Semantic Segmentation of Roof Superstructures

P2 Thesis proposal — MSc Geomatics

Irène Apra

1st supervisor: Ken Arroyo Ohori
2nd supervisor: Giorgio Agugiaro
External supervisors (Technical University of Munich):
Bruno Willenborg
Sebastian Krapf

January 26, 2022

1 Introduction

Remote Sensing (RS) refers to spatial data acquired remotely through various sensors, such as cameras, scanners or radars (Janssen 2004). During the last decades, these datasets are gaining in quality, quantity and availability thanks to technological advances. Needs for interpretation tools are expanding accordingly. A growing analysis technique involves Deep Learning (DL), a branch of Artificial Intelligence (AI) allowing to extract information from large data input with little manual work. DL has been increasingly applied to RS data, especially for land-cover and land-use classification through semantic segmentation of aerial images, and has demonstrated satisfying results (Yuan et al. 2021).

Data processing is necessary to elaborate more structured and therefore usable data. As a result, digital representations of the world (or "digital twins") can be generated, useful for various purposes such as urban planning, environmental analysis, disaster management etc. Semantic 3D city models are representing virtual urban environments, including building geometries but also descriptive semantic attributes. The open data model and exchange format CityGML defines four Levels of Details (LOD) for object depiction (OGC 2012), which have been further subdivided by Biljecki et al. 2014 (figure 1). Each of these refined LODs corresponds to different application purposes and needs therefore to be clearly defined. Some usages require more advanced geometries whereas others benefit from lighter representations (Peters et al. 2021).

Generating 3D models automatically is a challenging exercise, especially when it reaches high LODs. Aiming at automating laborious work, DL methods offer therefore high potentials for this task. Several approaches can be considered, such as edge detection to create a roof partition as conceptualised by Peters et al. 2021. Another approach is to detect roof installations through DL algorithms, which, once located, can be modelled in 3D through extrusion which matches elevation data. Such detailed models, capturing roof installations bigger than 2.5 m^2, correspond to LOD2.2 according to Biljecki et al. 2014. Application potentials of such models are various: energy demand estimation, roof insulation assessment by detecting irregularities and thermal bridges, urban wind flow analysis, runoff and noise diffusion modelling. Additionally, these models can be employed for urban planning, to assess the age and condition of buildings through their architectural characteristics (dormers, chimneys) or the potential for constructions to be vertically extended. Finally, they can be used to support energy transition, helping to determine roof surface suitable and available for photovoltaic (PV) or solar-thermal installations, which convert sunlight into electricity and heat respectively.

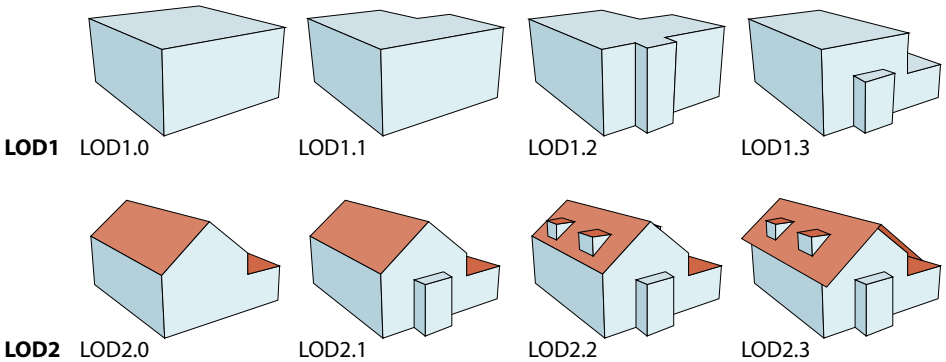

Figure 1: Level of Details 1 and 2 with their refinements, adapted from Biljecki et al. 2014

1.1 Motivation of the work

The need for automatic 3D modeling solutions and ML potential for this purpose inspired this research, which is carried out within an existing pipeline, developed since 2018 at the Chair of Automotive Engineering of the Technical University of Munich. The pipeline has been developed to provide a tool assessing PV potential of buildings in the German state of Bavaria. The project intends to support the development of sustainable charging stations for electric vehicles: solar stations could be regularly installed in cities if relevant buildings contribute to PV power generation. Therefore, our work benefits from existing implementation framework and datasets (including training data), while aiming at improving the tool.

The whole pipeline includes, among other aspects (cf. section 2.8.1), the evaluation of the physical suitability of roofs for PV panel installation based on their geometrical characteristics. The latter consists of roofs' azimuth and estimation of their available surface. Currently, both are independently estimated by means of Deep Neural Networks (DNN) applied to RGB aerial images. The azimuth categorises the orientation of the roof segment into 16 classes: North, East, South, West, and their variations. To estimate the roof surface available, the network detects installations on the roof - such as chimneys, dormers, existing PV installations, etc., which are further called "roof superstructures" - that are deduced from the total roof surface.

Since the pipeline uses basic DNN architectures applied to 2D data, the research motivation is to refine it by fusing 3D data sources for the segmentation of roof superstructures. This would improve the geometrical assessment of the pipeline. Additionally, the superstructures detected will be modelled in 3D in a simplified form, so the semantic 3D model, already available, can be upgraded from LOD 2.0 to LOD2.2, according to the classification of Biljecki et al. 2014.

1.2 Problem statement and hypothesis

This research aims at clarifying the added value of 3D data to detect roof superstructures through DL methods. For this, relevant 3D data is used which requires exploration of different height data. Additionally, different network architectures are tested, chosen after inspecting the state-of-the-art.

The hypothesis is that superstructures should be described by 3D data such as Light Detection And Ranging (LiDAR) point clouds and, therefore, integration of such data should improve the network performance. However, it is expected that each type of superstructure detected will benefit differently from it: for instance, chimneys are higher-profile obstacles than solar panels which will therefore be less detected through height differences.

1.3 Organisation of the document

Next section defines the main terms, presents relevant works related to DL and its application to aerial imagery combined with other data types, as well as the status of DL in the frame of the existing pipeline for PV-potential assessment. The third section introduces the research question and sub-questions. Then, the methodology is presented, followed by an organisational timeline and a description of the tools and datasets used.

2 Related work

This section introduces works related to neural network architectures that help us building a solution to the research problem. First of all, the main terms related to DL are defined, and the mathematical background for understanding the functioning of neural networks is introduced. Then, we present some types of network architectures related to our research,

including multimodal ones. Finally, we give an overview of the works achieved for the PV-assessement project and how it has applied DL so far. To conclude, we indicate how our approach is built upon what has been presented.

2.1 Deep Learning - Context and definitions

Artificial intelligence (AI) can be considered as a branch of computer science, aiming at simulating, extending and expanding human intelligence (Shi and Zheng 2006) and making machines that can reproduce human behavior based on the human-intelligence understanding. The applications are very wide, including robotics, voice and image recognition as well as natural language processing (Niu et al. 2016). The growth of research on AI exploded from the 1990's, which can mainly be explained by the increase of computational power (ibid.).

Artificial Neuron Networks (ANN), often just called Neural Networks (NN), were developed to build such "intelligent" machines. These networks are based on the biological neural network functioning (Gupta 2013). They can have various applications, including engineering, and can be used for pattern recognition or image segmentation as it is used in the case of this research. In supervised Machine Learning (ML), the model learns from its previous experience, obtained through labelled data, in order to output an answer (eg, classification tasks, assigning a category to the data); whereas in unsupervised ML, the model outputs a result simply based on the input, with no previous knowledge (eg, clustering tasks, grouping data based on common characteristics). We focus on the first type.

2.1.1 Neural Network

A NN is composed of neurons (or "units") implementing functions to map an input with an output. More precisely the input, comparable to a synapse, is multiplied by a weight computed with a mathematical function, whereas another function - the activation - determines the output of the neuron (Gupta 2013). A succession of neurons forms a "Neural Network". In the case of supervised ML, these are trained, so that their functions provide the desired output when fed with an input. The units located at a similar depth in the network form a hidden layer, whereas the first and last layers of the network are respectively called input and output layers (figure 2). When a network is composed of numerous layers it is called a "Deep Neural Network", from where derives the term "Deep Learning". The training data, usually manually labelled, allows the network to figure out the functions of the nodes mapping best the input x to the correct output $\hat{y}$ (Ng 2021).

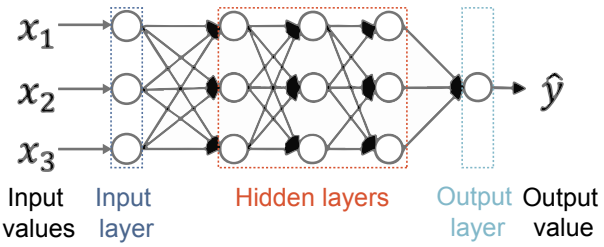

Figure 2: Standard Neural Network, adapted from Ng 2021

Different sorts of networks have been developed for different aims, and adapted to various input types (sound, images, etc.). One of the purposes is image classification, also called "semantic segmentation", which assigns an object-class probability value to each pixel on an image. Minaee et al. 2020 provides a review of the architectures which have been developed for this task until 2019.

2.1.2 Training, validation and test datasets

In the case of supervised ML, a "training dataset" is necessary to train the network and fit the model. Additionally, a verification and a test datasets are commonly used to improve and assess the network performance. The validation dataset is a sample of examples, independent from the training dataset, used to evaluate the fit of the model on the training dataset. It is still used in development stage and allows to adjust the network's higher-level hyperparameters. The test dataset, also independent from the two previous ones, allows to evaluate the final model and illustrates various real-world cases the model would face.

2.2 Mathematical background

DNN are often described as "black boxes" since it is not possible to apprehend what happens in each of his neurons. However, DNN are able to figure out the best parameter values through calculus. Mathematical aspects allow the practitioner to define blocks composing a network architecture, to assess its performance and rectify its parameters. Therefore, we introduce fundamentals that will help understanding the following presentation of some network architectures. This subsection is based on Ng 2021.

2.2.1 Neuron weight and bias

In an ANN, a layer is composed of several neurons which are connected to all or only part of the neurons of the next layer. For each neuron a weight and a bias are applied to the input, resulting in z, before computing the output through an activation function a (figure 3). The weight w and bias b are learnable parameters, where b is a constant.

