Semantic Labeling, 12 2020.

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. Technical report, 2016.

S. Khan, B. Phan, R. Salay, and K. Czarnecki. ProcSy: Procedural Synthetic Dataset Generation Towards Influence Factor Studies Of Semantic Segmentation Networks. Technical report, 6 2019. URL `https://uwaterloo.ca/wise-lab/procsy`.

F. Kong, B. Huang, K. Bradbury, and J. Malof. The Synthinel-1 dataset: a collection of high resolution synthetic overhead imagery for building segmentation. 1089, 2019. doi: 10.1007/978-3-030-29930-9. URL `http://link.springer.com/10.1007/978-3-030-29930-9`.

Y. Lecun, Y. Bengio, and G. Hinton. Deep learning, 5 2015. ISSN 14764687.

M.-Y. Liu and O. Tuzel. Coupled Generative Adversarial Networks. 6 2016. URL `http://arxiv.org/abs/1606.07536`.

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), 2017. ISSN 2072-4292. doi: 10.3390/rs9060522. URL https://www.mdpi.com/2072-4292/9/6/522.

E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez. Can Semantic Labeling Methods Generalize to Any City? The Inria Aerial Image Labeling Benchmark. Technical report, 2017. URL https://hal.inria.fr/hal-01468452.

A. E. Mulder. MSc thesis in Geomatics for the Built Environment Semantic Segmentation of RGB-Z Aerial Imagery Using Convolutional Neural Networks. Technical report, TU Delft, Delft, 2020.

S. I. Nikolenko. Synthetic Data for Deep Learning. Technical report, 2021. URL http://www.springer.com/series/7393.

PDOK. Basisregistratie Grootschalige Topografie (BGT), 12 2021.

O. Ronneberger, P. Fischer, and T. Brox. U-Net: Convolutional Networks for Biomedical Image Segmentation. In H. Navab Nassirand, W. M. Joachimand Wells, and A. F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI*, pages 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4.

G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez. The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes. Technical report, 2016.

C. Rosenberger, S. Chabrier, H. Laurent, and B. Emile. Unsupervised and supervised image segmentation evaluation. In *Advances in Image and Video Segmentation*, pages 365–393. IGI Global, 2006. ISBN 9781591407539. doi: 10.4018/978-1-59140-753-9.ch018.

A. Rozantsev, M. Salzmann, and P. Fua. Beyond Sharing Weights for Deep Domain Adaptation. 3 2016. doi: 10.1109/TPAMI.2018.2814042. URL http://arxiv.org/abs/1603.06432http://dx.doi.org/10.1109/TPAMI.2018.2814042.

S. Saito, T. Yamashita, and Y. Aoki. Multiple object extraction from aerial imagery with convolutional neural networks. *Journal of Imaging Science and Technology*, 60(1), 1 2016. ISSN 19433522. doi: 10.2352/J.ImagingSci.Technol.2016.60.1.010402.

S. Sankaranarayanan, Y. Balaji, A. Jain, S. N. Lim, and R. Chellappa. Learning from Synthetic Data: Addressing Domain Shift for Semantic Segmentation. 11 2017. URL http://arxiv.org/abs/1711.06969.

B. Sun, J. Feng, and K. Saenko. Return of Frustratingly Easy Domain Adaptation. Technical report, 2016. URL www.aaai.org.

K. Tempfli, G. C. Huurneman, W. H. Bakker, L. L. F. Janssen, W. F. Feringa, A. S. M. Gieske, K. A. Grabmaier, C. A. Hecker, J. A. Horn, N. Kerle, F. D. van der Meer, G. N. Parodi, C. Pohl, C. V. Reeves, F. J. A. van Ruitenbeek, E. M. Schetselaar, M. J. C. Weir, E. Westinga, and T. Woldai. *Principles of remote sensing : an introductory textbook*. ITC Educational Textbook Series. International Institute for Geo-Information Science and Earth Observation, Netherlands, 2009. ISBN 978-90-6164-270-1.

M. Wang and W. Deng. Deep Visual Domain Adaptation: A Survey. 2 2018. doi: 10.1016/j.neucom.2018.05.083. URL http://arxiv.org/abs/1802.03601http://dx.doi.org/10.1016/j.neucom.2018.05.083.

Y. Xu, L. Wu, Z. Xie, and Z. Chen. Building Extraction in Very High Resolution Remote Sensing Imagery Using Deep Learning and Guided Filters. *Remote Sensing*, 10(1), 2018. ISSN 2072-4292. doi: 10.3390/rs10010144. URL https://www.mdpi.com/2072-4292/10/1/144.

X. Yuan, J. Shi, and L. Gu. A review of deep learning methods for semantic segmentation of remote sensing imagery. *Expert Systems with Applications*, 169:114417, 5 2021. ISSN 0957-4174. doi: 10.1016/J.ESWA.2020.114417.

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, 1 2016. ISSN 1047-3203. doi: 10.1016/J.JVCIR.2015.10.012.