

This is the Title of my Thesis

Valdemar Edvard Sandal Rolfsen

Desember 2017

PROJECT

Department of Geomatics

Norwegian University of Science and Technology

Supervisor 1: The main supervisor

Supervisor 2: The co-supervisors (internal and external)

Preface

Here, you give a brief introduction to your work. What it is (e.g., a Master's thesis in RAMS at NTNU as part of the study program xxx and...), when it was carried out (e.g., during the autumn semester of 2021). If the project has been carried out for a company, you should mention this and also describe the cooperation with the company. You may also describe how the idea to the project was brought up.

You should also specify the assumed background of the readers of this report (who are you writing for).

Trondheim, 2012-12-16

(Your signature)

Ola Nordmann

Acknowledgment

I would like to thank the following persons for their great help during ...

If the project has been carried out in cooperation with an external partner (e.g., a company), you should acknowledge the contribution and give thanks to the involved persons.

You should also acknowledge the contributions made by your supervisor(s).

O.N.

(Your initials)

Remark:

Given the opportunity here, the RAMS group would recognize Professor Emeritus Marvin Rausand for the work to prepare this template. Some minor modifications have been proposed by Professor Mary Ann Lundteigen, but these are minor compared to the contribution by Rausand.

Executive Summary

Here you give a summary of your work and your results. This is like a management summary and should be written in a clear and easy language, without many difficult terms and without abbreviations. Everything you present here must be treated in more detail in the main report. You should not give any references to the report in the summary – just explain what you have done and what you have found out. The Summary and Conclusions should be no more than two pages.

You may assume that you have got three minutes to present to the Rector of NTNU what you have done and what you have found out as part of your thesis. (He is an intelligent person, but does not know much about your field of expertise.)

Contents

Preface	i
Acknowledgment	ii
Executive Summary	iii
1 Introduction	2
1.1 Background	2
1.2 Problem	3
1.3 Approach	3
1.4 Limitations	4
1.5 Outline	4
2 Theoretical background	6
2.1 Image segmentation	6
2.1.1 The Hough transform	6
2.1.2 Image segmentation using superpixels	8
2.2 Artificial Neural Networks	9
2.2.1 Feed-forward neural networks	9
2.2.2 Training a neural network	10
2.2.3 Activation functions	11
2.2.4 Loss functions	12
2.2.5 Hyperparameters	13
2.2.6 Convolutional neural networks	14
2.3 Height estimation using remote sensing	17
2.3.1 Synthetic Aperture Radar (SAR)	17
2.3.2 Light Detection And Ranging (LIDAR)	19
2.3.3 Height detection using shadows from imagery	19
3 Related Work	21
3.1 Standard methods for shape detection	21
3.1.1 Shape detection using the Hough Transform	21

CONTENTS	1
3.1.2 Simple Linear Iterative Clustering (SLIC)	21
3.2 The development of Convolutional neural networks	22
3.2.1 Early adaption of convolutional neural networks	22
3.2.2 Deep Convolutional Neural Networks	23
3.3 Semantic segmentation using convolution neural networks	27
3.4 Height estimation form a nadir perspective	29
3.4.1 Building height retrieval from VHR SAR Imagery	29
3.4.2 Height estimation from InSAR analysis	29
3.4.3 Height estimation using shadow measurement	30
4 Evaluating the methods	32
4.1 Available data	32
4.2 Amount of data required for analysis	33
4.3 Choosing a method	34
5 Conclusions	37
5.1 Summary and Conclusions	37
5.2 Discussion	37
5.3 Recommendations for Further Work	38
Bibliography	39

Chapter 1

Introduction

Semantic segmentation and feature extraction in remote sensing, has been a field of research for many decades. The ability to extract geospatial information directly from satellite imagery has been crucial for fields such as environmental and demographic research. Furthermore, a vast development within the field of machine learning and artificial intelligence the last years, has enabled many researchers to find useful applications of such algorithms within their own scientific fields.

With new satellite technology providing frequently updated imagery, with resolutions less than 0.5 meters, the amount of data that can be extracted is almost incomprehensible. This paper focuses on how automatic height estimation of ground objects can be achieved through remote sensing, semantic image segmentation and different height estimation techniques.

The main goal of this study is to investigate different methods for automatic detection and height retrieval of the roofs of oil tanks in Cushing, Oklahoma. By knowing the height difference between the tanks floating roofs and their actual height, it is possible to calculate their inventory.

1.1 Background

Satellites has been collecting earth observation data for decades. Since the satellite Explorer 6 took the first picture of the earth in 1959, millions of satellite images has been taken, processed and stored [Esa \(2009\)](#). This information however, has been very difficult to access, and even harder to analyze when accessible. However, during the last decade, the development of machine learning based methods for Earth Science applications has experienced a considerable leap forward [Lary \(2010\)](#).

In 2014 the first satellite in the new family of earth observation satellites, called the Sentinels, was launched from Kourou, French Guiana. Since then 7 different constellations, each consisting of two satellites, have been launched, and are now orbiting and monitoring the earths surface. The goal of these satellites is to produce a continuous stream of timely data for Europe's Copernicus program, which will be used for environmental monitoring. The different constellations have different missions when it comes to providing datasets for the Copernicus Service. While some provide very specific data, such as monitoring the earths atmosphere, other constellations provide more general data, such as multi-spectral, high-resolution imagery of the earths surface. In order to maximize the usage of these temporal datasets, they have all been provided free of charge to the public. Other satellite constellations such as SkySat ([Planet, 2017](#)) and WorldView ([DigitalGlobe, 2017](#)) provide even more accurate satellite imagery, with sub-meter resolutions. With these satellites providing frequently updated imagery over most parts of the earths surface, the amount of earth observation data that can be observed is huge.

This means that data that earlier required manual measurements in order to be analyzed, now can be gathered, processes and analyzed automatically at a high phase. One field where this can provide valuable information is within the oil industry.

Cushing Oklahoma is the major trading hub and famous price settlement point for West Texas Intermediate of the New York Mercantile Exchange. With inventory numbers not being continuously updated by Energy Information Administration (EIA), there exists a potential in retrieving these numbers automatically through remote sensing.

1.2 Problem

The paper aims to answer if it is possible, with todays methods and satellite technology to automatically provide frequently updated estimations of the total inventory of all the crude oil tanks in Cushing, Oklahoma.

1.3 Approach

This paper will focus on investigating theory and previous work related to semantic segmentation of satellite images and techniques for height detection using data collected through remote sensing. By combining these two elements of study, it is theorized that it can be possible to automatically locate and estimate the inventory of oil tanks in Cushing, Oklahoma.

Semantic segmentation

In terms of semantic segmentation the paper discuss different approaches ranging from standard methods, such as the Hough transform to newer, more advanced methods such as image segmentation using convolutional neural networks. The goal is to present the strengths and weaknesses of the different methods, in order to determine which one is most suitable to the task in hand.

Height estimation

There are also a large variety of different methods for height estimation using data from satellite imagery. While the methods investigated related to semantic segmentation only focus on multispectral images, both methods related to radar images and multispectral imaged are considered for height estimation. The methods that are investigated are SAR, InSar and shadow measurements.

1.4 Limitations

This study is conducted as a literature review, and does not present any implementation of the methods that are investigated.

1.5 Outline

The remaining part of this paper consists of the following chapters:

- Chapter 2. Theoretical background: This chapter gives a theoretical background of the different methods that are investigated later in the paper. It provides the basic knowledge required in order to understand the different methods presented in related work.
- Chapter 3. Related Work: In this chapter the previous work relevant for this paper is presented. It includes standard methods for image segmentation, an overview of the advances in convolutional neural networks and how they can be used for image segmentation. Furthermore, different works related to height estimation of buildings are presented.
- Chapter 4. Evaluating the methods: The focus in this chapter is to evaluate and compare the different methods that have been investigated in the previous chapters.

- Chapter 5. Conclusions, discussion, and ideas for further work: A conclusion is presented based on the previous chapters, the overall
- Bibliography

Chapter 2

Theoretical background

In this chapter we are going to investigate some of the key concepts related to pattern recognition, shape detection in imagery and height estimation from remote sensing.

2.1 Image segmentation

Identifying geometric shapes in computer vision has been a classical problem for decades. There are many theories related to what is the best way of detecting a particular shape in an image, with shapes defined as two dimensional features of an object that are invariant to scene factors, or whose variation can be modeled easily ([Moon et al., 2002](#)).

2.1.1 The Hough transform

In image analysis, the Hough Transform is a technique used for feature extraction. This technique uses a voting procedure from which object candidates are obtained as local maxima in a space constructed by the algorithm (called a accumulator space). Since the algorithm requires that the desired features are specified in some parametric form, the classical Hough transform is mostly used for the detection of regular curves (lines, circles ellipses, etc.).

The main idea of the algorithm is that each edge pixel contributes to a globally consistent solution, such as a curve. In order to detect a point's contribution to this solution, the algorithm performs a point-to-curve transformation from the cartesian image coordinate space, to a polar Hough parameter space. In the cartesian coordinate space, a line segment can be represented by [Equation 2.1](#).

$$x \cos(\theta) + y \sin(\theta) = r \quad (2.1)$$

In Figure 2.2 each point (x,y) on the line corresponds to a set of constant r and θ values.

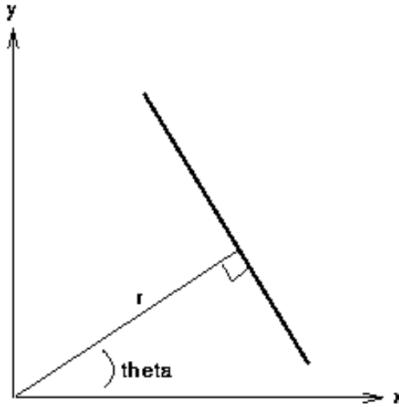


Figure 2.1: Parametric line represented by Equation 2.1 (Fisher et al., 2003)

Therefore when viewed in the Hough parameter space, points that are collinear in the cartesian space will yield curves which intersect at common r and θ values. Here bright areas (high degree of intersection) indicates collinearity between points in an image.

