UNIVERSITY OF OULU

FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

Nadir Bengana

Land Cover and Forest Segmentation using Deep Neural Networks

Master's Thesis
Degree Programme in Computer Science and Engineering
May 2019

Bengana N. (2019) Land Cover and Forest Segmentation using Deep Neural Networks. University of Oulu, Degree Programme in Computer Science and Engineering. Master's thesis, 53 p.

ABSTRACT

Land Use and Land Cover (LULC) information is important for a variety of applications notably ones related to forestry. The segmentation of remotely sensed images has attracted various research subjects. However this is no easy task, with various challenges to face including the complexity of satellite images, the difficulty to get hold of them, and lack of ready datasets. It has become clear that trying to classify on multiple classes requires more elaborate methods such as Deep Learning (DL). Deep Neural Networks (DNNs) have a promising potential to be a good candidate for the task. However DNNs require a huge amount of data to train including the Ground Truth (GT) data. In this thesis a DL pixel-based approach backed by the state of the art semantic segmentation methods is followed to tackle the problem of LULC mapping. The DNN used is based on DeepLabv3 network with an encoder-decoder architecture. To tackle the issue of lack of data the Sentinel-2 satellite whose data is provided for free by Copernicus was used with the GT mapping from Corine Land Cover (CLC) provided by Copernicus and modified by Tyke to a higher resolution. From the multispectral images in Sentinel-2 Red Green Blue (RGB), and Near Infra Red (NIR) channels were extracted, the 4th channel being extremely useful in the detection of vegetation. This ended up achieving quite good accuracy on a DNN based on ResNet-50 which was calculated using the Mean Intersection over Union (MIoU) metric reaching $0{,}53 MIoU$. It was possible to use this data to transfer the learning to a data from Pleiades-1 satellite with much better resolution, Very High Resolution (VHR) in fact. The results were excellent especially when compared on training right away on that data reaching an accuracy of $0{,}98$ and $0{,}85 MIoU$.

Keywords: Remote sensing, land cover, forestry, semantic segmentation, deep learning

TABLE OF CONTENTS

FOREWORD

This master thesis was written at the Center of Machine Vision and Signal Analysis (CMVS) as part of the Forestry One Stop Shop (FOSS) project funded by Business Finland (BF). I would like to thank my supervisor, Prof. Janne Heikkilä for his guidance throughout the process of research and writing the thesis. I would also like to thank the team from VTT Technical Research Centre of Finland at Espoo notably Anne Lönnqvist, Heikki Astola, and Lauri Seitsonen. Thanks to all the professors who helped me during my master studies.

I am also grateful to my family and friends and everyone who helped me during my studies.

Oulu, Finland May 4, 2019

Nadir Bengana

ABBREVIATIONS

AI	Artificial Intelligence
ANN	Artificial Neural Networks
AP	Average Precision
ASPP	Atrous Spacial Pyramid Pooling
ASPP	Atrous Spatial Pyramid Pooling
BF	Boundary F1
BIL	Band Interleaved by Line
BIP	Band Interleaved Pixel
BN	Batch Normalization
BSQ	Band SeQuential
BoVW	Bag Of Visual Words
CLC	Corine Land Cover
CNN	Convolutional Neural Networks
DL	Deep Learning
DNN	Deep Neural Networks
DT	Decision Trees
EO	Earth Observation
ESA	European Space Agency
FC	Fully Connected
FCN	Fully Convolutional Networks
FOSS	Forestry One Stop Shop
GDAL	the Geospatial Data Abstraction Library
GIS	Geographic Information Systems
GPUs	Graphical Processing Units
GT	Ground Truth
ILSVRC	The ImageNet Large Scale Visual Recognition Challenge
JAXA	The Japan Aerospace Exploration Agency
k-NN	The k Nearest Neighbour
LULC	Land Use and Land Cover Classification
LiDAR	Light Detection And Ranging
MIoU	Mean Intersection over Union
ML	Machine Learning
MLP	Multi Layer Perceptron
NASA	National Aeronautics and Space Administration
NDVI	Normalized Difference Vegetation Index
PA	Pixel Accuracy
PCA	Principle Component Analysis
PNG	Portable Network Graphics
RADAR	RAdio Detection And Ranging
RED-Net	Residual Encoder-Decoder Network
RGB	Red Green Blue
RS	Remote Sensing
ReLU	Rectified Linear Units
ResNet	Residual Nets
SVM	Support Vector Machines

TDRSS	Tracking and Data Relay Satellite System
VAE	Variational Auto-Encoders
VHR	Very High Resolution

1. INTRODUCTION

The mapping of land cover from satellite images is an ongoing research topic, especially when the type of cover in question is forest species. The images used for that purpose are taken using high resolution cameras on board Earth Observation (EO) satellites and other sensors such as radars. The process mentioned is dubbed remote sensing. The usage of satellite images is a good alternative to airplane photography which might require permits to flyby, or field work that would be costly. Automating this process is extremely useful to reduce the amount of work and the cost. What was nearly impossible at times due to the inaccessibility of the terrain or the border policies is possible thanks to the fact that satellites can image any place of the planet.

Achieving good results in land cover segmentation would be beneficial in various domains such as landscape changes monitoring, forest fires tracking, road extraction, military surveillance, etc. This thesis will focus on the forest types segmentation for the Forestry One Stop Shop (FOSS) project funded by Business Finland. It answers questions such as: where and how much there is forest? And which types of trees are in each forest?

The tools used to segment satellite images by land cover type are Deep Learning (DL) methods. DL is based on Deep Neural Networks (DNN) which have shown promising results when it comes image recognition, classification and segmentation [1] [2]. The method of segmentation that is going to be utilized is semantic segmentation consisting of labeling of each pixel in an image with a predetermined class label. DNNs require a large number of images for the training phase. This need is fulfilled with data from satellites providing high resolution and Very High Resolution (VHR) images. Satellites such as the Sentinel2 have their data freely accessible by anyone which simplifies the process.

In this thesis DL methods will be explored to apply semantic segmentation on satellite images for the purpose of land and forest cover segmentation. Chapter 2 will explore the background knowledge in the domain of Remote Sensing (RS), semantic segmentation, land cover, and forest cover mapping. Chapter 3 will tackle the data used in the thesis and study area where it is extracted from. Chapter 4 focuses on the methods and implementation of the solution for the problem in question. Chapter 5 reports on the results of the implementation and discuss them, before concluding in the final chapter.

2. REMOTE SENSING

Remote Sensing (RS) is the act of acquiring info about a phenomena or object without the need to physically come in contact with it [3].

Remote Sensing (RS) relies on a sensor, capable of measuring the energy from the electromagnetic spectrum reflected by an object without coming in contact with it. In the event that the energy reflected by the object studied has been emitted by the device measuring it the type of RS is called ative, as opposed to passive RS where the source of the energy is another entity such as sunlight or heat.

2.1. Sensors in Remote Sensing

Although RS can be applied to many fields of studies, the main field referred to when RS is mentioned is the use of sensors on board aircrafts or satellites to study objects on the surface of the Earth whether it is on the ground, the ocean, or the atmosphere. This process is called airborne remote sensing, also referred to as spaceborne in the case of satellites. The sensors used for the purpose mentioned range from passive sensors such as multispectal or hyperspectral cameras to active ones such as RAdio Detection And Ranging (RADAR) and Light Detection And Ranging (LiDAR).

2.1.1. Passive Sensors

Passive sensor is a device that detects and reacts to an input from the physical environment. In RS passive sensors detect the reflection of electromagnetic waves reflected by a distant object. The source of the radiation reflected is independent from the sensor. It could be a third party such as sunlight or the object itself such as its internal heat.

Multispectral and hyperspectral cameras are the bread-and-butter of passive remote sensing. Those sensors consist of dividing the electromagnetic spectrum to several bands, a dozen for multispectral and hundreds or even thousands for hyperspectral (Figure 1). The ability to distinguish between two close points in the electromagnetic spectrum is called Spectral resolution denoted as $\Delta\lambda$. Depending on the application these multispectral bands are given different priorities, usually Red Green Blue (RGB) aka the visible spectrum channels are given high priority by having a higher spatial resolution. This refers to the amount of detail that can be seen in each pixel. For example a $10m/px$ resolution means that the smallest distinguishable object in the image is $10m$. Other Passive sensors include sounders. They are microwave radiometer capable of measuring vertical distributions of atmospheric parameters such as temperature and pressure. These sensors are especially useful for weather and climate monitoring.

2.1.2. Active Sensors

In RS unlike passive sensors active sensors do not just receive signals in the form of electromagnetic waves. They instead are the source of that signal. The part of the

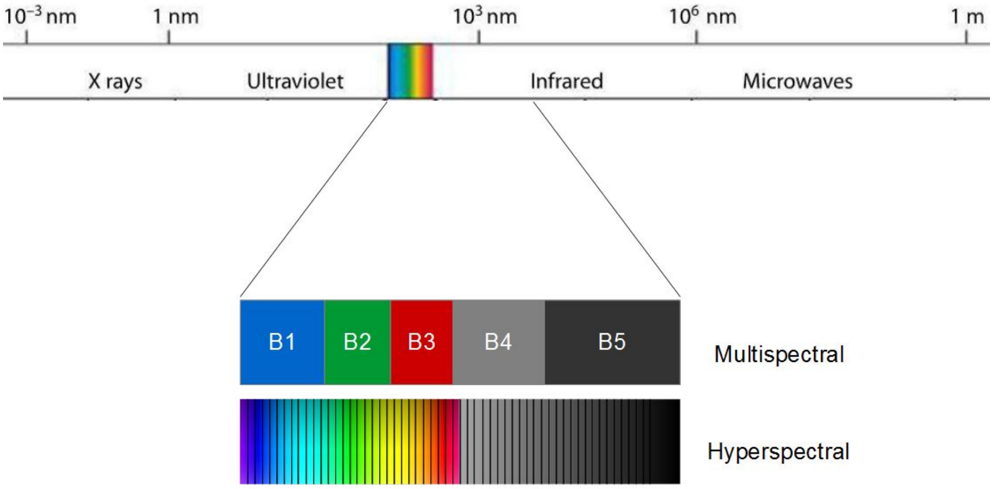


Figure 1. Bands in multispectral imagery vs bands in hyperspectral imagery.

electromagnetic spectrum these sensors operate in is the microwave portion for the most part, giving them the ability to penetrate objects such as clouds or water.

RADARs are the most well-known active sensors. These sensors contain a transmitter that emit microwave radiation in the from of pulses or continuous signal. The receiver usually the same as the transmitter then receives the reflection of that radiation. Radars can penetrate through clouds, fog, snow, rain all of which can block visible light. The distance to the target the radar is observing can be determined by calculating the time it takes for the emitted waves to reach the sensor again. This is particularly useful in having a 3d image by obtaining the depth measurement as well.

LiDARs sensors use lasers to emit light pulses and record the time it takes for them to reflect back. LiDARs cannot penetrate mediums like radars but are very precise in measuring the difference in elevation giving it better 3D imagery of the surface being imaged.

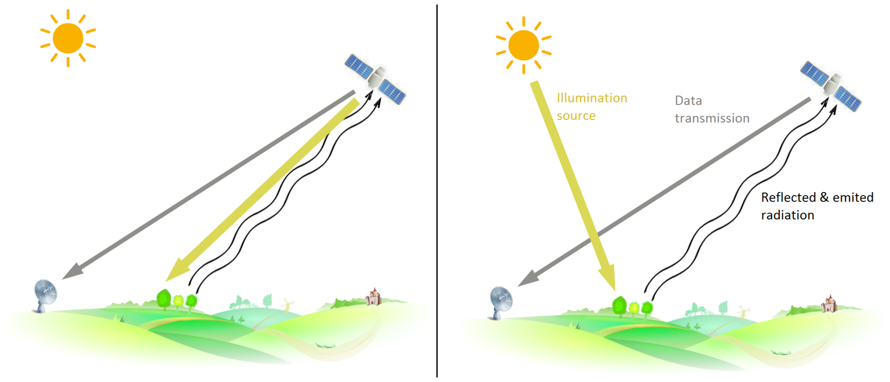


Figure 2. Left: Active Remote sensing where the sensor provides its own illumination; Right: Passive remote sensing where the sensor relies on another source for illumination like in this example the Sun.

2.2. Satellites and Remote Sensing

A satellite is an object orbiting another object. Satellites can be natural like the Moon orbiting Earth or artificial which are the man made objects put in orbit. In the context of remote sensing satellites refer to the latter type.

Remote Sensing has become almost synonymous with EO which is the gathering of information about Earth's physical, chemical and biological systems. The means by which EO is done varies but it is mostly done by satellites called Earth Observation Satellites. Satellites such as the TIROS and Landsat1 were amongst the first civilian EO satellites[4]. By 2018 over 700 EO satellites are in orbit [5].

EO Satellites with the goal of capturing images of Earth's surface are positioned in Sun-synchronous orbits. They are near polar orbits, positioned between $600km$ to $800km$ above sea level where the satellite revisits any point of the planet at the same local solar time. This is helpful in avoiding variation of illumination caused by sun. The satellites aiming at monitoring the weather are put in a different type of orbit usually in geo-stationary orbits. Located at an altitude of $36000km$ geo-stationary orbits orbit at the same rate Earth rotates making them seem fixed in the sky.

RS satellites are equipped with a variety of both passive and active sensors. Early satellites had a low spatial resolution due to the limitation of technology but also partly due to limitation by military agencies. The Landsat1 launched in 1972 had a $60m/px$ resolution while recent satellites such as WorldView-4 boasting a $0{,}31m/px$ resolution.

On top of the properties of RS cameras mentioned previously the Swath is a property related to the satellite itself. It represents the area imaged on the surface of the planet by said satellite. In sun-synchronous orbits the swath shifts westward covering new areas after each rotation. The nadir point is another property (Figure 3). It represents the point directly below the satellite. An orbit cycle is when the satellite visits the same nadir point again. The time it takes for a satellite to finish an orbit cycle is called the revisit period. This is important because it defines the temporal resolution of the images taken by the satellite[1].

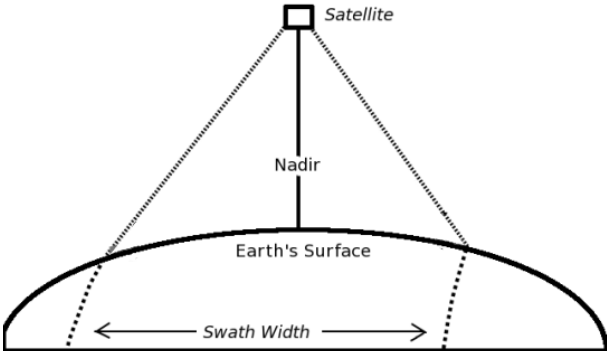


Figure 3. Swath and Nadir of a satellite.

Satellites give a massive advantage in areal remote sensing over other conventional methods like airplanes. That is because albeit being more expensive to launch, they are

[1] The actual temporal resolution isn't always the same as the revisit period. It depends on a variety of factors, including the satellite/sensor capabilities, the swath overlap, and latitude.

cheaper to use. They are capable of imaging large chunks of the planet in much less time with less cost and are not limited by borders.

2.2.1. Sentinel-2 Satellites

The Copernicus Sentinel-2 mission comprises a constellation of two polar-orbiting satellites placed in the same sun-synchronous orbit, phased at 180° to each other. It aims at monitoring variability in land surface conditions, and its wide swath width (290 km) and high revisit time (10 days at the equator with one satellite, and 5 days with 2 satellites under cloud-free conditions which results in 2-3 days at mid-latitudes) will support monitoring of Earth's surface changes [6]. Sentinel-2 is a very good candidate for remote sensing application notably land cover segmentation and by extent forest cover segmentation with their wide range of spectral bands, and with a 10m/px (Table 1). It has enough details to capture the differences of textures from various types of forests. Another major positive aspect of it is that data from the Sentinel-2 constellation is provided for free through the Copernicus Open Access Hub [7].

