Janne Mäyrä

Land cover classification from multispectral data using convolutional autoencoder networks

Master's Thesis in Information Technology

November 15, 2018

University of Jyväskylä

Faculty of Information Technology

Author: Janne Mäyrä

Contact information: `janne.e.o.mayra@student.jyu.fi`

Supervisors: Ilkka Pölönen, and Laura Uusitalo(SYKE)

Title: Land cover classification from multispectral data using convolutional autoencoder networks

Työn nimi: Maanpeiteluokittelu monispektridatasta konvolutiviisilla autoenkooderineuroverkoilla

Project: Master's Thesis

Study line: Computational Sciences

Page count: 77+7

Abstract: Since the mid 2000's, deep learning has received much attention and today its applications are almost everywhere. Around the same timespan the amount of freely available satellite data has grown, especially after Sentinel-2 missions started. This data has a lot of remote sensing applications, but the amount of produced data is practically impossible for humans to analyze or process. This thesis tested the viability of U-Net, a well-known neural network architecture, in land cover classification from multispectral satellite images to different classification levels in the Rakkolanjoki river drainage basin area. Classification results from only visible light bandwidths, all Sentinel-2 bands, precomputed spectral indices and all available features were compared, and best results were achieved with all available features. Even with next to none fine-tuning and short training time, implemented version of U-Net managed to accurately classify over 90% of the pixels for the easiest classification level (CORINE land cover level 1), and around 75% for the hardest level. Produced segmentation maps were also visually observed and compared to both ground truth labels and RGB-composites of the satellite image. As as conclusion, U-Net is a viable baseline for the needs of Finnish Environment Institute, and will later be developed further.

Keywords: Land cover classification, machine learning, neural networks, remote sensing

Suomenkielinen tiivistelmä: Syväoppiminen saanut paljon huomiota 2000-luvun puolivälistä alkaen, ja tänä päivänä sen sovelluksia on lähes kaikkialla. Samalla aikavälillä avoimen satelliittikuvadatan määrä on kasvanut, erityisesti Sentinel-2 satelliittien laukaisujen jälkeen. Tätä dataa voidaan hyödyntää useissa kaukokartoitussovellutuksissa, mutta tämän datamäärän analysointi ja käsittely on ihmisille käytännössä mahdotonta. Tässä tutkielmassa testattiin erään tunnetun neuroverkkoarkkitehtuurin, U-Netin, suorituskykyä Rakkolanjoen valuma-alueen maanpeiteluokittelussa monispektrisatelliittikuvista eri luokittelutarkkuuksille. Eri lähtodatoilla saatuja luokittelutarkkuuksia vertailtiin keskenään, ja parhaat luokittelutulokset saatiin hyödyntämällä sekä kaikkea Sentinel-2 dataa että erikseen laskettuja spektri-indeksejä. Huolimatta lähes olemattomasta verkkojen hienosäädöstä ja lyhyestä koulutusajasta saadut luokittelutulokset ovat varsin lupaavia helpoimman luokittelutason (CORINE land cover taso 1) tarkkuuden ollessa yli 90% ja haastavimmallakin yli 75%. Tuotettuja maanpeitekarttoja vertailtiin myös visuaalisesti sekä lähtötietoihin että satelliittikuviin. Johtopäätöksenä voidaan todeta, että U-Net on käyttökelpoinen malli Suomen Ympäristökeskuksen tarpeisiin, ja kehitettyä mallia tullaan jatkokehittämään edelleen.

Avainsanat: Maanpeiteluokittelu, koneoppiminen, neuroverkot, kaukokartoitus

Glossary

CLC	CORINE Land Cover
CNN	Convolutional Neural network
CORINE	”Coordination of information on the environment” programme coordinated by European Union
ESA	European Space Agency
HSI	Hyperspectral imaging
JI	Jaccard index, intersection over union
LiDAR	Light Detection and Ranging
LULC	Land use and land cover
MSI	Multispectral imaging
NDBI	Normalized Difference Built-up Index $NDBI = \frac{SWIR-NIR}{SWIR+NIR}$
NDMI	Normalized Difference Moisture Index $NDMI = \frac{NIR-SWIR1.5}{NIR+SWIR1.5}$
NDSI	Normalized Difference Snow Index, $NDSI = \frac{GREEN-SWIR1.5}{GREEN+SWIR1.5}$
NDTI	Normalized Difference Tillage Index $NDTI = \frac{SWIR1.5-SWIR2.0}{SWIR1.5+SWIR2.0}$
NDVI	Normalized Difference Vegetation Index $NDVI = \frac{RED-NIR}{RED+NIR}$
NIR	Near-infrared
ReLU	Rectified linear unit
ROI	Region of interest
SAR	Synthetic Aperture Radar
SYKE	Suomen ympäristökeskus, Finnish Environment Institute
SWIR	Short-wave infrared
VNIR	Visible light and near infrared

List of Figures

List of Tables

Contents

1 Introduction

This chapter gives an overview about the thesis topic, present the research problem and provide some use cases for remote sensing data. This chapter also briefly presents some recent results related to this thesis and describes the overall structure of this thesis.

Historically one of the biggest challenges in remote sensing has been the limited availability of viable data and lack of both storage space and computing power to analyze such data. This has changed during the last few years, and for instance European Space Agency provides pre-processed data collected with Sentinel-2 satellites. These satellites have temporal resolution of five days, totalling over 70 images each year from almost any location in the world. However, the amount of data has grown so much that it is impossible for humans to process and interpret even a fraction of it (Lillesand, Kiefer, and Chipman 2008; European Space Agency 2018c). Imagine trying to visually interpret all of Europe from images with spatial resolution of 10×10m, and add the fact that some land use and land cover classes are very hard to distinguish from each other. Lucky for us, different machine learning methods are viable for interpreting these images, and during the last few years deep learning has been applied to remote sensing tasks. These methods, unlike our eyes, can process and classify multi- and hyperspectral data and thus gain better results.

Artificial intelligence and deep learning have received a lot of hype in the past few years. Even though deep learning dates back in 1940s, it didn't have much attention before mid 2000s. First commercial applications of deep learning were presented in 1990s, but they were not considered to be a viable technology. Because computing power was much lower than nowadays and available datasets were too small for easy training, managing to successfully train a neural network required specialized expertise. During the mid 2000s a few breakthroughs in deep learning research reignited the interest, and neural network implementations started to beat classical machine learning methods in several artificial intelligence tasks. After AlexNet dominated ImageNet competition in 2012, interest in deep learning and convolutional neural networks started growing rapidly, and now their applications are everywhere (Goodfellow, Bengio, and Courville 2016; Krizhevsky, Sutskever, and Hinton 2012).

It is of course possible to perform land use and land cover classification with classical machine learning methods like K nearest neighbors or support vector machines, using each pixel's spectral signature as features. Unlike deep learning methods which are more or less ”black boxes”, it's possible to find out which features affect classifications the most. However, these methods don't necessarily utilize the spatial information of each location, such as that water pixels are typically surrounded by other water pixels rather than for instance broad-leaved forest. It is possible to modify these to use both spectral and spatial information and thus improve classification results (Fauvel et al. 2013). Even though classical machine learning methods are nowadays still used in land cover classification tasks, in the past few years deep learning has started to become the go-to choice due to their success in other visual recognition tasks. Especially previously mentioned convolutional neural networks have gained attention because of their natural ability to classify images and automatically detect hierarchical features from input data (Li, Zhang, and Shen 2017; Nogueira, Penatti, and Santos 2017).

While land cover classification could be performed like typical image classification, so that a centroid pixel of one smaller image is classified, this has same problems than pixel-by-pixel classification described earlier. Recently developed autoencoder-type networks, such as U-Net and SegNet solve these problems by classifying whole image at once (Ronneberger, Fischer, and Brox 2015; Badrinarayanan, Kendall, and Cipolla 2015). This thesis test the viability of U-Net, a network originally produced for grayscale medical image segmentation, in multispectral land cover classification.

Aside from land cover classification and change monitoring, large scale remote sensing data has its uses in other areas. Data from agricultural areas can be used to both classify different crops grown there and estimate the crop yield from these fields, and forestry has similar applications. Perhaps the area with most new uses for remote sensing data is ecosystems management, in which the land cover classification belongs. It's possible to estimate biodiversity of different areas partly based on satellite images, due to certain vegetation covers being often associated with high biodiversity and vice versa. As a last example, remote sensing data is used to estimate damages of some natural disaster, like heavy thunderstorm or wildfire (Jones and Vaughan 2010). For example, the route of a heavy thunderstorm that occurred on the end of July 2010 can be detected by comparing satellite images taken before and after the event (Härmä et al. 2014).

I received the topic for this thesis by coincidence. In March 2018, after traditional ice-fishing contest of SYKE Jyväskylä (which yours truly won overwhelmingly by catching both the most and the biggest fish) and the following post-competition dinner, one researcher asked me about the status of my studies. When I responded that I might be in a need of a master's thesis topic, possibly from the field of optimization, machine learning or data analysis, the response was practically ”I might have one for you”. Fast forward about three weeks and I was presented with broader topic ”machine learning methods in land cover classification”, which later was fined down into the current one. At that point, I had next to none experience in deep learning, remote sensing or spectral imaging. I started the thesis process on July 2018, and original goal was to finish it during autumn 2018. From July to September, I was employed full-time by SYKE as as thesis worker, and after that I finished this thesis as an external researcher.

1.1 Research problem

According to Lillesand, Kiefer, and Chipman (2008, 48-51), successful application of remote sensing involves at least the following steps:

1. Clear definition of a problem at hand
2. Evaluation of the potential for addressing the problem with remote sensing techniques
3. Identification of the remote sensing data acquisition procedures appropriate to the task
4. Determination of the data interpretation procedures to be employed and the reference data needed
5. Identification of the criteria by which the quality of information collected can be judged

First of all: do we have a clear definition of a problem? Finnish Environment Institute (SYKE) has at least two practical needs for land cover classification: land cover classification for Finnish Corine Land Cover 2018 project and acquiring suitable land cover classification in Rakkolanjoki river drainage basin area in order to calculate nutrient loads. While it would be unreasonable to provide perfect classifier for these (and all future) needs in one master's thesis (or even in one dissertation), at least proof-of-concept or simple baseline can be expected to be acquired and later expanded.

Data acquisition is not a problem any more, because since the launch of Sentinel-2A satellite in 2015, and more so after Sentinel-2B was launched in 2017, European Space Agency (ESA) has provided high-resolution multispectral data in regular intervals for open use. By having a method to interpret this data regularly it is possible to monitor land cover change and agriculture effectively, along with other possible remote sensing applications.

Identification of the remote sensing data and determination of the interpretation procedures are the topics for sections 3.1 and 3.2 respectively, whereas chapter 4 will be focused on judgement of criteria and quality of information collected.

To sum it up, the following research questions can be raised:

1. Does using more features (spectral bands, spectral indices) improve classification results?
2. Are autoencoder networks trained from scratch viable classifiers for Sentinel-2 data?
3. Is the quality of classifications good enough for remote sensing applications, such as nutrition load calculations?
4. Which land cover classes can be accurately classified?

In order to answer these questions, this thesis presents a workflow and method for land cover classification with convolutional neural networks using multispectral data as an input. Additionally, classification results for Rakkolanjoki drainage basin area are produced for multiple different classification levels. Usability of this method for other areas will be evaluated with classification results from region of interest.

Convolutional autoencoder-type networks have successfully been applied to land use and land cover classification. However, many of these implementations use only RGB-images or at most 4 or 5 channel images rather than full Sentinel-2 spectrum. One objective for this thesis is to present a method usable with multispectral satellite data utilizing multiple spectral bands and possibly spectral indices, especially from Sentinel-2 data. Kemker, Salvaggio, and

Kanan (2018) present one method and dataset for multispectral remote sensing, but scale and usage of their data is very different than needed in environmental monitoring. Also, neural network architecture used here is different than used by both Li, Zhang, and Shen (2017) and Ji et al. (2018).

As for the last two questions, the third one needs to be answered by another expert, as I am not nearly competent enough to answer that question. Thankfully, several of those are employed by SYKE, and feedback from them was collected. The fourth question can be answered by observing both the numerical results, such as metrics and confusion matrices, but also by looking at satellite images and comparing results to them.

2 Theory

This chapter is structured as follows: First section is a short primer for spectral imaging, with main focus being on multispectral imaging. Remote sensing and its applications are introduced briefly in section 2.2, with land use and land cover classification being introduced more thoroughly. Finally, in section 2.3 basic theory about neural networks and more specifically convolutional neural networks is presented. Note that section 2.3 is written under the assumption that the reader has basic knowledge of machine learning, but is not necessarily familiar with neural networks. Overall, this chapter is meant to give the reader such an understanding that methods used in this thesis can be followed through.

Main sources for sections 2.1 and 2.2 are ”Remote sensing and image interpretation” (Lillesand, Kiefer, and Chipman 2008) and ”Remote sensing of vegetation” (Jones and Vaughan 2010). Theory background for section 2.3 is mainly based on ”Deep learning book” (Goodfellow, Bengio, and Courville 2016) and lectures from Stanford University School of Engineering course ”CS231n: Convolutional neural networks for visual recognition” (Li, Johnson, and Yeung 2018). These are supplemented by recent scientific articles and research papers related spectral imaging, remote sensing and modern convolutional neural networks.

2.1 Spectral imaging

Like the name suggests, spectral imaging is a combination of spectroscopy and imaging. While imaging is the science of acquiring both spatial and temporal data from objects in the form of images, and spectroscopy is the science of observing electromagnetic spectrum (Figure 1), spectral imaging is simply acquiring and observing data with spectral pixel values. ”Normal” images, like everyday photographs, are composed of three bands from the visible light area of the electromagnetic spectrum, namely red, green and blue bands. So, for RGB images, each pixel has only three separate intensity values, but spectral images can have tens or even hundreds different values for each pixel, depending on the spectral resolution of the image. Perhaps the easiest way to visualize spectral images is as a three dimensional datacube, as seen in figure 2. This cube can be interpreted as several images taken from a certain wavelength, or a collection of pixels that each have their own spectra (Garini, Young, and McNamara 2006).

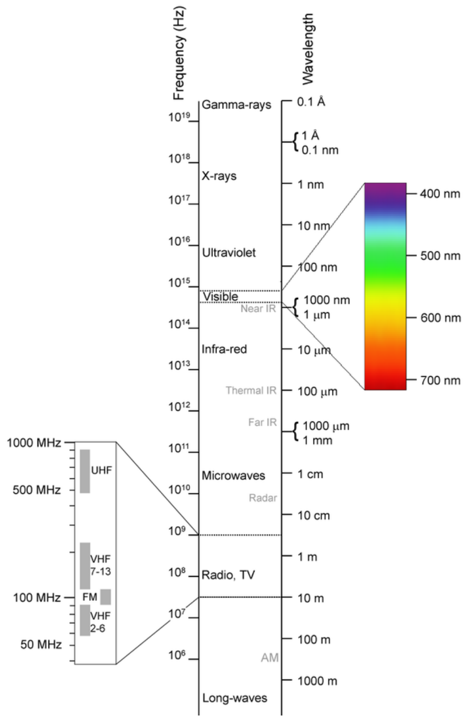


Figure 1. Electromagnetic spectrum (Commons 2018)

All surfaces both absorb and reflect electromagnetic radiation. If you have a blue coffee mug, it absorbs red and green wavelengths of the visible light part of the spectrum, and reflects blue wavelengths, thus making it look blue to you. Likewise, objects colored black absorb all of the visible (white) light and white objects reflect everything. All colors can be thought as a combination of the three main colors, that correspond to the three main bands from visible light: red, green and blue. However, if previously mentioned blue mug were to be put into a green light, it would look black, because there is no wavelength to reflect. Human eye has a receptor for each of these colors, and thus humans are able to differentiate between these colors and their combinations (Jones and Vaughan 2010).

The motivation for observing other wavelengths than those in the area visible to a human eye is that they usually hold valuable information about the object. Different materials may look the same to a human observer, such as artificial turf and natural grass both being green. If ones task would be to differentiate them from an aerial photograph, it would be almost

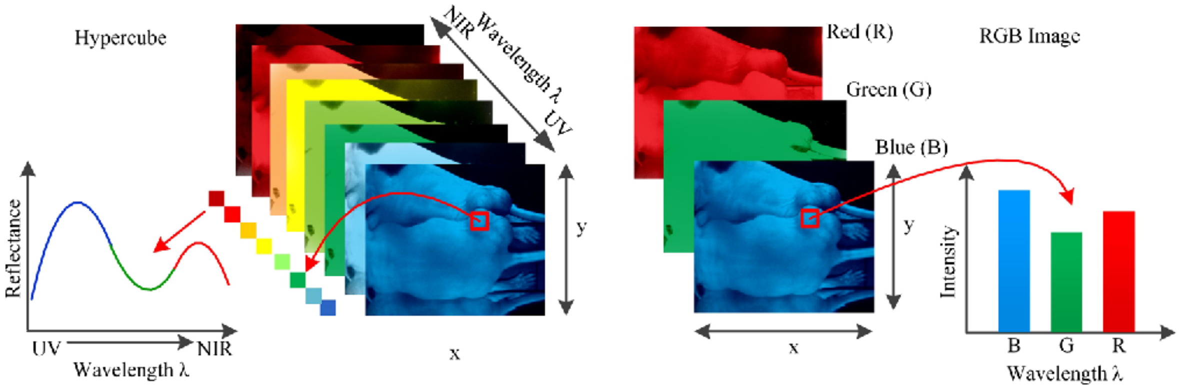


Figure 2. Comparison between spectral image (left) and rgb image (right) (Lua and Feia 2014)

impossible. However, their *spectral signature* is completely different, because only natural grass contains chlorophyll and thus reflects heavily in near infrared band (approximately 700nm, so called red edge). By having access to just one additional band, otherwise invisible to human eye, this task becomes trivial (Lillesand, Kiefer, and Chipman 2008).

Processing and analyzing images with more than three separate bands is called either *multispectral imaging* (MSI) or *hyperspectral imaging* (HSI). Electromagnetic spectrum can be thought as a collection of different bands, that in turn are a collection of different wavelengths within the bandwidth. If a band has a central wavelength of 560nm (green light) and a bandwidth of 45nm, then all wavelengths between 515nm and 605nm would belong to that band. MSI differs from HSI mainly in the bandwidth of acquired data and in the number of bands recorded. One definition for these terms is that multispectral imagers record more than three separate spectral bands, superspectral imagers record more than ten, and hyperspectral imagers record more than fifty. Because of that, hyperspectral bands have very low bandwidth, even as low as 0.1nm. This high spectral resolution enables construction of practically continuous reflectances from hyperspectral data. (Lillesand, Kiefer, and Chipman 2008; Jones and Vaughan 2010). According to Goetz (2009), bands being contiguous is the defining factor that separates HSI from MSI, not the number of spectral bands. Continuous data guarantees, that any possibly vital information is not lost.

For regular images, the resolution of the image typically means the *spatial resolution*, which is the minimum size of a distinguishable object. Typically spatial resolution is equal to pixel size. For spectral images, especially ones that are gathered from earth-orbiting satellites, resolution can also mean other things. *Spectral resolution* is the smallest difference between distinguishable wavelengths, and *temporal resolution* is the temporal difference between images (Jones and Vaughan 2010).

2.2 Remote sensing

Remote sensing means acquiring information about an object by analyzing data gathered with some external device that is not in contact with the object under investigation. Remote sensing as a term was first coined in 1960, meaning that the field is still rather new in the sense of research of it. Originally remote sensing referenced to observing an object without touching it, but it has since changed to be almost exclusively used related to earth observation. Nowadays it is most frequently performed by interpreting optical imagery, for example aerial photographs or satellite imagery. Data can also be acquired via microwave or LiDAR (*light detection and ranging*) sensing. Due to the topic of and methods used in this thesis, this section mostly focuses on multi- and hyperspectral sensing (Lillesand, Kiefer, and Chipman 2008; Jones and Vaughan 2010).

At first the only source for remote sensing data were aerial photographs, which were first taken from hot-air balloons and later, after airplanes had been invented, from planes. After that, cameras first moved to rockets, then to ballistic missiles and later to satellites. First satellite images of the earth were taken in around 1960s, and in 1972 ERTS-1, later known as Landsat-1, was launched. Landsat-1 was the first satellite solely designed to observe earth, and because it was also equipped with multispectral scanner, although only a four band one, it also started the age of spectral data in remote sensing. Interpretation of remote sensing data has also changed during the years. When only aerial photographs and such were available, only way to interpret images was naturally to look at them. Later, when digital images and more powerful computers became widespread, machine learning (K-means, Gaussian maximum likelihood etc) and later deep learning methods started to be used for different classification tasks (Goetz 2009; Lillesand, Kiefer, and Chipman 2008).

The advantage of using spectral data is the addition of infrared spectrum. Different surfaces, such as vegetation, water or barren soil each have their own distinct characteristic absorbance. For instance, vegetation can be detected by sharp increase of reflectance around the so called red-edge (700nm) between red and near infrared (NIR) bands of spectrum. (Jones and Vaughan 2010). This can be observed in figure 3, which contains two images of same agricultural area. Left one is composed of red, green and blue Sentinel-2 bands, and right one of NIR, red and green bands. In NIR-red-green -image most of the areas are colored red, because vegetation reflects this band heavily. Likewise, barren soil in the middle of the image doesn't reflect infrared, so it is colored green in false-color infrared image.

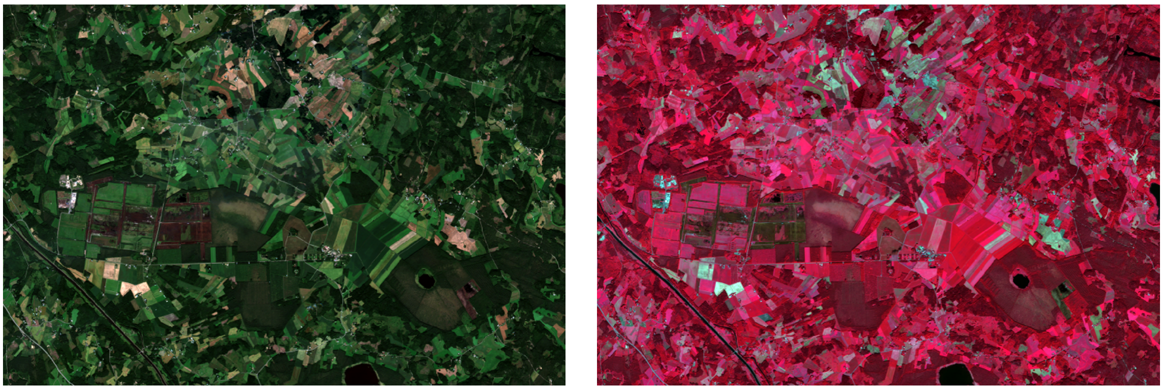


Figure 3. Left: RGB-image composed of red, green and blue bands. Right: False-color infrared -image composed of NIR, red and green bands (Copernicus Sentinel Data[2018], processing: SYKE)

These kinds of reflectance differences enable the derivation of spectral indices from two (or sometimes more) spectral bands. One of the most commonly used index is *Normalized Difference Vegetation Index* (NDVI), which is derived from NIR and red bands as shown in formula 2.1. This index has the advantages of being well-behaved, normalized and having values between -1 and 1, with values less than zero occurring in cloud, water or snowy surfaces. It also has the advantage of greatly simplifying data. Instead of inspecting whole spectrum, relevant information can be acquired from only one set of variables (Jones and Vaughan 2010). Several different spectral indices can be computed similarly, and in this thesis a total of five different indices are used. They are presented in section 3.1.

$$NDVI = \frac{NIR - RED}{NIR + RED} \tag{2.1}$$

2.2.1 Land use and land cover classification

One of the subfields of remote sensing is land use and land cover mapping (LULC). *Land cover* means the type of the feature covering the surface of the earth, whereas *land use* means the types of human activity. For example, lakes, fields or highways are types of land cover, and housing or agriculture are types of land use. Having knowledge about land use and land cover is essential for planning different activities (Lillesand, Kiefer, and Chipman 2008). For instance, original objective for this thesis was to perform land cover classification for Rakkolanjoki river drainage basin area. This information would be used to calculate nutrient loads from land areas affecting Rakkolanjoki river.