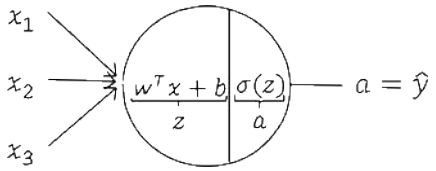

Figure 3: Operations occurring in a neuron; source: Ng 2021

2.2.2 Activation functions

The activation function acts like a synapse for the next layer, by activating it with its output. These activations can either be linear or non-linear: in early age of DL, the sigmoid function σ was usually implemented (function 1) whereas nowadays, depending on the application, rectifiers are rather used, such as the Rectified Linear Unit (ReLU). The latter does not modify the result if it is positive or outputs zero if it is negative.

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{1}$$

In the last layer, the function usually differs since it yields the network output, i.e. the final classification result. For binary classification, logistic regression is commonly used (based on a sigmoid function), whereas for multi-class outputs, softmax regression is used, generalising the logistic function to several dimensions. The softmax function normalizes the output to a probability distribution.

2.2.3 Cost and loss functions

The loss error function allows to assess how well is the algorithm performing, by defining the error between the prediction $\hat{y}$ and the ground truth y. The loss function L is defined for a single training example, whereas the cost function J does it for the entire training set, averaging the sum of the loss function applied to each training example.

The cost function can be defined in several ways. In case of a regression model, estimating a variable value based on the other, we can use a regression cost function: eg, the mean error (ME), the mean squared error (MSE) or the mean absolute error (MAE). A convex function can be used to enhance the optimisation problem, such as the logistic loss function, illustrated bellow with its corresponding cost function J:

$$L(\hat{y}, y) = -(ylog(\hat{y}) + (1-y)log(1-\hat{y})), \quad \text{and} \quad J(w,b) = \frac{1}{m}\sum_{i=1}^{m} L(\hat{y}^{(i)}, y^{(i)}) \tag{2}$$

where $\hat{y}$ is the predicted label, y is the ground truth, m the number of training examples, w and b the parameters considered.

To estimate the performance of the network on multi-classes, the cross-entropy (CE) function is commonly used, or binary cross entropy in case of only two possible output classes. In all cases, the aim is to determine the parameters w and b that minimise the overall cost function J. For this, w and b are initialised, and the model is trained in order to learn these parameters by converging to a global optimum.

2.2.4 Optimisation algorithms

For converging to this optimum, gradient descent algorithm can be applied. It is composed of a for- and backpropagation step which goes backwards through the layers, computing the gradients of the loss function J with respect to the weight parameters on the whole training set. The derivatives are used to compute the slope of the cost function J at the current state of the parameters. The parameters can then be accordingly incremented or decremented, allowing steps to be taken in direction of the steepest descent, until convergence. For controlling the steps taken between each elevation of gradient descent, the parameter α, called "learning rate", is used.

In logistic regression the two parameters are updated as follows:

$$w := w - \alpha\frac{dJ(w,b)}{dw}, \quad \text{and} \quad b := b - \alpha\frac{dJ(w,b)}{db}, \tag{3}$$

where α is the learning rate, $\frac{dJ(w,b)}{dw}$ is the derivative of J with respect to w and $\frac{dJ(w,b)}{db}$ is the derivative of J with respect to b.

Other optimisation algorithms include Stochastic Gradient Descent (SGD), gradient descent with momentum, Root Mean Square propagation (RMSprop) and the Adaptive Moment estimation (Adam). The latter, widely used, combines the effects of gradient descent with momentum and RMSprop. Although several parameters are involved, the learning rate α is common to them and needs to be adjusted by the practitioner.

2.3 Convolutional Neural Networks

In Computer Vision (CV) domain, DL has been extensively used for image recognition tasks such as image classification (predicting a label per image) and object detection (figuring out the location of an object and drawing a bounding box around it). However, images provide very large input features, and it would require too much computational power to train such

a network. Therefore, convolutional networks (*ConvNets*) have been developed, based on three types of layers: the Convolutional layer (*Conv*), the Pooling layer (*Pool*) and the Fully Connected layer (*FC*).

2.3.1 Convolutional Layer

The first type of layer relies on the "convolution operation" which "convolves" a filter on the pixels of the image, i.e. by sliding this window ("filter" or "kernel") from a pixel position to another. The filter - usually of size *3x3*, *5x5* or *7x7* - contains values allowing to detect a specific kind of feature on the image. The operation applied between the input pixel and the filter, denoted by an asterisk in figure 4, consists of an element-wise multiplication followed by summation of the values obtained.

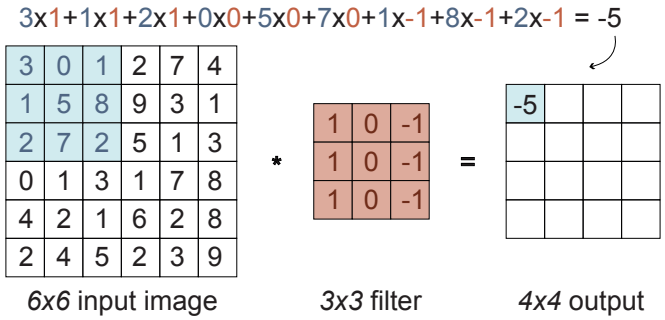

Figure 4: Convolution operation; adapted from Ng 2021

In figure 5, the *3x3* filter allows to detect vertical edges. The intuition is that the filter detects contrasts between left (bright pixels) and right side (dark pixels) of the central pixel considered. If this filter is rotated by 90 degrees, it would allow detection of horizontal edges. Similarly, a lot of different types of filter might be used to detect different features, and the filters' values can be learnt by the network, instead of entered manually, to be as efficient as possible.

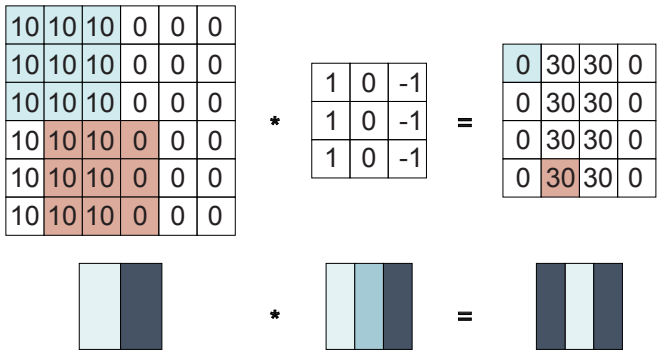

Figure 5: Convolution operation using a vertical edge detection filter; adapted from Ng 2021

The basic convolution operation can be altered by means of padding and strides. The first one, by adding some additional row(s) of pixel to the input image, prevents it from shrinking when implementing convolution. A padding size p of one would add one row of pixels all along (figure 6). Therefore, a convolution can be "valid" if no padding is applied; or "similar" ("same" convolution) if the output image is of equal size than the input. On the other hand, the stride s impacts the sliding distance of the kernel between two operations. The higher the stride value is, the smaller is the output size.

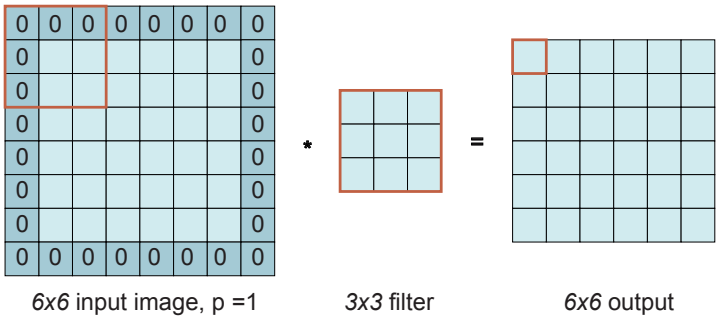

Figure 6: Padding operation, example of padding value p equal to one

A convolution operation can also be applied to a multi-dimensional input (eg, RGB images). In that case, the filter has the same number of channels (i.e. dimensions) than the input. The number of filters used results in the number of channels in the output (figure 7). Consequently, an RGB image convolved with a single *3x3* filter would result in an image of one channel. To turn the operation into a *ConvNet*, the convolved activations from the previous layer should be applied a non-linearity function (eg, ReLU) and a bias, resulting in the final layer output. The output from different filters are stacked together to form a *Conv* layer.

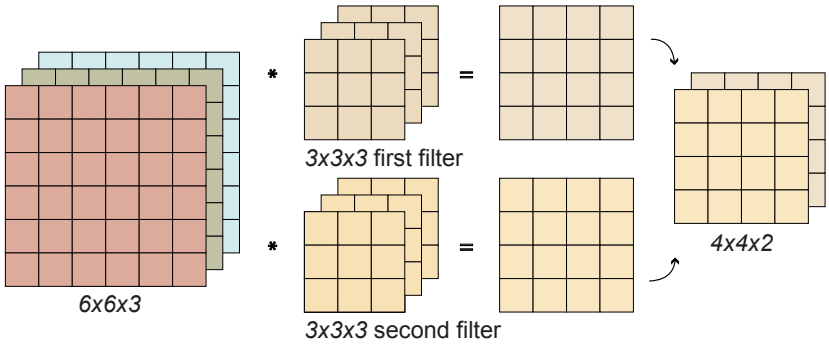

Figure 7: 3D Convolutional operation using two filters; adapted from Ng 2021

To summarise, a *Conv* layer is characterised by the filter size, its padding, its stride and the number of filters used, with each filter having the same number of channels as the input. The values of the filters stacked together correspond to the weights (used for linear operation), and the bias is unique per filter.

2.3.2 Pooling Layer

Although a CNN may only use *Conv* layers, *Pool*s and *FC*s are usually also implemented. *Pool*s allow to reduce the representation size, therefore speeding up the computation and making feature detection more robust. Essentially, the input is divided into regular regions, based on a filter size and stride value, and the maximum value (*Max Pooling*, figure 8) or average value (*Average Pooling*) per region is kept. The intuition is that by keeping a statistically meaningful value per piece, features detected are preserved. *Pool*s have some hyperparameters (filter size and stride) but no parameters to learn, and the operation preserves the number of dimensions. By convention a *Conv* layer followed by a pooling operation form together one layer, since only layers having weights are counted.