Table 1. Sentinel2-B properties

Bands	Bandwidth (nm)	Central wave-length (nm)	Spacial resolution (m)
B1: Coastal	21	442.3	60
B2: Blue	66	492.1	10
B3: Green	36	559	10
B4: Red	31	665	10
B5: Vegetation red edge	16	703.8	20
B6: Vegetation red edge	15	739.1	20
B7: Vegetation red edge	20	779.7	20
B8: NIR	106	833	10
B8A: Narrow NIR	22	864	20
B9: Water Vapour	21	943.2	60
B10: Short Wave Infra Red (SWIR)	30	1376.9	60
B11: SWIR	94	1610.4	20
B12: SWIR	185	2185.7	20

2.2.2. Pleiades-1 Satellites

The Pleiades constellation composes of two polar-orbiting satellites phased at 180° to each other. Pleiades satellites share the same orbital plane with SPOT 6 and 7 satellites providing a constellation of four satellites phased at 90° to each other [8].

Pleiades constellation is an excellent set of earth observation satellites providing a pixel resolution of as high as $50cm/px$. It also provides an NIR channel which is very useful for forestry applications (Table 2).

Table 2. Pleiades-1B properties

Bands	Bandwidth (nm)	Central wave-length (nm)	Spacial resolution (m)
B1: Panchromatic	350	655	0.5
B2: Blue	120	490	2
B3: Green	120	550	2
B4: Red	120	660	2
B5: NIR	200	850	2

2.3. Data Processing

Satellites orbit the planet at altitudes over $100km$, having speeds of about $28080km/h$ [9]. Conditions like that introduce several issues while imaging the surface such as occlusion or total coverage by clouds, distortions caused by the curvature of the planet, motion blur, and the atmosphere.

2.3.1. Data recovery

The first "EO" satellite was the Corona spy satellite(s) launched by the U.S. military in the late 50s and early 60s. Some issues in imaging surfaces of the planet from a height of over $100km$ above sea level appeared. One of those issues is the stabilization the satellites so that the images are not affected by motion blur. The solution was using stars as reference points. This method is called three-axis body stabilization which is still used to this day.

Early Satellites like the Corona sent the data by physically dropping the film to the surface, meaning there was no real-time transmission. Nowadays data transmission is done via the Tracking and Data Relay Satellite System (TDRSS). The TDRSS is composed of a network of geostationary satellites providing continuous coverage for low earth orbiters that include the remote sensing satellites[10]. TDRSS satellites being geostationary can have direct communication to ground receiving stations much like TV satellites.

2.3.2. Data format

Unlike old satellites that used analog cameras and stored the data in films, today's EO satellites use digital format of the data introducing another property for the RS satellites cameras which is radiometric resolution. Radiometric resolution defines the number of binary bits required for each pixel. Most cameras use the $8bit/pixel$ format but RS satellites requiring more details have more. The Sentinel2 satellite has a radiometric resolution of $12bit/pixel$ [11].

The data is stored in the form of rasters which are matrices of pixels arranged in three main formats, Band Interleaved Pixel (BIP), Band Interleaved by Line (BIL) or Band SeQuential (BSQ). BIP is saved in a manner where each pixel is stored by row then the individual pixels of each band respectively. BIL stores the pixel values by row then the whole band of that row. BSQ format stores the pixels by going through the whole band sequentially row by row moving to the next band. The data is then encoded in a digital form such as LGSOWG (Landsat Ground Station Operators Working Group) or Super Structured Format, GeoTIFF (Geographic Tagged Image File Format) which is the most popular, or HDF (Hierarchical Data Format).

2.3.3. Preprocessing

Preprocessing is a crucial step in obtaining the RS satellite data. The images suffer from distortions caused by the atmosphere, planet's curvature, and the sensor. Thus distances need to be known to project the images into a plane. Geo-referencing, radiometric and atmospheric corrections are the usual steps performed in preprocessing satellite images.

Radiometric corrections

Using sun-synchronous orbits permits the satellites to take images where the illumination angle surface of the planet underneath it to be almost the same. The reason it is just almost and not always is due to other factors affecting the illumination on the surface.

Atmospheric corrections

Atmospheric correction is done by calibrating each band so that its minimum value is a 0 pixel value. The result of that is elimination of atmospheric haze.

Geo-referencing

Geo-referencing is done by knowing geographic coordinates of control points and then transform the image data according to those points [12]. This will assign real world coordinates to each pixel in the raster image.

2.3.4. Processing levels

In order to make it easy to recognize the type of processing done on a product NASA (National Aeronautics and Space Administration) defined processing levels for the RS imagery [13]. Now image distribution agencies have adopted a similar set of processing levels ranging from level 0 to level 4.

Level 0

Level 0 usually represents the raw data from the sensor without any processing or correction. Agencies rarely provide this form of data.

Level 1

Level 1 is divided into at least two steps of processing, level 1A and level 1B. Level 1A does the illumination calibration between the sensor values on board. Essentially 1A includes radiometric corrections and not geometric corrections [13] [14]. In the European Space Agency (ESA)'s Sentinel mission level 1A does not include radiometric correction, that is instead referred to as Level 1B, Level 1A only includes decompressing packets [15]. Level 1B includes more radiometric corrections such as brightness temperature. It also includes geometric correction. Geometric corrections deal with the distortions caused by the sensor or the movement of the satellite or the planet. Level 1B is referred to as Level 1C in the Sentinel mission.

Level 2

NASA defines it as derived geophysical variables at the same resolution and location as the Level 1 source data. The Japan Aerospace Exploration Agency (JAXA) shares somewhat the same definition [13] [14]. This is fancy wording for geo-referencing. ESA's level 2 processing includes scene classification and atmospheric correction [15].

Level 3

Level 3 includes geo-referrecing of the image data using ground control points as well as mapping on a uniform time scale. It also includes fixing of missing point in the image. This is the data that is mostly available for end users [16].

Level 4

Level 4 is the model output of the analysis of the previous data. Any sort of extra processing on top of the previous levels is defined as level 4 processing.

3. LAND COVER CLASSIFICATION AND SEGMENTATION

Land cover represents the biophysical and physical cover on the planet's surface. It ranges from bareback soil and water surfaces to vegetation and man-made structures [17]. Forest cover on the other hand is a sub-category of land cover. It represents the area covered by forests which are defined as large areas dominated by trees.

3.1. Land cover classes

Land cover classes can be distinguished by either being natural or man-made. The man-made cover classes are often referred to as land use, defining how humans utilize the land. Land use can refer to both Urban and Agricultural land.

The amount of land cover classes depends on the application. It could range from being artificial or natural to dozens or even hundreds. Corine, a land cover program initiated by the European Union defined 5 main land cover classes (Artificial, Agricultural, Forest, Wetlands, and Water). These classes are in turn divided into a total of 44 sub-classes [18]. The main classes and sub-classes can be seen in Figure 4.

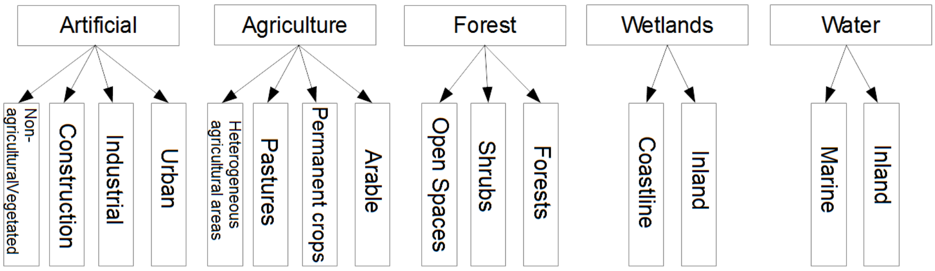


Figure 4. Land cover classes from Corine.

3.2. Land cover classification

Classification is the categorization of items into different classes. Classification done with predetermined classes is called supervised classification. This latter requires training data to learn a mapping function to the classes, additionally called labels. Unsupervised classification however does not require training data and the classes are not predetermined beforehand.

In the domain of land cover classification, satellite images or sections of them are labeled with the appropriate class. This is a challenging problem since many classes have similar features, in addition of having the same class appear different under different scales. There are various different approaches to be taken to classify remotely sensed images. The Machine Learning (ML) approach rely on feature extraction followed by the classification. Many ML methods have been applied in land cover and land use classification such as Random forests, k-means, Support Vector Machines (SVM), etc.

3.2.1. Classification methods overview

Classification methods in ML vary in complexity and effectiveness. Hence it is important to know which one would be best for the application at hand. These methods dubbed classifiers vary from linear classifiers such as logistic regression to more complex one such as DNN.

The most famous unsupervised segmentation method is the k-means classification. K-means starts from a preset number of classes known as centroids in the feature space[1] and finds the closest point to each centroid, then updates the centroid to be average point recursively. In supervised classification which is the one mostly used, several classifiers compete with each other. Linear classifiers try to find a separating line between the classes in a feature space. Decision Trees (DT) simply break down the data into subsets while building a tree related to those subsets to be options to decide from depending on the condition set by the DT. The k Nearest Neighbour (k-NN) method is a more elaborate method that uses distances in the feature space. It labels an item or point to a certain class based on how many neighbouring previously labeled points belong to that class. From Artificial Intelligence (AI) comes Neural Networks, also known as Artificial Neural Networks (ANN) containing what are called artificial neurons capable of mapping a linear function. ANNs consist of stacked artificial neurons with non linear functions aimed at finding the best mapping between the data and the corresponding class.

3.2.2. Classification of land cover

Land cover classification aims to find the dominant class in a satellite image. The research already done in the domain of image classification has been transferred to apply for land cover classification also known as Land Use and Land Cover Classification (LULC). Yang *et al.* [19] applied Bag Of Visual Words (BoVW) methods to classify satellite images. BoVW saves image features called words and checks the occurrence of each word in a histogram then selects which class is dominant. But like almost any other machine vision application, DNNs dominate the state of the art. With the launch of several high resolution imagery satellites available for everyone, and new datasets like the EuroSAT [20] it became possible for DNNs to thrive since they highly rely on voluminous data. Lei *et al.* [21] reviewed 173 scientific papers related to object based classification of remote sensing images concluding that although SVM and RF were good classifiers, DL showed potential and is expected to improve further.

3.3. Land cover segmentation

Land cover segmentation in remote sensing is a subset of image segmentation, which is defined as the partitioning of images with the goal of simplifying said images or labeling parts of them [22]. In land cover segmentation, areal or satellite images of the Earth's surface are segmented by land cover types.

[1] Feature space in an n-dimensional space containing variables mapped from the data through feature extraction

The difference between land cover segmentation and classification is that land cover classification labels patches of the image or the whole image as a class, while segmentation tries to outline the actual edges of the object (Figure 5).

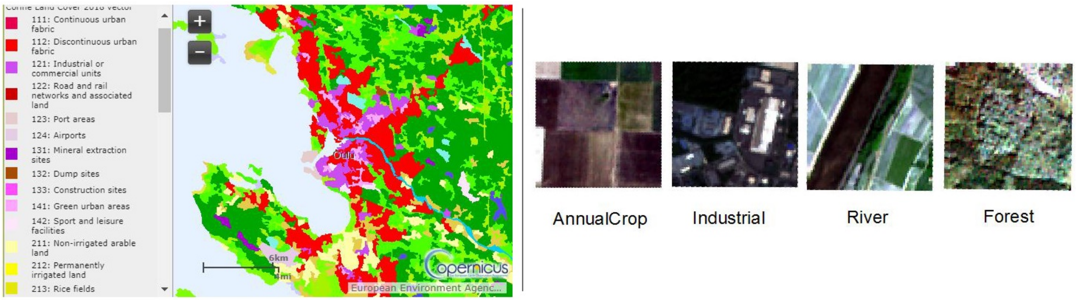

Corine Land Cover 2018 vector
🟥 111: Continuous urban fabric
🟥 112: Discontinuous urban fabric
🟪 121: Industrial or commercial units
🟥 122: Road and rail networks and associated land
123: Port areas
124: Airports
🟪 131: Mineral extraction sites
🟫 132: Dump sites
🟪 133: Construction sites
141: Green urban areas
142: Sport and leisure facilities
🟨 211: Non-irrigated arable land
🟨 212: Permanently irrigated land
🟨 213: Rice fields
+
−
Dubl
6km
4mi
copernicus
European Environment Agenc...
AnnualCrop
Industrial
River
Forest

Figure 5. Left: Land cover segmentation from Corine; Right: Example of land cover classes from EuroSat Dataset.

3.3.1. Segmentation methods

The segmentation can be done such as each pixel is given a class. This is called pixel-based segmentation. The case where the segmentation is done by region is called region based segmentation.

The non-semantic segmentation is mostly termed unsupervised segmentation referring to the segmentation where the algorithm does not have any prior information about the classes in the image. The previously mentioned K-means is one of the most well-known unsupervised algorithms, which belongs to the clustering algorithm types.

3.3.2. Semantic segmentation

Semantic segmentation is just image segmentation where the segments represent predetermined classes. It assigns a semantic label to each relevant segment in the image. Semantic segmentation is a part of supervised classification meaning that the algorithm has some prior knowledge about the classes in the form of Ground Truth (GT) segments. Among all the approaches ML methods are the leading ones in this category. They are algorithms that improve based on inference rather than being hard-coded. This requires a training data which contains the labeled pixels which the algorithm needs to learn to mimic. Notable ML based segmentation algorithms are DT, SVM, ANN, and from those come DNN which are the best performing amongst them [23] [24] [25]. Milestones in DL have gotten state of the art results in semantic segmentation time after time. Architectures such as Fully Convolutional Networks (FCN), ResNet [26] and Atrous Spacial Pyramid Pooling (ASPP) [27] have shown great improvement in the field.

Because of the vast possibility of applications that can be enabled via supervised segmentation a good classification requires experience and experimentation. Therefore the selection of the optimal segmentation scheme is crucial [28]. More often than not the objects semantically segmented in images have regular shape. In this case the pixel's spatial property is taken into consideration before assigning a label to it. In other words the segmentation is done as a group of pixels together. On the other end pixel based segmentation assesses each pixel individually.

Semantic segmentation methods are evaluated in different ways depending on the application. Even so in the same application domain diverse assessment metrics are used. Supervised and unsupervised methods for example cannot be compared directly.

3.3.3. Previous work

Land cover segmentation also referred to as image classification for land-cover mapping purposes is one of the most common applications of remote sensing, attracting significant attention in recent years. Over the last decade multiple studies and research have been conducted with different configuration of sensors, classification algorithms, and accuracy assessment methods. It is therefore important to find reviews that compare them against each other. Unfortunately comparative studies and surveys are quite uncommon in this domain, this does not come as a surprise since the methods of assessment for each study varies quite a bit from the others [29]. Qualitative assessment based on visual interpretation is a widely-used method but on top of being time consuming and subjective it makes it quite difficult to compare studies against each other. Räsänen *et al.* [30] concluded that one has to decide what is needed from the segmentation and use the according evaluation method with care.