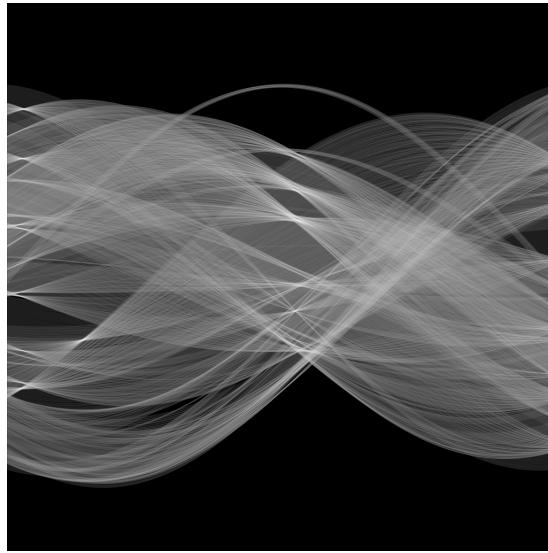


Figure 2.2: Parametric line represented by Equation 2.1 (Fisher et al., 2003)

For detecting circles the computational complexity of the algorithm increases, because the parametric equation representing a circle requires a three dimensional Hough parameter space (see Equation 2.2).

$$(x - a)^2 + (y - b)^2 = r^2 \quad (2.2)$$

2.1.2 Image segmentation using superpixels

Most images are based on a raster format, meaning that pixels in the image are structured as an array or grid, where each pixel is associated with a position (row and column), and a numeric value. Raster images can represent a range of different shapes, where a point can be represented by a single pixel and a circle by a contiguous collection of pixels ([Worboys, 2003](#)). Even though rasters are easy to work with in most computer systems, since they are represented the way that they are, they do not contain any information about the topology of the objects in the image. For example, there is no way of knowing if a pixel is contained within a certain object or not. In order to identify objects, one approach is therefore to detect edge pixels.

In recent years shape detection algorithms have come to increasingly rely on superpixel algorithms, which groups pixels into perceptually meaningful atomic regions ([Achanta et al., 2012](#)). Such regions replace the regular, rigid structure of the raster grid, as shown in [Figure 2.3](#).



Figure 2.3: Visualization of image segmentation using SLIC ([Achanta et al., 2012](#))

When constructing superpixels, there are some properties of the algorithm that are desirable, regardless of the problem that are being solved. These are, according to ([Achanta et al., 2012](#)), the following three points:

- Superpixels should adhere well to image boundaries
- When used to reduce computational complexity as a preprocessing step, superpixels should be fast to compute, memory efficient, and simple to use.

- When used for segmentation purposes, superpixels should both increase the speed and improve the quality of the results.

2.2 Artificial Neural Networks

In recent years, the development of machine learning, a branch of artificial intelligent systems, has become increasingly important in terms of pattern and object recognition in remote sensing and image analysis in general. One of the most commonly used approaches for data mining in remote sensing, has been artificial neural networks [Lary et al. \(2016\)](#).

The basic principle behind artificial neural networks (ANNs) is that it is built up by a network of many simple units that are working in parallel with no centralized control unit. The networks primary means of storage are the weights between the individual units, and the network learns by updating these weights in relation to being provided with a training example.

In order to understand the behavior of ANNs it is important to understand their structures. The most common structure for an ANN is what is called a Feed-forward neural network structure.

2.2.1 Feed-forward neural networks

A feed-forward neural network is build up by a given number of connected layers. A layer is a collection of simple units, called artificial neurons. All networks must consist of an input and output layer, but can also contain an optional number of hidden layers. A network that only consists of an input and output layer is called a perceptron, and it has been shown that these types of networks are only able to model linear functions ([Minsky and Papert, 1969](#)).

Input Layer The input layer is the layer that feeds the information into the network. Here the number of neurons are typically equal to the number of features in the data.

Hidden layer The hidden layers in an ANN is what enables the network to learn and model nonlinear functions. It is the weights on the connections between the different layers that enables the network to encode the information extracted from the training data.

Output layer In order to extract the answer or prediction from the model, an output layer has to be present. Depending on the setup of the neural network, the output value can either be a real value or a set of probabilities. The output type is dependent on the activation that is chosen

for the layer. What an activation function is, and how it effects the output of a layer will be discussed later in this section.

A feed forward network can either be fully or partly connected. In a fully connected network, all the neurons in each layer has a connection to all neurons in the previous layer, and all neurons in the next layer, while in a partly connected layer only some of the neurons are connected.

2.2.2 Training a neural network

The main purpose of a well trained ANN is to be able, by using its weighted connections, to amplify the signal and dampen the noise of the data it has been trained on. It does so by altering the different weights and biases in a way that allocates significance to some features and removing it from other. This way the model can learn which features are tied to which outcomes.

Neural networks learn these relationships by making a guess based on the input, weights and biases, and then get feedback on how accurate the guess was. It is the loss function related to the model, such as stochastic gradient decent (SGD), which gives this feedback by rewarding good guesses and penalizing bad ones.

The most common learning algorithm associated with neural networks is the *backpropagation learning algorithm*.

Backpropagation learning

The backpropagation algorithm learns by first trying to compute a training examples output value, by taking a forward pass through the network. If the output matches the label associated with the example nothing happens, but if it does not the weights has to be updated.

In order to update the weights in the network, [Equation 2.3](#) is used.

$$W_{j,i} \leftarrow W_{j,i} + \alpha * a_j * Err_i * g'(input_sum_i) \quad (2.3)$$

[Equation 2.3](#) is called the weight update rule for the connection between neuron j and i as seen in [Figure 2.4](#). Furthermore, α is the learning rate (discussed in [subsection 2.2.5](#)), a_j is the incoming activation function, Err_i is the error in i and $g'(input_sum_i)$ is the derivative of the activation function over the input sum as seen in [Equation 2.4](#).

$$a_j = g(input_sum_j) \quad (2.4)$$

where the input sum is given by:

$$\text{input_sum}_i = W_i * A_i + b \quad (2.5)$$

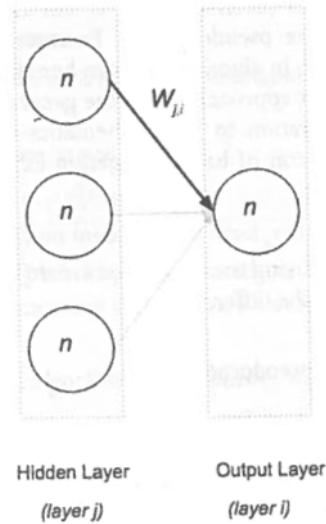


Figure 2.4: The two last layers of an multilayer feed-forward neural network [Patterson and Gibson \(2017\)](#)

The backpropagation algorithm traverses the network backwards, updating the weight of connection between each layer, as described in [Equation 2.3](#), until it reaches the input layer. This way the weights and biases that have been assigned the blame for the error are reduced, while the ones that supporting the correct answer are strengthened.

2.2.3 Activation functions

Activation functions are scalar-to-scalar functions which are used to propagate the output of one layer to the next, and it is what enables the network to model nonlinear functions. This section will discuss some of the most common activation functions used in ANNs.

Linear The linear activation function, $f(x) = Wx$ is the simplest activation function and is often associated with the input layer of the network. It says that the dependent variable x is proportional with the independent variable Wx .

Sigmoid The sigmoid activation function belongs to the class of logistic activation functions. It reduces extreme values or outliers in the example data, without removing them. One could

see the sigmoid function as a machine that converts independent variables of near infinite range into probabilities between 0 and 1.

Tanh (and Hard Tanh) Another class of activation functions are the hyperbolic trigonometric functions. The tanh function represents the ratio between the hyperbolic sine sine to the hyperbolic cosine, which means that unlike the sigmoid function, it has a normalized range between -1 and 1. The hard tanh activation function simply adds hard caps to the range, setting all values larger than 1 and smaller than -1 to respectively 1 and -1. The advantage of these functions are that they can deal more easily with negative numbers.

Softmax The softmax function, also referred to as the normalized, exponential function, is a generalization of the logistic function. Its function is that it returns the probability distribution over mutually exclusive output classes. For example, if the softmax activation function was applied to the vector [1, 2, 3, 4, 1, 2, 3], the result would be [0.024, 0.064, 0.175, 0.475, 0.024, 0.064, 0.175,]. As seen from this example, the function is most often used to weight the largest value and dampen values that are considerably smaller.

ReLU The rectified linear unit function is currently considered the state of the art activation function. It can be described as $f(x) = \max(0, x)$, meaning that it, above a certain threshold the output has a linear relationship with the dependent variable, it is else zero.

2.2.4 Loss functions

When working with artificial neural networks, we often talk about the ideal state of the network, meaning the state that classifies all the examples correctly. The loss function is a way of quantifying how close a network is to this ideal state. This is done by aggregating the errors produced by the networks prediction over the entire dataset, and average this value to get a single number that represents how close the network is to its ideal state. In other words, by minimizing the loss function, the network gets as close as possible to its ideal state, resulting in an optimization problem where the solution can be approximated well with iterative optimization algorithms.

There are different loss functions that are appropriate for regression and classification problems, however for the purpose of this project report it is only relevant to discuss the ones related to classification problems.

Hinge loss For hard classification, for example with discrete classes [sell=0, buy=1], the hinge loss function is most commonly used. It is also used by a class of models called maximum-margin classification models such as support vector machines, which are discussed in ??.

Logistic loss Often probabilities are of greater interest than hard classifications, in such situations the logistic loss function is of more value. An important remark when computing probabilities, is that all values has to be in the range 0 to 1 and that the probability of mutually exclusive outcomes should sum up to 1. Therefore, it is essential that the last layer uses the softmax activation function.

Optimizing the logistic loss function is the same as optimizing the "maximal likelihood", which means that the algorithm should maximize the probability that it predict the correct class, and do so for each single sample in the dataset.

NOT FINISHED HERE

2.2.5 Hyperparamenters

In machine learning, hyperparameters deal with controlling the optimazation functions during learning, making sure that they neither overfit or underfit the data, but at the same time learns as quickly as possible.

Learning rate The learning rate, as seen in [Equation 2.3](#), is a coefficient that scales the size of each weight update step. In other words, the learning rate decides how much of the computed gradient that should be used for each step.

Regularization Regularization is important in order to control what is called out-of-control parameters. This is done by controlling the trade-off between finding a good fit on the training data and limiting the weight on features with high-polynomials. This is because such features tend to overfit the training examples.

Momentum Momentum is often described as the the learning rate of the learning rate. What it does is to prevent the learning algorithm from getting stuck in local minimas.

Sparsity Lastly, the sparsity hyperparameter helps recognize which features of the input examples that are relevant. This is important because in some datasets, the feature arrays for each example will be very different in terms of values.

2.2.6 Convolutional neural networks

In recent years, convolutional neural networks (CNNs) have been recognized as very suitable for object recognition in images [Patterson and Gibson \(2017\)](#). One of the main reasons why the world, research society recognizes the power of deep learning have been the efficacy of CNNs image recognition capabilities. The name comes from the networks use of convolutions, a mathematical operation on two functions in order to produce a third.

Biological Inspiration

As all neural networks, CNNs are very inspired by the biological neurons in animal brains. The CNNs are especially inspired by the visual cortex, which cells are very sensitive to small subregions of the input. One often say that these cells act as local filters over the input space, which also is the case for CNNs.

