


TECHNISCHE UNIVERSITÄT MÜNCHEN
Photogrammetrie und Fernerkundung

# Using Deep Convolutional Neural Networks for the Identification of Informal Settlements to Improve a Sustainable Development in Urban Environments

Thomas Stark

Master's Thesis

2018

TECHNISCHE UNIVERSITÄT MÜNCHEN
Photogrammetrie und Fernerkundung

# Using Deep Convolutional Neural Networks for the Identification of Informal Settlements to Improve a Sustainable Development in Urban Environments

Thomas Stark

Vollständiger Abdruck der von der Ingenieurfakultät Bau Geo Umwelt der Technischen Universität München zur Erlangung des akademischen Grades eines

Master of Science (M.Sc.)

genehmigten Master's Thesis.

| | |
|---|---|
| Prüfer der Master's Thesis: | Prof. Dr.-Ing. Uwe Stilla |
| Betreuer am DLR: | Dr. Michael Wurm<br>Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR)<br>Deutsches Fernerkundungsdatenzentrum |

Die Master's Thesis wurde am 15.02.2018 bei der Technischen Universität München eingereicht und durch die Ingenieurfakultät Bau Geo Umwelt am 15.02.2018 angenommen.

# Abstract

Currently about one-quarter of the world's urban population live in slums. Slums are defined by the United Nations (UN) as informal settlements or areas deprived of access to water, sanitation and durable housing. The buildings in slums are overcrowded and lack land tenure security. Slum-identification studies are very much driven by the persistence and growth of slums and the emergence of new slums being inexorably part of contemporary urbanization processes, particularly in the global south where rapid slum development is linked to the failure of formal land markets and low planning capacity. Identifying slums is an import aspect in urban environments of mega-cities. The information on location, boundaries and population in informal settlements is of great need for social economic studies and thus providing beneficial insight for a sustainable urban development. Beyond the identification of informal settlements and their physical parameters it is of great interest to provide these areas with an optimal fresh water-pipe infrastructure, since their supply of water is very limited.

The view from above using remote sensing data makes it possible to grasp the physical spatial settlement structures and, accordingly, to approach the characterizing parameters of slums and with this in mind image class segmentation on slum mapping can be done using different approaches. In recent years mainly object based, machine learning and texture classification approaches have been used to identify slums in urban areas. Regular machine learning tasks are limited because of their manually designed features. Another disadvantage of those methods is the inability to transfer the classifier to different datasets. This study provides a combination of methods in deep learning to achieve respectable accuracies in mapping informal settlements. Detected slums provide the prerequisite for establishing an optimal water supply network for all informal settlements. Since this procedure depends very much on the input geo-data, multiple ways of slum mapping using deep convolutional neural networks are presented and the cost of an optimal water-pipe network supplying all slum dwellers with water is calculated for Mumbai and Delhi.

Class segmentation performance was evaluated using overall and class based accuracy metrics. Using a pre-trained fully convolutional network resulted in an overall Pixel Accuracy for informal settlements of 78% and a mean Intersection over Union of 68%, while fine-tuned FCNs could achieve an overall Pixel Accuracy for informal settlements of 75% and a mean intersection over union of 63%. Using the best performing FCN a water supply infrastructure was built optimized to the shortest path connecting all slums using different approaches. The investment of a pipeline network providing clean water would cost about 16 million € for Mumbai and 12 million € for Delhi after 10 years of operation.

# Kurzfassung

Derzeit lebt etwa ein Viertel der urbanen Bevölkerung in Slums. Slums werden von den Vereinten Nationen (UN) als informelle Siedlungen definiert, in denen ein massiver Mangel an Zugang zu Wasser, sanitären Anlagen und dauerhaftem Wohnraum besteht. Die Behausungen in Slums sind überfüllt und es fehlt ihnen an baulicher Sicherheit. Studien zur Identifizierung von Slums basieren auf der Einbeziehung von Ausbreitung, Beständigkeit und Neuentstehung von Slums in die heutigen Urbanisierungsprozesse. Der Hauptgrund für die rasche Slumentwicklung, insbesondere im globalen Süden, ist das Versagen der Stadtplanung in Hinsicht auf den stetig wachsenden Zuzug in die Großstädte. Die Kartierung von Slums in Mega-Cities liefert wichtige Informationenen zu Ort, Grenzen und Bevölkerungszahl in informellen Siedlungen. Die daraus resultierenden Erkenntnisse sind das Fundament für sozialökonomische Studien und eine nachhaltige Stadtentwicklung. Durch eine möglichst genaue Identifizierung von Slums, kann somit auch deren größter Mangel, der limitierte Zugang zu Wasser, durch eine optimiertes Wasserversorgungs-Netzwerk behoben werden.

Der Blick von oben, ermöglicht es die physischen Siedlungsstrukturen zu erfassen und räumliche Parameter zu erhalten. Die Segmentierung von Slums kann mit unterschiedlichen Methoden erfolgen. In den letzten Jahren wurden vor allem objektbasierte Methoden, maschinelles Lernen und Textur-basierte Ansätze zur Kartierung von informellen Siedlungen verwendet. Bisherige Methoden basieren lediglich auf nutzerdefinierten Bildmerkmalen, des Weiteren ist die Übertragung der Klassifizierungsmethode auf andere Datensätze nur begrenzt möglich. In dieser Thesis werden mehrere Neurale Netzwerke verwendet, um eine höhere Genauigkeit bei der Kartierung von informellen Siedlungen zu erreichen. Diese Daten dienen als Grundlage für die Errichtung einer optimierten Wasserversorgungskette.

Die Segmentierung wurde anhand von gesamt- und klassenbasierten Genauigkeitsmetriken bewertet. Die Verwendung von *pre-trained fully convolutional neural networks* (FCN) ergab eine Pixelgenauigkeit informeller Siedlungen von 78% und einer *mean Intersection over Union* von 68%. *Fine-tuned* FCNs konnten eine Pixelgenauigkeit informeller Siedlungen von 75% erreichen und eine *mean Intersection over Union* von 63%. Mit dem besten FCN wurde eine Wasserversorgungsinfrastruktur geplant, welche auf einen kürzesten Weg optimiert ist und alle Slums mit Wasser versorgt. Die Investition eines Wasserversorgungsnetzwerks für Mumbai würde eine Ivestition von 16 Millionen € nach zehnjähriger Laufzeit zur Folge haben während ein Wasserversorgungsnetzwerk für Delhi rund 10 Millionen € nach zehn Jahren kosten würde.

# Contents

# List of Abbreviations

| Abbreviation | Description | Page |
|---|---|---|
| UN | United Nations | 15 |
| WHO | World Health Organization | 42 |
| NN | Neural Network | 19 |
| ANN | Artificial Neural Network | 19 |
| CNN | Convolutional Neural Network | 22 |
| DCNN | Deep Convolutional Neural Network | 23 |
| VGG-19 | CNN Architecture from the Visual Geometry Group in Berkley | 36 |
| FCN | Fully Convolutional Network | 29 |
| FCN-vgg19 | Fully Convolutional Neural Network using the vgg19 architecture | 36 |
| ReLU | Rectified Linear Units | 22 |
| Adam | Adaptive Moment Estimation Optimizer | 40 |
| GLCM | Gray Level Co-occurrence Matrix | 48 |
| RF | Random Forest Decision Tree Classifier | 34 |
| GSD | Ground Sampling Distance | 27 |
| VHR | Very High Resolution | 45 |
| NDVI | Normalized Differential Vegetation Index | 46 |
| IoU | Intersection Over Union | 53 |
| oPA | Overall Pixel Accuracy | 53 |
| OD | Origin Destination | 56 |

# List of Figures

# List of Tables

# 1 Background and problem statement

Humanity is in a process of migration that has resulted in dramatic changes to the global settlement landscape [Taubenböck & Wurm, 2015a]. The city of today is permanently changing and more dynamic than ever before. Suburbs are endless, cities merge and the centres grow into the sky. Urbanization is an elementary part of global change and since 2007 more people live in cities than in rural areas [UnitedNations, 2011]. Currently, about one quarter of the world's urban population lives in slums, which are defined by the United Nations (UN) as informal settlements or areas deprived of access to safe water, acceptable sanitation and durable housing [Kuffer et al., 2016]. The supply with fresh and clean water for peopling living in these areas is one of the main goals of modern civilization according to the UN [UnitedNations, 2015a].

The dynamic of urbanization varies locally. Thus, especially in developing countries, urban regions are experiencing rapid growth. This speed of today's growth is astounding. Cities such as Delhi, Lagos or Dhaka grow by 300,000 inhabitants per year [Burdett & Rhode, 2010]. This is even more extensive if one compares the population development with the available area in urban space. Cities occupy only a tiny space in relation to the entire land surface. Around four billion urbanites are concentrated in a space of about $0.24 - 2.75\%$ of the earth's surface [Schneider et al., 2009]. So what drives this process of urbanization? Migration of people can be explained by two effects: pull factors attract people from rural areas propelling them into the city. This can be statistically proven by a significant correlation between urbanization and well-being [Glaeser, 2010]. There are agglomeration effects that can provide a highly diversified supply and jobs, health care, educational and cultural offerings and infrastructure [Taubenböck & Wurm, 2015a]. These location factors give the city a radiance that magically attracts the rural population. The pull factors have mainly positive effects on the city and the newly arrived population, but there are still about one billion people living in slums of big cities. This can be attributed by so-called push factors, which pressures people into urban agglomerations. One example for such a push factor is the rise of new technologies for agricultural industrialization, which in turn means fewer jobs in the primary sector. This pressure from large agricultural companies is often flanked by globalisation of the economy [Harvey, 2013]. Thus, the rural population is forced to seek new job opportunities in the city by losing jobs in agriculture.

This gigantic migration into cities has significant consequences for the urban population, its influence on our environment and the face of urban planning in respect to general infrastructure. Today, more than half of the population live in urban areas and produce more than 80% of global gross domestic product [UnitedNations, 2011]. This can be particularly harmful to the environment, since $60 - 80\%$ of the global energy is consumed in cities and in turn causes about 75% of global $CO_2$ emissions [Kamal-Chaoui & Robert, 2009]. This challenge must be considered when planning the infrastructure of big cities in the future.

At the same time, the design of physical space has a profound impact on social coexistence and social cohesion [Burdett & Rhode, 2010]. The rapid increase in people moving to urban areas has led to extremely diverse developments in how this concentration of immigrants is absorbed. Depending on the location, the change in our cities has taken on threatening forms. In developed countries, for example, the urban population only increases moderately and an orderly urban planning is possible. In contrast, developing countries are experiencing rapid growth. Between 1990 and 2000, urban space grew by about 50% in developing countries [Angel et al., 2005]. With this extreme area growth, structured urban planning is very demanding. Thus questions arise as to whether there is enough information about the effect of global urbanization and how to spot such an explosion in population growth in urban areas in a timely manner so that global urbanization does not get out of control.

This rapid growth in mega cities can lead to informal settlements, where slums are forming a spatially disordered cluster with no uniform infrastructure [Friesen et al., 2017]. These circumstances negatively affect the physical and psychological health of the slum dwellers [Snyder et al., 2014]. This led the United Nations to record their goals for sustainable development. One goal addresses the right of every human to access to water. In reality this is not an easy task, especially in informal settlements where not much official information exists concerning location, boundaries and the number of inhabitants.

## 1.1 Global urbanization and its effects on urban poverty

Many people think of slums as poor, neglected, very dense and randomly arranged cottage settlements. In our perception a slum is first a place with a certain physical expression. This is also stated in the United Nations report [UnitedNations, 2009] as a combination of physical and socio-structural parameters. According to their definition, a slum is described as having poor access to clean water, lack of sanitation, poor infrastructure and physical parameters such as poor terrain, overcrowding and unsafe status in relation to the place of residence. The names of slums vary strongly across countries and literature, e.g. *favelas, barriadas, shantytowns, informal / spontaneous / marginal / squatter settlements, gececondus, ashwa'iyyat, bidonville and townships* are used to describe urban poverty in various physical manifestations featuring a wide range of built-up structures [Wurm & Taubenböck, 2018]. In this thesis the term slum and informal settlements will be used synonymously.

Although this description applies to a large part of globally distributed slums, in general the physical structure of informal settlements is always different than formal buildings, but at the same time slums also always differ from each other. This heterogeneity makes it very difficult to identify all poverty areas on earth. Depending on continent and culture, slums never have exactly the same physical structure. For example, slums in Mumbai can be used as a single-storey hut with a corrugated iron roof; in Bucharest, pointed and hipped roofs are the norm; and in São Paulo multi-storeybrick structures are standard [Taubenböck & Kraff, 2015]. A universal physical definition is therefore very difficult to find [Wurm & Taubenböck, 2018]. Using state-of-the-art earth observation data from remote sensing satellites, slums can be identified on a large scale. To develop successful slum mapping methods it is especially important to deal with the properties of informal settlements to understand the behavior of their physical parameters.

Today, around the world, a quarter of the urban population live in slums [UnitedNations, 2016]. In developing countries 881 million urban residents live in slum conditions. In 1990, this figure was 689 million. This represents an increase of 28% in slum population over the past 26 years, even though the proportion of the urban population in developing countries living in slums has declined from 39% to 30% during the same period. This shows the sheer dimension of

agglomeration effects in urban areas. In Asia and the Pacific, home to half of the urban population of the world, 28% of the urban population resides in slums [UnitedNations, 2016]. Considering the predicted migration waves and urbanization rates, cities will not be able to oppose the huge demand for living space in the future [Taubenböck & Kraff, 2015].

Informal settlements are areas deprived of access to safe water, acceptable sanitation and durable housing [Kuffer et al., 2016]. These circumstances negatively affect the physical and psychological health of the slum dwellers [Snyder et al., 2014]. The supply with fresh and clean water for people living in these areas is one of the main goals of modern civilization according to the UN [UnitedNations, 2015a]. This led the United Nations to record their goals for sustainable development where one goal addresses the right of every human to access to water. The aim is to support urban planning globally in such a way that it is both prepared for migration and urbanization in a sustainable manner. Another aim is to recognize the current state of global urban poverty and to improve it through recent urban planning methods with the help of which enough fresh water could be provided for slum dwellers. At the moment, however, there is only rudimentary knowledge about the location, quantity, area size, growth, settlement structures or population of the slums all around the world and hence providing a water supply chain for slum dwellers is not an easy task. The duties of a global and urban generation must therefore be able to spatially grasp these areas and accurately measure their structural features. This is why the collection, monitoring and water supply of slums has become the focus of interest for the work with the Millennium Goal 7D [UnitedNations, 2015b]. With the increasing availability of accessible satellite data, modern methods can be used to significantly increase the understanding of slum processes and to help cities improving living conditions in slums with sustainable urban planning.

## 1.2 Introduction to the study areas

The goal of this study is working towards a method that can be used in multiple scenarios rather than concentrating on one specific location. The methodology is tested and validated for the cities of Mumbai, India and New Delhi, India. Both cites share the same architectural structure, but its informal settlements are different enough to test the method on various geospatial entities. As seen in table 1.1 both cites are extremely big and experienced an exceptional growth in the last 60 years.

| | Mumbai | | Delhi | |
|---|---|---|---|---|
| Census | Population | Growth rate [%] | Population | Growth rate [%] |
| 1950 | 2,857,000 | 0.0 | 1,369,000 | 0.00 |
| 1975 | 7,082,000 | 21.87 | 4,426,000 | 25.35 |
| 1990 | 12,436,000 | 19.68 | 9,726,000 | 32.78 |
| 2000 | 16,367,000 | 14.37 | 15,732,000 | 26.80 |
| 2010 | 19,422,000 | 8.56 | 21,935,000 | 17.49 |
| 2017 | 21,690,000 | 3.07 | 27,197,000 | 5.81 |

Table 1.1: According to the census in 2011 [Office of the Registrar General & Census Commissioner, 2011], the population of the city of Mumbai and Delhi in comparison.

Mumbai is the capital city of the Indian state of Maharashtra. It is one of the most populous cities in India with an estimated city center population of 12.4 million as of 2011. Along with

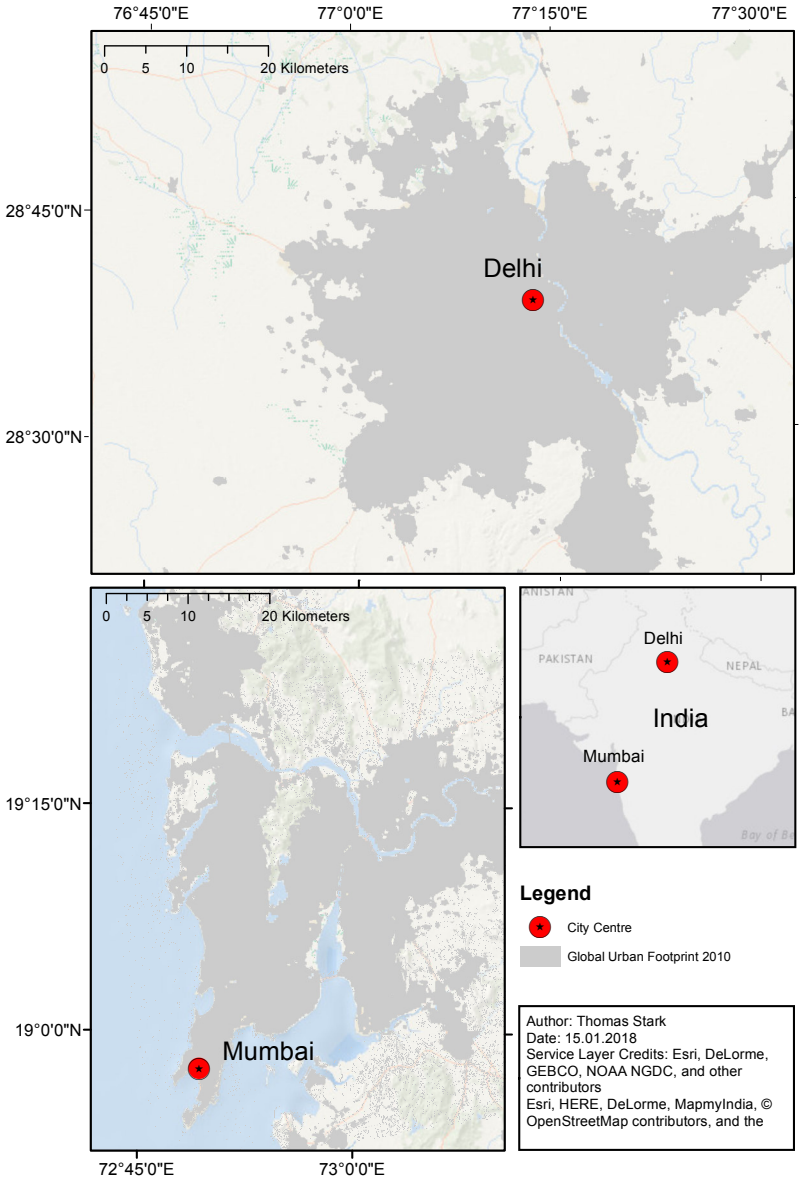


Figure 1.1: Overview of the study areas used in this thesis. The illustration on the top shows the city of Delhi, while the one on the bottom left shows Mumbai.

the neighboring regions of the Mumbai Metropolitan Region, it is the second most populous metropolitan area in India, with a population of 21.3 million as of 2016. Mumbai is the financial, commercial and entertainment capital of India. It is also one of the world's top ten centers of commerce in terms of global financial flow [Rashmi, 2011], generating 6.16% of India's GDP [MMRDA, 2008] and accounting for 25% of industrial output. All these factors are pull-factors bringing in people from rural areas in search for work and success.

New Delhi is the capital of India and one of the eleven districts of Delhi City. Although colloquially Delhi and New Delhi are used interchangeably to refer to the National Capital Territory of Delhi, they are two distinct entities, with New Delhi forming a small part of Delhi. Delhi has served as the political and financial centre of several empires of ancient India and the Delhi Sultanate, most notably of the Mughal Empire from 1649 to 1857. During the early 1900s, a proposal was made to the British administration to shift the capital of the British Indian Empire,

as India was officially named, from Calcutta to the east coast, to Delhi. Delhi is the largest commercial center in northern India. As of 2016 recent estimates of the economy of the Delhi urban area have ranged from $167 to $370 billion (PPP metro GDP) ranking it either the most or second-most productive metropolitan area of India.

## 1.3 Background on deep learning and neural networks

The perceptron based model is the foundation of the earliest Neural Networks (NNs). This bio-inspired model for binary classification aims to mathematically formulize how a biological neuron works, which can be seen in an comparison of perceptron and artificial neural networks (ANN) in figure 1.2. Neurons are the basic computational units of the human brain, which receive input signals from dendrites and send output signals through axons. This coarse mathematical model of linear combination can be seen in equation 1.1. The centre of every ANN consists of the input data $\mathbf{x}$ and learnable weights $\mathbf{w}$, where weights act excitatory or inhibitory. If these integrated weighted inputs exceed a certain threshold, the neuron fires and carries the information through the network. This firing rate is controlled by the activation function.

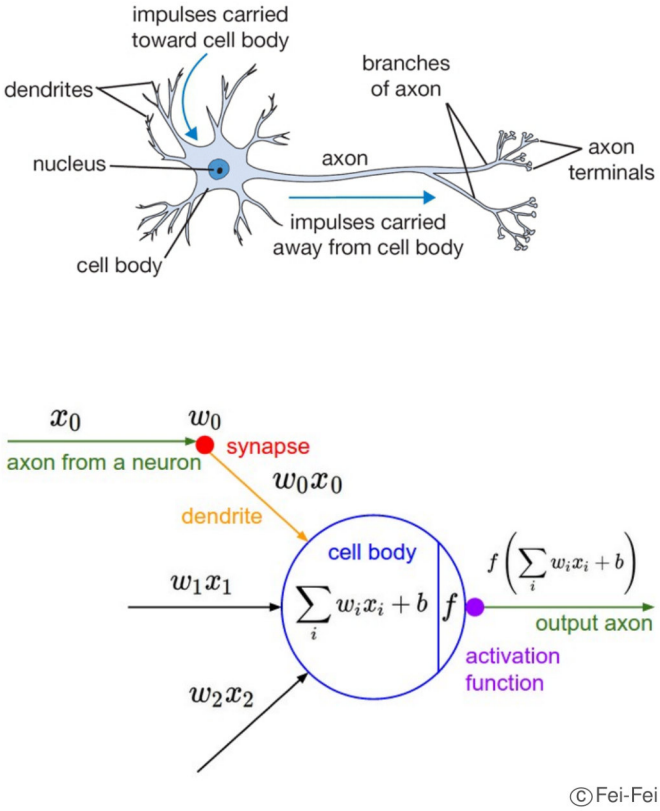


Figure 1.2: A comparison of a bio-inspired perceptron model and an artificial neural network [Körner, 2016].

$$\begin{aligned} &\mathbf{x}^T\mathbf{w} + w_0 \\ &\quad \mathbf{x}^T : \text{input data} \\ &\quad \mathbf{w} : \text{learnable weights} \end{aligned} \tag{1.1}$$

Perceptrons were developed in the 1950s and 1960s by the scientist Frank Rosenblatt [Rosenblatt, 1958], inspired by earlier work of Warren McCulloch and Walter Pitts [McCulloch & Pitts, 1943]. Today, it is more common to use sigmoid neurons seen as the cell body in figure 1.2. The sigmoid neuron has multiple inputs $x_0, x_1, x_2, \ldots$, but instead of being just 0 or 1 as present in perceptrons, these inputs can also take on any values between 0 and 1. Also just like a perceptron, the sigmoid neuron has weights for each input $w_0, w_1, w_2, \ldots$ and an overall bias $b$. But the output is not 0 or 1. Instead, it is $\sigma(wx + b)$, with $\sigma$ being called the activation function.

The general terminology of the architecture in ANNs makes it possible to name different parts of a network. This can be exemplified with illustration 1.3. The leftmost layer in this network is called the input layer, and the neurons within the layer are called input neurons. The rightmost or output layer contains the output neurons, or, as in this case, a single output neuron. The middle layer is called a hidden layer, since the neurons in this layer are neither inputs nor outputs. In a feedforward neural network, which means that there are no loops in the network an information is always fed forward, an algorithm finds weights and biases so that the output from the network approximates $y(x)$ for all training inputs $x$. To quantify how well this goal is achieved, a loss function is defined as seen in equation 1.2.

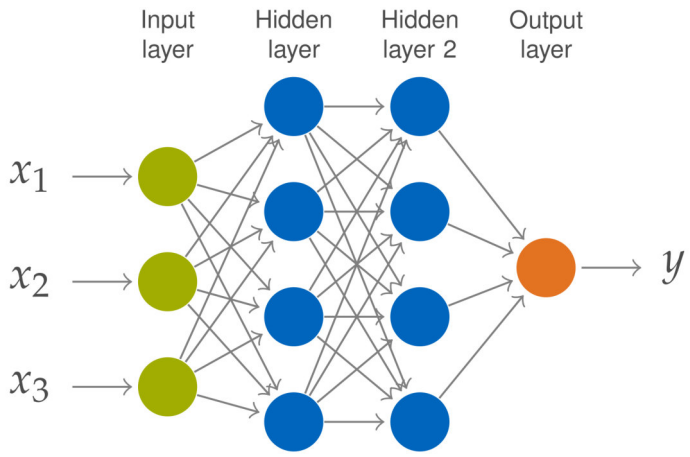


Figure 1.3: Layer-wise organization of an artificial neural network containing input-, hidden- and output layers. [Körner, 2016].

$$C(w,b) = \frac{1}{2n} \sum_x ||y(x) - a||^2 \tag{1.2}$$

$C(w,b)$ : Quadratic cost function or mean squared error

Inspecting the form of the quadratic cost function, $C(w,b)$ is non-negative, since every term in the sum is non-negative. Furthermore, the cost $C(w,b)$ becomes smaller, i.e., $C(w,b) \approx 0$, precisely when $y(x)$ is approximately equal to the output $a$ for all training inputs $x$. So the training algorithm is effective if it can find weights and biases so that $C(w,b) \approx 0$. By contrast, it is not effective if $C(w,b)$ is large; that would mean that $y(x)$ is not close to the output for a large number of inputs. Consequently the aim of that training algorithm will be to minimize the cost $C(w,b)$ as a function of the weights and biases. In other words, a set of weights and biases which make the cost as small as possible is searched. Using an algorithm known as gradient descent this process can be achieved.

