Høgskulen
på Vestlandet

UNIVERSITAS BERGENSIS

Machine Learning based Detection and Identification of Trees using High Resolution Satellite Images

Sindre Aalhus

Supervisor: Prof. Reza Arghandeh

Master's Thesis in Software Development Western Norway

University of Applied Sciences

May, 2021

Acknowledgement

I would like to acknowledge everyone who played a role in this thesis. Without the great help from the very talented members of the Ci2Lab, including my supervisor Prof. Reza Arghandeh, my colleagues (Atle Følid, Mojtaba Yousefi and especially Michele Gazzea), and of course my family, this thesis and my Master's Degree would not have been possible. This project is supported partly by the European Space Agency.

Abstract

Vegetation detection and species identification around infrastructure networks such as power lines, roadways, and pipelines are essential for reducing the risk of damage imposed by vegetation and the safety concerns for wildfires. The primary approach for monitoring vegetation around infrastructure networks is sending ground-based crew and flying helicopters and drones. Such methods are laborious, time-consuming, and cost-inefficient. With the rise of satellite imagery services and artificial intelligence, this thesis proposes utilizing satellite images to create a more accessible, faster, and more cost-efficient way of monitoring vegetation. This thesis develops an automated approach using multi-spectral images, tree location data, and forest inventory to locate and identify species of trees and forests. The proposed method is validated using actual data for an area in the western part of Norway. The outcomes of this study can be utilized to lower the cost of vegetation monitoring and consequently lower the environmental impact of vegetation management plans.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This section will give an understanding of the problems this thesis aims to solve, and the reason for their importance. It will provide a short overview of the project, the motivation for this research, how it is conducted, and its structure.

1.1 Objective and Motivation

Vegetation growth can have massive impacts on infrastructures networks. Trees can interfere with power lines and cause outages, or they can fall over roads and cut off transportation. The impact of bad vegetation management can lead to great economical losses, and cause several inconveniences for companies and locals of any community. This thesis looks into the possibility to use any available data to automate the process of monitoring vegetation, and tries to manufacture a smart and accurate solution to locate and identify a set of tree species using remote sensing technology. We believe remote sensing combined with satellite imagery, vegetation measurements, and forest inventories, has the innate capability to both locate trees, and identify their species.

1.2 Research Questions

This thesis tries to go in depth of the potency of using remote sensing techniques to make vegetation monitoring more time-, and cost-efficient. These are the research questions that need to be answered in order to formulate a concrete conclusion.

Research Question I

How can today's available satellite image data be effectively utilized to monitor vegetation?

Research Question II

What are some of the most effective satellite image analysis techniques when applied to vegetation monitoring?

1.3 The GridEyeS Project

This master thesis is done in cooperation with an ESA Business Applications project, namely GridEyes. The goal of this project is to create a platform for monitoring power grids using satellites combined with other sources to provide several tools for risk assessment, infrastructure management, and pre-event resilience assessment. The project is an effort to automate the gathering and analysis of information on specific geographical areas, train machine learning models with data from these areas, and provide these outcomes through a user interface to end user companies. (Fig.

1.1) In this way communities can have better insight in the status and integrity of their infrastructure and how to further manage it.

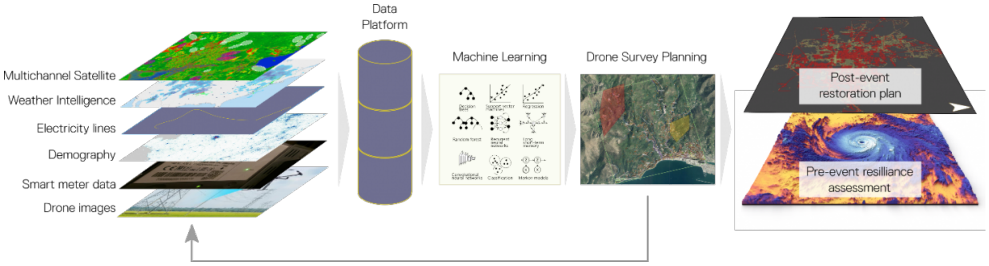


Figure 1.1: This image shows the proposed architecture for the GridEyeS resilience assessment approach. ([1])

The main focus of this thesis is the machine learning aspects on the respective task; to implement deep learning methods and algorithms in order to automatically estimate location and species of vegetation, in hopes that this can help determine the risk they pose. This research will only consider the vegetation aspect of this encompassing project. Resulting methodologies of using different machine learning technologies should provide some knowledge on how to utilize remote sensing and local data in order to monitor vegetation effectively. For practical consideration this methodology might help strengthen the resilience and security of our society, and improve the ability to handle unexpected events.