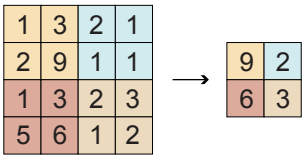

Figure 8: Example of MaxPooling; adapted from Ng 2021

2.3.3 Fully Connected Layer

An *FC* layer contains units similar to the ones described earlier, which are densily connected to the activations output from the previous layer. *FC*s have therefore a lot of parameters. On the contrary, *Conv* layers use shared parameters (that are the values in a filter) and their connections are sparse, making them advantageous. As illustrasted in figure 9, *Conv* layers followed by *Pool* layers, with *FC* at the end, before a softmax activation function, is a common pattern for CNNs.

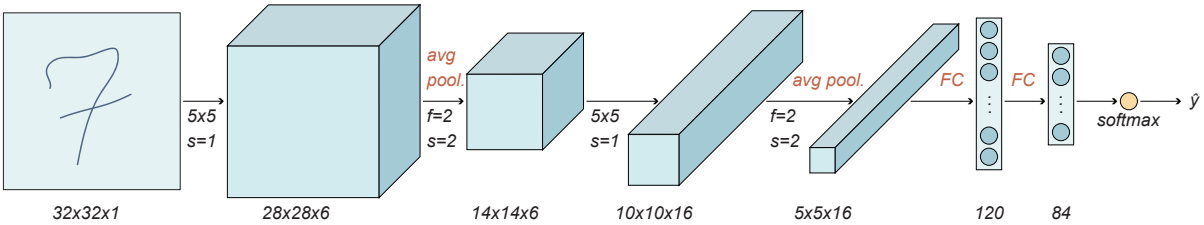

Figure 9: Typical ConvNet; adapted from LeNet-5 network (Le Cun et al. 1998)

2.4 CNN architectures

The three building blocks, *Conv, Pool and FC*, are combined together to form architectures, which CV community is researching to reach better performances. Some architectures relevant for this research are introduced.

First of all, LeNet-5, illustrated in figure 9, has been developed for document recognition purpose (Le Cun et al. 1998). In 2012, AlexNet extended LeNet-5 into a deeper CNN, achieving great performance on image classification tasks (Krizhevsky et al. 2012). A few years later, the very deep network VGG-16 was created for large-scale image recognition tasks, 16 referring to the number of weighted layers the network contains (Simonyan and Zisserman 2015). The uniformity of this architecture - with systematic rates of size decrease and channel increase between layers - makes it attractive. However, it contains a large number of parameters to train.

2.4.1 ResNets

Very deep networks are hard to train because of the problem of vanishing and/or exploding gradients, which makes unexpectedly the training error getting worse in deep layers. To tackle this issue, skip connections have been researched, allowing to pass the activations functions from a layer to another one, deeper in the network. Residual blocks using this concept (figure 10) have been proposed by He et al. 2015. A *ResNet* is built by stacking together several of these blocks.

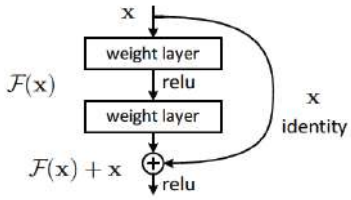

Figure 10: Building block using residual learning; source: He et al. 2015

2.4.2 Network in Network

The concept of "Network in Network" was developed by Lin et al. 2014, inspiring many architectures later on, including the Inception architecture (Szegedy et al. 2014). The idea is to use a *1x1* convolutional layer (also called "Network in Network"), allowing to modify the number of channels according to the number of filters, but not the size (height and width) of an image (figure 11). If the number of filters corresponds to the number of input channels, then *1x1 Conv* is useful to learn more complex functions.

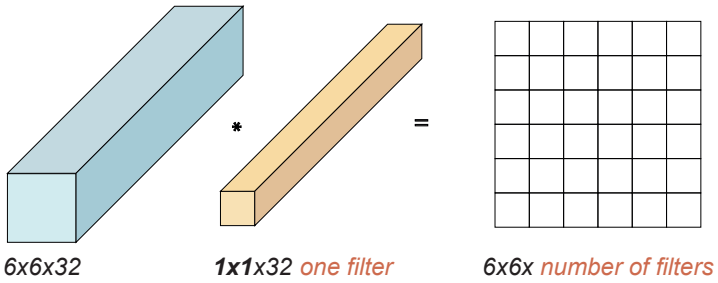

Figure 11: *1x1* Convolutional layer; adapted from Ng 2021

2.5 Encoder-decoder network architectures

Until now we have described classification networks, that go deeper through feature detection layers in order to output a label class for the whole image. However, for semantic segmentation, the final output should have a similar size than the input image, assigning a probability per class per pixel. For this purpose, a second "decoding" part is stacked to the "encoding" part of the network. The first part refers to the encoding of image pixels into low-level feature-maps detected through filters (e.g. lines, textures) resulting in the increase of channel number; whereas decoding refers to the opposite process, decoding the features back to the image size, allowing high-level, global information detection (e.g. object classification as a whole). The decoder part projects the encoder information on the pixel space to output a dense classification.

The Fully Convolutional Network (FCN), extending *ConvNets* image classification purpose to segmentation tasks, has been defined to combine deep semantic information and shallow appearance information through the use of skip connections (Long et al. 2015). The authors introduce the concept of deconvolution to upsample the image back to its original size, allowing pixel-wise prediction (figure 12). The final prediction layer is linked to lower layers, so that local dense predictions are coherent with the global structure.

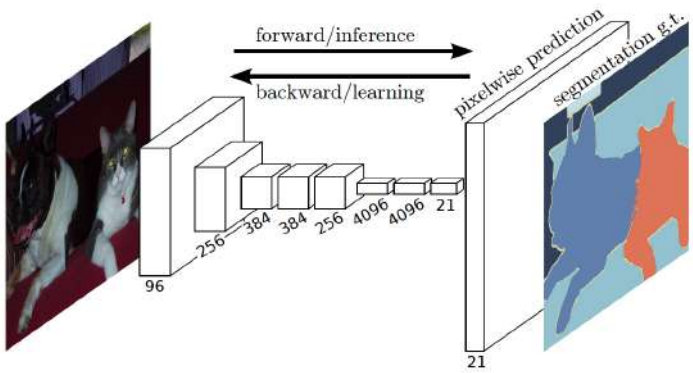

Figure 12: FCN architecture, allowing pixel-wise prediction; source: Long et al. 2015

2.5.1 U-Net

U-Net, built upon FCN, elaborates on the decoding part. It uses transpose convolutions for the upsampling process, making the network symmetric (Ronneberger et al. 2015). The shape of the figure 13, illustrating U-Net functioning, has given its name to the network. For simplification purpose, each rectangle represents the height (vertical axis) and number of channel of the image (horizontal axis) whereas its width is omitted. Therefore, the evolution of the image size and its number of channel can easily be understood.

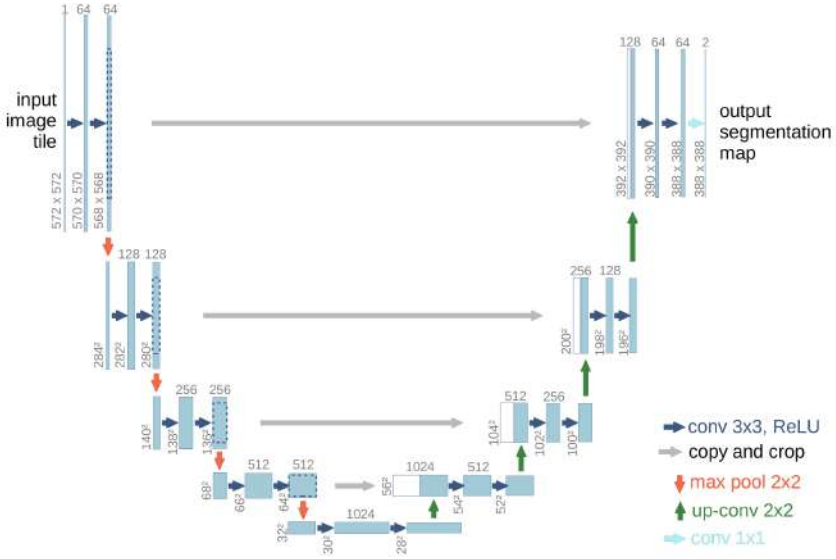

Figure 13: U-Net architecture; source: Ronneberger et al. 2015

2.6 Multimodal network architectures

Another aspect of the research on network performances involves multimodalities. Those refer to different datatypes: image, sound, video, etc. Their combination within NNs opens new and countless possibilities to improve networks' performance since these diverse inputs offer complementary information. Multimodal NNs offer various possibilities: eg, modelling two modalities jointly, generating a modality from another one or using a modality as label for the second one (eg, the sound of a video, labelled by the corresponding images) (Dean 2017).

In the case of RS and 3D city modeling, modalities include aerial images, hyperspectral data (HIS, capturing light within wider bands of the electromagnetic spectrum than RGB

images), Light Detection And Ranging (LiDAR) point clouds and semantic 3D models. As reviewed by Yuan et al. 2021, several works have researched multimodalities to improve semantic segmentation of aerial images. Some of them use HIS data fusion to enhance land-cover classification (Roy et al. 2020) or roof material identification (Nimbalkar et al. 2018).

2.6.1 Height data fusion to CNNs

LiDAR point clouds provide discrete 3D data acquired by measuring the range of light pulses emitted by a laser-scanning device. Usage of those points in ML can either be achieved through 3D architectures including 3D CNN - which is an application of CNN theory to 3 dimensions (Guo et al. 2020) -, or through converting it to 2D information by means of raster representation. For this, points can be projected to depict a "height map" in case of parallel data or "depth map" in case of perspective data. The first implementation case allows 3D shape classification, object detection or points segmentation (Guo et al. 2020), whereas the second allows to enhance 2D segmentation results by providing complementary data. Techniques for the second approach are explored in this subsection.