Difference from regular feed-forward multilayer neural networks

A well known problem when it comes to analyzing image data using regular feed-forward multilayer neural networks, is that they do not scale well with increasing image sizes. Imagine an color image with the size 400x400 pixels (which would be a regular sized picture) that are used as input for a ANN. Such an image, represented as a vector, would create $400 * 400 * 3 = 480000$ different weight connections for each neuron in the first hidden layer. For a fully connected network, this would be the case for all layers to come, which would create a tremendous amount of weight connections.

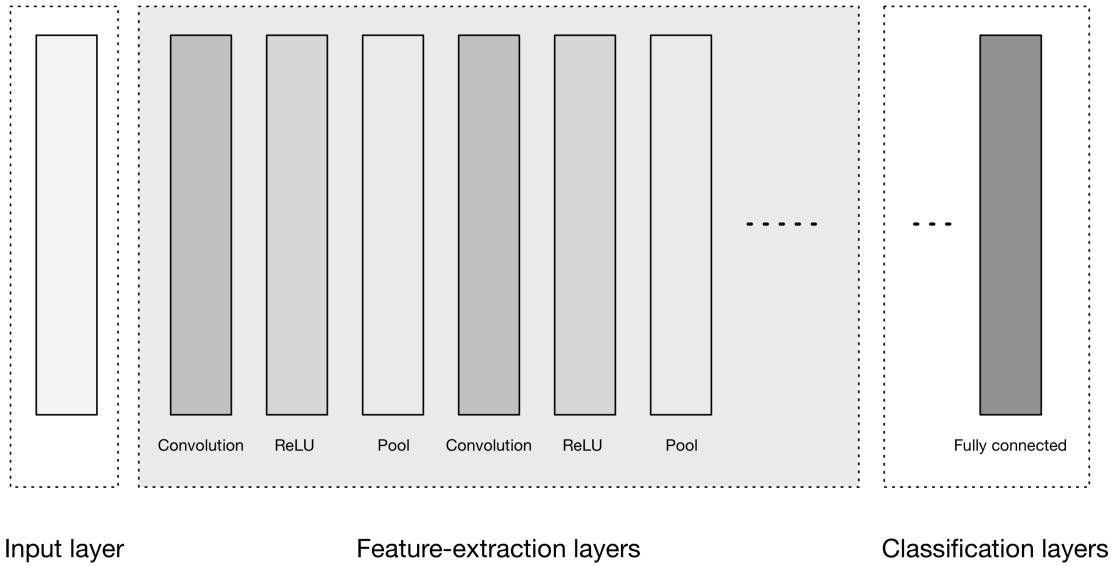
Convolutional neural networks solve this issue by representing the images in a three-dimensional structure, meaning that the input data is represented as a three-dimensional matrix with:

- Image width in pixels
- Image height in pixels
- RGB channels in depth

As will be discussed later, this structure is how CNNs have evolved from previous feed-forward networks, in terms of computational efficiency.

Architecture

The purpose of the network is to transform the input examples through a series of connected layers, into a set of class scores. A general architecture is presented in [Figure 2.6](#).



[Figure 2.5: A general presentation of the architecture of CNNs Patterson and Gibson \(2017\)](#)

As seen in [Figure 2.6](#) the network can be divided into three sections; Input, Feature-extraction and Classification layer(s). The most interesting part of this structure is the feature-extraction layers, which is used to identify a number of features in the images, and from these construct higher-order features. The strategy of constructing high-order features is one of the key aspects of deep learning.

Remark: It is important to note that in a CNNs layers, the neurons are arranged in a three-dimensional structure, in order to match the input data, as described earlier.

Convolutional layers

The convolution layers, seen in [Figure 2.6](#), detect features in an image through what is called the convolution operation. As mentioned in the beginning of this chapter is a convolution operation a mathematical operation that transforms two functions (or sets of information) into one,

through Fourier transformations. The way that this works, is that the layer applies certain filters, called kernel filters, to segments of its input using a technique called sliding window.¹ The convolution operator is described in [Equation 2.6](#), where I is the input data and K is the kernel filter of size $h * w$.

$$(I * K)_{xy} = \sum_{i=1}^h \sum_{j=1}^w K_{ij} \cdot I_{x+i-1, y+j-1} \quad (2.6)$$

Such filters can for example be an edge kernel, which only passes through information containing edges. In most cases applying a filter means reducing the size of the data, thus reducing the number of neurons in each, upcoming layer.

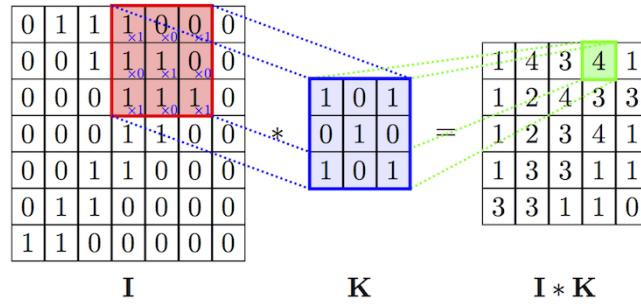


Figure 2.6: The convolution operation (applying a kernel filter) [Cambridge \(2017\)](#)

ReLU layers

As seen in [Figure 2.6](#), ReLU activation functions are often used in separate layers. This layer does not change the dimension of the input volume, but can change some of the pixel values.

Pooling layers

The pooling layers is an other important part of the convolutional neural networks. They help prevent overfitting, by reducing the size of the input data using what is called max pooling, as shown in [Figure 2.7](#)

[Figure 2.7](#) presents a max pool with a 2x2 filter size, and a stride of 2, meaning that 2x2 pixels are compared and that the sliding window moves 2 pixels for each comparison. I practice this means that the 75% of the activations from the previous layer is discarded.

¹A technique that slides over a set of data, only analyzing a pre-defined patch size at a time [Stanford \(2017\)](#)

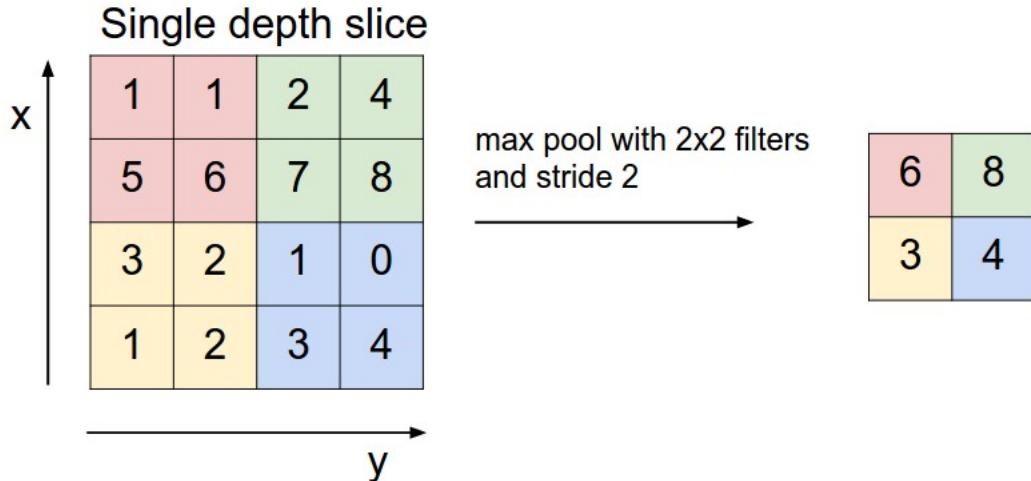


Figure 2.7: Example of max pooling operation [Karpathy \(2017\)](#)

Links:

<https://www.pyimagesearch.com/2014/10/20/finding-shapes-images-using-python-opencv/>
<https://www.pyimagesearch.com/2016/02/08/opencv-shape-detection/>
<http://melvincabatuan.github.io/SLIC-Superpixels/>
<https://www.gislounge.com/shadows-angles-measuring-object-heights-satellite-imagery/>

2.3 Height estimation using remote sensing

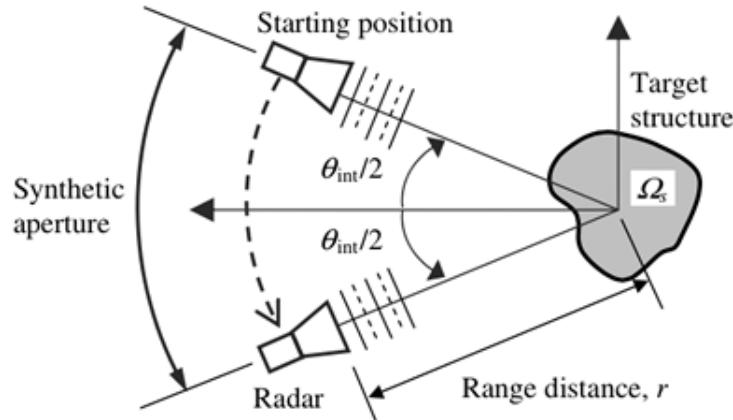
There are many different ways of estimating heights using satellite data, which involves using lidar data, SAR data and regular satellite multispectral imagery. This section will give the theoretical background for some of the most commonly used techniques.

2.3.1 Synthetic Aperture Radar (SAR)

The first technique involves using a Synthetic Aperture Radar on a moving platform, in this case a satellite. By sequentially transmitting electromagnetic waves onto the earth's surface and collecting the reflected echos, the SAR satellites are able to collect high-precision, 3-dimensional data.

One of the key advantages of the SAR satellites is that they take advantage of the fact that they are moving quickly. Since the transmission and reception occurs at different times, the platform

has moved, thus creating a synthetic aperture that is much larger than the satellite antenna (see [Figure 2.8](#)).



[Figure 2.8: Concept of syntetic aperture ?](#)

The technique provides a finer spatial resolution to the collected data, making it possible to do height estimations with a sub-decimeter accuracy.

Interferometric synthetic aperture radar (InSAR)

While SAR makes use of amplitude and absolute phase of the returned signal, InSAR use a differential phase of the reflected signal, represented in what is called a phase image.

Looking at a phase image completely isolated will not prove very useful, as it would appear visually random. This is because in practice, the phase of the return signal is affected by a lot of different factors, resulting in no apparent correlation between the pixels in the image.

In order to get useful information these phase images, some of the factors, discussed above, has to be removed. The process of doing so is referred to as interferometry, and it uses two phase images taken from the exact same position in order to generate a interferogram (see [Figure 2.9](#)).

Producing an inferiogram consists of multiple steps:

Co-registration Using a correlation function the two images are co-registered by finding the offset and difference in geometry between them.