Assuming a function $C(v)$ that has to be minimized and at the same time being a function of just two variables $v_1, v_2$, the point where $C$ achieves its global minimum has to be found. Using calculus to try to find the minimum analytically is ruled out since in a real scenario the $C(v)$ does not have just two variables but billions of weights. After a random starting point is chosen the momentum is moved a small amount $\Delta v_1$ in the $v_1$ direction, and a small amount $\Delta v_2$ in the $v_2$ direction. The gradient descent algorithm works by repeatedly compute the gradient $\nabla C$, and then to move in the opposite direction. To make gradient descent work correctly, choosing the right learning rate $\eta$ is crucial, where $\Delta C \approx \nabla C \Delta v$ would be a good approximation. At the same time, $\eta$ shouldn't be to too small since that will make the changes $\Delta v$ tiny and thus the gradient descent algorithm would work very slowly. In practical implementations, $\eta$ is often varied, recognizing that $\Delta C \approx \nabla C \Delta v$ remains a fine approximation, but not slowing down the algorithm. This makes it possible to follow the gradient to a minimum, even if $C$ is a function of many variables, by repeatedly applying the update rule, which can be seen in equation 1.3. It enables to repeatedly change the position $v$ in order to find a minimum of the function $C$ this is a powerful way of minimizing the cost function.

$$\begin{aligned} v \rightarrow v' =& v - \eta \nabla C \\ \nabla C =& (\frac{\partial C}{\partial v_1}, ..., \frac{\partial C}{\partial v_n}) \end{aligned} \tag{1.3}$$

Training neural networks is an optimization problem containing a forward and backward propagation through the network. For each neuron in the network extremal values in an objective function $f(x)$ are searched and its partial derivates express the sensitivity of $f$ on each parameter. During forward propagation the data goes straight through each layer. Back-propagation is about understanding how changing the weights and biases in a network changes the cost function. Ultimately, this means computing the partial derivatives. Back-propagation is based around computing both the error and the gradient of the cost function, which can be seen in the four fundamental equations in 1.4 [Nielson, 2015].

$$\begin{aligned} \delta_j^L =& \frac{\partial C}{\partial a_j^C} \sigma'(z_j^L) \\ & \delta_j^L : \text{Error in the output layer } L \text{ and } j\text{-th output activation } \textbf{[I]} \\ \\ \delta^l =& ((w^{l+1})^T) \delta^{l+1} \circledast \sigma'(z_j^L) \\ & \delta^l : \text{Error in the next layer } \textbf{[II]} \\ \\ \delta_j^l =& \frac{\partial C}{\partial b_j^l} \\ & \delta_j^l : \text{Rate of change of the cost with respect to bias } \textbf{[III]} \\ \\ a_k^{l-1} \delta_j^l =& \frac{\partial C}{\partial w_{jk}^L} \\ & a_k^{l-1} \delta_j^l : \text{Rate of change of the cost with respect to weight in the network } \textbf{[IV]} \end{aligned} \tag{1.4}$$

The first equation in 1.4 is a very natural expression. The first term on the right, $\partial C / \partial a_j^L$ just measures how fast the cost is changing as a function of the $j$-th output activation, while the

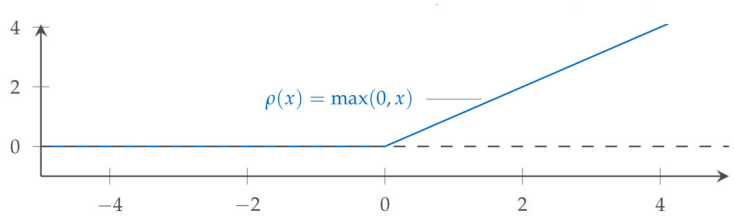


Figure 1.4: Range of a rectified linear units [Körner, 2016].

second term on the right, $\sigma'(z_j^L)$, measures how fast the activation function $\sigma$ is changing at $z_j^L$, which is the weighted input to the neurons in layer $L$ for the $j$-th output activation. This first equation is a component wise expression for $\delta^L$. The second equation is for the error $\delta^l$ in terms of the error in the next layer $\delta^{l+1}$. The error $\delta^{l+1}$ at the $l+1$-th layer is known and the transpose weight matrix $(w^{l+1})^T$, can be intuitively thought of as moving the error backward through the network. This provides some sort of measure of the error at the output of the $l$-th layer. The Hadamard product $\circledast\sigma'(z_j^L)$ moves the error backwards through the activation function in layer $l$, supplying the error $\delta^l$ in the weighted input to layer $l$. By combining the first two equations the error $\sigma^l$ can be computed for any layer in the network. At first equation 1.4[I] is used to compute $\sigma^L$, then equation 1.4[II] will compute $\sigma^{L-1}$. Equation 1.4[II] is computed again to gain $\sigma^{L-2}$, and so on, all the way back through the network. The third equation 1.4[III] establishes the rate of change of the cost with respect to any bias in the network. That is, the error $\delta_j^l$ is exactly equal to the rate of change $\partial C/\partial b_j^l$. The last equation 1.4[IV] has the purpose to measure the rate of change of the cost with respect to any weight in the network. This shows how to compute the partial derivatives $\partial C/\partial w_{jk}^L$ in terms of the quantities $\delta^l$ and $a^{l-1}$. Here it is understood that $a^{l-1}$ is the activation of the neuron input to the weight $w$, and $\delta_j^l$ is the error of the neuron output from the weight $w$.

There are other insights along these lines which can be obtained from the four equations in 1.4. Starting with illustration 1.4 and considering the term $\sigma'(z_j^L)$ in equation 1.4[I]. The rectified linear unit (ReLU) activation functions are in the range of $[0, \infty]$. When the activation function is less than 0, the term $\sigma'(z_j^L)$ is also 0. Consequently a weight in a layer will stop learning for negative values and increase for positive values. In the first case it's common to say the output neuron has saturated and, as a result, the weight has stopped learning. Similar remarks hold also for the biases of neurons.

To adapt a ANN to a Convolutional Neural Network (CNN) some changes to its architectures are necessary. A CNN consist of various types of layers all working together and being equally important to the success of the classification. Each layer accepts a 3D volume as input and produces another 3D volume as output by differential functions. A CNN consists of convolutional, fully connected, pooling and activation layers. While convolutional and fully connected have learnable parameters that can't be tuned by the user, the aforementioned and pooling layers have hyper-parameters that can change their definition of passing data through the network by adapting their 3D volume parameters. The following list gives an overview on the most common layer types:

- Convolutions compute a weighted sum $\mathbf{w}^T\mathbf{x}$ of input pixels from a small region that can be regarded as a linear convolution operation. The source regions can overlap, which is considered as a stride. Convolutions extract low-level, mid-level and high-level features throughout the network.

- Pooling layers usually follow convolutional layers and reduce the size of an image patch by downsampling. Most prominently by a max operator.
- Fully connected layers are the final output layer and compute a vector of class probabilities. These layers reduces the full image to a vector of class scores.
- Activation functions, most commonly rectified linear units (ReLU) act as a firing unit to carry information through the network when a certain threshold is exceeded. ReLUs accelerate the learning with their linear structure and are zero centered which adds the benefit of having not only positive gradients during the training operation.

## 1.4 Working hypothesis on identifying slum areas in mega cities

The aim of this study is to explore the capabilities of deep learning techniques to detect slums using very high resolution optical imagery. For this purpose two mega cities in India, Mumbai and Delhi, are chosen for investigation and covering in total almost $200qm^2$. The area of interest contains about 341 official named slums of various sizes, structures and are within a different urban context between the two cities.

In a broad experimental setup, the ability to transfer knowledge learned from a deep convolutional neural network (DCNN) is used to detect informal settlement in different urban environments. Results are compared with multiple training techniques from the same DCNN. Large scale training data is created, containing at its core a random forest classification using texture based and spectral features. Since slum dwelling often occupy only a small share of a city's total area [Taubenböck & Wurm, 2015b], a pixel based classification has to take this imbalanced class distribution into account [Wurm et al., 2017]. This common field of data mining [Weiss, 2004] also occurs in remote sensing applications [Wright & Gallant, 2007], [Williams et al., 2009]. This imbalance of informal settlements and formal build-up structures is reworked using enforced learning techniques during the training phase and its classification method of a DCNN.

A comprehensive work flow is used to produce a high quality mask for informal settlements in Mumbai and Delhi. These results provide geodata which is used as an input for a mathematical optimization to find optimal routes for a water supply chain connecting all informal settlements to a fresh water pipe network. Prior studies researching optimal water supply networks for informal settlements used only large slums and pipeline networks not along the road infrastructure [Rausch et al., 2018]. In this thesis a thorough approach on connecting all detected informal settlements to an optimal water supply network along the road infrastructure is presented.

This study of slum mapping using a DCNN for different learning techniques and connecting all detected informal settlements with an optimal fresh water pipe network is guided by the following research questions:

- Is it possible to use DCNNs to differentiate formal from informal settlements?
- Using transfer learning, which are the optimal scenarios for best possible slum mapping results?
- How can a water supply infrastructure be positioned in an optimal network connecting all detected informal settlements?

At the beginning of this thesis the geospatial structure of informal settlements and its textural features is discussed in section 2.1.1. Afterwards a state of the art review on deep learning in

remote sensing context is given in section 2.2. The proposed method of class-segmentation on urban poverty to provide water for the poor in explained in section 3. The classification process is described in detail in section 3.1 including its data preprocessing and the creation of high quality ground truth data and the used DCNN for the classification of informal settlements. Section 3.2 explains the mathematical optimization of the fresh water pipe network. An extensive experimental set up was done to provide large-scale research data on transfer learning methods of different variation using the DCNN and is present in section 4. The final results are shown in section 5 with a following discussion and conclusion in section 6.

# 2 State of the art

There is a growing international motivation to reduce the number of slum dwellers by gathering information of informal settlements for policy relevant organizations [UnitedNations, 2015a]. With more available high and very high resolution satellite data and advances in geospatial processing tools, a growing number of methods for urban classification are present [Kuffer et al., 2016]. Even though informal settlements show different morphological characteristics as seen in Table 2.1, slum mapping still suffers from rather low identification accuracies [Vaz & Berenstein, 2004]. This challenge is part of the motivation of providing better identification results than previous methods by using state of the art classification algorithms and combine these with expert knowledge to identify informal settlements.

| Features | Slum areas | Formal built up areas |
|---|---|---|
| Size | Small buildings/huts | Larger building sizes |
| Density | High roof coverage density<br>Lack of public (green) spaces | Low to moderate building density<br>Provision of public (green) spaces |
| Pattern | No orderly infrastructure arrangement | Regular and planed infrastructure |
| Site characteristics | Hazardous locations (flood prone or steep slope) | Land has basic suitability for built up areas |

Table 2.1: Morphological features typical for slum areas according to [Kuffer et al., 2014] and [Baud et al., 2010].

## 2.1 Geospatial structure of urban poverty

Very high resolution (VHR) remote sensing imagery provides a detailed representation of the physical elements and characteristics of informal settlements. Since the definition of what constitutes as a slum is very complex, multiple variations exist depending on global, continental or regional factors [Risbud, 2002]. This is proven by the fact that all mapped informal settlements are characterized by incomes below the poverty line [Wurm & Taubenböck, 2018]. The houses or huts in informal settlements and the resulting unorganized structure in theses areas do not share a universal form. A comparison of selected houses in slums reveals significant differences: These are masonry one-storey buildings with corrugated iron roofing (in Dharavi, Mumbai, India), or pointed roof and hipped roof (in Tei Toboc, Bucharest, Romania) over multi-storey brick structures (in Paraisopolis, Sau Paulo, Brazil) or a makeshift collection of different building materials up to two stories per cottage (in Kayekitsha, Cape Town, South Africa). Figure 2.1 shows four informal settlements for both Mumbai and Delhi and their differences within informal settlements and the change to its counterpart of formal buildings indicates that a universal definition in terms of physical structure is very ambiguous.


Figure 2.1: Informal settlements in Mumbai and Delhi illustrating the differences between each other and their surrounding formal buildings. The top row shows slums in Delhi, while bottom row shows slums in Mumbai.

Nevertheless, a physical approach to identify and characterize slums is applicable. Especially useful is the availability of modern earth observation data in the style of the Big Data Revolution in area-wide and consistent datasets. These datasets make it possible to record small-scale urban structures and to aim for cross-city analyses. Therefore, the aim is to record the location and morphological characteristics of exemplary informal settlements. The focus of this study is to determine how slums can be physically detected from earth observation data. Furthermore gaining knowledge of spatial analyses and location characteristics to improve slum mapping. Lastly it is of great interest to gain information about morphological differences of informal settlements between different cities on a single continent. The view from above makes it possible to grasp the physical and spatial settlement structures. This makes it possible to identify the characterizing parameters of slums. In order to localize urban poverty with earth observation data, a connection between the top down view and the physical spatial features of informal settlements must be produced. This can be seen in 2.1, the data excerpts show poverty districts in different cultivated areas in very high-resolution optical satellite data. In the process, the high density of the building structures, which are also unevenly arranged and have very heterogeneous building types, emerges as a visual characteristic.

### 2.1.1 Urban geographic analysis and geospatial metrics significant to slum mapping

Space plays a central role in urban geographic analyses and comparisons. Against this background, [Taubenböck & Kraff, 2015] give an overview of the structural and morphological analysis of informal settlements, which are subsequently carried out on two spatial levels; the level of the city as a whole and the level of the slums. As seen in illustration 2.1 it is visible that informal settlements contain a high building density. These buildings are unequally distributed with a high diversity their structures.

[Taubenböck & Kraff, 2014] state that a building density for slums is with about 75% significant higher than the 40% for formal building structures. The variance of building sizes and heights is significantly lower in informal areas, with consistently lower building size and height. This exemplary, morphological proof, that poor building structures can be differentiated from earth observation data from formal bites, establishes the basis for classifying informal settlements in a comprehensively monitored manner.

[Hollis, 2013] raises the question if slums are a universal phenomenon or whether each neglected neighbourhood arises due to individual reasons and its own history and accordingly form the varying settlement structures. A comparison of [Taubenböck & Kraff, 2014] aims at whether the phenomenon of slums actually creates a universal settlement structure. The measured building densities in slums are very high compared to planned settlements. The medians consistently show building densities higher than 50%. Likewise, a fundamentally low building size is evident in all examined slums. The medians are approximately 20 to 35 square meters per building. The dimension of this combination - high density and small buildings - is also clear from the fact that, projected on a square kilometer in Paraisópolis, São Paulo Brasil, more than 33.200 buildings are present. The intensity of this land use is extremely high in all informal settlements. However, the sometimes high variances in building densities and sizes in slums show that organic settlement development do not form homogeneous structures, but rather an individual, heterogeneous network of settlements develops within each slum. The fact that organic settlement development does not form homogeneous structures becomes quantitatively measurable through the heterogeneity of informal settlements.

A physical analysis is therefore useful to spatially identify and characterize areas and to understand the morphological processes. To come to a conclusion, there are physical features that globally characterize the settlement structures of slums, but there are no universally valid parameters.

### 2.1.2 Recent slum mapping approaches

The complexity of physical slum characteristics requires advanced sensor systems for mapping purposes. The following section provides an overview of the requirements and recent trends in slum mapping. As stated above the physical parameters of slums are explored in a high building density and small building sizes, so the spatial and radiometric requirements for slum mapping are quite high. Spectrally most of the optical imagery have 2-3 bands in the visible range and 1-2 bands in the infra-red. [Jacobsen & Büyüksalih, 2008] report that the ground sampling distance (GSD) for building objects should be $2m$, however detailed building object information requires a GSD of $0.5m$ and a sufficient contrast between buildings and its surroundings according to [Jensen & Cowen, 1999]. In a detailed study [Kuffer et al., 2016] present an overview of recent trends in slum mapping. Among the reviewed studies, multiple methods have been used to classify slums. The majority of slum classification analyses study the extraction on entire slum areas as seen in table 2.2. Apart from object based image analysis, visual image representation and standard pixel based image classification a recent trend shows an increase in machine learning methods, where researchers used neural networks [Persello & Stein, 2017][Dell'Acqua et al., 2006], random forest [Wurm et al., 2017] or support vector machines [Huang et al., 2015]. Machine learning tasks are information driven approaches that allow for a repetitive learning from a large and rich set of training data [Niebergall et al., 2008].

Since the report on an expert meeting on slum mapping in 2008 [Sliuzas et al., 2008] more methods have been explored, expanding the global knowledge repository of slum characteristics and their variability. According to [Brito & Quintanilha, 2012], feature extraction of optical data is the most commonly used for slum mapping, but there is no clear agreements on the

| | Methods | | | | |
|---|---|---|---|---|---|
| | Machine Learning | Object-Based | Statisical Model | Texture Morphology | Visual Image Interpretation |
| Identification of slum areas | 11 | 15 | 2 | 9 | 11 |

Table 2.2: Frequency of methods for slum mapping using VHR imagery according to [Kuffer et al., 2014].

most successful method. Thus there is a strong need for new approaches in automatic image understanding on remote sensing data interpretation.

The last dimension of the analysis deals with the performance of these methods, measured by their reported accuracy levels. According to the review of [Kuffer et al., 2016], advanced approaches, such as mathematical and morphological analysis, have a better performance than standard classification methods [Giada et al., 2003]. The highest mean accuracy is obtained by machine learning approaches, but also texture and statistical based methods, while the variance of object based approaches is rather large, due to the very complex and heterogeneous environments of informal settlements.

As of right now, only few other studies used convolutional neural networks in their approach to detect informal settlements [Persello & Stein, 2017][Mboga et al., 2017]. While these recent studies show promising results of using patched based CNNs and fully convolutional networks (FCNs) with overall pixel accuracies from 81% up to 92%, both use rather shallow networks of $3-6$ hidden layers. When working towards a wider continental approach of slum mapping, where the transferability of learned knowledge in DCNNs becomes necessary, no study has been done yet generalizing semantic class segmentation of multiple cites to classify informal settlements, since the use of transfer learning is very successful in various scenarios [Oquab et al., 2014].

In conclusion machine learning methods tend to work very good if aiming at extracting slum areas at a city scale and object based approaches were found quite successful to extract single buildings in informal settlements. However, there are no studies on evaluating the transferability of using DCNNs on a large scale slum mapping approach of different cities.

## 2.2 Deep learning for remote sensing satellite data

Most machine learning methods work well because of human-designed representations and features, in this case machine learning becomes optimizing weights to make an optimal prediction. Representation learning attempts to automatically learn good features or representations, which works well for small problems. But manually designed features are often over-specified, incomplete, and take a long time to design and validate. Deep learning algorithms attempt to learn multiple levels of representations and outputs [Körner, 2016]. So it is no surprise that deep learning is one of the fastest growing trend in machine learning tasks as seen in illustration 2.2. Deep learning is driven by neural networks which exploit feature representation exclusively learned from its input data. Recent advances in the field have proven deep learning a very successful set of tools, sometime even able to surpass human ability to solve highly computational tasks [Zhu et al., 2017]. Especially for image representation convolutional neural networks have proven to excel at extracting mid- and high level abstract features from raw images. Recent studies indicate that the feature representations learned by CNNs is greatly effective in large scale image recognition

[Krizhevsky et al., 2012][Simonyan & Zisserman, 2014], object detection [Girshick et al., 2016] and especially relevant for this study semantic segmentation [Long et al., 2015].

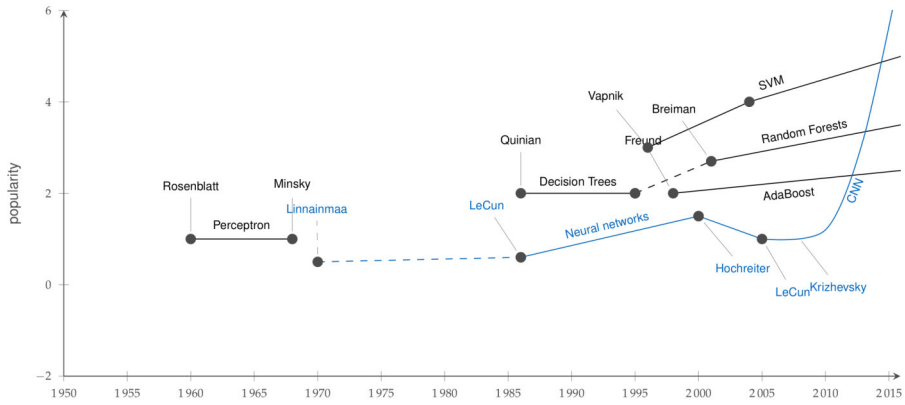


Figure 2.2: Popularity of research papers in machine learning and the rise of neural networks [Körner, 2016].

### 2.2.1 Semantic image segmentation with convolutional neural networks

Semantic segmentation means understanding an image at pixel level i.e., each pixel of an image is assigned an object class. Before deep learning took over computer vision, people used approaches like Support Vector Machine [Mountrakis et al., 2011] and Random Forest based classifiers [Wurm et al., 2017][Belgiu & Drăguţ, 2016] for semantic segmentation with remote sensing datasets. As with image classification, convolutional neural networks (CNN) have had enormous success in solving segmentation problems. One of the popular initial deep learning approaches was patch classification where each pixel was separately classified into classes using a patch of images around it. The main reason to use patches was that classification networks usually have full connected layers and therefore required fixed size images. In 2014, Fully Convolutional Networks (FCNs) [Long et al., 2015] popularized CNN architectures for dense predictions without any fully connected layers. This allowed segmentation maps to be generated for images of any size and it was also much faster compared to the patch classification approach. Almost all subsequent state of the art approaches on semantic segmentation adopted this paradigm. Apart from fully connected layers, one of the main problems using CNNs for segmentation are the pooling layers. Pooling layers increase the field of view and are able to aggregate the context while discarding the location information. However, semantic segmentation requires the exact alignment of class maps and thus, needs the 'where' information to be preserved. Two different classes of architectures evolved in the literature to tackle this issue [Sasank, 2017]. While encoder-decoder architectures work with an encoder which gradually reduces the spatial dimension with pooling layers and a decoder that gradually recovers the object details and spatial dimension, fully convolutional networks use so-called dilated/atrous convolutions [Yu & Koltun, 2015] and do away with pooling layers.

The work from [Long et al., 2015] for fully convolutional neural networks is probably the most important work in deep learning for semantic segmentation. The key observation is that fully connected layers in classification networks can be viewed as convolutions with kernels that cover their entire input regions. This is equivalent to evaluating the original classification network on overlapping input patches but is much more efficient because computation is shared over the overlapping regions of patches. To perform this task, the output of the final fully connected layers of the CNN must be of the same pixel size as the input and not a vector shape assigning pictures

to classes. The network of [Long et al., 2015] introduces many significant ideas, for example like end-to-end learning, where an up-sampling algorithm down-samples the activations size and then up-samples it again. Using fully convolutional architectures allows the network to use input images of arbitrary sizes as an input since there is no fully connected layer at the end that requires a specific size of activations. And lastly the FCN introduces skip connections as a way of fusing information from different depth in the network for a multi-scale interference. Section 3.1.2 on page 36 provides a closer look into the architecture of the used FCN in this study.

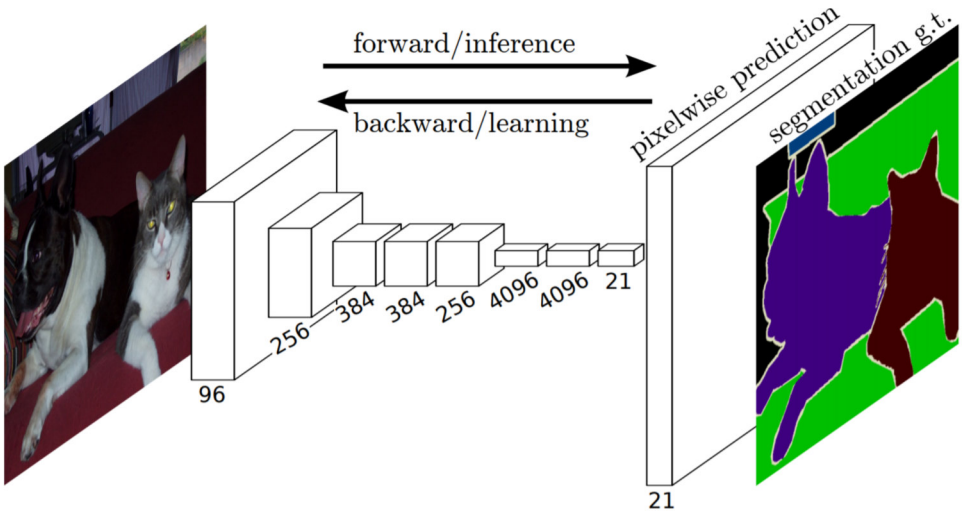


Figure 2.3: Architecture of the fully convolutional network from [Long et al., 2015]. This network introduced end to end learning for semantic segmentation, with the help of upsampling with deconvolutional layers.

### 2.2.2 Deep learning in remote sensing

Deep learning in remote sensing presents new challenges, since satellite image analysis raises some unique questions that have to be considered when using this new methods. A review on deep learning for remote sensing tasks from [Zhu et al., 2017] tackles some of these ambitious questions:

- Remote sensing data are geo-located and every pixel corresponds to a spatial coordinate. This opens up very interesting procedures of multi sensor data fusion when combining satellite imagery with geo-tagged images from social media. At the same time one has to make sure that during the training process of the image data the geo-location information is kept with the trained image due to random shuffling of the data tiles.
- Remote sensing data are geodetic measurements with controlled quality, which enable the user to retrieve geo-parameters with confidence estimates. However, different from purely data-driven approaches, the role of prior knowledge about the sensors adequacy and data quality becomes even more crucial. Even when using same ground sampling distances if the inclination angles are different the learning process can be more enduring.
- Remote sensing also faces the big data challenge. In the Copernicus era, very large and ever-growing data volumes are often available on a global scale. For example, even if they were launched in 2014, Sentinel satellites have already acquired about 25 Peta Bytes of data. Not only with the Copernicus program the amount of freely available satellite image data calls for global applications, as a consequence, algorithms must be fast enough and sufficiently transferable to be applied for the whole earth surface. On the other hand, these data are

well annotated and contain plenty of metadata. Hence, in some cases, large training data sets might be generated (semi-) automatically.