Combining RGB and height data can be achieved through data fusion, implemented for the first time by Hazirbas et al. 2017 for RGB-Depth camera images. Gu et al. 2021 distinguishes two types of data fusion to the network architecture. The most frequent one adopts a bottom-up approach (figure 14, left), where the additional datatype is encoded through another branch to the network. The top-down approach (figure 14, right) feeds encoded information of the different branches to the corresponding decoding depth.

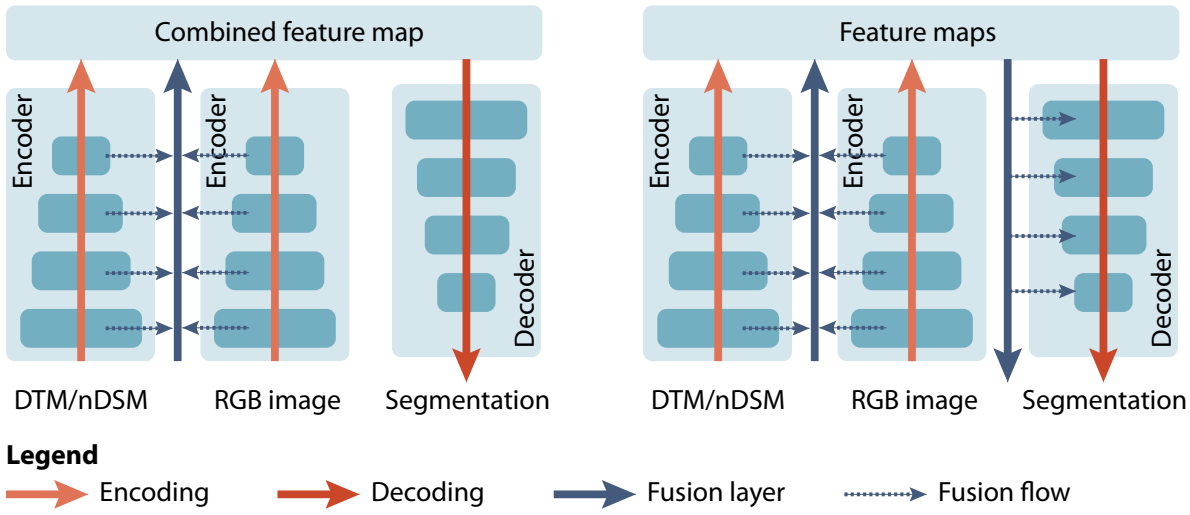

Figure 14: Two different fusion approaches: left, the bottom-up and right, the top-down fusion methods; adapted from Gu et al. 2021

2.6.2 Bottom-up Fusion Network architectures

Hazirbas et al. 2017 implemented depth data fusion for the purpose of perspective image segmentation. The author tested stacking depth information on an additional channel, and fusing it, using a parallel encoder (or "auxiliary branch") which activations are aggregated to the ones of the main RGB branch before decoding (figure 15). Data fusion yielded better results.

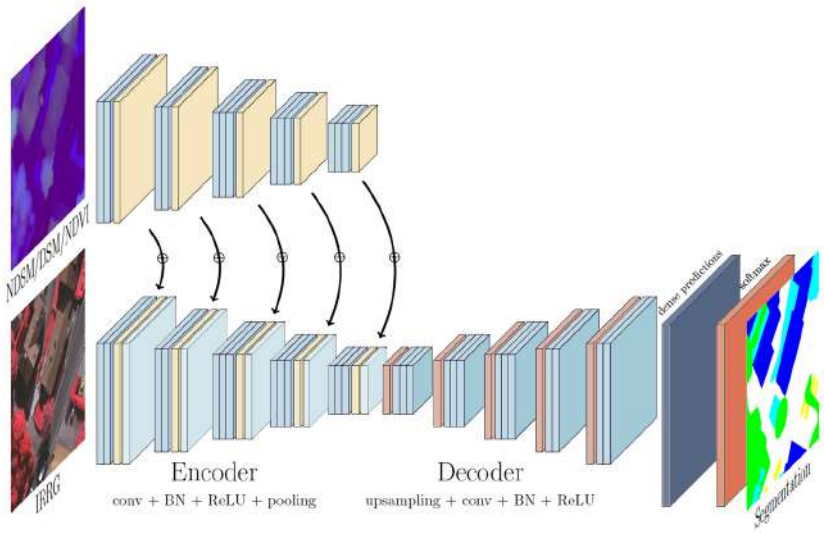

Figure 15: FuseNet architecture encoding auxiliary data to the main datasource; source: Audebert et al. 2018 adapted from Hazirbas et al. 2017

Similarly, for land-cover classification, other authors tested height data fusion to RGB images: Digital Terrain Model (DTM), Digital Surface Model (DSM) and normalised DSM (nDSM) which depicts height data relative to ground level. For instance, Virtual-FuseNet was developed to fuse nDSM, DSM or NDVI (difference between visible and near-infrared) data to aerial images, tackling the asymmetry issue of FuseNet (Audebert et al. 2018). The main and auxiliary branches are not differentiated anymore thanks to the use of a third "virtual" branch, aggregating activations of both branches through fusion layers (figure 16).

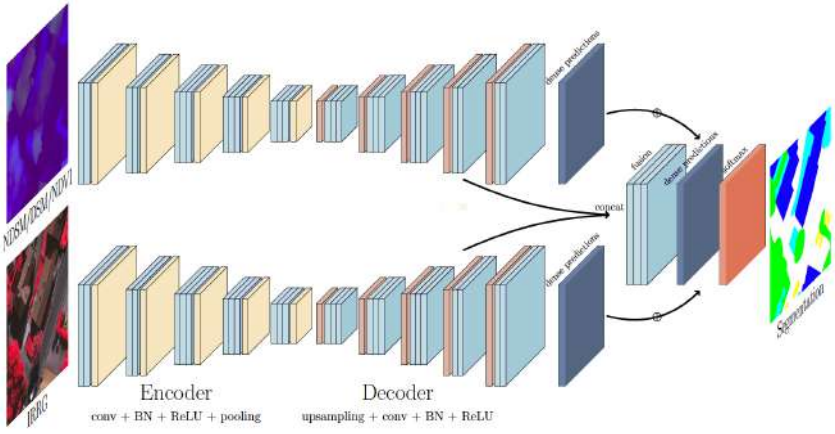

Figure 16: Virtual-FuseNet architecture, a symmetric version of data fusion; source: Audebert et al. 2018

Another author compared land-cover classification performance of a network using respectively additional Digital Terrain Model (DTM), Digital Surface Model (DSM), normalised DSM (nDSM) and standardised nDSM (StdnDSM) fused to RGB images (figure 17, Zhou et al. 2019). In StdnDSM, height values are rescaled within the bounds of RGB values present on the image (0 to 255), so both RGB and height information have an equal impact on the network output. Fusion of StdnDSM yielded better results.

Accordingly, Mulder 2020 concluded that height information fused to the network, rather than stacked, improves semantic segmentation results on RGB images for specific classes (i.e.

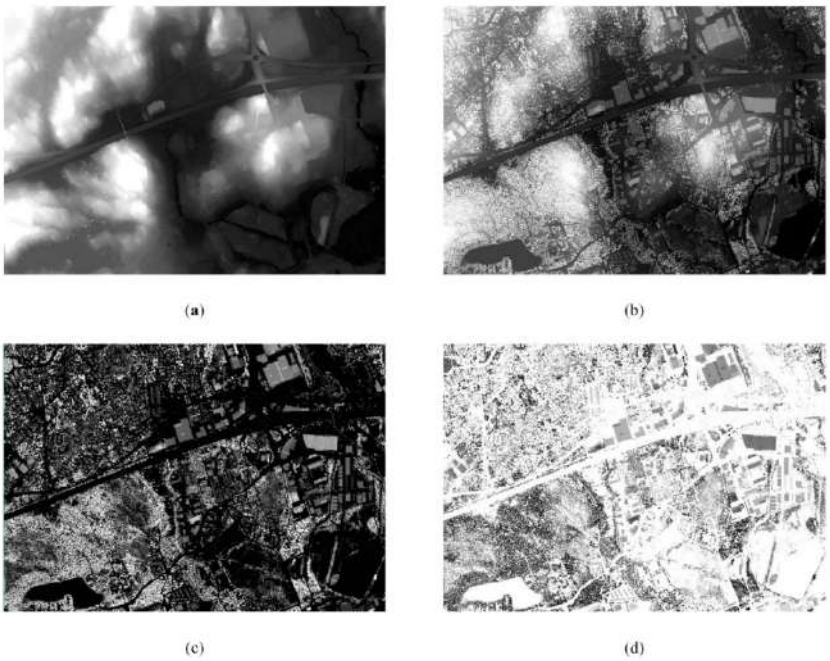

Figure 17: Comparison of height data sources: a) DTM, b) DSM, c) nDSM and d) standardised nDSM; source: Zhou et al. 2019

buildings but not water class). Best results are reached using relative height, obtained through pixel-level subtraction of DTM from DSM.

2.6.3 Top-down Fusion Network architectures

Gu et al. 2021 developed a light-weighted Top-down Pyramid Fusion Network (TdPFNet), tackling the issue of information loss during decoding phase, happening in usual multi-branches fusion approaches, as well as the complexity of such networks. Their solution allows to guide the fusion of low-level texture information (in shallow layers) with the high-level semantic of deeper layers. This is possible through the use of fusion layers at each depth of the network, gathering information from the different encoder-branches and feeding them to the corresponding decoding depth (figure 14, right). Finally, cumulated high and low feature information from multi-levels are arranged and fused to obtain dense segmentation results. Tests yield best segmentation results for Potsdam land-use classification when fusing Open Street Map (OSM) dataset, and showed that TdPFNet is less complex than other state-of-the arts models (FuseNet, V-FuseNet, etc.), being therefore easier to train.

2.7 Roof superstructure detection and modelisation

Although (multimodal) CNNs have been extensively researched for land-cover and land-use semantic segmentation and prove satisfying results, they are seldom used for roof segmentation tasks and even less for roof superstructure detection. This can be explained by the necessity of high-resolution input data and the need of large training datasets, which require extensive preparation work.