Some of the related studies to GridEyeS by our research group, the Connectivity, Information Intelligence Lab (www.ci2Lab.com), are available in ”Automated Satellite-based Assessment of Hurricane Impacts on Roadways” [2], ”Developing city-wide hurricane impact maps using real-life data on infrastructure, vegetation and weather” [3], ”Automated Power Lines Vegetation Monitoring using High-Resolution Satellite Imagery” [4], ”Post-Hurricanes Roadway Closure Detection using Satellite Imagery and Semi-Supervised Ensemble Learning” [5], and ”Leveraging Remote Sensing Indices for Hurricane-induced Vegetative Debris Assessment: A GIS-based

Case Study for Hurricane Michael". [6]

1.4 Design Science Research

This thesis follows a design science research approach with a small set of artifacts. Design science should provide a nominal process for conducting the research, keep the results consistent with prior findings, and streamlined towards a clear objective. [7] Within the definitions and constraints of design science research, 5 activities are defined to help guide the research process.

- *Identification of Problem:* Current solutions for monitoring infrastructure networks and the vegetation around it is slow and economically costly. Further explanation is found in chapter 2 and 3.
- *Objective Definition:* Define an automated approach to monitor vegetation with available data.
- *Design:* Develop a set of remote sensing models trained to solve the identified problem.
- *Demonstration:* Demonstrate the developed models on an area of interest.
- *Evaluation:* Evaluate the models and their efficacy with well defined metrics.

Every item in this list will be described in depth during the thesis, design is defined in methodology, demonstration and evaluation of the artifacts are found in the results and discussion chapter.

1.5 Thesis Structure

A summary of the thesis structure.

Chapter 1 - Introduction:

Introduces the circumstances of the research, its objectives, motivation, and questions.

Chapter 2 - Background:

Describes the context of the research and the fundamental technologies used to accomplish it.

Chapter 3 - Use Case:

Provides information on the problem, the use case, and the data.

Chapter 4 - Methodology:

In depth explanation of the methods developed in this thesis.

Chapter 5 - Results and Discussion:

Empirical results and analysis.

Chapter 7 - Conclusion:

Conclusion of the work, its contribution and any future work.

Chapter 2

Background

This section should establish the context of the research, briefly explain what approaches already exist, which ones are to be used in order to close in on research goals, as well as justify the need to put forward any findings in the field.

2.1 Overview of Vegetation Monitoring

There are several domains where vegetation surveillance is becoming an increasingly more important task. For example in power grid management, vegetation is a big risk factor in system malfunctions and outages. Errors in managing such infrastructure can leave entire cities with unplanned outages for unknown amounts of time, and occasionally even cause wildfires [8]. According to "Risk Analysis for Assessment of Vegetation Impact on Outages in Electric Power Systems" [9] the most prominent cause for malfunctions in power grids are surrounding vegetation impacting the power lines. Vegetation around towers and along lines can interfere with the constructions by being blown over by wind, degenerate due to old age, or the lines can sag due to heavy precipitation during winter. These factors make it abundantly clear that vegetation monitoring is a fundamentally important service as the consequences of climate change, and the higher frequency of extreme weather events are more present than ever [10]. Other researchers aim to improve a society's recovery time after an extreme event like hurricanes by providing local responders with information on fallen trees, loose debris, or toppled powerlines. [2] [6] Information such as this can help alleviate several problems in the aftermath of an extreme weather event. [3] Recover roadways, locate broken equipment, and generally help fully open and restore the infrastructure.