A vast number of approaches have been followed in the field of LULC classification to find land cover mapping and changes. Tehrany *et al.* [31] compared 3 methods on an area of $1750km^2$. (1) DT which are a hierarchical classification method, coupling the Normalized Difference Vegetation Index (NDVI) [32] with information from the other spectral bands of the SPOT5 satellite to get a per pixel classification. (2) SVM classification that looks for a plane to separate classes in the feature space. (3) k-NN classifier. The assessment was done with the accuracy metric with k-NN being on top. More recently Khatami *et al.* [33] showed a comparative study of various supervised pixel-based segmentation methods, including SVM, DT, a simple Artificial Neural Network , maximum likelihood (a statistical model for classification), and others. The results they came out with showed that SVM outperformed all the other approaches with ANN not falling far behind but hinting that ANN would perform better with more data. To avoid the issue of different data type, study area, classifier properties, the results were reported only when comparisons have been done with the same data between pairs of classifiers.

Just like any other computer vision field DL has become a ubiquitous tool in land cover segmentation. The recent surge of high resolution free satellite data has given DNN and environment to thrive in. Otávio *et al.* [34] showed that CNNs vastly outperform the classical ML methods when it comes to land cover classification. DeepGlobe challenge, a challenge for RS segmentation akin to challenges in object semantic segmentation challenges such as COCO has been introduced in 2018 [35]. The challenge contains a track for land cover segmentation using images provided by DigiGlobe from WorldView satellites. The state of the art in this challenge is unsurprisingly dominated by DL. [36] holds the best result in the challenge with a deep fusion net [37]. [38]

used the state of the art semantic segmentation method to date and ended up with quite similar results to [36]. Hasan *et al.* [39] used similar architecture as [38] adding Lidar data that considerably improved upon the results.

3.4. Forest segmentation

While land cover segmentation refers to segmenting the images by the covers mentioned previously, forest cover focuses on segmenting forests by tree or vegetation species.

Forest cover segmentation is a more complex task compared to land cover segmentation due to the fact that the variance between tree species is much less apparent than the variance between other land cover classes. The challenge does not stop there, according to a study done by Beech *et al.* [40] there are over 60,000 tree species in the planet. Many of those species are very similar looking and some of which are so rare that they are countable in the single digits [40]. Adding to that, many tree species are location specific. In Europe alone the tree species vary a lot between north and south [41]. Hence it comes to no surprise that the studies on this field are scarce to say the least but by no means this is a new field. In 1985 Compton *et al.* [42] used spectral indices made by testing different combinations of the NOAA-7 satellite's spectral bands including NDVI to segment land covers in Africa including various vegetation cover zones such as the Savanna, tropical forest, and southern Sahel. The bulk of research done in forest segmentation is done alongside other land cover classes and only seldom the classes are exclusively tree species. Lung *et al.* [43] analyzed the development of land cover including various type of forest covers (Pine, Bischoffia, Terminalia etc,) using ML classifier, having varying accuracies depending on the species.

One of the studies that focuses solely on the classification and segmentation of tree species is the one done by Waser *et al.* [44]. Using airborne cameras (ADS40, RC30) they segmented airborne multispectral imagery and LiDAR data based of 4 to 7 tree species. Multinomial regression was used as the method of classification for the tree species. The usage of airborne cameras is existent in multiple studies [45] [46] [47] [48]. Classification and segmentation per forest type with satellite images is also existent albeit more leaning towards simple classification rather than semantic segmentation. This is largely due to the lack of datasets that provide labeled segmented data of tree species, and even if there is, it might be specific to an area not necessarily relevant to another study. Some of the freely available datasets are the Forest type mapping Data Set [49] for classification purposes containing Japanese species of forests. The National Forest Type Dataset [50] is used for semantic segmentation with a $250m/px$ resolution of the entire United States (Figure 6). Using [49] Sabanci *et al.* [51] compared multiple machine learning methods in the classification of forest types in Ibaraki prefecture, Japan, with at least four tree species having an ANN also known as Multi Layer Perceptron (MLP) on top followed by k-NN. The study fails to explains the architecture of the MLP though. In a more recent study by Pasquarella *et al.* [52] RF were used to segment spectral-temporal Landsat images from western Massachusetts in the U.S. of eight forest types.

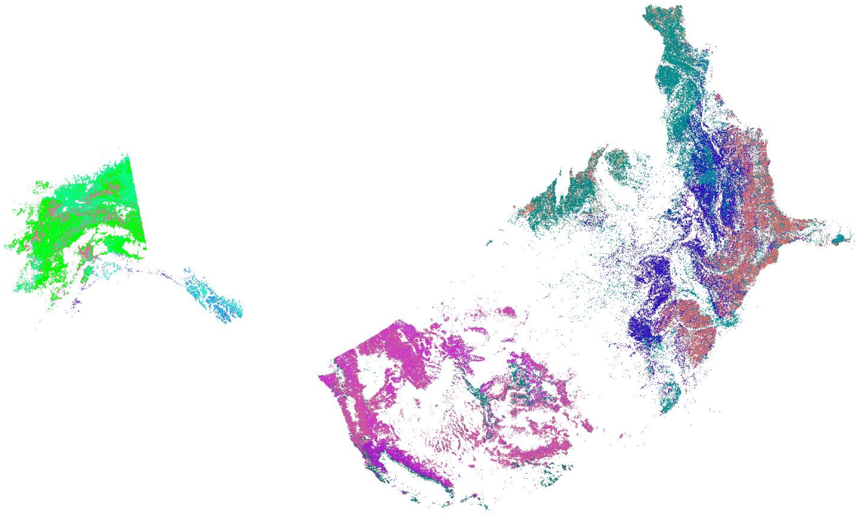


Figure 6. National Forest Type Dataset raster over the U.S. [50].

The main issue that is apparent when comparing these previous studies is that they all are in different areas with varying forest types, and on top of that they vary with the number of species they included in the research. Segmenting land cover as forest and non-forest for example could be considered forest cover mapping but it is a far easier task than segmenting by the specific tree species type. The assessment methods range from visual interpretation to metrics such as accuracy which in turns makes it difficult to compare but nonetheless it is important to note that forest cover segmentation is an ongoing research lacking GT data.

4. DEEP LEARNING BASED LAND COVER SEGMENTATION

4.1. Deep Learning

DL is a subset of machine learning which in itself is a subset of artificial intelligence. DL is based on DNNs composed of a series of artificial neurons mimicking the biological neurons in the brain. Unlike machine learning where the features are extracted before the classification step, DL does everything in one network. That makes DL very versatile in tackling many domains.

DNNs are ANNs with multiple hidden layers. ANNs have different forms but all are based upon artificial neurons, each one of those neurons sums the product of its inputs with a specific weight (Figure 7). What makes ANNs differ from each other is how these neurons are connected to each other. Fully connected networks is the most well-known architecture where each neuron is connected to all the neurons in the previous one.

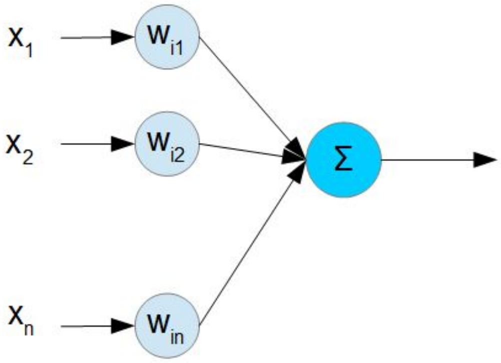


Figure 7. Artificial Neuron (right) and its corresponding weights (left) with inputs x_1 to x_n.

The idea behind DNNs dates from the late 70s but hasn't been implemented efficiently until 2012. That delay was caused by both hardware limitation and algorithm limitations. Nowadays DNNs are trained with Graphical Processing Units (GPUs) providing fast multi-core solutions. Algorithm-wise the main culprit in preventing NN from being too deep is the vanishing gradient issue [53]. The vanishing gradient occurs when gradient descent (Figure 8) is applied in training the DNN, due to the nature of it of using backpropagation which is the backward pass through the network to update its parameters. Gradient descent updates the parameters following this equation:

$$\theta_j := \theta_j - \alpha \frac{\delta}{\delta \theta_j} J(\theta)$$

where $J(\theta)$ is the cost function, α is the learning rate, θ is the parameter updated. One of the main ways the vanishing gradient "vanished" is the adoption of different network architecture such as Convolutional Neural Networks (CNN), Residual Nets (ResNet) and activation functions such as Rectified Linear Units (ReLU).

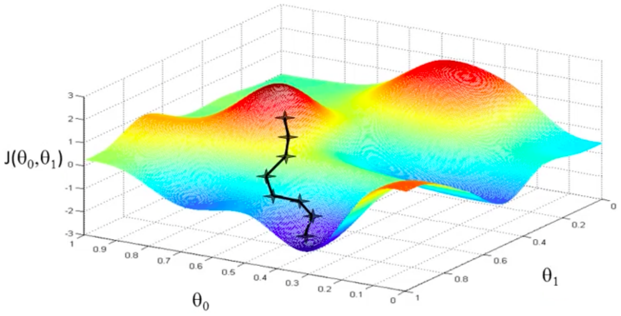


Figure 8. Gradient descent(shown as steps in graph) through the cost function.

4.1.1. CNN

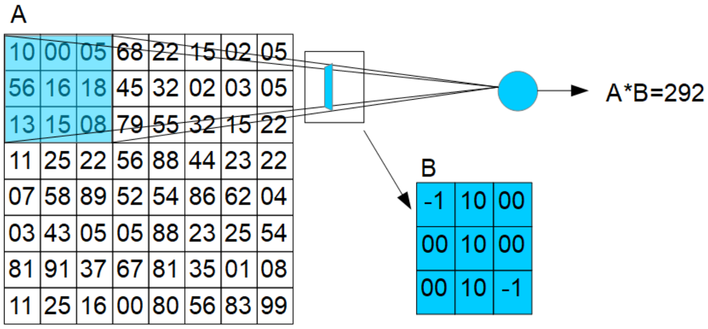
The concept for CNNs was introduced in 1989 [54], they were represented as being inspired by the biological visual cortex. CNNs differ from fully connected ANN by having each neuron being connected to a limited number of neuron in the previous layer. CNNs assume the input is an image and looks for features through a kernel which is the number of inputs the neuron is connected to. This is described as the size of input the neuron "can see". The weights are shared amongst all these neurons in the layer to have the same kernel (Figure 9). This gives the ability to create a feature map which shows the activation where the specific feature the kernel is looking for exists. The detection is done through convolution between the input and the kernel thus the term convolutional neural networks, the equation for that convolution is:

$$F(n) = (x * w)[n] = \sum_{a=-\infty}^{+\infty} x[a]w[a+n]$$

where w is the kernel and x is the input. This equation is the cross correlation equation as opposed to the actual discrete convolution

$$F(n) = (x * w)[n] = \sum_{a=-\infty}^{+\infty} x[a]w[a-n]$$

however this is not an issue, more of a notation issue but since the weights are trainable they would simply adjust accordingly.



10	00	05	68	22	15	02	05
56	16	18	45	32	02	03	05
13	15	08	79	55	32	15	22
11	25	22	56	88	44	23	22
07	58	89	52	54	86	62	04
03	43	05	05	88	23	25	54
81	91	37	67	81	35	01	08
11	25	16	00	80	56	83	99

-1	10	00
00	10	00
00	10	-1

Figure 9. A neuron in CNN, Left is the input image with A representing the area the neuron can see; B represents the kernel.

The repeated kernel makes a convolutional filter. A set of stacked convolutional filters makes a convolutional layer. Convolutional layers are followed by activation functions whose job is to have linear or non linear output from each conv layer. Each convolutional layer is tasked to extract features from the input in the layer previous to it, so in order to detect higher level features pooling layers are introduced. Pooling layers reduce the resolution of the previous input to combine several features for the next convolutional layer.

A CNN thus contains a stack of convolutional layers followed by activation functions and pooling layers, and finally an addition of one or more Fully Connected (FC) layers [55]. FC layers are useful for the purpose of classification when the goal is to select amongst a set of finite classes for each image.

CNNs showed great improvement over FC ANNs in image processing and proved to be the "go-to" architecture in the field of computer vision. It reduced the amount of parameters needed while processing images compared to FC networks. The shared weights not only makes CNN translation invariant but it also drastically decreases the amount of computations needed. CNNs are presently so ubiquitous that it is now almost impossible to find a DNN that is not based upon them.

4.1.2. Activation functions

Activation functions are the answer to parsing non linear functions into the ANN. They are put after the output of neurons as an additional condition on the result (Figure 10). Many options exist for activation functions such as the sigmoid function, tanh function, ReLU. Although each activation has its advantages and disadvantages the ReLU activation function reigns over the others in the DL field thanks to its simplicity and its nature being linear makes it avoid gradient vanishing [56]. The ReLU's function is as follows:

$$\varphi(x) = max(0, x)$$

It is now considered the go to activation for deep learning in a vast number of applications [57]. Other variations of ReLU exist including Leaky ReLU, Noisy ReLU, etc. All of which has an advantage that aims to improve upon ReLU or combine with the advantages with other activation functions. The rule of thumb is only use the alternatives when ReLU is not performing well.

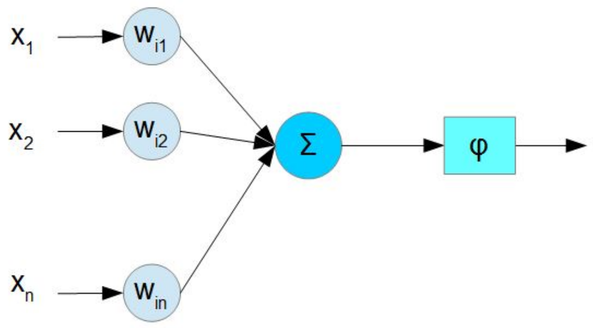


Figure 10. Artificial neuron with activation function φ.

4.2. Typical DL architectures

DL has improved and developed to the point where one would not just stack a bunch of layers on top of each other and hope for good results. Many architectures have been tailored for specific functions just like how CNNs are best for images. This created a set of well-known DNN architectures that hold state of the art results in various fields.

4.2.1. Residual Networks

Although CNN and ReLU made it possible for ANNs to become deep it was not without issues. The degradation problem caused the accuracy of DNNs to be saturated with the depth increasing, then degrades rapidly. Kaiming *et al.* [2] explained the reason behind it by presenting a shallow network and a deeper counterpart with the same desired output. The extra layers should improve upon the shallow version and in the worst case they should have the same performance but in practice the deeper counterpart degrades. The reason behind this is that instead of linear mapping it is much harder to approximate it with non linear functions. If the function of the ANN is $y = F(x)$ and $y = x$ then there should be a function $F(x) = x$, but adding a residual connection such as $y = F(x) + x$ makes the function $y = x$ as simple as making $F(x) = 0$. (Figure 11) shows a shallow ANN and its deep counterparts, residual and non residual.