2.7.1 Input data resolution and availability

Roof superstructure detection through DL method requires (Very) High-Resolution (V-HR) images and/or LiDAR data. In remote sensing, HR to VHR satellite images involves a representation from 30cm to 5m per pixel according to Kraetzig 2021. In the finer case, a chimney would approximately be represented by 4 pixels, illustrating the need for VHR images for roof superstructure detection. Since the launch of the first HR satellite sensor in 1999 (Ikonos), finer multispectral and panchromatic images are acquired each year (Marcello and Eugenio 2019), yielding corresponding research and development of various use-cases. Marcello and Eugenio 2019 gathers such works and mentions, among other processes, image segmentation and classification that can be applied to change detection, land monitoring and urban mapping. Although VHR images exist for multiple areas, they are not always publicly available nor provided as orthophotos, making them complex to use as combined to other datasets.

Additionally, LiDAR data acquisition requires extensive work since airborne laser scanners have to fly over the whole area considered. Then, points processing and classification is necessary to make them usable for different purposes. Availability and quality of such datasets are therefore limited or very recent. Despite these challenges, European countries are increasingly investing to acquire them: e.g. in the Netherlands, the fourth national LiDAR dataset (AHN4) is available since 2021 with a resolution of 10 to 12 points/m^2 (AHN 2022); in France (13 times larger than the Netherlands), a LiDAR national dataset is being acquired since 2021, fully available by 2026 with an average resolution of 10 points/m^2 (IGN 2022); in Germany, to our knowledge, no national LiDAR dataset is available, but Bavaria proposes one of at least 4 points/m^2 (LDBV 2021b) that will be used for this research.

2.7.2 Roof superstructure detection through CNNs

First of all, semantic segmentation on RGB images at building scale has been used for geometrical purposes: to detect building footprints (Wei et al. 2020), extract roof segments (surfaces with their orientation) (Lee et al. 2019) or roof edges (Ahmed and Byun 2019). The latter can be used to model buildings in 3D: Alidoost et al. 2020 use DL to extract nDSM and rooflines from mere RGB input to reconstruct LOD2 buildings.

Secondly, DL methods applied on roof images have been used for specific application cases: e.g. PV-mapping by detecting solar panels already installed on roofs (Ioannou and Myronidis 2021), or roof material analysis using various spectral data input (Nimbalkar et al. 2018). However, to our knowledge, no DL method has been developed to detect simultaneously all types of installations on roofs. Therefore, no training data has been found for such purpose and this work uses data manually labelled by the team since 2018.

2.7.3 Roof superstructure modelisation

Once detected, roof superstructures can be modelled in 3D. To describe the nuances of such a model, the four LODs described by CityGML standards have been refined by Biljecki et al. 2014. Each subdivided LOD depicts a model suitable for different application purposes. 3D city models are available in Europe at different LODs, but mostly LOD1 or LOD2 (e.g. Bavaria). Availability of LOD2.2 models that include dormers and chimneys, is rare on large areas since complex to generate automatically. To our knowledge, the Netherlands is the only country that has such a model nationally available (3D BAG) (tudelft3d 2021). A 3D cadastral dataset has been researched, downloadable since 2021 in 3 different LODs - 1.2, 1.3 and 2.2 (tudelft3d 2021). Peters et al. 2021 describes the method used for automatic generation of 3D

buildings, based on cadastral and AHN LiDAR point cloud datasets: although the process requires building edges detection, no ML technique is used.

Photogrammetry is a method commonly used to reconstruct buildings in 3D from photographs taken at different angles (e.g. Google Earth). However, Kudinov 2021a proposes a ML-based method developed upon R-CNN (Gkioxari et al. 2020) to reconstruct 3D-mesh buildings including roof superstructures from mere orthophotos (Kudinov 2021a, Kudinov 2021b). Input data include RGB images and nDSM which efficiency are tested separately and combined. The latter provides best results (figure 18), allowing even to model overhangs, therefore reaching LOD2.3 representation according to Biljecki et al. 2014. However, it has a tendency to smooth out the edges, is sensible to image noises and to shadows. As a result, the use-cases of such a mesh model - unreliable and complex - are reduced. Nonetheless, it illustrates the potential of ML for accurate and detailed 3D modelisation in the near future.

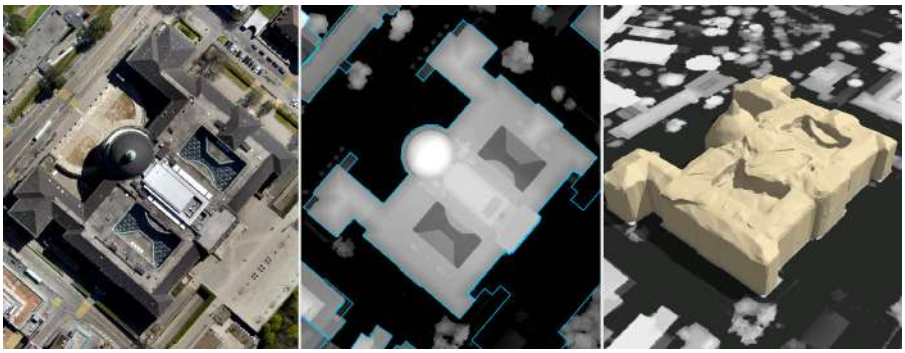

Figure 18: 3D mesh-reconstruction (right) obtained by fusing RGB aerial image (left) and nDSM (middle); source: Kudinov 2021b

2.8 Current status of the pipeline

Finally, we examine how ML theory is implemented in the existing pipeline for buildings' PV-potential assessment, and present different works carried out in the frame of the project. Implementation details and results until 2021 are provided by Krapf et al. 2021.

2.8.1 Pipeline overview

As illustrated in figure 19, the whole pipeline for PV-potential assessment includes estimation of physical suitability of roofs for PV panel installation based on radiation data, and the geometrical potential of roofs based on their azimuth and the estimation of available surface. Furthermore, technical aspects of PV installations as well as electricity prices on the market are considered.

The geometrical assessment consists of two aspects: roof azimuth and roof surface available. Each is estimated by means of a DNN applied to RGB aerial images. The azimuth categorises the orientation of the roof into 16 classes: North, East, South, West and their variations. To estimate the roof surface available, the network detects installations on the roof (i.e. superstructures) that are deduced from the total roof surface.

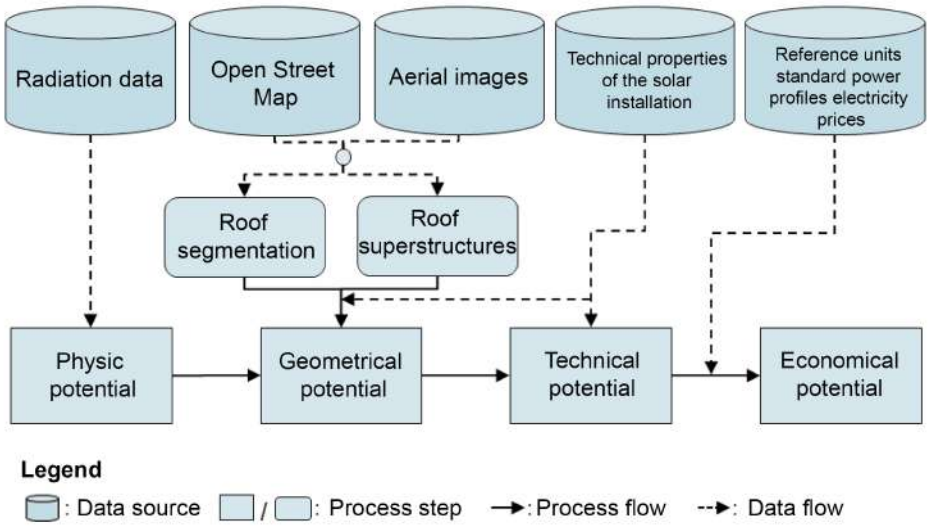

Figure 19: General pipeline for PV potential assessment; adapted from Prummer 2021

2.8.2 Geometrical assessment implementation

First of all, segmentation of the roofs into azimuth (figure 20) has been implemented by Kemmerzell 2020, based on Lee et al. 2019, using a U-Net architecture with a ResNet-152 backbone. The same author has also implemented PV technical and economic assessment.

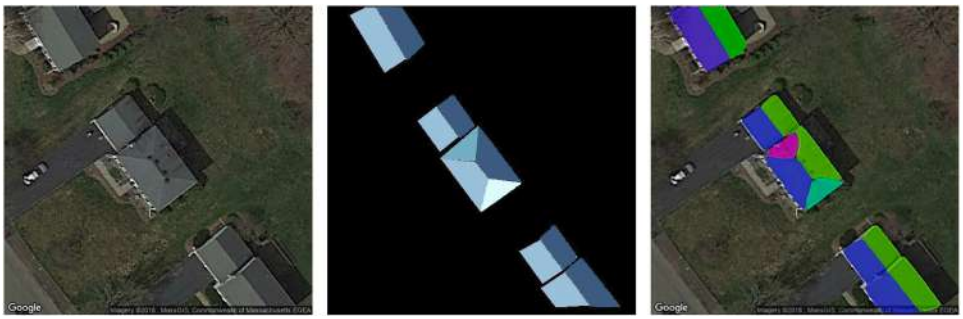

Figure 20: From left to right: Google aerial photographs, segments ground truth and segments detected by the network; source: Syed 2020

Later, the detection of roof superstructures is implemented by Syed 2020, using a U-Net architecture and resulting in a single and/or multi-class segmentation as illustrated in figure 21. Both CNN use images obtained through Google API as input and focus on the city of Wartenberg in Bavaria from which training, validation and test datasets are generated. Since no training datasets is publicly available for roof superstructures, the latter has been manually labeled by the team and its extension and improvement is an ongoing process since 2018 (Krapf et al. 2021). Next, Prummer 2021 carries out a data-centric approach to improve the network performance.

Finally, in parallel with this work, an automatic generation of training data for roof azimuth segmentation based on the 3D model available is researched. For this, several encoder backbones are evaluated within a U-Net architecture implementation. Additionally, a thesis is carried out to model in 3D the chimneys and dormers detected by the network for superstructure segmentation.