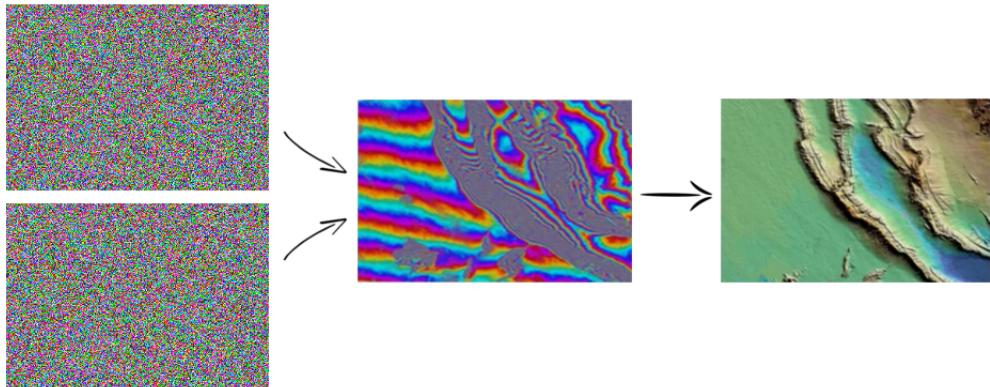


Figure 2.9: Generating an interferogram

Re-sampling In the re-sampling step, one of the images (referred to as the slave) is re-sampled in order to match the geometry of the other (called the master). What this means in practice is that each pixel represents the same area of ground in both images.

Cross-multiplication and flattening After re-sampling the images the interferogram is generated by taking the cross product of each pixel, and the interferometric phase due to the curvature of the earth is removed (flattening).

Filtering Lastly, it is common to filter the basic interferogram in order to amplify the phase signal, and to interpolate over phase jumps in order to produce a continuous deformation field.

2.3.2 Light Detection And Ranging (LIDAR)

Another high precision technique for generating height data using satellites is using LIDAR. LIDAR is a surveying method that uses a pulsed ultraviolet, visible or near infrared light laser to map an area or an object. In the same way as SAR and InSAR, it used the reflected signal to measure the distance from the transmitter to the object.

2.3.3 Height detection using shadows from imagery

For height estimation of specific objects, shadow estimations can be applied. This technique uses a high resolution image, as well as knowledge about the position of the sun, the position of the image projection center and the length of the projected shadow in order to estimate an objects height ([Figure 2.10](#)).

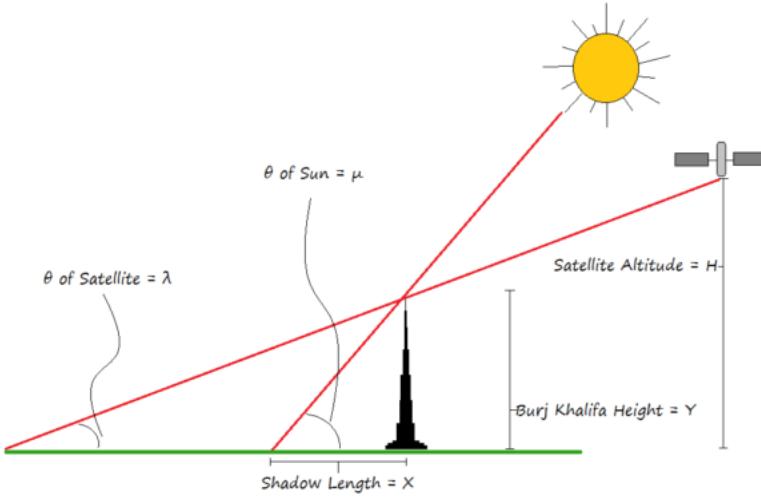


Figure 2.10: Height estimation of Burj Khalifa using shadow estimation ([GISLounge, 2014](#))

Looking at Figure 2.10 the height of the object can be found by:

$$\tan(\mu) = \frac{H_{object}}{L_{shadow}} \Rightarrow L_{shadow} = \frac{H_{object}}{\tan(\mu)} \quad (2.7)$$

Since the satellite image rarely is taken with the projection center directly above the object (nadir angle = 0°), the blocked length of the shadow has to be estimated. Since the height of the object is the unknown, estimating the blocked length of the shadow is an iterative process, where a new calculated, temporary value for the height is used in each iteration. Calculating the length of the blocked shadow is also very straight forward:

$$\tan(90 - \lambda) = \frac{L_{blocked\ shadow}}{H_{object}} \Rightarrow L_{blocked\ shadow} = H_{object} * \tan(90 - \lambda) \quad (2.8)$$

In Equation 2.8 $90 - \lambda$ defined as the Off-nadir angle.

It is important to remember that if this technique shall produce exact results, some parameters has to be known in the moment of the capture:

- The position of the satellite
- The time
- The position of the object with the unknown height

Chapter 3

Related Work

3.1 Standard methods for shape detection

Before looking at the use of neural networks for shape recognition in images, some more standard approaches are presented.

3.1.1 Shape detection using the Hough Transform

Shape detection and feature extraction in images, using Hough Transforms has been a subject of research for many years. The algorithm was first popularized in computer vision in 1981 by [Ballard \(1981\)](#) which applied a generalized version of the algorithm in order to detect of curves in gray scale images.

The article describes how boundary detection plays a crucial role for feature extraction in images, and how a generalized Hough algorithm can use edge information to define a mapping from the orientation of an edge point to a reference point related to the shape.

3.1.2 Simple Linear Iterative Clustering (SLIC)

[Achanta et al. \(2012\)](#) proposes a superpixel-segmentation algorithm (SLIC), which in their opinion is best suited to meet these demands. They compare their algorithm to a variety of state-of-the-art superpixel methods, and conclude that none of the existing methods are satisfactory in regards to the points.

The SLIC algorithm is fairly simple to understand. One of its key principles is that, by limiting the search space for each cluster center (points in the regular raster grid), it reduces the search speed significantly. This is achievable due to the fact that one of the primary goals of algorithm is to create a set of approximately equal sized superpixels. Thus, instead of searching the whole raster grid for each cluster center, the algorithm only has to search for edge pixels at a distance equal to D , as shown in [Equation 3.1](#).

$$D' = \sqrt{\left(\frac{d_c}{m}\right)^2 + \left(\frac{d_s}{S}\right)^2} \quad (3.1)$$

In [Equation 3.1](#) d_c is the euclidean distance between two pixels in terms of color and d_s is the pixels euclidean, spatial distance. Furthermore, S is the sampling interval of the cluster centers ($S = \sqrt{N/k}$, where N is the number of pixels in the grid and k is the desired number of superpixels) and m is a fixed constant based on the color diversity in the image.

Since the algorithm generates superpixels by clustering pixels based on their color and spatial proximity, creating a 5 dimensional, *labxy* space, one would think that the distance could be found by simply taking the 5D euclidean distance. However, it turns out that for large superpixels, spatial distance outweigh the color proximity. Which is why the two distances d_c and d_s are weighted.

3.2 The development of Convolutional neural networks

Since McCulloch and Pitts created what is acknowledged as the first neural network in 1943, using simple electrical circuits ([McCulloch and Pitts, 1943](#)), they have played an important role in the field of pattern recognition.

Since the late 1990's the idea of using convolutional operations in these networks has been considered more and more prominent ([Le Cun et al., 1998](#)), and this approach is still one of the leading research fields within ANN research ([Wu et al., 2017](#)). This section will discuss the development of the convolutional neural networks the last two decades.

3.2.1 Early adaption of convolutional neural networks

The earliest attempts of using convolutional operations for pattern and object recognition in neural networks was first done nearly twenty years ago.

MORE HERE (LeNet)

3.2.2 Deep Convolutional Neural Networks

It was however, not until Alex Krizhevsky, Geoffrey Hinton, and Ilya Sutskever won the ImageNet¹ 2012 competition (ILSVRC'12), that CNNs became acknowledged as one of the most sophisticated approaches for image recognition. Their deep convolutional neural network (commonly called AlexNet) consisted of five convolutional layers, each followed by a pooling layer (max-pooling), and three fully-connected softmax layers (Krizhevsky et al., 2012). In order to reduce overfitting, the network applied two different methods; Data Augmentation and Dropout layers.

Even though the AlexNet was a big breakthrough for the convolutional neural networks, it was criticized for not presenting a good ground for understanding what was happening inside the network, thus making it hard to improve. One of the solutions to this problem are the ZF Net, which applies deconvolutional neural networks (Zeiler et al., 2011) in order to map the dense feature space produced by a CNN back to its original pixel space (Zeiler and Fergus, 2013). Figure 3.1 shows how a deconvolutional network can help visualizing the feature space of a CNN.

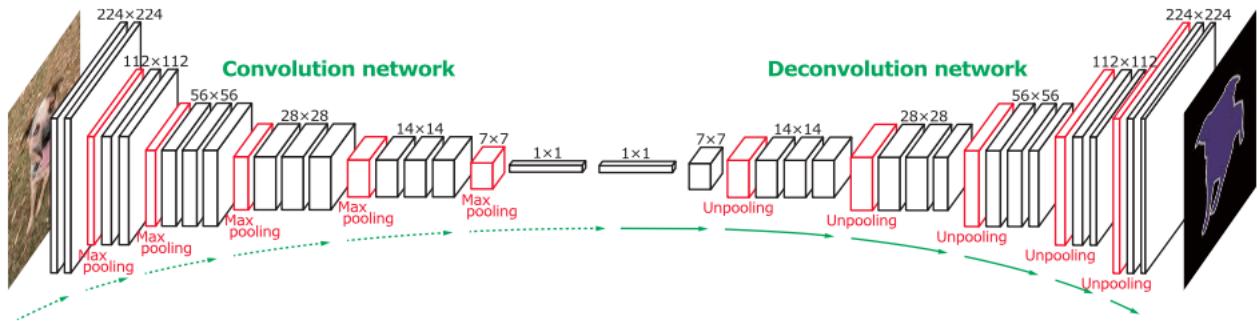


Figure 3.1: Process of visualizing the feature space of a CNN (Noh et al., 2015)

In order to do so, the ZF Net successfully unpool, rectify and filter the feature map, generated by the CNN, in order to reconstruct the activity inside the network.

Unpooling In general, the max-pooling operation is non-reversible, since there are no tracking of the positions of the selected features, as seen in section 2.2.6. Therefore, in order to reconstruct the feature map the location of each selected maxima had to be stored.

Rectification Their CNN used the common ReLU activation function for each Convolution Layer, therefore the reconstruction layers also need to use this activation function in order to prevent negative values.

¹ImageNet is a very large dataset consisting of 15 million labeled, high-resolution images divided into over 22.000 categories.