Thanks to ResNet DNNs are able to grow deeper than the networks preceding it, before it VGG19 and GoogleNet were the deepest nets used with 19 and 22 layers respectively, while ResNet manages to have over 100 layer. Many sample ResNet architectures have been proposed including ResNet 18, 34, 50, 101, and 152 where the numbers represent the depth of the network. These networks are presented as stacks of residual convolutional layers known as blocks. Layers in the same block share the same depth for the most part (Figure 12). ResNets proved to be powerful when they

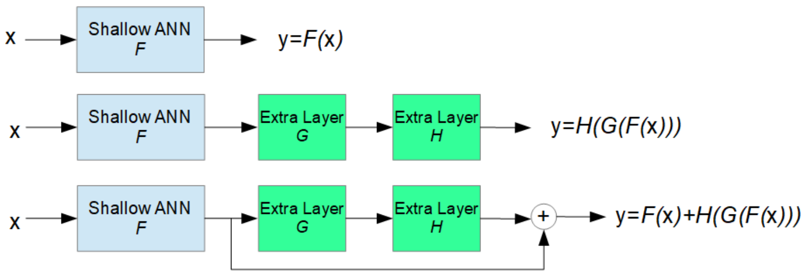
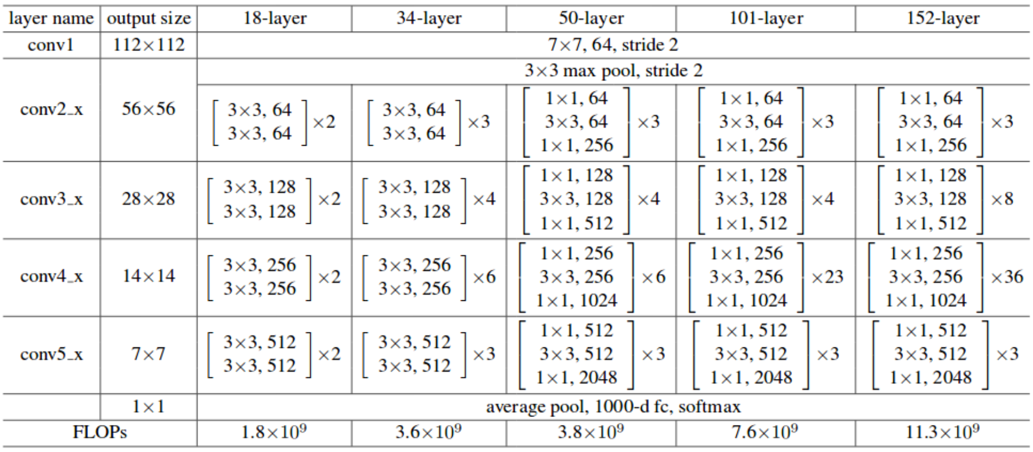


Figure 11. Top: shallow ANN; Middle: Deep counterpart; Bottom: same Deep counterpart with a residual connection; if all networks have the same result it would be hardest for the middle network to learn the mapping while the bottom one only sets G and H to 0.



layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		[3×3, 64 3×3, 64] ×2	[3×3, 64 3×3, 64] ×3	[1×1, 64 3×3, 64 1×1, 256] ×3	[1×1, 64 3×3, 64 1×1, 256] ×3	[1×1, 64 3×3, 64 1×1, 256] ×3
conv3_x	28×28	[3×3, 128 3×3, 128] ×2	[3×3, 128 3×3, 128] ×4	[1×1, 128 3×3, 128 1×1, 512] ×4	[1×1, 128 3×3, 128 1×1, 512] ×4	[1×1, 128 3×3, 128 1×1, 512] ×8
conv4_x	14×14	[3×3, 256 3×3, 256] ×2	[3×3, 256 3×3, 256] ×6	[1×1, 256 3×3, 256 1×1, 1024] ×6	[1×1, 256 3×3, 256 1×1, 1024] ×23	[1×1, 256 3×3, 256 1×1, 1024] ×36
conv5_x	7×7	[3×3, 512 3×3, 512] ×2	[3×3, 512 3×3, 512] ×3	[1×1, 512 3×3, 512 1×1, 2048] ×3	[1×1, 512 3×3, 512 1×1, 2048] ×3	[1×1, 512 3×3, 512 1×1, 2048] ×3
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 12. Architecture of well-known ResNet networks.

won the The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2016 with $96{,}4\%$ accuracy.

ResNets uses Batch Normalization (BN) [58] layers whose job is to normalize all the features of the layer before it. To put it simply this is done by subtracting the batch mean and dividing it by the standard deviation. However the actual equation is a little bit more complex than that:

$$BN_{\gamma,\beta}(x_i) \equiv \gamma \frac{x_i - \mu}{\sqrt{\sigma + \epsilon}} + \beta$$

where γ, β are trainable parameters and μ, σ are the mean and variance of the batch. The point in using BN is reducing the amount by which the hidden layers unit values shift, also referred to as covariance shift [58]. This serves as not only increasing the maximum accuracy but also increase the speed by which the network reaches it (more than 10 times faster [58]).

4.2.2. Variational Auto-Encoders

The DNNs mentioned previously take data in any form, mostly images and converts them to a dense tensor. This is useful in classification when the decisions are categorical and limited. But in the case of a more complex desired output such as generating new images or semantic segmentation, that approach is not enough. Variational Auto-Encoders (VAE) solve this issue by having another network after the classifier called the decoder network. The first network in this case is called the encoder network. VAEs go from a dense representation and learns to generate data from it. Figure 12 shows an example of a VAE network compared to a normal classifier.

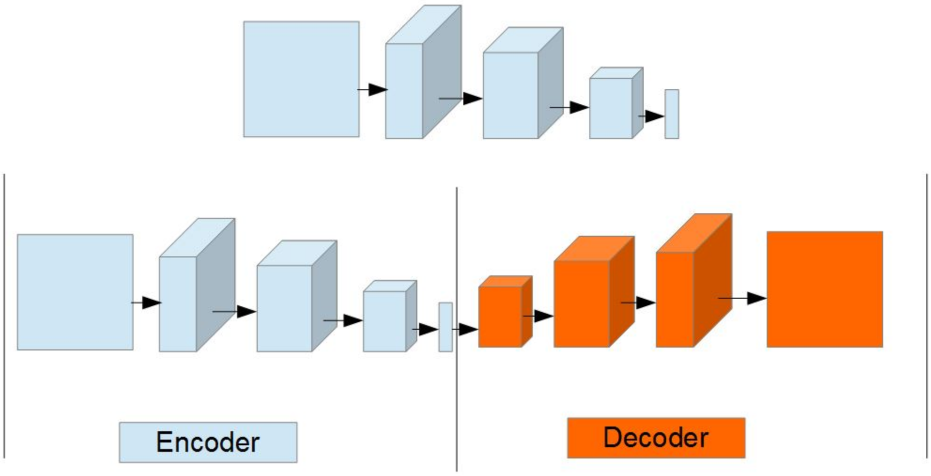


Figure 13. *Top:* Classifier network. *Bottom:* VAE counterpart.

The decoder network is usually a mirror copy of the encoder with pooling layers replaced by upsampling layers and convolutional layers replaced with deconvolutional layers. Deconvolutional layers or transposed convolution layers do the opposite of convolution where they go from a dense matrix into a higher resolution one [59].

4.2.3. Transfer Learning

Transfer learning is a method used in ML and by extension DL where a model developed for a certain task is used as the starting point for a different task. Transfer learning in DNN is widely used, so much so that using a pretrained model is the norm. The reason why transfer learning is used so much is the lack of datasets since DNNs require a large dataset. ImageNet is an example of such dataset containing over 1 million images with 1000 classes. Even if a dataset large enough is available, it would take quite a long time on most GPUs to train a DNN such as ResNet101 from scratch.

There are different types of transfer learning, of which three are mainly used. Pretrained models is probably the simplest, it consists of taking the weights from a checkpoint and starting with those on a different dataset. This will permit the network to be fine-tuned completely. This approach however requires a large enough dataset to prevent overfitting. In the case of a smaller dataset the second type of transfer learning would fit better. Fine-tunning is how it is usually referred to, instead of retraining the whole network some layers would be frozen, usually the first layers responsible for generic features. The third type is where the whole DNN is frozen and treated as a feature extractor. In this case only the FC layers are trained on the new data. This is extremely useful when the new data resembles the data trained on a lot.

4.3. Semantic Segmentation with DL

Semantic segmentation is a very important step in scene understanding used in various fields. Therefore it is crucial to have the best result possible in it. Semantic image segmentation has been tackled with traditional computer vision methods but with the rise of deep learning the tables have turned on the previous methods [60]. CNNs have been the backbone of any DNN applied in this field. [61] [62] [63] [64] [65] along with many more researches show how CNN has taken over that field.

4.3.1. Atrous Spatial Pyramid pooling

Consecutive pooling and strided convolution in deep CNNs are helpful in capturing more complex features but causes the resolution to be reduced which impacts the semantic segmentation negatively. Atrous convolution is a solution to overcome that problem [27]. Atrous [1] convolution removes the downsampling caused by strided convolution while keeping the ability to see larger features by increasing the filter size and inserting holes between the weights (Figure 14). The appearance of atrous convolution gives it the name of dilated convolution. This architecture permits to eliminate the need for pooling layers by increasing the field of view without decreasing the spatial dimensions. The amount of holes between weights defines at which scale the convolution is done. A normal convolutional kernel is an atrous kernel with rate 1, while rate 2 means there's one hole between each weight. a 3×3 atrous kernel with rate 2 has the same field of view as a 5×5 convolutional kernel (Figure 14).

Atrous convolution goes one step further to handle the problem of objects at different scales. This is solved by Atrous Spatial Pyramid Pooling (ASPP), where parallel atrous convolutional layers having different rates are used to capture information at different scales (Figure 15).

4.3.2. VAE architectures in semantic segmentation

Due to the nature of semantic segmentation requiring high resolution features to be kept the encoder-decoder architecture has been a favorable one for that purpose. Networks such as DeconvNet [66] where the encoder network is a VGGNet [67]. At the decoder network deconvolution layers are used to recover high resolution features for

[1] Atrous means with holes or holed in French.

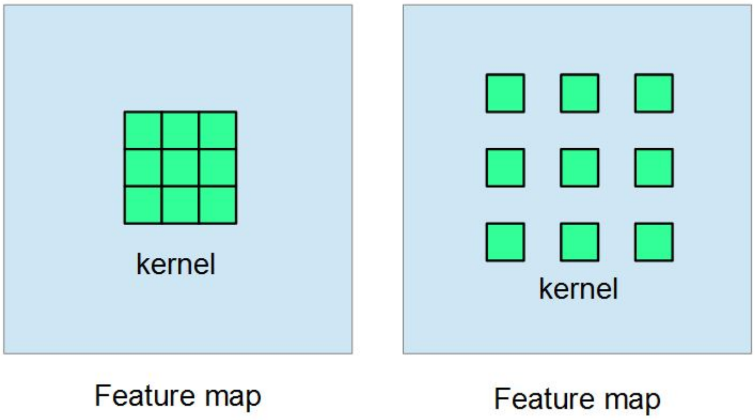


Figure 14. *Left:* typical 3×3 convolutional filter. *Right:* Atrous 3×3 convolutional filter.

the semantic segmentation. SegNet [68] is another network with the encoder decoder architecture using VGGNet as the encoder network. SegNet differs from DeconvNet by omitting the fully connected layers to conserve the high resolution features and drastically improve the performance. The decoder is a mirrored version of the encoder, having the pooling layers replaced by upsampling layers which are connected to the pooling layers of the encoder using shortcut connections (Figure 16). The decoder is followed by a softmax layer for a pixelwise segmentation. U-Net [69] is a semantic segmentation network used mainly for biomedical application. The encoder is this network is called the contraction path, while the decoder is called the expansion path. The contraction path is a classifier while the expansion path is a series of deconvolution layers that are connected to the encoder before every max pooling layer. The final layer is a 1×1 convolutional layer to reduce the dimensions to 2 (cell and membrane). Residual Encoder-Decoder Network (RED-Net) [70] is an example of a recent iteration in the encoder-decoder architecture used for semantic segmentation. The network used as an encoder in this case is ResNet. Skip or shortcut connections are made between the encoder and decoder, which helps in recovering high resolution features that could be lost due to pooling or strided convolution. Many more networks use the encoder-decoder architecture [71] [1] for semantic segmentation making it the go to architecture for that purpose.

4.3.3. DeepLab

DeepLabv3 is the current holder of the state of the art in semantic segmentation on the PASCAL VOC 2012 and Cityscape dataset [27]. It is the third iteration of DeepLab by Google thus the number 3 at the end of the name. It comprises a ResNet network either ResNet-50 or ResNet-101 with the final layers omitted and replaced with ASPP. This permits the output stride [2] to be 16 with ASPP rates $(6, 12, 18)$ or 8 with ASPP rates $(12, 24, 36)$ then the resulting logits are upsampled to the GT resolution.

[2] Output stride is the ratio between the input dimension and the output dimension.

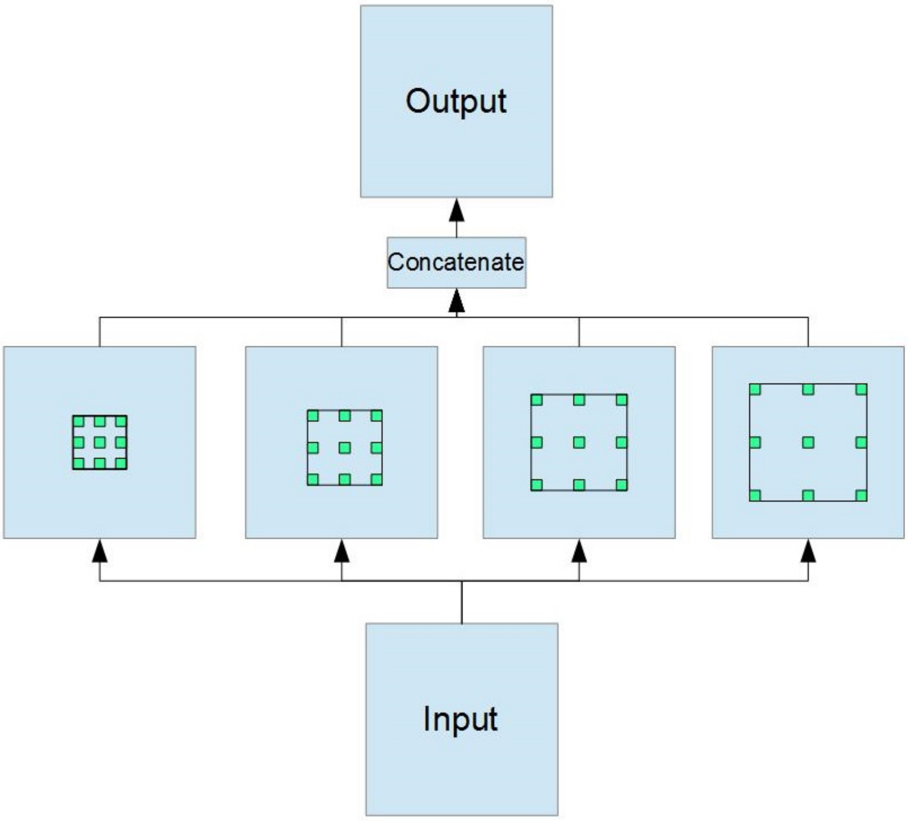


Figure 15. Atrous spatial pyramid pooling.

DeepLabv3 ads batch normalization to the ASPP layers which requires large batch sizes. This was solved by having the output stride 16 instead of 8.

4.3.4. Networks for Land Cover Segmentation

The idea for a network entirely specific to land cover segmentation has not been completely developed. Therefore the architectures mentioned previously are as much land cover segmentation networks as they are cityscape segmentation networks. That being said the nature of land cover classes limits the networks to a certain type.

The encoder-decoder network architecture based on CNNs is a dominating common point in the land cover segmentation field with multiple research published using it [72] [73] [74] [39]. This is largely due to objects like forest stands not having regular shapes and thus the need for high resolution features.

The issue of a lack of an imprecise dataset to train a land cover DNN is a limiting factor in the domain. Maggiori *et al.* [75] tackled the issue by training on a large dataset of inaccurate labels and then refined the results with a small set of accurate labels.