Scene classification of VHR satellite images, which aims to automatically assign a semantic label to each scene or pixel in an image, has been an active research topic in the field of high-resolution satellite images in the past decades. Generally, scene classification can be divided into two steps: feature extraction and classification. With the growing number of images, training a complicated non-linear classifier is time consuming. Hence, to extract a holistic and discriminative feature representation is the most significant part for scene classification. Traditional approaches are mostly based on the Bag-of-Visual-Words model [Sivic & Zisserman, 2003][Zhu et al., 2016], but their potential for improvement was limited by the ability of experts to design the feature extractor and the expressive power encoded. The deep learning architectures discussed in Section 2.2.1 have been applied to the problem of scene classification of high-resolution satellite images and led to state-of-the-art performance [Zou et al., 2015][Penatti et al., 2015][Castelluccio et al., 2015]. As deep learning is a multi-layer feature learning architecture, it can learn more abstract and discriminative semantic features as the depth grows and achieve far better classification performance compared with the mid-level approaches [Zhu et al., 2017].

Training deep learning-based methods can be done using three different methods. Using **pre-trained networks** trained on a image dataset, e.g., OverFeat [Sermanet et al., 2013], GoogLeNet [Szegedy et al., 2015] or ImageNet [Deng et al., 2009], have led to impressive results on scene classification of high-resolution satellite images [Hu et al., 2015][Zou et al., 2015]. Making a pre-trained model adapt to the specific conditions observed in a dataset under study, one can decide to **fine-tune** it on a smaller labeled dataset of satellite images. For example, [Nogueira et al., 2017] fine-tuned some high-level layers of the GoogLeNet [Szegedy et al., 2015]. This can help to further exploit the intrinsic characteristic of satellite images [Zhu et al., 2017]. **Training new networks** from scratch in addition to the above-mentioned ways to use deep learning methods for classifying satellite images. For example, [Castelluccio et al., 2015] train the networks by only using the existing satellite image dataset. This suffers a drop in classification accuracy compared with using the pre-trained networks as global feature extractors or fine-tuning the pre-trained networks.

# 3 Using deep learning to identify slums for an optimal water supply infrastructure

The supply of fresh water for the poor in areas of urban poverty is one of the main millennium development goals for modern civilization [UnitedNations, 2015b]. To provide an optimal water supply chain for inhabitants of informal settlements, a multidisciplinary approach is necessary. In this thesis a method is presented that combines a state of the art deep convolutional neural network to identify informal settlements in urban agglomeration areas and uses this information for designing a mathematical optimization to find optimal water supply structures. Since the result is very dependent on the input data a comparison between the original mapped slum data and the result from a DCNN segmentation is done. The general work-flow for supplying fresh water for slum dwellers using geodata from DCNN segmentation can be seen in figure 3.1. The method is split into two parts, first a DCNN is used to identify the informal settlements and in a second step the information about the slum's area and boundaries is used to calculate a water supply network optimized by its shortest path to provide each informal settlement with enough fresh water for its inhabitants.

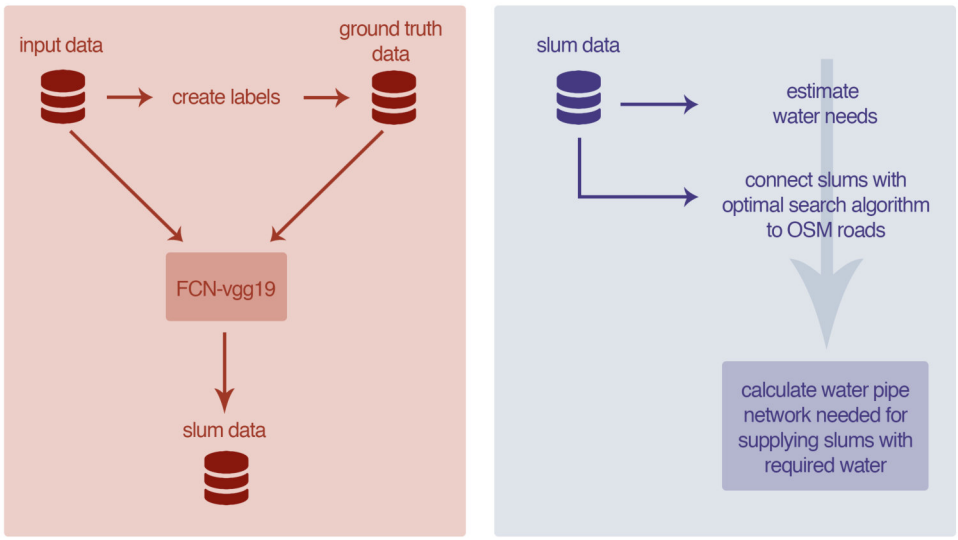


Figure 3.1: Proposed procedure for using deep learning to identify slums for an optimal water supply infrastructure.

## 3.1 Class segmentation of informal settlements

Semantic class segmentation means understanding an image at pixel level for specific groups, where each pixel in the image is assigned an object class. In this thesis multiple land use and land cover classes are used to classify an optical remote sensing image and extract detected informal settlements as polygons for a water supply chain optimization. The segmentation result contains land cover classes, which are considered the observed physical cover on the surface, and land use classes, which are characterized by the arrangements and inputs people undertake in a certain land cover type. The class segmentation contains vegetation, water and built-up land covers, where the built-up land use is split into formal building structures, informal buildings and lastly roads, railways and bare soil are combined into a ground land use class. Table 3.1 gives an overview on the used land use and land cover classes.

| Class | Label |
|---|---|
| 1 | Formal buildings |
| 2 | Ground |
| 3 | Vegetation |
| 4 | Water |
| 5 | Informal buildings / slums |

Table 3.1: Labels used for class segmentation of a DCNN.

### 3.1.1 Large scale ground truth data

Training a DCNN for class segmentation of remote sensing data requires a lot of reference data. Figure 3.2 illustrates the work-flow to create large scale high quality ground truth data used for training multiple DCNNss for slum mapping.

Quad-tree based segmentation: A quad-tree segmentation is used to assign the first four land use / land cover classes from table 3.1 to the resulting image objects. To produce the quad-tree image segmentation, the remote sensing VHR image is recursively split into quadrants and subquadrants until all pixels in a subquadrant meet the criterion of homogeneity (e.g., if all the pixels in the block are within a specific dynamic range [Finkel & Bentley, 1974]). Quad-tree segmentation performs very fast on large scale remote sensing data.

Decision tree based classification: The segmentation result uses domain specific features and a mutli-threshold procedure to create labels for a training dataset to perform a supervised classification. Feature construction is one of the key steps in data representation and largely conditioning the success of the following machine learning classification. For this purpose spectral, line and texture features are selected for data representation. Table 3.2 gives an overview on the applied features for a decision tree classification. A training dataset is created using a multi-threshold approach for land use / land cover classes formal buildings, ground, vegetation and water. A random forest (RF) classifier [Breiman, 2001] is applied to predict unlabelled image objects of the quad-tree dataset. The RF classifier creates 300 individual decision trees based on randomly picked samples from all training observations. During the classification phase an unlabelled observation is determined by the most frequent result of all trees.

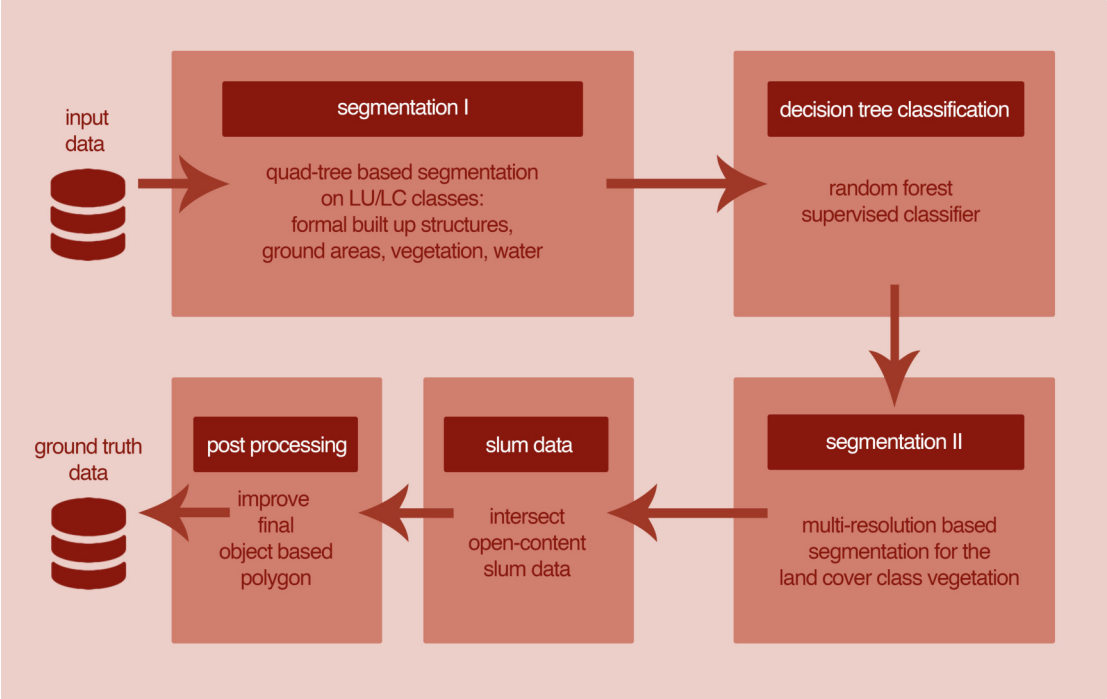


Figure 3.2: Work-flow to create a reference dataset for training DCNNs. The process is split into five parts, where the input dataset is used to create large cover ground truth data.

| Features | Type |
|---|---|
| Spectral | red, green, blue, near-infrared, ndvi |
| Line | canny edge detection, median filter on edges |
| Texture | contrast feature of a gray level co-occurrence matrix on all available spectral bands |

Table 3.2: Feature construction for a training dataset to classify with a decision tree.

Multi-resolution segmentation: Although the quad-tree segmentation performs very well on rectangular objects, the segmentation lacks precision on round object shapes. To improve the ground truth dataset on vegetation image objects a separate segmentation using a multi-resolution approach is used. The multi-resolution segmentation algorithm locally minimizes the average heterogeneity of image objects for a given resolution of image objects [Trimble, 2014]. This algorithm consecutively merges pixels based on a pairwise region merging technique. The resulting image objects present more precise boundaries between vegetation and the other land use / land cover classes.

Intersect slum reference data: The ground truth data for the land use class of informal settlements was acquired from Wikimapia [Koriakine & Saveliev, 2006], which utilizes an interactive web map with a geographically referenced wiki-system. Since official data about slums and its boundaries is very rare, Wikimapia provides a special category for slums where all districts containing informal settlements are tagged. If slums are clearly visible in the used remote sensing scene the boundary is vectorized on the basis of the used satellite data. Since Wikimapia is a privately owned open-content collaborative mapping project some inconsistencies are present and

depending on the date of the satellite image slums are either not yet present or have already transformed to formal settlements.

### 3.1.2 Class segmentation of informal settlements using deep convolutional neural networks

Deep convolutional neural networks are driving advances in recognition. DCNNs are not only improving whole-image classification [Krizhevsky et al., 2012][Szegedy et al., 2015], but are also making progress going from coarse to fine inference where a prediction is made at every pixel. In this section a fully convolutional network is introduced which is used to classify urban environments with the task to differentiate formal buildings from informal buildings.

#### Fully convolutional network FCN-vgg19

FCNs, first introduced by [Long et al., 2015] make it possible to train end-to-end and pixel-to-pixel on semantic segmentation for predicting dense outputs from arbitrary sized input images. Both learning, which is considered fitting the hyper parameters of the model for all examples and inference, which reflects the task of learning the values of the latent variables for a specific example are performed at a whole image by dense feedforward computation and backpropagation. Within the network upsampling layers enable a pixelwise prediction and learning with subsampled pooling.

The network in figure 3.3 uses the proven classification architecture VGG19 network from the *Visual Geometry Group* of Oxford university [Simonyan & Zisserman, 2014]. Throughout the whole CNN rather small $3x3$ receptive fields are used which are convolved with the input at every pixel. In this way a stack of two $3x3$ convolutional layers has an effective receptive field of $5x5$ (See illustration 3.4). And four such layers have a $9x9$ effective receptive field. This gives the advantage of incorporating four non-linear rectification layers instead of a single one, which makes the decision function more discriminative. Secondly, it decreases the number of parameters. $4(3^2C^2) = 36C^2$ produces less trainable weights than a single $9x9$ convolutional layer $9^2C^2 = 81C^2$.

To adapt the architecture from the vgg19 DCNN to a FCN some adaptions are required. The final classifier layer is discarded and replaced with a $1x1$ convolution and a channel dimension of the number of used classes. Afterwards deconvolutional layers are introduced to bilineary upsample coarse outputs to pixel dense outputs. In this case upsampling through deconvolutional layers means using backwards strided convolutions (transpose convolutions). This operation simply reverses the forward and backward passes of the convolution. Upsampling is performed for end-to-end learning by backpropagation from a pixelwise loss [Long et al., 2015].

As seen in figure 3.3 the FCN-vgg19 uses skips, which combines the final prediction layer with lower level layers with finer strides. Fusing fine layers and coarse layers lets the model make local predictions that respect global structure. The FCN fuses the output of the vgg19 network architecture with the predictions computed on top of the forth convolutional block (Layers 9-12) at stride 32 by adding a 2 times upsampling layer and summing both predictions (see figure 3.3 first transpose convolution). The upsampling is initialized by bilinear interpolation with learnable parameters. The next step is to fuse predictions from the third pooling layer with a two times upsampling of predictions fused from the forth pooling layer and 18th fully connected layer, building the complete FCN-vgg19 architecture.

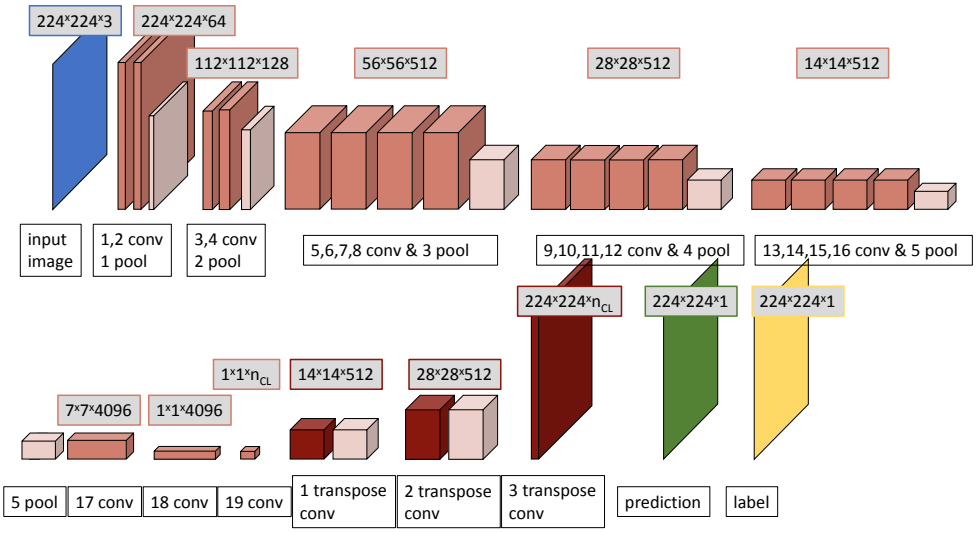


Figure 3.3: Architecture of FCN-vgg19 from [Long et al., 2015] with 19 convolutional (conv), 5 pooling (pool) and 3 transpose convolutional layers.

### Training methodology for the fully convolutional network

As presented in section 2.2.2 CNNs can be trained using three different methods. In this study learning from pre-trained networks and fine-tuning on different layers of the FCN are used in various ways.

- Pre-trained FCN: The FCN is trained on all layers with pre-trained Image-Net weights. ImageNet is an image dataset organized according to the WordNet hierarchy. As of 2016, over ten million images have been hand-annotated by ImageNet to indicate what objects are pictured. Training from scratch often does not improve classification results significantly [Nogueira et al., 2017]. Also it takes considerably more time to train a new dataset from scratch, which is why pre-trained weights for a vgg19 CNN trained on the ImageNet dataset are used. All layers are trainable by backwards propagation through the whole network. The FCN is trained for 100 epochs, where one epoch is considered a complete pass through of the given dataset. These results will work as a reference for the other training methods. In Addition a combined dataset from both cites will be used to test, if the FCN can generalize to multiple cities [Maggiori et al., 2017].

- Fine-tuned FCN: The FCN is fine-tuned from one cities dataset to the other and vice versa from two different points in the FCN. In addition the pre-trained FCN of the combined dataset of multiple cities is used to fine-tune the network for each city. The FCNs are initialized from the previous best performing pre-trained networks. From the third and fourth convolutional block in figure 3.3 layers are fully trainable while the other convolutional blocks are frozen for learning with the pre-existing weights as seen in table 3.3.

- Enforced fine-tuned FCN: Since an imbalanced class distribution is expected for slum mapping approaches, where informal settlements only cover a fraction of the complete samples, an enforced dataset is created and fine-tuned based on the results of the pre-trained FCN. The enforced dataset only uses image tiles containing labels for informal settlements.

Table 3.3 gives an overview on the training methodology for the FCN. The naming procedure used for training the pre-trained networks is labeled as $FCN\_dataset_{xyz}\_n_E$, *dataset* being the

image data for each city, $xyz$ the image data channels used for feeding input data through the network and $n_E$ being the number of epochs used for training. When using fine-tuning approaches the nomenclature is changed to $FTd\_dataset_{xyz}\_n_E\_Ln$, where $FTd$ stands for fine-tuned on dataset for Mumbai $M$, Delhi $D$ and $MD$ for the combined dataset ($dataset_E$: enforced learning on the dataset) and $Ln$ indicates from which convolutional layer block learning was enabled. When the pre-trained Mumbai FCN is fine-tuned to the Delhi dataset for 100 epochs with learnable layers from the fifth FCN block the resulting data would be labeled as $FTM\_Delhi\_100\_L5$.

| | FCN | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| FCN block | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Layer | conv | conv | conv | conv | conv | fc | fc | conv' | conv' | conv' |
| $FCN\_dataset\_n_E$ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| $FTx\_dataset\_n_E\_L4$ | x | x | x | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| $FTx\_dataset\_n_E\_L5$ | x | x | x | x | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| $FTx\_dataset_E\_n_E\_L4$ | x | x | x | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| $FTx\_dataset_E\_n_E\_L5$ | x | x | x | x | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

Table 3.3: Training methodology for the fully convolutional network. The checkmark indicates learnable weights, while the cross freezes the FCN block for learning. *conv* represents convolutional layers, *fc* fully connected layers and *conv'* transpose convolutional layers.

**Experimental framework**

Deep learning framework: The implementation of the FCN is based on the TensorFlow framework [Shekkizhar, 2017]. TensorFlow is an open-source software library for machine intelligence and commonly used for deep learning. The FCN is trained on a single Nvidia Titan-X GPU with 12gb ram.

Data augmentation: Deep networks need a large amount of training data to achieve good performance. To build a powerful image classifier for remote sensing, where often limited training data is available, image augmentation is usually required to boost the performance of deep networks. Image augmentation artificially increases training images through splitting the data with an overlap of 28 pixels into multiple images tiles. The input data is split into image tiles of $224x224$ pixels and with a radiometric resolution of 8bit unsigned values and three image channels.

Receptive fields: The receptive field is defined as the region in the input space that the FCNs feature is affected by. The FCN uses $3x3$ receptive fields throughout the whole network as seen in illustration 3.4. A receptive field of a feature can be fully described by its center location and its size. The number of output features in each dimension can be calculated using equation 3.1 [Dumoulin & Visin, 2016]. The number of features learned by the FCN are approximately 140 million parameters.

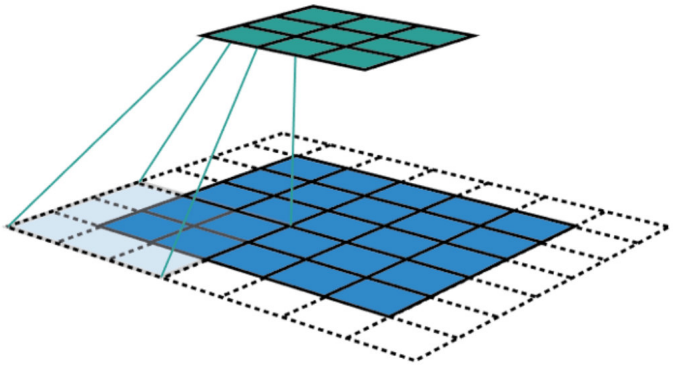

Figure 3.4: This figure shows a receptive field, applying a convolution *conv* with kernel size $k = 3x3$, padding size $p = 1x1$ to extend the original image boundaries, stride $s = 1x1$ on an input map $5x5$, an output feature map $3x3$ (green map) is created [Dang, 2017].

$$
\begin{aligned}
n_{out} &= (\frac{n_{in} + 2p - k}{s}) + 1 \\
n_{out} &: \text{Number of output features} \\
n_{in} &: \text{Number of input features} \\
k &: \text{Convolution kernel size} \\
p &: \text{Convolution padding size} \\
s &: \text{Convolution stride size}
\end{aligned}
\tag{3.1}
$$

Loss function: The FCN uses cross entropy for its loss function. Cross entropy loss, or log loss, measures the performance of a classification model whose output is a probability value between $0-1$. The output from a neuron is $a = \sigma(z)$, where $z$ is the weighted sum of the inputs in equation 3.2. The cross-entropy cost function for this neuron is defined by equation 3.3. This cost function is always non-negative. All individual terms in the sum in 3.3 are negative, since both logarithms are of numbers ranging from $0 - 1$ and there is a minus sign in front of the sum. Also if the neuron's actual output is close to the desired output for all training inputs $x$ the cross-entropy will be close to zero. Summing up, the cross-entropy is positive and tends toward zero as the neuron gets better at computing the desired output $y$ for all training inputs $x$. Another benefit of using cross entropy as a loss function is, that it avoids the problem of learning slowing down [Nielson, 2015].

$$
\begin{aligned}
z &= \sum_j w_j x_j + b \\
z &: \text{weighted sum of the inputs}
\end{aligned}
\tag{3.2}
$$

$$
\begin{aligned}
C &= -\frac{1}{n}\sum_{x} ylna + (1-y)ln(1-a) \\
C &: \text{Cross entropy loss for neuron } j \\
n &: \text{Total number of items in training data} \\
x &: \text{Training input} \\
y &: \text{Corresponding desired outputs}
\end{aligned}
\tag{3.3}
$$

Optimization: Gradient descent algorithms are one of the most popular options to perform optimization and by far the most common way to optimize neural networks. Gradient descent is a way to minimize an objective function $J(\theta)$ parameterized by a model's parameters by updating the parameters in the opposite direction of the gradient of the objective function $\nabla J(\theta)$ with respect to the parameters $\theta$. The learning rate $\eta$ determines the size of the steps taken to reach a (local) minimum. To optimize in the right direction the cross entropy loss is minimized with the Adaptive Moment Estimation Optimizer (Adam) [Kingma & Adam, 2017]. This gradient descent algorithm computes adaptive learning rates and keeps an exponentially decaying average of past gradients and squared gradients.

$$
\begin{aligned}
\theta_{t+1} &= \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + 10^{-8}}\hat{m}_t \\
\theta &: \text{Parameter} \\
\hat{v}_t &: \text{Exponentially decaying average of past squared gradients} \\
\hat{m}_t &: \text{Exponentially decaying average of past gradients} \\
\eta &: \text{learning rate}
\end{aligned}
\tag{3.4}
$$

k-fold cross validation: 4-fold cross-validation is used for training and validation of the data. The dataset is randomly partitioned into four equal sized subsamples. Of the four subsamples, a single subsample is retained as the validation data for testing the FCN, and the remaining three subsamples are used as training data. The cross-validation process is then repeated four times (the folds), with each of the four subsamples used exactly once as the validation data. The four results of the folds are averaged to produce a single estimation. The advantage of this method over repeated random sub-sampling is that all observations are used for both training and validation, and each observation is used for validation exactly once.

## 3.2 Optimal water supply infrastructure

Informal settlements are among others characterized by a lack of access to water, sanitation or electricity. To ensure sustainable development in informal settlements and providing water for the poor a multidisciplinary approach is necessary. In general, supplying the slum dwellers with fresh water is a complex problem that different studies try to solve by finding an optimal water supply strategy [Snyder et al., 2014][Subbaraman et al., 2014]. For this study an optimal water infrastructure with pipes of different diameters is modeled. Since an optimal water supply network for slums has not been modelled along the street network in a recent study by [Friesen et al., 2017], in this thesis a procedure to calculate the cost of investing in a completely new water pipe network to provide each slum with fresh water is presented. The pipeline networks are optimized towards a shortest possible path using multiple network approaches.

### 3.2.1 Optimal water pipe network for informal settlements

Slum systems are determined by the geographical position and its area extracted by the FCN. Based on the size of a slum and an average population density the expected daily need of water is calculated for each slum. Running the optimization solver leads to a network showing the chosen connections between the slums. Each slum is connected to the water works station by a path of all slums. In this thesis three approaches for the general water supply networks are proposed to find an optimal solution for each method, illustration 3.5 shows the basic procedure of building these pipe connections.

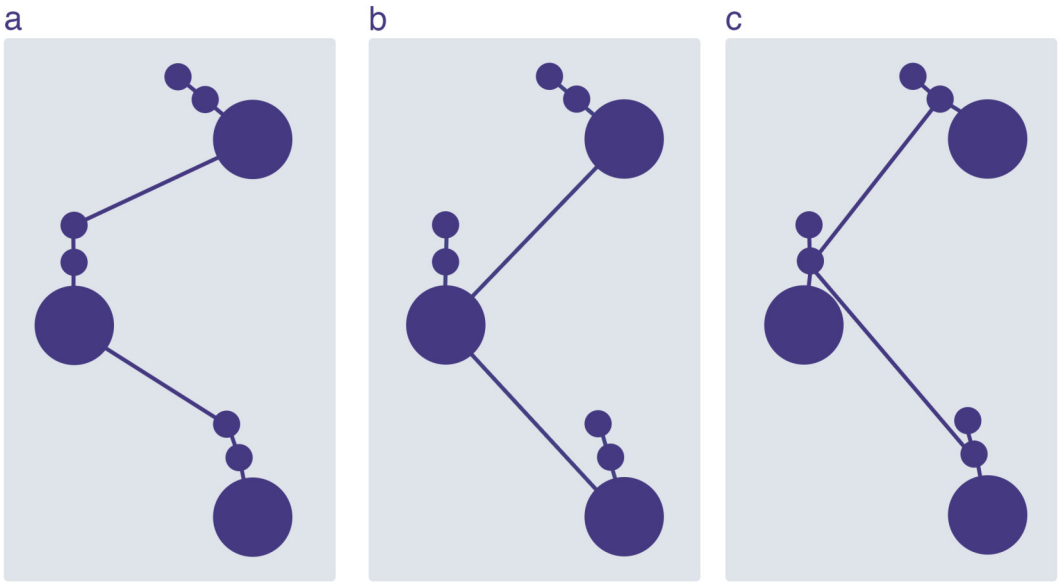


Figure 3.5: Water pipe network solutions. (a) All slums are connected via a shortest path, (b) hierarchical approach with clusters connected through the biggest slums, (c) hierarchical approach with clusters connected through the center slums

- Shortest path connecting all slums: For this network in figure 3.5(a) all informal settlements are connected optimized for the shortest possible path. There is no sorted order to connect slum areas by any criteria whatsoever.
- Geospatial cluster & large slums form separate network: All slums are clustered according to the geographical location and a hierarchical system is built as seen in illustration 3.5(b). The largest slums form an underlying network connected by large roads and a second network using all roads to connect all slums in their cluster. The general idea is to use pipes with a large diameter for the underlying network and smaller pipes for the cluster networks to lower the investment cost.
- Geospatial cluster & center slums form separate network: Instead of using the largest slums of each cluster, this approach uses the centre of each cluster to form the underlying network of the hierarchical solution in figure 3.5(c). These are connected via the road network using only large roads and the cluster networks use all roads for the pipe network optimization.