In remote sensing, typical objective of image classification is not to find out whether the image is a picture of a lake, but rather segment each pixel of the image into one predetermined land cover classes - precisely what LULC is. Each pixel has their own spectral pattern, which is the collection of reflectance measurements in available wavelengths. Image classification can be performed using only this pattern, so that pixels with similar patterns belong to the same class. However, pixels also have their own spatial pattern and temporal pattern, latter in case there are multiple measurements from different dates. Spatial pattern recognition utilizes the information from the neighboring pixels to classify one. For instance, if a pixel is surrounded by other pixels that have the spectral pattern of water, it is highly likely that the centroid is also water. Temporal pattern recognition, on the other hand, utilizes observations from different dates to classify. Disregarding snow, one way to distinguish coniferous and deciduous trees from each other could be, that coniferous trees are green all year, whereas deciduous trees shed their leaves on winter (Lillesand, Kiefer, and Chipman 2008).

LULC methods utilize multiple pattern recognition techniques rather than only one. For instance, previous example about coniferous and deciduous trees is almost impossible with just temporal pattern, but when combined with spectral pattern it becomes rather trivial. Historically most common combination has been spectral and spatial, but final combination depends on available data and other resources (Lillesand, Kiefer, and Chipman 2008).

Most often LULC is performed as *hard classification*, meaning that each pixel can have only one label. However, because spatial resolution for large-scale remote sensing data is typically so coarse that in reality each pixel contains multiple different land cover classes. These so called *mixed pixels* are often a source of classification error, because their spectral signature can more closely resemble different classes. This can be solved by either performing *soft classification* or *subpixel classification*. Soft classification assigns each pixel a probability value for belonging in certain class and subpixel classification outputs the fraction of each label in corresponding pixel (Jones and Vaughan 2010). Even though approach utilized in this thesis outputs probability distribution for each pixel, all segmentation maps are produced with hard classification method.

The most significant LULC project in Europe is CORINE (Coordination of information on the environment). CORINE programme was initialized by European Environment Agency in 1985. The three main objectives of the programme are

- to compile information on the state of the environment with regard to certain topics which have priority for all the Member States of the Community
- to coordinate the compilation of data and the organization of information within the Member States or at international level
- to ensure that information is consistent and that data are compatible

CORINE land cover project (CLC) is one part of the CORINE programme. Previously, there has been four CLC projects, and CLC 2018 is now active. Spatial resolution for classification is 100 meters and the smallest unit mapped must be at least 25 hectares (European Environment Agency 1985; Härmä et al. 2014). However, this resolution is not usable for many use cases, and higher resolution data is typically produced for national use. In CLC2012 project the Finnish Environment Institute produced high resolution data in addition to CLC2012 data. This high resolution data has spatial resolution of 20 meter, and some CLC classes are aggregated to fourth level. Also, some CLC classes that aren't found in Finland, such as "glaciers", are removed from national data (Härmä et al. 2014).

Finnish CLC products are not produced from only one source, but rather gathered from existing national databases. For instance, forest data is gathered from National Forest Inventory (updated by Forest Research Institute), buildings from the databases of Population Register Centre and roads from Finnish Transport Agency. European-wide CLC data is then produced

Level 1	Level 2	Level 3	Level 4
3. Forests and semi-natural areas	3.1. Forests	3.1.1. Broad-leaved forest	3.1.1.1 Broad-leaved forest in mineral soil
			3.1.1.2. Broad-leaved forest in peat soil
		3.1.2. Coniferous forest	3.1.2.1. Coniferous forest in mineral soil
			3.1.2.2. Coniferous forest in peat soil
			3.1.2.3. Coniferous forest in bedrock

Table 1. Example of aggregating CLC classes for class 3. Forests (European Environment Agency 1985; Härmä et al. 2014)

from these national produts (Härmä et al. 2014).

2.3 Convolutional neural networks

Convolutional neural networks (CNNs) are a specific type of deep learning methods, which in turn is a specific type of machine learning methods. The basic example of deep learning is a deep feedforward network or feedforward neural network, also known as *multilayer perceptron* (MLP). Simply put, MLPs are functions mapping a set of input values to some output values. The difference between linear machine learning functions and MLPs is that the functions MLPs represented are a chain of simpler functions. For example, consider a simple linear score function $f = Wx$, where W is arbitrary weight matrix and x a vector so that Wx is defined. This can be ”expanded” into a two-layer neural network simply by chaining another function, so the scores are calculated with $f = W_2max(0,W_1x)$. By adding another function, this then becomes a three-layer network, and the scores are given with $f = W_3max(0,W_2max(0,W_1x))$ and so on. The last layer of the network, the *output layer*, gives the final result of the network, such as class scores for classification tasks. The number of these *hidden layers* between input and output layers define the depth of the network (Goodfellow, Bengio, and Courville 2016; Li, Johnson, and Yeung 2018).

It's unusual for real world problems to be correctly represented with linear functions, and for those that are can be solved without deep learning methods. Neural networks apply non-linearity usually with affine transformation $W^\top x + b$, where W is the matrix containing learned weights and b is a vector containing the biases. This transformation is followed by fixed nonlinear function, the *activation function* that is applied to each element of vector $W^\top x + c$. This forms an affine transformation from vector x to vector y: $y = g(W^\top x + c)$. Most commonly used and the ”default” activation function is called rectified linear unit (ReLU), which is simply $g(x) = max(0, x)$. This function is partially linear. Other widely used activation functions are for example leaky ReLU ($g(x) = max(\alpha x, x), \alpha$ usually 0.01 or smaller), sigmoid ($g(x) = \frac{1}{1+e^{-x}}$) and tanh ($g(x) = tanh(x)$) (Goodfellow, Bengio, and Courville 2016). Different activation functions and their plots in interval [-5,5] are visualized in figure 4.

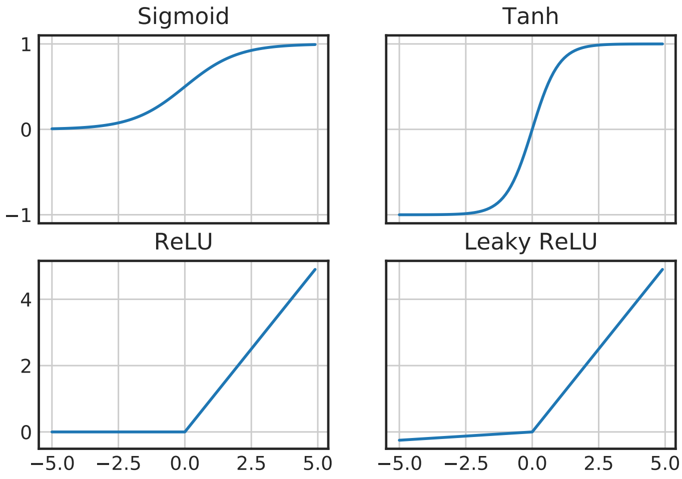


Figure 4. Plots of different activation functions

As it was mentioned previously, the matrix $W^\top$ in the equation contains the learned weights, but how does the network learn these weights? First of all, weights are usually initialized randomly, for example with He normal initialization (He et al. 2015). Classification scores are then calculated using these weights, and the function that measures the difference between these classifications and correct ones is called cost function. When cost function reaches zero, then classifier does perfect work. Minimizing cost function is usually performed by gradient-based optimization, *gradient descent*. Because the local minimum of a function is in the direction of the negative gradient, adjusting the weights like so drives the cost function down. However, weights are not adjusted with full value of the negative gradient, and it is adjusted with learning rate (either static or decreasing over time) before computing new weights. Typically gradient descent is not performed for the whole dataset, but for some minibatch of it. This algorithm for is called *stochastic gradient descent* (SGD). SGD combined with *backpropagation* (BP) is an effective way to learn the weights (Goodfellow, Bengio, and Courville 2016). BP is, simply put, recursively applying the chain rule of calculus, let $y = g(x)$ and $z = f(g(x)) = f(y)$. Then $\frac{dz}{dx} = \frac{dz}{dy}\frac{dy}{dx}$, to calculate gradients for each weight of neural network (Goodfellow, Bengio, and Courville 2016; Rumelhart, Hinton, and Williams 1986).

Goodfellow, Bengio, and Courville (2016) define CNNs to be neural networks that use convolution instead of general matrix multiplication in at least one of their layers. This greatly reduces the computational cost for training the network. *Convolution* operation is usually denoted with $*$, and mathematical formula for it is shown in equation 2.2 (Goodfellow, Bengio, and Courville 2016).

$$s(t) = (x * w)(t) = \int_{-\infty}^{\infty} x(a)w(t-a)da \tag{2.2}$$

Here, the argument x is known as *input*, w is called *kernel* or *filter*, and output is called *feature map*. For machine learning applications, it can be assumed that both x and w are zero everywhere else but on a finite number of points, and variable t can only take integer values. Because of this, the integral can be computed as a summation over finite elements. Also, because machine learning data is usually at least two dimensional array, called tensor, both image and kernel are at least two dimensional. Equation 2.3 shows this operation with two-dimensional input I and two-dimensional kernel K. Note that kernel is usually smaller than input (Goodfellow, Bengio, and Courville 2016).

$$S(i,j) = (I*K)(i,j) = \sum_{m}\sum_{n} I(m,n)K(i-m,j-n) = \sum_{m}\sum_{n} I(i-m,i-n)K(m,n) = (K*I)(i,j) \tag{2.3}$$

Equation 2.3 shows also the commutative property of convolution. This property is attained by ”flipping” the kernel relative to the input. Before kernel-flip the index into the input decreases and index into the kernel increases, and afterwards it is vice versa. However, there is no reason to flip the kernel except attaining commutative property which is useful for mathematical proofs but not at all relevant for machine learning purposes. Because of that, machine learning libraries usually implement *cross-correlation* (Equation 2.4), which is the same as convolution without the kernel-flip. In machine learning frameworks however, both of these operations are usually called convolution. Because convolution is typically used with non-commutative functions, the combination doesn’t commute regardless whether convolution or cross-correlation is used (Goodfellow, Bengio, and Courville 2016).

$$S(i,j) = (I*K)(i,j) = \sum_{m}\sum_{n} I(i+m,j+n)K(m,n) \tag{2.4}$$

CNNs were first presented by LeCun (1989), and first commercial applications of CNNs were developed in early 1990’s, when AT&T utilized CNNs for bank check reading system, and later, motivated by this according to LeCun, Kavukcuoglu, Farabet, et al. (2010), Microsoft developed handwriting recognition system with CNNs. It is worth noticing, that computational effiency of CNNs was an important factor of their success in 1990’s (Goodfellow, Bengio, and Courville 2016; LeCun, Kavukcuoglu, Farabet, et al. 2010). CNNs are mainly used for image classification and processing tasks, but they have also been utilized in natural language processing, such as speech recognition and text classification (Bhandare et al. 2016; Goodfellow, Bengio, and Courville 2016).

A major breakthrough for CNNs in image recognition was AlexNet (Krizhevsky, Sutskever, and Hinton (2012)), which was first CNN to win ILSVRC (ImageNet Large Scale Visual Recognition Competition) in 2012. Their implementation, containing eight layers, managed to reduce top-5 test error rate to 15.6%, compared to 26.2% that previous non-CNN implementations had performed. Since then CNNs have been utilized in each winning implementation of the competition, and by 2014 error rate had dropped to less than 7% (Russakovsky et al. 2015). 2017 competition was won by a model called SENet (Hu, Shen, and Sun (2017)) with classification error of approximately 2.2%. For reference, according to Russakovsky et al. (2015), human annotator is able to have classification error of around 5%.

Convolutional layers consist of three stages: Convolution, detector and pooling stages. Depending on terminology used, these stages can also be referred as layers. *Convolution stage* differ from traditional networks fully connected layers by usually having *sparse interactions*. Because kernel is made to be smaller than input image, convolutional layers both store significantly smaller amount of parameters and have smaller runtime compared to fully connected layers. For example, if we have m inputs and n outputs, then matrix multiplication stores $m \times n$ parameters and has runtime of $O(m \times n)$, but by limiting the number of connections by making the kernel have a size of $k \times k$, these numbers are now $m \times k$ and $O(m \times k)$ respectively. Other ways convolutional layers improve from traditional hidden layers are parameter sharing (same parameter is utilized more than once) and equivariance ($f(x)$ is equivariant to g if $f(g(x)) = g(f(x))$) (Goodfellow, Bengio, and Courville 2016).

Convolutional stages each contain K kernels, also known as filters, which are typically of size 3×3 or 5×5. These filters are "slided" across the input, with *stride* of S and *padding* of P. Stride means the amount of cells kernel is moving at once. Padding is the number of zeros added to each side of input. By varying these variables, it is possible to control the size of the output. An example summary of a two-dimensional convolutional layer is summarized in table 2 and figure 5.

After convolution has been performed, output is run through a *detector stage*, which applies the activation function, typically ReLU to output values. Third stage of a convolutional layer is *pooling stage*, which is typically performed as max pooling. Pooling stage has two hyperparameters: spatial extent of a neighborhood F and stride S, which are similar than in convolution stage. Max pooling outputs the maximum value of a neighborhood and then moves S steps. Aside from max pooling, also for example average pooling or L^2 norm pooling are possible. (Goodfellow, Bengio, and Courville 2016). An example of a max pooling layer with $F = 2$ and $S = 2$ is summarized in table 3 and figure 6.

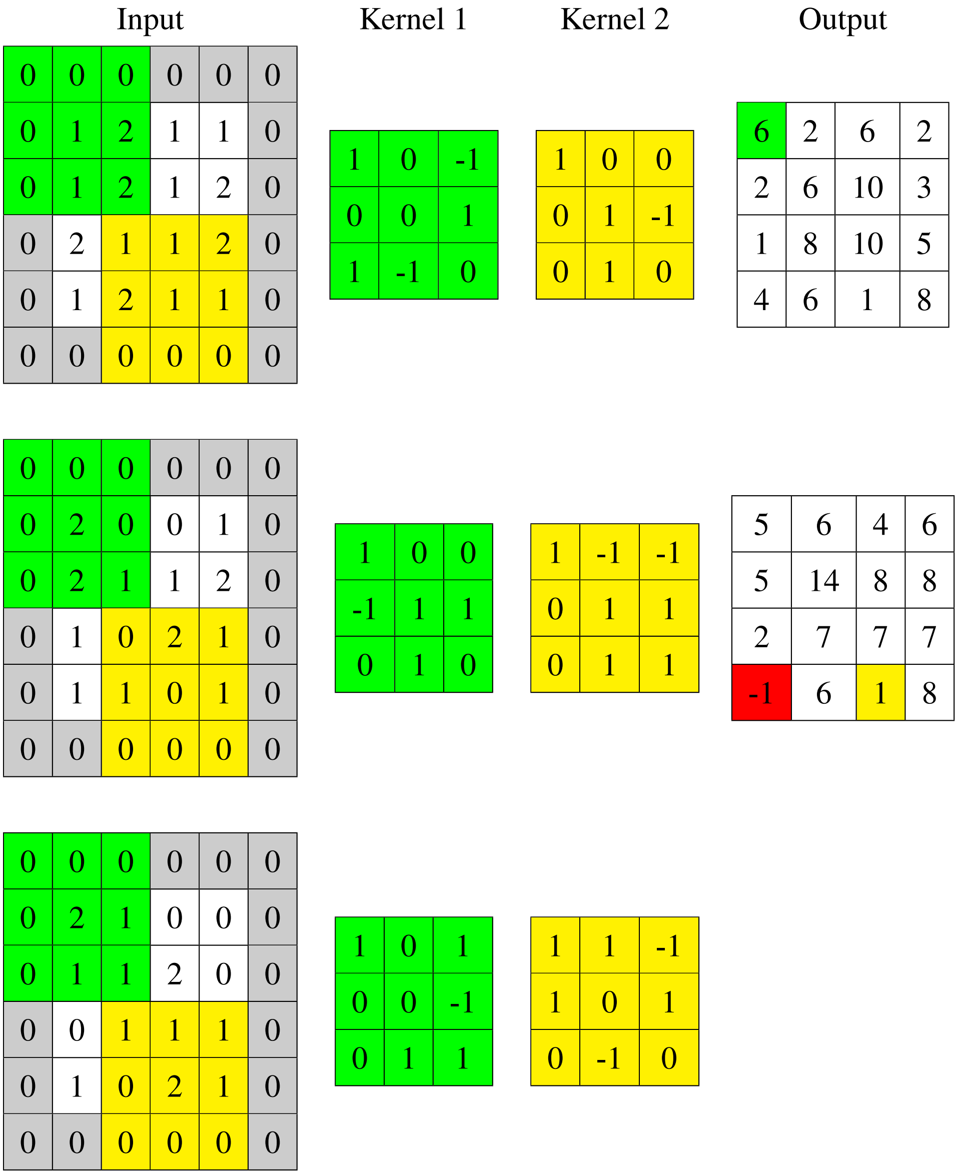
Description	**symbol**	**example**
Input volume	$W_1 \times H_1 \times D_1$	$4 \times 4 \times 3$
Number of kernels	K	2
Spatial extent of a kernel	F	3
Stride	S	1
Amount of zero padding	P	1
Output width	$W_2 = \frac{W_1 - F + 2P}{S} + 1$	$4 = \frac{4-3+2\cdot 1}{1} + 1$
Output height	$H_2 = \frac{H_1 - F + 2P}{S} + 1$	$4 = \frac{4-3+2\cdot 1}{1} + 1$
Output depth	$D_2 = K$	2
Output volume	$W_2 \times H_2 \times D_2$	$4 \times 4 \times 2$
New weights per kernel	$F \cdot F \cdot D_1$	$3 \cdot 3 \cdot 3 = 27$
Total new parameters from layer	$(F \cdot F \cdot D_1) \cdot K$	$(3 \cdot 3 \cdot 3) \cdot 2 = 54$

Table 2. Summary of a convolutional stage performing 2D convolution to three-channel data (Goodfellow, Bengio, and Courville 2016)

Description	**symbol**	**example**
Input volume	$W_1 \times H_1 \times D_1$	$4 \times 4 \times 2$
Spatial extent of neighborhood	F	2
Stride	S	2
Output width	$W_2 = \frac{W_1 - F}{S} + 1$	$2 = \frac{4-2}{2} + 1$
Output height	$H_2 = \frac{H_1 - F}{S} + 1$	$2 = \frac{4-2}{2} + 1$
Output depth	$D_2 = D_1$	2
Output volume	$W_2 \times H_2 \times D_2$	$2 \times 2 \times 2$

Table 3. Summary of a two dimensional max pooling stage (Goodfellow, Bengio, and Courville 2016)

After desired number of convolutional layers, the feature maps are then processed into a classifiable form. This can be done for instance by flattening the spatial dimensions and feeding the result to a feedforward network or with specified pooling operation. In any case, for classification tasks the last step is acquiring a probability distribution for possible classes, usually with *softmax* function (Equation 2.5). Softmax function is normalized exponential



Input

0	0	0	0	0	0
0	1	2	1	1	0
0	1	2	1	2	0
0	2	1	1	2	0
0	1	2	1	1	0
0	0	0	0	0	0

Kernel 1

1	0	-1
0	0	1
1	-1	0

Kernel 2

1	0	0
0	1	-1
0	1	0

Output

6	2	6	2
2	6	10	3
1	8	10	5
4	6	1	8

0	0	0	0	0	0
0	2	0	0	1	0
0	2	1	1	2	0
0	1	0	2	1	0
0	1	1	0	1	0
0	0	0	0	0	0

1	0	0
-1	1	1
0	1	0

1	-1	-1
0	1	1
0	1	1

5	6	4	6
5	14	8	8
2	7	7	7
-1	6	1	8

0	0	0	0	0	0
0	2	1	0	0	0
0	1	1	2	0	0
0	0	1	1	1	0
0	1	0	2	1	0
0	0	0	0	0	0

1	0	1
0	0	-1
0	1	1

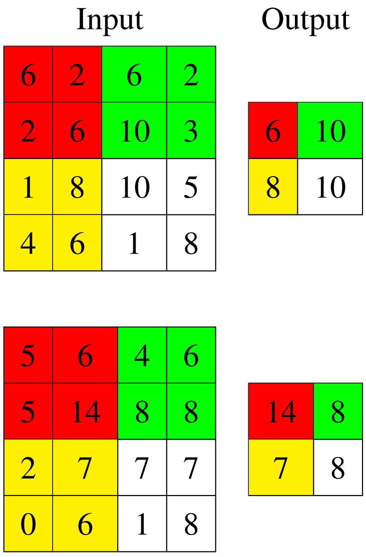
1	1	-1
1	0	1
0	-1	0

Figure 5. 2D convolution stage for three-channel data

Green and yellow cells show the areas and kernels that correspond certain output value. Red cell highlights a value that changes to 0 after ReLU.

function which rescales its input vector so that all entries add up to 1 (Goodfellow, Bengio, and Courville 2016).

$$softmax(x)_i = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}} \tag{2.5}$$



6	2	6	2
2	6	10	3
1	8	10	5
4	6	1	8

6	10
8	10

5	6	4	6
5	14	8	8
2	7	7	7
0	6	1	8

14	8
7	8

Figure 6. Two dimensional max pooling stage

Colors correspond to neighborhoods

2.3.1 Image segmentation with CNNs

Image segmentation can be divided into two different subcategories: *semantic segmentation*, which aims to label all pixels in the image into predetermined subclasses, and *instance segmentation*, which in addition to labeling all pixels into subclasses, also aims to differentiate between different instances of same subclass, such as labeling them as ”animal 1” and ”animal 2”. Land cover classification can be considered to be one form of semantic segmentation, so this section focuses on it.

Labelling pixels can be thought as a classification task, where each pixel is labelled one-by-one, with some amount of surrounding pixels as a spatial pattern to help to classify it. This approach, also known as *sliding window approach*, while successful has some shortcomings, such as performance speed. Recent research has shown that pooling stages and fully connected layers can be replaced with convolution stages and thus improve performance, at least in image segmentation tasks. As described earlier, by varying stride S and padding P in convolution stage it is possible to alter the output size. Fully connected layers can be replaced by 1×1 convolutions, and by averaging this layer the classification scores are achieved (Springenberg et al. 2015).

It is possible to upsample images with transposed convolution (known as *backwards convolution* or *deconvolution*), so this kind of approach is applicable also for different segmentation tasks (Long, Shelhamer, and Darrell 2015). After up-sampling the output is not a single label for one image, but rather an array of labels. Another way to upsample images is simply just expand regions into larger areas, such as each cell in 2×2 matrix into 4 cells in 4×4 matrix. However, transposed convolution has the advantage of learning how to upsample input instead of it being fixed (Long, Shelhamer, and Darrell 2015).

One of the most successful implementations of this kind of encoder-decoder (autoencoder) type networks is U-Net by Ronneberger, Fischer, and Brox (2015), which was originally developed for biomedical image segmentation. This network consists of encoding and decoding paths. Encoding path follows the standard CNN structure: repeated 3×3 unpadded convolutions with ReLU as activation functions and standard 2×2 max pooling operations with different number of kernels. Decoding path, however, replaces max pooling operations with transpose convolutions, which doubles the resolution of each feature map. In addition, each upsampled feature map is concatenated with cropped feature maps from the ”same level” of encoding path. Feature maps from encoding path must be cropped, because unpadded convolutions reduce size by two. Final layer of U-Net is 1x1 convolution with softmax activation, which produces a segmentation map for the image. Full architecture for Ronneberger, Fischer, and Brox (2015) implementation can be seen in image 7.

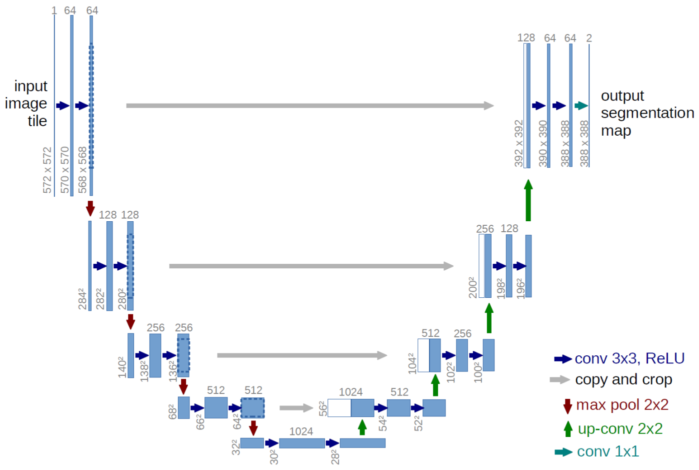


Figure 7. U-Net architecture (Ronneberger, Fischer, and Brox 2015)

Even though the original purpose of U-Net is biomedical image segmentation, it has been successfully applied to other fields, such as remote sensing (Zhang, Liu, and Wang 2018) and segmentation of photographs (Siam et al. 2018). Other commonly used autoencoder-type networks are Fully-Convolutional Network (FCN) introduced by Long, Shelhamer, and Darrell (2015) and SegNet by Badrinarayanan, Kendall, and Cipolla (2015), which differ slightly from U-Net. However, the basic principle is the same: Encoding path follows some typical CNN structure (VGG16 in the case of U-Net), and decoding path upsamples images in some way, producing full segmentation mask as output.