Filtering Since CNNs apply kernel filters to each convolution layers input volumes, the filtering has to be inverted when reconstructing. In order to do so, the same filters are transposed (remember that the filters are matrices) and applied to the rectified activation maps.

GoogLeNet

Another network, whose creators have criticized the standard structure of the convolutional neural networks, is the GoogLeNet (Szegedy et al., 2014). The authors of the article claims that their network was significantly more accurate than the AlexNet, while at the same time only using one twelfth of the parameters.

The GoogLeNet introduces what is called a Inception architecture (see Figure 3.3), which main idea is to cluster neurons in the network which have highly correlated outputs. This is because in images, the correlation between pixels tend to be local.

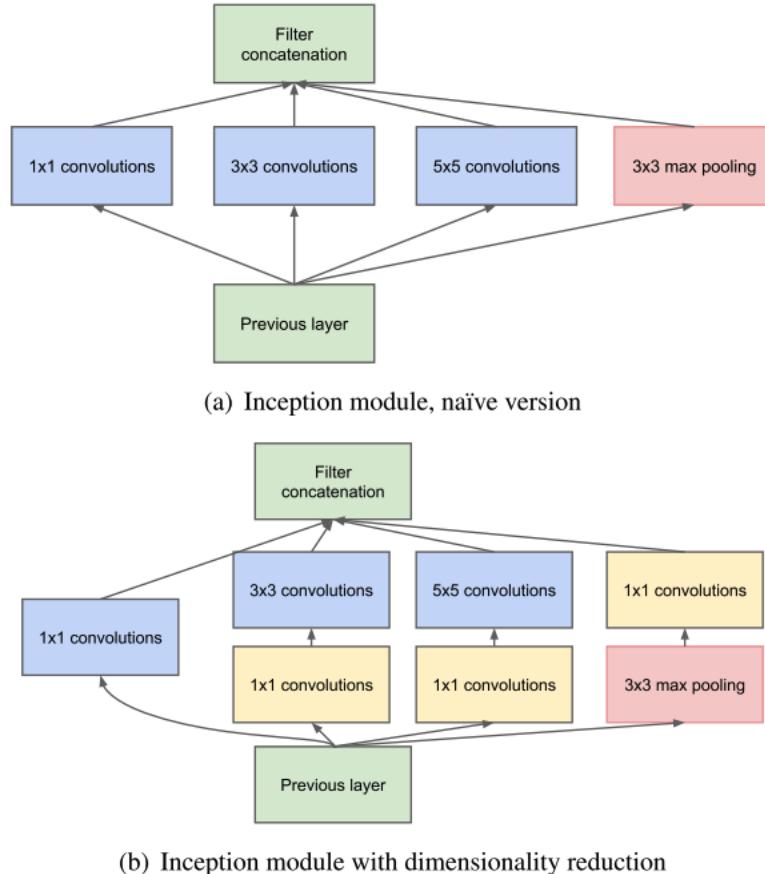


Figure 3.2: Two versions of the inception module (Szegedy et al., 2014)

The basic structure of the inception module is to do multiple convolution operations, in parallel,

with different sized filters, and then concatenate the results before passing them on to the next layer. Additionally, a parallel pooling operation is added to each inception module, because such operations have proven them selves successful in other CNNs.

The naive version of these "micro networks" is shown in [Figure 3.3](#) (a). One issue with this naive version is that, even with a modest number of 5x5 convolutions, the computational cost can be quite expensive. This is solved by keeping the representation of the information as sparse as possible, and only compress the signals when they have to be aggregated. In order to do so, 1x1 convolutions are used to compute reductions before the 3x3 and 5x5 convolutional operations are applied ([Figure 3.3](#) (b)).

VGGNet

Another example of a successful deep convolutional neural network is the VGGNet. This network does not present any new concepts, but the authors argue that by stacking multiple layers, doing small convolutional operations (3x3 and a few 1x1) they can outperform the other discussed networks ([Simonyan and Zisserman, 2014](#)).

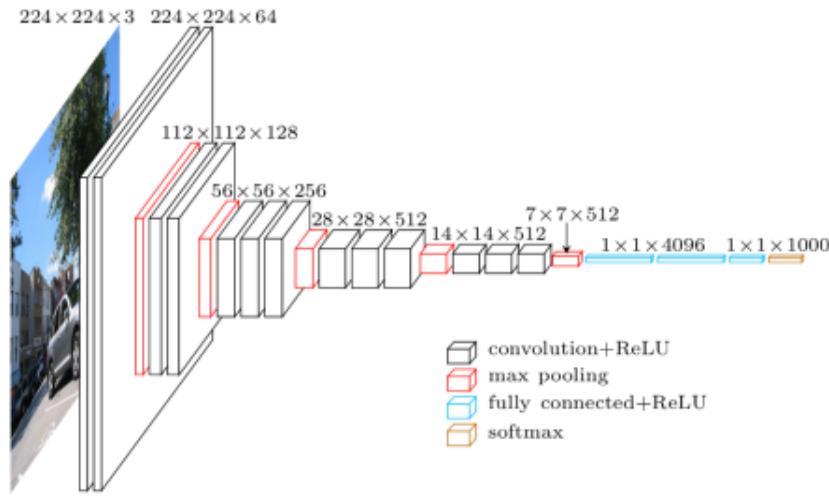


Figure 3.3: The 16-layer VGGNet ([Frossard, 2016](#))

An important difference between this network and previous networks, is that the creators focuses on depth, thus calling the network a very deep convolutional network.

ResNet

A problem that arises with deeper convolutional neural networks is the degradation problem. As depth increases in the network, the accuracy decreases ([Figure 3.4](#)).

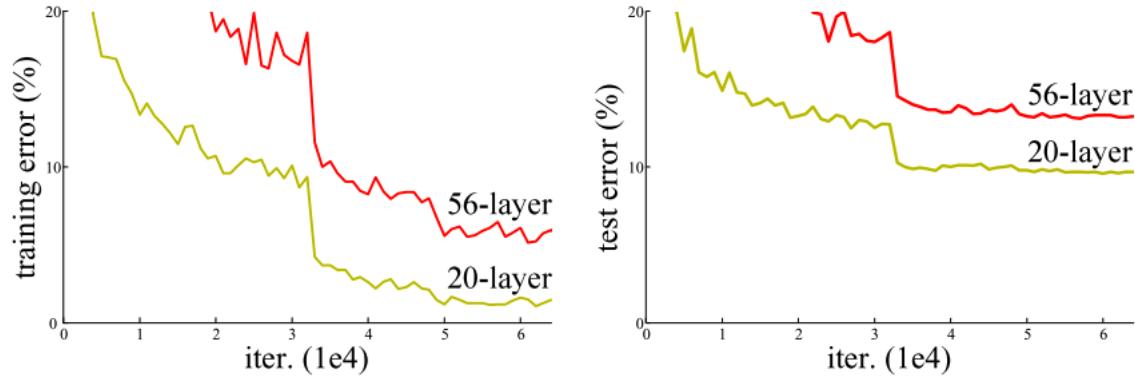


Figure 3.4: Degradation problem ([Wu et al., 2017](#))

Surprisingly enough this decrease in accuracy is not due to overfitting, as adding more layers effects the training error as well ([Wu et al., 2017](#)). In theory any deep network should be able to perform at least as good as a shallower network, only by seeing the layers that differentiate the two networks as identity mappings.² This however, does not seem to be the case in practice, suggesting that networks have problems learning identity mappings by multiple, non-linear layers.

In order to solve the degradation problem, the authors of the paper introduces two new concepts, which sets the foundation for their residual neural network (ResNet).

Residual mapping The paper hypothesize that it is easier to optimize residual mapping, thus introducing the residual mapping function:

$$F(x) := H(x) - x \quad (3.2)$$

where $H(x)$ is the desired, underlying mapping function after 2 weight layers.

Shortcut connections Connections which skip one or more layers, as seen in [Figure 3.5](#), in order to obtain the desired mapping function $H(x)$.

Using these concepts, the authors were able to create the winning 152-layer deep ResNet of the ILSVRC'15.

Check out <http://yann.lecun.com/exdb/publis/pdf/lecun-89e.pdf>

²The map which assigns every member of a set A to the same element id_A . It is the same as the identity function: $id(x) = x$

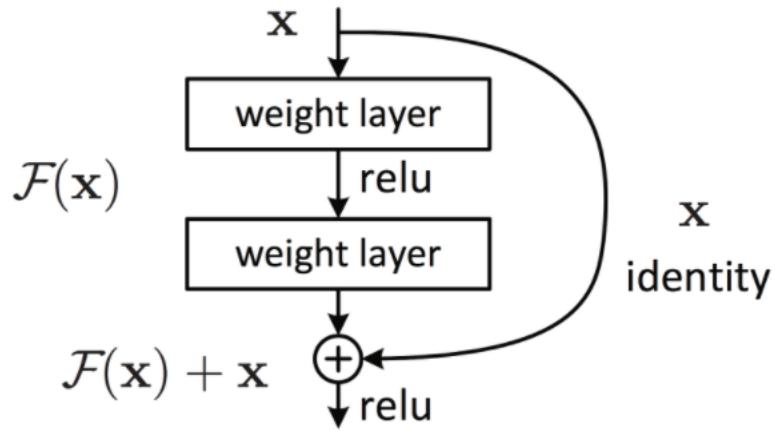


Figure 3.5: A residual building block (Wu et al., 2017)

3.3 Semantic segmentation using convolution neural networks

In later years there have been a significant progress in the field of edge detection in imagery has been made due to advances in deep learning (Yu et al., 2017). Using an end to end deep semantic edge learning architecture based on ResNets (Wu et al., 2017) and further extending the networks architecture with a new skip-layer, Yu et al. (2017) are able to not only identify, but also categorize edges in an image (see Figure 3.6).

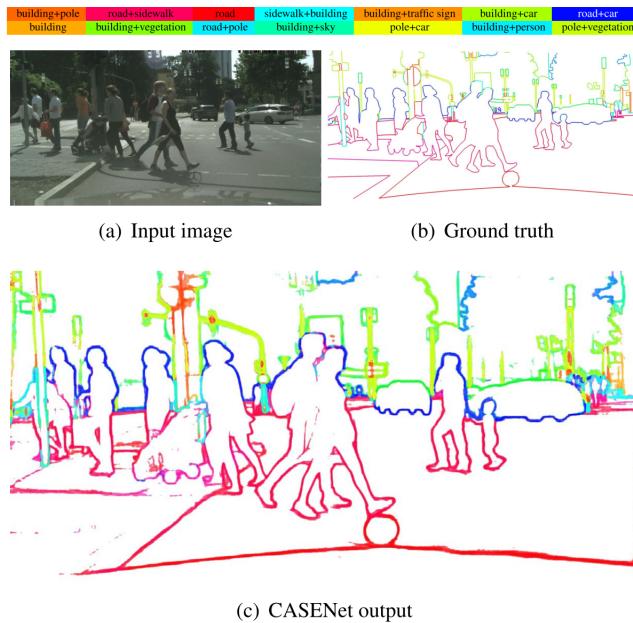


Figure 3.6: Edge detection using CASENets (Yu et al., 2017)

The skip-layer uses category-wise edge activations, which are added on top of convolution layer,

which share and are fused with the same set of bottom layer features. Using this approach the network produces n different edge maps, where n is the number of defined categories. Each map indicates the edge probability of e certain category.