There are ways of monitoring vegetation growth, but most of them are very expensive and usually not fully automated. Nowadays, most electrical power companies utilize manned surveys of the grid system. This means paid employees visually inspect the grid by foot or by vehicle. Some provide aerial imagery for inspection, but there seems to be a lack of automated surveying methods for this domain. Several attempts have been made to produce such processes such as using satellite or aerial images [11, 12], Light Detection And Ranging data (LiDAR) [13].

Many of the scientific approaches to solve the problem of detecting and analysing trees from satellite images suffer from being dependent on image quality. Pointed out in "3D reconstruction from very high resolution satellite stereo and its application to object identification" [14] a wide range of features can be used to compliment the drawbacks from satellite images, but satellite images are considered not sufficient by itself. This thesis looks into correlation that can be found between high resolution images and aerial LiDAR data as well as tree species data to expand on any resourceful features beyond just satellite. If there exists sufficient correlation between satellite images and species and/or LiDAR, a resulting machine learning algorithm can have the potential to turn a high resolution image into a medium that can precisely locate and recognize tree species.

A study from The Norwegian Institute of Bioeconomy (NIBIO) found that specific species of vegetation have a higher risk of causing damage to a power grid, and their solution included species identification of individual trees to assess the risk they posed to infrastructural systems. [15] With exact forest inventory data an algorithm can automatically classify trees by species from any pan-chromatically captured area (where satellite images are available), and can be used to analyze how likely they are to intervene with structures in the future. This fact can play a big role in GridEyeS's goal for pre-event resilience assessment, and can provide crucial information.

As proposed in ”Use of satellite imagery for DEM extraction, landscape modeling and GIS application” [16] stereoscopic images can be used to extract digital elevation models (DEMs). Using two slightly offset cameras to photograph the same location provides enough context to generate a 3-dimensional representation of the landscape, however these resources are costly and only a subset of satellites provide this service. Given the growing number of satellites in orbit equipped with high resolution cameras, quality satellite data for any area around the globe have become increasingly more available to private actors. [17] Generally using satellite imagery to detect vegetation is a well-studied topic, but most of these studies are done in small scale areas where vegetation are easily distinguished, i.e. sparsely spread. [18] Nature does not necessarily behave so in a larger scale, especially not in particularly forested areas like most of Norway.

Previous studies mentioned in this section are almost entirely parts of prolonged processes, and very few of them can perform on demand. Taking LiDAR surveys as an example, even though it is a popular approach to monitor power grids, is an expensive process and usually only performed once per 5-10 years.[19] Introducing machine learning algorithms to perform on data that is easily updated and less prone to being outdated for long periods of time will provide an effective way to monitor vegetation.

Reviewing the data and the previous works surrounding this domain, it seems clear that it is lacking a cheap automated approach to accurately locate trees/forests and identify their species. With all this data available to us, machine learning, or more specifically remote sensing techniques may provide the functionality needed to comply with these demands.

The findings of other works that have attempted remote sensing on vegetation and vegetation species have found that the use of LiDAR data and multi-spectral satellite images to train a machine learning algorithms for semantic segmentation have proved feasible. [20] Most works in remote sensing revolve around using satellite with some ground truth labels, and many of these point in the direction of convolutional neural networks being the most prominent technique. [21]

2.2 Overview of Machine Learning Techniques

Machine learning is an area of information science which concerns automatically evolving algorithms that through weighted statistical equations may learn how to perform a task. In short it is the ability to teach a computer how to expertly perform a very specific task, given very specific input. This can range from learning the statistics of the housing market in order to predict sales prices (i.e. regression), recognizing what objects an image contains (i.e classification), and so on. An instance of such a trained algorithm is called a model. Machine learning (ML) plays a vital role in the work of automating numerous mundane processes, but also carries great potential to solve new and challenging problems.