3 Materials and methods

In this chapter, the workflow to produce segmentation maps is presented. Section 3.1 presents materials used, how they were acquired and what preprocessing steps were performed to them. In addition, labels are inspected and their aggregation from original labels is described. Section 3.2 presents implemented neural network structure and metrics for evaluating its performance. Hardware and tools used are also described there, as well as monitored training and validation losses for trained neural networks. Even though several LULC segmentation works perform some kind of postprocessing for segmentation masks, such as edge enhancing or removing areas smaller than some threshold, in this thesis only ”raw” segmentation masks are presented and inspected in chapter 4.

3.1 Materials

Multispectral satellite images used in this thesis are Sentinel-2 products. Sentinel-2 (S2) mission consists of two polar-orbiting satellites Sentinel-2A (S2A) and Sentinel-2B (S2B). S2A was launched first, on 25 June 2015 and S2B later, on 7 March 2017. Both satellites carry an optical instrument sampling 13 spectral bands (Table 4), ranging from VNIR to SWIR. These images are divided into different levels of products. Of these, levels 0 through 1B are raw data not available to end users. Level-1C product contains top of atmosphere (TOA) reflectance values, and Level-2A contains bottom of atmosphere (BOA) generated from Level-1C products. S2 band B10 is used only for cirrus cloud detection, and thus it’s absent from Level-2A products. Both Level-1C and 2A products are composed of 100x100m^2 tiles. Level-1C products can be downloaded from Copernicus Open Access Hub, along with some amount of Level-2A data. Level-2A data can also be generated with Sentinel Application Platform (SNAP) and Sen2Cor extension. All Sentinel-2 products can be used for non-commercial uses, regulated and allowed under EU law, provided that ESA is credited properly (European Space Agency 2018c).

Processed satellite data was stored in University of Jyväskylä network drives and SYKE network drives, as well as laptop provided by SYKE. The research data policy of SYKE aims, that the data will be shared through SYKE Metadata and Research Data services with CC BY 4.0 license, but at the time of writing the sharing process is still uncertain. This thesis doesn't use any non-open data, nor any data that uses personal information.

Band number	S2A	S2B	Spatial resolution	Name
B1	443.9 ± 13.5nm	442.3 ± 22.5nm	60m	Aerosol
B2	496.6 ± 49nm	492.1 ± 49nm	10m	Blue
B3	560.0 ± 22.5nm	559.0 ± 23nm	10m	Green
B4	664.5 ± 19nm	665.0 ±19.5nm	10m	Red
B5	703.9 ± 9.5nm	703.8 ± 10nm	20m	Red-edge 1
B6	740.2 ± 9nm	739.1 ± 9nm	20m	Red-edge 2
B7	782.5 ± 14nm	779.7 ± 14nm	20m	Red-edge 3
B8	835.1 ± 72.5nm	833.0 ± 66.5nm	10m	NIR_{wide}
B8a	864.8 ± 16.5nm	864.0 ± 16nm	20m	NIR_{narrow}
B9	945.0 ± 13nm	943.2 ± 13.5nm	60m	Water vapor
B10	1373.5 ± 37.5nm	1376.9 ± 38nm	60m	Cirrus
B11	1613.7 ± 71.5nm	1610.4 ± 141nm	20m	SWIR 1
B12	2202.4 ± 121nm	2185.7 ± 119nm	20m	SWIR 2

Table 4. Overview of Sentinel-2A and Sentinel-2B multispectral instruments (European Space Agency 2018c)

Acquiring data for this thesis was pretty straightforward, because research engineer Markus Törmä from SYKE Geoinformatics Research generously Sen2Cor processed several images from April-October 2016 and 2017. In addition, Törmä provided different mosaics for NDVI, NDTI, NDMI, NDBI and NDSI dated similarly than Level-2 products, and CLC2012 Level 4 labels for the region of interest (ROI). These indices are rescaled between 0 and 200 instead of -1 and 1. Most of Level-2 products are heavily covered in clouds, and finding even one cloudless image is almost impossible. In order to get cloudless mosaics (Level-3 product), these images were then processed with Sen2Three processor tool. Sen2Three processes a time series of Sen2Cor corrected images by replacing all bad pixels (for example clouds and cloud shadows) with better pixels. Depending on the algorithm used, bad pixels are replaced either with the good pixels from later scenes or with the average of all good pixels

(European Space Agency 2018b). Mosaic used in this thesis was generated with temporal homogeneity algorithm. While Sen2Three doesn't produce perfect mosaics, it still makes a satisfactory job at removing clouds and making usable images for this purpose. After generating the cloudless mosaic, due to the spatial resolution of labels being 10m pixels, all other bands were then downscaled to this resolution using SNAP. Images were then exported to GeoTIFF format.

Final training and testing dataset contains both cloudless 12 band reflectance mosaic and pre-computed spectral index mosaic from summer 2017 composed from images from June to August. Spectral index values are selected according to maximum NDVI value observed during this period. Summer mosaic was selected because it contains all of the target classes and they are able to be distinguished. For instance, water vegetation may not be present during spring or autumn, and agricultural areas can be identified. Description of provided spectral indices is presented in table 5.

Spectral index	Derivation	Usage
NDVI	$\frac{RED-NIR}{RED+NIR}$	Green vegetation detection
NDTI	$\frac{SWIR1.5-SWIR2.0}{SWIR1.5+SWIR2.0}$	Dry vegetation detection
NDBI	$\frac{SWIR-NIR}{SWIR+NIR}$	Urban area detection
NDMI	$\frac{NIR-SWIR1.5}{NIR+SWIR1.5}$	Water detection
NDSI	$\frac{GREEN-SWIR1.5}{GREEN+SWIR1.5}$	Snow/ice detection

Table 5. Provided spectral indices

As mentioned before, the main reason for ROI being in the Finnish-Russian border is Rakkolanjoki river. Land cover classification in this drainage basin area could be used to calculate nutrient loads, provided it exists. While this classification is available in the Finnish side of the border, due to Russia not being part of EU and CORINE almost half of the drainage basin area is unclassified. Target land cover classes are selected to be suitable for water quality modelling, meaning their nutrient loads are different from each other, and areas producing similar loads are grouped together.

ROI contains approximately 18 million labelled pixels and out of 48 CLC level 4 classes, ROI contains pixels from 41 of them. These labels were reduced to 11 target classes, and

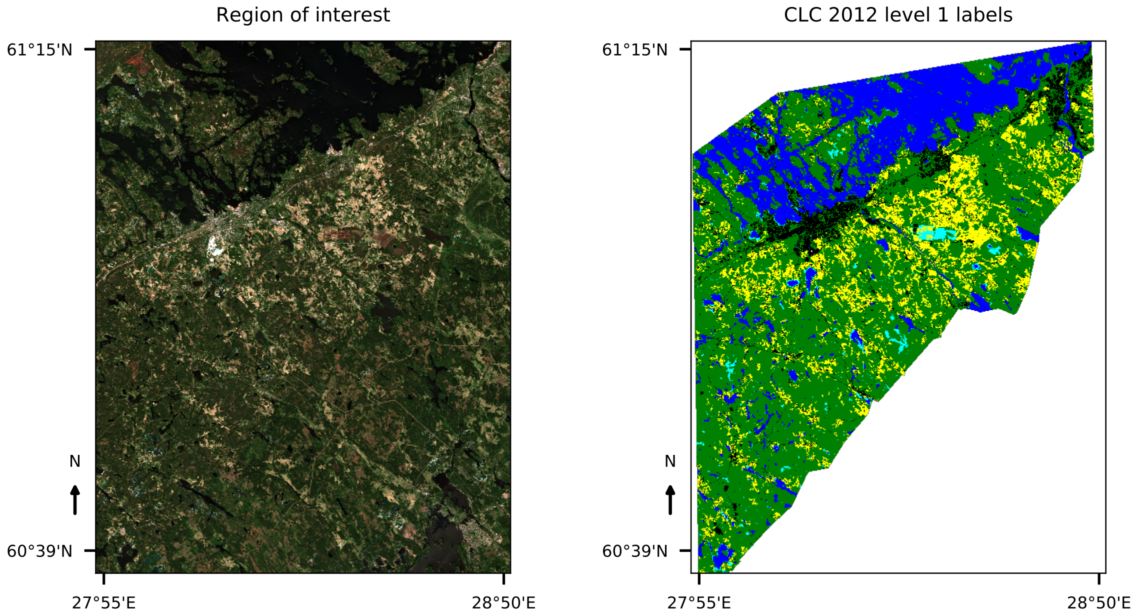


Figure 8. Left: Map of the region of interest. (Copernicus Sentinel Data[2018]). Right: CLC level 1 labels for ROI.

also to CLC level 1 and CLC level 2 labels for testing and comparison purposes. Each of these classifications is heavily unbalanced, with most common CLC level 1 and CLC level 2 label being "Forest", around 60% of all labels, and most common target class label being "Coniferous forest", around 38% of all labels. Rarest classes are wetlands for CLC level 1 (around 1.7% of labels), pastures for CLC level 2 (around 0.03 %) and water vegetation for target classes (around 0.6%). Because some classes are extremely underrepresented, classifying them correctly will be very difficult. Distributions of classes are presented in figure 9, and aggregation of CLC level 4 classes to CLC level 1, CLC level 2 and target classes is presented in appendix A.

CLC level 1 classes are possibly the easiest ones to distinguish from each other using only average reflectance values, which is understandable due to there being only five of them, and each of those have different spectral signature, with next to none overlap between average reflectance values. Out of these, most absorbent class is, not surprisingly, water and most reflective is arable land. Normalized spectral indices show similar behavior. Due to water being absorbent for all bandwidths, values for it hover near zero, whereas other classes have

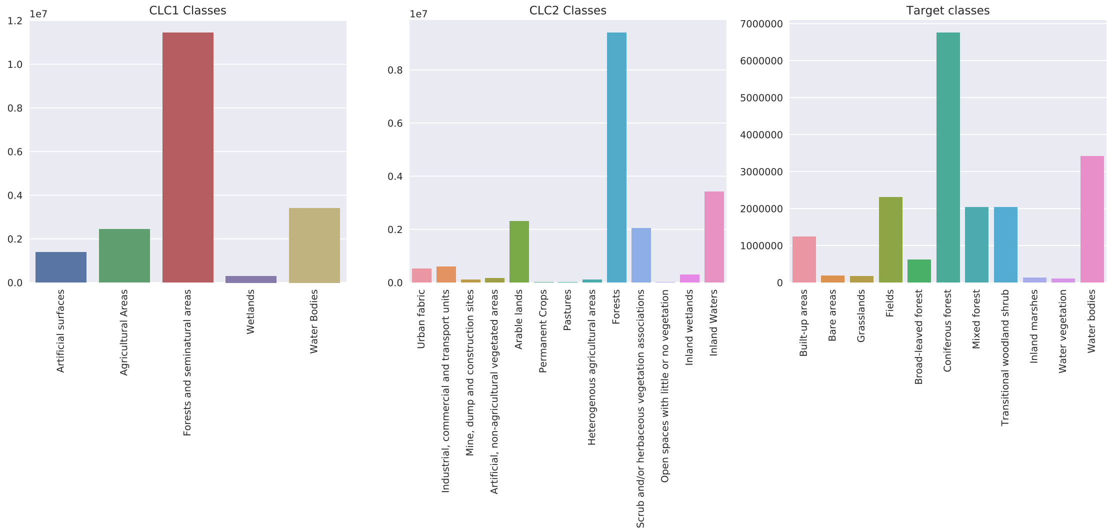


Figure 9. Distribution of land cover class labels in ROI

values over 0.5 for NDVI, but almost -1.0 for NDBI.

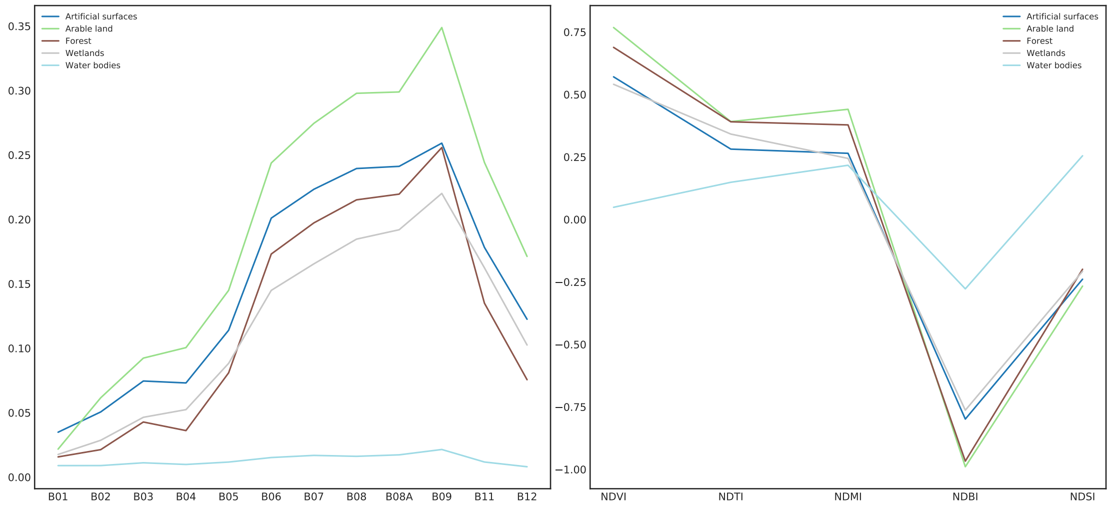


Figure 10. Average reflectances and normalized spectral index values for CLC level 1 classes

CLC level 2 classes show much more overlap than CLC level 1 classes, which is understandable due to there being 13 different classes instead of 5. Water still behaves similarly than before, and classes aggregated from arable land are the most reflective ones. Some classes aggregated from same CLC level 1 class show a lot of overlap, such as permanent crops and pastures, and it would be reasonable to expect them to mix up or be correctly classified at all, especially because they are so underrepresented.

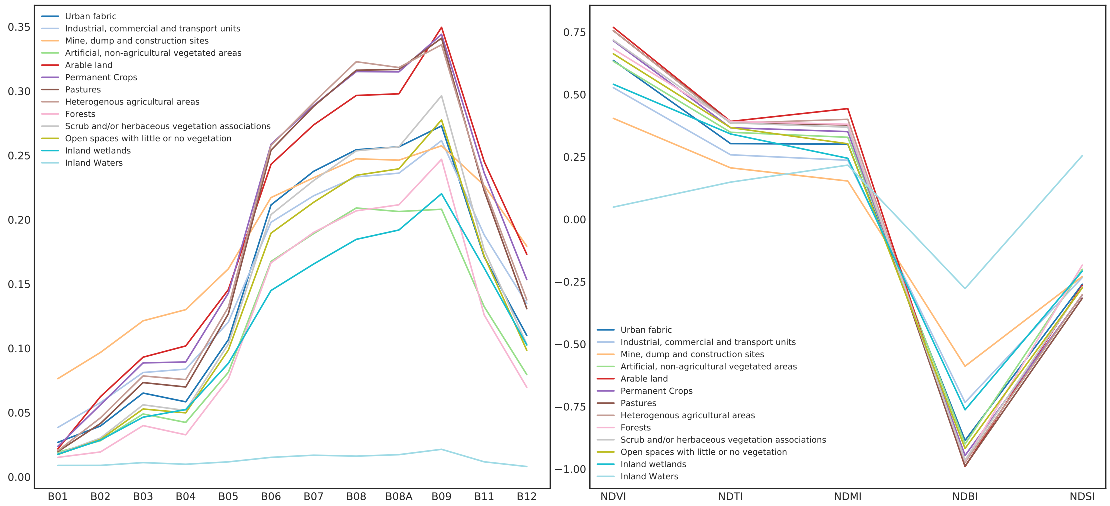


Figure 11. Average reflectances and normalized spectral index values for CLC level 2 classes

Target classes are a bit simpler to distinguish than CLC level 2 classes, at least based on average values. Even though there are three different forest types, their spectral signatures differ clearly. Coniferous forest is least reflective, and broad-leaved forest the most with mixed forest in between. Classes with most overlap are broad-leaved forest and transitional woodland shrub. It is worth noticing, that these classes may contain subclasses from multiple CLC level 1 classes. For example, class bare areas consist of dumps and construction sites (CLC level 1: Artificial surfaces) but also of peat production areas (CLC level 1: Wetlands), and grasslands contains both golf courses (CLC level 1: Artificial surfaces) and pastures (CLC level 1: Arable land).

From this dataset, a smaller subset was extracted near Lappeenranta, seen in figure 13. This subset, sized 19.2×38.4 km contains approximately 30% of all provided labels. Smaller 128×128 px (1.28km) images were extracted with 64px stride from this area. This window size is chosen rather arbitrary, and larger windows may work better. This was then augmented so that each image was flipped both horizontally and vertically, and also rotated 90 degrees both ways, thus fivefolding the size of all sets. All values were then cast to 32 bit floating point numbers, and S2 bands were divided with 10 000 and spectral indices with 200 in order to get the values approximately between 0 and 1. Afterwards, this set was divided into training, validation and test sets sized 5760, 1440, and 1800 images respectively. Of course, all remaining data outside training and validation sets can and will later be used for testing.

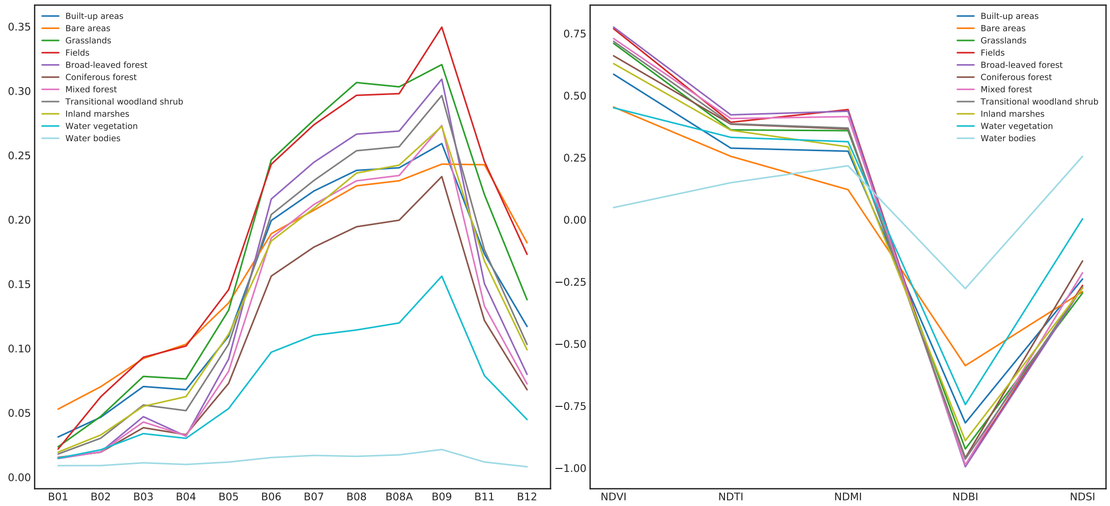


Figure 12. Average reflectances and normalized spectral index values for target classes

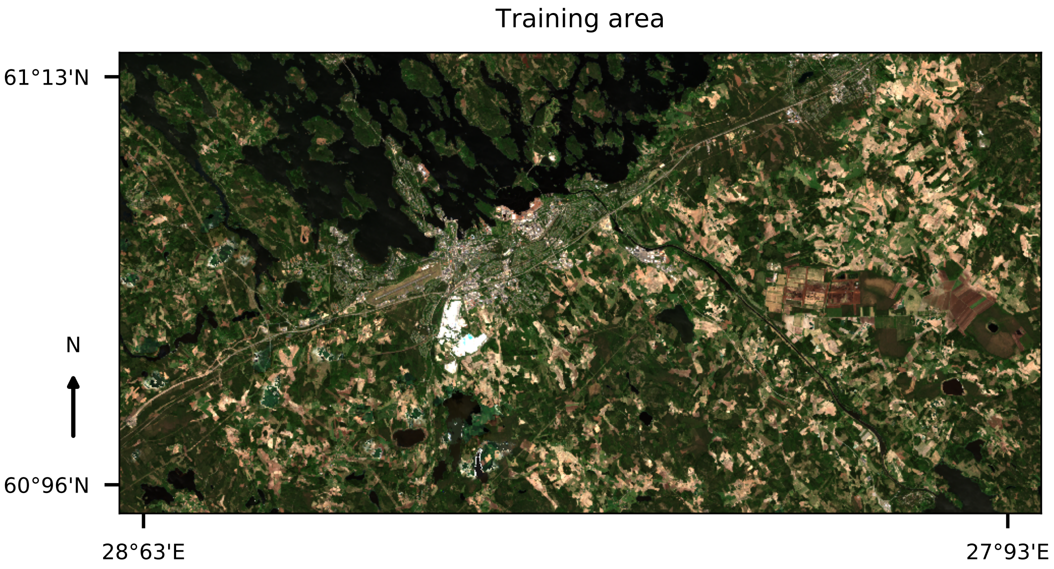


Figure 13. Area used in network training process. Notice small errors due to Sen2Three processing (Copernicus Sentinel Data[2018]).

3.2 Methods

Aside from resampling and subsetting performed with SNAP, all data preprocessing and network training were performed with platforms provided by University of Jyväskylä Faculty of Information Technology using Anaconda distribution of Python 3.6. Jupyter Notebook was heavily used, especially for inspecting data. All graphs and visualizations were produced with matplotlib and Seaborn Python libraries (Hunter 2007; Waskom et al. 2017). Networks were implemented with Keras using Tensorflow as backend (Chollet et al. 2015; Abadi et al. 2016) and trained with Nvidia GeForce GTX 1080 GPU.

3.2.1 Evaluation metrics

For semantic segmentation, the most common measures are accuracy, precision, recall, Jaccard index and F1-score. Of these, accuracy can be calculated as *overall pixel accuracy* (OP) or *per-class accuracy* (PC). OP is just a proportion of correctly labelled pixels, and PC is OP averaged with the number of classes. Precision tells the proportion of correct positive predictions (true positives, TP) from all TP and FP (false positives), while recall is the proportion of TP from all positive labels (TP and false negative (FN) predictions). The de-facto standard of segmentation tasks is *Jaccard index* (JI), also known as intersection over union (IoU), which accounts also the false alarms and missed values unlike different accuracy measurements. For multiclass problems, JI is not only computed for each class separately, but also for the whole dataset by calculating the arithmetic mean of JI scores for each class. F1-score (also known as *Dice coefficient*) is the harmonic mean of precision and recall (Csurka et al. 2013). Equations for different metrics are show in equation 3.1.

$$\begin{aligned} &OP = \frac{TP+TN}{TP+FP+FN+TN}, \quad PC = \frac{1}{L}OP \\ &Precision = \frac{TP}{TP+FP}, \quad Recall = \frac{TP}{TP+FN} \\ &F1(Dice) = \frac{2 \cdot precision \cdot recall}{precision + recall}, \quad JI(IoU) = \frac{TP+TN}{TP+FP+FN} \end{aligned} \tag{3.1}$$

Out of the metrics provided here, it is possible to calculate *micro*, *macro* and *weighted* scores for Pre, Rec, and F1-score when evaluating the full dataset. Macro calculates the mean of these metrics with each class having equal weight, whereas micro sums the dividends and divisors that contribute to each labelwise score and uses them to compute the overall metric. Weighted simply weights each score with corresponding classes presence in the full data sample. Micro-averaging scores favors bigger classes and macro-averaging treats all classes equally (Sokolova and Lapalme 2009). In this thesis, if not mentioned otherwise, all overall Pre, Rec and F1-scores presented are micro scores.