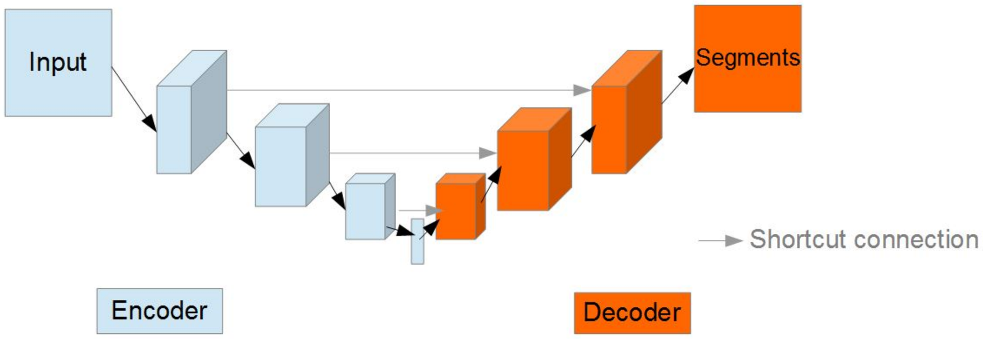


Figure 16. Encoder-Decoder Network for semantic segmentation with shortcut connections.

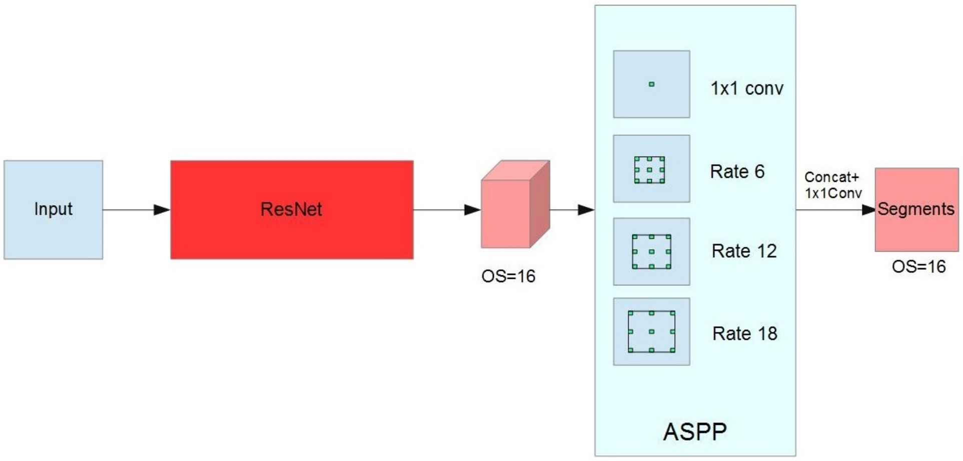


Figure 17. DeepLabv3.

5. EXPERIMENTAL EVALUATION

In this chapter we will tackle experimentation on handling the task of land cover semantic segmentation on satellite imagery using the DL tools discussed previously. It is clear that not a single method is exactly better than another in this field and that research and experimentation still has to be done to find the best solution. All we can deduce from the literature review is that state of the art semantic segmentation on all types of objects is dominated by DNN. The encoder-decoder architecture is yet another shared method. On top of all that ASPP has proved to be an excellent replacement to FCN in order to avoid losing information by reducing the resolution. Another thing that will be explored is the usage of all the data from the EO satellite images which usually have a radiometric resolution of 12bit floating point. Therefore encoding the images in 16bit unsigned integer format would keep all the information held by the data, whereas almost all other research focuses on using 8bit images. The main reason 8bit is used is to make it easier to transfer the learning from pretrained networks on non-remote sensing objects. This will limit out ability to transfer the learning which will require us to use a large amount of data.

5.1. Datasets

5.1.1. Sentinel-2 data

The level 3 Sentinel-2 data is provided for free on the Copernicus Open Access Hub from which 14 rasters were downloaded to build the dataset. The images selected were all captured during the summer, and with 0% cloud cover. The area covered by these rasters is a little over $157000km^2$ mostly captured in 2018 (Table 3). The exact area covered is shown in Figure 18.

Table 3. Rasters taken from Sentinel-2 satellites

Year	Number of rasters	Satellite
2015	3	Sentinel-2A
2017	2	Sentinel-2B
2018	9	Sentinel-2A

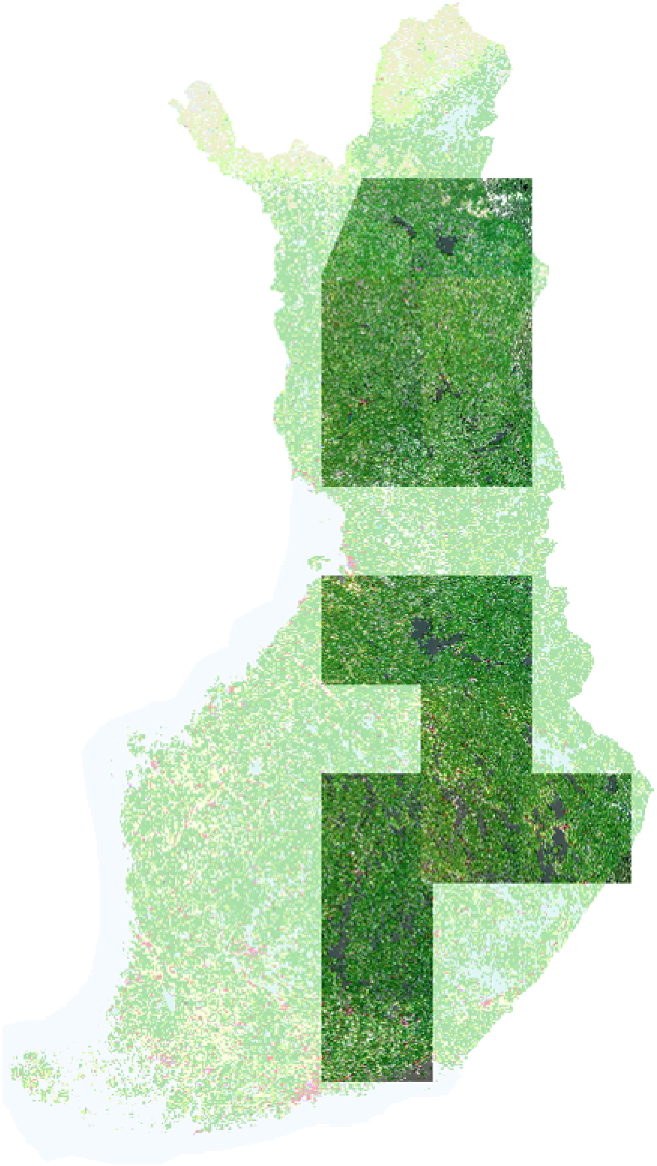


Figure 18. Sentinel-2 rasters downloaded from Finland area.

5.1.2. Pleiades-1 data

A total of 6 level 3 rasters from the Pleiades constellation were used. Just like the Sentinel-2 data the images selected were all captured during the summer of 2015, and with 0% cloud cover. The area covered by these rasters is a little over $52000km^2$. Figure 19 shows one of these rasters.

Unlike Sentinel-2 data Pleiades data is mostly not available for free which explains the reason why not so much data was used compared to the Sentinel-2 data.

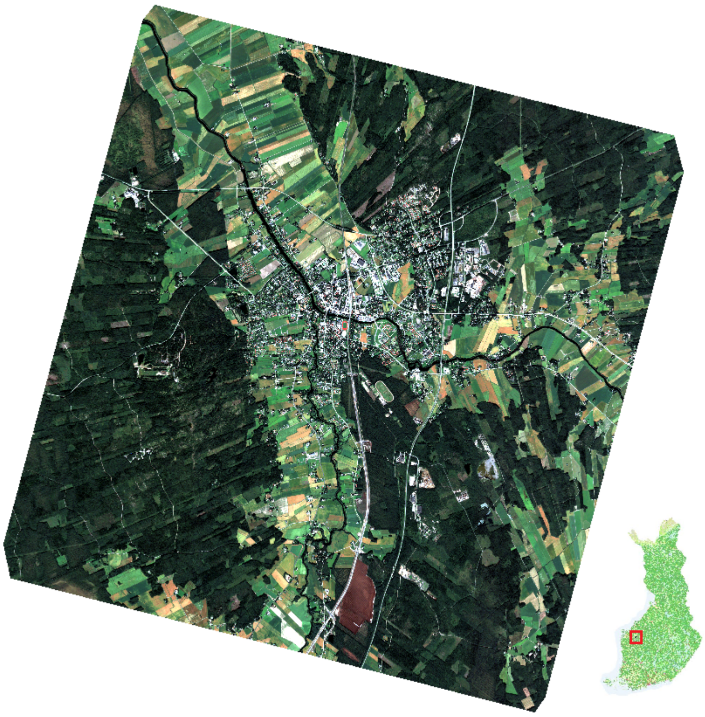


Figure 19. Pleiades-1 raster sample.

5.1.3. Label data

The GT segments were taken from the Corine Land Cover(CLC) 2018 [76] which has a $100m/px$ resolution. However there exists a version exclusively on the area of Finland with a $20m/px$ resolution [77]. CLC contains 44 classes ranging from man-made covers such as industrial areas or buildings to natural covers such as forests and grasslands. The version used in this experiment is a modified version with 17 classes instead shown in Table 4. These classes sum together some classes and omit others, for example all urban fabrics including airports, ports, buildings, and roads and bundled into one class.

5.2. Preprocessing

5.2.1. Raster to images conversion

From the Sentinel-2 data downloaded the first step taken in preprocessing it is selecting the appropriate bands. The bands selected were $2, 3, 4, and$ 8 which corresponds to Blue, Green, Red, and NIR. all these bands have a resolution of $10m/px$ (Table 1). Next step was the combination of the bands using the Geospatial Data Abstraction Library (GDAL) and taking patches from these rasters with 224×224 resolution which comes at about 30567 images. The images were saved as 3 channel RGB images in PNG format and 16bit unsigned integer depth and a single channel NIR image with

Table 4. Classes of the modified CLC2018

Class number	Class name	RGB value
1	Buildings	229 0 76
2	Artificial greenery w/ buildings	255 220 154
3	Artificial greenery	254 229 254
4	Arable land	245 254 167
5	Olive groves	241 165 0
6	Pastures	241 241 67
7	Agriculture mixed with natural vegetation	230 204 77
8	Agro-Forestry area	242 204 166
9	Broad leaved forest	127 254 0
10	Coniferous forest	0 165 0
11	Mixed forest	0 190 0
12	Natural grasslands	203 229 76
13	Moors and heathland	205 205 102
14	Transitional woodland	165 229 0
15	Sand and open spaces	220 220 220
16	Wetlands	165 165 254
17	Water surfaces	0 203 229

16bit unsigned integer depth to preserve all the data from the 12bit floating point depth sensor onboard the Sentinel-2. The conversion of the data type was done using the Geographic Information Systems (GIS) software QGIS.

The same steps were taken with the Pleiades data. The RGB, and NIR bands with $2m/px$ were used. The data format was converted to 16bit. The rasters were divided in 224×224 patches resulting in 3591 image. These images were saved as 3 channel RGB 16bit Portable Network Graphics (PNG), and 1 channel NIR 16bit PNG.

5.2.2. Data Augmentation

$30K$ images is not quite enough to train a deep neural network. Therefore data augmentation was needed to increase the size of the dataset. The augmentation consisted of rotation and mirroring to get the data to 152835 images. From this data 80% of it was used for training and 20% was used for validation.

The same procedure was repeated for the Pleiades data, which was augmented to 21546 using rotation and mirroring. Similarly 80% of it was used for training and 20% was used for validation.

5.2.3. Label data

In addition to the class combinations mentioned previously, the preprocessing done on the CLC 2018 consisted of upsampling due to it being half the pixel resolution of the Sentinel-2 data and the tenth of the Pleiades-1 data. It was also augmented in the same fashion as the RGB and NIR images. It was encoded to a single channel PNG where each pixel represents the number of the class.

5.2.4. Final input

Lastly the data was inputted in the form of a 4 channel 16bit tensor with the mean of each channel subtracted for normalization. The dataset was saved as TensorFlow Records (TFR) which allows for various and flexible data formats [78].

5.3. Architectures

The architecture used in this experiment to train on the Sentinel-2 dataset is based on the DeepLabv3 network [27] and the runner up of the DigiGlobe challenge [38]. It comprises of an encoder-decoder architecture with a ResNet network as its main encoder network. It replaces the FC layers with ASPP layers. Since this is just like DeepLabv3 there is not much need to a decoder network but as shown by [38] the result could improve by having a decoder to recover even higher resolution features. Therefore a decoder network containing 2 convolutional layers and an upsample layer to recover features from the first and second ResNet block was added. The full architecture can be seen in Figure 20. The ResNets tested were ResNet-18, ResNet-34, and ResNet-50 to explore how different depths of the DNN would affect the results. ResNet-101 and ResNet-152 were omitted due to the lack of data.

Although using a pretrained network on any dataset would be beneficiary the nature of our data being 4 channel instead of 3 makes using a pretrained set a bit tricky. There are a few solutions to that such as reducing the number of channels or adding a layer before the ResNet. However, it was chosen to train from scratch since land cover images are different than ImageNet which is the dataset mostly used in pretrained ResNets and having 150k images could prove large enough to train ResNet-50.

5.4. Evaluation metrics

The next step is to choose the assessment method. Semantic segmentation uses a varying set of metrics to measure how accurate it is compared to the GT data. Those include Mean Intersection over Union (MIoU), Average Precision (AP), Pixel Accuracy (PA), and Boundary F1 (BF) score. In this experimentation two of those were used MIoU and PA.

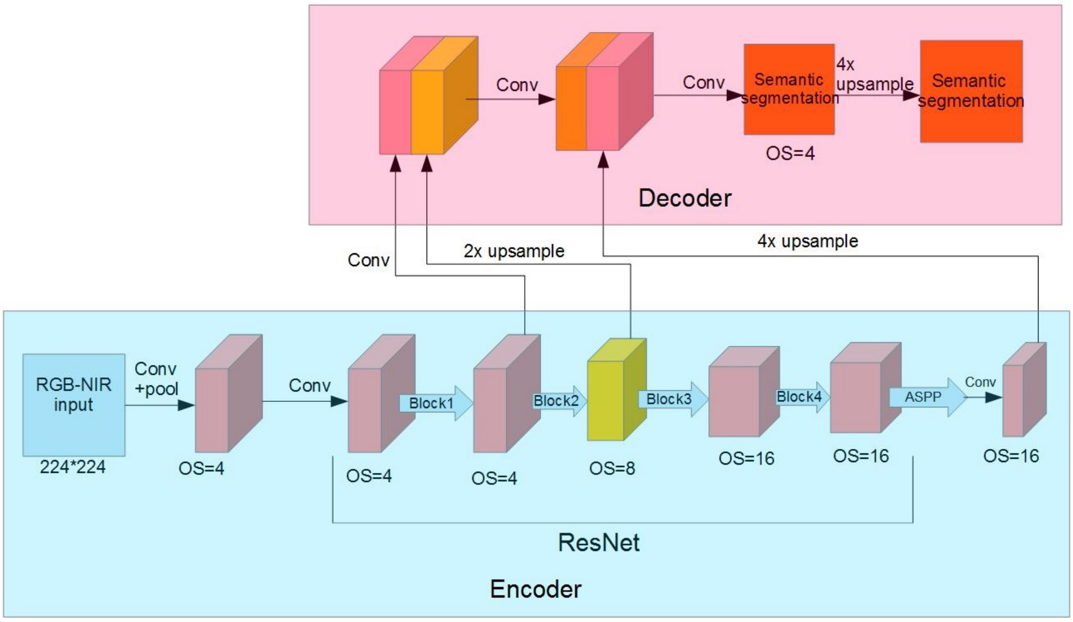


Figure 20. Architecture used.