For each network solution of figure 3.5 an origin - destination (OD) matrix is calculated. A network-based spatial analysis algorithm solves this complex routing problem calculating all pos-

sible lengths along the road infrastructure for all informal settlements. The OD cost matrix finds and measures the least-cost paths along the network from multiple origins to multiple destinations. The results of the OD cost matrix analyses become the input for a spatial analyses where the network cost is measured by the length along the street infrastructure.

Kruskal's algorithm is used to find the optimal path connecting all informal settlements selected in the OD cost matrix [Kruskal, 1956]. This procedure is a minimum-spanning-tree algorithm which finds an edge of the least possible weight of all connections. It is a greedy algorithm in graph theory as it finds a minimum spanning tree for a connected weighted graph adding increasing cost arcs at each step. This means it finds a subset of the edges forming a tree that includes every vertex, where the total weight of all the edges in the tree is minimized. If the graph is not connected, it finds a minimum spanning forest. The algorithm creates a graph, where each vertex in the graph is a separate tree. This can be seen in the six nodes in figure 3.6. Then a set containing all the edges in the graph is created. Nine connections in the OD matrix of figure 3.6 form the weights of the graph. If the edge connects two different trees with minimum weight it is added to the forest, combining two trees into a single tree.

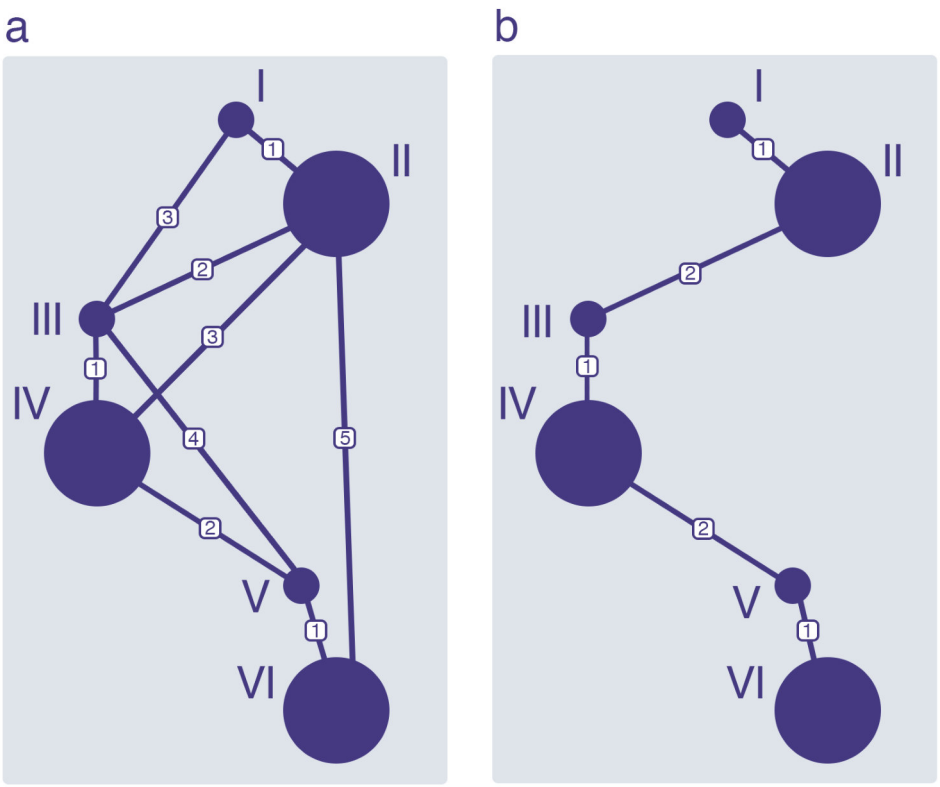


Figure 3.6: (a) Origin-Destination Matrix showing a cost variable in length of the road networks between informal settlements. (b) illustrates the result of Kruskal's algorithm finding the shortest path to connect all informal settlements with each other.

### 3.2.2 Cost functions for network structure

Guidelines of the World Health Organization (WHO) for drinking water quality, provide guidance on good practices for ensuring that water distribution is adequate for human health. Every person needs a minimum of 20 liters of water per day to meet the minimum basic requirements, although

this amount may still lead to health concerns. Governments and authorities should therefore aim to guarantee at least 50 to 100 liters of water per person per day. [Sule, 2003] goes even further and reports an average usage of 135 liters per consumer per day for the city of Mumbai. This information is used to calculate the investment cost of supplying each slum dweller with enough fresh water. To find an economical optimal infrastructure, it is necessary to define cost functions for the different options. Different types of pipes are assumed that can be chosen using a diameter of $60mm$ to $600mm$. The cost functions for these pipes are taken from [Marchionni et al., 2015] who developed cost functions for pipes in Portugal. These cost function were adapted to informal settlements from [Friesen et al., 2017]. The costs contain variable $C_{var}(d,l)$(operating time depended per day) and fix costs $C_{fix}(d,l)$. To model the variable costs of a pipe network the common dissipation model for pipeflows is used, calculating the dissipated power with the following equation, assuming turbulent flow [Spurk & Aksel, 2004]. $Q$ is the volume flow, $\Delta p$ the pressure loss in result of dissipation, $d$ the diameter and $l$ the length of the pipe.

$$C(d,l) = C_{var}(d,l) + C_{fix}(d,l) \tag{3.5}$$

$$\begin{aligned} C_{fix}(d,l) &= (32.59 + 0.11d + 0.00053d^2)l \\ C_{fix}(d,l) &: \text{Fixed costs } € \\ d &: \text{Pipe diameter [m]} \\ l &: \text{Pipe length [m]} \end{aligned} \tag{3.6}$$

$$\begin{aligned} C_{var}(d,l) &= \frac{1}{\eta} C_{kWh} P_{Diss}(d,l) t \\ P_{Diss} &= Q\Delta p \\ C_{var}(d,l) &: \text{Variable costs } € \\ C_{kWh} &: \text{Energy price [€/kWh]} \\ P_{Diss} &: \text{Dissipated Power [watt]} \\ Q &: \text{Volume flow [l/s]} \\ \Delta p &: \text{Pressure loss [m]} \\ d &: \text{Pipe diameter [m]} \\ l &: \text{Pipe length [m]} \end{aligned} \tag{3.7}$$

# 4 Experiments

To test the proposed method from section 3 several experiments were performed. Their aim was to test the DCNNs robustness to detect informal settlements not only for case study applications but rather to transfer the learned process from the DCNN to two mega cities. The general idea is to train on very high resolution optical data with a large scale ground truth data for the class segmentation of slums and afterwards using transfer learning strategies to identify informal settlements in other cites of the same country. Using mega cities in the same country assures that the DCNN only has to retrain its weights in order to detect slums and not a complete different architectural type of city structures. The optical very high resolution satellite data contains multiple pansharpened QuickBird scenes from Mumbai and Delhi in India, which are introduced in section 4.1. The ground truth data is created using the method from section 3.1.1 and described in section 4.1.1. Once the ground truth data is validated the DCNN and its training process is presented in 4.2. The FCN-vgg19 is trained on small images tiles for a Mumbai-, Delhi- and a combined dataset of the two mega-cities. Transfer learning is introduced to the FCN in order to test the FCNs possibility of learning geo-spatial structures between the Mumbai and Delhi dataset. Once the informal settlements are classified the method to provide a water supply network from section 3.2 is used to test the effects of different input geodata from the extracted slums on the water pipe network and its changes in financial investment. These results are shown in section 4.4.

## 4.1 Dataset

### 4.1.1 Satellite dataset

For the two Indian mega cities Mumbai and Delhi, pan-sharpened QuickBird Scenes were acquired. All available QuickBird imagery products are 4-band pan-sharpened images and combine the visual information of four multispectral bands (blue ($450-520$nm), green ($520-600$nm), red ($630-690$nm), near infra-red ($760-890$nm)) with 2.4m ground sampling distance (GSD) and the spatial information of the panchromatic band of 60cm GSD. The QuickBird scene for Mumbai was acquired on 17.11.2008 covering an area of roughly $103km^2$. For the study area in Delhi a scene from 26.04.2007 was acquired. Seven areas of interest are selected due to frequent appearance of informal settlements with a combined area of $96km^2$. Figure 4.1 presents both datasets used for training the FCN.

### 4.1.2 Ground truth dataset

Deep learning methods are very dependent on good training data. To ensure that the network learns in a correct manner the labels in the reference dataset need to be of high quality. In this thesis the main goal is to identify and differentiate slums from formal city structures in an urban environment, but since there are more land use / land cover classes to be found in this habitat the FCN is trained for multiple classes. The labels used for training the network should represent the

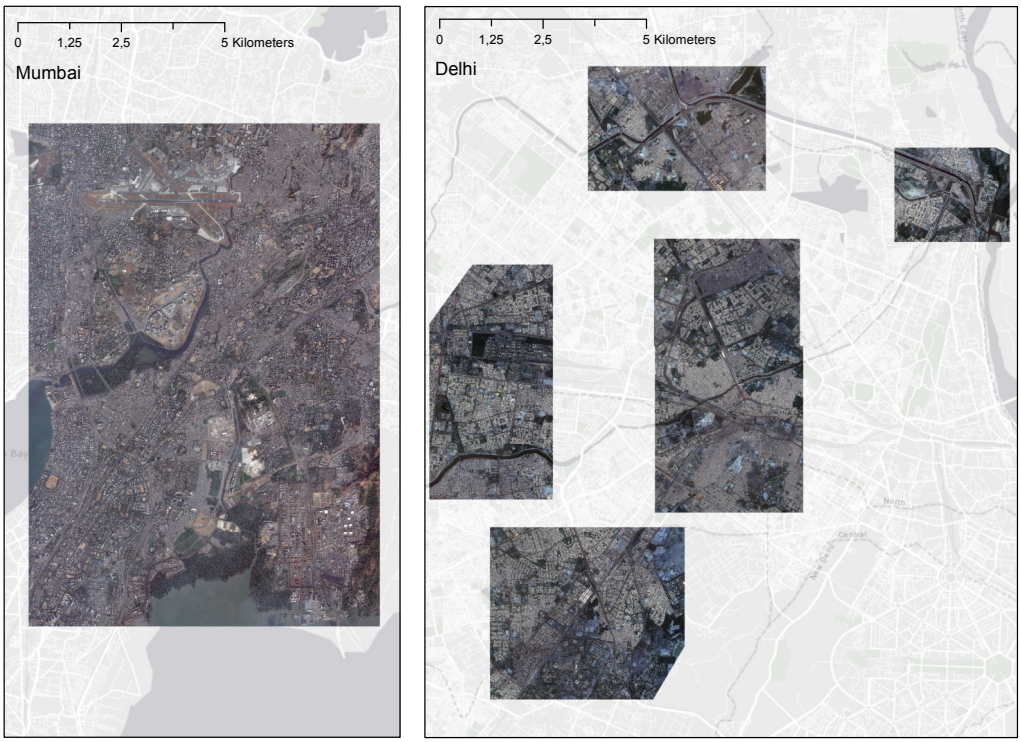


Figure 4.1: Pansharpened true colour composite QuickBird scenes for the study areas in Mumbai and Delhi.

common structure of a global mega-city with a focus on differentiating slums areas from the rest of the structures as good as possible. The five land use / land cover classes introduced in table 3.1 of section 3.1 were used for the class-segmentation. The ground truth dataset was created using the method presented in figure 3.2. Illustration 4.2 to 4.4 gives an overview of the work-flow to create large scale ground truth data for an area of interest in Delhi containing all land use / land cover classes but water. Validation of the reference dataset is presented in section 4.1.2.

**Ground truth data for the study areas.**

The experimental work-flow for the ground truth data is designed to elevate accuracy of previous reference data. For a pixel-to-pixel based class-segmentation using a FCN a processed image patch should contain no background or missing entities. To achieve highest possible segmentation accuracy for each image dataset a ground truth dataset was created. Figure 4.2 to 4.4 illustrates the development of each step.

In figure 4.2 a quad-tree based image segmentation process is used to split the remote sensing VHR image (Segment (a)) recursively into quadrants and subquadrants until all the pixels in a subquadrant meet the criterion of homogeneity of all four available channels (blue, green, red, near infra-red) in image (b). For every object, domain specific image features are constructed. (c) displays the Normalized Difference Vegetation Index (NDVI), high values represent reputable vegetation, whereas dark/low values indicate no flora. (d) represents a median ($25x25$pixel) filter of a canny edge detector for the blue channel. The canny edge detector is known as an optimal detector, where the algorithm aims to satisfy three main criteria of a low error rate, meaning a good detection of only existent edges, good localization, represented by the distance between edge pixels detected and a minimal response, where only one detector response per edge is accepted.

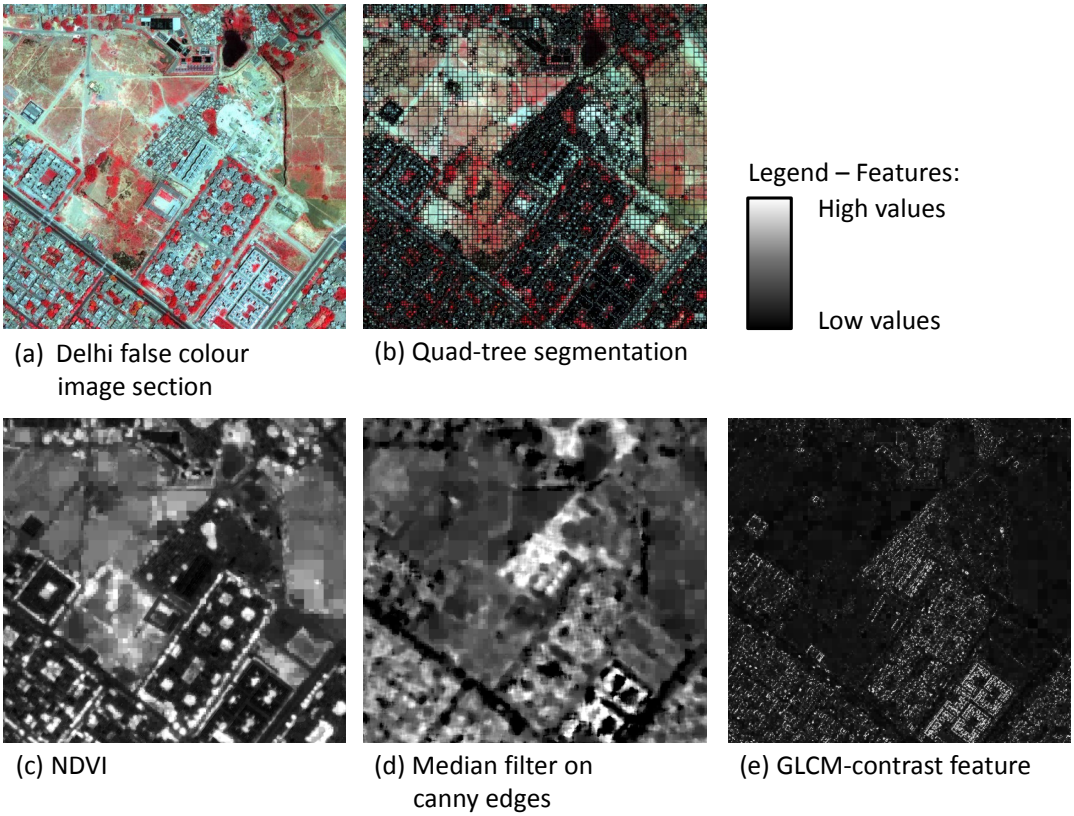


Figure 4.2: Ground truth segmentation process and feature selection. (a) shows an area of interest in Delhi in a false color image of the near infrared, red and green band. (b) presents the result from the Quad-tree segmentation. In (c), (d) and (e) features for training the random forest are shown.

The result is smoothed with a median filter since it preserves edges while filtering the image [Arias-Castro et al., 2009]. Edges are of critical importance for differentiating built-up structures from other entities. This is why edge features need to be optimal. High values represent multiple detected edges in one image objects whereas dark spots are considered as object without edges. Lastly for every image channel the GLCM contrast features were constructed ((e) illustrates the feature for the blue channel). GLCMs are filter functions which provide a statistical view of texture based on the image histogram [Haralick et al., 1973]. Several statistics provide information about the texture of an image. In this case the contrast feature is used to measure the local variations in the gray-level co-occurrence matrix to separate built-up structures from other entities. Bright spots represent a high index for the GLCM contrast feature in one image objects while dark spots are considered as objects of low local variations.

$$
\begin{aligned}
Label_{building} &= NDVI < x \;\&\; GLCM_{contrast} > y \;\&\; edge_{canny} > z \\
Label_{ground} &= NDVI > x \;\&\; NDVI < y \;\&\; edge_{canny} < z \\
NDVI &: \text{Normalized difference vegetation index} \\
GLCM_{contrast} &: \text{GLCM contrast haralick feature} \\
edge_{canny} &: \text{Canny edge detection} \\
x, y, z &: \text{Feature dependent threshold}
\end{aligned}
\tag{4.1}
$$

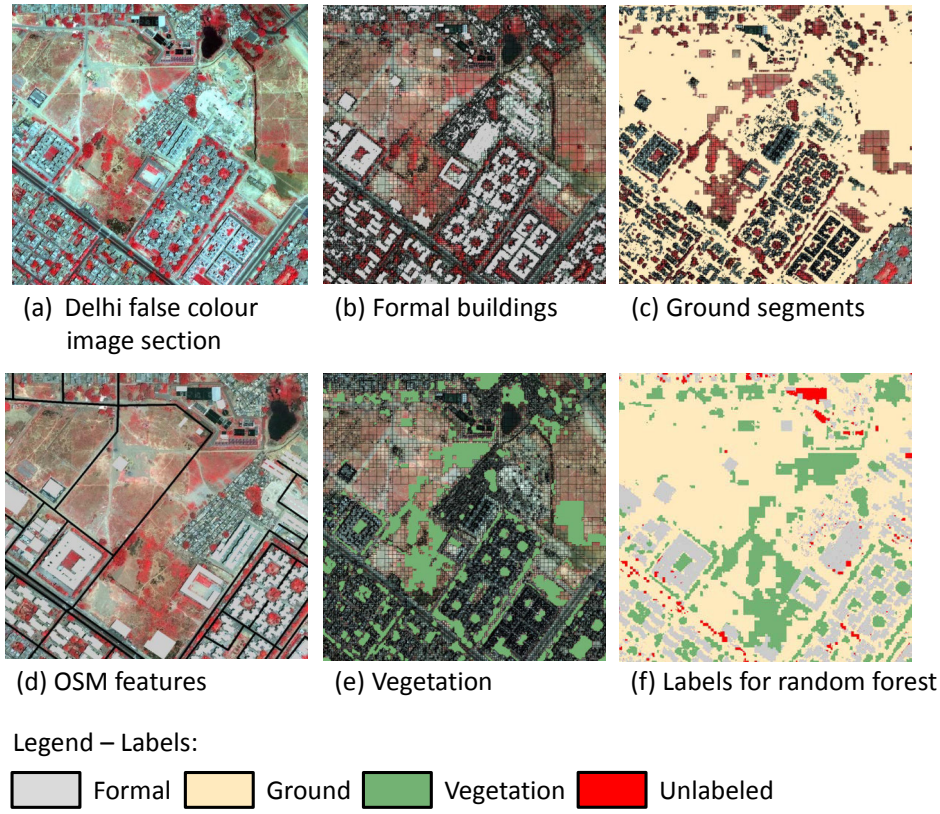


Figure 4.3: Ground truth segmentation process and production of labels for a supervised classification. (a) presents an area of interest in Delhi in a false colour image of the near infrared, red and green band. (b) Triple threshold using edge-, NDVI- and gray level co-occurence matrix (GLCM) features. (c) Triple threshold using two NDVI thresholds and one edge feature condition. (d) Intersection with open street map data. (e) Labels for vegetation based on the NDVI and lastly (f) illustrates the dataset used for training a random forest decision tree classifier.

Figure 4.3 illustrates how the domain specific features introduced above are utilized to create labels for a training dataset to perform a supervised classification. Since feature construction is one of the key steps in data representation and largely conditioning the success of the following machine learning classification, labels for a decision tree classification need to be of high quality. In (b) a the triple threshold function from the first equation in 4.1 is used to label formal urban structures. A similar threshold operation in the second equation of 4.1 is used to label all ground segments (bare soil, streets and other non flora ground segments) in (c). To improve the robustness of labels in (b) and (c) Open Street Map data is used to label buildings and selected roads. A simple NDVI threshold is used for labelling vegetation in (e). (b)-(e) are used to train a decision tree classifier in figure 4.4. Red labels in (f) represent image objects to be predicted by the random forest classifier.

A device for classifying unlabeled observations from a feature space is the random forest classifier [Breiman, 2001], which is based on decision trees. A Random Forest creates a collection of individual decision trees based on randomly picked samples from all training observations trees. In the training phase 300 classification trees are built. In the classification phase, an unlabeled observation is classified with all trees. The class which is ultimately assigned to the observation is determined by the most frequent result of all trees. In figure 4.4 the result can be seen in (b). All previous unlabeled image objects are assigned to a prediction. The current segmentation process only contains four land cover / land use classes. Since informal settlements offer great

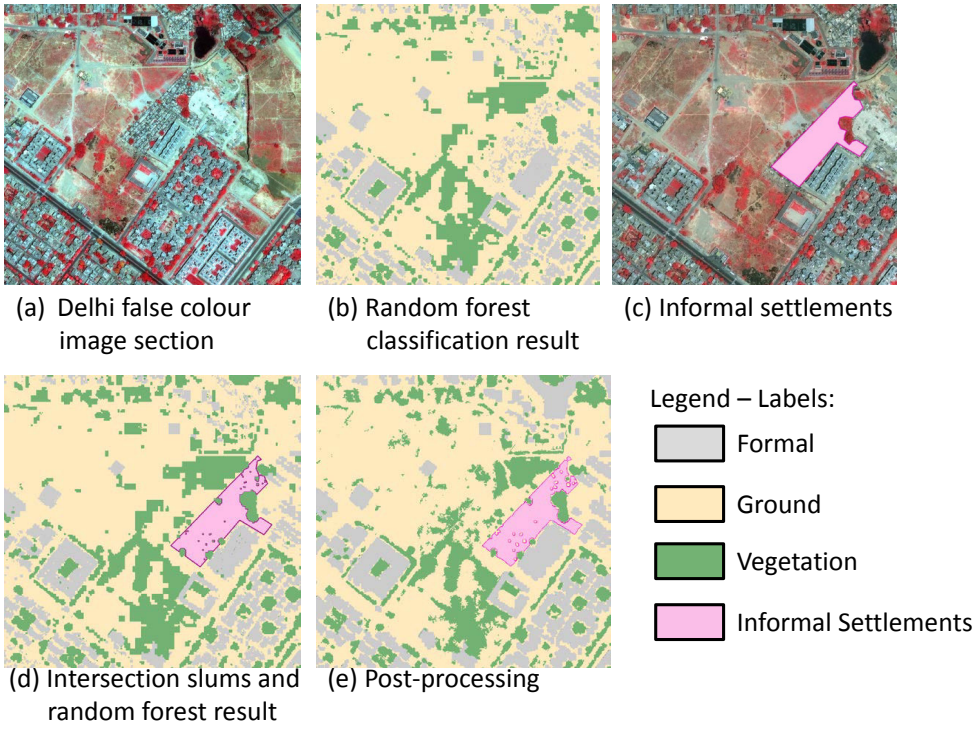


Figure 4.4: Ground truth segmentation process, prediction from a random forest classifier and the final ground truth dataset after post processing. (a) presents an area of interest in Delhi in a false color image of the near infrared, red and green band. (b) Prediction from a random forest classifier. (c) Polygon data for informal settlements. (d) Intersection of slums and the random forest result (e) Final ground truth dataset used for training the DCNN.

heterogeneity it is very compute intensive and feature dependent to successfully detect slums with regular machine learning algorithms like random forest or linear discriminant analysis [Wurm et al., 2017]. For an accurate representation of informal settlements in the ground truth dataset open access geo-data is intersected with the predicted result from the random forest classifier as illustrated in (c) and (d). Wikimapia [Koriakine & Saveliev, 2006], which utilizes an interactive web map with a geographically referenced wiki system, is used for the slum geo-data, since official datasets about slums and its boundaries are very rare and can suffer from inconsistencies. Wikimapia provides a special category for districts containing slums, if slums are clearly visible in the QuickBird scene of Mumbai and Delhi the boundary is polygonized. Since wikimapia is a privately owned open-content collaborative mapping project some inconsistencies are present. Additionally depending on the date of the satellite image slums are either not yet present or already advanced into formal settlements. The outcome of the random forest classification and its intersection geo-data of informal settlements is still prone to mediocre results in vegetation and in general objects that represent round structures, since the quad-tree based segmentation can only approximate round features through smaller quadrants. In a post-processing mechanism a multi-resolution based segmentation algorithm replaces vegetation features and improves a more natural shape of greenery entities. The final result is rasterized and co-registered to the exact same extent and pixel size of the VHR image data. The ground truth dataset shows an imbalance of land use / land cover classes as seen in table 4.1. In Mumbai significantly more geo-data of informal settlements was available. The difference of formal built-up structures and informal

settlements is represented by a percentage increase of 888%, whereas in Delhi the increase in the housing situation makes up a immense percentage increase of 1743%. This difference may be the result of inconsistencies in the wikimapia reference. In total slums in Mumbai make up 10% of the complete ground truth dataset, while in Delhi they only make up roughly 2%.

| Class | Mumbai | Delhi |
|---|---|---|
| slum | 10.12% | 1.80% |
| formal | 25.44% | 33.29% |
| ground | 22.94% | 33.62% |
| vegetation | 35.93% | 29.65% |
| water | 5.57% | 1.63% |
| Sum | 286 074 829 [pixel] | 220 499 966 [pixel] |

Table 4.1: Class imbalance for the ground truth dataset of Mumbai and Delhi. Mumbai contains about 65 million more total pixels and is with an area of $103km^2$ 23% larger than the Delhi AOI with $79km^2$.