Deep learning refers to the area of machine learning which makes use of neural networks/nets. A Neural net is a densely interconnected net of processor nodes, also called neurons. This archetype of learning has been in and out of fashion since the mid 1940s, but due to the latest improvements and affordability of computing power, neural nets have undergone a resurgence. Neural networks were first proposed by Warren McCullough and Walter Pitts in 1944. Their work were heavily influenced by neuro-science and how neural nets are loosely modeled after the inner workings of the brain, hence the choice of terminology. [22]

As mentioned, a neural network is built up by thousands, sometimes even millions of connected neurons structured in layers. Neurons are nodes that have weighted connections to neurons of the next layer in the network. These weights determine whether the neuron feeds the value forward, multiplying it with its weighted connection, or it stops the value indicating no related pattern. During training these real number weights are adjusted according to a loss function as they learn to make the right prediction. All these connections and neurons work in parallel to recognize patterns along the network. Exactly how an instance of a network does this recognition is considered a black box problem, or not easily analyzed and understood, but the logical assumption is that shallow layers will recognize basic patterns, and deeper layers recognize more complex, domain-specific patterns. Deep learning architectures are capable of learning very intricate patterns in data, and with neurons and connections being the only building blocks, a neural net is scalable for almost all use cases. Such deep learning structures are exceptional at recognizing visual patterns and have in many cases been used to analyse and extract information out of satellite images.

There are a multitude of ways to build and structure such neural nets. This thesis will only scrape the surface of neural net architecture, but there are some distinguished types. Firstly there are Fully Connected Neural Networks (FCNN) where every neuron of each layer is connected to every neuron of the next layer. (Fig. 2.1) This makes the architecture structure agnostic by nature, meaning no assumptions need to be made about the input beforehand, making it applicable to nearly all tasks, but it is worth mentioning that FCNNs will almost always fall short when compared to special-case networks designed and tuned to a specific task.

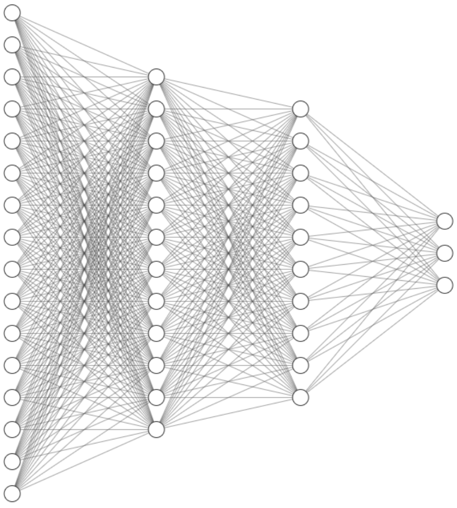


Figure 2.1: Example structure of a Fully Connected Neural Network

Another common type of neural networks for image processing are Convolutional Neural Networks (CNN). These are mainly used for semantic segmentation or object detection tasks on images. [23] [24] As this thesis is dealing with a segmentation problem, CNNs will be the main actor in the thesis methodology, as they are designed to explicitly take images, or image-like data as input, and are popular methods for satellite image analysis. These are built by mainly 3 pieces; convolutional layers, pooling layers, upsampling layers, and sometimes a fully connected layer for output. (see Fig. 2.2)