5.4.1. Pixel accuracy

PA is defined as the percentage of pixels in the image that are correctly classified. It is calculated using the following equation:

$$pixel\_accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

where TP means true positive which represents a pixel that is correctly predicted to belong to the given class. TN is true negative representing a pixel that is correctly identified as not belonging to the given class. FP is false positive, it is the pixels wrongly predicted to belong to a certain class. FN means false negative representing pixels wrongly predicted to not belong to the given class.

5.4.2. Mean intersection over union

MIoU is a widely used metric for the evaluation of semantic segmentation. It computes the mean of the rate of overlap between the GT segments and the resulting segmentation. MIoU is obtained using the following equation:

$$MIoU = \frac{1}{n}\sum_{i=1}^{n}\frac{GT_i \cap Output_i}{GT_i \cup Output_i}$$

where n is the number of classes. Another way to write the formula is:

$$MIoU = \frac{1}{n}\sum_{i=1}^{n}\frac{TP}{TP + FP + FN}$$

5.5. Training

5.5.1. Sentinel-2

The training was performed on an Nvidia Tesla V100 GPU [79] with 16GB of video memory. The training covered 315 epochs with a batch size of 32 resulting in about $1{,}2 million$ iteration. The batches were randomized for every iteration in the epoch.

The training was performed on 3 ResNet architecture, ResNet-18, ResNet-34, and ResNet-50 to test the effect of the depth on the results. All using the same data, learning rate, optimization function, and loss function. The learning rate was set to decrease polynomialy from 7×10^{-3} to 1×10^{-6}. The optimization function was to the momentum optimizer [80] . Lastly the loss function used is as follows:

$$loss = \sum GT \log(out) + W_D \sum_{i=1}^{n} \frac{V_i^2}{2}$$

where $\sum GT \log(out)$ represents the cross entropy with out being the model output and GT the ground truth. W_D is the weight decay and $\sum_{i=1}^{n} \frac{V_i^2}{2}$ is the L2 loss for each variable V_i.

No transfer learning was performed for the Sentinel-2 data therefore the weights were initialized randomly.

5.5.2. Pleiades-1

The training was performed on the same Nvidia Tesla V100 GPU. The number of epochs was 315 with a batch size of 32 resulting in about $1{,}2 million$ iteration. The batches were randomized for every iteration in the epoch.

In addition to training on the Pleiades-1 data from scratch transfer learning was applied by initializing the weights from the Sentinel-2 network to tackle the issue of limited data. This is done to compare how well would transfer learning work from a different satellite with less pixel resolution yet more data.

5.6. Results

5.6.1. Sentinel-2

The results after training the network with the three different architectures for ResNet on the Sentinel-2 data is shown in Table 5 displaying both MIoU and PA scores with the time it took for the training to converge. To translate the numbers in Table 5 Figure 21 and Figure 22 shows the difference in segmenting a test raster.

Table 5. Results with Sentinel-2 data

Architecture	MIoU	PA	Time to train
ResNet-18	0.30	0.72	2d
ResNet-34	0.30	0.72	2d10h
ResNet-50	0.52	0.78	7days

5.6.2. Pleiades-1

The results after training the the pretrained network with the three different architectures for ResNet on the Pleiades-1 data are shown in Table 6. Figure 23, Figure 24, and Figure 25 show the difference in segmenting test images from Pleiades-1.

Table 6. Results with Pleiades-1 data

Architecture	MIoU	PA	Time to train
ResNet-18	0.34	0.76	5h
ResNet-34	0.35	0.76	12h
ResNet-50	0.33	0.75	16h
ResNet-18TL	0.39	0.78	5h
ResNet-34TL	0.40	0.78	9h
ResNet-50TL	0.85	0.98	15h

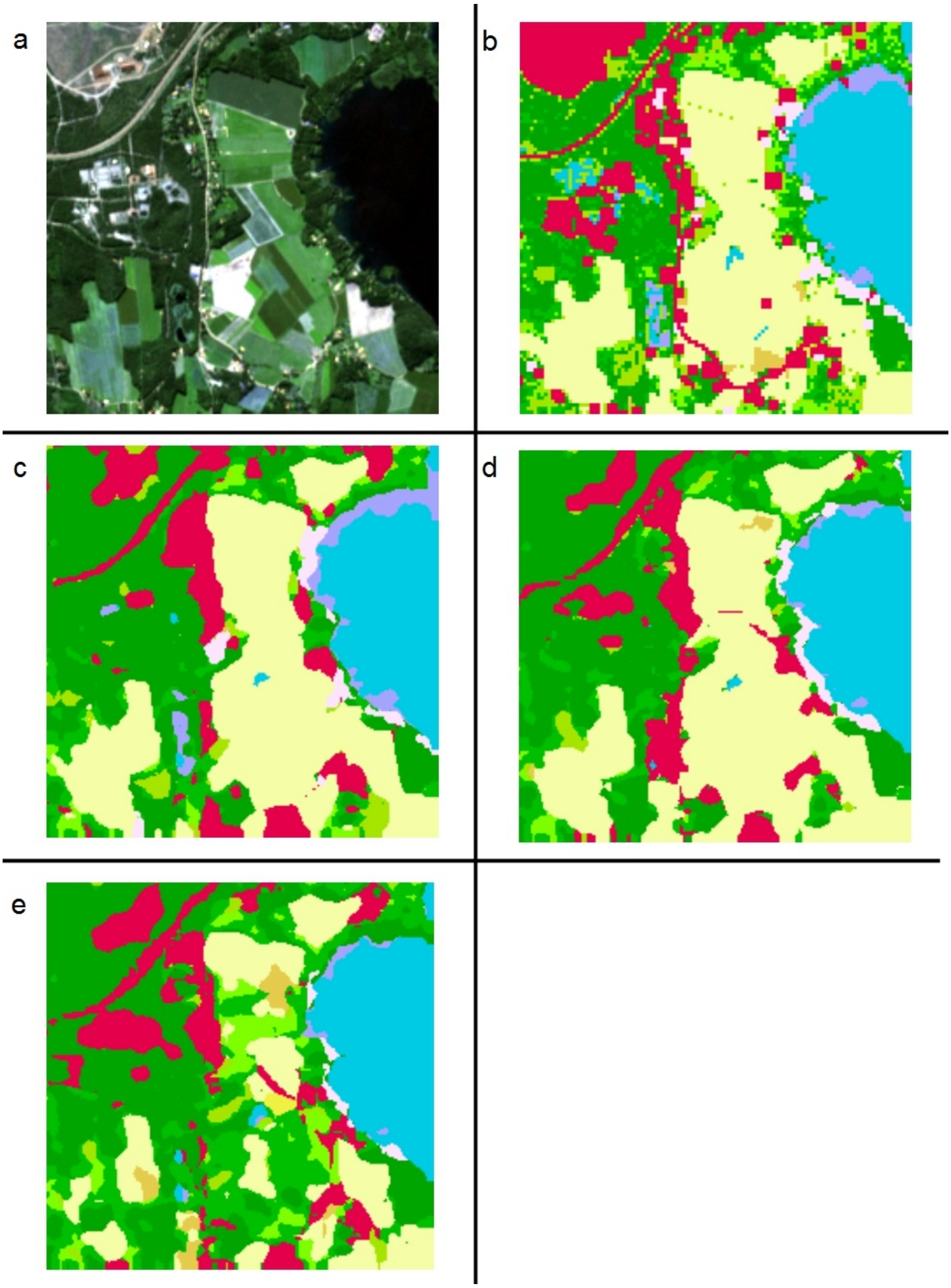


Figure 21. Sample output of model from Sentinel-2 test image. *a:* Sentinel-2 image. *b:* GT label from modified CLC. *c:* Output of ResNet-50 based model. *d:* Output of ResNet-34 based model. *e:* Output of ResNet-18 based model.

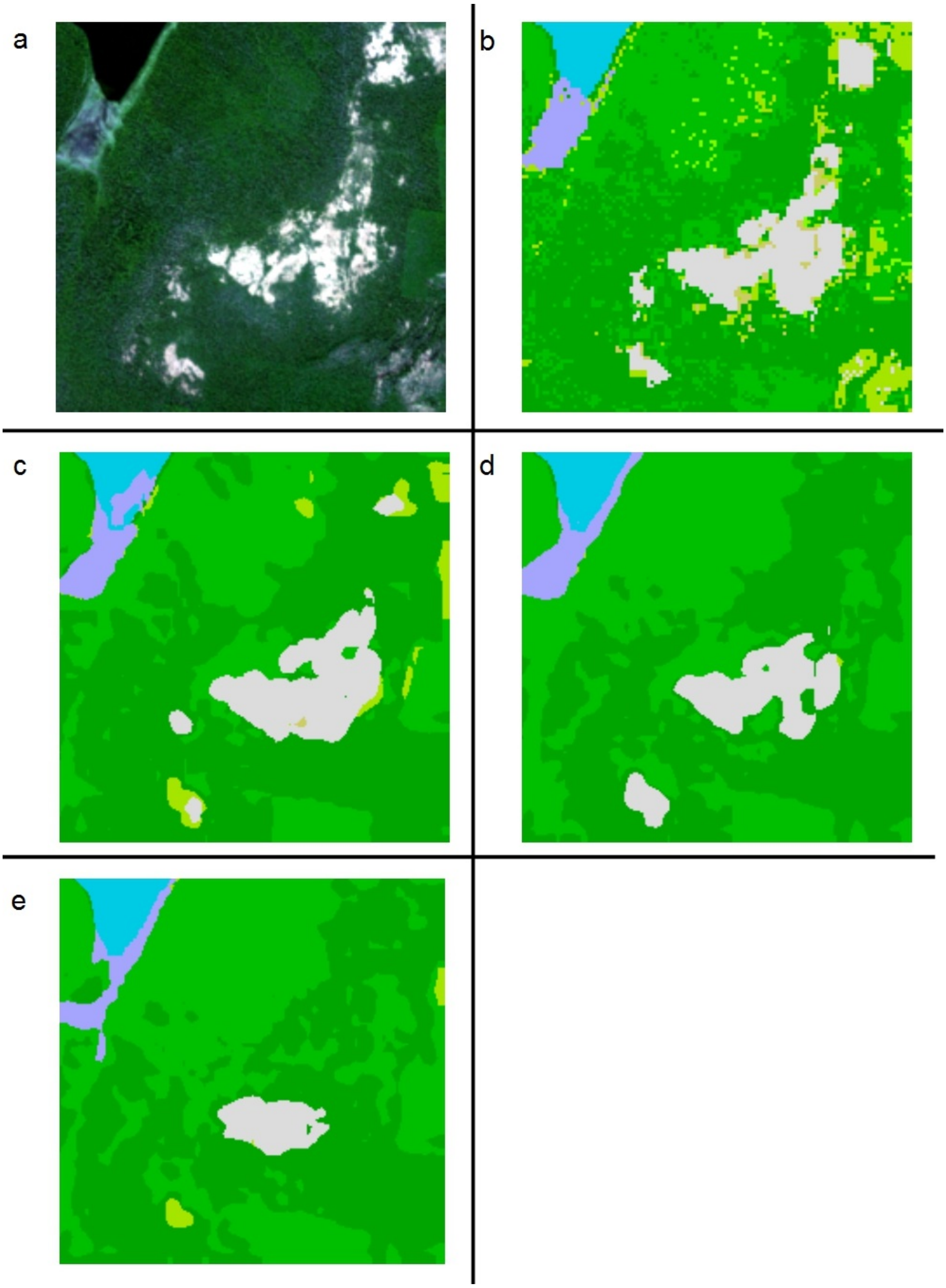


Figure 22. Sample output of model from Sentinel-2 test image. *a:* Sentinel-2 image. *b:* GT label from modified CLC. *c:* Output of ResNet-50 based model. *d:* Output of ResNet-34 based model. *e:* Output of ResNet-18 based model.

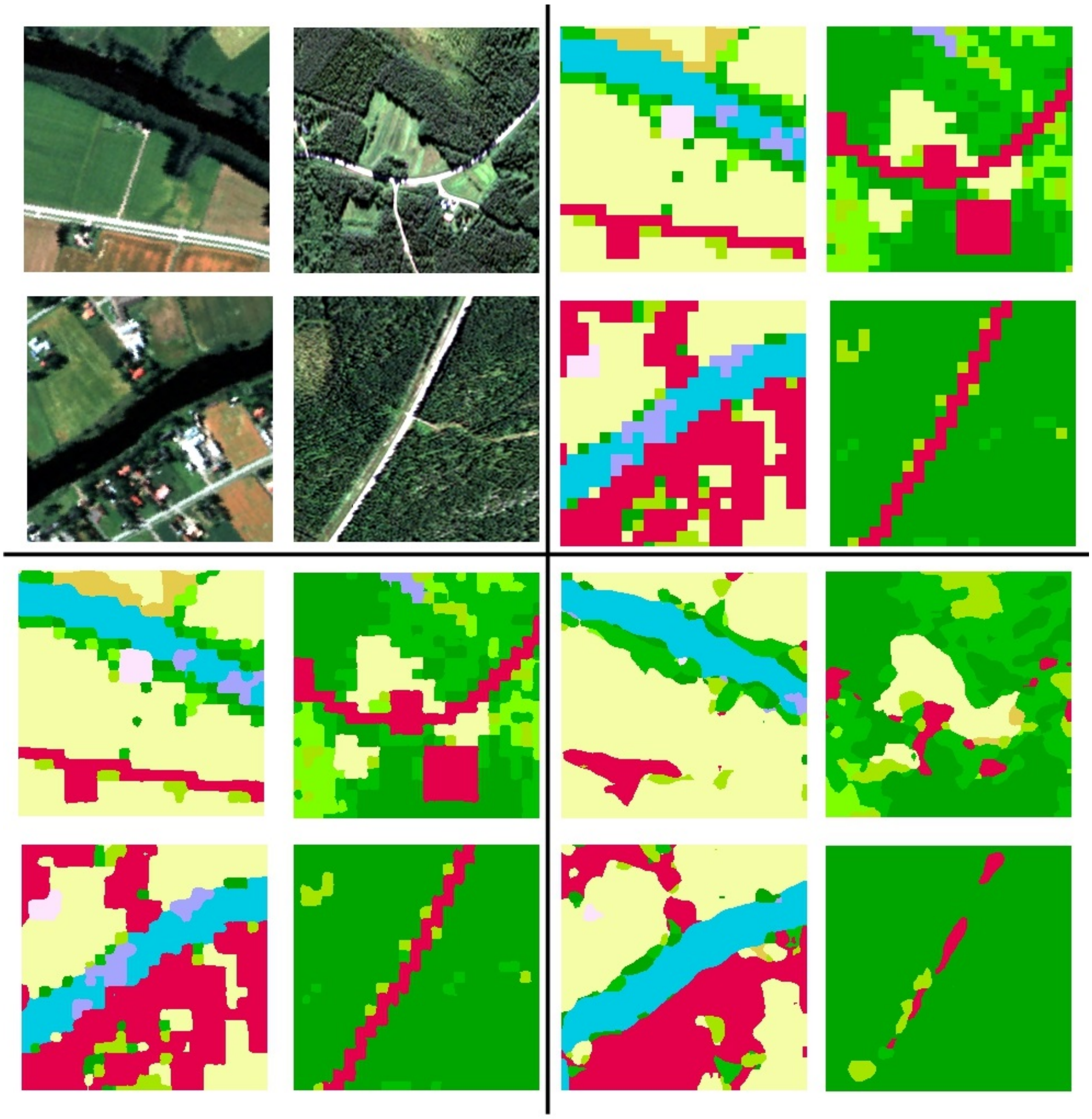


Figure 23. Output of model based on ResNet-50. *Top left:* Satellite images from Pleiades-1. *Top right:* GT labels from modified CLC. *Bottom left:* Output of model with transfer learning. *Bottom right:* Output of model without transfer learning.