The proposed CASENet is a modification of the ResNet-101, with some modification to the convolutional blocks, in order to preserve low-level edge information. The authors compare their network to other similar architectures, and show that their architecture outperforms state-of-the-art networks.

In terms of aerial image segmentation [Kaiser et al. \(2017\)](#) present a solution, which deals with the lack of labeled training data available. In order to do so, the authors make use of the open source map data library OpenStreetMap, in order to automatically derive weakly labeled training data. By matching this data with aerial imagery from Google Maps, and training a fully convolutional neural network they are able to classify buildings and roads at a high level of accuracy (see [Figure 3.7](#)).

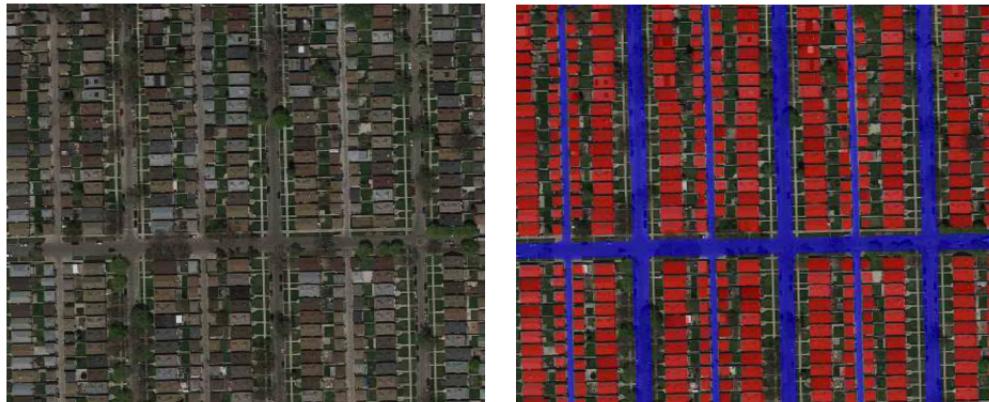


Figure 3.7: Results from classification of buildings and roads ([Kaiser et al., 2017](#))

Other attempts for semantic segmentation in remote sensing have been done with different variants of deep convolutional neural networks ([Kemker et al., 2017](#)). In their paper, they propose an approach for using the Digital Imaging and Remote Sensing Image Generation (DIRSIG) modeling software to generate a large quantity of synthetic multispectral imagery and corresponding labels. For the semantic segmentation, the authors adapt two fully-convolutional neural networks called SharpMask ([Pinheiro et al., 2016](#)) and RefineNet ([Lin et al., 2016](#)). Their results show that it is possible to use synthetic imagery can be used to assist in terms of training semantic segmentation networks when there is not enough annotated image data.

3.4 Height estimation from a nadir perspective

As seen in section 2.3 there are many different techniques that can be applied in order to estimate the height of a building. This section will focus on different attempts to retrieve height data, using satellite imagery.

3.4.1 Building height retrieval from VHR SAR Imagery

After the launch of TerraSAR-X in 2007, a satellite that was designed to acquire high-resolution X-band radar images of the entire planet, it was possible to gather SAR data with a resolution of down to 1 meter ([Airbus, 2017](#)). In contrast to other optical, spaceborne sensors, such as Ikonos, Quickbird and WorldView, satellites using SAR overcome the difficulties of weather conditions and lack of sun illumination.

Using the SAR images provided by the satellite [Brunner et al. \(2008\)](#) was able to estimate the height of man-made structures with a sub-meter precision by automatically reconstruct 3-D models, using a "hypothesis generation-rendering-matching" procedure. The basic principle was that using a optimization algorithm, the height of a building is found by testing different height hypothesis against a single SAR image. The estimation is done without modeling its exact radiometry, since this would require extensive apriori knowledge about the roughness parameters and dielectric constants of the surfaces. Generating this height model becomes an optimization problem [Equation 3.3](#).

$$\hat{h} = \underset{h, \vec{s}}{\operatorname{argmax}} \left\{ M \left[\hat{X}_{\vec{s}}(\vec{H}), X \right] \right\} \quad (3.3)$$

In [Equation 3.3](#) X is the true SAR image, \hat{X} is the simulated SAR image at height h , M is the matching function, and \vec{H} is the simulated hypothesis.

[Brunner et al. \(2008\)](#) tested their method on different types of buildings, where flat buildings gave a mean accuracy of $0.3 \pm 2.1m$.

3.4.2 Height estimation from InSAR analysis

Another approach for height detection using SAR satellites, is interferometric SAR (InSAR) analysis, as attempted by [Liu et al. \(2015\)](#). In their paper they used images taken on December 5 and 27, 2007 in order to generate an interferogram over San Francisco. The technique was based on extracting potential layover areas from the interferogram and use these areas to measure the

height of the buildings using the relationship between the height of an object and the layover observed in the interferogram ([Equation 3.4](#)).

$$\Delta\Phi = \frac{4\pi B_N}{\lambda H \sin\theta} \Delta R \quad (3.4)$$

In [Equation 3.4](#) $\Delta\Phi$ is the phase difference within one pixel in the layover area, B_N is the perpendicular baseline distance, λ is the radar wavelength, H is the satellite altitude and θ is the angle of incidence (see [Figure 3.8](#)).

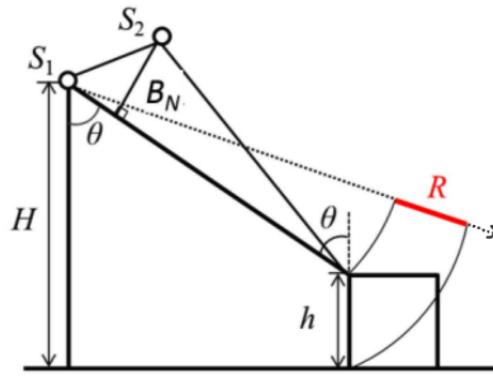


Figure 3.8: A schematic image of geometrical characteristics for a building in a slant-range SAR image ([Liu et al., 2015](#))

Using this method [Liu et al. \(2015\)](#) were able to estimate the height of high-rise buildings in a crowded area with a RMS of 13m and an average difference between detected results and the reference values of 6.6m.

3.4.3 Height estimation using shadow measurement

While some height extraction methods require precise specifications regarding the geometrical properties of remotely sensed data, [Comber et al. \(2012\)](#) present a method for determining height using building shadows. By classifying buildings and their associated shadows based on a series of different measures (see [Figure 3.9](#)), and applying [Equation 3.5](#) the authors were able to give relative good height estimations.

$$H = \frac{W}{\frac{\cos(\phi_{sun} + 90 + \phi_{az})}{\tan(\phi_{sun})}} \quad (3.5)$$

Other attempts have also been made in terms of estimation heights using shadows, such as [Shao et al. \(2011\)](#) who addresses the issue of distinguishing between shadows and water in aerial

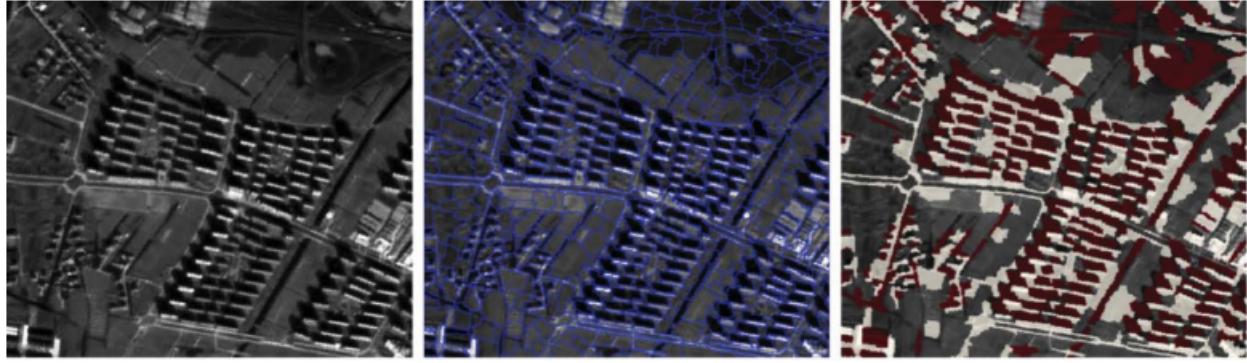


Figure 3.9: Classification of buildings and their assiated shadows ([Comber et al., 2012](#))

photographs. In their approach they merged the two classes to a single shadow/water class, and used different spatial indices such as object size, shape and spatial neighbour information in order to separate shadow and water. They further used a standard trigonometrical approach in order to estimate the building heights. Their results can be seen in [Figure 3.10](#).

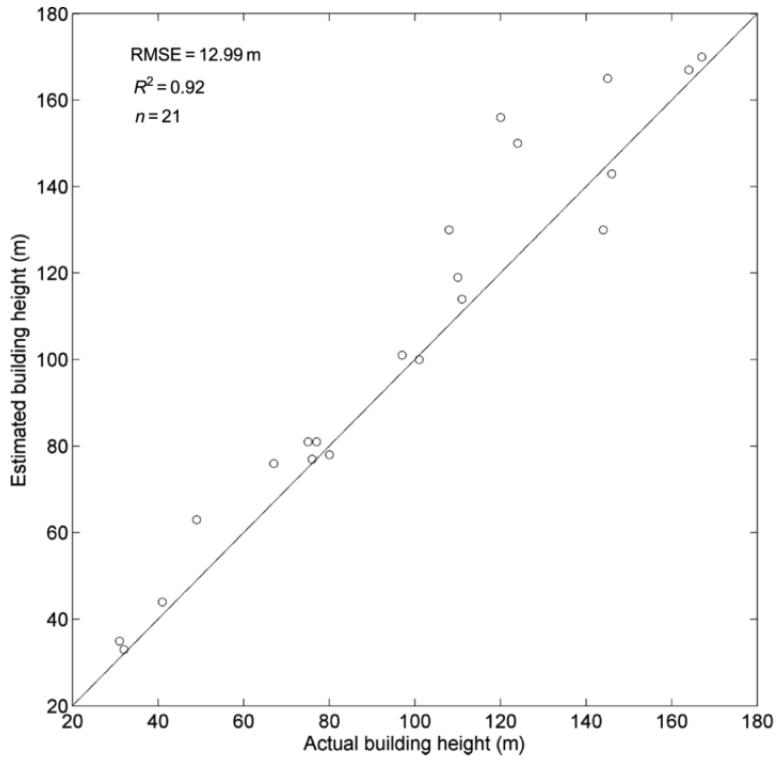


Figure 3.10: Accuracy evaluation of building height estimation ([Shao et al., 2011](#))

Chapter 4

Evaluating the methods

There are evidently a large variety of different methods available for solving the task at hand. Based on the work done in the previous chapter, section of the paper will discuss possible solutions to the problem.