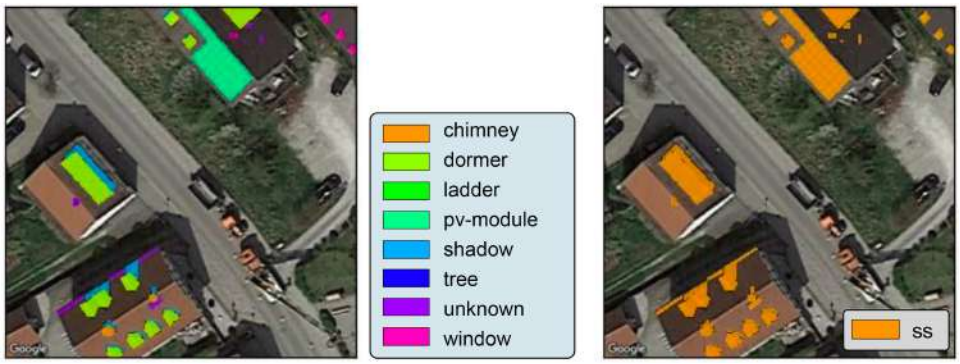

Figure 21: Roof superstructure detection into multi-class (left) and single-class (right); source: Prummer 2021

2.9 Conclusion on the State-of-the-Art

To conclude with, this research aims to improve the current U-Net implementation of the pipeline for PV-potential assessment by migrating to an architecture allowing data fusion. For this, state-of-the-art architectures will be used and their performance with diverse inputs will be evaluated for roof superstructure segmentation. This purpose, which requires finer information and image understanding than land-cover or land-use classification, has been hardly explored until now.

3 Research questions

The main research question is the following: How can 3D data be most efficiently integrated to a Convolutional Neural Network on RGB aerial images to improve the semantic segmentation of roof superstructures?

Some sub-questions regarding the input, architectures and output are considered:

- Which types of height information are the most relevant to integrate for detecting roof superstructures?
- How should the input data be processed? How to deal with different resolutions of available data (point cloud and images) and temporal mismatches?
- How should the existing pipeline be modified to fuse height data efficiently to the network? Which architecture gives better results?
- How to evaluate the results accuracy for single- and multi-class superstructure segmentation, and the added value of each input respectively?

4 Methodology

4.1 Research process overview

Regarding network architectures, the performance of several of them (FuseNet, Virtual-FuseNet) will be compared for roof superstructure segmentation. Regarding height data, nDSM data will be generated and standardised to match the extents of RGB values. Several ways to generate nDSM will be tested to match the resolution of the aerial images based on different interpolation methods. Additionally, fusion of several data sources will be

tested, eg. nDSM and slope per segment. Finally, the polygons detected will be extruded in 3D, matching the height information obtained from LiDAR dataset, and combined to the existing LOD2 model. Figure 22 depicts the overall research process.

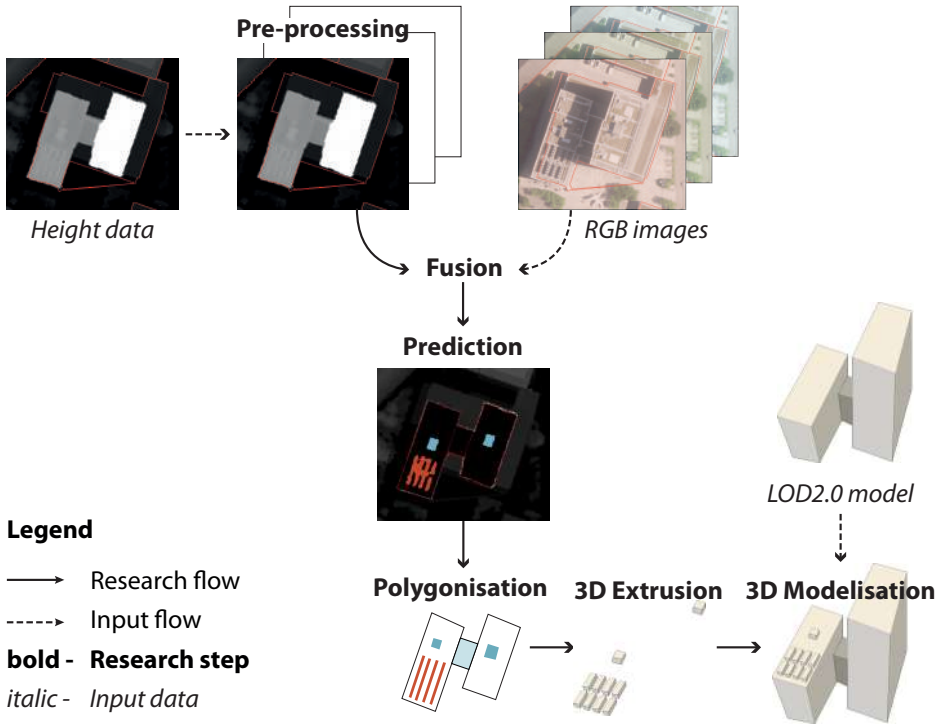

Figure 22: Overview of the research process; base-image adapted from Kudinov 2021b

4.2 Research steps

4.2.1 Exploration of the current network and its results

First of all, the existing network is trained and run on the test dataset available to get acquainted with the pipeline and current results. The results per class are analysed using existing scripts (figure 25), computing the proportion of pixels per class and their accuracy compared to the ground truth. This step requires installing packages and libraries, which might be done by means of a Docker container that should be created.

4.2.2 Height data preparation and assessment

The datasets available are processed in order to generate height data. First, nDSM is obtained by subtracting DTM to DSM. Interpolation methods are used to obtain the same resolutions as the aerial images. Secondly, the slope information per segment can be generated based on the LOD2 model. Additionally, information content of the data generated can be evaluated and compared thanks to the Information Entropy (IE) criteria as described by Zhou et al. 2019.

4.2.3 Training data

The existing labelled data is examined and the relevant superstructure classes are kept: *shadow* and *tree* classes might be discarded. The infrared band can help discarding trees whereas the shadows should be learnt by the network through height data input. Additionally, the classes relevant for final 3D modelling, i.e. *chimneys* and *dormers*, should be considered. Therefore, the

ladder and *window* classes could be discarded as well, whereas the *pv-module* class is relevant for PV-potential analysis. As a result, the final classes include at least *chimney*, *dormer*, *pv-module* and *other*.

4.2.4 Height data integration

Different network architectures (Fuse-Net, V-FuseNet) are implemented for height data fusion to RGB. The model is fitted thanks to the existing training data and run for the test dataset using different height inputs and architectures.

4.2.5 Evaluation and comparison of the results

The results are analysed thanks to existing scripts computing the Intersection over Union (IoU) and the confusion matrices to determine the Overall Accuracy (OA). These criteria allow to assess the quality of predictions as compared to the ground truth. Tables are generated to compare the different architectures and height inputs.

4.2.6 3D model generation

The detected superstructures are extruded in 3D, and added upon the existing gml model available in LOD2.0. The extrusions, including semantics describing the superstructure type, allow to determine the roof area still available for PV installations. The enhanced model can be further used for various applications.

5 Time planning

The research phase (figure 23) is composed of a groundwork including a machine learning course (Ng 2021) and extended literature review, followed by an exploration of the current network, training data and segmentation classes. The latter are refined to fit the research purpose. Then, height data is prepared and fusion architectures are implemented. Finally, an evaluation of the results is carried out.

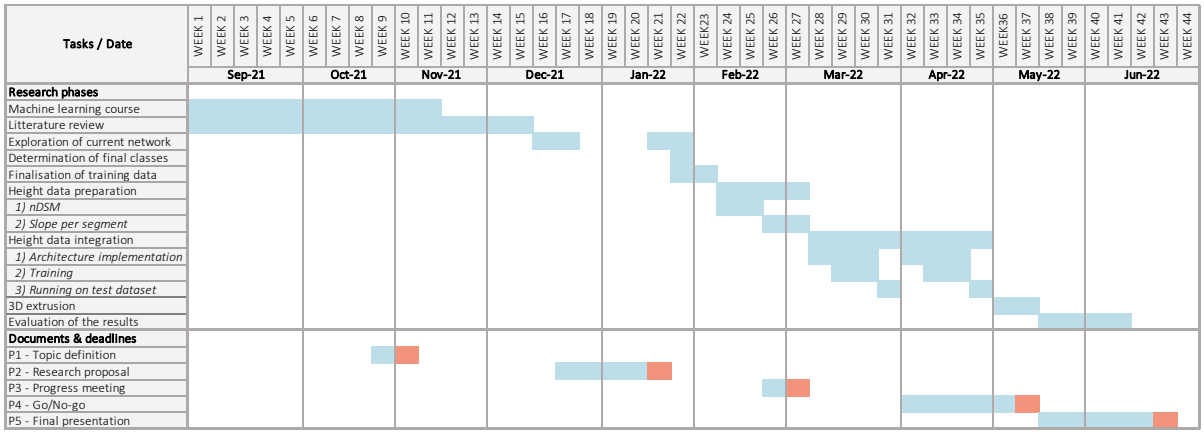

Figure 23: Gantt chart

6 Tools and datasets used

6.1 Datasets available

The project aims at using datasets openly available on LDBV 2021c for Bavaria, so the research can be scalable. However, the LOD2 3D model, DTM and DSM, not publicly available, were specially provided for this project by the State Office for Digitization, Broadband and Surveying (LDBV). The datasets available include the following:

- Aerial Photography (AP): They can be downloaded through Web Map Service (WMS) from the Bavarian geoservices portal (LDBV 2021c). Although they are provided as true orthophotos, their resolution is only of 20cm/pixel, which is less than Google Application Programming Interface (API) images (10cm/pixel), that are not ortho-rectified.
- LiDAR point cloud: According to LDBV 2021b, it has been acquired for Bavaria between 2011 and 2021. The training data area is part of the Taufkirchen zone, which has been acquired in 2012. The resolution is of at least 4 points/m^2. The points are divided into seven classes (last pulse) and eight classes (first pulse) including ground, building, water bodies, objects and bridges (LDBV 2016).
- Semantic LOD2 model (CityGML): Existing since 2012 for Bavaria, it is generated from the building footprints of the cadastral map combined with the point cloud. The latest model was generated between 2016 and 2021 depending on the location in Bavaria. The buildings are fitted to one of the 13 typical roofshapes described in the ALKIS information system (LDBV 2021a). If height information is missing due to temporal data acquisition mismatches, the roofs are modelled flat with a height relative to the roof size (more or less than 25 m^2).
- Digital Terrain Model (DTM): Generated from the LiDAR dataset (ground points), it has a resolution of 1m/pixel (LDBV 2021c).
- Digital Surface Model (DSM): It is generated from the LDBV APs (20cm/pixel resolution) using photogrammetry techniques. The DSM resolution is of 40cm/pixel, mismatching the DTM resolution (LDBV 2021c).