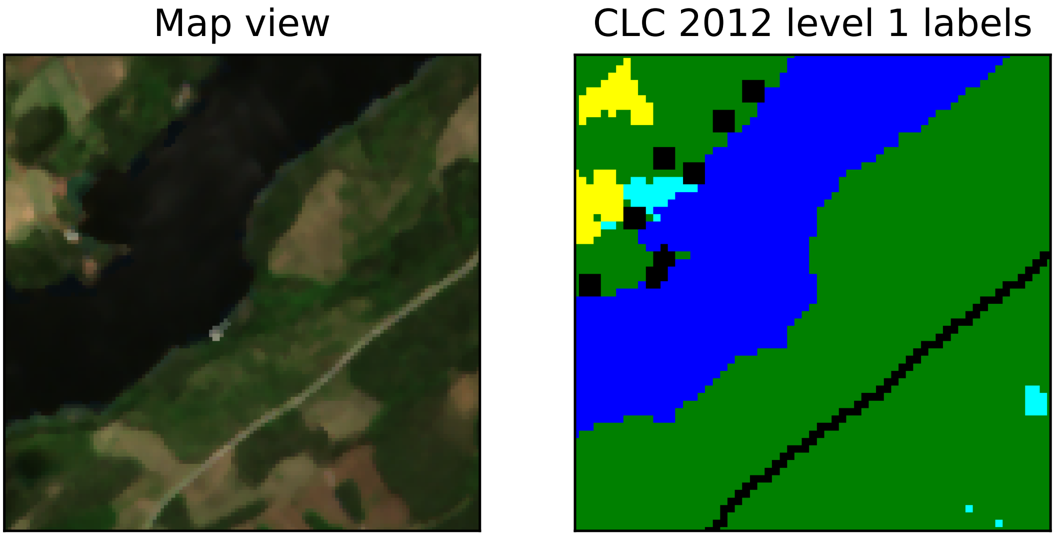


Figure 14. Example of ground truth labelling error. Black areas correspond to CLC-1 class ”Artificial surfaces” and are labelled to be larger than in reality (Copernicus Sentinel Data[2018]).

Even though ground truth labels are provided, they should not be trusted blindly. First of all, labels are from CLC 2012 project which ended in 2015, and it is highly likely that since then some changes have already happened. Also, CLC2012 labels are not 100% accurate. Compared to LUCAS 2012 (Land use and land cover survey by Eurostat), CLC level 1 classes have overall accuracy of 93%, CLC level 2 classes have 83% and CLC level 3 classes have 61%, while compared to Finnish National Forest Inventory sample points, the overall accuracy was around 92% (Härmä et al. 2014). Due to how these classifications are produced, some built-up areas are marked as being larger than they are in reality, as seen in figure 14. Because of this, metrics using provided labels as ground truth should be considered as a guideline, not a certain truth.

Because JI accounts also the incorrect predictions for each class, it is considered to be the de-facto evaluation metric for segmentation tasks. However, it has the problem of not necessarily evaluating how accurate the segmentation borders are Csurka et al. (2013). Because of this, and also the slight uncertainty of ground truth labels, results will be evaluated based on accuracy, precision, recall, JI-score and F1-score, but also by visually interpreting segmentation masks and comparing them to CLC 2012 masks.

3.2.2 Neural network implementation and training

At first, I tried to utilize 3D CNNs, sliding window approach and time series of data for classification, as described in (Ji et al. 2018). However, while classification results for single points were promising, full image classification was both slow and inaccurate, even for CLC level 1 classes. Because of that, the method was changed from sliding window to fully convolutional network. For the new network architecture baseline, U-Net was chosen due to its ability to get good results with limited data. Ronneberger, Fischer, and Brox (2015) originally trained their model with only 30 labelled images sized 512×512 pixels. Even though my dataset is much larger, for land cover mapping it is not always the case, so network should be able to learn from limited data. Also, U-Net and its variations have been succesfully used in remote sensing tasks. Another reason to use fully-convolutional architecture is that it can be adapted to work with arbitrary input size as long as all pooling operations are performed for even-numbered height and width, whereas model containing fully-connected layers needs to have fixed input size due to matrix multiplications instead of convolutions.

There are few differences between original U-Net and presented implementation. First of all, while original U-Net is developed for binary classification, presented network performs multi-class classification. This means that output mask is multidimensional instead of being true-false mask. Second, presented network utilized padded convolutions in each layer, so that output mask has same size than original image. This also means that cropping the feature maps is not needed when concatenating them. Third, the number of kernels is halved for each convolution and deconvolution block in order to reduce the number of parameters of the network. Finally, there are some differences between hyperparameters, such as input image and batch size. The original U-Net had an input size of 572×572 pixels and batch size of one, but presented network was trained with smaller 128×128px images and batch size of four. Batch normalization was also added to each convolution block. Both networks use SGD as optimizer function, but U-Net has constant momentum of 0.99 while presented network utilizes Nesterov momentum. Both implementations use categorical crossentropy as loss function and He normal (He et al. 2015) initializer for weight initialization (Ronneberger, Fischer, and Brox 2015). Used network summary is presented in table 6. Its structure follows the original, but because of added zero padding, it is able to process arbitrary sized images as long as both height and width can be halved four times so that each step is an integer.

Number of layers	Trainable parameters	Non-trainable parameters	Total parameters
36	7,765,661	2,840	7,768,501

Table 6. Used neural network summary

Each network was trained from scratch for 100 epochs, and it took between one hour (CLC level 1 classes) to two hours (CLC level 2 classes) to train each of them. Overall, each network improved their performance until the end of training without strong signs of overfitting, even though improvement on validation metrics slowed down during the end. Graphs of training metrics for each networks are presented in appendices B, C and D. Best models were selected according to lowest validation loss during training, which occurred during the last few epochs for all networks.

4 Results

In this chapter, the results from neural network classification are presented and inspected. For each classification level, different results acquired with only red (B04), green (B03) and blue (B02) bands, all S2-Bands, precomputed spectral indices, and both S2-bands and spectral indices combined are compared to each other and evaluated according to the metrics presented in section 3.2.1. In addition, different segmentation maps are visually observed and compared.

Networks were first evaluated with previously described test set, and afterwards the full dataset is classified using the best performing model. This final evaluation provides not only predictions that can be compared with existing labels, but also classifications from unlabelled area. The quality of the classifications for the Russian side of the border can be estimated with results from the Finnish side. For all classification levels, best results were acquired with S2 bands and spectral indices as input features, and therefore all full dataset classifications produced with only those neural networks.

Because original data has such dimensions that selected window size is unable to fully classify it without overlapping, data was extrapolated with mirroring. For instance, for used window size of 128 $\times$ 128px, image was padded with 57 pixels on top and bottom and 38 pixel on left and right, making the total image have size of 5248 $\times$ 6656 pixels. These paddings were removed after classification. It is also worth noticing that spectral indices are not provided for the full image, so for full image classification it's likely that borders regions are incorrectly classified even after removing areas without spectral index data.

4.1 Results for CLC level 1 classification

Identifier	Class	Support
1	Artificial surfaces	3014061
2	Agricultural areas	5791164
3	Forests and seminatural areas	16359436
4	Wetlands	505891
5	Water bodies	3820648
	Total	**29491200**

Table 7. CLC level 1 class distribution in the test set

	Training		Validation		Testing	
Features	Acc	Loss	Acc	Loss	Acc	Loss
RGB bands	0.92	0.20	0.90	0.27	0.90	0.27
All bands	0.92	0.20	0.90	0.25	0.90	0.25
Indices	0.93	0.19	0.90	0.26	0.90	0.26
Combined	0.93	0.18	0.92	0.22	0.92	0.22

Table 8. Accuracy and loss for different features in CLC level 1 classification

	RGB only				Bands only				Indices only				Combined			
	Pre	Rec	F1	IoU	Pre	Rec	F1	IoU	Pre	Rec	F1	IoU	Pre	Rec	F1	IoU
1	0.74	0.69	0.71	0.56	0.76	0.68	0.72	0.56	0.74	**0.74**	0.74	0.59	**0.78**	0.71	**0.74**	**0.59**
2	0.89	0.89	0.89	0.81	0.89	0.92	0.90	0.82	0.88	0.91	0.90	0.81	**0.91**	**0.92**	**0.92**	**0.85**
3	0.92	0.93	0.92	0.86	0.92	0.93	0.93	0.87	**0.93**	0.92	0.92	0.86	0.92	**0.95**	**0.94**	**0.88**
4	0.84	0.64	0.73	0.57	0.81	0.69	0.74	**0.59**	0.76	**0.72**	0.74	0.59	**0.84**	0.66	**0.74**	0.59
5	0.96	0.96	0.96	0.92	0.96	0.96	0.96	0.92	0.97	0.96	0.96	0.93	**0.97**	**0.96**	**0.97**	**0.94**
Mic	0.90	0.90	0.90	-	0.90	0.90	0.90	-	0.90	0.90	0.90	-	**0.91**	**0.91**	**0.91**	-
Mac	0.87	0.82	0.84	0.74	0.87	0.84	0.85	0.75	0.86	**0.85**	0.85	0.75	**0.89**	0.84	**0.86**	**0.77**
Wei	0.90	0.90	0.90	-	0.90	0.90	0.90	-	0.90	0.90	0.90	-	**0.91**	**0.91**	**0.91**	-

Table 9. Test set results for CLC level 1 classes

CLC level 1 classes are by far the easiest to classify, which is, of course, not surprising at all. No matter which features were used for classification, overall accuracies, precision, recall and F1-score were at least 0.9 or higher. Labelwise results are also promising, with three out of five classes having F1-score 0.9 or higher, and other two classes having satisfactory performance with F1-score of around 0.74. Easiest class to predict is ”Water bodies” and toughest one either ”Artificial surfaces” or ”Wetlands” depending on features used. mIoU varies between 0.75 and 0.77 for whole dataset, and labelwise IoU between 0.56 (”Artificial surfaces” using only S2-bands) and 0.93 (”Water bodies” using both bands and spectral indices). All results are presented in tables 8 and 9. These show that network using both S2 bands and spectral indices outperforms other networks, but only by a little.

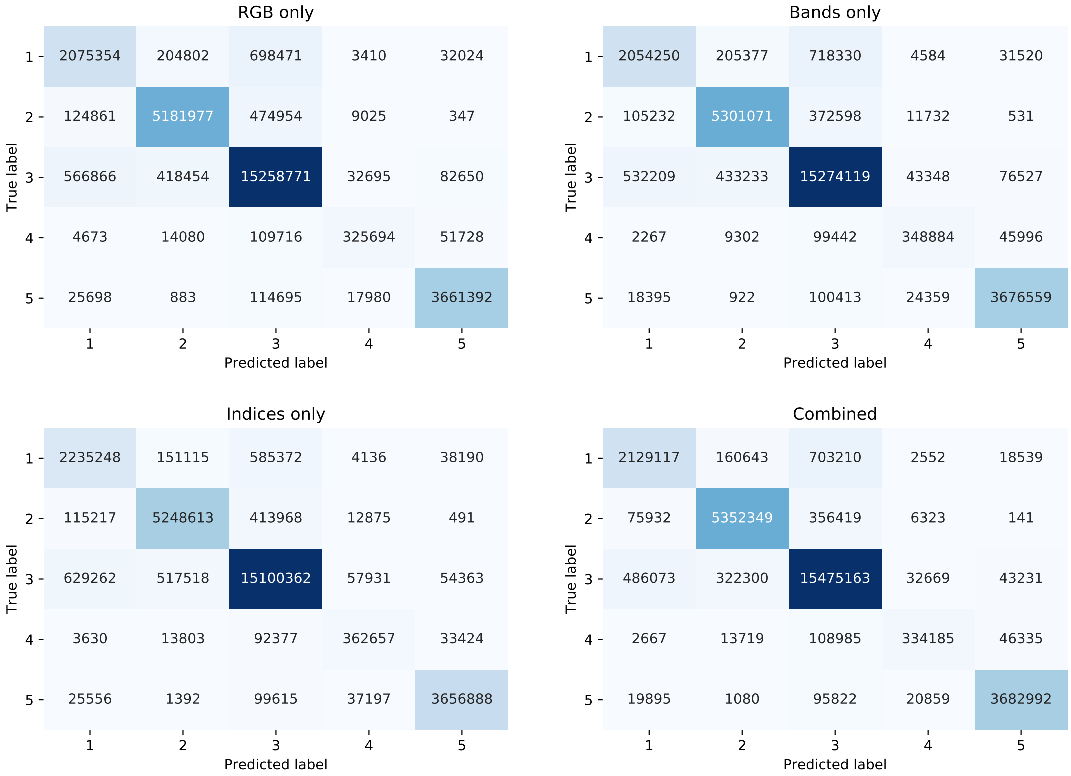


Figure 15. Confusion matrices for CLC level 1 classes

As results show, different classes for this level can be distinguished from each other quite well. Compared to provided labels, most common mix-ups between ”Forests and seminatural areas” and ”Artificial surfaces”, which can for some part be explained with labelling errors seen in figure 14, but much larger factor are narrow roads as seen in figure 16. Also, no matter what features were used, around 20% of pixels with true label ”Wetlands” were classified with label ”Forests and seminatural areas”. Confusion matrices for CLC level 1 classifications provided in figure 15.

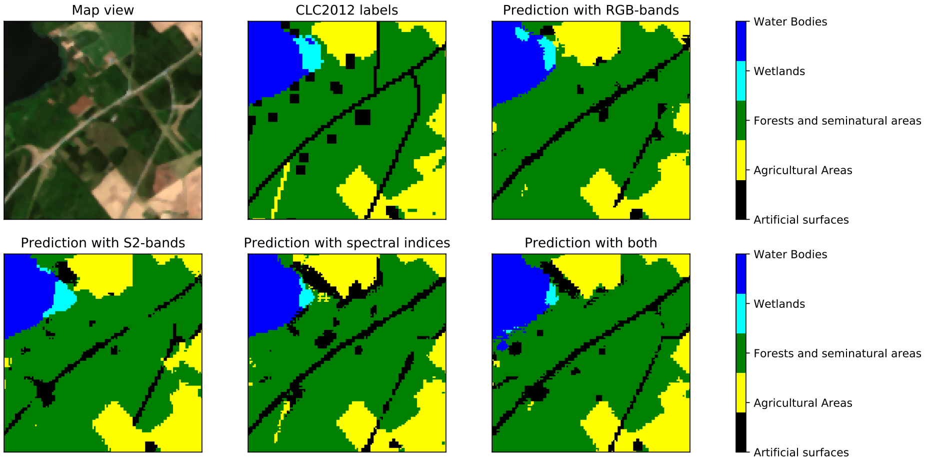


Figure 16. An example of CLC level 1 classifications in non-urban area

Visual observation shows more differences between different networks. As seen in figure 16, all classifiers have problems labelling the roads in the scene. All classifiers manage to more or less find the biggers road running across, but all predict it to be a little bit wider than in labels. None of the classifiers manage to find the northernmost road, but all of them are able to label the road in the lower part of the scene. Water bodies, forests and agricultural areas are classified similarly between different classifiers, and they mostly follow both CLC2012 labels and ground truth from map.

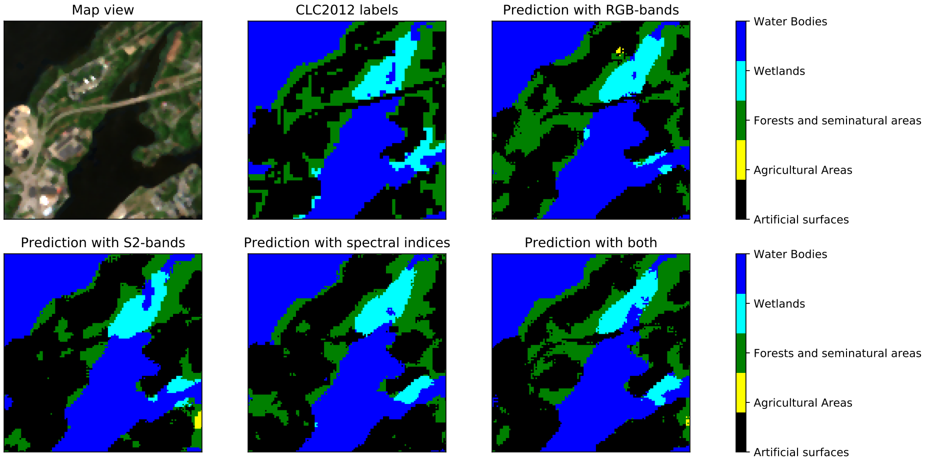


Figure 17. An example of CLC level 1 classifications in urban area

Figure 17 shows an example of a typical challenge for these networks. Densely populated urban areas are classified as large clusters, and some forests inside these areas stay undetected. Also, the bridge in the middle of the image is tough to classify, except for RGB only classifier. Classifier using both bands and indices manages to label most of the bridge, while other classifiers miss it entirely.

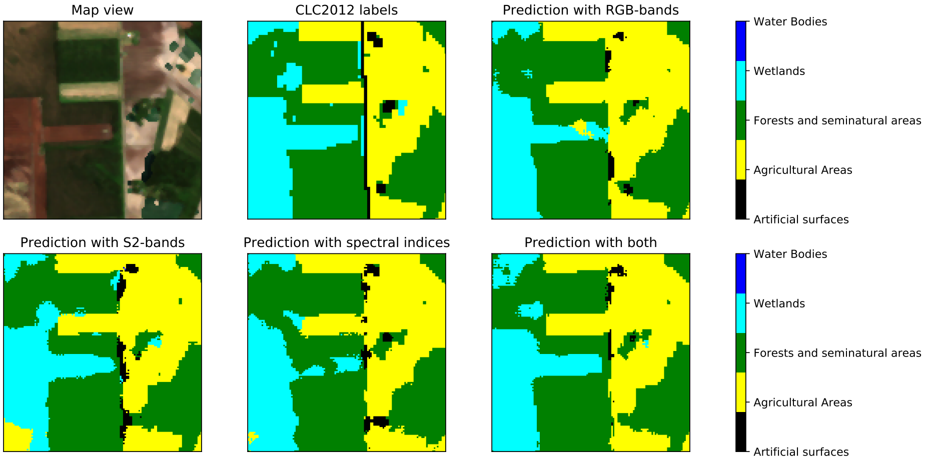


Figure 18. An example of differences with label borders for CLC level 1 classes

Figure 18 shows a situation where borders between different classes are more or less straight lines, and they are also separated by a straight road. Each classifier manages to give accurate predictions for borders between forests and agricultural areas, but borders between wetlands and forests are a lot trickier. Like before, the road on the middle remains undetected. Even though image has some anomalies from sen2three processing, they don’t seem to have any effect in classification.

Classification results for full dataset are presented in table 10. Even though micro and weighted overall scores are a bit better than for the test set, macro score is significantly worse. Only forests and water bodies have equal or better scores than for the test set, and all other labels have their scores decrease. One possible explanation for this is that full dataset and augmented test set have different proportions of data. For instance, test set has almost twice the amount of pixels labelled as wetlands than the full dataset. Nevertheless, overall accuracy is still almost 0.92, albeit mostly because of high accuracy for forests.

Confusion matrix (Figure 19) looks mostly identical to the one for the test set. About a third

Class	Pre	Rec	F1	IoU	Support
1	0.73	0.63	0.67	0.50	1385998
2	0.90	0.89	0.89	0.81	2430871
3	0.93	0.96	0.94	0.89	11445516
4	0.72	0.50	0.59	0.41	297871
5	0.97	0.97	0.97	0.94	3414808
Micro	0.92	0.92	0.92	-	18975064
Macro	0.85	0.79	0.81	0.71	18975064
Weighted	0.92	0.92	0.92	-	18975064

Table 10. Results for full image classification for CLC level 1 classes

of wetlands are classified as a forest, and around a quarter of artificial surfaces are confused with forests, most likely due to reasons stated previously. As expected, CLC level 1 classes are rather easy to classify correctly, even with minimal preprocessing or postprocessing.

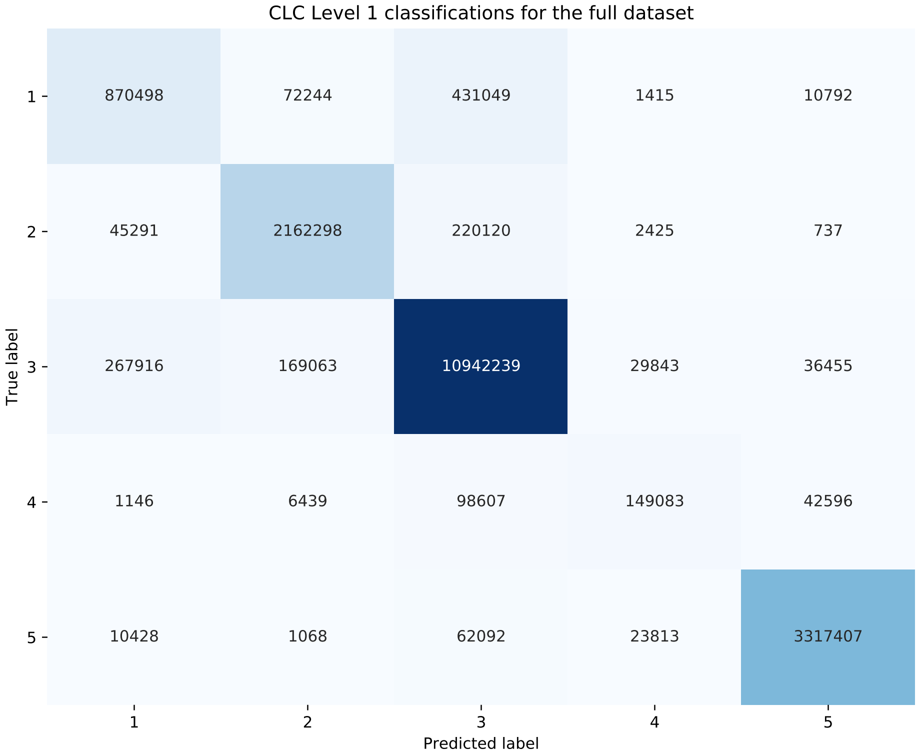


Figure 19. Confusion matrix for CLC level 1 full dataset classification

From full segmentation map, seen in figure 20, it can be observed that predictions match CLC 2012 labels pretty well. On the upper right corner of the image, few parts of lake Saimaa are incorrectly labelled as wetlands. This is constant for all classification levels and it is due to spectral index data being corrupted there. This behaviour is expanded more in section 5.2. Classification differences are visualized in figure 21, and as theorized previously, most of the misclassifications are either narrow roads or border regions between two different classes rather than large misclassified areas. It also seems that anomalies from sen2three processing don't affect classifications significantly. Also, as expected, borders of the labelled area have a lot of misclassifications due to missing spectral indices outside of them.

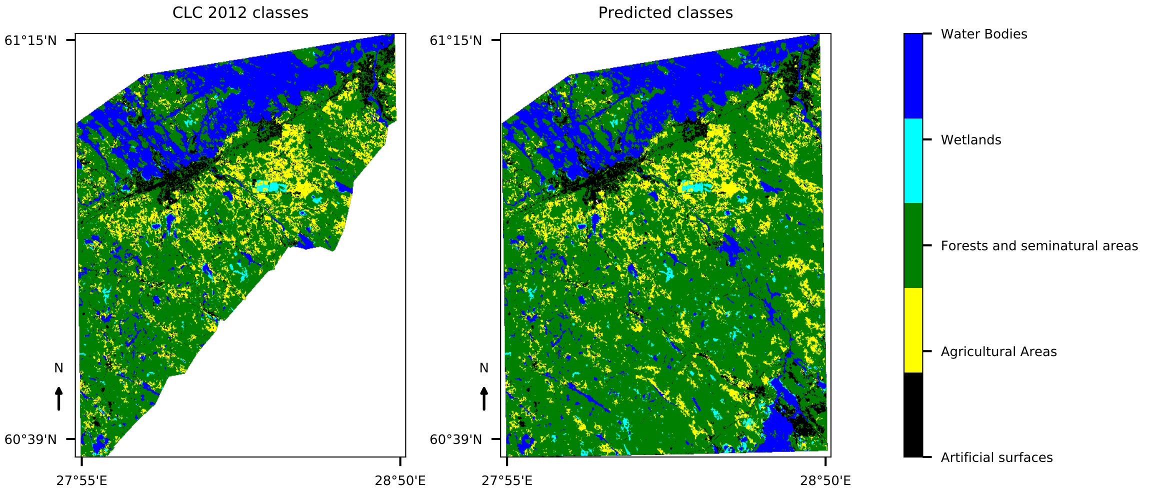


Figure 20. Left: CLC 2012 level 1 segmentation mask. Right: Predicted CLC level 1 segmentation mask

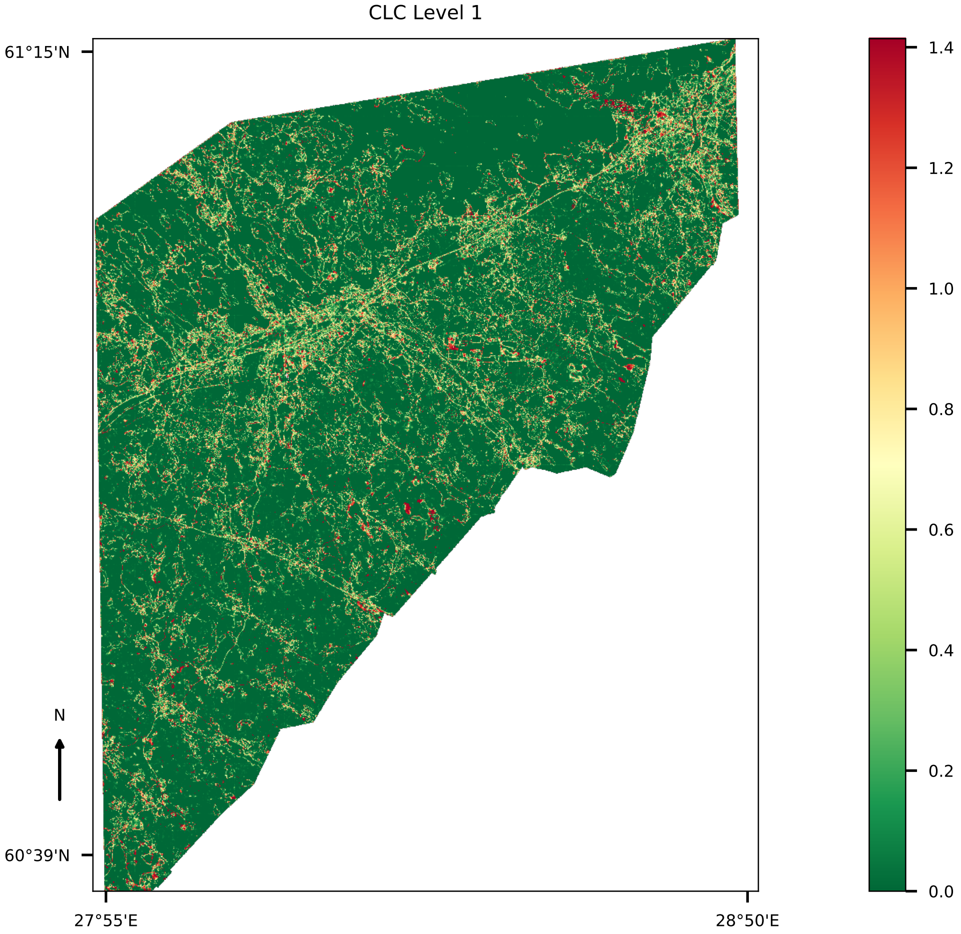


Figure 21. L2 distances between CLC 2012 labels and predictions.