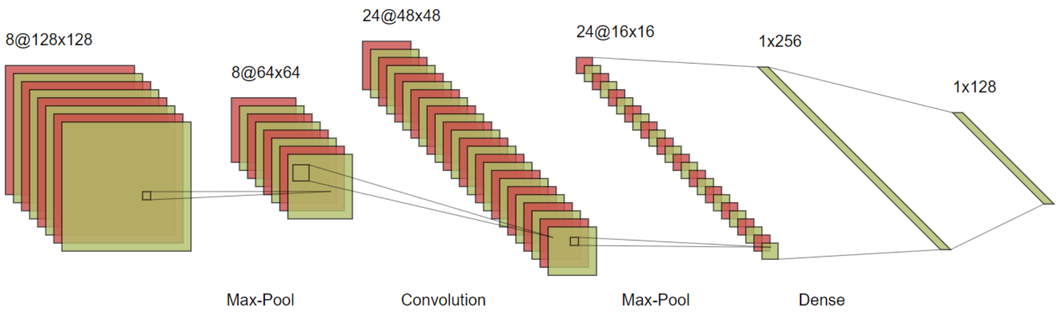


Figure 2.2: Example of a Convolutional Neural Network

Convolutional layers apply a filter to an input to conjure a feature map of any features detected in the input.[25] The filters can be static, but in most cases they are ”learned” and changes during training such that they obtain the ability to detect patterns. For image segmentation the filters are a two dimensional array fitted with weights. (see Fig. 2.3) When inputted an image the convolution layer will iterate the array over the pixels in the image, skip a number of pixels per iteration (also referred to as stride), multiply their value by the layer weights before adding all values together, then output it as another feature map. Several of these convolution layers in tandem will after enough training epochs be able the recognize features in the data. [24]

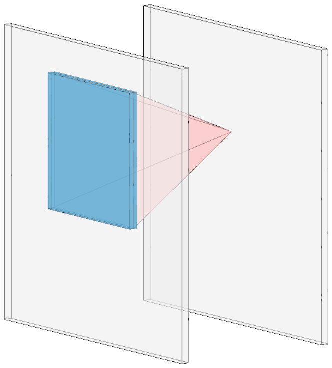
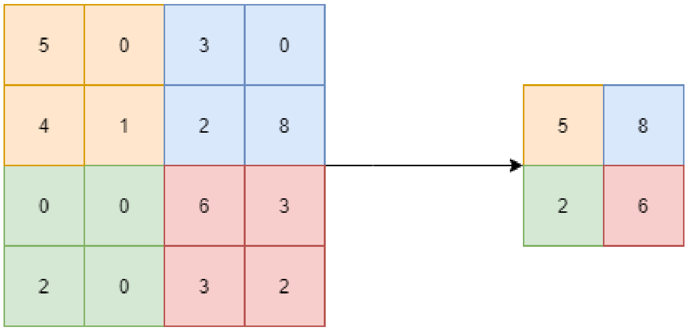


Figure 2.3: A visualization of the convolution process. The grey slates are feature maps, and the blue slate represents the weighted filter.

Maxpooling is another building block of convolutional networks. It's purpose is to accumulate features of the feature map into a smaller and more compact feature space, (see Fig. 2.4) thus its name downsampling. The process takes an image or a feature map from a previous layer, iterates over the map and returns the maximum value of traditionally non-overlapping areas. This reduces the spatial resolution of the map, but retains the most important information. [26]



5	0	3	0
4	1	2	8
0	0	6	3
2	0	3	2

5	8
2	6

Figure 2.4: A 2x2 max pool process keeping the largest value and the general area it was found.

Deconvolution, upsampling, or backwards strided convolution are some of the names for the final fundamental piece of a convolutional neural network. The motivation for this layer is to accurately recreate a higher resolution feature map when inputted a downsampled feature map. Upsampling can be viewed as an inverse of downsampling as it tries to recreate the feature map with bigger dimensions rather than smaller. Some techniques in convolutional segmentation networks uses a memory of maxpooling locations to further improve the upsampling (also called un-pooling, see Fig. 2.5).[27] In some networks the upsampling layers are exchanged for transposed convolution layers. These layers are trainable layers much like the traditional convolution, but will also upsample the input in terms of feature map dimensions.