**Accuracy assessment for the ground truth data**

To validate the reference dataset used for training the network an appropriate sample unit was selected. To get a good representation of the dataset when using polygons as reference data, equation 4.2 should be used to get the total number of samples according to [Congalton & Green, 2008]. Since five land cover / land uses classes were used, 817 samples are round up to 1000, providing 200 samples per class needed to validate the dataset. For $\Pi_i$ a confidence level of 95% and four $(classes - 1)$ degrees of freedom were used. $B$ is calculated with equation 4.3.

$$
\begin{aligned}
n &= \frac{B\Pi_i(1-\Pi_i)}{b_i^2} = \frac{0.3593 * 0.35 * (1-0.35)}{0.01^2} = 817 \\
\text{where } n &: \text{Number of total samples} \\
B &: \chi^2_{4,0.99} \\
\Pi_i &: \text{Area of biggest class in \%} \\
b_i^2 &: \text{Margin of error}
\end{aligned}
\tag{4.2}
$$

$$
\begin{aligned}
\chi^2_{k-1,index} &= 1 - \frac{\alpha}{k} = 1 - \frac{1-0.95}{5} = 0.99 \\
\text{where } \chi^2_{k-1,index} &: \text{Value for index} \\
\alpha &: \text{1-confidence value} \\
k &: \text{Number of classes}
\end{aligned}
\tag{4.3}
$$

Table 4.2 and 4.3 show the confusion matrix for the reference dataset for Mumbai and Delhi using 200 samples per class. These samples were chosen randomly throughout the whole dataset. The confusion matrix for the ground truth dataset for Mumbai and Delhi show an overall accuracy of 87% and a kappa value of 84% for both Datasets. Although precision scores for the land use class slums score with over 90% very respectable, the recall values with over 83% give an indicator that maybe not all available informal settlements are present in the ground truth dataset. Another

interesting insight can be seen in a significant overlap of the land use classes formal buildings and ground segments. Both classes score lowest throughout the confusion matrix. Especially in Mumbai the ground truth data reveals that repeatedly ground segments are present in formal labeled objects.

| Classification | Formal | Ground | Vegetation | Water | Slum | Sum | Precision |
|---|---|---|---|---|---|---|---|
| Formal | 146 | 30 | 10 | 0 | 14 | 200 | 73% |
| Ground | 8 | 172 | 14 | 0 | 6 | 200 | 86% |
| Vegetation | 4 | 14 | 178 | 0 | 4 | 200 | 89% |
| Water | 0 | 2 | 0 | 198 | 0 | 200 | 99% |
| Slum | 4 | 10 | 0 | 2 | 184 | 200 | 92% |
| Sum | 162 | 228 | 202 | 200 | 208 | 1000 | |
| Recall | 90% | 75% | 88% | 99% | 88% | | |

Table 4.2: Confusion matrix for the accuracy assessment of the Mumbai ground truth dataset.

| Classification | Formal | Ground | Vegetation | Water | Slum | Sum | Precision |
|---|---|---|---|---|---|---|---|
| Formal | 166 | 16 | 0 | 0 | 18 | 200 | 83% |
| Ground | 12 | 162 | 10 | 0 | 16 | 200 | 81% |
| Vegetation | 0 | 10 | 188 | 2 | 0 | 200 | 94% |
| Water | 0 | 22 | 0 | 176 | 2 | 200 | 88% |
| Slum | 6 | 6 | 6 | 0 | 182 | 200 | 91% |
| Sum | 184 | 216 | 204 | 178 | 218 | 1000 | |
| Recall | 90% | 75% | 92% | 98% | 83% | | |

Table 4.3: Confusion matrix for the accuracy assessment of the Delhi ground truth dataset.

## 4.2 Training the fully convolutional neural network FCN-vgg19 for slum mapping

The production of training data is often an expensive and laborious task. In this study large scale ground truth data was available by virtue of the work-flow from the previous section. Using data augmentation image tiles with an overlap are created to increase the dataset used for training the DCNN. The aim of this section is to demonstrate the training procedure for each approach presented in section 3.1.2 and how to work with an imbalanced dataset.

| Dataset | training image tiles | validation image tile |
|---|---|---|
| *Mumbai* | 5616 | 1872 |
| *Delhi* | 4370 | 1457 |
| *Mumbai & Delhi* | 9986 | 3329 |

Table 4.4: Dataset for training and validation and number of annotation tiles containing informal settlements.

The dataset is trained on one single Nvidia Titan-X Gpu with 12 Gb of GDDR5 Ram. Although this GPU delivers a very high compute capability training the FCN-vgg19 is very demand-

ing. In this case only two 8bit images of $224x224$ pixels can be trained simultaneously. These two images define the batch-size of the FCN. Both datasets are converted from a 16bit four channel scene into two datasets of 8bit three channel image-tiles with an overlap of 28pixels (1/8 of the tile). The advantage of using an overlap is that it delivers an increased number of total image-tiles for training. The two dataset consist of a true colour (channels red, green, blue) and a false colour composite (channels nir, red, green) image stack. The dataset for Mumbai contains in total 5615 training images and Delhi consists of 4370 training images as seen in table 4.4. Training the FCN for one complete pass through the dataset is called one epoch. For Mumbai one epoch takes $5616/batchsize = 2808$ iterations, while for Delhi it only takes $4370/batchsize = 2185$ iterations. Both datasets are trained on all available layers of the network for 100 epochs.

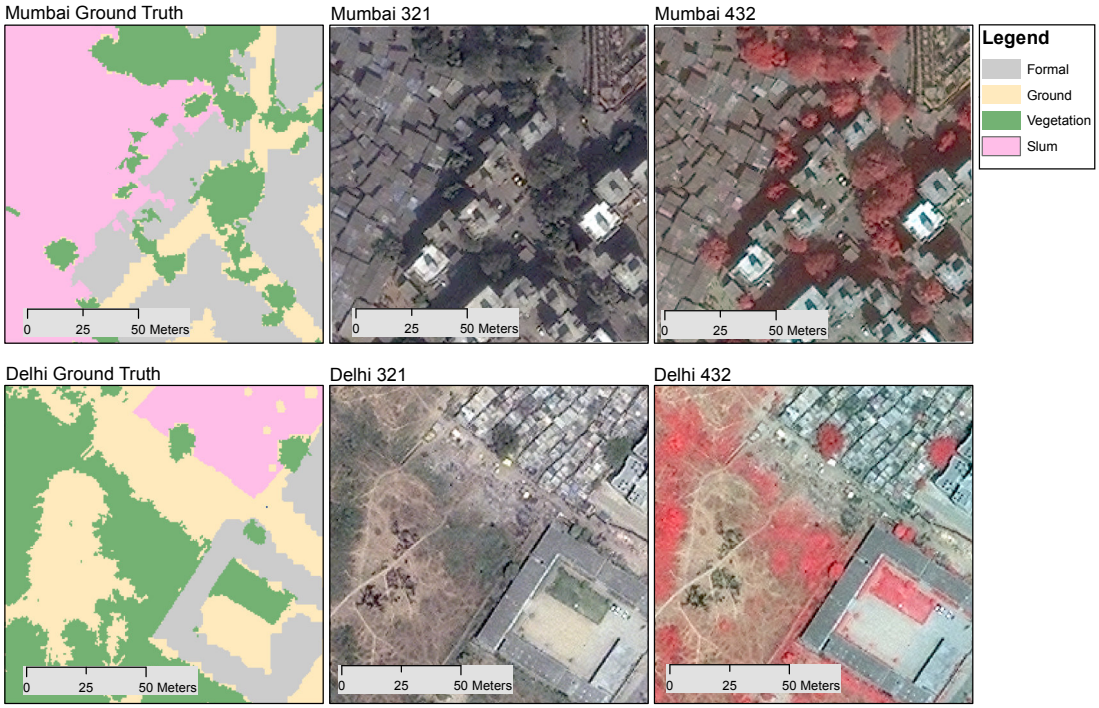


Figure 4.5: Datasets used for training the FCN-vgg19. The first row represents images from the Mumbai dataset, while the bottom row shows images from Delhi. The first column shows the ground truth data and the second and third column is an image tile of the size of $224x224$ pixel for a red, green and blue 8 bit composite and a false colour near infra-red, red and green 8 bit composite.

In this thesis multiple procedures for fine-tuning the FCN are proposed. Until now, slum mapping was often only used on small areas for investigating methodological development, but rarely methods are capable of exhaustive slum mapping in multiple cities [Graesser et al., 2012]. Presently, few methods successfully detect the diversity of slums in multiple cities. Thus, a more robust image-based systematic exploration of potential methods is required for the development of a slum inventory spreading across multiple cities [Kuffer et al., 2016]. Table 4.5 presents multiple methods of training the FCN-vgg19 used in this study to ascertain optimal class segmentation results. The FCN is trained on all available layers for the city of Mumbai, Delhi and on a combined dataset. These three networks are trained for 100 epochs. Two methods of fine-tuning are introduced to see if class segmentation results differ from a regular pre-trained FCN and a fine-tuned FCN. To test if FCNs can successfully generalize between different cities a FCN is using

pre-trained weights from one city and then it is transfer-learned to another city and vice versa. Another option of fine-tuning is using an enforced dataset, where the FCN relearns on image-tiles only containing informal settlements. This methods can improve class segmentation results, but the imbalance of informal settlements leads to fewer training image-tiles.

| pre-trained | fine-tuned | enforced-learning |
|---|---|---|
| $FCN\_Mumbai_{321}\_100$ | $FT_D\_Mumbai_{321}\_50\_L_5$ | $FT_D\_Mumbai^E_{321}\_50\_L_5$ |
| $FCN\_Mumbai_{432}\_100$ | $FT_D\_Mumbai_{321}\_50\_L_4$ | $FT_{MD}\_Mumbai^E_{321}\_50\_L_5$ |
| $FCN\_Delhi_{321}\_100$ | $FT_M\_Delhi_{321}\_50\_L_5$ | $FT_{MD}\_Delhi^E_{321}\_50\_L_5$ |
| $FCN\_Delhi_{432}\_100$ | $FT_M\_Delhi_{321}\_50\_L_4$ | |
| $FCN\_MD_{321}\_100$ | | |

Table 4.5: Training methodology for the fully convolutional network.

## 4.3 Performance evaluation of the FCN-vgg19

### 4.3.1 Accuracy measures

In the field of supervised image classification, a machine learning model is applied to classify multiple satellite image scenes. To assess the performance of the FCN it is of major importance to determine the quality of the prediction process. Only then it is possible to receive a quantitative impression of the class segmentation quality. The accuracy of remote sensing image class segmentation can be assessed using different measures. Facing an imbalance of the land cover / land use classes in this study, some accuracy measures do not reflect the visual impression of the classification quality. Since informal settlements only make up a small percentage of classes in the dataset the accuracy of the metrics which are most important is under-represent by an imbalanced class distribution [Mosley, 2013]. To address this shortcoming, class specific accuracy measures can be calculated.

#### Overall accuracy measures

Accuracies are reported using five evaluation metrics commonly applied in semantic segmentation and scene parsing tasks. Let $n_{ij}$ be the number of pixels of class $i$ predicted to belong to class $j$, where there are $n_{cl}$ different classes, and let $t_i = \sum_j n_{ij}$ be the total number of pixels of class $i$. The most common measure of determining the accuracy of a classified image is the overall Pixel Accuracy (oPA) computed with equation 4.4.

$$oPA = \frac{\sum_i n_{ii}}{\sum_i t_i} \tag{4.4}$$

One significant limitation of the oPA measure is its bias in the presence of very imbalanced classes. The Intersection over Union (IoU) is an evaluation metric used to measure the accuracy of an image segment in a particular dataset. The IoU thus takes into account both false alarms and the missed values for each class [Csurka et al., 2013]. This solves the issue concerning oPA and it is nowadays the standard metric to evaluate the PASCAL VOC challenge [Everingham et al., 2010]. Computing the Intersection over Union in equation 4.5 is dividing the area of all true positives $\sum_i n_{ii}$ by the area of union. In this case $n_{ij}$ is the number of pixels in class

$i$ predicted to belong to class $j$ and $t_i$ the total number of pixels of class $i$. The mean IoU is straightforwardly the average over all classes. Because of this, the IoU defines an evaluation metric that rewards predicted bounding boxes for heavily overlapping with the ground-truth as seen in figure 4.6. Predicted bounding boxes that heavily overlap with the ground-truth bounding boxes have higher scores than those with less overlap. This makes the mean IoU an excellent metric for evaluating class segmentation results. Scores above 50% can be considered moderate while scores above 60% reflected respectable results [Rosebrock, 2016].

$$
\begin{aligned}
IoU &= \frac{\sum_i n_{ii}}{t_i + \sum_j n_{ji} - n_{ii}} \\
n_{ii} &: \text{number of correctly classified pixels} \\
n_{ij} &: \text{number of pixels wrongly classified} \\
t_i &: \text{total number of pixels of class } i
\end{aligned}
\tag{4.5}
$$

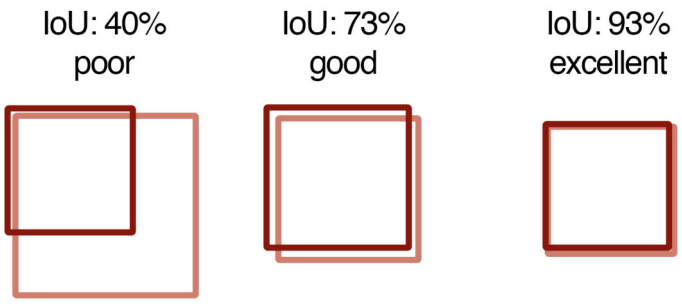


Figure 4.6: The Intersection over Union for various bounding boxes. Predicted bounding boxes that heavily overlap with the ground-truth bounding boxes have higher scores than those with less overlap [Rosebrock, 2016].

Another measure widely used for quality assessment in remote sensing segmentation studies is the Kappa value $\kappa$ [Cohen, 1960]. It is used to estimate the difference between an achieved classification result and random chance. The $\kappa$ score expresses the level of agreement between two annotators on a classification problem. It is defined as present in equation 4.6, where $p_0$ is the empirical probability of agreement on the label assigned to any sample (the observed agreement ratio), and $p_e$ is the expected agreement when both annotators assign labels randomly. $p_e$ is estimated using a per-annotator empirical prior over the class labels.

$$
\kappa = \frac{p_0 - p_e}{1 - p_e} \tag{4.6}
$$

Specifically for statistical accuracy measurements the confusion matrix is employed for validation purposes. The confusion matrix is a specific table layout that allows visualization of the performance of a supervised machine learning classifier. Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class. In this matrix true positives $tp$ and true negatives $tn$ are the observations that are correctly predicted, whereas false positives $fp$ and false negatives $fn$, occur when the ground truth class contradicts with the predicted class. With these informations the precision and recall accuracy metric can be produced as seen in equation 4.7. Precision is the ratio of correctly predicted positive observations

to the total predicted positive observations. High precision relates to the low false positive rate. Recall is the ratio of correctly predicted positive observations to the all observations in actual class.

$$\begin{aligned} precision &= \frac{tp}{tp + fp} \\ recall &= \frac{tp}{tp + fn} \\ tp &: \text{True positive} \\ fp &: \text{False positive} \\ fn &: \text{False negative} \end{aligned} \tag{4.7}$$

**Class specific accuracy measurements**

The accuracy metrics from above provide an estimate of the classification as a whole. Since slum mapping suffers from imbalanced class distribution, where informal settlements are under-represented, class specific accuracy measures are used to compensate the imbalance of overall accuracy metrics. For this purpose the accuracy metric oPA, mean IoU, precision and recall are calculated using pixel masks for all predictions containing informal settlements.

### 4.3.2 Evaluation of a mosaic created from the FCN prediction

The FCN is limited by a small processing image-tile of $224x224$ pixels. To produce large scale geo-data of informal settlements a fully trained FCN is used to predict a dataset specificity set up for this task. Mosaicking is the process of combining multiple, individual images into a single scene, where the process yields a new raster dataset. To counter misbehaviours in classification near the image edges, tiles overlap to the extent of 150 pixels. Using this method every predicted pixel overlaps with six other available predictions and a modal operator produces a majority based mosaic. For both mosaics random samples are selected in a similar fashion as presented in section 4.1.2 to present a confusion matrix to validate the mosaics to reality data from its input optical image data.

## 4.4 Water supply infrastructure based on results of different geo-data sources

Slum systems are determined by the geographical position and its area. The settlements present in the mosaic created from overlapping image-tiles predicted by the FCN are used as input data to fabricate an optimal fresh water supply chain connecting all informal settlements to each other. Using a origin destination (OD) matrix the length of each possible pipes along the street network can be calculated as seen in table 4.6. With the OD Matrix Kruskal's algorithm is used to calculate the shortest path of the three configurations presented in section 3.2.1. Figures 4.7 to 4.9 show the procedure for an optimal water supply chain for informal settlements contained in the ground truth dataset. Three methods are proposed to built an optimal water supply chain. A shortest path connecting all slums with each other is calculated. A pipe diameter between $60mm$ to $600mm$ can be chosen depending on the water needed to supply all slum dwellers. First a water supply network is determined using a single pipeline connecting all slums via the shortest path. The second methods uses a hierarchical structure for the network model. Geospatial clusters and large slums form two separate networks. All slums are clustered according to the geographical location and a hierarchical system is built. The largest slums form an underlying

network connected by only large roads and providing water for all slums. The water supply network for each cluster form a network providing water for all slums within each cluster except the largest slum. The general idea is to use pipes with a large diameter for the underlying network and smaller pipes for the cluster networks to lower the cost of investment. The third and last option is a geospatial cluster, where centre slums form a separate network. Instead of using the largest slums of each cluster, this approach uses the centre of each cluster to lower the length of the network providing water for all slums. Centre slums are again connected via the road network using only large roads and the cluster networks use all roads for the pipe network optimized to a shortest path. Figure 4.7 illustrates all informal settlements for both Mumbai and Delhi in the ground truth dataset. The slums are clustered into four groups by a Delaunay triangulation, which ensures all group members to be proximal. This warrants that all slums in the same group will have at least one natural neighbour in common with another slum in the same group. In figure 4.8 the origin destination matrix (OD) is calculated for all three approaches and measures the least-cost paths along the network from multiple origins to multiple destinations along a given road network. Figure 4.8 only shows the OD matrices for the underlying networks of the hierarchical approaches. Table 4.6 shows the OD matrix for the largest slums in the ground truth dataset for Delhi. The OD matrix is used for the input weights to optimize Kruskal's algorithm for the shortest possible path along a given road infrastructure. Figure 4.9 shows the shortest path in a hierarchical approach. Slums are either connected through the largest or closest to a clusters centre using only large roads. All other informal settlements are connected using all available roads.

| Origin/Destination | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 0m | 8829m | 5224m | 2588m |
| 2 | 8829m | 0m | 1153m | 7154m |
| 3 | 5524m | 11538m | 0 | 6361m |
| 4 | 2588m | 7154m | 6361m | 0m |

Table 4.6: OD matrix showing the weights of possible direction calculated by the distance along the road infrastructure

A water supply network for Mumbai and Delhi is calculated with the help of cost functions introduced in section 3.2.2. Results are calculated using population densities from various researches to identify the water needed for all slum dwellers. Studies report an approximate area of $0.4m^2$ available per person in certain slums of Mumbai [Chinmayi & Madhavi, 2013], whereas other report densities of $280000pax/km^2$ to $350000pax/km^2$ [Fernando, 2009]. For this study a population density introduced by [Taubenböck & Wurm, 2015b] of $0.22pax/m^2$ is used. For the three available network approaches water needs are calculated with the area of each slum and an average population density. The cost functions in equation 4.8 to 4.10 are depended on the pipe diameter and pipeline length. Kruskal's shortest path delivers the total water supply length for each network approach. The pipeline diameter is calculated depending on how much water a supply network has to carry. All three water supply network approaches use different pipeline diameters. Since water needed to supply all slum dwellers stays the same for all three approaches only the length determines the diameter of the pipes, where shorter networks need a larger diameter and vice versa. This gives an interesting insight if the cost of water supply networks is more depended on pipeline length or its diameter. Using a flow nomogram for polyethylene pipes and assuming an water velocity of $1m/s$ the volume flow $Q$ and pressure loss $\Delta p$ can be obtained. These results form the input for the cost function to calculate the investment in setting up a completely new water pipeline infrastructure. Applying the cost functions to the ground truth dataset and calculating the cost for a water supply network connecting all informal settlements

in Mumbai via a shortest path results in the following equations 4.8 to 4.10. The shortest path connecting all slums is $118932m$ long according to the calculations from Kruskal's algorithm in section 3.2.1. Since this methods needs to supply all informal settlements with fresh water a diameter of $370mm$ is chosen. The cost functions includes all variables for the network construction of the transmission and distribution pipes. The parameters in equation 4.8 are based on polyethylene pipes, which are best suited for a diameter of $0.06 - 0.7m$ [Marchionni et al., 2015]. To model variable costs in equation 4.9 the dissipated power for pipeflows is cast with the volume flow $Q$ and the pressure loss $\Delta p$. The volume flow $Q$ is determined by the diameter of pipes in use and the velocity of water in pipes of $1m/s$. The pressure loss $\Delta p$ is calculated with a hydraulic gradient from a flow nomogram for pipework systems. A pipe diameter of $370mm$ with a flow rate of $90l/s$ corresponds to a hydraulic gradient of $0.25m/100m$. The total pressure loss along the network is the hydraulic gradient multiplied by the length of all pipes. Total investment for a water supply network using informal settlements present in the ground truth dataset of Mumbai and assuming a operation time of 8 hours a day for 365 days would cost about 5 million €.

$$
\begin{aligned}
C_{fix}(d,l) &= (32.59 + 0.11d + 0.00053d^2)l \\
C_{fix}(d,l) &= (32.59 + 0.11 * 0.37m + 0.00053 * 0.37^2m) * 118932m \\
C_{fix}(d,l) &= \mathbf{3886812.99€} \\
C_{fix}(d,l) &: \text{Fixed costs [€]} \\
d &: \text{Pipe diameter [m]} \\
l &: \text{Pipe length [m]}
\end{aligned}
\tag{4.8}
$$

$$
\begin{aligned}
C_{var}(d,l) &= \frac{1}{\eta} C_{kWh} P_{Diss}(d,l) t \\
P_{Diss} &= Q\Delta p \\
P_{Diss} &= 90l/s * 297.33m = 251541W \\
C_{var}(d,l) &= \frac{1}{0.55} * 0.00757€/kWh * 251541W * 8h \\
C_{var}(d,l) &= \mathbf{2946.48624€} \\
C_{var}(d,l) &: \text{Variable costs per day[€]} \\
C_{kWh} &: \text{Energy price [€/kWh]} \\
P_{Diss} &: \text{Dissipated Power [watt]} \\
Q &: \text{Volume flow [l/s]} \\
\Delta p &: \text{Pressure loss [m]} \\
\eta &: \text{Pump efficiency} \\
t &: \text{Operating time per day [h]} \\
d &: \text{Pipe diameter [m]} \\
l &: \text{Pipe length [m]}
\end{aligned}
\tag{4.9}
$$

$$
C(d,l) = C_{var}(d,l) * 365days + C_{fix}(d,l) = \mathbf{4959333.99€} \tag{4.10}
$$

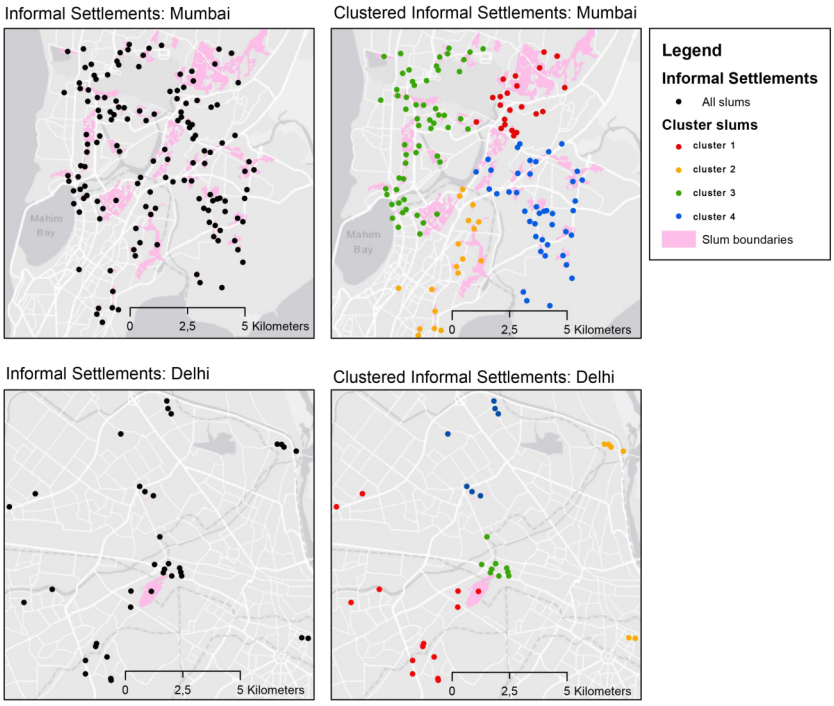


Figure 4.7: Informal settlements in the ground truth dataset of Mumbai and Delhi. The slums are clustered into four groups by a Delaunay triangulation.