4.2 Results for CLC level 2 classification

Identifier	Class	Support
1	Urban fabric	1140489
2	Industrial, commercial and transport units	1335162
3	Mine, dump and construction sites	238805
4	Artificial, non-agricultural vegetated areas	299605
5	Arable land	5533164
6	Permanent crops	17403
7	Pastures	14698
8	Heterogenous agricultural areas	225899
9	Forests	12915108
10	Scrub and/or herbaceous vegetation associations	3429604
11	Open spaces with little or no vegetation	14724
12	Inland wetlands	505891
13	Inland waters	3820648
	Total	**29491200**

Table 11. CLC level 2 classes in the test set

CLC level 2 classes are far more challenging than CLC level 1 classes. While ”Water bodies” and ”Inland wetlands” are equal to CLC level 1 classes in this area, other classes are divided into three or five subclasses. Also, three of these labels are almost non-existent both in the test set and even in the whole area, so they can be expected to be poorly classified.

	Training		Validation		Testing	
Features	Acc	Loss	Acc	Loss	Acc	Loss
RGB bands	0.85	0.39	0.82	0.52	0.82	0.52
All bands	0.85	0.41	0.83	0.49	0.82	0.50
Indices	0.86	0.38	0.82	0.50	0.82	0.50
Combined	0.86	0.38	0.84	0.45	0.84	0.45

Table 12. Accuracy and loss for different features in CLC level 2 classification

	RGB only				Bands only				Indices only				Combined			
	Pre	Rec	F1	IoU	Pre	Rec	F1	IoU	Pre	Rec	F1	IoU	Pre	Rec	F1	IoU
1	0.69	0.64	0.67	0.50	0.69	0.65	0.67	0.50	0.67	0.70	0.69	0.52	**0.69**	**0.70**	**0.69**	**0.53**
2	0.61	0.57	0.59	0.42	0.63	0.52	0.57	0.40	0.64	0.59	0.62	0.45	**0.64**	**0.62**	**0.63**	**0.46**
3	0.72	0.62	0.67	0.50	0.77	**0.63**	0.69	0.53	0.80	0.61	0.69	0.53	**0.84**	0.61	**0.70**	**0.54**
4	0.57	0.34	0.43	0.27	0.61	0.33	0.43	0.27	**0.64**	0.31	0.42	0.27	0.59	**0.37**	**0.46**	**0.30**
5	0.87	0.90	0.89	0.80	0.87	0.92	0.90	0.81	0.86	0.92	0.89	0.80	**0.90**	**0.93**	**0.91**	**0.84**
6	0.77	0.42	0.54	0.37	0.77	0.39	0.51	0.34	0.76	0.40	0.52	0.35	**0.83**	**0.42**	**0.56**	**0.39**
7	0.58	0.09	0.15	0.08	**0.87**	0.00	0.01	0.00	0.64	0.02	0.05	0.02	0.59	**0.18**	**0.28**	**0.16**
8	0.56	0.27	0.37	0.22	0.51	0.31	0.38	0.24	0.54	0.34	0.42	0.26	**0.57**	**0.34**	**0.43**	**0.27**
9	0.82	0.91	0.86	0.76	0.83	0.91	0.87	0.77	0.84	0.91	0.87	0.77	**0.85**	**0.91**	**0.88**	**0.78**
10	0.67	0.45	0.54	0.37	**0.68**	0.48	0.56	0.39	0.65	0.46	0.53	0.36	0.67	**0.52**	**0.58**	**0.41**
11	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
12	**0.84**	0.61	0.71	0.55	0.79	0.69	0.74	0.59	0.78	0.69	0.73	0.58	0.81	**0.70**	**0.75**	**0.60**
13	0.95	0.96	0.96	0.92	0.96	0.96	0.96	0.92	0.96	0.96	0.96	0.93	**0.97**	**0.97**	**0.97**	**0.94**
Mic	0.82	0.82	0.82	-	0.82	0.82	0.82	-	0.82	0.82	0.82	-	**0.84**	**0.84**	**0.84**	-
Mac	0.67	0.52	0.57	0.44	0.69	0.52	0.56	0.44	0.68	0.53	0.57	0.45	**0.69**	**0.56**	**0.60**	**0.48**
Wei	0.81	0.82	0.81	-	0.81	0.82	0.82	-	0.81	0.82	0.81	-	**0.83**	**0.84**	**0.83**	-

Table 13. Test set results for CLC level 2 classes

As expected, overall results (Tables 12 and 13) are worse than for CLC level 1 classifications, but not by a lot considering that there are over twice as many labels. Overall accuracies are all over 0.8, and as before network using all available features produces the best accuracies and lowest loss. Labelwise results, on the other hand, vary a lot. Easiest classes are still similar than before (”Water bodies”, ”Arable land” and ”Forests”), and interestingly ”Water bodies” are classified here more accurately than for CLC level 1 classes. Of the three most underrepresented classes, ”Open spaces with little or no vegetation” is totally undetected, while ”Pastures” has F1-score of 0.56 and IoU of 0.39 and ”Permanent crops” has scores 0.28 and 0.16 respectively.

Confusion matrices (Figure 22) show that most common mix-ups are between ”Forest” and ”Scrub and/or herbaceous vegetation associations”, which is understandable due to them both being aggregated from same CLC level 1 class. Misclassifications are similar for each network, but the exact amounts differ. For instance, no matter what features are used, around 50% of all ”Scrubs etc” are classified as a ”Forest”.

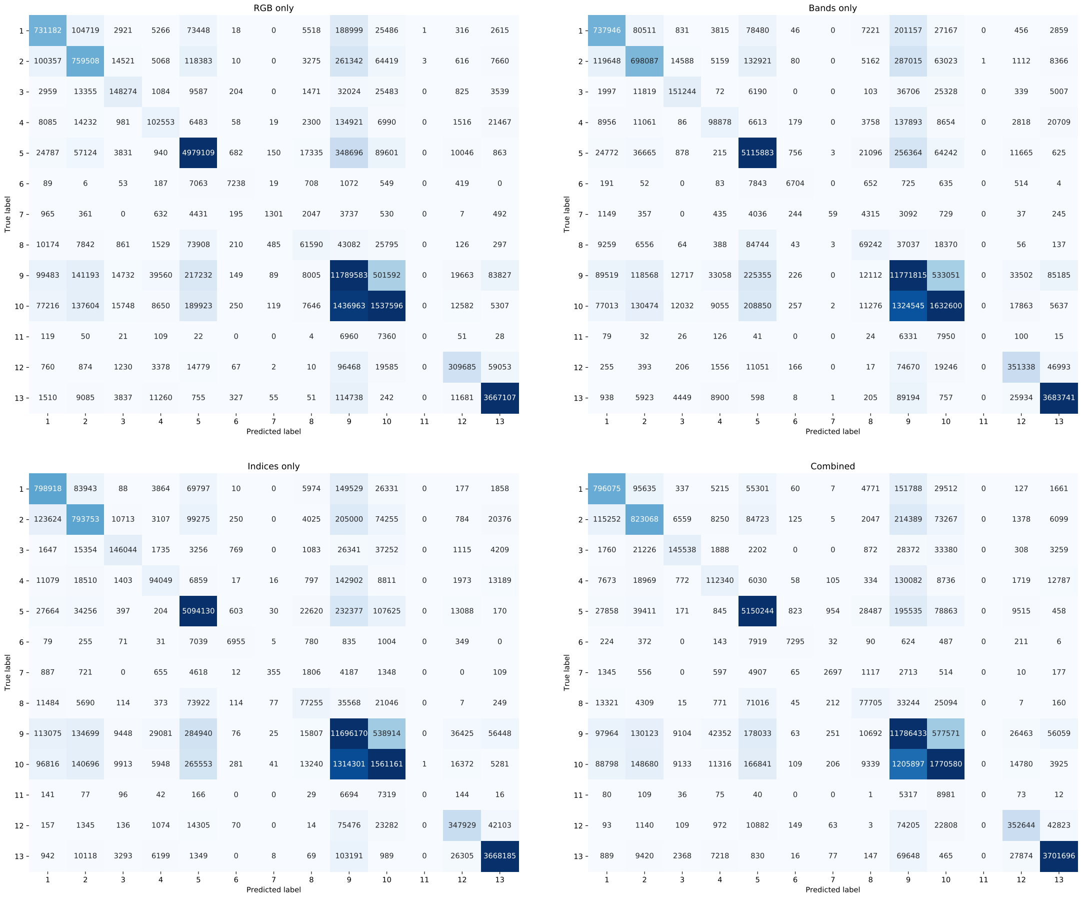


Figure 22. Confusion matrices for CLC level 2 classes

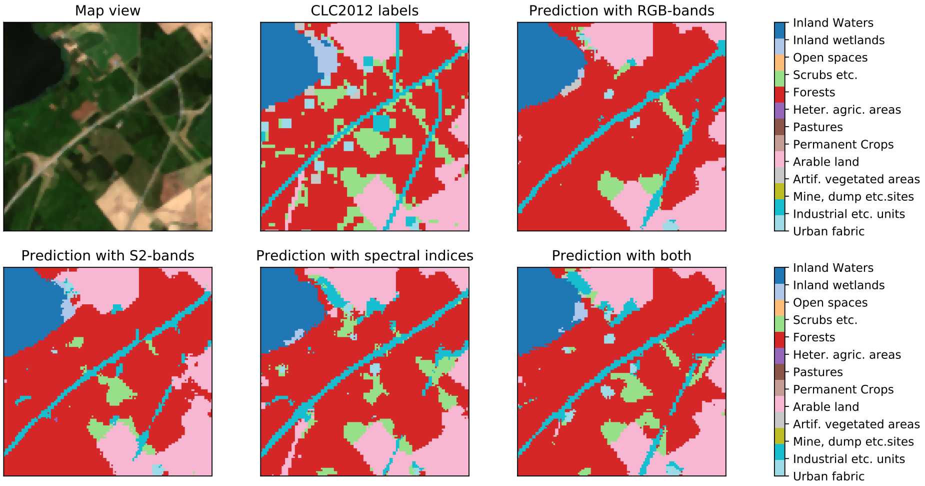


Figure 23. An example of CLC level 2 classifications in non-urban area

Figure 23 shows classifications for same image than figure 16, but for CLC level 2 classes, and the results are similar. The northernmost road stays undetected, middle road is detected but labelled to be wider than CLC 2012 labels suggest, and only some parts of the southernmost road are detected. Water bodies, forests and arable land are both labelled fairly correctly compared to CLC 2012 and satellite image.

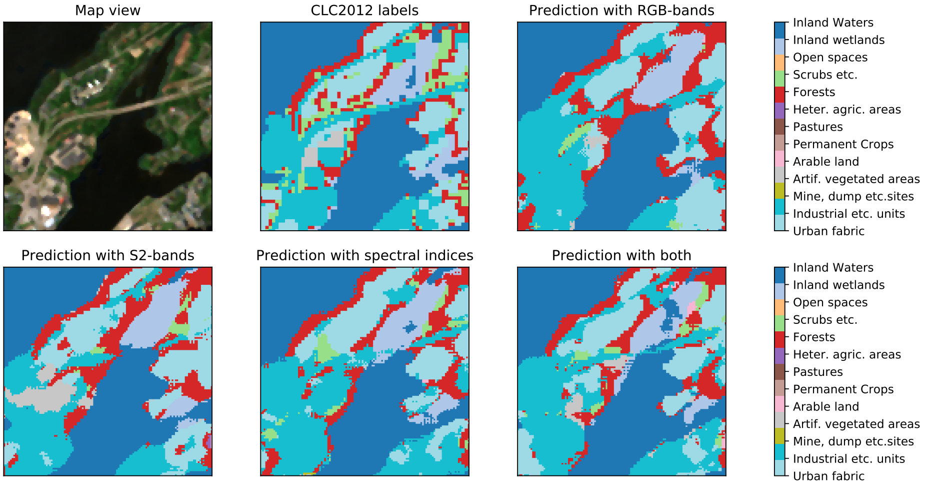


Figure 24. An example of CLC level 2 classifications in urban area

Likewise, figure 24 is from the same area than figure 17 but with different labels. Urban areas (”Urban fabric” and ”Industrial, commercial and transport units”) are still tough to classify similarly to for CLC2012 labels, and the bridge in the middle is still tough for most of the networks. While previously RGB only classifier was able to more or less detect this bridge and label it correcly, now it labels approximately half of it as ”Forest”. All classifiers agree, that there are ”Scrubs etc” at the lower-middle part of the image, above the arable land, but the size and shape of them differ from each other.

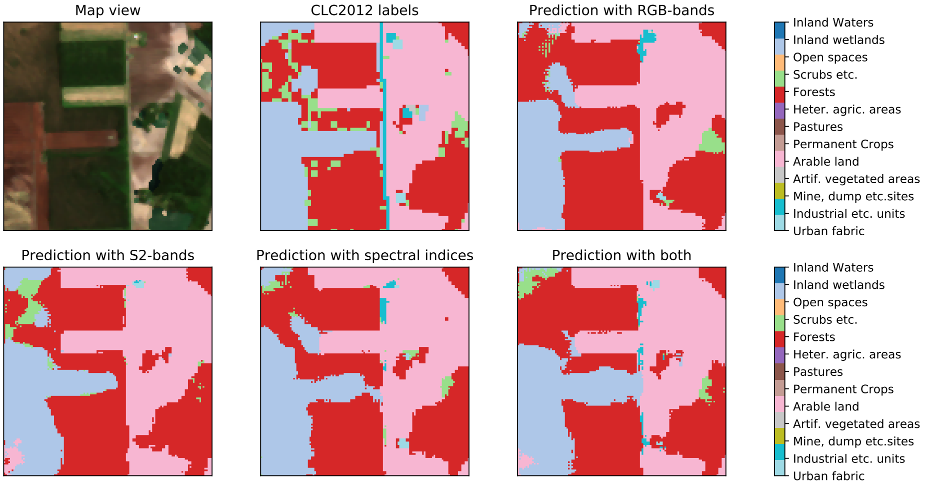


Figure 25. An example of differences with label borders for CLC level 2 classes

Scene with straight borders shows several differences between networks. Surprisingly RGB only classifier has the best result for wetlands, but it entirely misses the road running across the image. Only classifier that even somehow manages to detect the road is the one using all available features, but even that only finds less than half of it. Like for CLC level 1, small anomalies on the upper right corner don't seem to influence classification at all.

Full classification results are presented in table 14 and confusion matrix in figure 26. Compared to testing results, each class except "Forest" has worse performance in at least one metric when classifying full image. Also, micro and weighted overall accuracies are equal to testing results, but macro averages and mIoU show significant drop. Macro Rec and F1-score are now less than 0.5, but it can be explained by looking at support counts of the worst performing classes. Classes with F1-score less than 0.5 represent only 2.1% of all pixels, and are here even more underrepresented than in the test set (again, due to data augmentation). Other classes have rather moderate changes in their scores. Typically these classes are mixed with some similar, more common class, such as "Mine, dump and construction sites" are mixed with "Industrial, commercial and transport units". Reasons for their misclassification may be that the test set contains only the easiest instances to detect, and after augmentation there are multiple instances of those.

Overall segmentation map (27) looks at first glance mostly identical to CLC 2012 labels, with

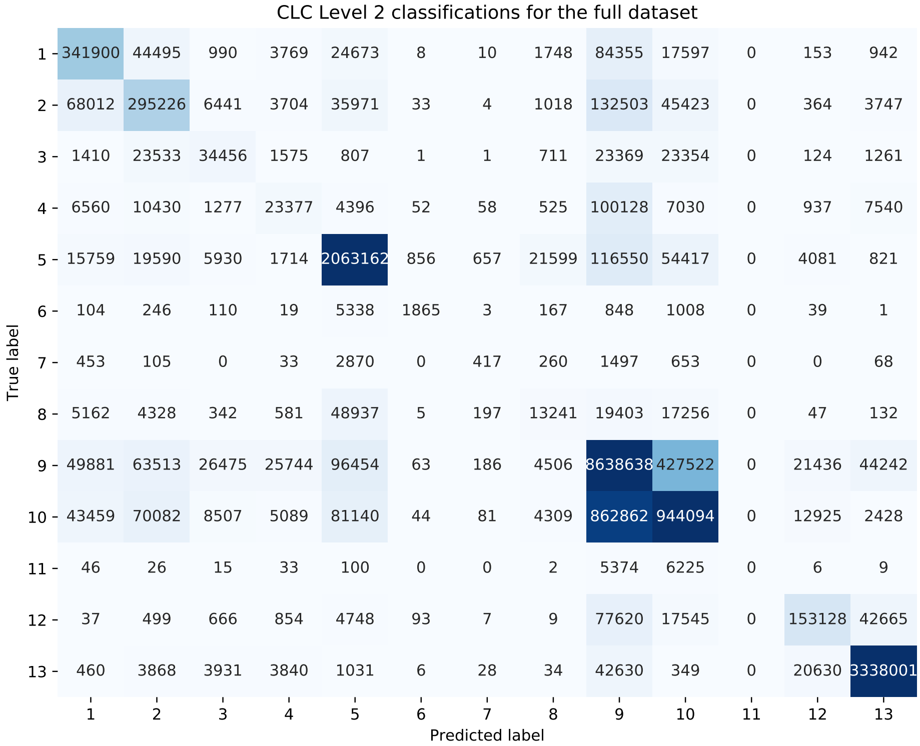
Class	Pre	Rec	F1	IoU	Support
1	0.64	0.66	0.65	0.48	520640
2	0.55	0.50	0.52	0.35	592446
3	0.39	0.31	0.35	0.21	110602
4	0.33	0.14	0.20	0.11	162310
5	0.87	0.90	0.88	0.79	2305136
6	0.62	0.19	0.29	0.17	9748
7	0.25	0.07	0.10	0.05	6356
8	0.28	0.12	0.17	0.09	109631
9	0.85	0.92	0.89	0.80	9398660
10	0.60	0.46	0.52	0.36	2035020
11	0.00	0.00	0.00	0.00	11836
12	0.72	0.51	0.60	0.43	297871
13	0.97	0.98	0.97	0.95	3414808
Micro	0.84	0.84	0.84	-	18975064
Macro	0.54	0.44	0.47	0.37	18975064
Weighted	0.82	0.84	0.83	-	18975064

Table 14. Results for full image classification for CLC level 2 classes

similar problems than CLC level 1 labels. Small, narrow areas are misclassified, although for some it may also be because of land cover change. Anomaly in the upper right corner is now smaller than previously, and interestingly classified as a ”Mine, dump and construction sites” instead of wetlands. Misclassified points are, again, mostly along the roads or between different land cover classes, with only few larger areas. These large areas are more common than for CLC level 1 classes, which is natural with over twice as many classes to predict.

4.3 Results for target level classification

Target level classes are in some ways tougher than CLC level 2 classes, mostly because of multiple different forest types. On the other hand, some rarer classes have been combined


CLC Level 2 classifications for the full dataset

True label \ Predicted label	1	2	3	4	5	6	7	8	9	10	11	12	13
1	341900	44495	990	3769	24673	8	10	1748	84355	17597	0	153	942
2	68012	295226	6441	3704	35971	33	4	1018	132503	45423	0	364	3747
3	1410	23533	34456	1575	807	1	1	711	23369	23354	0	124	1261
4	6560	10430	1277	23377	4396	52	58	525	100128	7030	0	937	7540
5	15759	19590	5930	1714	2063162	856	657	21599	116550	54417	0	4081	821
6	104	246	110	19	5338	1865	3	167	848	1008	0	39	1
7	453	105	0	33	2870	0	417	260	1497	653	0	0	68
8	5162	4328	342	581	48937	5	197	13241	19403	17256	0	47	132
9	49881	63513	26475	25744	96454	63	186	4506	8638638	427522	0	21436	44242
10	43459	70082	8507	5089	81140	44	81	4309	862862	944094	0	12925	2428
11	46	26	15	33	100	0	0	2	5374	6225	0	6	9
12	37	499	666	854	4748	93	7	9	77620	17545	0	153128	42665
13	460	3868	3931	3840	1031	6	28	34	42630	349	0	20630	3338001

Figure 26. Confusion matrix for CLC level 2 full dataset classification

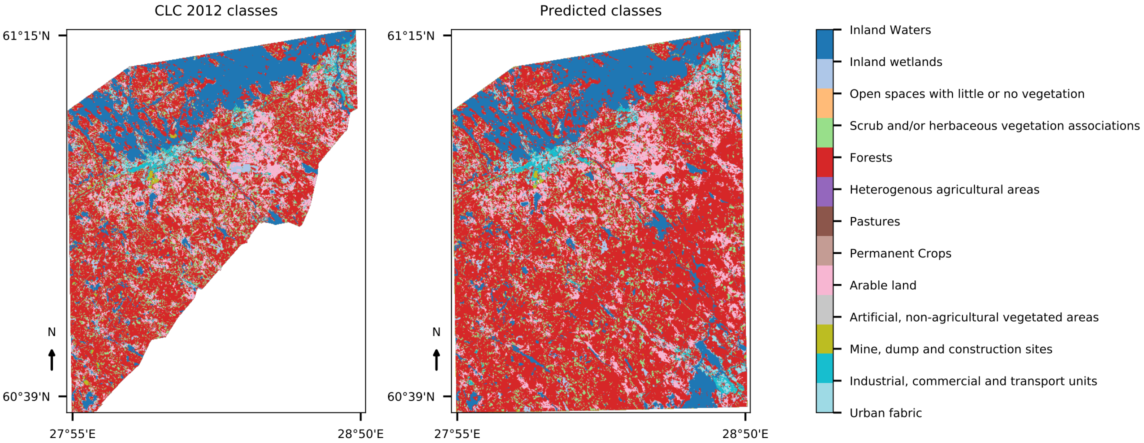

CLC 2012 classes
Predicted classes
61°15'N
60°39'N
27°55'E
28°50'E
N
61°15'N
60°39'N
27°55'E
28°50'E
N
Inland Waters
Inland wetlands
Open spaces with little or no vegetation
Scrub and/or herbaceous vegetation associations
Forests
Heterogenous agricultural areas
Pastures
Permanent Crops
Arable land
Artificial, non-agricultural vegetated areas
Mine, dump and construction sites
Industrial, commercial and transport units
Urban fabric

Figure 27. Left: CLC 2012 level 2 segmentation mask. Right: Predicted CLC level 2 segmentation mask

into one and total number of classes is smaller. Comparing these results straight to other predictions is not straightforward, because as stated before target level classes are not aggregated from other CLC classes, but rather combine several of them.