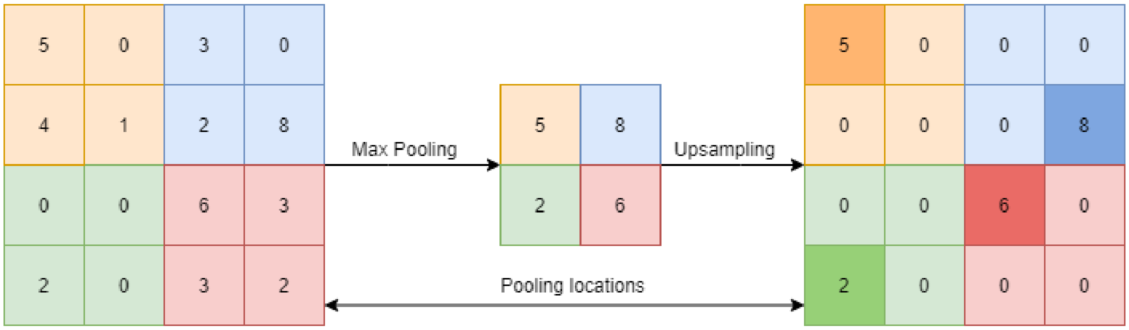
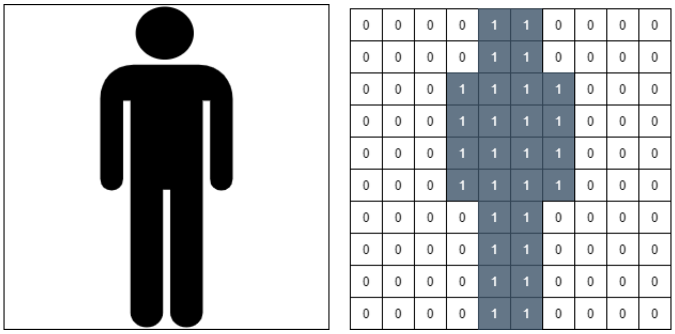


Figure 2.5: An example visualization of an un-pooling process where the location of the maxpooling is used to improve upsampling.

Semantic Segmentation is a field of machine learning which concerns locating objects in an image, and classifying what object that is. Segmentation algorithms generate an output commonly referred to as a ”mask”. These masks provide pixel specific classification of entire images based on target labels and classes the algorithm was provided during training (see Fig. 2.6). Segmentation masks can be very useful to quickly analyse and describe images, and can be a powerful tool when used in practical areas like geographical mapping or analysis, as they can learn the distribution and patterns in any objects, and cover large images/areas if a proper dataset is provided. The approaches in this thesis will be heavily focused on semantic segmentation techniques, specifically convolutional neural network segmentation algorithms.



0	0	0	0	1	1	0	0	0	0
0	0	0	0	1	1	0	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	0	1	1	0	0	0	0
0	0	0	0	1	1	0	0	0	0
0	0	0	0	1	1	0	0	0	0
0	0	0	0	1	1	0	0	0	0

Figure 2.6: An example of how a generated mask corresponding to an image of a "man sign/symbol" would look, given 0 = Background, and 1 = Man.

An activation function is a simple mathematical function that defines how the weighted sum of an input is transformed into the output of a node in a neural network. They tend to vary in functionality and complexity, but for this methodology Rectified Linear Units (ReLU, see Fig. 2.7) are used as they are the common and often the most effective choice for convolutional networks. [28]

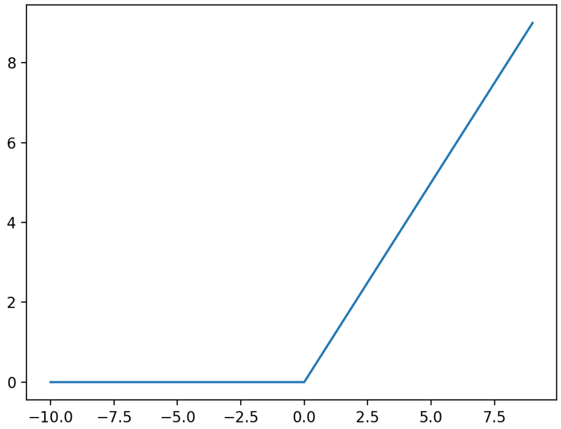


Figure 2.7: Rectified Linear Unit Activation Function displayed as a graph.

Activation functions for output layers are different in the sense that they produce a prediction instead of an input to the next node, and the common choice in convolutional networks is often a softmax activation.

An optimizer is an algorithm designed to continuously make changes to model weights during training to reach the lowest loss and the highest accuracy. There exists an abundance of optimizer algorithms for different use cases. One example is Adaptive Moment Estimation (Adam)[29], an optimizing algorithm well suited for tasks that are large in terms of data and with noisy and/or sparse labels, and is the optimizer implemented in the methodology.