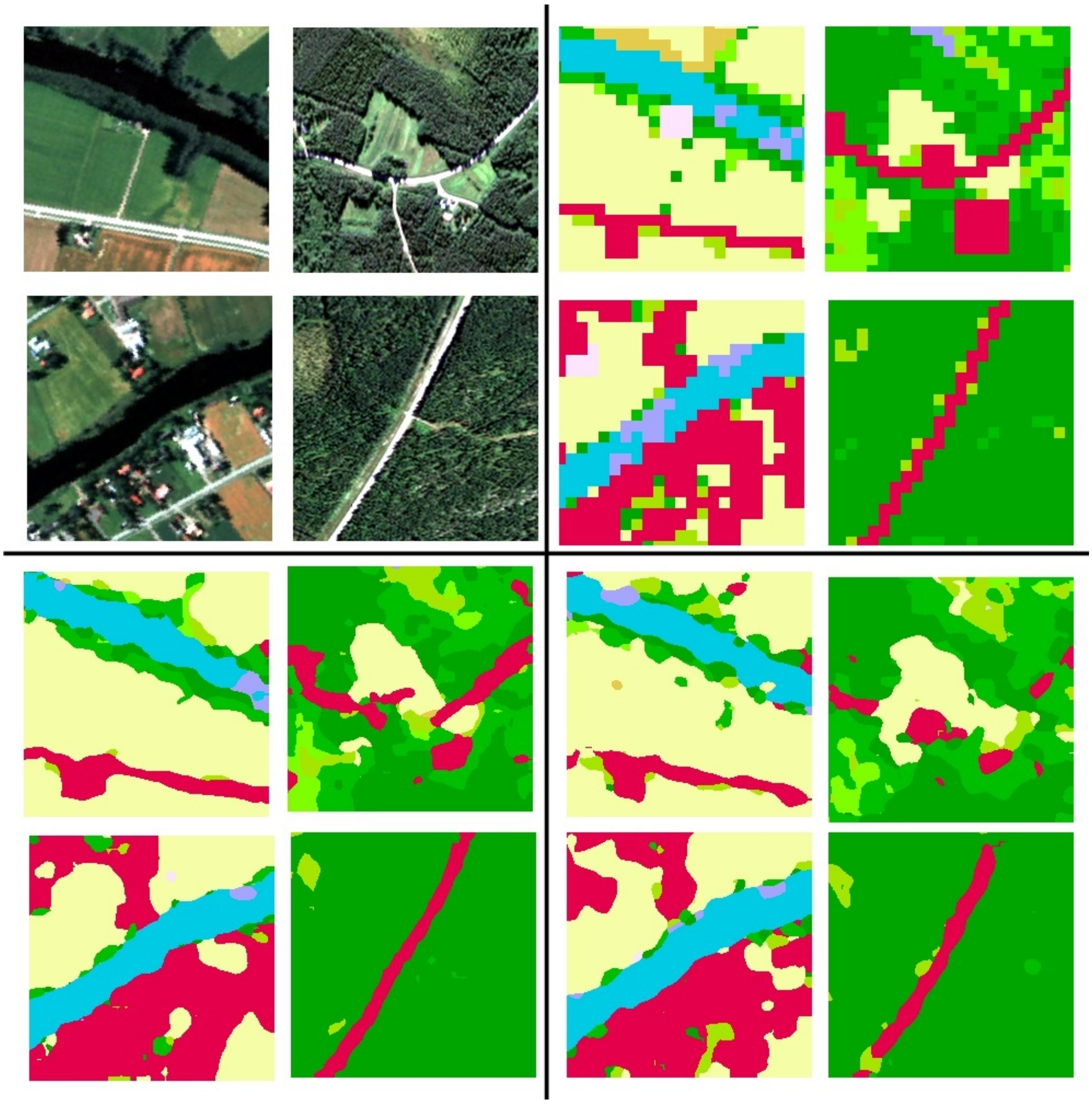


Figure 24. Output of model based on ResNet-34. *Top left:* Satellite images from Pleiades-1. *Top right:* GT labels from modified CLC. *Bottom left:* Output of model with transfer learning. *Bottom right:* Output of model without transfer learning.

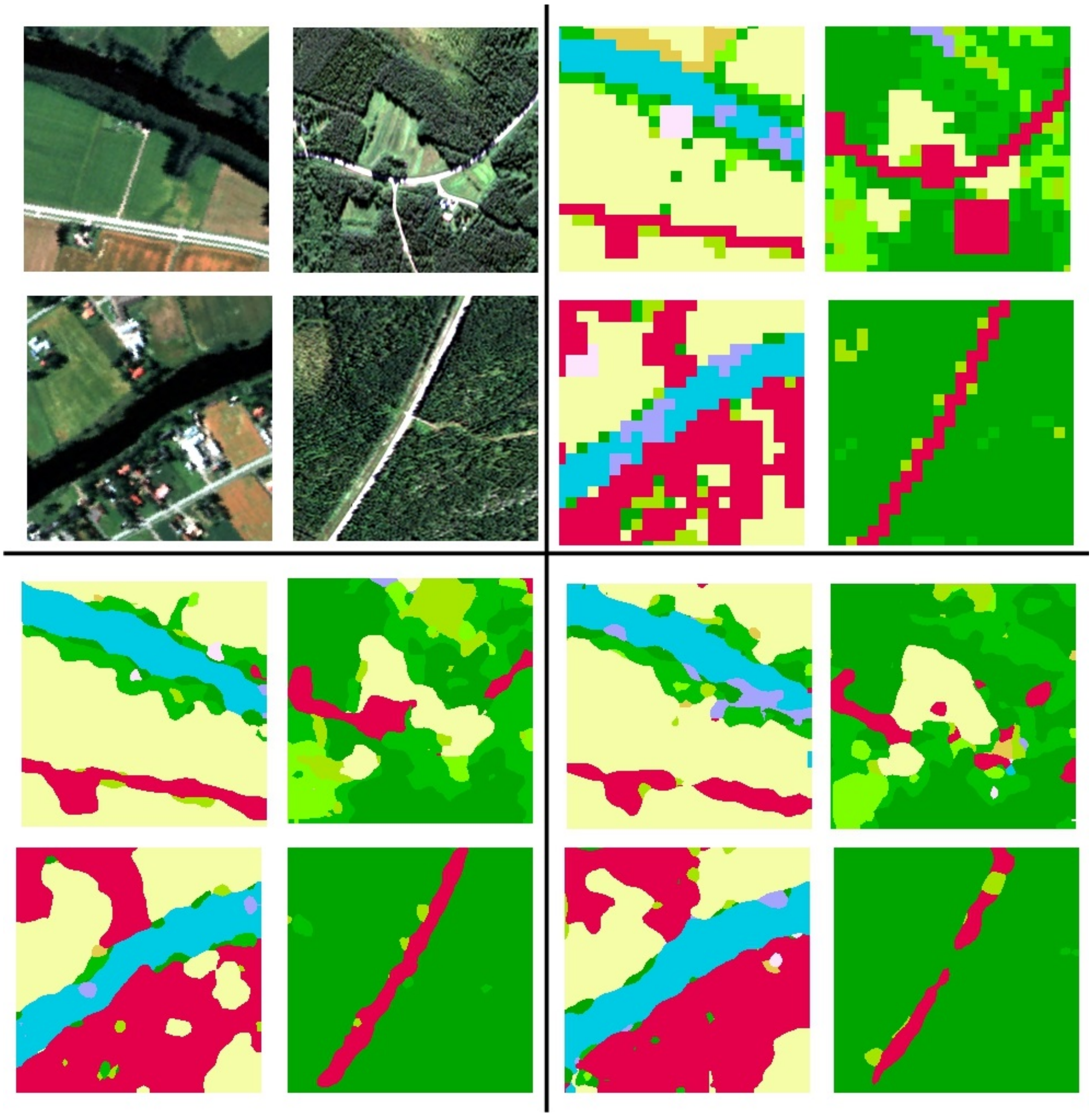


Figure 25. Output of model based on ResNet-18. *Top left:* Satellite images from Pleiades-1. *Top right:* GT labels from modified CLC. *Bottom left:* Output of model with transfer learning. *Bottom right:* Output of model without transfer learning.

6. DISCUSSION

In this thesis an encoder-decoder DNN approach based on Deeplab with transfer learning was followed to apply semantic segmentation on RS satellite images. The DNN takes advantage of the large amount of data available on one dataset to train on a different somewhat limited dataset to improve upon the results that would have been acquired without it.

6.1. Sentinel-2 results

The amount of data used to train the DNN was enough to train a network as deep as ResNet-50. The results achieved with the Sentinel-2 based dataset although not easily comparable with other studies are quite good. ResNet-50 achieved a MIoU of $0{,}53$ an and PA $0{,}78$. There is quite the jump in performance between ResNet-34 to ResNet-50 with ResNet-34 achieving a $0{,}30MIoU$ and $0{,}72PA$ so not so good all across the board. However this jump does not exist between ResNet-18 to ResNet-34, in fact it is quite the opposite since ResNet-18 reached a very similar score to ResNet-34 with $0{,}30MIoU$ and $0{,}72PA$ even though the difference in depth is the same. This is an issue that needs more experimentation. This could be attributed to the fact that there are quite a lot of classes for a land cover segmentation application compared to previous studies. The lack of 1×1 convolution in those shallower versions of ResNet might also be a limiting factor. It is clear that ResNet-18 and ResNet-34 are not favored compared to the other versions of ResNet, not performing better than VGG or GoogLeNet in many occasions. One more thing to point out is the difference in result between MIoU and PA. This is caused by the bias of PA in reporting how well the ANN identifies when the classes are not present rather than when they are which is more important.

6.2. Pleiades-1 results

Pleiades-1 data is better than Sentinel-2's when it comes to pixel resolution. However the Pleiades-1 data used in this experiment is quite limited even with a lot of augmentation which required the need for transfer learning. This is very much seen in the results with the ones gotten by transfer learning vastly outperforming the ones without it, for the most part. The ResNet-50's MIoU jumped from $0{,}33$ to an impressive $0{,}85$ showing the difference in results. This is even better than the results achieved by the Sentinel-2's data alone. A theme that repeats itself over all the ResNet backbones used although not to the same extent. As for the PA results, the pretrained model based on ResNet-50 reached a near perfect $0{,}98$. Although it is clear that accuracy is not very representative of the actual results in semantic segmentation it is still worth pointing out. The reason transfer learning worked so well in this case is the similarity of data between both datasets. Both Sentinel-2 and Pleiades-1 use RGB,and NIR channels with similar central wavelengths, in addition to being on roughly the same area covering parts of Finland. The improvement over Sentinel-2 data is due to the fact that Pleiades data has 10 times more pixel resolution.

6.3. Limitations

The main limitation for this research is the lack of GT data. CLC proved to be a very good resource for that purpose but it is still not good enough. The $20m/px$ resolution is still lower than both Sentinel-2's $10m/px$ resolution and more so Pleiades-1's $2m/px$. In addition the $20m/px$ is only exclusive to the Finnish area while the CLC provided by Copernicus for the whole of Europe is only $100m/px$. So even though Sentinel-2's data is available for free for the whole world it will not be very useful with that kind of label data. This could limit generalizing the network for Europe or even at a global scale to span the whole world.

The VHR data of satellites such as the Pleiades-1 is not available for free which limits the potential for it and it could cause wrong conclusions. Some of this data is available for free [81] however it is very limited and sometimes outdated.

The Geographic Information Systems (GIS) softwares are a huge contributor to RS data preprocessing and even processing so the lack of some functions or limitation in those can waste a lot of time and possibilities. QGIS is a free software that keeps getting better but still is quite limited and clumsy in handling the RS data. There exists ways to bypass using them by using the GDAL library in Python for example but those can also have their flaws. The softwares that are more stable such as ArcGIS are very costly which limits the potential of research in RS and by extension this very thesis.

6.4. Improvements

Some more tests need to be done to find the optimal configuration for this task. Trying to omit the 1×1 convolution is one example to see if they are contributing to the bottleneck in performance in both ResNet-18 and ResNet-34. To improve upon the results, firstly trying with deeper versions of ResNet such as the ResNet-101 and ResNet-152. This will clearly require more video memory which will cause the batch size to decrease. That would affect the batch normalization, which can be fixed with more epochs. Another possible improvement is raising the output stride to 8 instead of 16 which will also require more memory, much more in fact so finding a balance between batch size and output stride is a challenge. applying data reduction methods such as Principle Component Analysis (PCA) would contribute in reducing the need for video memory.

Of course having more diverse from areas all over the world would generalize the application to a global scale. Having more classes including various tree species would greatly improve the usage of the solution provided by this thesis. Therefore finding better GT data would improve the results quite well.

7. CONCLUSION

Deep Learning (DL) has vastly improved machine vision application. Therefore it is only natural to explore its advantages over semantic segmentation in a challenging domain such as remote sensing, specifically land and forest cover segmentation. Semantic segmentation in of itself is quite a broad and complex subject and adding to it the irregularity of objects in satellite images such as forest stands. This thesis presented an experimentation based on applying the state of the art DL methods in semantic segmentation. A deep neural network based of CNN, ResNets, ASPP, and encoder-decoder architecture was built to semantically segment images taken from the Sentinel-2 satellite based on 17 classes built from CLC. On top of having free data from the Sentinel-2 constellation of satellites, the availability of labels from CLC helped a lot in executing that task, reaching a very good MIoU score. This was useful with Pleiades-1 as satellite images as well, even though are not available for free for the most part. Thanks to transfer learning the model trained on Sentinel-2 data was able to be translated to Pleiades-1's much more limited data. The results with transfer learning on Pleiades-1 data outperformed the results without it. This is not surprising because the Pleiades dataset is quite limited to train a DNN. However it even managed to outperform the original network from Sentinel-2 by a large margin due to the higher resolution features that it bought to the model, thus making it more precise.

In conclusion DL proved to be quite useful in LULC segmentation, especially with a somewhat high number of classes. Deeper neural networks outperformed shallower versions showing the importance of DNNs in handling that task. Transfer learning is also a key method to follow in the case of scarcity of data where a large amount of data is required.

8. REFERENCES

[1] Chen L., Papandreou G., Kokkinos I., Murphy K. & Yuille A.L. (2018) Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Transactions on Pattern Analysis and Machine Intelligence 40, pp. 834–848.

[2] He K., Zhang X., Ren S. & Sun J. (2015) Deep residual learning for image recognition. CoRR abs/1512.03385. URL: `http://arxiv.org/abs/1512.03385`.

[3] Fischer W., Hemphill W. & Kover A. (1976) Progress in remote sensing (1972–1976). Photogrammetria 32, pp. 33 – 72. URL: `http://www.sciencedirect.com/science/article/pii/0031866376900132`.

[4] Lin S.K. (2013) Introduction to remote sensing. fifth edition.by james b. campbell and randolph h. wynne, the guilfordpress, 2011; 662 pages. price: £80.75, isbn 978-1-60918-176-5. Remote Sensing 5. URL: `http://www.mdpi.com/2072-4292/5/1/282`.

[5] Ucs satellite database. URL: `https://www.ucsusa.org/nuclear-weapons/space-weapons/satellite-database`.

[6] Sentinel-2 - missions - sentinel online. URL: `https://sentinel.esa.int/web/sentinel/missions/sentinel-2`.