4.1 Available data

One important aspect that should be considered is the amount of data available for the different methods. A key criteria for the success is that there exists satellite constellations that are able to provide frequently updated image data over the requested area. New images should be available on a weekly basis, and the resolution of the images has to be approximately one meter or less for the results to be accurate enough.

Multispectral images

There are many different satellite constellations that provide frequently updated multispectral imagery of the requested area, such as the Sentinel-2 ([ESA](#)), SkySat ([Planet, 2017](#)) and WorldView ([DigitalGlobe, 2017](#)) satellite constellations. All three constellations have a subweek revisit time over any point on earth, however the ground sampling distance (GSD) vary with a factor of more than ten.

While both SkySat and WorldView provide a submeter accuracy on their images, the Sentinel-1 constellation has a expected GSD of 10x10 meters, which will not be sufficient in terms of shadow measurement. Furthermore, it has to be taken into consideration that while data provided from the Sentinel constellations are free of charge, data from both SkySat and WorldView

comes at a certain cost. Data provided by the Sentinel-2 constellation might however be accurate enough in order to locate the oil tanks.

A general issue with multispectral images is that they are sensitive to cloudy weather and day light. This can pose as an issue if the system is expected to continuously provide data without deviations.

Radar images

Another type of images that is not affected by clouds or the time of day are radar images. In terms of radar based methods, there are also a large quantity of satellite constellations that can be of interest. For example while Sentinel-2 provide multispectral images, the Sentinel-1 constellation provides radar-images with resolutions down to 5x5 meters ([ESA](#)).

WRITE MORE HERE

4.2 Amount of data required for analysis

An important difference between the different methods presented in the previous sections is the amount of data needed in order to provide an accurate analysis. For example simple image segmentation can be done in a single image using Simple Linear Iterative Clustering ([Achanta et al., 2012](#)) as seen in [Figure 4.1](#). Furthermore, by knowing the approximate size of the oil tanks, a general Hough Transform ([Ballard, 1981](#)) can be used to detect their circular shapes as seen in [Figure 4.2](#). Image segmentation and shape detection using machine learning based methods require large amounts of training examples, which can be very hard to acquire. But as seen in [Kaiser et al. \(2017\)](#) and [Kemker et al. \(2017\)](#) it might be possible to automatically collect and label training data.

Advanced methods in machine learning, such as Convolutional Neural Networks, have proven themselves as very efficient in image segmentation, but also in terms of classifying the different segments and even associating values with them if given the correct training data. This means that by acquiring the correct training data, which in this setting would be the the actual inventory of an oil tank at a certain time, and relate these values to aerial photographs of the same tanks, it would be possible to train a network into doing the actual inventory estimation directly.

When it comes to height estimation using radar images, [Brunner et al. \(2008\)](#) presented a method that only acquired a single radar image in order to estimate the height of buildings using the SAR principle. In order to generate a interferogram two phase images are required as explained in

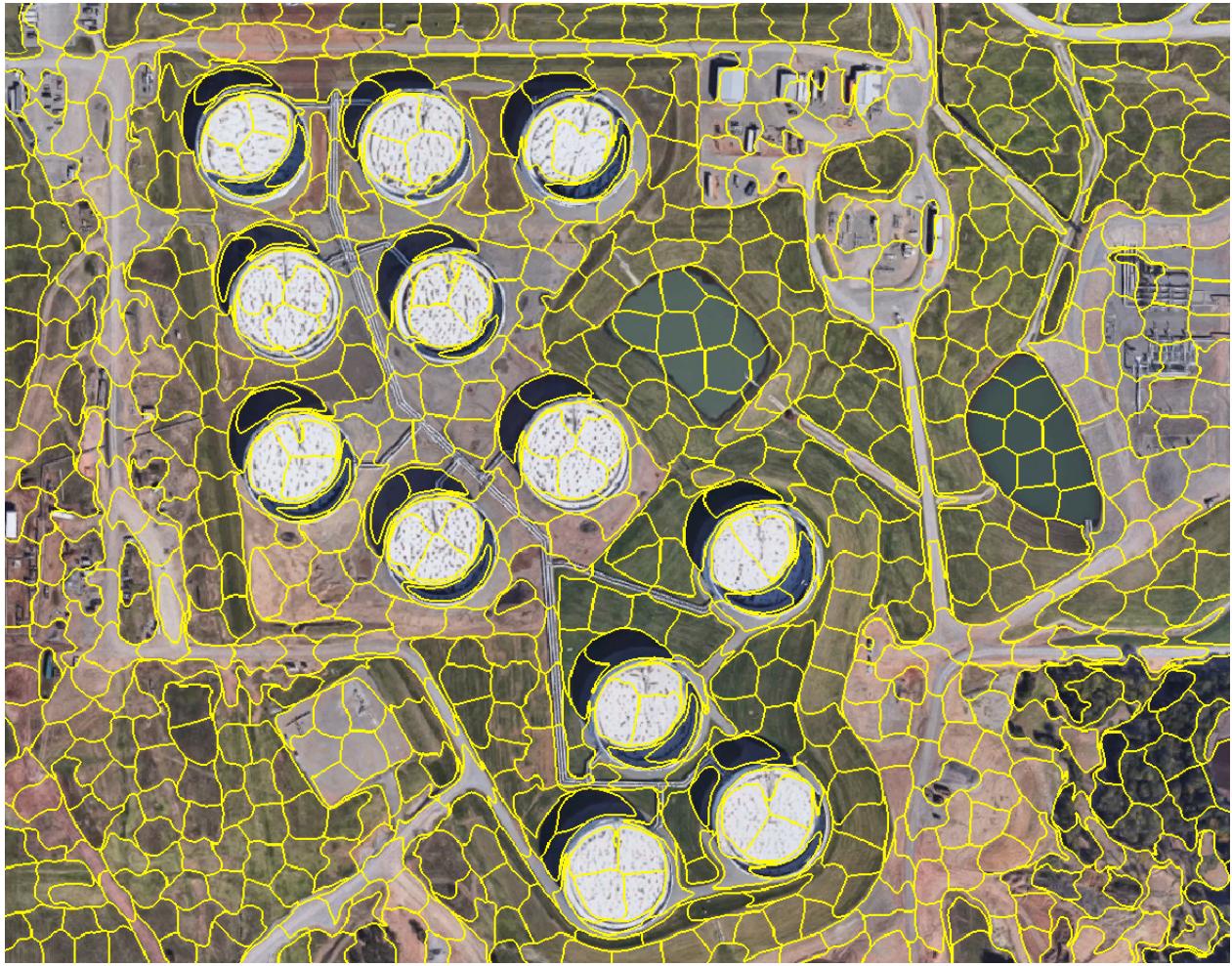


Figure 4.1: Simple implementation of SLIC superpixel segmentation on image taken from Google Earth

[Liu et al. \(2015\)](#). Furthermore, height estimation using shadow measurement can also be done using single, multispectral images as shown by [Comber et al. \(2012\)](#) and [Shao et al. \(2011\)](#).

4.3 Choosing a method

By taking the above comparison into consideration, it becomes clear that in terms of pure circle detection, the generalized Hough transform stands out as the most efficient method. The fact that we have pre-knowledge about the shape of the objects we are trying to locate, makes it much easier to use a standard approach. However, the generalized Hough transform isolated will not be able to estimate any heights, and would have to be combined with another method for height estimation. By looking at [Figure 4.2](#), height estimation using shadow measurements

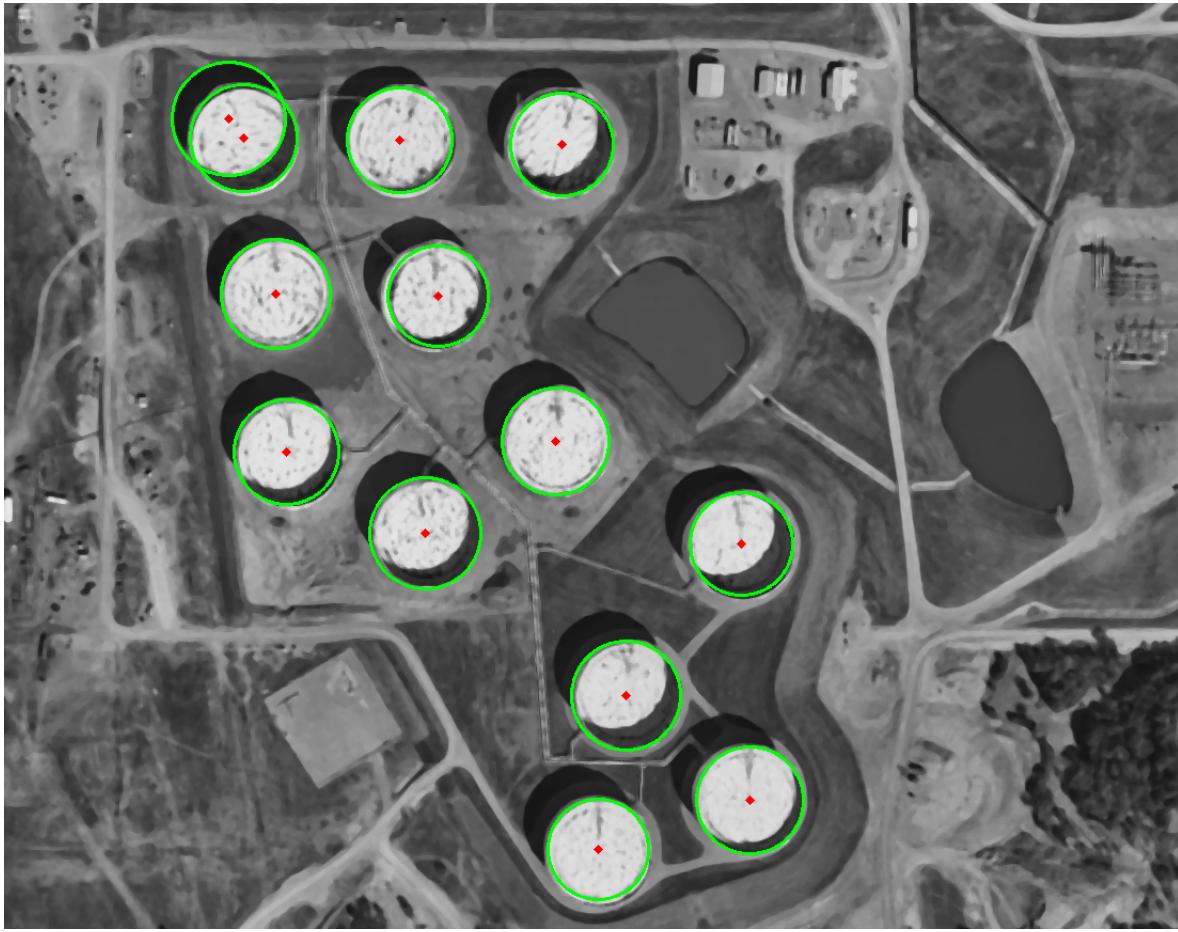


Figure 4.2: Detecting circles of similar size in an image taken from Google Earth using a generalized hough transform

might be the best approach. This is because the green circles limits the search area in terms of locating shadows, since we are only interested in the shadow that is cast inside the tanks.