Loss is the heart of training a model, as it is the driving force for the algorithm to keep improving. A Loss function is a way of measuring how far off the algorithm's predictions are from the real truth. During training a machine learning model goes through an entire dataset several times (an iteration over an entire dataset is called an epoch), and does some predictions on the validation data set after each epoch. The loss function takes those predictions, compares them to the actual ground truth, and generates a number. Conventionally the higher the number the worse the predictions are, and thus a worse performance by the algorithm. This loss, or rather the change in loss is used to guide the updates done to the trainable parameters of the model in order to improve performance. There are a number of different loss functions conceived by mathematicians over the years which solve different problems. Finding the right loss function for the dataset and task is crucial for the algorithm to train properly and for results to be relevant. For instance in segmentation tasks with multiple classes there might be cases where some classes are much more populated than others, producing a skewed dataset. Using traditional loss functions can result in models unable to learn the unpopulated classes. Dice loss is just one loss function which aims to solve such issues.

Dice loss function has its origins from the Sørensen-Dice Coefficient, a statistics developed in the 1940s to measure the similarity between two sets. The coefficient was later made to calculate similarities in two images, and in 2016 adapted into a loss function to combat highly unbalanced classes in segmentation tasks. [30]

$$Dice\,Loss(y_{true}, y_{pred}) \;=\; 1 - \frac{2\sum_{pixels} y_{true} y_{pred}}{\sum_{pixels} y_{true}^2 + \sum_{pixels} y_{pred}^2}$$

When used in multi-class segmentation tasks, a generalized dice loss is used, which takes into account every class in the ground truth and weights them in accordance to their volume in the dataset. The higher volume a class is the lower the weight, the lower the volume of a class the higher the weight. Effectively this makes learning classes with a smaller representation in the dataset much more important to a model than classes with higher volumes. This does not falter the ability to learn high volume classes though, since each case of it will recur more often in training than small volume classes. Dice loss is therefore a very useful tool when dealing with imbalanced datasets.

In machine learning there are plenty of ways to measure how good the models are at what they need to do. Without going into too much detail, here is a brief explanation of the three metrics used in this thesis.

Accuracy is a metric which measures how often a classification algorithm predicts a data point correctly considering all predictions. If the cost of having miscalculations in the task is small, this metric should do just fine. The problem with Accuracy is that a model can gain a high accuracy score by classifying high amounts of insignificant classes (like background or ”No Tree”) correctly, but completely fail on important classifications. Problems like this makes having the correct collection of metrics in order to gauge the performance of a classification algorithm crucial to success.

Precision is a metric which measures how often a classification algorithm predicts a data point of a certain class correctly considering all the predictions of said class. This metric is great for tasks where the cost of false positives is relatively high. Precision determines how good a model is at predicting correctly if it predicts something at all. For context in a species recognition algorithm, precision calculates the amount of correctly predicted ”Pines” among all predicted ”Pines”. In order to understand the functions below, consider this: True positives are when the predicted class exists in the ground truth, false positives are when the predicted class does not exist in the ground truth, True negatives are when nothing is predicted and no class is present in the ground truth, and false negative are when nothing is predicted but there do exists a class in the ground truth.

$$Precision = \frac{True\,Positive}{True\,Positive + False\,Positive}$$

Recall is a metric which measures how often a classification algorithm predicts a data point of a certain class correctly considering all the data points of said class. This metric is great for tasks where the cost of false negatives is very high. Again for context, in classification, recall measures the amount of correct predictions the model makes on a class over every case it should have predicted said class.

$$Recall = \frac{True\,Positive}{True\,Positive + False\,Negative}$$

Under-, and overfitting are common problems when training a machine learning model. The goal of a model is to accurately predict the true data given an input of the same distribution. In some cases models are too general, not being completely wrong in their predictions, but also not very accurate. This is called underfitting, and is when the model generalizes too much. The opposite, overfitting, is when the model learns the training data too well, and tries to make too specific predictions that often are completely wrong, but seems correct based on individual cases in the training data. Regularization is a method to combat overfitting models. In this thesis l2-regularization is used which keeps weights from getting unrealistically big, yet never zero so they lose their value. Dropout is another method used against overfitting. Dropout is a functionality in neural networks that during training it drops the weights of random nodes from changing per training case.[31] This functionality has a regularizing effect in training and a great tool when facing overfitting models.