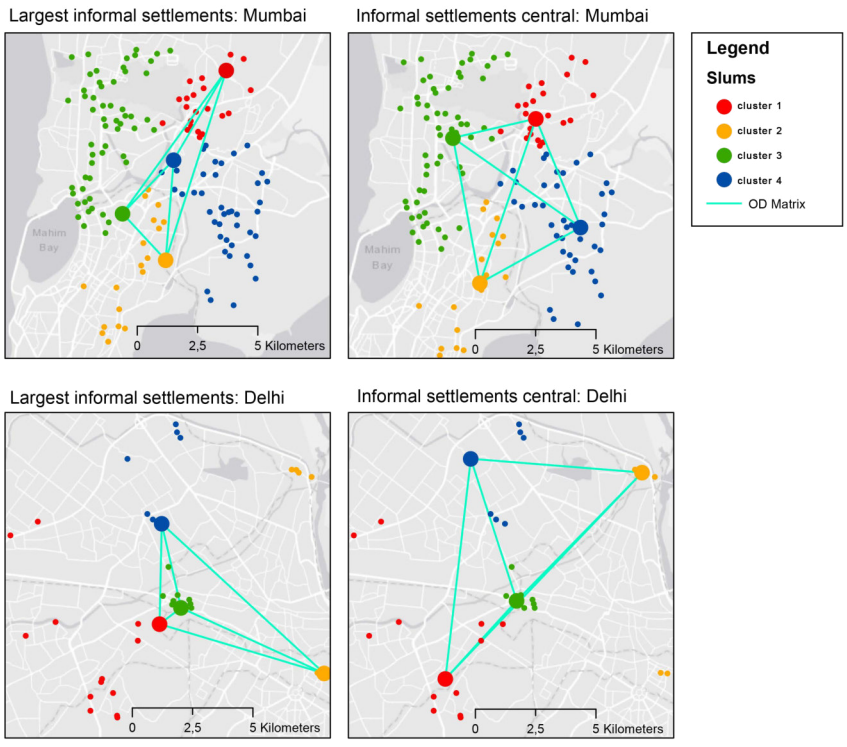


Figure 4.8: The origin destination matrix for Mumbai and Delhi for the largest and for the centre slums of each cluster.

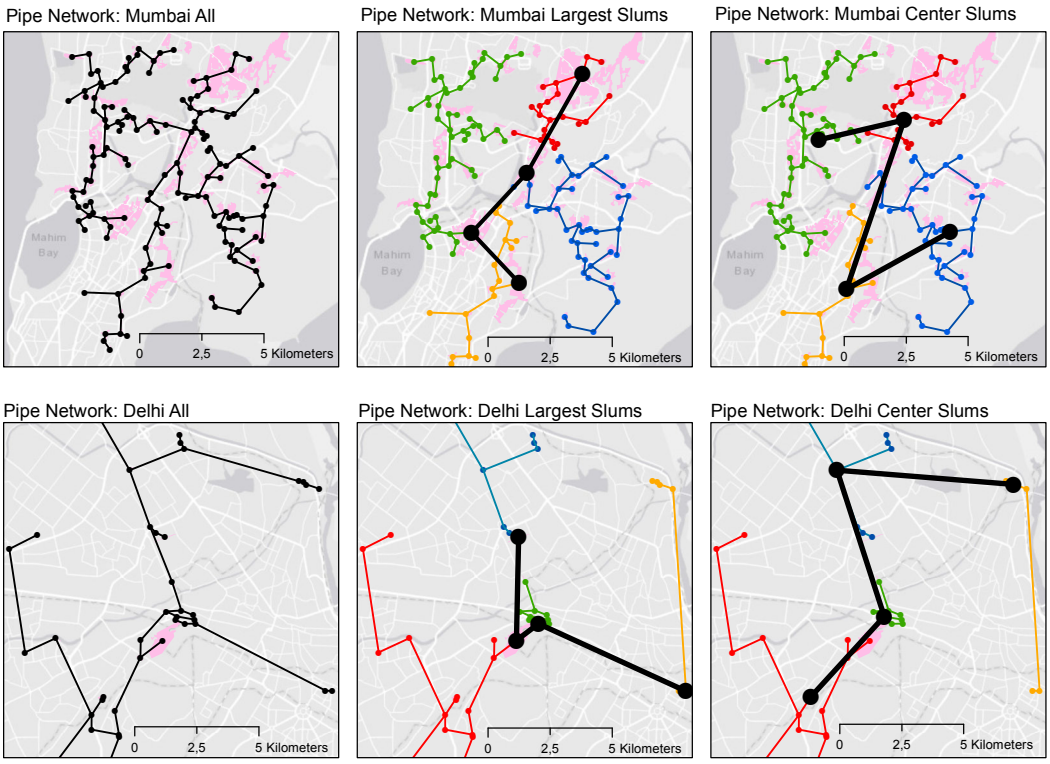


Figure 4.9: Kruskal's algorithm is used to find the optimal path connecting all informal settlements to the road infrastructure. The illustration shows the shortest path connecting all informal settlements using three different water supply network approaches.

# 5 Results

Following the experimental setup, the class segmentation results are presented in section 5.1. The classification results are divided into overall accuracy measurements and metrics specifically calculated for informal settlements. The best performing FCN is used to calculate the investment in setting up a water pipeline infrastructure for slum dwellers.

## 5.1 Class segmentation of informal settlements

### 5.1.1 Overall accuracy measurements

For the assessment of the overall classification quality, the measures overall pixel accuracy (oPA), mean Intersection over Union (mIoU), Kappa estimate ($\kappa$), precision and recall values are calculated. The accuracy metrics are presented in table 5.1 for pre-trained FCNs, in table 5.2 for transfer-learned FCNs and for fine-tuned FCNs using a enforced dataset in table 5.3. The pre-trained FCNs were both trained on a true (321) and a false (432) colour composite for 100 epochs. A combination of both true colour datasets from Mumbai and Delhi is trained for 100 epochs to test if a FCN can generalize between multiple cities. Transfer-learning learning is only tested on datasets a FCN has not been trained on to avoid overfitting. All fine-tuned networks are only trained on selected layers $L_n$ for 50 epochs, where $n$ represents the first convolutional block available for learning, while the other weights and biases are not updated from its original pre-trained FCN.

| FCN | $oPA$ | $mIoU$ | $\kappa$ | $precision$ | $recall$ |
|---|---|---|---|---|---|
| $Mumbai_{321}$_100 | 81.59% | 57.34% | 75.75% | 82.40% | 82.45% |
| $Mumbai_{432}$_100 | 82.96% | 59.30% | 75.61% | 81.91% | 82.03% |
| $Delhi_{321}$_100 | 84.14% | 61.20% | 75.62% | 83.31% | 83.32% |
| $Delhi_{432}$_100 | **88.41%** | **69.37%** | **83.39%** | **88.39%** | **88.41%** |
| $MD_{321}$_100 | 83.18% | 60.12% | 77.41% | 83.42% | 83.50% |

Table 5.1: Overall accuracy measurements for all pre-trained FCNs. The networks are trained for 100 epochs.

In general false colour composite datasets deliver better accuracies for both cities on all measurements. The overall pixel accuracy ranges from 81% to 88% and shows respectable results. For class segmentation methods the mean intersection over union provides a very important metric for comparing boundaries of detected object classes. All FCNs score higher than 57%, with the FCN $Delhi_{432}$_100 achieving a $mIoU$ of 69%. The FCN using the combined true colour image data $MD_{321}$_100 scores with an overall pixel accuracy of 83% and a mIoU of 60%.

The kappa estimate $\kappa$ of 75% to 83% indicate a substantial strength of agreements for all pre-trained FCNs. The precision is intuitively the ability of the FCN to not label a positive sample as a negative. With scores from 81% to 88% the precision proves to be a strong indicator for correctly predicted pixels. The recall score, which presents the reliability of classes in the predicted image, is the fraction of correctly predicted pixels with regard to all classified pixels. The recall scores vary from 82% to 88% with the FCN $Delhi_{432}\_100$ achieving the highest score again.

| FCN | $oPA$ | $mIoU$ | $\kappa$ | $precision$ | $recall$ |
|---|---|---|---|---|---|
| $FT_D\_Mumbai_{321}\_50\_L_5$ | 71.63% | 44.55% | 58.36% | 69.49% | 69.50% |
| $FT_D\_Mumbai_{321}\_50\_L_4$ | 77.21% | 51.24% | 67.72% | 76.18% | 76.17% |
| $FT_M\_Delhi_{321}\_50\_L_5$ | 78.49% | 52.64% | 68.95% | 78.38% | 78.82% |
| $FT_M\_Delhi_{321}\_50\_L_4$ | **81.12%** | **56.67%** | **69.51%** | **79.32%** | **78.94%** |

Table 5.2: Overall accuracy measurements for all transfer learned FCNs. Networks are initialized on one pre-trained FCN and fine-tuned to another city's dataset. The fine-tuned FCNs are only trained on the last (convolutional block 5) and penultimate (convolutional block 4) layer of the network.

The fine-tuned FCNs in table 5.2 show overall slightly worse accuracy scores than the pre-trained FCNs. With a Kappa score of 58% to 69% the results are still moderately acceptable, but the overall pixel accuracy drops down to 10% compared to pre-trained FCNs. Equipping the FCN with more features to learn increases accuracy measurements across all fine-tuned FCNs. Accuracy gains of about 5% can be measured for all accuracy metrics when more layers are available for learning. The mean intersection over union drops below 50% for the $FT_D\_Mumbai_{321}\_50\_L5$ FCN.

| | | | | | |
|---|---|---|---|---|---|
| $FT_{MD}\_Mumbai^E_{321}\_50\_L_5$ | 93.11% | 75.71% | 90.57% | 93.28% | 93.31% |
| $FT_{MD}\_Delhi^E_{321}\_50\_L_5$ | 93.07% | 77.93% | 91.41% | 93.76% | 93.77% |
| $FT_D\_Mumbai^E_{321}\_50\_L_5$ | 68.44% | 38.22% | 56.41% | 68.38% | 68.12% |

Table 5.3: Overall accuracy measurements for all fine-tuned FCNs on an enforced dataset containing only images with informal settlements. The fine-tuned and enforced FCNs are trained only on the last (convolutional block 5) layer of each FCN.

Results for enforced learning techniques are presented in table 5.3. Using only ground truth images containing informal settlements a pre-trained FCN is fine-tuned a very limited set of data. These FCNs are only trained for 50 epochs on the last (convolutional block 5) layer of each FCN. Since fine-tuning the combined dataset $MD_{321}\_100$ on already learned image tiles the results for both FCNs $FT_{MD}\_Mumbai^E_{321}\_50\_L_5$ and $FT_{MD}\_Delhi^E_{321}\_50\_L_5$ achieve accuracy metrics of over 90% oPA and the mean IoU of over 75%. These metrics will be not considered in the final evaluation since they can contain training images due to a random shuffel of images. Fine tuning from the Delhi dataset to the enforced Mumbai dataset results in a oPA of 68% with a mIoU of only 38%.

### 5.1.2 Accuracy measurements for informal settlements

Because informal settlements are underrepresent in the ground truth dataset, overall accuracy measurements tend to distort metrics of an imbalanced dataset, especially in Delhi where slums only make up 1.8% of all available land use / land cover classes in the ground truth dataset. The following metrics are specifically set up to extract accuracy measurements for slums. With the overall Pixel Accuracy (oPA), mean Intersection over Union (mean IoU), precision and recall similar metrics are chosen for benchmarking all trained FCNs.

| FCN | $oPA_{slums}$ | $mIoU_{slums}$ | $precision_{slums}$ | $recall_{slums}$ |
|---|---|---|---|---|
| $Mumbai_{321}\_100$ | 73.90% | 65.05% | **88.53%** | 86.84% |
| $Mumbai_{432}\_100$ | 77.54% | 66.12% | 78.86% | 85.10% |
| $Delhi_{321}\_100$ | 44.43% | 39.98% | 58.38% | 59.41% |
| $Delhi_{432}\_100$ | 53.25% | 48.85% | 59.86% | 61.27% |
| $MD_{321}\_100$ | **77.70%** | **67.65%** | 78.07% | **86.95%** |

Table 5.4: Accuracy measurements for informal settlements of all pre-trained FCNs. The networks are trained for 100 epochs.

Table 5.4 shows accuracy metrics for informal settlements on pre-trained FCNs. The FCNs show great differences in the used dataset. The overall Pixel Accuracies for the Mumbai dataset range from 74% to 77%, with a mean IoU of 65%. The FCNs trained on the Delhi dataset perform in an unsatisfactory manner. Precision and recall metrics for both Delhi datasets present low precision and under classification of informal settlements. The combined dataset $MD_{321}\_100$ assumes to the task of correctly mapping a slum's boundaries in different cites with an mean IoU of 67%, although its precision of 78% indicates that some informal settlements were missed.

| FCN | $oPA_{slums}$ | $mIoU_{slums}$ | $precision_{slums}$ | $recall_{slums}$ |
|---|---|---|---|---|
| $FT_D\_Mumbai_{321}\_50\_L_5$ | 66.70% | 55.79% | 75.84% | 75.36% |
| $FT_D\_Mumbai_{321}\_50\_L_4$ | **75.11%** | **62.54%** | **77.25%** | **81.90%** |
| $FT_M\_Delhi_{321}\_50\_L_5$ | 39.45% | 35.76% | 38.37% | 15.08% |
| $FT_M\_Delhi_{321}\_50\_L_4$ | **53.08%** | **45.11%** | **82.96%** | **38.80%** |

Table 5.5: Accuracy measurements for informal settlements for all transfer learned FCNs. Networks are initialized on one pre-trained FCN and fine-tuned to another city's dataset. The fine-tuned FCNs are only trained on the last (convolutional block 5) and penultimate (convolutional block 4) layer of the network.

Transfer-learned FCNs gradually decline in accuracy as seen in table 5.5. A drop of 5% to 10% in overall pixel accuracy occurs if only the last layer is fine-tuned to a new dataset. The oPA improve considerably when more layers are available for learning and the $FT_D\_Mumbai_{321}\_50\_L_4$ FCN reaches almost same pixel accuracy of a pre-trained FCN with 75%. Similar trends of improvement can be seen for the other FCNs when fine-tuning the FCN from the fourth convolutional block. Most strikingly the FCN $FT_M\_Delhi_{321}\_50\_L_4$ showing dissatisfactory results is able to score very high in its precision metric of 83%, combined with a poor recall metric of just 38%. To sum it up while most informal settlements are detected, a great deal of miss-classification is also present.

| FCN | $oPA_{slums}$ | $mIoU_{slums}$ | $precision_{slums}$ | $recall_{slums}$ |
|---|---|---|---|---|
| $FT_{MD}\_Mumbai^{E}_{321}\_50\_L_5$ | 93.37% | 85.92% | 95.73% | 97.24% |
| $FT_{MD}\_Delhi^{E}_{321}\_50\_L_5$ | 90.19% | 81.21% | 96.38% | 97.68% |
| $FT_{D}\_Mumbai^{E}_{321}\_50\_L_5$ | 71.85% | 57.15% | 80.86% | 85.55% |

Table 5.6: Accuracy measurements for informal settlements

Results for enforced learning techniques are presented in table 5.6. Using only ground truth images containing informal settlements a pre-trained FCN is fine-tuned on a very limited set of data. These FCNs are only trained for 50 epochs on the last (convolutional block 5) layer of each FCN. Since fine-tuning the combined dataset $MD_{321}\_100$ on already learned image tiles the results for both FCNs $FT_{MD}\_Mumbai^{E}_{321}\_50\_L_5$ and $FT_{MD}\_Delhi^{E}_{321}\_50\_L_5$ score with accuracy metrics of over 90% oPA and the mean IoU of over 80%. They will be not considered in the final evaluation since these FCNs can contain training images due to a random shuffel of images. Training accuracy still scores higher with about 5% than validation image tiles for both FCNs. An enforced learning technique to fine-tune the FCN $FT_{D}\_Mumbai^{E}_{321}\_50\_L_5$ from the Delhi dataset to the Mumbai dataset is tested with only 1652 image tiles for training and 552 validation images. Overall pixel accuracy is 4% worse than the best scoring fine-tuned FCN $FT_{D}\_Mumbai_{321}\_50\_L_4$. The difference can be caused by learning on a complete dataset of 5616 training images and 1872 validation image tiles. Nevertheless the enforced FCN scores better in oPA and mean IoU then the fine-tuned FCN only trained on the last layer of a pre-trained network of $FT_{D}\_Mumbai_{321}\_50\_L_5$. Moreover $FT_{D}\_Mumbai^{E}_{321}\_50\_L_5$ proves best in class precision and recall scores for informal settlements with scores of 81% and 86% respectively.

Figure 5.1 and 5.2 present an overview of error bars for accuracy metrics of informal settlements of all trained FCNs. The oPA in illustration 5.1 shows that all FCNs suffer from high standard deviations. Similar measurements are present in the accuracy scores for the mean IoU for slums in figure 5.2. In general both illustrations confirm an improvement when training a pre-trained FCN on a false colour composite. $Mumbai_{432}\_100$ and $Delhi_{432}\_100$ prove that FCNs can adapt from a true color ImageNet data to false colour remote sensing images. Furthermore when using fine-tuning techniques FCNs tend to score better if more layers are available for training. Both FCN versions of the $FT_{x}\_y_{321}\_50\_L_4$ FCN, where $x$ and $y$ represent the dataset for each study area, deliver better accuracy metrics than their $FT_{cityA}\_cityB_{321}\_50\_L_5$ counterpart. Training a FCN on a combined dataset with the $MD_{321}\_100$ FCN shows that applying it universally to multiple cities is possible with highest scoring results for informal settlements with an oPA of 77.70% and a mean IoU of 67.65%.

## 5.2 Using fully convolutional networks for large scale slum mapping

Images used for training all variations of the FCN-vgg19 are commonly only of small sizes, since the learning process within the neural network is very compute intensive. Since remote sensing images cover large areas, the scenes used for the prediction informal settlements were split into $150,000$ tiles for Mumbai and $300,000$ tiles for the complete Delhi scene. With an overlap of 150 pixels for each tile enough data for an majority based mosaicking method was chosen. The area for classifying Mumbai stays with $103km^2$ the same, while the Delhi area increases from $80km^2$

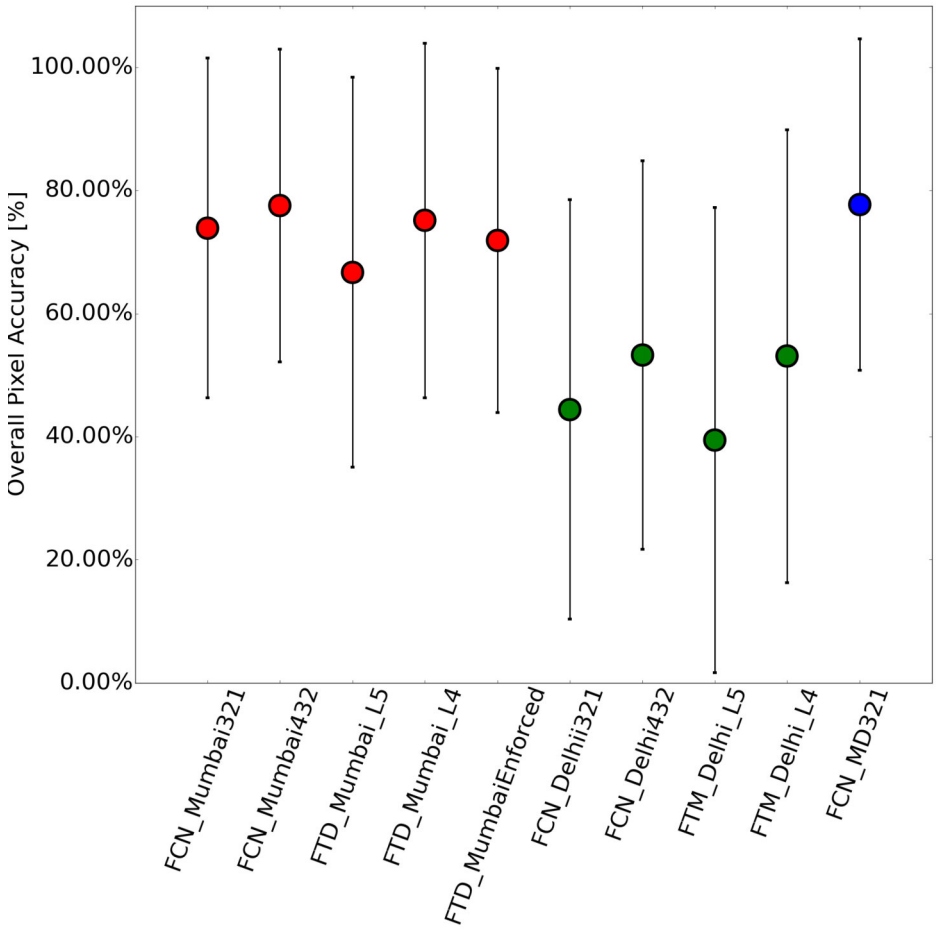


Figure 5.1: Comparative alignment of error bars representing the oPA and its standard deviation for all FCNs. Red error bars correspond to the Mumbai dataset, while green bars apply to the Delhi dataset. The blue error bar describes the metric for the combined dataset of both cities.

used for training the FCN to $300km^2$ for classification. Class segmentation results can be seen in figure 5.3 for the pre-trained $FCN\_MD_{321}\_100$.

Using the same random sample accuracy evaluation used for testing the ground truth data in section 4.1.2 the mosaics predicted by the $FCN\_MD_{321}\_100$ are tested in table 5.7 for the Mumbai mosaic and in table 5.8 for the Delhi mosaic. The confusion matrices compare the FCN prediction to the reality of the VHR satellite image input data. An increase in performance compared to section 5.1 can be seen throughout the accuracy metrics. Overall accuracy for Mumbai is 90.6% with a Kappa value of 88.3% and for Delhi 89.3% with a Kappa value of 86.6%. Precision scores for informal settlements are with 93% for Mumbai and 92% for Delhi very high. Recall values confirm the high accuracy metrics scored by the FCN with values of 96% and even 98%. With a sample size of 1000 random points accuracy metrics are represented by a 95% confidence level with 1% margin of error.

Extracting informal settlements from the mosaics in figure 5.3 yields the input geo-data for the process to provide water to the map slums via a holistic water supply network. Illustration 5.4 presents an overview on detected informal settlements in both scenes compared to the ground truth dataset for each dataset. The y-axis showing the size of detect slums is logarithmic due a large range in smallest and larges informal settlements. The error bars illustrate that the FCN

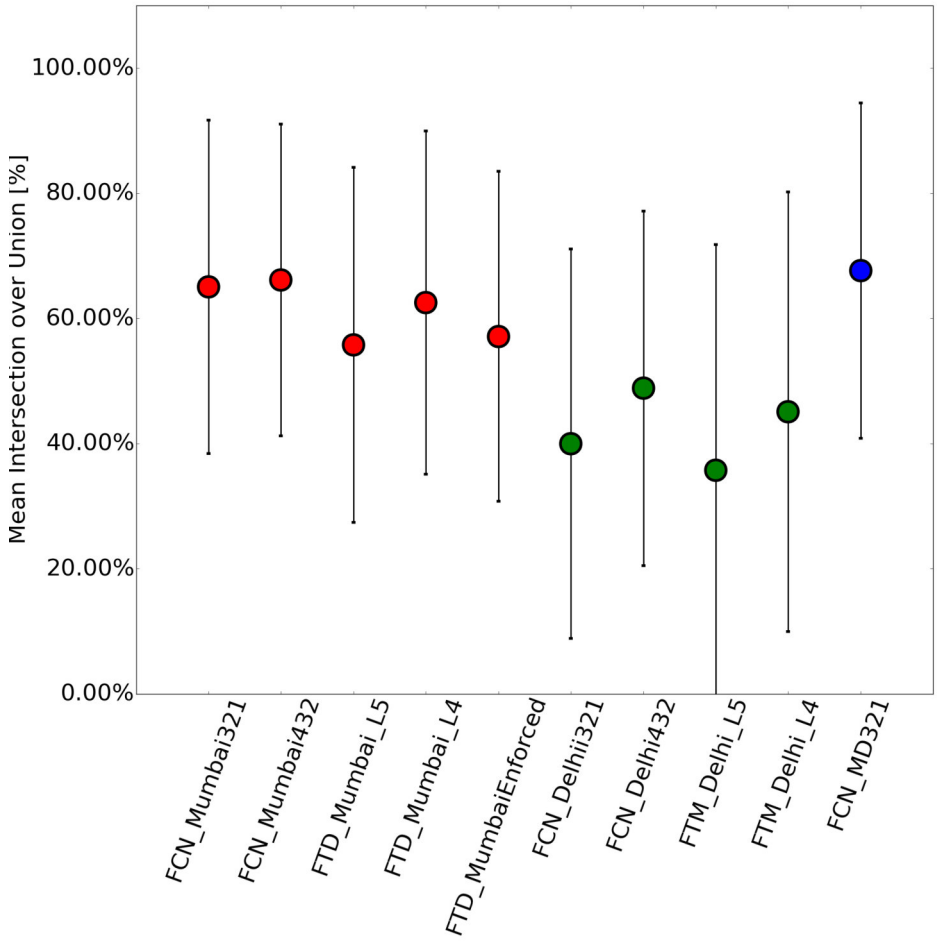


Figure 5.2: Comparative alignment of error bars representing the mIoU and its standard deviation for all FCNs. Red error bars correspond to the Mumbai dataset, while green bars apply to the Delhi dataset. The blue error bar describes the metric for the combined dataset of both cities.

| Classification | Formal | Ground | Vegetation | Water | Slum | Sum | Precision |
|---|---|---|---|---|---|---|---|
| Formal | 172 | 19 | 6 | 1 | 2 | 200 | 86% |
| Ground | 8 | 179 | 10 | 2 | 1 | 200 | 90% |
| Vegetation | 7 | 6 | 184 | 0 | 3 | 200 | 92% |
| Water | 0 | 6 | 8 | 186 | 0 | 200 | 93% |
| Slum | 10 | 4 | 1 | 0 | 185 | 200 | 93% |
| Sum | 197 | 214 | 209 | 189 | 191 | 1000 | |
| Recall | 87% | 84% | 88% | 98% | 96% | | |

Table 5.7: Confusion matrix for the accuracy assessment of the Mumbai FCN.

is able to detect even small patches of informal settlements. The FCN contains 113 more slums than the ground truth dataset in Mumbai and in Delhi the difference is 178. Minimum size for slums in Mumbai in the ground truth dataset is $1017m^2$, while the FCN can detected slums as small as $474m^2$. In Delhi smallest detect slums by the FCN are with $130m^2$ $30m^2$ smaller than in the ground truth dataset.