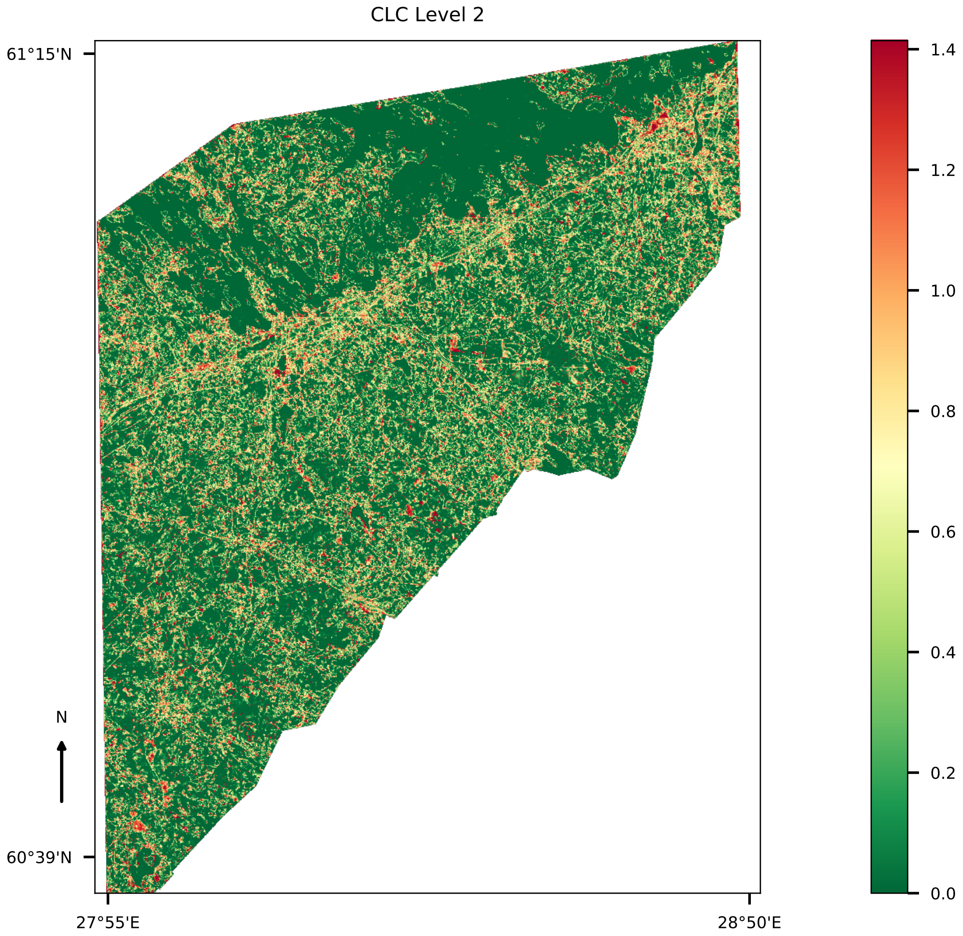


Figure 28. L2 distances between CLC 2012 labels and level 2 predictions.

Both accuracy and loss scores are worse for these classes than for CLC level 2 classes, with best overall accuracy and F1-score for test set both being 0.76. Even though overall accuracy and F1-score are lower than for CLC level 2 classes, mean IoU score and overall macro averages are better. One possible explanation for this is different number of forest type classes, as well as predicted results for CLC level 2 omitting some classes totally. Like for other classification levels, best results are acquired by using all Sentinel-2 bands and precomputed spectral indices and worst results with only RGB bands.

Best performing classes follow the trend from previous classifications. Water bodies are still the easiest ones to classify, followed by fields and then coniferous forest. Hardest classes are broad-leaved forest and mixed forest, followed by grasslands. Overall, F1-scores and

Identifier	Class	Support
1	Built-up areas	2655204
2	Bare areas	411277
3	Grasslands	378052
4	Fields	5533164
5	Broad-leaved forest	1190813
6	Coniferous forest	8390724
7	Mixed forest	3333571
8	Transitional woodland shrub	3429604
9	Inland marshes	189864
10	Water vegetation	158279
11	Water bodies	3820648
	Total	**29491200**

Table 15. Target level classes in the test set

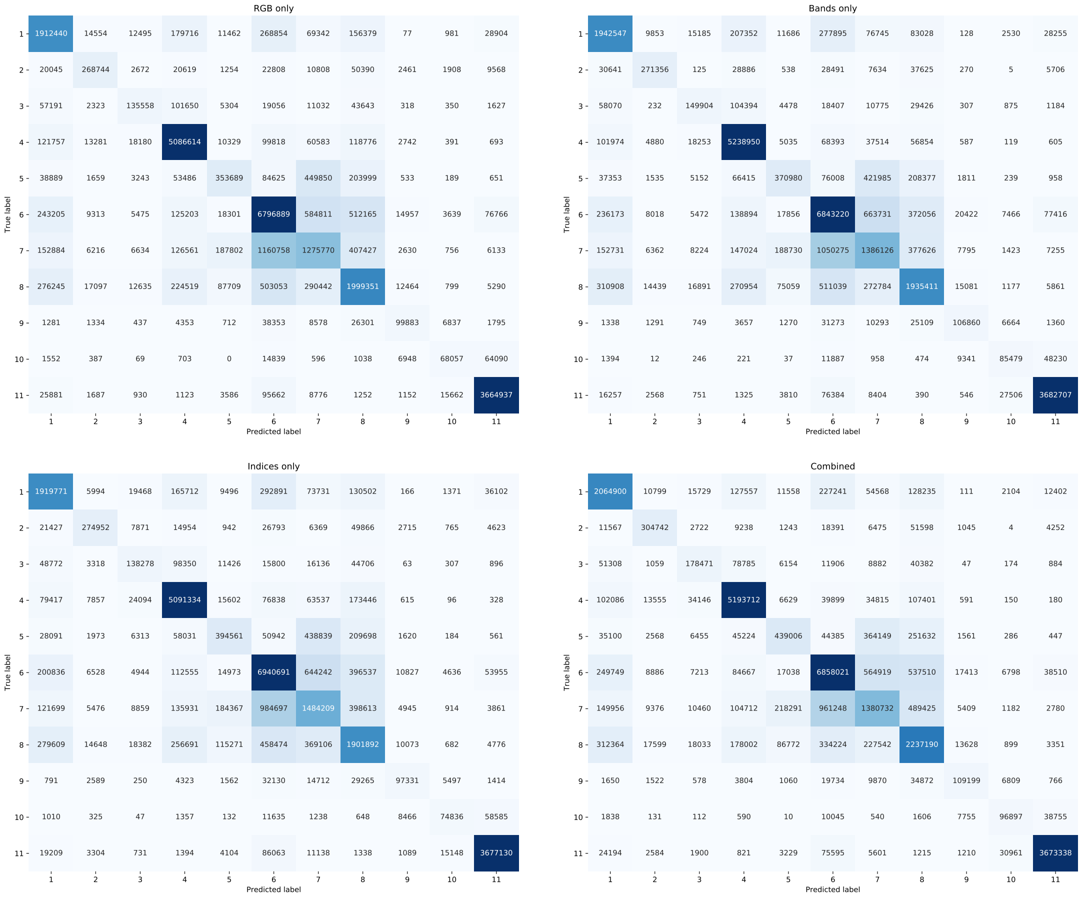
	Training		Validation		Testing	
Features	Acc	Loss	Acc	Loss	Acc	Loss
RGB bands	0.77	0.61	0.74	0.73	0.73	0.74
S2 bands	0.77	0.61	0.75	0.70	0.75	0.70
Indices	0.78	0.59	0.75	0.71	0.75	0.71
Combined	0.79	0.58	0.76	0.65	0.76	0.65

Table 16. Accuracy and loss for different features in target level classification

IoU-scores are much more even than for CLC level 2 classes, partly due to there not being any extremely underrepresented class among target classes. Also, highest precision score for each class is at least 0.5. Given that class imbalance was not taken into account when training neural networks, these results are promising. All results are presented in tables 16 and 17.

	RGB only				Bands only				Indices only				Combined			
	Pre	Rec	F1	IoU	Pre	Rec	F1	IoU	Pre	Rec	F1	IoU	Pre	Rec	F1	IoU
1	0.67	0.72	0.69	0.53	0.67	0.73	0.70	0.54	**0.71**	0.72	0.71	0.56	0.69	**0.78**	**0.73**	**0.57**
2	0.80	0.65	0.72	0.56	**0.85**	0.66	0.74	0.59	0.84	0.67	0.74	0.59	0.82	**0.74**	**0.78**	**0.64**
3	0.68	0.36	0.47	0.31	**0.68**	0.40	0.50	0.33	0.60	0.37	0.46	0.29	0.65	**0.47**	**0.55**	**0.38**
4	0.86	0.92	0.89	0.80	0.84	0.95	0.89	0.81	0.86	0.92	0.89	0.80	**0.89**	**0.94**	**0.91**	**0.84**
5	0.52	0.30	0.38	0.23	0.55	0.31	0.40	0.25	0.52	0.33	0.41	0.25	**0.56**	**0.37**	**0.44**	**0.28**
6	0.75	0.81	0.78	0.64	0.76	0.82	0.79	0.65	0.77	**0.83**	0.80	0.67	**0.80**	0.82	**0.81**	**0.68**
7	0.46	0.38	0.42	0.26	0.48	0.42	0.44	0.29	0.48	**0.45**	**0.46**	0.30	**0.52**	0.41	0.46	**0.30**
8	0.57	0.58	0.58	0.40	**0.62**	0.56	0.59	0.42	0.57	0.55	0.56	0.39	0.58	**0.65**	**0.61**	**0.44**
9	0.69	0.53	0.60	0.43	0.65	0.56	0.61	0.43	0.71	0.51	0.59	0.42	**0.69**	**0.58**	**0.63**	**0.46**
10	0.68	0.43	0.53	0.36	0.64	0.54	0.59	0.41	**0.72**	0.47	0.57	0.40	0.66	**0.61**	**0.64**	**0.47**
11	0.95	0.96	0.95	0.91	0.95	0.96	0.96	0.92	0.96	0.96	0.96	0.92	**0.97**	**0.96**	**0.97**	**0.94**
Mic	0.73	0.73	0.73	-	0.75	0.75	0.75	-	0.75	0.75	0.75	-	**0.76**	**0.76**	**0.76**	-
Mac	0.69	0.60	0.64	0.49	0.70	0.63	0.65	0.51	0.70	0.62	0.65	0.51	**0.71**	**0.67**	**0.68**	**0.55**
Wei	0.72	0.73	0.73	-	0.74	0.75	0.74	-	0.74	0.75	0.74	-	**0.76**	**0.76**	**0.76**	-

Table 17. Test set results for target level classification



RGB only

	1	2	3	4	5	6	7	8	9	10	11
1	1912440	14554	12495	179716	11462	268854	69342	156379	77	981	28904
2	20045	268744	2672	20619	1254	22808	10808	50390	2461	1908	9568
3	57191	2323	135558	101650	5304	19056	11032	43643	318	350	1627
4	121757	13281	18180	5086614	10329	99818	60583	118776	2742	391	693
5	38889	1659	3243	53486	353689	84625	449850	203999	533	189	651
6	243205	9313	5475	125203	18301	6796889	584811	512165	14957	3639	76766
7	152884	6216	6634	126561	187802	1160758	1275770	407427	2630	756	6133
8	276245	17097	12635	224519	87709	503053	290442	1999351	12464	799	5290
9	1281	1334	437	4353	712	38353	8578	26301	99883	6837	1795
10	1552	387	69	703	0	14839	596	1038	6948	68057	64090
11	25881	1687	930	1123	3586	95662	8776	1252	1152	15662	3664937

Bands only

	1	2	3	4	5	6	7	8	9	10	11
1	1942547	9853	15185	207352	11686	277895	76745	83028	128	2530	28255
2	30641	271356	125	28886	538	28491	7634	37625	270	5	5706
3	58070	232	149904	104394	4478	18407	10775	29426	307	875	1184
4	101974	4880	18253	5238950	5035	68393	37514	56854	587	119	605
5	37353	1535	5152	66415	370980	76008	421985	208377	1811	239	958
6	236173	8018	5472	138894	17856	6843220	663731	372056	20422	7466	77416
7	152731	6362	8224	147024	188730	1050275	1386126	377626	7795	1423	7255
8	310908	14439	16891	270954	75059	511039	272784	1935411	15081	1177	5861
9	1338	1291	749	3657	1270	31273	10293	25109	106860	6664	1360
10	1394	12	246	221	37	11887	958	474	9341	85479	48230
11	16257	2568	751	1325	3810	76384	8404	390	546	27506	3682707

Indices only

	1	2	3	4	5	6	7	8	9	10	11
1	1919771	5994	19468	165712	9496	292891	73731	130502	166	1371	36102
2	21427	274952	7871	14954	942	26793	6369	49866	2715	765	4623
3	48772	3318	138278	98350	11426	15800	16136	44706	63	307	896
4	79417	7857	24094	5091334	15602	76838	63537	173446	615	96	328
5	28091	1973	6313	58031	394561	50942	438839	209698	1620	184	561
6	200836	6528	4944	112555	14973	6940691	644242	396537	10827	4636	53955
7	121699	5476	8859	135931	184367	984697	1484209	398613	4945	914	3861
8	279609	14648	18382	256691	115271	458474	369106	1901892	10073	682	4776
9	791	2589	250	4323	1562	32130	14712	29265	97331	5497	1414
10	1010	325	47	1357	132	11635	1238	648	8466	74836	58585
11	19209	3304	731	1394	4104	86063	11138	1338	1089	15148	3677130

Combined

	1	2	3	4	5	6	7	8	9	10	11
1	2064900	10799	15729	127557	11558	227241	54568	128235	111	2104	12402
2	11567	304742	2722	9238	1243	18391	6475	51598	1045	4	4252
3	51308	1059	178471	78785	6154	11906	8882	40382	47	174	884
4	102086	13555	34146	5193712	6629	39899	34815	107401	591	150	180
5	35100	2568	6455	45224	439006	44385	364149	251632	1561	286	447
6	249749	8886	7213	84667	17038	6858021	564919	537510	17413	6798	38510
7	149956	9376	10460	104712	218291	961248	1380732	489425	5409	1182	2780
8	312364	17599	18033	178002	86772	334224	227542	2237190	13628	899	3351
9	1650	1522	578	3804	1060	19734	9870	34872	109199	6809	766
10	1838	131	112	590	10	10045	540	1606	7755	96897	38755
11	24194	2584	1900	821	3229	75595	5601	1215	1210	30961	3673338

Figure 29. Confusion matrices for target level classes

Confusion matrixes (Figure 29) show clear mix-ups between different forest types. Broad-leaved forest is most commonly confused with mixed forest or transitional woodland shrub, but each network manages to differentiate between broad-leaved and coniferous forest surprisingly well. Mixed forest and coniferous forest are sometimes confused with each other, with mixed forest being labelled as coniferous forest more often than vice versa. Otherwise matrices show similar results than previous ones: classes that are typically next to each other are also mixed up (for instance built-up areas and coniferous forest), while classes that are geographically far from each other are not (for instance bare areas and water vegetation).

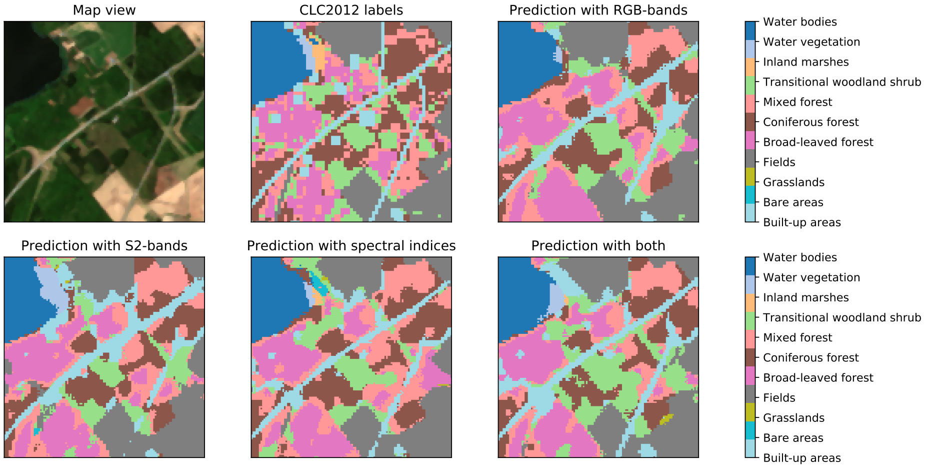


Figure 30. An example of target level classifications in non-urban area

As seen in figure 30, similar challenges than before occur for target level classifiers. Roads are still labelled wider than CLC 2012 labels and only middle road is found constantly. However, it seems that this level finds the other roads better than other levels, at least when all available features are used. Almost all of the bottom road is labelled, along with approximately half of the topmost one. Different forest types are tough to distinguishing from satellite image, but all classifiers somewhat agree on them, and compared to CLC2012 they look believable. One thing common in all classification levels for this scene is that network using all available features detects large area labelled either ”Artificial surfaces”, ”Industrial, commercial and transport units” or ”Built-up areas” on the upper-right beach area, while according to CLC2012 labels that area is really marshland.

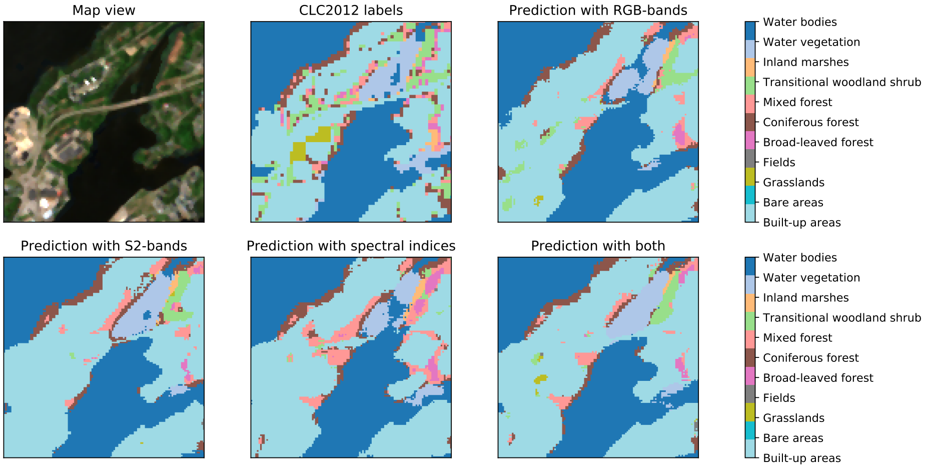


Figure 31. An example of target level classifications in urban area

Comparing figure 31 to figures 17 and 24, similar challenges as before can be observed. Urban areas are still classified to be larger clusters that hide smaller non-urban locations inside them. This behaviour is even more noticeable for this classification level. Forests between urban area and water are more narrow than for either CLC level, as are forest locations inside built-up areas. Network using all features is able to detect some grasslands inside built-up areas that remain undetected for other networks. It is also able to find the bridge on the middle and label it correctly, whereas for instance network using S2 bands labels a part of it as ”Coniferous forest”

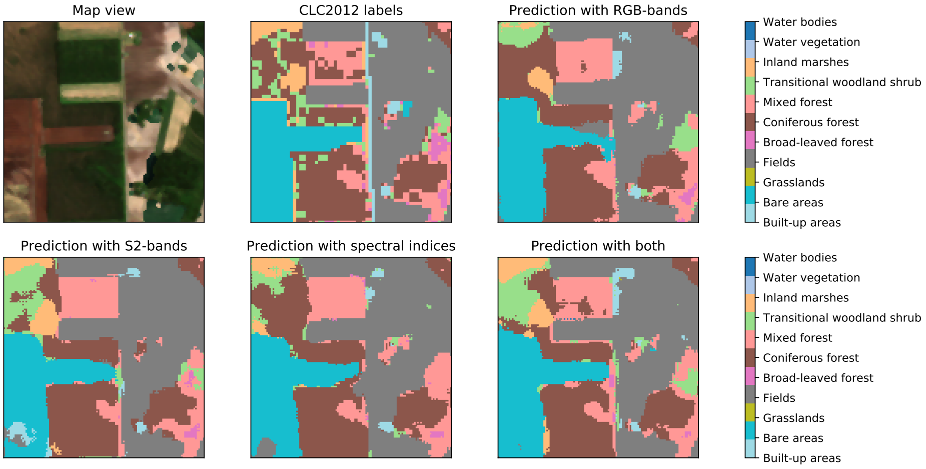


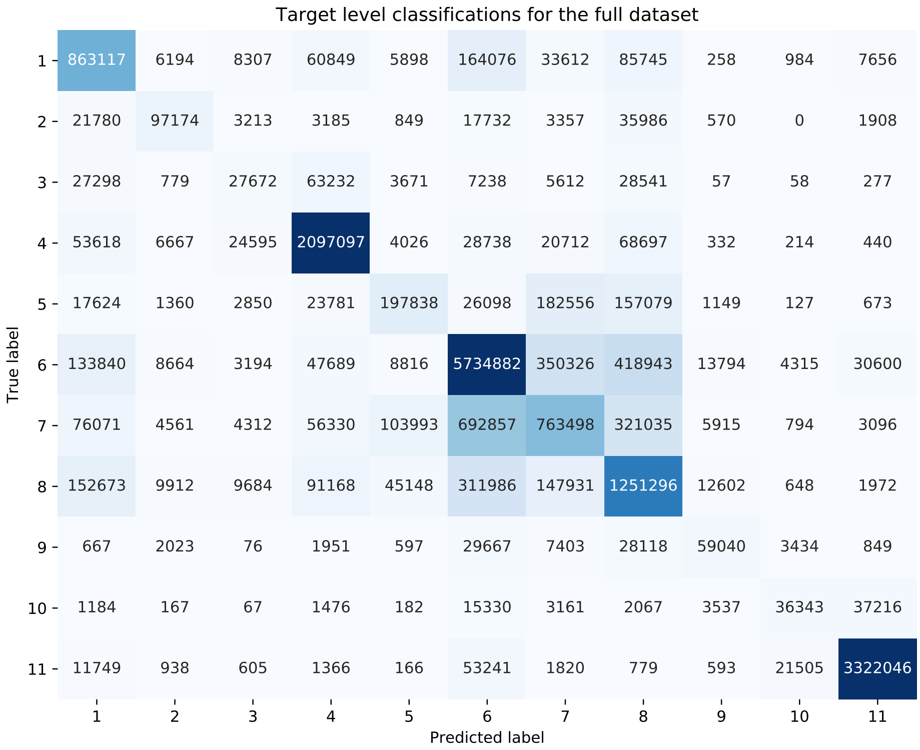
Figure 32. An example of differences with label borders for target level classes

In figure 32, peat production area has before been in the same class than marshes near it, but for target level it has been moved to the class ”Bare areas”. Most of the classifiers manage to distinguish between these classes and also find the borders between marsh and different forests. One thing that remains constant for each classification level is the inability to detect the road running in the middle. Networks here also differ in forest labels, with RGB only and index only predictions being mostly coniferous forest and others having noticeably more transitional woodland shrub. All networks seem to agree, that smaller rectangular areas of ”Transitional woodland shrub” are non-existent inside forest located on the lower-middle part of this scene.

Class	Pre	Rec	F1	IoU	Support
1	0.63	0.70	0.66	0.50	1236696
2	0.70	0.52	0.60	0.43	185754
3	0.33	0.17	0.22	0.13	164435
4	0.86	0.91	0.88	0.79	2305136
5	0.53	0.32	0.40	0.25	611135
6	0.81	0.85	0.83	0.71	6755063
7	0.50	0.38	0.43	0.27	2032462
8	0.52	0.61	0.56	0.39	2035020
9	0.60	0.44	0.51	0.34	133825
10	0.53	0.36	0.43	0.27	100730
11	0.98	0.97	0.97	0.94	3414808
Micro	0.76	0.76	0.76	-	18975064
Macro	0.64	0.57	0.59	0.46	18975064
Weighted	0.75	0.76	0.75	-	18975064

Table 18. Results for full image classification for target level classes

Again, full image classification has at least one metric deteriorate for most of the classes, and as before the classes with the worst performance are typically those with lowest support count. Micro and weighted averages are similar than before, but macro averages are significantly better than for CLC level 2 classes. Out of classes that are identical with some CLC level 2 levels, namely ”Fields” and ”Water bodies”, performance is almost identical for both networks. Like for the test set, broad-leaved and coniferous forests are distinguished from each other pretty well, and also other forest type mix ups are still similar.