Since Google APs of 2018 have been used for labeling the training data until now, there might be some temporal mismatches between them and height data available. Spatial mismatches can also occur since these APs are not true orthophotos.

6.2 Training data

The data has been annotated on the city of Wartenberg, Bavaria (figure 24), since it offers standard data conditions and would ensure the project is scalable for the whole state. About 1800 buildings have been annotated during the last three years, based on Google API image dataset of 2018. Since the labeling process is expensive, these annotations are used for this research. The same labels cannot be used on 2021 images due to different image distortions caused by the slight angle photos were taken from. Additionally, lighting conditions would impact the network performance if we would use 2021 APs as test data. Therefore, 2018 dataset should be used.

Labeled classes include *solar panels*, *windows*, *chimneys*, *dormers*, *ladder*, *trees* (hanging over the roofs), *shadows* and *unknown* (figure 21). The labeled dataset includes both masks (*512x512* pixel images, one per roof) and csv files containing annotation details:

Figure 24: Bavaria with Wartenberg localisation in red (left), Wartenberg outlines (right)

multipolygon geometry, latitude-longitude coordinates (as of Google API), an id per mask and the id of the roof segment area it belongs to (stored in another csv). Masks contain an array per pixel assigning a value for each of the eight classes.

The training set include 60 percents of theses annotations, which are isolated in the algorithm by generating a random point in Wartenberg and including buildings within a growing radius until reaching 60 percents of them. Then, 20 percents is used for validation and 20 for model testing.

6.3 Programming tools

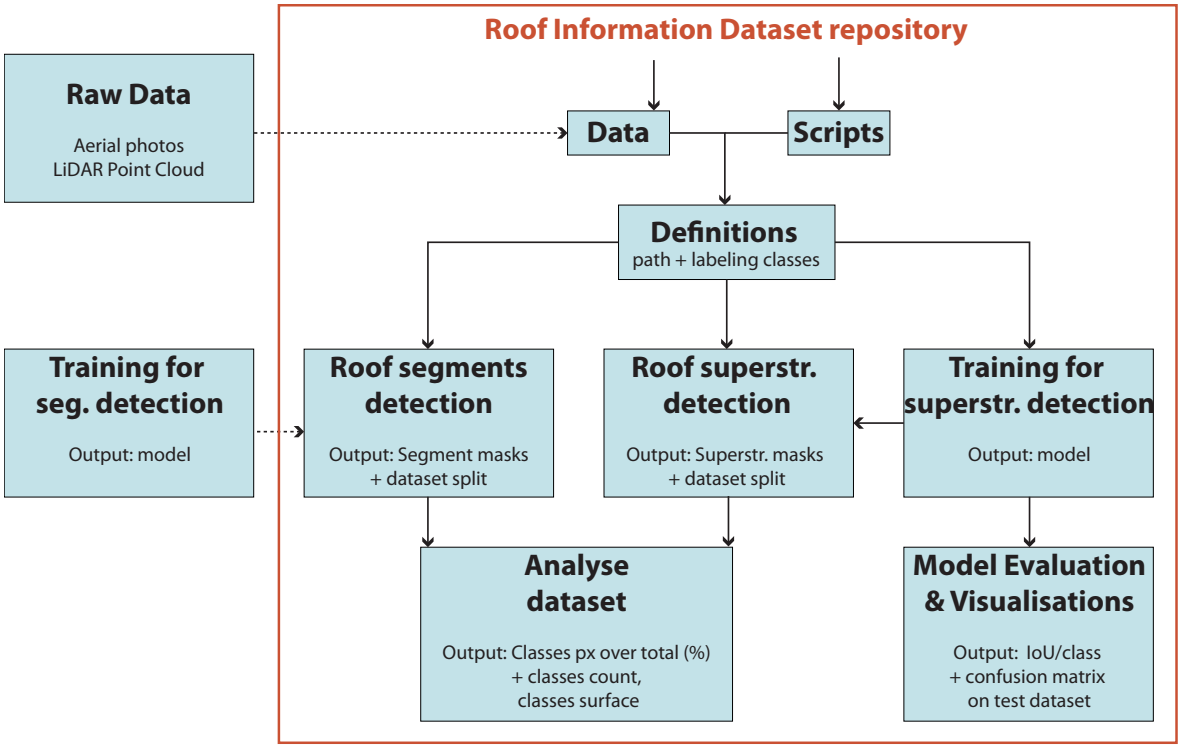

Figure 25: Organisation of the working repository

The existing pipeline for geometric assessment of roofs includes several python scripts which purpose and relations are described in the figure 25. This repository is hosted on gitLab. The DL scripts, including training and prediction, use Keras, an API built upon Tensorflow, in python language (Chollet et al. 2015). The U-Net model is imported from there.

6.4 Computational power

Training the model can be done on a remote server, using one of the three Graphics Processing Units (GPU) available through the university. The connection to the GPU server is possible using the remote session manager MobaXterm which generates, among other services, the necessary SSH keys. The data, generated using an existing script, and the training script are uploaded to the server and the requirements installed (or a Docker image built), before running the script from the command line. As a result, an h5 file containing the converged parameters is generated, which can later be used - associated to the corresponding ANN - for predictions on the test data.

References

Ahmed, Aneeqa, and Yung-Cheol Byun. 2019. "Edge Detection using CNN for Roof Images." In *Proceedings of the 2019 Asia Pacific Information Technology Conference*, 75–78. APIT 2019. New York, NY, USA: Association for Computing Machinery, January 25, 2019. ISBN: 9781450366212, accessed January 16, 2022. https://doi.org/10.1145/3314527.3314544. https://doi.org/10.1145/3314527.3314544.

AHN. 2022. "Actueel Hoogtebestand Nederland (AHN)." Accessed January 14, 2022. https://www.ahn.nl/.

Alidoost, F., et al. 2020. "Y-SHAPED CONVOLUTIONAL NEURAL NETWORK FOR 3D ROOF ELEMENTS EXTRACTION TO RECONSTRUCT BUILDING MODELS FROM A SINGLE AERIAL IMAGE." *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* V-2-2020 (August 3, 2020): 321–328. ISSN: 2194-9050, accessed December 15, 2021. https://doi.org/10.5194/isprs-annals-V-2-2020-321-2020. https://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/V-2-2020/321/2020/.

Audebert, Nicolas, et al. 2018. "Beyond RGB: Very high resolution urban remote sensing with multimodal deep networks." *ISPRS Journal of Photogrammetry and Remote Sensing*, Geospatial Computer Vision, 140 (June 1, 2018): 20–32. ISSN: 0924-2716, accessed December 10, 2021. https://doi.org/10.1016/j.isprsjprs.2017.11.011. https://www.sciencedirect.com/science/article/pii/S0924271617301818.

Biljecki, Filip, et al. 2014. "Formalisation of the level of detail in 3D city modelling." *Computers, Environment and Urban Systems* 48 (November 1, 2014): 1–15. ISSN: 0198-9715, accessed December 20, 2021. https://doi.org/10.1016/j.compenvurbsys.2014.05.004. https://www.sciencedirect.com/science/article/pii/S0198971514000519.

Chollet, François, et al. 2015. "Keras (github repository)." https://github.com/fchollet/keras.

Dean, Victoria. 2017. *Multimodal Deep Learning*. Accessed December 13, 2021. https://www.youtube.com/watch?v=6QewMQT4iMM.

Gkioxari, Georgia, et al. 2020. "Mesh R-CNN." *arXiv:1906.02739 [cs]* (January 25, 2020). Accessed December 12, 2021. arXiv: 1906.02739. http://arxiv.org/abs/1906.02739.

Gu, Yuhang, et al. 2021. "Top-Down Pyramid Fusion Network for High-Resolution Remote Sensing Semantic Segmentation." *Remote Sensing* 13, no. 20 (January): 4159. Accessed December 14, 2021. https://doi.org/10.3390/rs13204159. https://www.mdpi.com/2072-4292/13/20/4159.

Guo, Yulan, et al. 2020. "Deep Learning for 3D Point Clouds: A Survey." *arXiv:1912.12033 [cs, eess]* (June 23, 2020). Accessed January 17, 2022. arXiv: 1912.12033. http://arxiv.org/abs/1912.12033.

Gupta, Neha. 2013. "Artificial Neural Network." *IISTE* 3 (1). Accessed December 4, 2021. https://core.ac.uk/reader/234686479.

Hazirbas, Caner, et al. 2017. "FuseNet: Incorporating Depth into Semantic Segmentation via Fusion-Based CNN Architecture." In *Computer Vision – ACCV 2016*, edited by Shang-Hong Lai et al., 10111:213–228. Cham: Springer International Publishing. ISBN: 9783319541808 9783319541815, accessed December 10, 2021. https://doi.org/10.1007/978-3-319-54181-5_14. http://link.springer.com/10.1007/978-3-319-54181-5_14.

He, Kaiming, et al. 2015. "Deep Residual Learning for Image Recognition." *arXiv:1512.03385 [cs]* (December 10, 2015). Accessed December 12, 2021. arXiv: 1512.03385. http://arxiv.org/abs/1512.03385.

IGN. 2022. "LiDAR HD: vers une nouvelle cartographie 3D du territoire." Accessed January 14, 2022. https://www.ign.fr/institut/lidar-hd-vers-une-nouvelle-cartographie-3d-du-territoire.