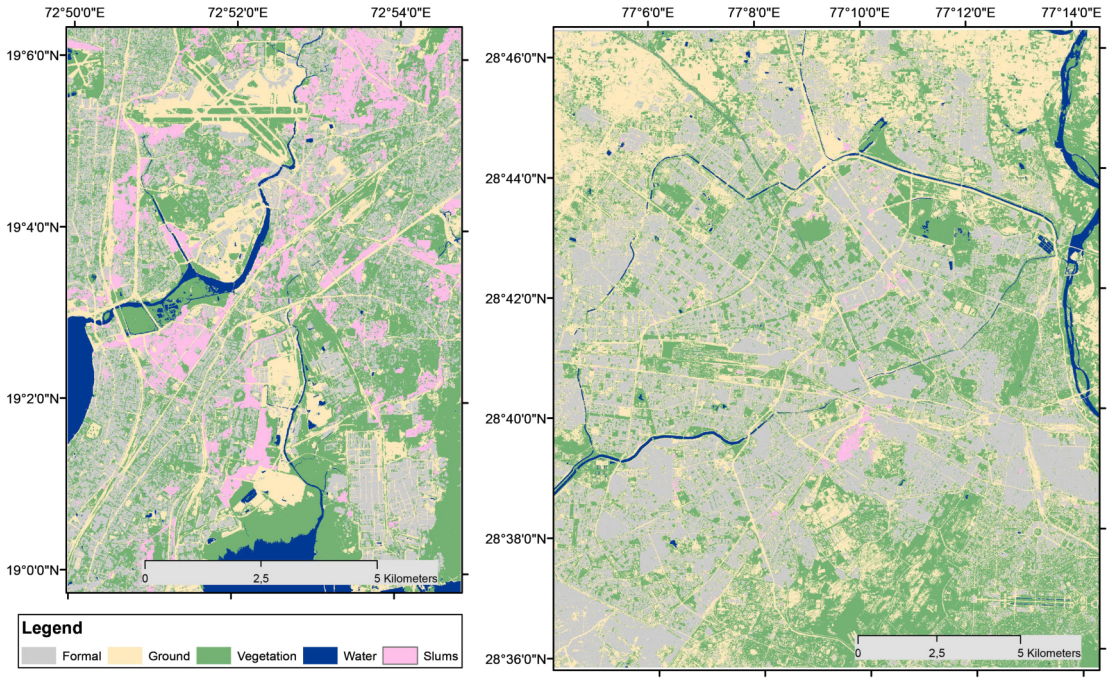


Figure 5.3: ...

| Classification | Formal | Ground | Vegetation | Water | Slum | Sum | Precision |
|---|---|---|---|---|---|---|---|
| Formal | 174 | 22 | 2 | 0 | 2 | 200 | 87% |
| Ground | 16 | 172 | 10 | 2 | 0 | 200 | 86% |
| Vegetation | 0 | 18 | 180 | 2 | 0 | 200 | 90% |
| Water | 0 | 8 | 8 | 184 | 0 | 200 | 92% |
| Slum | 5 | 11 | 1 | 0 | 183 | 200 | 92% |
| Sum | 195 | 231 | 201 | 188 | 185 | 1000 | |
| Recall | 90% | 74% | 89% | 98% | 98% | | |

Table 5.8: Confusion matrix for the accuracy assessment of the Delhi FCN

## 5.3 Investment for water supply infrastructure

Informal settlements detected by the $FCN\_MD_{321}$ are used as input geodata to calculate the cost for three approaches optimized to a shortest path connecting the water supply network to all slums. Investment cost is depended on the total length of the network and its diameter, which is defined by the volume needed to supply slum dwellers with enough water. Since all three network approaches use different parameters for their infrastructure, cost differs depending on pipeline length, diameter and duration of the excepted service life. Results can be seen in table 5.9 for the water supply network in Mumbai and in table 5.10 for Delhi. Table 5.9 shows an expected increase in the investment for the water supply network using geodata from the FCN since it contains more informal settlements. After 10 years of operation the water supply network using the simplest approach connecting all slums with a pipeline via a shortest path and one unique diameter of $300mm$ for all pipe costs roughly $16,4$ million €. Both hierarchical approaches are $3-4$ million € more expensive. Differences in both hierarchical approaches are minimal with 1million € after a 10 year duration. The variable cost $C_{var}$ increases more over time the larger the pipe diameters are. Thus the water supply network connecting the largest slums is able

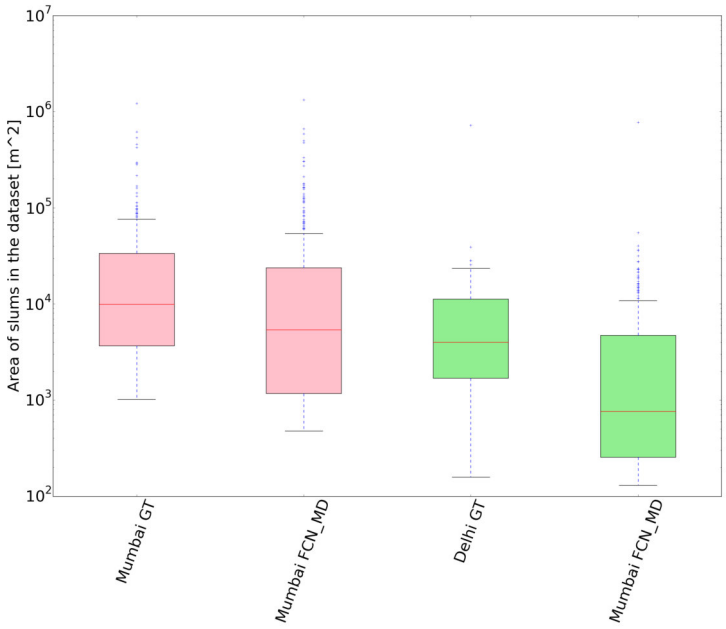


Figure 5.4: Boxplot showing difference in slum present in the ground truth dataset and slums detected by the FCN.

to use smaller pipe diameters for the cluster networks and in result making the network more cost efficient over time. Table 5.10 shows a very big increase in the investment for the water supply network in Delhi using geodata from the FCN. Since the FCN detects 178 more informal settlements investment cost increases considerably for the water supply network. After 10 years of operation the water supply network connecting all slums with pipeline via a shortest path and using one unique diameter of $117mm$ for all pipes costs roughly 13 million €. Both hierarchical approaches are $300,000$-2million € more expensive. Again the hierarchical approach connecting all slums through the largest informal settlements of each cluster costs less than a connection through the centre slums.

| Mumbai | Water supply network (WSN) | $GT_{data}$ | $FCN\_MD_{321}$ |
|---|---|---|---|
| 1 Day | $WSN^{all}_{slums}$ | 3,886,812€ | 5,858,511€ |
| | $WSN^{large}_{slums}$ | 4,835,210€ | 6,800,944€ |
| | $WSN^{centre}_{slums}$ | 4,767,982€ | 6,597,901€ |
| 1 Year | $WSN^{all}_{slums}$ | 4,932,815€ | 6,909,855€ |
| | $WSN^{large}_{slums}$ | 5,759,028€ | 8,091,021€ |
| | $WSN^{centre}_{slums}$ | 5,779,271€ | 8,011,707€ |
| 10 Years | $WSN^{all}_{slums}$ | 14,373,357€ | 16,398,605€ |
| | $WSN^{large}_{slums}$ | 14,096,237€ | 19,733,611€ |
| | $WSN^{centre}_{slums}$ | 14,905,878€ | 20,770,923€ |

Table 5.9: Investment for a water supply network for informal settlements in Mumbai. A comparison of cost for the ground truth dataset and geodata predicted by a FCN for different operating times.

| Delhi | Water supply network (WSN) | $GT_{data}$ | $FCN\_MD_{321}$ |
|---|---|---|---|
| 1 Day | $WSN^{all}_{slums}$ | 2,314,813€ | 6,341,265€ |
| | $WSN^{large}_{slums}$ | 2,534,046€ | 7,022,936€ |
| | $WSN^{centre}_{slums}$ | 2,807,210€ | 7,020,193€ |
| 1 Year | $WSN^{all}_{slums}$ | 2,604,498€ | 7,002,613€ |
| | $WSN^{large}_{slums}$ | 2,821,550€ | 7,178,039€ |
| | $WSN^{centre}_{slums}$ | 3,199,596€ | 7,777,477€ |
| 10 Years | $WSN^{all}_{slums}$ | 5,218,825€ | 12,971,088€ |
| | $WSN^{large}_{slums}$ | 5,416,195€ | 13,303,030€ |
| | $WSN^{centre}_{slums}$ | 6,740,774€ | 14,611,763€ |

Table 5.10: Investment for a water supply network for informal settlements in Delhi. A comparison of cost for the ground truth dataset and geodata predicted by a FCN for different operating times.

# 6 Discussion and Conclusion

## 6.1 Discussion

In this section the results of the study are discussed. First, the suitability of using FCNs to identify informal settlements in urban environments is evaluated concerning the the different training techniques of pre-trained FCNs and fine-tuned FCNs. Subsequently, an interpretation of the investments to produce a large scale water pipe line infrastructure is discussed.

### 6.1.1 Interpretation of the results for FCN training techniques

#### Slum mapping using pre-trained FCNs

The analysis of class segmentation accuracies of multiple pre-trained FCNs reveals that deep learning methods are very capable for slum mapping in an urban environment. The overall Pixel Accuracies for slums reach up to 77% and up to 67% for the mean Intersection over Union. Thus FCNs deliver a reliable classification result, considering the difficulty of extracting geo-spatial properties of informal settlements in different cities. The difference between a false colour composite dataset and its true colour counterpart perform very similar with only minor improvements using a false colour dataset. Comparing the best results of each FCN for the true and false colour composite shows differences of lower than one percent. This aspect is decidedly interesting since all pre-trained FCNs are initialized with weights from the ImageNet dataset containing 3 channel true colour composite pictures. This infers that the used FCN is quite capable adapting from non remote sensing pictures to multi spectral satellite images.

Illustration 6.1 gives an overview of all pre-trained FCNs. In general the FCNs perform very reliable for large scale slum mapping. Boundaries are smoother than presented in the ground truth and show a more true to reality structure. All FCNs can recover the fine structures present in its input image and seem to spot informal settlements independent of its size in the image tile. Most interestingly poor results from the FCNs trained on the Delhi dataset can be explained due to the fact of very difficult differences in formal buildings and slums. $FCN\_Delhi_{321}\_100$ in row four shows the FCN detected an informal settlement while the ground truth dataset is labeled for formal buildings. Neither official data nor visual inspections can interpret the result for this tile. In the Delhi dataset formal buildings can show similar morphological structures to slums, which makes class segmentation non trivial. These effects can be seen throughout the Delhi dataset. The results show that the accuracies depended highly on the quality of the ground truth data. This effect is made more challenging due to the fact that only few informal settlements are present in the ground truth data. This can be seen in very low recall scores throughout the FCNs trained on the Delhi dataset. The FCNs tends to perform better than the ground truth data considering some visible artefacts of the quad-tree segmentation are neglected and adapted to the reality of the input image. This can cause poor accuracy results when comparing the prediction to the ground truth data.


Figure 6.1: Comparative alignment of all pre-trained FCNs.

A combination of image tiles from Mumbai and Delhi in $MD_{321}\_100$ shows that a FCN can generalize reliable between different datasets as seen in the last row of figure 6.1. The FCN delivers with 67% the highest mIoU of all pre-trained FCNs. Semantic class segmentation of informal settlements can be achieved using a generalized FCN containing image-tiles of different cites. The Mumbai and Delhi dataset presents morphological structures different enough in the ground truth data to propose a challenge for the FCN to predict semantic classes. Especially in the aspect of the land use class of interest, because slums are not only different from its formal

counterpart but are also always different from each other. The FCN $MD_{321}\_100$ manages to show that this difficult task can be tackled.

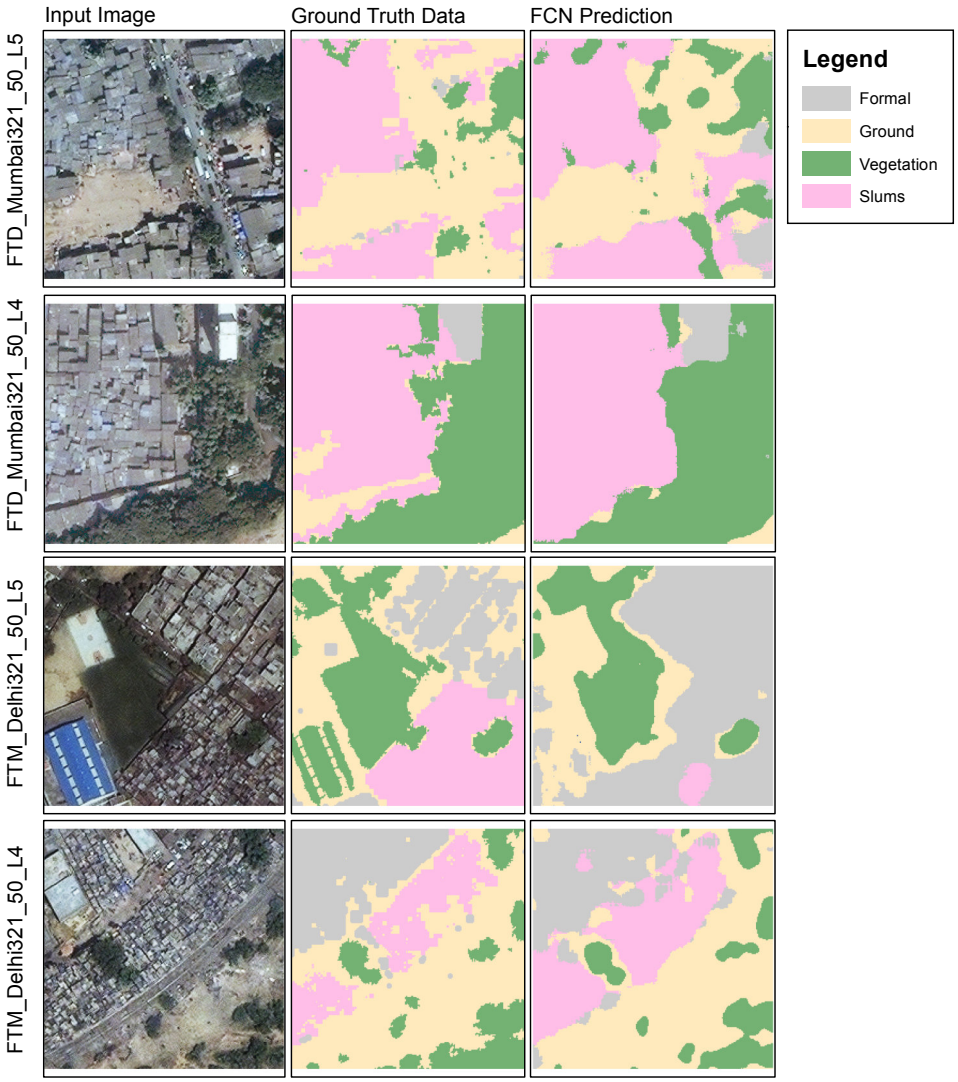


Figure 6.2: Comparative alignment of all fine-tuned FCNs.

### Applying transfer-learning to pre-trained FCNs

Using transfer-learning techniques to test the ability of FCNs to generalize to different cities, each city's dataset is fine-tuned to another dataset. As shown in section 5.1.2 fine-tuned FCNs perform with about 2% to 5% for all accuracy metrics slightly worse than pre-trained networks. A comparative alignment in figure 6.2 shows the transfer-learned FCNs still present respectable segmentation performance. With regard to informal settlements both FCNs trained on the Delhi ground truth data and transfer learned to Mumbai show that enough data on informal settlements should be present to counter the problem of an imbalanced class distribution. This effect can cause sever problems when training the FCNs the other way around where the segmentation in row 3 of figure 6.2 shows that the $FTM\_Delhi_{321}\_50\_L5$ could not identify the slum present in the ground truth data. The FCNs trained from the fourth layer of the FCN show better results when adapting the learning process to other cities. $FTM\_Delhi_{321}\_50\_L5$ shows the importance of layers for learning to detect informal settlements. While in $FTM\_Delhi_{321}\_50\_L5$ informal

settlements of the ground truth data are often missed, in $FTM\_Delhi_{321}\_50\_L4$ the classification performs better with about 10% for all accuracy metrics. Small differences in geomorphological structures between formal and informal buildings present a complex challenge when fine-tuning a FCN, while the other way around does not affect the FCN in the same manner. This reinforced the importance of having plenty and precise ground truth data especially when dealing with an imbalanced class distribution. Comparing the amount of slums in the ground truth data of 10% in Mumbai and only 2% in Delhi shows that this imbalance provides a limit below the 10% where class segmentation can be affected by low accuracy scores.

### 6.1.2 Analysis for large scale slum mapping using a FCN

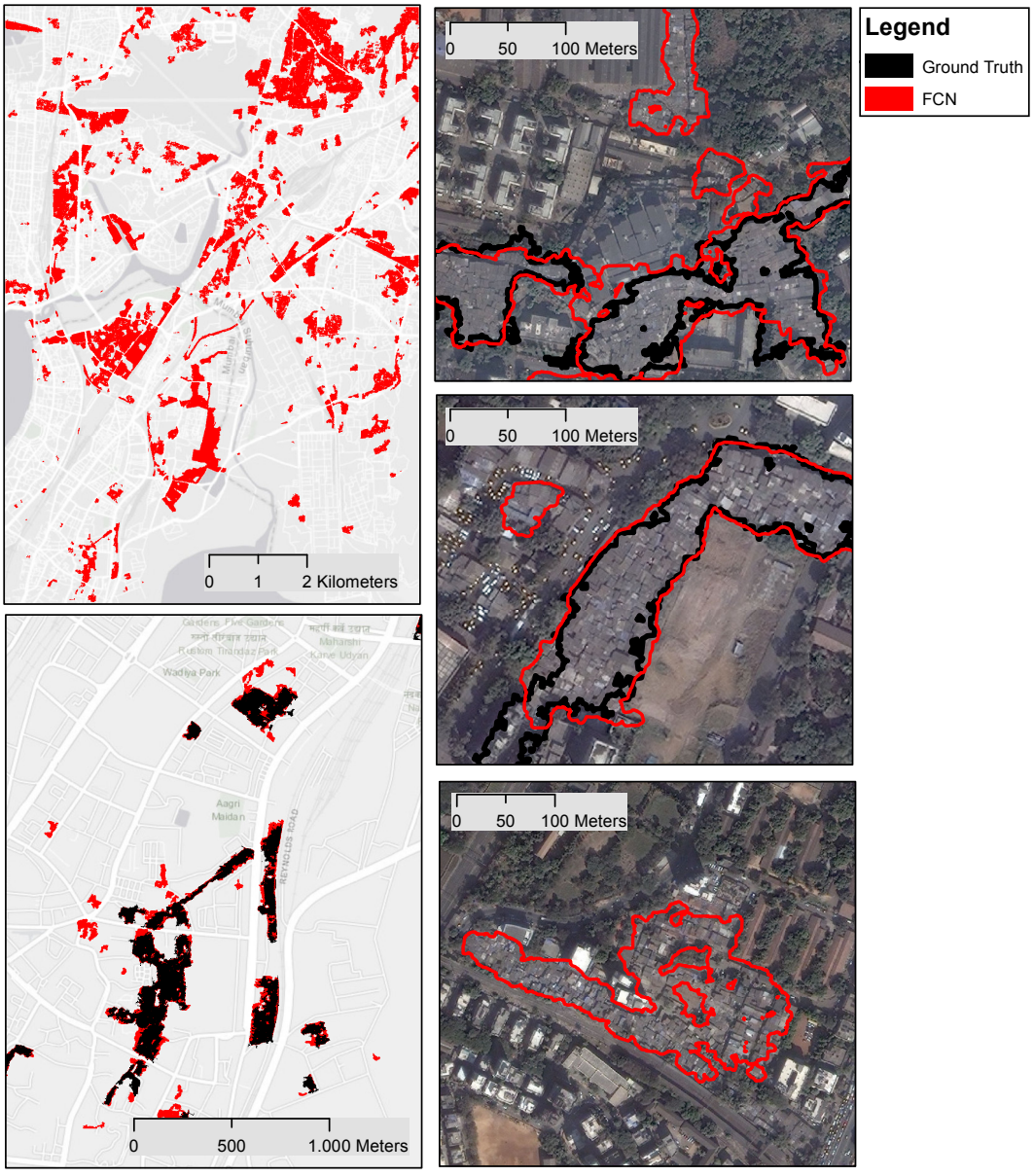


Figure 6.3: Comparison of informal settlements detect by the $FCN\_MD_{321}$ and slums present in the ground truth dataset in Mumbai.

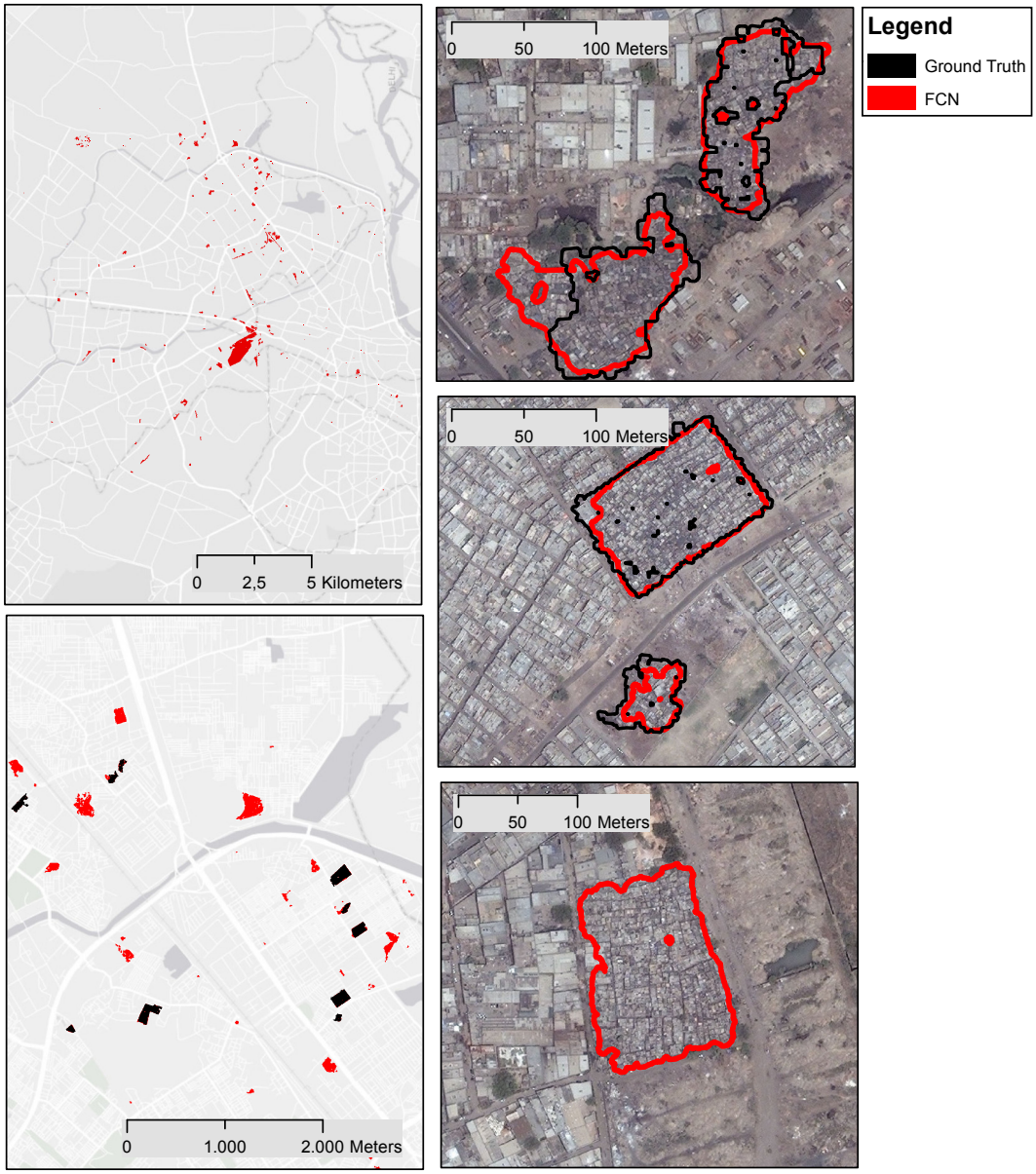


Figure 6.4: Comparison of informal settlements detect by the $FCN\_MD_{321}$ and slums present in the ground truth dataset in Delhi.

Figure 6.3 and 6.4 show the difference in detected informal settlements in Mumbai and Delhi in comparison to available slums in the ground truth data. The top left images show all detected slums in the classified image. The differences between Mumbai and Delhi are quite big. While in Mumbai slums take up much more space in Delhi informal settlements are smaller and not as dense. The median size of slums in Mumbai is with $5385m^2$ 606% higher than the size of slums in Delhi with $762m^2$. For both figures the bottom left image shows a subsection comparing the detected slums from the FCN in red with the slums present in the ground truth data in black. In figure 6.3 the subsection in Mumbai shows that although many slums are present in the ground truth dataset, some smaller patches of informal settlements could only be detected through the FCN. In figure 6.4 the bottom left subsection show that many informal settlements are missing in the ground truth dataset. On the right side of figure 6.3 and 6.4 three examples of a slum's

boundaries are presented. The FCN prediction is highlighted in red, while the ground truth dataset is labelled in black boundaries. In both instances the FCN is able to form better and more detailed borders around detected informal settlements.

### 6.1.3 Investment for a water pipeline infrastructure

The optimal water supply network for Mumbai and Delhi is a holistic approach to provide water for the poor. Following a multidisciplinary way of working, the investment is compared for a water pipe line infrastructure between the informal settlements in the ground truth dataset and the best performing FCN. A mathematical approach for an optimization of a shortest path along the existing street network is the basis for the cost functions to calculate the investment for different periods of operating time. Comparing the cost of investment for all three approaches for the two dataset the straightforward method of connecting all informal settlements with one unique water pipeline can be considered the cheapest solution. Using a more complicated hierarchical approach with multiple pipelines of variable pipeline diameters the cost increases. Water supply networks for Mumbai and Delhi can be seen in figure 6.5 and 6.6.