Target level classifications for the full dataset

True label \ Predicted label	1	2	3	4	5	6	7	8	9	10	11
1	863117	6194	8307	60849	5898	164076	33612	85745	258	984	7656
2	21780	97174	3213	3185	849	17732	3357	35986	570	0	1908
3	27298	779	27672	63232	3671	7238	5612	28541	57	58	277
4	53618	6667	24595	2097097	4026	28738	20712	68697	332	214	440
5	17624	1360	2850	23781	197838	26098	182556	157079	1149	127	673
6	133840	8664	3194	47689	8816	5734882	350326	418943	13794	4315	30600
7	76071	4561	4312	56330	103993	692857	763498	321035	5915	794	3096
8	152673	9912	9684	91168	45148	311986	147931	1251296	12602	648	1972
9	667	2023	76	1951	597	29667	7403	28118	59040	3434	849
10	1184	167	67	1476	182	15330	3161	2067	3537	36343	37216
11	11749	938	605	1366	166	53241	1820	779	593	21505	3322046

Figure 33. Confusion matrix for target level full dataset classification

Predicted segmentation map is, at least for the main features (water bodies, urban areas etc) similar to CLC 2012 labels. Similar classification differences can still be detected than for example in CLC level 2 classification. On the western side of Lappeenranta there’s an area that according to CLC 2012 labels is ”Bare areas”, but is predicted as ”Built-up areas” (Figure 35). For CLC level 2, this classification change was from ”Mine, dump and construction sites” into ”Industrial, commercial and transport units”. This may indicate that there was a construction zone in the time of CLC 2012 project which has now finished. Interestingly, misclassification in the upper right corner of the segmentation map is smallest for this classification level.

Because overall accuracy for this level is about 75%, much more than only border regions between different classes are misclassified. Because CLC 2012 labels have a large number

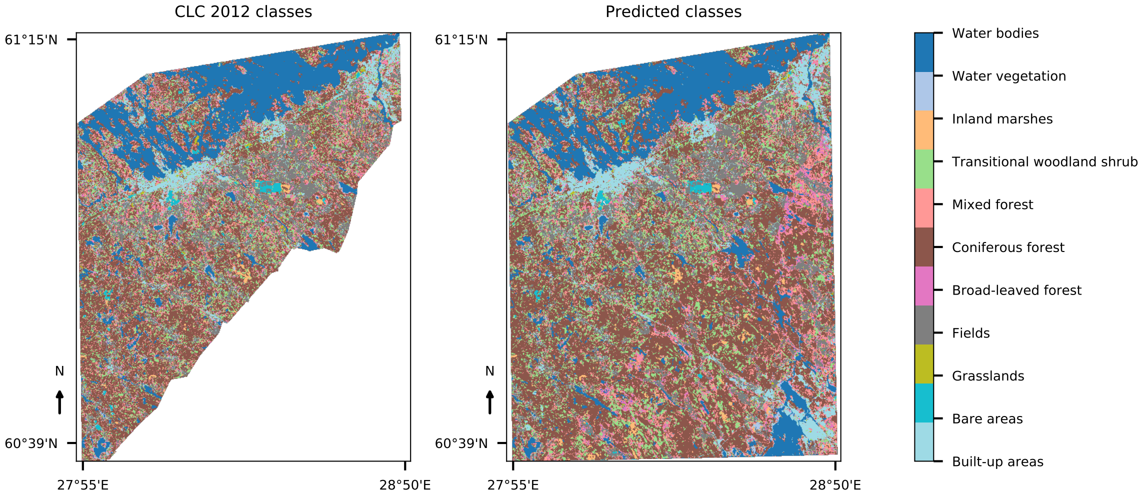


Figure 34. Left: CLC 2012 target level segmentation mask. Right: Predicted target level segmentation mask

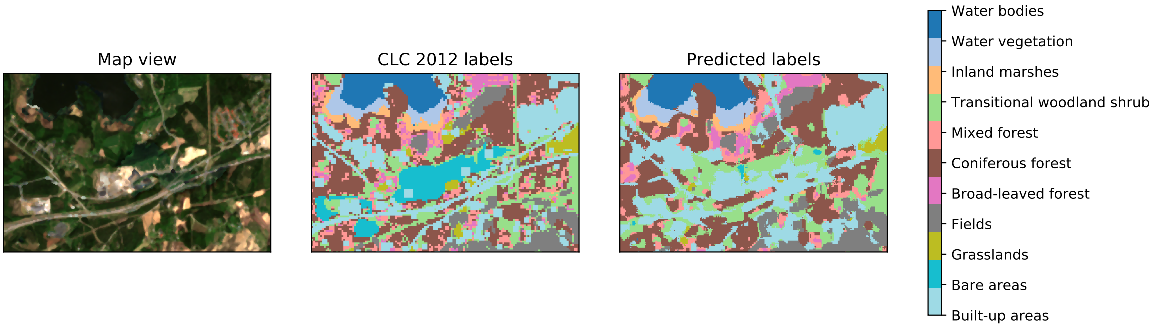


Figure 35. Example of classification differences that can be explained with land cover change

of lone pixels of one forest type inside another (typically broad-leaved or mixed forest inside coniferous forest) and network used here has problems with narrow areas, many misclassifications are made because of this. In addition, for this classification level there are also larger areas that are classified differently than in CLC 2012 labels. A lot of misclassifications are located in the areas between water and land, possibly due to ”Water vegetation” differences.

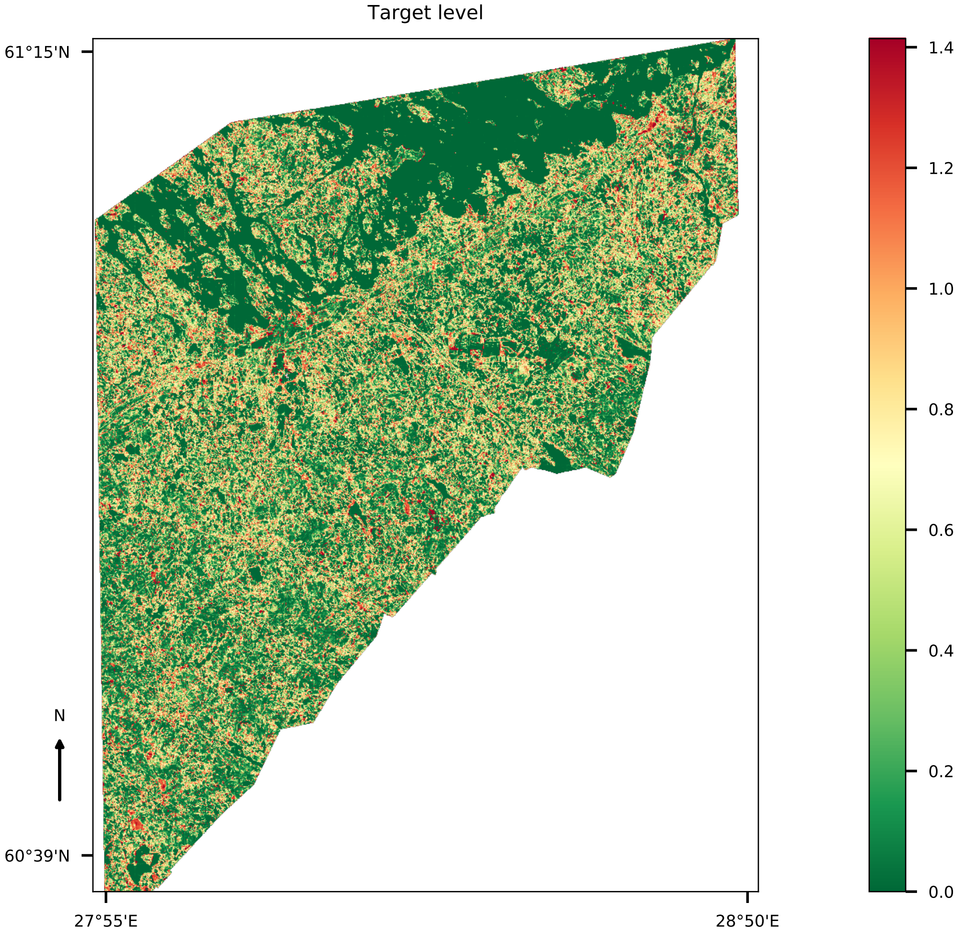


Figure 36. L2 distances between CLC 2012 labels and target level predictions.

5 Discussion

In this chapter, the conclusions from acquired results are presented briefly, and possible ways to improve classification results are theorized. Ways to improve are based on recent research of utilization of deep learning in remote sensing tasks. Challenges for this kind of dataset are also discussed, and some ideas about how to solve them are presented.

5.1 Advantages in using multiple features

Neural networks using all available features outperformed other tested neural networks for all classification levels, and networks using only red, green and blue S2 bands had the worst performance. Amount of improvement depends on the classification level, but overall F1-scores improved by one or two percent and IoU improved by two to four percent points. Networks using only spectral indices or only all S2 bands had pretty much identical overall results, and both outperform other for some classes and some metrics. Typically spectral index networks yield better recall, while S2 band network had better precision scores and thus their F1-scores are similar.

Comparing RGB classifications and S2 band classifications, there is no single class that shows clear benefit from using more features, but rather all classes improve by a few percent points. Adding spectral indices to features doesn't change this behavior, and again all class scores improve by a couple of percent points. Largest improvement for F1-score when moving from S2 bands to S2 bands and indices is 6%, which occurs for CLC level 2 class ”Industrial, commercial and transport units”. However, difference between index only and combined classifier for this class is only 1%. It is worth to keep in mind, that these results are acquired with pretty much untuned neural network, and section 5.3 focuses on how to improve further.

5.2 Challenges of the used dataset

As stated in 3.2.1, training data labels are not 100% accurate, and estimated accuracy drops quickly when classification level becomes more specific. In addition to this, land cover classification sets are always extremely unbalanced and dataset used here is no exception. For instance, in Finland almost 75% of all land cover belongs to CLC level 1 class "Forests and seminatural areas" (Härmä et al. 2014). For the dataset used in this thesis, even though labels are provided in 10×10m spatial resolution, these labels are produced in 20×20m resolution and then resampled to 10×10m resolution. 3.2.1 also presents some other problems with original labels, such as faulty built-up area labels. These factors, along with possible land cover changes after initial classification, make analyzing the results very difficult. For example, while I can be certain that water labels are almost 100% correct (some borders between water and land may be incorrectly classified due to different resolution), same can't be said for agricultural areas.

One of the reasons to use deep learning methods for land cover classification is their robustness to labelling errors. Rolnick et al. (2017) show that instead of just memorizing noise, CNNs are able to generalize and acquire over 90% accuracy on MNIST dataset even with 100 randomly labelled images for each correctly labelled one. Of course, this is true only when training set is sufficiently large, and having more errors also means that batch size should be decreased.

As with all remote sensing tasks using satellite data, there's always the problem with clouds and cloud shadows. Even though I had over 30 satellite images to choose from, only two of three of them were even close to being usable. In this thesis this was solved by generating synthetic cloudless mosaic from several atmospheric corrected images, but for accurate, low-interval LULC monitoring it would be better to be able to classify images with clouds. After all, S2A and S2B produce images every five days, so theoretically almost real-time monitoring is possible. Sen2cor produces cloud confidence masks that give each pixel some probability value of containing clouds, so it could be possible to set all points with cloud confidence value larger than some threshold. Sen2cor version 2.4 however has a known problem of assigning large cloud confidence values for built-up areas, so it might mask non-clouded, built-up areas in addition to clouds. This has been partially fixed in later versions (European

Space Agency 2018a). It may be that clouds and built-up areas have similar spectral signatures, because networks trained here always classify clouds as some artificial surface, as shown in figure 37. Another way is to train a network with images containing clouds, and classify them as their own class ”Clouds or unknown”.

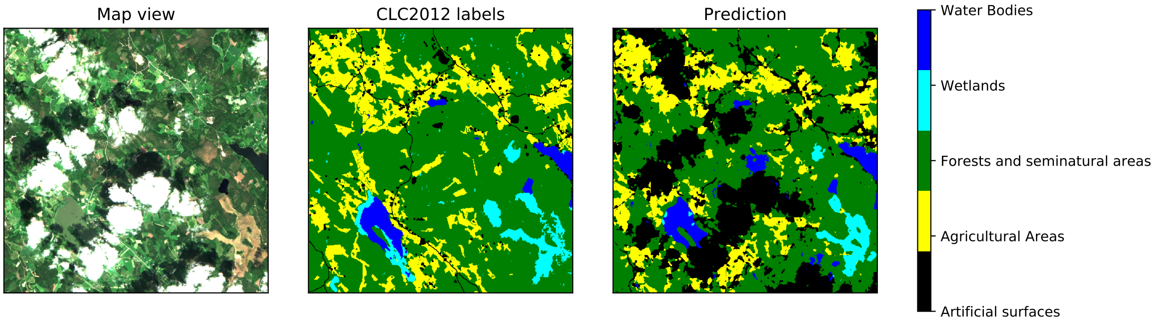


Figure 37. Example classification of a clouded scene

5.3 How to improve classification results

In this section ways to improve these results are theorized and presented, both from neural network tuning and input data perspective. While the results presented above show that this kind of method is suitable for LULC from multispectral images, but how could these results be improved? Common baseline for LULC accuracy assessment is 85% overall accuracy with no class having accuracy less than 70% (Foody 2002), and presented method doesn’t quite reach that even for CLC level 1 classes. Note that while presented neural network classifies multiple classes at once, but another possible method is to form an ensemble of classifier that each perform binary classification for some of the classes. Binary classification is in some ways easier to evaluate than multiclass classification, because for instance ROC (*receiver operating charasteristic*) curve and AUC (*Area under curve*) metrics are well defined and easier to use for binary classification. On the other hand, multiclass classification confusion matrices give clear information about which classes are hard to distinguish from each other. Nevertheless, ensemble classification might improve results.

Even though proposed neural networks acquired rather good results, at least for the easiest classification level, in all honesty they are completely untuned for this task. Used loss function, unweighted categorical cross entropy, is not well-suited for datasets with an extreme imbalance like in dataset used here. This can be observed from the results, where larger classes have better and smaller correspondingly worse results. By changing the loss function to be either weighted categorical cross entropy or to either recently introduced loss functions *focal loss* (Lin et al. 2017) or Lovasz-Softmax loss (Matthew and Blaschko 2018) and retraining networks it is likely that better classification results are achieved. How much do they improve is uncertain and worth testing.

Another way to improve network performance is to use pretrained weights from some well-performing model in the encoding path rather than train networks from scratch. For instance, Nogueira, Penatti, and Santos (2017) show that using pretrained model and then fine tuning it produces better results than fully training a new network. This approach has also the advantage of being a lot less time-consuming and computational resource intensive, because now only the transposed convolutions and final layers need to be trained from scratch.

A vital part of neural network implementation and training is the choice of hyperparameters. Presented network mostly follows original U-Net identically with only slight modifications under the assumption that developers of U-Net have tuned this architecture well enough. Still, possible ways to improve training process is to adjust some of these, such as learning rate and gradient optimizer, as well as vary input image size. Presented network is trained with only 128 × 128px images, but it's able to process almost arbitrary sized images with good success. Presented network also doesn't incorporate any regularization, like dropout layers, or decay of weights and learning rate. Networks were also trained for only 100 epochs and even though validation metrics started to slow down during the end of training, they were still improving slowly. Now that presented network has shown to be a good baseline, next logical step would be to fine-tune the hyperparameters.

As a side note about window sizes, while networks were trained with 128 × 128px windows, for some reason using larger window sizes may actually lead to better results. I quickly tested classifications with 64 × 64px, 256 × 256px, 512 × 512px, 1024 × 1024px, 2048 × 2048px windows and full image at once. Overall metrics for target level classification are presented in table 19, and as seen there, overall metrics first improved a little, but stayed the same for 512 × 512px and larger. Even if 128 × 128px window had the second-worst performance, I still decided to use same window size for final classification than used for training, because the results would then be easier to compare to each other. One possible way to utilize this networks' ability to process (almost) arbitrary sized images could be to train the network different sized images. For instance, use 128 × 128px images as input for the first 100 epochs. After this, change the input data to be size 256 × 256px and continue training for sufficient amount of epochs.

Size	OA	Pre	Rec	F1	mIoU
64	0.76	0.64	0.55	0.58	0.45
128	0.76	0.64	0.57	0.59	0.46
256	0.76	0.64	0.58	0.60	0.47
512	0.76	0.64	0.59	0.61	0.47
1024	0.76	0.64	0.59	0.61	0.47
2048	0.77	0.64	0.59	0.61	0.47
Full image	0.77	0.64	0.59	0.61	0.47

Table 19. Full classification results for target level classification using different window sizes. Precision, recall and F1-score are macro averages

Segmentation maps presented in this thesis were produced by making predictions of smaller images and then assembling them into a grid. However, as seen in figure 38, especially in CLC level 2 classes, window borders can be clearly seen. One way get rid of this would be to extract the images with smaller strides, such as half of the image width and average the predictions for each pixel. Rakhlin et al. (2018) use this kind of approach, and they also have several other steps that could improve these results, such as equibatch sampling (making sure that each class is present once every *C* batches) and utilizing previously mentioned Lovasz-softmax loss function. Their classification task is a bit harder than CLC level 1, but way easier than target level classification task, and they managed to get pretty much as good results with only RGB-images as I did with S2-bands and indices.

Despite all hype about deep learning and neural networks, it may very well be that classical machine learning methods perform better for this kind of tasks. De Alban et al. (2018) utilize random forests in a large land cover change analysis combining multispectral satellite data and L-band SAR data with good success. They show that combining these features improve

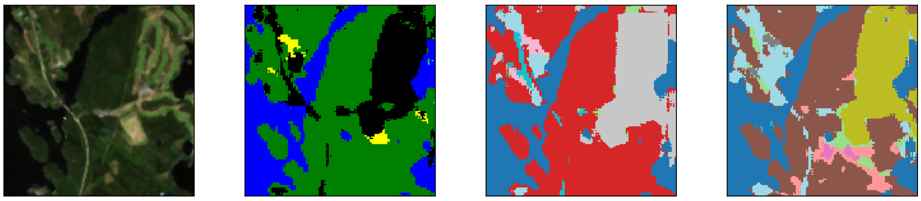


Figure 38. Without overlap sliding window borders can be detected on the left side of the scene.

classification compared to only Landsat satellite data. However, they also mention that recent improvements in CNN development should be tested with similar features. Neural network structure presented in this thesis combined with either L-band SAR or LiDAR data would, according to results presented in De Alban et al. (2018), improve classification even further.

Last point I'd like to highlight is the temporal dimension for classification. This thesis originally tried to use 3D CNNs and image stacks containing images from several different time steps, but that strategy yielded poor results outside of testing set. While there have been 3D implementations of U-Net, they are meant for volumetric segmentation of 3D images and thus are not suited for land cover classification. However, there are a couple of ways to utilize scenes from different times for U-Net type of classifier. The first and most simple one is to stack images from different time steps and classify those stacks. Second option is to train separate classifiers for each time step, be it month or season, and average the predictions to acquire segmentation maps. Third and in my opinion the most promising one is to combine recurrent neural networks (RNNs) with U-Net architecture, like R2U-Net proposed by Alom et al. (2018). RNNs are special kind of neural networks that are meant for sequential data processing (Goodfellow, Bengio, and Courville 2016). Like original U-Net, R2U-Net is also developed for medical image segmentation, but it doesn't mean that it can't be used in other fields. After all, U-Net has been adopted almost as a de-facto standard for segmentation networks. By adding the temporal dimension to images, it can be expected that for instance distinguishing between coniferous and deciduous trees will improve.

6 Conclusions

In this chapter, overall thesis process is reflected and successful application steps from section 1.1 are rechecked and evaluated. Even though I decided to completely change the used network structure and the type of classification performed in the midway of the thesis process, overall process was rather straightforward. Typical time allocation in data mining tasks is often described such as 80% of the time taken is spent on interpreting and processing data and all other steps take the remaining 20%, and this work was not an exception to this. Most of the time went to choosing viable images, preprocessing them and generating synthetic images.

In section 1.1 I stated that there is a clear definition of a problem, or rather several of those. I set the goal to test the viability of U-Net type classifier for LULC tasks using multispectral input data, and as a side product get segmentation maps for Rakkolanjoki drainage basin. As a conclusion for this question I can say, that U-Net is an excellent method for LULC tasks, and after testing and incorporating the improvements from section 5.3 it could improve even further. Questions two and three are nowadays almost trivial, because large amounts of exellent quality satellite images are freely available. However, for some tasks satellite images are not enough, but either SAR or LiDAR data is needed. For large scale LULC tasks Sentinel-2 or other satellite images are, in my opinion, sufficient enough.

I was lucky to have large amounts of training data for this task. In addition, this data was not from some well known LULC dataset that has been analyzed and classified thorouhgly. Had this testing been done in for example Pavia University dataset, same accuracies than I achieved with Kaakonkulma could have been considered a failure. Using non-estabilished dataset has some difficulties, because the accuracy of the training data is not as good as in some estabilished sets. This challenge was solved with comparing the produced segmentations to satellite images in addition to ”ground truth” segmentations and confusion matrices. These results still need more analyzing, because at least for me it’s practically impossible to distinguish between different forest types from satellite images, especially if they are only few pixels in size.

If I had to highlight two of the most important factors to focus in order to improve these results, they would be to test different loss functions more viable for unbalanced data and test some pretrained model for encoding path. After the model achieved the baseline for results (85% overall accuracy, all classes at least 70% accurate), then it would be reasonable to implement RNNs to it and see if temporal dimension improves these results even further.

Deep learning has many applications in different remote sensing tasks, and the development in last ten years has been fast. Lillesand, Kiefer, and Chipman (2008) mention artificial neural networks with only a few paragraphs, considering them to be a subject of continuing research in image interpretation. Ten years later neural networks, especially CNNs, are the default method for image interpretation, and new ways to utilize them in different areas are constantly found. In the near future, I'll possibly be researching new applications for neural networks in remote sensing. Around a week after I held a short presentation for SYKE Machine Learning network about CNNs and their usage in land cover classification, one of the senior researchers contacted me and asked if I was interested in participating in an Academy of Finland funding application as a Ph.D. student. If the application is approved, then I'll continue to work with this subject and implement the suggestions presented in section 5.3, and test method presented here for hyperspectral and LiDAR data.

Bibliography

Abadi, Martín, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. "Tensorflow: a system for large-scale machine learning." In *OSDI,* 16:265–283.

Alom, Md. Zahangir, Mahmudul Hasan, Chris Yakopcic, Tarek M. Taha, and Vijayan K. Asari. 2018. "Recurrent Residual Convolutional Neural Network based on U-Net (R2U-Net) for Medical Image Segmentation". *CoRR* abs/1802.06955. http://arxiv.org/abs/1802.06955.

Badrinarayanan, Vijay, Alex Kendall, and Roberto Cipolla. 2015. "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation". *CoRR* abs/1511.00561. arXiv: 1511.00561. http://arxiv.org/abs/1511.00561.

Bhandare, Ashwin, Maithili Bhide, Pranav Gokhale, and Rohan Chandavarkar. 2016. "Applications of Convolutional Neural Networks". *International Journal of Computer Science and Information Technologies:* 2206–2215.

Chollet, François, et al. 2015. *Keras.* https://keras.io.

Commons, Wikimedia. 2018. *File:Electromagnetic-Spectrum.svg — Wikimedia Commons, the free media repository".* Online; accessed 15-November-2018. https://commons.wikimedia.org/w/index.php?title=File:Electromagnetic-Spectrum.svg&oldid=293570842.

Csurka, Gabriela, Diane Larlus, Florent Perronnin, and France Meylan. 2013. "What is a good evaluation measure for semantic segmentation?." In *BMVC,* 27:2013. Citeseer.

De Alban, Jose Don T., Grant M. Connette, Patrick Oswald, and Edward L. Webb. 2018. "Combined Landsat and L-Band SAR Data Improves Land Cover Classification and Change Detection in Dynamic Tropical Landscapes". *Remote Sensing* 10 (2). ISSN: 2072-4292. doi:10.3390/rs10020306. http://www.mdpi.com/2072-4292/10/2/306.

European Environment Agency. 1985. "CORINE land cover". Visited on June 29, 2018. https://www.eea.europa.eu/publications/COR0-landcover.

European Space Agency. 2018a. *2nd Sentinel 2 validation team meeting.* https://elib.dlr.de/119319/1/LouisJ_S2VT02%20-%2020180125.pdf.