[7] Open access hub. URL: `https://scihub.copernicus.eu/`.

[8] Pleiades - eoportal directory - satellite missions. URL: `https://directory.eoportal.org/web/eoportal/satellite-missions/p/pleiades`.

[9] Swinerd G. (2008) How Spacecraft Fly: Spaceflight Without Formulae, Astronomy, astrophysics, Springer New York. pp. 103–104. URL: `https://books.google.fi/books?id=FU0zWjX1CAUC`.

[10] Teles J., Samii M. & Doll C. (1995) Overview of tdrss. Advances in Space Research 16, pp. 67 – 76. URL: `http://www.sciencedirect.com/science/article/pii/027311779598783K`, orbit Determination and Analysis.

[11] space Agency E., Sentinel-2 - missions - resolution and swath - sentinel handbook. URL: `https://sentinel.esa.int/web/sentinel/missions/sentinel-2/instrument-payload/resolution-and-swath`.

[12] Hackeloeer A., Klasing K., Krisp J.M. & Meng L. (2014) Georeferencing: a review of methods and applications. Annals of GIS 20, pp. 61–69. URL: `https://doi.org/10.1080/19475683.2013.868826`.

[13] Aeronautics N. & Administration S.

[14] EORC J. (accessed 06.3.2019), Definition of processing levels | products amp; algorithms | data products | gcom-c@eorc. JAXA EORC. URL: https://suzaku.eorc.jaxa.jp/GCOM_C/data/product_def.html.

[15] Agency E.S. (accessed 06.3.2019), User guides - sentinel-2 msi - processing levels - sentinel online. ESA Sentinel Online. URL: https://sentinel.esa.int/web/sentinel/user-guides/sentinel-2-msi/processing-levels.

[16] University T.P.S. (accessed 20.3.2019), Exploring imagery and elevation data in gis applications. www.e-education.psu.edu. URL: https://www.e-education.psu.edu/geog480/node/497.

[17] Schmullius C., Earth system monitoring and modelling. URL: https://earth.esa.int/documents/973910/987578/cs2_schmullius.pdf.

[18] Barbara Kosztra György Büttner G.H.S.A.

[19] Yang Y. & Newsam S. (2010) Bag-of-visual-words and spatial extensions for land-use classification. In: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS '10, ACM, New York, NY, USA, pp. 270–279. URL: http://doi.acm.org/10.1145/1869790.1869829.

[20] Helber P., Bischke B., Dengel A. & Borth D. (2017) Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. CoRR abs/1709.00029. URL: http://arxiv.org/abs/1709.00029.

[21] Ma L., Li M., Ma X., Cheng L., Du P. & Liu Y. (2017) A review of supervised object-based land-cover image classification. ISPRS Journal of Photogrammetry and Remote Sensing 130, pp. 277 – 293. URL: http://www.sciencedirect.com/science/article/pii/S092427161630661X.

[22] Linda G. Shapiro G.C.S. (2001) Computer Vision, New Jersey, Prentice-Hall. p. 279–325.

[23] Lu Z., Chen D. & Xue D. (2018) Survey of weakly supervised semantic segmentation methods. pp. 1176–1180.

[24] Papandreou G., Chen L., Murphy K. & Yuille A.L. (2015) Weakly- and semi-supervised learning of a DCNN for semantic image segmentation. CoRR abs/1502.02734. URL: http://arxiv.org/abs/1502.02734.

[25] Chen L., Papandreou G., Kokkinos I., Murphy K. & Yuille A.L. (2016) Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. CoRR abs/1606.00915. URL: http://arxiv.org/abs/1606.00915.

[26] He K., Zhang X., Ren S. & Sun J. (2016) Deep residual learning for image recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[27] Chen L., Papandreou G., Schroff F. & Adam H. (2017) Rethinking atrous convolution for semantic image segmentation. CoRR abs/1706.05587. URL: `http://arxiv.org/abs/1706.05587`.

[28] Perumal K. & Bhaskaran R. (2010) Supervised classification performance of multispectral images. CoRR abs/1002.4046. URL: `http://arxiv.org/abs/1002.4046`.

[29] Costa H., Foody G.M. & Boyd D.S. (2018) Supervised methods of image segmentation accuracy assessment in land cover mapping. Remote Sensing of Environment 205, pp. 338 – 351. URL: `http://www.sciencedirect.com/science/article/pii/S0034425717305734`.

[30] Räsänen A., Rusanen A., Kuitunen M. & Lensu A. (2013) What makes segmentation good? a case study in boreal forest habitat mapping. International Journal of Remote Sensing 34, pp. 8603–8627. URL: `https://doi.org/10.1080/01431161.2013.845318`.

[31] Tehrany M.S., Pradhan B. & Jebuv M.N. (2014) A comparative assessment between object and pixel-based classification approaches for land use/land cover mapping using spot 5 imagery. Geocarto International 29, pp. 351–369. URL: `https://doi.org/10.1080/10106049.2013.768300`.

[32] Jiang Z., Huete A.R., Chen J., Chen Y., Li J., Yan G. & Zhang X. (2006) Analysis of ndvi and scaled difference vegetation index retrievals of vegetation fraction. Remote Sensing of Environment 101, pp. 366 – 378. URL: `http://www.sciencedirect.com/science/article/pii/S0034425706000290`.

[33] Khatami R., Mountrakis G. & Stehman S.V. (2016) A meta-analysis of remote sensing research on supervised pixel-based land-cover image classification processes: General guidelines for practitioners and future research. Remote Sensing of Environment 177, pp. 89 – 100. URL: `http://www.sciencedirect.com/science/article/pii/S0034425716300578`.

[34] Penatti O.A.B., Nogueira K. & dos Santos J.A. (2015) Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? In: 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 44–51.

[35] Demir I., Koperski K., Lindenbaum D., Pang G., Huang J., Basu S., Hughes F., Tuia D. & Raskar R. (2018) Deepglobe 2018: A challenge to parse the earth through satellite images. CoRR abs/1805.06561. URL: `http://arxiv.org/abs/1805.06561`.

[36] Tian C., Li C. & Shi J. (2018) Dense fusion classmate network for land cover classification. pp. 262–2624.

[37] Huang G., Liu Z. & Weinberger K.Q. (2016) Densely connected convolutional networks. CoRR abs/1608.06993. URL: `http://arxiv.org/abs/1608.06993`.

[38] Kuo T.S., Tseng K.S., Yan J., Liu Y.C. & Wang Y.C.F. (2018) Deep aggregation net for land cover classification. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) , pp. 247–2474.

[39] Arief H.A., Strand G.H., Tveite H. & Indahl U.G. (2018) Land cover segmentation of airborne lidar data using stochastic atrous network. Remote Sensing 10. URL: `http://www.mdpi.com/2072-4292/10/6/973`.

[40] Beech E., Rivers M., Oldfield S. & Smith P.P. (2017) Globaltreesearch: The first complete global database of tree species and country distributions. Journal of Sustainable Forestry 36, pp. 454–489. URL: `https://doi.org/10.1080/10549811.2017.1310049`.

[41] Institute E.F. (accessed 11.3.2019.), Tree species maps for european forests. URL: `https://www.efi.int/knowledge/maps/treespecies`.

[42] Tucker C., Townshend J. & E. Goff T. (1985) African land-cover classification using satellite data. Science (New York, N.Y.) 227, pp. 369–75.

[43] Lung T. & Schaab G. (2010) A comparative assessment of land cover dynamics of three protected forest areas in tropical eastern africa. Environmental Monitoring and Assessment 161, pp. 531–548. URL: `https://doi.org/10.1007/s10661-009-0766-3`.

[44] Waser L., Ginzler C., Kuechler M., Baltsavias E. & Hurni L. (2011) Semi-automatic classification of tree species in different forest ecosystems by spectral and geometric variables derived from airborne digital sensor (ads40) and rc30 data. Remote Sensing of Environment 115, pp. 76 – 85. URL: `http://www.sciencedirect.com/science/article/pii/S0034425710002464`.

[45] Hyyppä J., Hyyppä H., Leckie D., Gougeon F., Yu X. & Maltamo M. (2008) Review of methods of small-footprint airborne laser scanning for extracting forest inventory data in boreal forests. International Journal of Remote Sensing 29, pp. 1339–1366.

[46] Maltamo M., Næsset E. & Vauhkonen J. (2014) Forestry applications of airborne laser scanning. Concepts and case studies. Manag For Ecosys 27, p. 460.

[47] Féret J.B. & Asner G.P. (2013) Tree species discrimination in tropical forests using airborne imaging spectroscopy. IEEE Transactions on Geoscience and Remote Sensing 51, pp. 73–84.

[48] Solberg S., Naesset E. & Bollandsas O.M. (2006) Single tree segmentation using airborne laser scanner data in a structurally heterogeneous spruce forest. Photogrammetric Engineering & Remote Sensing 72, pp. 1369–1378.

[49] Johnson B., Forest type mapping data set. URL: http://archive.ics.uci.edu/ml/datasets/Forest+type+mapping.

[50] Clearinghouse U.F.S.F., National forest type dataset. URL: https://data.fs.usda.gov/geodata/rastergateway/forest_type/.

[51] Sabanci K. & Polat K. (2016) Classification of different forest types with machine learning algorithms.

[52] Pasquarella V.J., Holden C.E. & Woodcock C.E. (2018) Improved mapping of forest type using spectral-temporal landsat features. Remote Sensing of Environment 210, pp. 193 – 207. URL: http://www.sciencedirect.com/science/article/pii/S0034425718300762.

[53] Hochreiter S. (1998) The vanishing gradient problem during learning recurrent neural nets and problem solutions. Int. J. Uncertain. Fuzziness Knowl.-Based Syst. 6, pp. 107–116. URL: http://dx.doi.org/10.1142/S0218488598000094.

[54] LeCun Y., Boser B., Denker J.S., Henderson D., Howard R.E., Hubbard W. & Jackel L.D. (1989) Backpropagation applied to handwritten zip code recognition. Neural Comput. 1, pp. 541–551. URL: http://dx.doi.org/10.1162/neco.1989.1.4.541.

[55] Krizhevsky A., Sutskever I. & E. Hinton G. (2012) Imagenet classification with deep convolutional neural networks. Neural Information Processing Systems 25.

[56] Glorot X., Bordes A. & Bengio Y. (2011) Deep sparse rectifier neural networks. In: G. Gordon, D. Dunson & M. Dudík (eds.) Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, Proceedings of Machine Learning Research, vol. 15, PMLR, Fort Lauderdale, FL, USA, Proceedings of Machine Learning Research, vol. 15, pp. 315–323. URL: http://proceedings.mlr.press/v15/glorot11a.html.

[57] Ramachandran P., Zoph B. & Le Q.V. (2017) Searching for activation functions. CoRR abs/1710.05941. URL: http://arxiv.org/abs/1710.05941.

[58] Ioffe S. & Szegedy C. (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. CoRR abs/1502.03167. URL: http://arxiv.org/abs/1502.03167.

[59] Shi W., Caballero J., Theis L., Huszar F., Aitken A.P., Ledig C. & Wang Z. (2016) Is the deconvolution layer the same as a convolutional layer? CoRR abs/1609.07009. URL: http://arxiv.org/abs/1609.07009.

[60] Garcia-Garcia A., Orts-Escolano S., Oprea S., Villena-Martinez V. & Rodríguez J.G. (2017) A review on deep learning techniques applied to semantic segmentation. CoRR abs/1704.06857. URL: http://arxiv.org/abs/1704.06857.

[61] Gupta S., Girshick R.B., Arbelaez P. & Malik J. (2014) Learning rich features from RGB-D images for object detection and segmentation. CoRR abs/1407.5736. URL: `http://arxiv.org/abs/1407.5736`.

[62] Gidaris S. & Komodakis N. (2015) Object detection via a multi-region and semantic segmentation-aware cnn model. In: The IEEE International Conference on Computer Vision (ICCV).

[63] Noh H., Hong S. & Han B. (2015) Learning deconvolution network for semantic segmentation. In: The IEEE International Conference on Computer Vision (ICCV).

[64] Hazirbas C., Ma L., Domokos C. & Cremers D. (2017) Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture. In: S.H. Lai, V. Lepetit, K. Nishino & Y. Sato (eds.) Computer Vision – ACCV 2016, Springer International Publishing, Cham, pp. 213–228.

[65] Linmans J., Winkens J., Veeling B.S., Cohen T.S. & Welling M. (2018) Sample efficient semantic segmentation using rotation equivariant convolutional networks. CoRR abs/1807.00583. URL: `http://arxiv.org/abs/1807.00583`.

[66] Noh H., Hong S. & Han B. (2015) Learning deconvolution network for semantic segmentation. CoRR abs/1505.04366. URL: `http://arxiv.org/abs/1505.04366`.

[67] Simonyan K. & Zisserman A. (2014) Very deep convolutional networks for large-scale image recognition. CoRR abs/1409.1556.

[68] Badrinarayanan V., Kendall A. & Cipolla R. (2015) Segnet: A deep convolutional encoder-decoder architecture for image segmentation. CoRR abs/1511.00561. URL: `http://arxiv.org/abs/1511.00561`.

[69] Ronneberger O., Fischer P. & Brox T. (2015) U-net: Convolutional networks for biomedical image segmentation. CoRR abs/1505.04597. URL: `http://arxiv.org/abs/1505.04597`.

[70] Jiang J., Zheng L., Luo F. & Zhang Z. (2018) Rednet: Residual encoder-decoder network for indoor RGB-D semantic segmentation. CoRR abs/1806.01054. URL: `http://arxiv.org/abs/1806.01054`.

[71] Jégou S., Drozdzal M., Vázquez D., Romero A. & Bengio Y. (2016) The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. CoRR abs/1611.09326. URL: `http://arxiv.org/abs/1611.09326`.

[72] Wu G., Shao X., Guo Z., Chen Q., Yuan W., Shi X., Xu Y. & Shibasaki R. (2018) Automatic building segmentation of aerial imagery using multi-constraint fully convolutional networks. Remote Sensing 10, p. 407.

[73] Ghosh A., Ehrlich M., Shah S., Davis L. & Chellappa R. (2018) Stacked u-nets for ground material segmentation in remote sensing imagery. pp. 252–2524.

[74] Rakhlin A., Davydow A. & Nikolenko S. (2018) Land cover classification from satellite imagery with u-net and lovasz-softmax loss. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.

[75] Maggiori E., Tarabalka Y., Charpiat G. & Alliez P. (2017) Convolutional neural networks for large-scale remote-sensing image classification. IEEE Transactions on Geoscience and Remote Sensing 55, pp. 645–657.

[76] Clc 2018 — copernicus land monitoring service. URL: https://land.copernicus.eu/pan-european/corine-land-cover/clc2018.

[77] Corine land cover 2018. URL: http://metatieto.ymparisto.fi:8080/geoportal/catalog/search/resource/details.page?uuid=\%7B26EEEBBB-FB5C-4045-B6DF-439F9B7D5C46\%7D.

[78] TensorFlow (accessed 11.4.2019.), Importing data. URL: https://www.tensorflow.org/guide/dataset.

[79] Nvidia tesla v100 data center gpu. URL: https://www.nvidia.com/en-us/data-center/tesla-v100/.

[80] Qian N. (1999) On the momentum term in gradient descent learning algorithms. Neural Networks 12, pp. 145 – 151. URL: http://www.sciencedirect.com/science/article/pii/S0893608098001166.

[81] Agency E.S. (accessed 11.4.2019.), Esa 3rd party missions overview - earth online - esa. URL: https://earth.esa.int/web/guest/missions/3rd-party-missions/overview.