#### Mumbai water supply network

The ground truth dataset is more biased towards larger informal settlements. While the minimum size for slums in the reference data is roughly $1000m^2$ the FCN is capable of to identify even smaller slum patches of $470m^2$. Figure 5.4 shows these differences, where the FCN is more capable of slum mapping for smaller areas. This obviously enlarges the amount of informal settlements. This aspect alone makes the investment cost of the pipeline network more expensive. Figure 6.5 illustrates the results for a water supply network optimized for a shortest path for all three proposed approaches using different geodata. The first row represents the pipeline network using the reference data as input, while the bottom row calculates the water supply network using informal settlements mapped by the FCN. The first column shows some similarities with the networks approach. All slums are connected by their shortest path along the road infrastructure. The FCN could detected about 100 more informal settlements not contained in the ground truth data. This has a more branched appearance. The second row in figure 6.5 shows the hierarchical approach using two combined pipeline networks. While one network with a large diameter provides water for all slums through the largest settlements of a geospatial cluster, the water supply network for all slums of each cluster are connected through a shortest path, where pipes use smaller diameters. The second hierarchical approach shown in the third column connects all informal settlements with a separate network through the centre slums of each geospatial cluster. Most interestingly the networks using the same hierarchical approach show no similarities in the underlying network connecting all clusters. The shortest path connecting each cluster is thoroughly dependent on the clusters structure and location.

#### Delhi water supply network

For Delhi the difference in detected informal settlements is even greater than for the Mumbai dataset. As seen in figure 5.4 the ground truth dataset only contains 98 slums while the FCN could map 276. This increase in detected informal settlements has an effect on the water supply network infrastructure. Whereas the a network connecting all slums via a shortest path showed some similarities for the Mumbai dataset, the difference in connected slums for Delhi results in a completely new water pipeline network as seen in figure 6.6. With a pipeline length of 194km in the FCN dataset the length for the network connecting all slums via a shortest path is 123km longer than the 71km pipeline connecting all slums of the ground truth dataset. The second row in figure 6.6 shows the hierarchical approach using two combined pipeline networks. While one

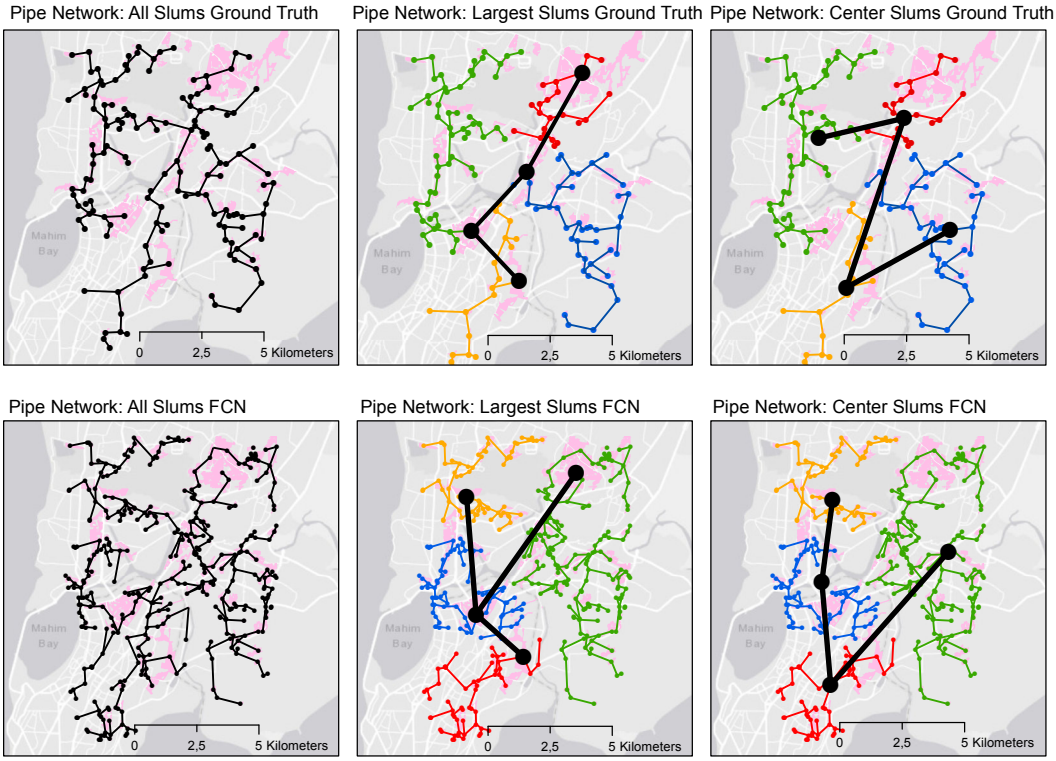


Figure 6.5: Comparative alignment of all water supply networks for Mumbai. The first row show results for a pipe network optimzed for the ground truth data, while the second row presents the rust for geodata aquired from a FCN.

network with a large diameter provides water for all slums through the largest settlements of a geospatial cluster, the water supply network for all slums of each cluster are connected through a shortest path, where pipes use smaller diameters. The second hierarchical approach shown in the third column connects all informal settlements with a separate network through the centre slums of each geospatial cluster.

## 6.2 Conclusion

With the ongoing trend of urbanization, the pressure to the cities of the world is growing. Huge urban areas with more than 10 million inhabitants are emerging. In the next two decades the number of such mega cities is predicted to increase to 41 [UnitedNations, 2014]. Alongside, cities face various challenges, like increasing usage of infrastructure, increasing demand for jobs or health risks. The growing trend of urbanization also affects the conditions of the living environment. Especially in developing countries with less possibilities to counteract these challenges, the urbanization leads to the development of illegal or informal settlements. Slums are predominantly located on unappealing or even polluted land, feature no durable housing and lack access to clean water. Until 2020 one and a half billion people worldwide will live under such sub-standard living conditions [Arimah, 2010]. These circumstances negatively affect the physical and psychological health of the slum dwellers [Snyder et al., 2014]. This led the United Nations to record their goals for sustainable development. One goal addresses the right of every human to access of water. To provide access to water, the United Nations formulated the need for analysing, mapping and monitoring the development of slums. The methodology of geographical remote sensing using satellite images enables land use / land cover mapping of large areas. Often only small areas of

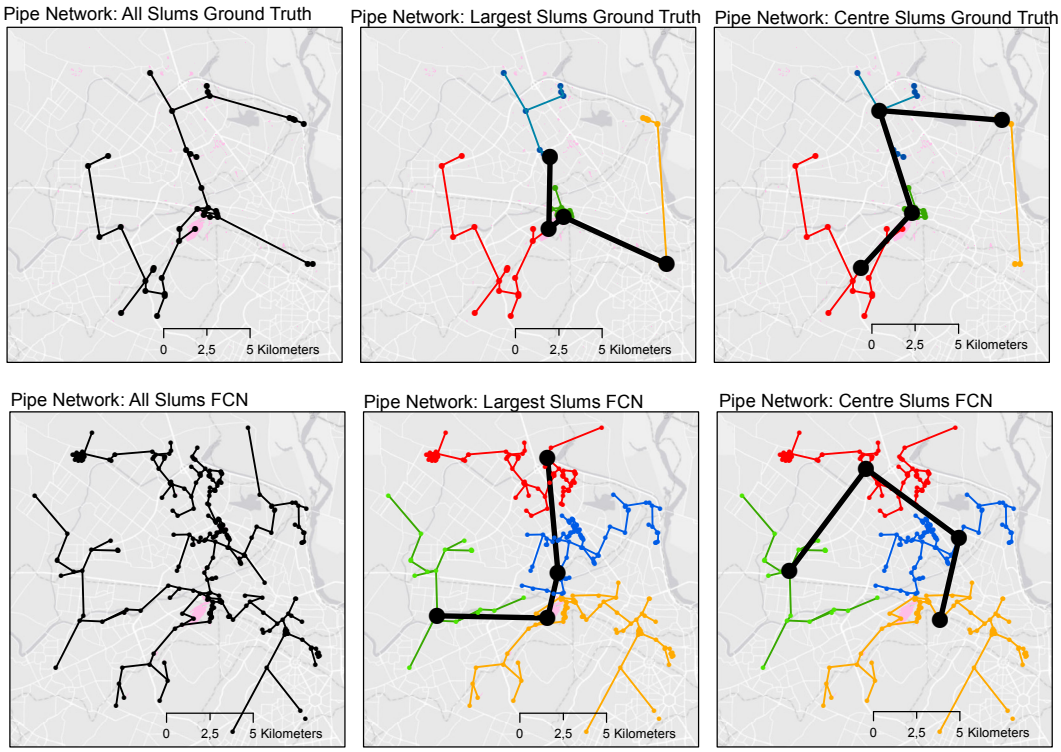


Figure 6.6: Comparative alignment of all water supply networks for Delhi. The first row show results for a pipe network optimzed for the ground truth data, while the second row presents the rust for geodata aquired from a FCN.

investigation were chosen for methodological development but few exhaustive city-wide mappings were conducted.

With the help of recent trends in deep learning, fully convolutional networks can provide valuable results for slum mapping. To investigate the capability of FCNs for extensive identification of informal settlements in urban areas, the mega cities Mumbai and Delhi were chosen as study areas. In order to train and validate the segmentation networks, an area wide reference dataset was created. This ground truth dataset is used for the training of multiple fully convolutional networks. In a broad experimental setup of mapping slums pre-trained FCNs are used for class segmentation of each dataset in Mumbai an Delhi. Overall Pixel Accuracy for each dataset achieves up to 88% for all land use / land cover classes and 77% for slums. Using fine-tuning techniques to study the FCNs ability for transferring learned knowlegde to different cities could achieve up to 75% Pixel Accuracy when a FCN is trained on one city and fine-tuned on another. With a mean Intersection over Union of up to 62% the FCN-vgg19 is very capable of accurately extracting boundaries of informal settlements from very high resolution optical data. Even though it is possible to classify slums in a large area, the conducted experiments strongly rely on an extensive reference dataset. This is especially present in the Delhi ground truth dataset, where only few informal settlements are covered in the reference data, which can lead to detecting slums not present in the ground truth dataset.

Analysing the extracted geographical data provided by the FCN's segmentation a fluid system with water pipes is determined. A mathematical optimization algorithm finds the shortest path connecting informal settlements to the road network. Using adapted cost functions the investment for a water supply network can be calculated. Modelling various networks a cost of 16 million € for

Mumbai and 14 million € for Delhi is necessary to supply all slum dwellers with enough clean water over a time span of ten years.

With the aim of mapping, analysing and monitoring slum areas in mega cities to provide water for the poor, further research could focus on transferring the results obtained in this study to even more test sites including other cities of other cultural regions towards a global dataset. Moreover, since the current approach is very dependent on sufficient available data of informal settlements, further optimization of an improved ground truth dataset could benefit segmentation results profoundly.

# Bibliography

Angel S, Sheppard S, Civco DL, Buckley R, Chabaeva A, Gitlin L, Kraley A, Parent J, Perlin M (2005) The dynamics of global urban expansion. World Bank, Transport and Urban Development Department Washington, DC.

Arias-Castro E, Donoho DL et al. (2009) Does median filtering truly preserve edges better than linear filtering? The Annals of Statistics, 37 (3): 1172–1206.

Arimah BC (2010) The face of urban poverty: Explaining the prevalence of slums in developing countries. Number 2010, 30. Working paper//World Institute for Development Economics Research.

Baud I, Kuffer M, Pfeffer K, Sliuzas R, Karuppannan S (2010) Understanding heterogeneity in metropolitan India: The added value of remote sensing data for analyzing sub-standard residential areas. International Journal of Applied Earth Observation and Geoinformation, 12 (5): 359–374.

Belgiu M, Drăguţ L (2016) Random forest in remote sensing: A review of applications and future directions. ISPRS Journal of Photogrammetry and Remote Sensing, 114: 24–31.

Breiman L (2001) Random forests. Machine learning, 45 (1): 5–32.

Brito P, Quintanilha J (2012) A literature review, 2001-2008, of classification methods and inner urban characteristics identified in multispectral remote sensing images. Proceedings of the 4th GEOBIA, : 586–591.

Burdett R, Rhode P (2010) Living in the urban age. In: Living in the endless city (pp. 8–43). Phaidon Verlag GmbH.

Castelluccio M, Poggi G, Sansone C, Verdoliva L (2015) Land use classification in remote sensing images by convolutional neural networks. arXiv preprint arXiv:1508.00092.

Chinmayi S, Madhavi R (2013) 80000 people/sq km even in plush towers. The Times of India.

Cohen J (1960) Kappa: coefficient of concordance. Educ. Psych. Measurement, 20 (37).

Congalton RG, Green K (2008) Assessing the accuracy of remotely sensed data: principles and practices. CRC press.

Csurka G, Larlus D, Perronnin F, Meylan F (2013) What is a good evaluation measure for semantic segmentation?. In: BMVC, 27: 2013.

Dang HTH (2017) A guide to receptive field arithmetic for Convolutional Neural Networks. https://medium.com/@nikasa1889/a-guide-to-receptive-field-arithmetic-for-convolutional-neural-networks-e0f514068807 (17.01.2018).

Dell'Acqua F, Stasolla M, Gamba P (2006) Unstructured human settlement mapping with SAR sensors. In: Geoscience and Remote Sensing Symposium, 2006. IGARSS 2006. IEEE International Conference on: 3619–3622.

Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L (2009) Imagenet: A large-scale hierarchical image database. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on: 248–255.

Dumoulin V, Visin F (2016) A guide to convolution arithmetic for deep learning. arXiv preprint arXiv:1603.07285.

Everingham M, Van Gool L, Williams CK, Winn J, Zisserman A (2010) The pascal visual object classes (voc) challenge. International journal of computer vision, 88 (2): 303–338.

Fernando V (2009) In the heart of Bombay: the Dharavi Slum. dialogs, proposals, stories for global citizenship.

Finkel RA, Bentley JL (1974) Quad trees a data structure for retrieval on composite keys. Acta informatica, 4 (1): 1–9.

Friesen J, Rausch L, Pelz PF (2017) Providing water for the poor-towards optimal water supply infrastructures for informal settlements by using remote sensing data. In: Urban Remote Sensing Event (JURSE), 2017 Joint: 1–4.

Giada S, De Groeve T, Ehrlich D, Soille P (2003) Information extraction from very high resolution satellite imagery over Lukole refugee camp, Tanzania. International Journal of Remote Sensing, 24 (22): 4251–4266.

Girshick R, Donahue J, Darrell T, Malik J (2016) Region-based convolutional networks for accurate object detection and segmentation. IEEE transactions on pattern analysis and machine intelligence, 38 (1): 142–158.

Glaeser E (2010) Triumph of the city (S. 352).

Graesser J, Cheriyadat A, Vatsavai RR, Chandola V, Long J, Bright E (2012) Image based characterization of formal and informal neighborhoods in an urban landscape. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 5 (4): 1164–1176.

Haralick RM, Shanmugam K et al. (1973) Textural features for image classification. IEEE Transactions on systems, man, and cybernetics, 3 (6): 610–621.

Harvey D (2013) Rebellische Städte. Suhrkamp Verlag.

Hollis L (2013) Cities are good for you: The genius of the metropolis. Bloomsbury Publishing USA.

Hu F, Xia GS, Hu J, Zhang L (2015) Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. Remote Sensing, 7 (11): 14680–14707.

Huang X, Liu H, Zhang L (2015) Spatiotemporal detection and analysis of urban villages in mega city regions of China using high-resolution remotely sensed imagery. IEEE Transactions on Geoscience and Remote Sensing, 53 (7): 3639–3657.

Jacobsen K, Büyüksalih G (2008) Topographic mapping from space. In: Proceedings of the 4th Workshop of EARSeL on Remote Sensing for Developing Countries/GISDECO, Istanbul, Turkey: 4–7.

Jensen JR, Cowen DC (1999) Remote sensing of urban/suburban infrastructure and socio-economic attributes. Photogrammetric engineering and remote sensing, 65: 611–622.

Kamal-Chaoui L, Robert A (2009) Competitive cities and climate change. OECD Regional Development Working Papers, 2009 (2): 1.

Kingma D, Adam BJ (2017) A Method for Stochastic Optimization. Cornell University Library. arXiv preprint arXiv:1412.6980.

Koriakine A, Saveliev E (2006) Wikimapia - we describe the world. www.wikimapia.org.

Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems: 1097–1105.

Kruskal JB (1956) On the shortest spanning subtree of a graph and the traveling salesman problem. Proceedings of the American Mathematical society, 7 (1): 48–50.

Kuffer M, Barros J, Sliuzas RV (2014) The development of a morphological unplanned settlement index using very-high-resolution (VHR) imagery. Computers, Environment and Urban Systems, 48: 138–152.

Kuffer M, Pfeffer K, Sliuzas R (2016) Slums from space—15 years of slum mapping using remote sensing. Remote Sensing, 8 (6): 455.

Körner M (2016) Neural Networks & Deep Learning - Image Understanding – Recent Trends in Machine Learning. Moodle TUM.

Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition: 3431–3440.

Maggiori E, Tarabalka Y, Charpiat G, Alliez P (2017) Can Semantic Labeling Methods Generalize to Any City? The Inria Aerial Image Labeling Benchmark.

Marchionni V, Cabral M, Amado C, Covas D (2015) Water supply infrastructure cost modelling. Procedia Engineering, 119: 168–173.

MathWorks (2005) Image Processing Toolbox User's Guide - Using Quadtree Decomposition. http://matlab.izmiran.ru/help/toolbox/images/enhanc12.html.

Mboga N, Persello C, Bergado JR, Stein A (2017) Detection of Informal Settlements from VHR Images Using Convolutional Neural Networks. Remote sensing, 9 (11): 1106.

McCulloch WS, Pitts W (1943) A logical calculus of the ideas immanent in nervous activity. The bulletin of mathematical biophysics, 5 (4): 115–133.

MMRDA (2008) Mumbai Urban Infrastructure Project. Mumbai Metropolitan Region Development Authority.

Mosley L (2013) A balanced approach to the multi-class imbalance problem.

Mountrakis G, Im J, Ogole C (2011) Support vector machines in remote sensing: A review. ISPRS Journal of Photogrammetry and Remote Sensing, 66 (3): 247–259.

Niebergall S, Loew A, Mauser W (2008) Integrative assessment of informal settlements using VHR remote sensing data—the Delhi case study. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 1 (3): 193–205.

Nielson M (2015) Neural Networks and Deep Leanring. Determination Press.

Nogueira K, Penatti OA, dos Santos JA (2017) Towards better exploiting convolutional neural networks for remote sensing scene classification. Pattern Recognition, 61: 539–556.

Office of the Registrar General & Census Commissioner I (2011) 2011 Census Data. Ministry of Home Affairs, Government of India.

Oquab M, Bottou L, Laptev I, Sivic J (2014) Learning and transferring mid-level image representations using convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition: 1717–1724.

Penatti OA, Nogueira K, dos Santos JA (2015) Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops: 44–51.

Persello C, Stein A (2017) Deep Fully Convolutional Networks for the Detection of Informal Settlements in VHR Images. IEEE geoscience and remote sensing letters, 14 (12): 2325–2329.

Rashmi N (2011) Mumbai, a land of opportunities. The Times of India.

Rausch L, Friesen J, Altherr LC, Meck M, Pelz PF (2018) A Holistic Concept to Design Optimal Water Supply Infrastructures for Informal Settlements Using Remote Sensing Data. Remote Sensing, 10 (2): 216.

Risbud N (2002) Policies for tenure security in Delhi. Holding Their Ground-Secure Land Tenure for the Urban Poor in Developing Countries; Durand-Lasserve, A., Royston, R., Eds, : 59–74.

Rosebrock A (2016) Image search image guide - Resource guide. pyimagesearch.

Rosenblatt F (1958) The perceptron: A probabilistic model for information storage and organization in the brain. Psychological review, 65 (6): 386.

Sasank C (2017) A 2017 Guide to Semantic Segmentation with Deep Learning. http://blog.qure.ai/notes/semantic-segmentation-deep-learning-review (14.01.2018).

Schneider A, Friedl MA, Potere D (2009) A new map of global urban extent from MODIS satellite data. Environmental Research Letters, 4 (4): 044003.

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

Shekkizhar S (2017) FCN.tensorflow. GitHub https://github.com/shekkizh/FCN.tensorflow.

Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Sivic J, Zisserman A (2003) Video Google: A text retrieval approach to object matching in videos. In: null: 1470.

Sliuzas R, Mboup G, de Sherbinin A (2008) Report of the expert group meeting on slum identification and mapping. Report by CIESIN, UN-Habitat, ITC, 36.

Snyder RE, Jaimes G, Riley LW, Faerstein E, Corburn J (2014) A comparison of social and spatial determinants of health between formal and informal settlements in a large metropolitan setting in Brazil. Journal of Urban Health, 91 (3): 432–445.

Spurk JH, Aksel N (2004) Strömungslehre, volume 5. Springer.

Subbaraman R, Nolan L, Shitole T, Sawant K, Shitole S, Sood K, Nanarkar M, Ghannam J, Betancourt TS, Bloom DE et al. (2014) The psychological toll of slum living in Mumbai, India: A mixed methods study. Social Science & Medicine, 119: 155–169.

Sule S (2003) Understanding our civiv issues - Mumbai's water supply. Bombay Community Public Trust.

Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition: 1–9.

Taubenböck H, Kraff N (2014) The physical face of slums: a structural comparison of slums in Mumbai, India, based on remotely sensed data. Journal of Housing and the Built Environment, 29 (1): 15–38.

Taubenböck H, Kraff N (2015) Das globale Gesicht urbaner Armut? Siedlungsstrukturen in Slums. In: Globale Urbanisierung (pp. 107–119). Springer.

Taubenböck H, Wurm M (2015a) Globale Urbanisierung–Markenzeichen des 21. Jahrhunderts. In: Globale Urbanisierung (pp. 5–10). Springer.

Taubenböck H, Wurm M (2015b) Ich weiß, dass ich nichts weiß–Bevölkerungsschätzung in der Megacity Mumbai. In: Globale Urbanisierung (pp. 171–178). Springer.

Trimble (2014) Trimble eCognition Developer 9.0 Reference Book. Trimble Germany GmbH, TrimbleGermany GmbH, Arnulfstrasse126, D-80636Munich, Germany, document version 9.0.3 edition.

UnitedNations H (2009) Slum Upgrading Facility. Land and Slum Upgrading.

UnitedNations H (2011) World urbanization prospects, the 2011 revision. Final Report with Annex Tables. New York, NY: United Nations Department of Economic and Social Affairs.

UnitedNations H (2014) World urbanization prospects: The 2014 revision, highlights. department of economic and social affairs. Population Division, United Nations.

UnitedNations H (2015a) Habitat III Issue Paper 22 Informal Settlements. New York, NY: United Nations, : 1–8.

UnitedNations H (2015b) United Nations Millennium Development Goals.

UnitedNations H (2016) World Cities Report. UN-Habitat.

Vaz LF, Berenstein J (2004) Morphological diversity in the squatter settlements of Rio de Janeiro. Suburban form: An international perspective, : 61–72.

Weiss GM (2004) Mining with rarity: a unifying framework. ACM Sigkdd Explorations Newsletter, 6 (1): 7–19.

Williams DP, Myers V, Silvious MS (2009) Mine classification with imbalanced data. IEEE Geoscience and Remote Sensing Letters, 6 (3): 528–532.

Wright C, Gallant A (2007) Improved wetland remote sensing in Yellowstone National Park using classification trees to combine TM imagery and ancillary environmental data. Remote Sensing of Environment, 107 (4): 582–605.

Wurm M, Taubenböck H (2018) Detecting social groups from space–Assessment of remote sensing-based mapped morphological slums using income data. Remote Sensing Letters, 9 (1): 41–50.

Wurm M, Taubenböck H, Weigand M, Schmitt A (2017) Slum mapping in polarimetric SAR data using spatial features. Remote Sensing of Environment, 194: 190–204.

Yu F, Koltun V (2015) Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122.

Zhu Q, Zhong Y, Zhao B, Xia GS, Zhang L (2016) Bag-of-visual-words scene classifier with local and global features for high spatial resolution remote sensing imagery. IEEE Geoscience and Remote Sensing Letters, 13 (6): 747–751.

Zhu XX, Tuia D, Mou L, Xia GS, Zhang L, Xu F, Fraundorfer F (2017) Deep Learning in Remote Sensing: A Comprehensive Review and List of Resources. IEEE Geoscience and Remote Sensing Magazine, 5 (4): 8–36.

Zou Q, Ni L, Zhang T, Wang Q (2015) Deep learning based feature selection for remote sensing scene classification. IEEE Geoscience and Remote Sensing Letters, 12 (11): 2321–2325.

# Acknowledgment

It is a pleasure to thank those who made this thesis possible.

I would like to show my gratitude to Prof. Dr.-Ing. Uwe Stilla who made this unique cooperation possible. I also would like to thank him for his constructive criticism and support.

I would like to express my special appreciation and thanks to my supervisor Dr. Michael Wurm who supported me during the thesis. In numerous discussions I experienced his continuous encouragement and constructive criticism which contributed to the success of this thesis.

Furthermore, I am very grateful for the help of Julian Zeidler and especially Adam Fathalrahman for letting me use the deep learning infrastructure in their department.

This study was compiled in cooperation with the Deutsches Fernerkundungsdatenzentrum (DFD, German Remote Sensing Data Center) at the Deutsches Zentrum für Luft- und Raumfahrt (DLR, German Aerospace Center) in Oberpfaffenhofen, Germany.

I am very grateful for the support, encouragement and revisions of Nina Stark and special thanks to my family for their emotional support and during the last few years.

Finally, I want to thank Angela Peron for cheering me up and being there for me during this challenging endeavour, without you this wouldn't have been possible.

# Eidesstattliche Erklärung

Eidesstattliche Erklärung

Familienname: Stark
Vorname: Thomas
Geburtsdatum: 31.01.1987

Ich erkläre hiermit an Eides statt, dass ich die vorliegende Arbeit eigenständig ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen direkt oder indirekt übernommenen Daten, Konzepte und anderen Inhalte sind unter Angabe des Literaturzitats gekennzeichnet. Ich weiß, dass die Arbeit in digitalisierter Form daraufhin überprüft werden kann, ob unerlaubte Hilfsmittel verwendet wurden und ob es sich – insgesamt oder in Teilen – um ein Plagiat handelt. Zum Vergleich meiner Arbeit mit existierenden Quelle darf sie in eine Datenbank eingestellt werden und nach der Überprüfung zum Vergleich mit künftig eingehenden Arbeiten dort verbleiben. Weitere Vervielfältigungs- und Verwertungsrechte werden dadurch nicht eingeräumt. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form einer anderen Prüfungsbehörde vorgelegt, war bisher nicht Bestandteil einer Studien- oder Prüfungsleistung und ist noch nicht veröffentlicht.

Ort, Datum

Unterschrift