———. 2018b. *Sen2Three.* http://step.esa.int/main/third-party-plugins-2/sen2three/.

———. 2018c. “Sentinel-2 – Missions”. https://sentinel.esa.int/web/sentinel/missions/sentinel-2.

Fauvel, Mathieu, Yuliya Tarabalka, Jon Atli Benediktsson, Jocelyn Chanussot, and James C Tilton. 2013. “Advances in spectral-spatial classification of hyperspectral images”. *Proceedings of the IEEE* 101 (3): 652–675.

Foody, Giles M. 2002. “Status of land cover classification accuracy assessment”. *Remote Sensing of Environment* 80 (1): 185–201. ISSN: 0034-4257. doi:https://doi.org/10.1016/S0034-4257(01)00295-4. http://www.sciencedirect.com/science/article/pii/S0034425701002954.

Garini, Yuval, Ian T Young, and George McNamara. 2006. “Spectral imaging: principles and applications”. *Cytometry Part A: The Journal of the International Society for Analytical Cytology* 69 (8): 735–747.

Goetz, Alexander FH. 2009. “Three decades of hyperspectral remote sensing of the Earth: A personal view”. *Remote Sensing of Environment* 113:S5–S16.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning.* http://www.deeplearningbook.org. MIT Press.

Härmä, Pekka, Suvi Hatunen, Markus Törmä, Elise Järvenpää, Minna Kallio, Riitta Teiniranta, Tiia Kiiski, and Jaakko Suikkanen. 2014. *GIO Land Monitoring 2011–2013 in the framework of regulation (EU) No 911/2010 Final Report Finland.* Finnish Environmental Institution.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification”. In *Proceedings of the IEEE international conference on computer vision,* 1026–1034.

Hu, Jie, Li Shen, and Gang Sun. 2017. “Squeeze-and-excitation networks”. *arXiv preprint arXiv:1709.01507* 7.

Hunter, J. D. 2007. “Matplotlib: A 2D graphics environment”. *Computing In Science & Engineering* 9 (3): 90–95.

Ji, Shunping, Chi Zhang, Anjian Xu, Yun Shi, and Yulin Duan. 2018. “3D Convolutional Neural Networks for Crop Classification with Multi-Temporal Remote Sensing Images”. *Remote Sensing* 10 (1): 75. ISSN: 2072-4292. doi:10.3390/rs10010075. http://www.mdpi.com/2072-4292/10/1/75.

Jones, Hamlyn G., and Robin A. Vaughan. 2010. *Remote sensing of vegetation.* Oxford university press.

Kemker, Ronald, Carl Salvaggio, and Christopher Kanan. 2018. “Algorithms for semantic segmentation of multispectral remote sensing imagery using deep learning”. *ISPRS Journal of Photogrammetry and Remote Sensing.* ISSN: 0924-2716. doi:https://doi.org/10.1016/j.isprsjprs.2018.04.014. http://www.sciencedirect.com/science/article/pii/S0924271618301229.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton. 2012. “Imagenet classification with deep convolutional neural networks”. In *Advances in neural information processing systems,* 1097–1105.

LeCun, Yann. 1989. “Generalization and network design strategies”. *Connectionism in perspective:* 143–155.

LeCun, Yann, Koray Kavukcuoglu, Clément Farabet, et al. 2010. “Convolutional networks and applications in vision.” In *ISCAS,* 2010:253–256.

Li, Fei–Fei, Justin Johnson, and Serena Yeung. 2018. *CS231n Lecture 4: Backpropagation and Neural Networks.* Stanford University school of engineering. http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture04.pdf.

Li, Ying, Haokui Zhang, and Qiang Shen. 2017. “Spectral–Spatial Classification of Hyperspectral Imagery with 3D Convolutional Neural Network”. *Remote Sensing* 9 (1): 67. ISSN: 2072-4292. doi:10.3390/rs9010067. http://www.mdpi.com/2072-4292/9/1/67.

Lillesand, Thomas M., Ralph W. Kiefer, and Jonathan W. Chipman. 2008. *Remote sensing and image interpretation.* John Wiley & Sons.

Lin, Tsung-Yi, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. 2017. “Focal Loss for Dense Object Detection”. *CoRR* abs/1708.02002. arXiv: 1708.02002. http://arxiv.org/abs/1708.02002.

Long, Jonathan, Evan Shelhamer, and Trevor Darrell. 2015. “Fully convolutional networks for semantic segmentation”. In *Proceedings of the IEEE conference on computer vision and pattern recognition,* 3431–3440.

Lua, Guolan, and Baowei Feia. 2014. “Medical hyperspectral imaging : a review”.

Matthew, Maxim Berman Amal Rannen Triki, and B Blaschko. 2018. “The Lovász-Softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks”.

Nogueira, Keiller, Otávio AB Penatti, and Jefersson A dos Santos. 2017. “Towards better exploiting convolutional neural networks for remote sensing scene classification”. *Pattern Recognition* 61:539–556.

Rakhlin, Alexander, OU Neuromation, Alex Davydow, and Sergey Nikolenko. 2018. “Land Cover Classification from Satellite Imagery With U-Net and Lovász-Softmax Loss”. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops,* 262–266.

Rolnick, David, Andreas Veit, Serge Belongie, and Nir Shavit. 2017. “Deep learning is robust to massive label noise”. *arXiv preprint arXiv:1705.10694.*

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. 2015. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI),* 9351:234–241. LNCS. (available on arXiv:1505.04597 [cs.CV]). Springer. http://lmb.informatik.uni-freiburg.de/Publications/2015/RFB15a.

Rumelhart, David E, Geoffrey E Hinton, and Ronald J Williams. 1986. “Learning representations by back-propagating errors”. *nature* 323 (6088): 533.

Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. 2015. “Imagenet large scale visual recognition challenge”. *International Journal of Computer Vision* 115 (3): 211–252.

Siam, Mennatullah, Mostafa Gamal, Moemen Abdel-Razek, Senthil Yogamani, and Martin Jagersand. 2018. “RTSeg: Real-time Semantic Segmentation Comparative Study”. *arXiv preprint arXiv:1803.02758.*

Sokolova, Marina, and Guy Lapalme. 2009. “A systematic analysis of performance measures for classification tasks”. *Information Processing & Management* 45 (4): 427–437. ISSN: 0306-4573. doi:https://doi.org/10.1016/j.ipm.2009.03.002. http://www.sciencedirect.com/science/article/pii/S0306457309000259.

Springenberg, J, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. 2015. “Striving for Simplicity: The All Convolutional Net”. In *ICLR (workshop track).*

Waskom, Michael, Olga Botvinnik, Drew O’Kane, Paul Hobson, Saulius Lukauskas, David C Gemperline, Tom Augspurger, et al. 2017. *mwaskom/seaborn: v0.8.1 (September 2017).* doi:10.5281/zenodo.883859. https://doi.org/10.5281/zenodo.883859.

Zhang, Zhengxin, Qingjie Liu, and Yunhong Wang. 2018. “Road extraction by deep residual u-net”. *IEEE Geoscience and Remote Sensing Letters.*

Appendices

A Aggregation of classification labels

If a cell is empty, then the label is same than the previous one.

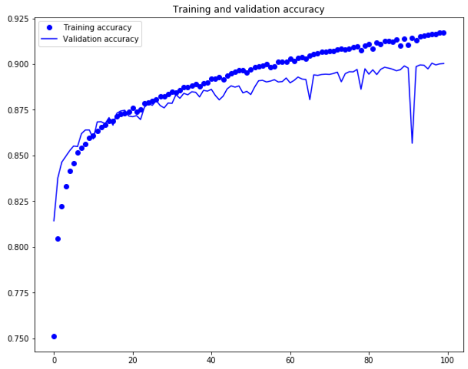
CLC2012 level 4	**CLC2012 Level 2**	**CLC2012 Level 1**	**Target**
Continuous urban fabric	Urban fabric	Artificial Surfaces	Built-up areas
Discontinuous urban fabric			
Commercial units	Industrial, commercial, and transport units		
Industrial units			
Road and rail networks and associated land			
Port areas			
Airports			
Mineral extraction sites	Mine, dump and construction sites		Bare areas
Open cast mines			
Dump sites			
Construction sites			
Summer cottages	Artificial non-agricultural vegetated areas		Grasslands
Sport and leisure areas			
Golf courses			

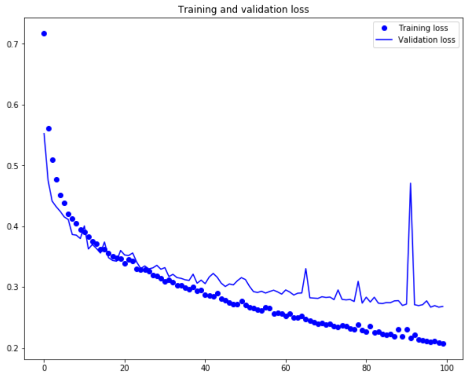
Trotting tracks			
Non-irrigated arable land	Arable land	Agricultural areas	Fields
Fruit trees and berry plantations	Permanent crops		Grasslands
Pastures	Pastures		
Natural pastures			
Land principally occupied by agriculture	Heterogenous agricultural areas		
Agro-forestry areas			
Broad-leaved forest on mineral soil	Forests	Forests and semi-natural areas	Broad-leaved forest
Broad-leaved forest on peatland			
Coniferous forest on mineral soil			Coniferous forest
Coniferous forest on peatland			
Mixed forest on mineral soil			Mixed forest
Mixed forest on peatland			
Mixed forest on rocky soil			
Natural grassland	Shrub and/or herbaceous vegetation associations		Transitional woodland shrub

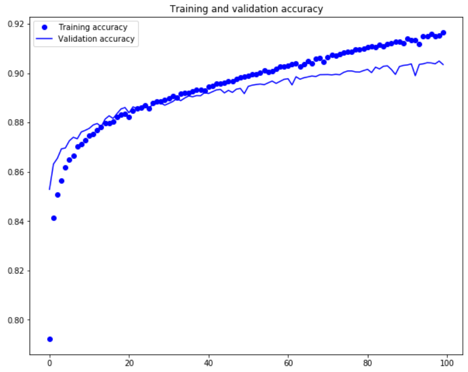
Moors and heathland			
Transitional woodland shrub (all cc)			
Beaches, dunes and sand plains	Open spaces with little or no vegetation		Bare areas
Bare rock			
Sparsely vegetated areas			
Inland marshes, terrestrial	Inland wetlands	Wetlands	Wetlands
Inland marshes, aquatic			Water vegetation
Peatbogs			Wetlands
Peat production sites			Bare areas
Water courses	Inland waters	Water bodies	Water bodies
Water bodies			

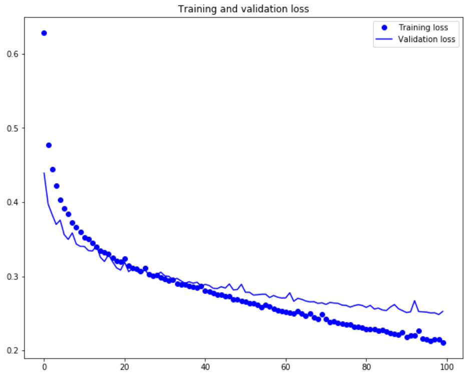
B Training and validation metrics for CLC level 1 classes

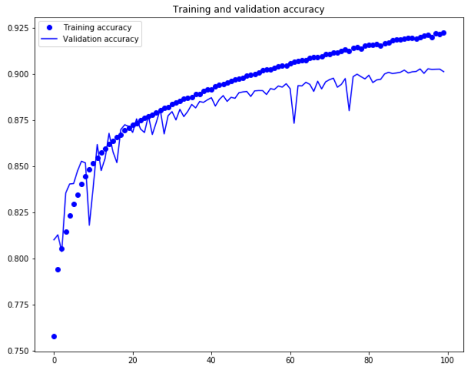
This appendix contains the training and validation metrics for CLC level 1 neural network training. Graphs are presented in the following order: RGB, S2-bands, spectral indices, combined.

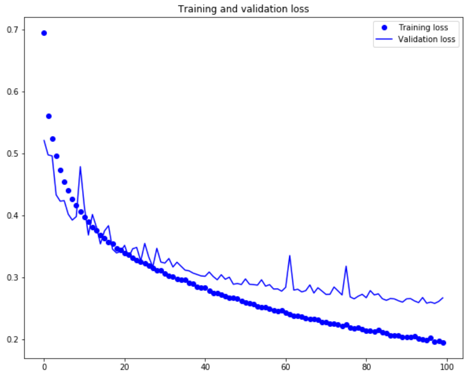

Training and validation accuracy
Training accuracy
Validation accuracy
0.925
0.900
0.875
0.850
0.825
0.800
0.775
0.750
0
20
40
60
80
100

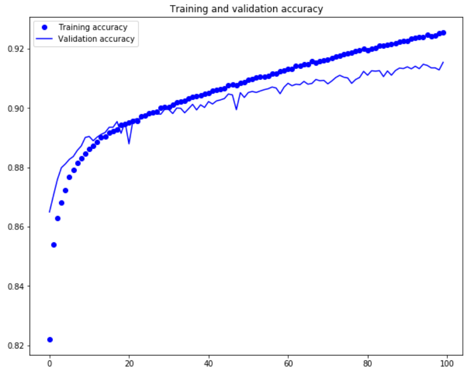

Training and validation loss
Training loss
Validation loss
0.7
0.6
0.5
0.4
0.3
0.2
0
20
40
60
80
100

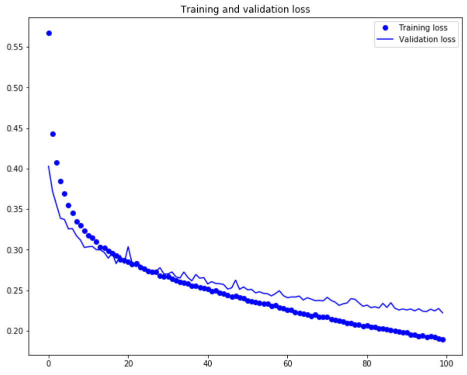

Training and validation accuracy
Training accuracy
Validation accuracy
0.92
0.90
0.88
0.86
0.84
0.82
0.80
0
20
40
60
80
100


Training and validation loss
Training loss
Validation loss
0.6
0.5
0.4
0.3
0.2
0
20
40
60
80
100


Training and validation accuracy
Training accuracy
Validation accuracy
0.925
0.900
0.875
0.850
0.825
0.800
0.775
0.750
0
20
40
60
80
100

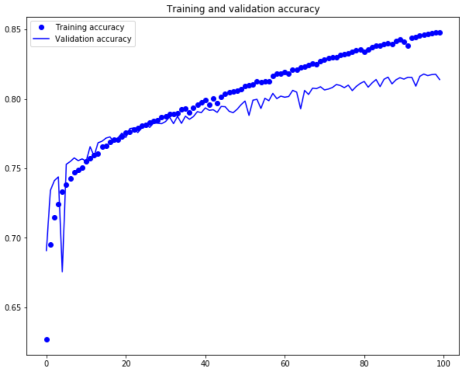

Training and validation loss
Training loss
Validation loss
0.7
0.6
0.5
0.4
0.3
0.2
0
20
40
60
80
100

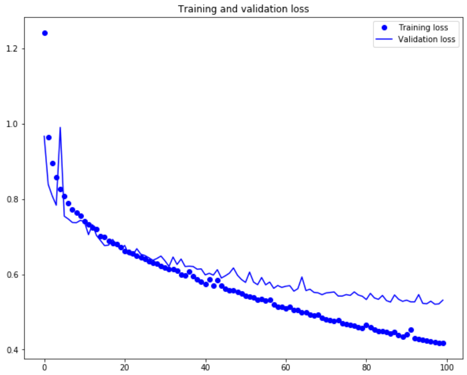

Training and validation accuracy
Training accuracy
Validation accuracy
0.92
0.90
0.88
0.86
0.84
0.82
0
20
40
60
80
100

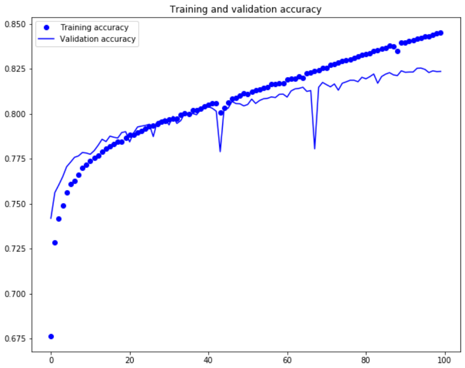

Training and validation loss
Training loss
Validation loss
0.55
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0
20
40
60
80
100

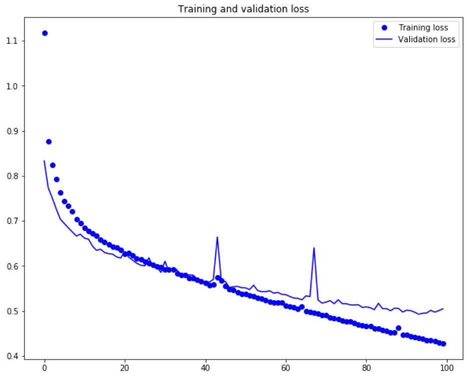
C Training and validation metrics for CLC level 2 classes

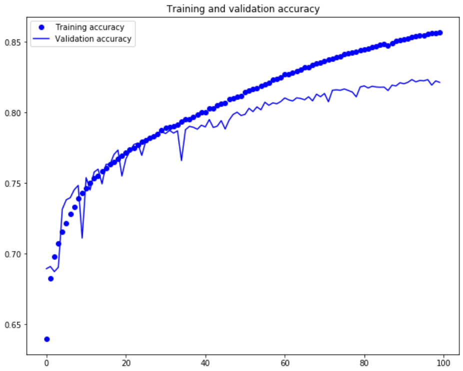
This appendix contains the training and validation metrics for CLC level 2 neural network training. Graphs are presented in the following order: RGB, S2-bands, spectral indices, combined.

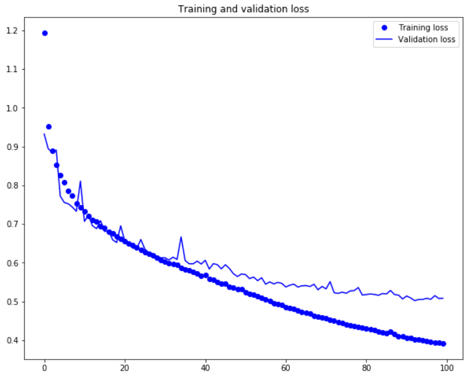


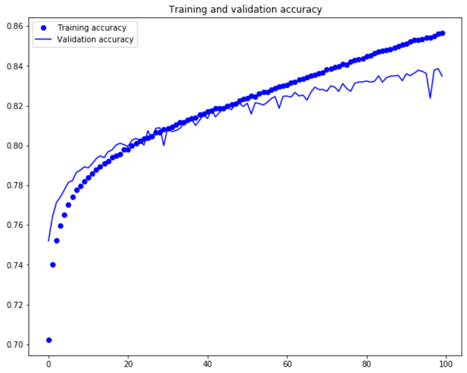


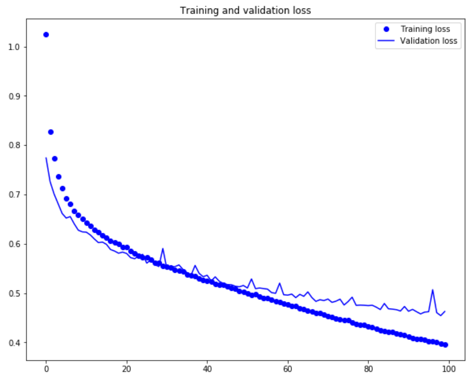






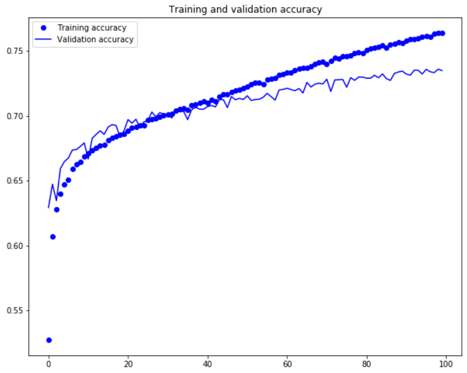


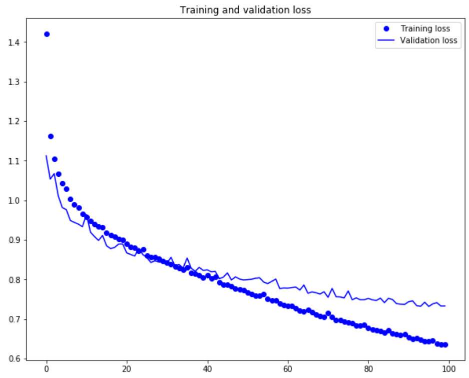


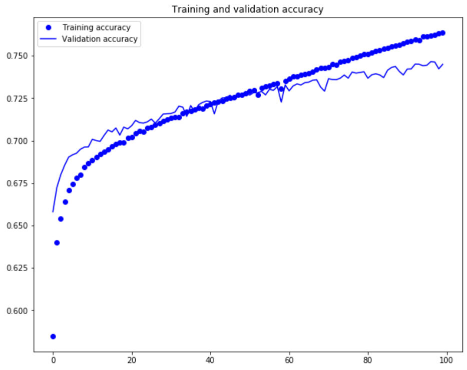


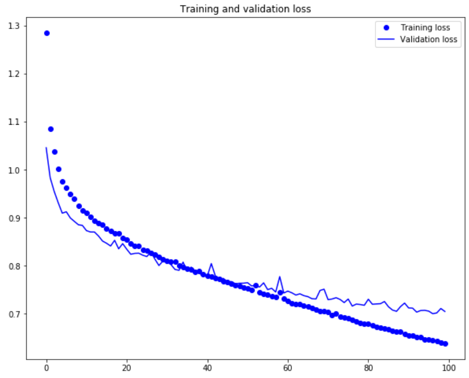
D Training and validation metrics for target level classes

This appendix contains the training and validation metrics for target level neural network training. Graphs are presented in the following order: RGB, S2-bands, spectral indices, combined.

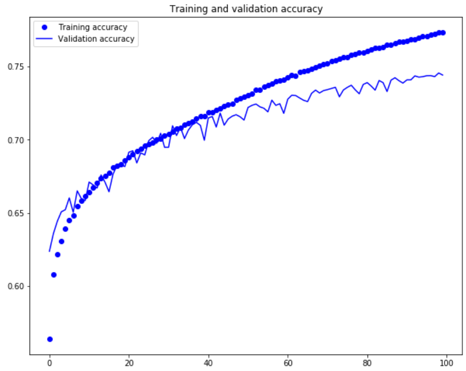




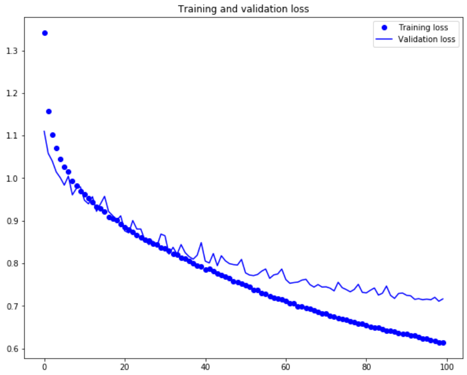




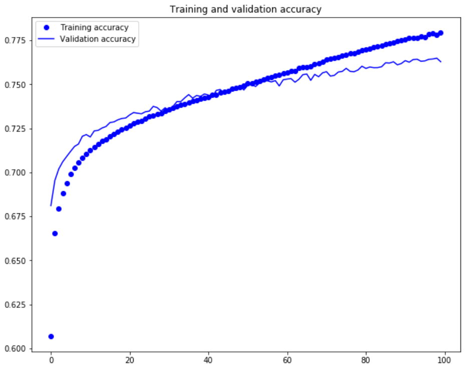
Training and validation accuracy



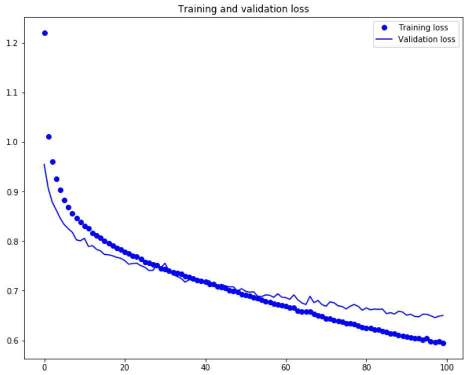
Training and validation loss



Training and validation accuracy



Training and validation loss


Training loss
Validation loss