An important issue regarding the generalized hough transform is the fact that if the tanks vary in size, it might not be able to find all of them, as seen in [Figure 4.3](#).

Looking at [Figure 4.1](#) it seems like the SLIC algorithm has potential in terms of image segmentation as well, but an issue is that it is hard to extract the correct edges without any prior knowledge. And the same issue regarding height estimation arises, since the algorithm will have to be combined with another method.

As previously stated, it has been shown that convolutional neural networks outperform the state-of-the-art methods for image segmentation in terms of accuracy, if they are provided with enough good training examples. There are several ways of acquire training data, such as automatically extract training examples from OpenStreetMap as described by [Kaiser et al. \(2017\)](#).

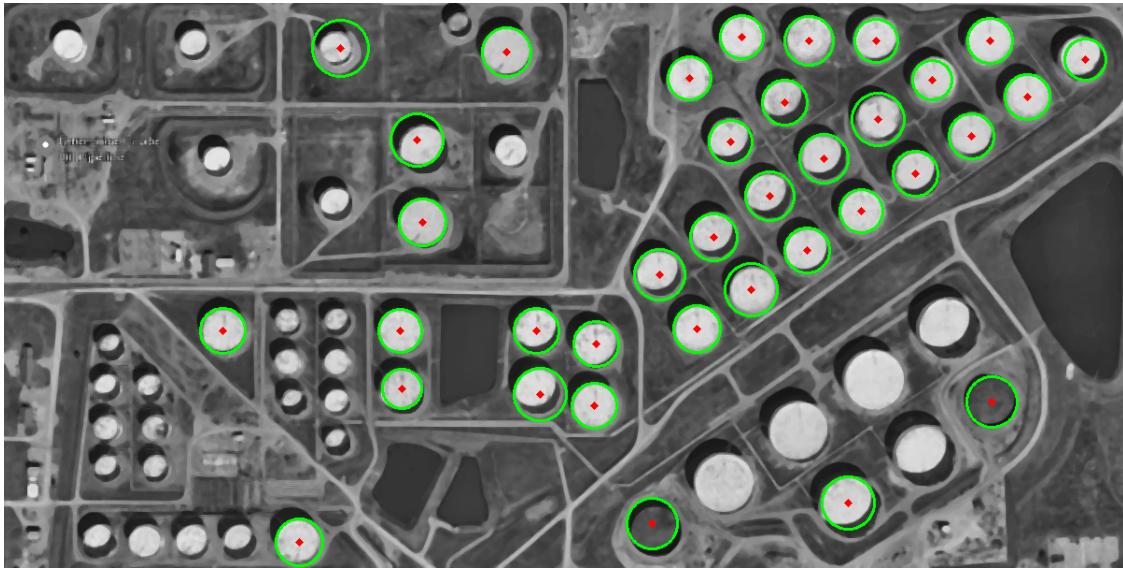


Figure 4.3: Detecting circles of different size in an image taken from Google Earth using a generalized hough transform

Since 2012 convolutional networks have been the leading method for object recognition and image segmentation. Furthermore, very deep convolutional neural networks such as the ResNet (Wu et al., 2017) have shown an incredible ability to create very complex functions for image analysis. The potential of these networks have yet not been fully explored, and there are still many applications that have not been discovered. One of them being building height extraction from aerial imagery.

After comparing the different methods, it has become clear that there are many ways to attack the presented problem. Below x different methods are presented as potential approaches to solving the problem:

- Direct estimation of the tanks inventory from shadow measurements using a modified ResNet
- Using a generalized Hough transform for tank detection, thus limiting the shadow search space, and then applying a modified ResNet for estimating a single tanks inventory.

Chapter 5

Conclusions, Discussion, and Recommendations for Further Work

Many attempts have been made in terms of semantic segmentation, and advances are still being made.

5.1 Summary and Conclusions

In this paper we have been investigating different approaches in order to estimate the total inventory of the famous price settlement point for West Texas Intermediate of the New York Mercantile Exchange in Cushing, Oklahoma. In order to do so, two different problems have been identified, which is locating the oil tanks in from an aerial image and estimating the inventory based on the height of the tanks floating roofs.

First the theoretical background for different methods are presented, including standard approaches for semantic segmentation, theory related to artificial neural networks, height estimation using SAR and InSAR analysis and trigonometrical height estimation using building shadows cast by the sun.

Then some related work related to this theory is presented.

5.2 Discussion

Here, you may discuss your findings based on your results, their strengths and limitations. Note that this discussion is more high level than discussions made in relation to results you have

achieved and presented in the previous chapter. The discussion here should put your work in larger context. You may address if you achieved what you had intended to do, why not (if you did not), if you got results in which you did not expect, why the results are important, why there are limitations in using the results, or if there are opportunities to transfer your results and findings into other domains, and so on.

5.3 Recommendations for Further Work

You should give recommendations to possible extensions to your work. The recommendations should be as specific as possible, preferably with an objective and an indication of a possible approach.

The recommendations may be classified as:

- Short-term
- Medium-term
- Long-term

Bibliography

- Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. (2012). SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2281.
- Airbus (2017). Technical Information : TerraSAR-X.
- Ballard, D. H. (1981). Generalizing the Hough Transform To Detect Arbitrary Shapes*. *Pattern Recognition*, 13(2):111–122.
- Brunner, D., Lemoine, G., and Bruzzone, L. (2008). Building Height Retrieval From VHR SAR Imagery Based on an Iterative Simulation and Matching Technique. 7110(3):71100F–71100F–12.
- Cambridge (2017). Convolutional Neural Networks with Keras.
- Comber, A., Umezaki, M., Zhou, R., Ding, Y., Li, Y., Fu, H., Jiang, H., and Tewkesbury, A. (2012). Using shadows in high-resolution imagery to determine building height. *Remote Sensing Letters*, 3(7):551–556.
- DigitalGlobe (2017). Our Constellation - DigitalGlobe.
- ESA. Overview - Copernicus.
- Esa (2009). 50 years of Earth Observation.
- Fisher, R., Perkins, S., Walker, A., and Wolfard, E. (2003). Image Transforms - Hough Transform.
- Frossard, D. (2016). VGG in TensorFlow.
- GISLounge (2014). Measuring Object Heights from Satellite Imagery.
- Kaiser, P., Wegner, J. D., Lucchi, A., Jaggi, M., Hofmann, T., Schindler, K., and Member, S. (2017). Learning Arial Image Segmentation from Online Maps. pages 1–15.
- Karpathy (2017). Convolutional Neural Networks for Visual Recognition.

- Kemker, R., Salvaggio, C., and Kanan, C. (2017). Algorithms for Semantic Segmentation of Multispectral Remote Sensing Imagery using Deep Learning. pages 1–45.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances In Neural Information Processing Systems*, pages 1–9.
- Lary, D. J. (2010). Artificial Intelligence in Geoscience and Remote Sensing. *RFID Technology, Security Vulnerabilities, and Countermeasures*, pages 75–100.
- Lary, D. J., Alavi, A. H., Gandomi, A. H., and Walker, A. L. (2016). Machine learning in geosciences and remote sensing. *Geoscience Frontiers*, 7(1):3–10.
- Le Cun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-Based Learning Applied to Document Recognition.
- Lin, G., Milan, A., Shen, C., and Reid, I. (2016). RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation.
- Liu, W., Suzuki, K., and Yamazaki, F. (2015). Height estimation for high-rise buildings based on InSAR analysis. *2015 Joint Urban Remote Sensing Event, JURSE 2015*, (June 2010):1–4.
- Mcculloch, W. S. and Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biology*.
- Minsky and Papert (1969). *Perceptrons*.
- Moon, H., Chellappa, R., and Rosenfeld, A. (2002). Optimal edge-based shape detection. *IEEE Transactions on Image Processing*, 11(11):1209–1227.
- Noh, H., Hong, S., and Han, B. (2015). Learning deconvolution network for semantic segmentation. *Proceedings of the IEEE International Conference on Computer Vision*, 2015 Inter:1520–1528.
- Patterson, J. and Gibson, A. (2017). *Deep Learning - A Practitioner's Approach*.
- Pinheiro, P. O., Lin, T. Y., Collobert, R., and Dollár, P. (2016). Learning to refine object segments. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9905 LNCS:75–91.
- Planet (2017). SkySat - Constellations.
- Shao, Y., Taff, G. N., and Walsh, S. J. (2011). Shadow detection and building-height estimation using IKONOS data. *International Journal of Remote Sensing*, 32(22):6929–6944.

- Simonyan, K. and Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. pages 1–14.
- Stanford (2017). Sliding Windows - Stanford University.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., Hill, C., and Arbor, A. (2014). Going Deeper with Convolutions. pages 1–9.
- Worboys, M. F. (2003). *GIS: A computing perspective*.
- Wu, S., Zhong, S., and Liu, Y. (2017). Deep residual learning for image steganalysis. *Multimedia Tools and Applications*, pages 1–17.
- Yu, Z., Feng, C., Liu, M.-Y., and Ramalingam, S. (2017). CASENet: Deep Category-Aware Semantic Edge Detection.
- Zeiler, M. D. and Fergus, R. (2013). Visualizing and understanding convolutional networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8689 LNCS(PART 1):818–833.
- Zeiler, M. D., Taylor, G. W., and Fergus, R. (2011). Adaptive deconvolutional networks for mid and high level feature learning. *Proceedings of the IEEE International Conference on Computer Vision*, pages 2018–2025.