Ioannou, Konstantinos, and Dimitrios Myronidis. 2021. "Automatic Detection of Photovoltaic Farms Using Satellite Imagery and Convolutional Neural Networks." *Sustainability* 13, no. 9 (May 10, 2021): 5323. ISSN: 2071-1050, accessed January 16, 2022. https://doi.org/10.3390/su13095323. https://www.mdpi.com/2071-1050/13/9/5323.

Janssen, Lucas L. F. 2004. *Principles of remote sensing: an introductory textbok.* OCLC: 265702962. Enschede: ITC. ISBN: 9789061642275.

Kemmerzell, Nils. 2020. "Automatische Analyse des ökonomischen PV-Potenzials mittels GIS und Luftbildern - Automatic analysis of economic pv-potential via aerial images and GIS." Master thesis.

Kraetzig, Nikita Marwaha. 2021. "A Definitive Guide to Buying and Using Satellite Imagery." UP42 Official Website. Accessed January 16, 2022. https://up42.com/blog/tech/a-definitive-guide-to-buying-and-using-satellite-imagery.

Krapf, Sebastian, et al. 2021. "Towards Scalable Economic Photovoltaic Potential Analysis Using Aerial Images and Deep Learning." *Energies* 14, no. 13 (June 24, 2021): 3800. ISSN: 1996-1073, accessed January 14, 2022. https://doi.org/10.3390/en14133800. https://www.mdpi.com/1996-1073/14/13/3800.

Krizhevsky, Alex, et al. 2012. "ImageNet Classification with Deep Convolutional Neural Networks." In *Advances in Neural Information Processing Systems,* vol. 25. Curran Associates, Inc. Accessed December 13, 2021. https://papers.nips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html.

Kudinov, Dmitry. 2021a. "3D Buildings from Imagery with AI: Part 1." GeoAI, October 28, 2021. Accessed December 12, 2021. https://medium.com/geoai/3d-buildings-from-imagery-with-ai-fbbc1852e4dd.

———. 2021b. "3D Buildings from Imagery with AI: Part 2." GeoAI, October 28, 2021. Accessed December 12, 2021. https://medium.com/geoai/3d-buildings-from-imagery-with-ai-part-2-ef129dca6dc.

LDBV. 2016. *Laserdaten Beschreibung der Punktklassen (Laserdata Description of the point classes).* Accessed December 22, 2021. https://www.ldbv.bayern.de/file/pdf/10574/Laserdaten%20-%20Beschreibung%20der%20Punktklassen.pdf.

———. 2021a. "ALKIS - Geodaten online." Accessed December 22, 2021. https://geodatenonline.bayern.de/geodatenonline/seiten/dfkalkis_info.

———. 2021b. "BayernAtlas." https://geoportal.bayern.de/bayernatlas/?topic=ba&lang=de&bgLayer=atkis&w=100%25&h=500px&catalogNodes=11&layers=luftbild,KML%7C%7Chttps:%2F%2Fwww.geodaten.bayern.de%2Fdownload%2Fuebersicht_DGM%2FLaserscanningbefliegungen.kml%7C%7Ctrue,luftbild_parz,tk_by&E=723070.17&N=5417849.58&zoom=3.5166666666666435&layers_visibility=false,true,true,false.

LDBV. 2021c. "Landesamt für Digitalisierung, Breitband und Vermessung." Accessed December 21, 2021. https://www.ldbv.bayern.de/.

Le Cun, Yann, et al. 1998. "GradientBased Learning Applied to Document Recognition." Accessed December 27, 2021. http://vision.stanford.edu/cs598_spring07/papers/Lecun98.pdf.

Lee, Stephen, et al. 2019. "DeepRoof: A Data-driven Approach For Solar Potential Estimation Using Rooftop Imagery." In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2105–2113. KDD '19: The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. Anchorage AK USA: ACM, July 25, 2019. ISBN: 9781450362016, accessed December 20, 2021. https://doi.org/10.1145/3292500.3330741. https://dl.acm.org/doi/10.1145/3292500.3330741.

Lin, Min, et al. 2014. "Network In Network." *arXiv:1312.4400 [cs]* (March 4, 2014). Accessed December 27, 2021. arXiv: 1312.4400. http://arxiv.org/abs/1312.4400.

Long, Jonathan, et al. 2015. "Fully Convolutional Networks for Semantic Segmentation." *arXiv:1411.4038 [cs]* (March 8, 2015). Accessed December 12, 2021. arXiv: 1411.4038. http://arxiv.org/abs/1411.4038.

Marcello, Javier, and Francisco Eugenio. 2019. *Very High Resolution (VHR) Satellite Imagery: Processing and Applications.* OCLC: 1163815192. ISBN: 9783039217571, accessed January 16, 2022. https://openresearchlibrary.org/content/db813f58-5522-443e-a178-36eb0d48671c.

Minaee, Shervin, et al. 2020. "Image Segmentation Using Deep Learning: A Survey." *arXiv:2001.05566 [cs]* (November 14, 2020). Accessed December 8, 2021. arXiv: 2001.05566. http://arxiv.org/abs/2001.05566.

Mulder, Amber. 2020. "Semantic Segmentation of RGB-Z Aerial Imagery Using Convolutional Neural Networks." Master thesis. Accessed December 3, 2021. https://repository.tudelft.nl/islandora/object/uuid%3Ab936953b-4c73-4ce1-a897-7da4287ff79a.

Ng, Andrew. 2021. "Neural Networks & Deep Learning (Coursera)." Course. Accessed December 4, 2021. https://www.coursera.org/specializations/deep-learning.

Nimbalkar, Prakash, et al. 2018. "Optimal Band Configuration for the Roof Surface Characterization Using Hyperspectral and LiDAR Imaging." *Journal of Spectroscopy* 2018 (April 18, 2018): e6460518. ISSN: 2314-4920, accessed December 14, 2021. https://doi.org/10.1155/2018/6460518. https://www.hindawi.com/journals/jspec/2018/6460518/.

Niu, Jiqiang, et al. 2016. "Global Research on Artificial Intelligence from 1990–2014: Spatially-Explicit Bibliometric Analysis." *ISPRS International Journal of Geo-Information* 5, no. 5 (May): 66. Accessed December 4, 2021. https://doi.org/10.3390/ijgi5050066. https://www.mdpi.com/2220-9964/5/5/66.

OGC. 2012. "OGC City Geography Markup Language (CityGML) En-coding Standard." Accessed January 17, 2022. https://portal.ogc.org/files/?artifact_id=47842.

Peters, Ravi, et al. 2021. "Automated 3D reconstruction of LoD2 and LoD1 models for all 10 million buildings of the Netherlands." *arXiv:2201.01191 [cs, eess]* (December 30, 2021). Accessed January 14, 2022. arXiv: 2201.01191. http://arxiv.org/abs/2201.01191.

Prummer, Johanna. 2021. "Untersuchung eines datenzentrierten Ansatzes zur Dachaufbauerkennung mittels Deep Learning - Assessment of a Data-Centric Approach for Roof Superstructure Detection using Deep Learning." Master thesis, Technical University of Munich.

Ronneberger, Olaf, et al. 2015. "U-Net: Convolutional Networks for Biomedical Image Segmentation." Version: 1, *arXiv:1505.04597 [cs]* (May 18, 2015). Accessed December 3, 2021. arXiv: 1505.04597. http://arxiv.org/abs/1505.04597.

Roy, Swalpa Kumar, et al. 2020. "FuSENet: fused squeeze-and-excitation network for spectral-spatial hyperspectral image classification." *IET Image Processing* 14 (8): 1653–1661. ISSN: 1751-9667, accessed December 12, 2021. https://doi.org/10.1049/iet-ipr.2019.1462. https://onlinelibrary.wiley.com/doi/abs/10.1049/iet-ipr.2019.1462.

Shi, Zhong-Zhi, and Nan-Ning Zheng. 2006. "Progress and Challenge of Artificial Intelligence." *Journal of Computer Science and Technology* 21, no. 5 (September 1, 2006): 810. ISSN: 1860-4749, accessed December 4, 2021. https://doi.org/10.1007/s11390-006-0810-5. https://doi.org/10.1007/s11390-006-0810-5.

Simonyan, Karen, and Andrew Zisserman. 2015. "Very Deep Convolutional Networks for Large-Scale Image Recognition." *arXiv:1409.1556 [cs]* (April 10, 2015). Accessed December 12, 2021. arXiv: 1409.1556. http://arxiv.org/abs/1409.1556.

Syed, Khawaja Haseeb Uddin. 2020. "A Deep Learning Approach to Roof Superstructure detection for PV potential estimation." Master thesis, Technical University of Munich, November 11, 2020.

Szegedy, Christian, et al. 2014. "Going Deeper with Convolutions." *arXiv:1409.4842 [cs]* (September 16, 2014). Accessed December 15, 2021. arXiv: 1409.4842. http://arxiv.org/abs/1409.4842.

tudelft3d. 2021. "3D BAG." Accessed December 8, 2021. https://3dbag.nl/en/viewer.

Wei, Shiqing, et al. 2020. "Toward Automatic Building Footprint Delineation From Aerial Images Using CNN and Regularization." *IEEE Transactions on Geoscience and Remote Sensing* 58, no. 3 (March): 2178–2189. ISSN: 0196-2892, 1558-0644, accessed January 16, 2022. https://doi.org/10.1109/TGRS.2019.2954461. https://ieeexplore.ieee.org/document/8933116/.

Yuan, Xiaohui, et al. 2021. "A review of deep learning methods for semantic segmentation of remote sensing imagery." *Expert Systems with Applications* 169 (May 1, 2021): 114417. ISSN: 0957-4174, accessed December 10, 2021. https://doi.org/10.1016/j.eswa.2020.114417. https://www.sciencedirect.com/science/article/pii/S0957417420310836.

Zhou, Keqi, et al. 2019. "CNN-Based Land Cover Classification Combining Stratified Segmentation and Fusion of Point Cloud and Very High-Spatial Resolution Remote Sensing Image Data." *Remote Sensing* 11, no. 17 (January): 2065. Accessed December 12, 2021. https://doi.org/10.3390/rs11172065. https://www.mdpi.com/2072-4292/11/17/2065.