2.3 CNN Architectures for Image Analysis

In this section an introduction to the different neural network architectures explored in this thesis will be provided. The following techniques are mostly used for image segmentation, in this case on multi-band satellite imagery.

The *Unet* architecture is a convolutional neural net which consists of a contracting path and an expansive path working as an encoder-decoder structure .[32] It gets its name from its visual shape (see Fig. 2.8), which looks like the letter 'U'. The encoder half is made up of several instances of two-dimensional convolutions, an activation unit, and a max pooling operation for downsampling. The decoder part of Unet has several steps of upsampling, a concatenation with the corresponding feature map from the encoder, and convolutions followed by an activation unit. This enables the model to localize data and use a greater area of contextual data to predict the output for each pixel. The final layer is a 1x1 convolution to map the feature vector into the desired number of classes. As the structure and specifications provided in the cited paper by Ronnerberger et al. is considered the baseline/"vanilla" architecture, several pieces of this are open to be tuned and changed in slight manners.

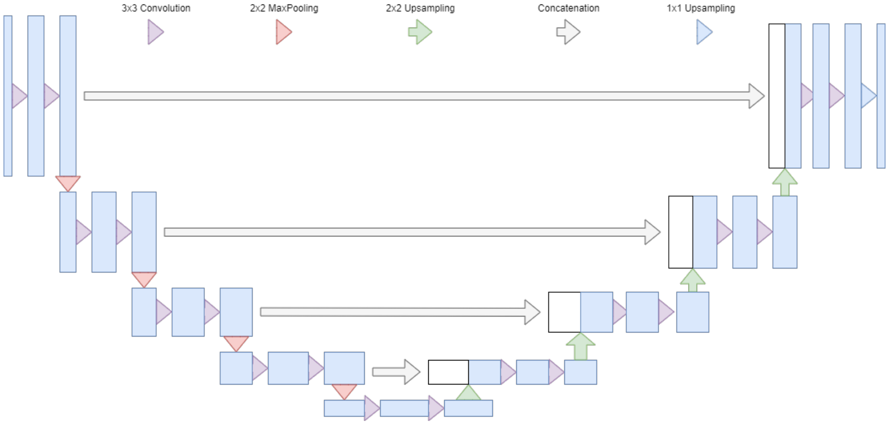


Figure 2.8: Unet structure visualization complete with the different operations on the feature maps.

Unet++ is a further development of the Unet architecture where instead of connecting each block in the contracting path to the corresponding block in the expanding path, the architecture nests them together (see Fig. 2.9).[33] The nesting is done by adding dense skip connections between blocks of the encoding and blocks of the decoding, blurring the lines of the encoder-decoder principles. This will theoretically allow the algorithm to carry more information over the gap between the encoder feature map and the decoder. Zhou et al. have redesigned the skip connections in this architecture to include a dense convolution. The convolution itself depends on the layer of the architecture.

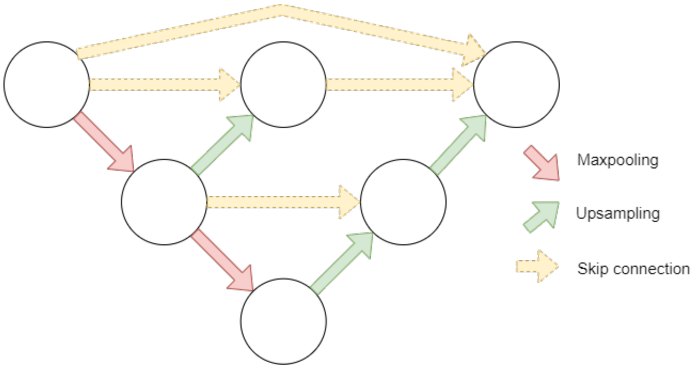


Figure 2.9: A visualization of a small Unet++ architecture to show the structure

In Unet++ the feature map undergoes several convolutions, both down-, and up-samplings at several stages before an output is generated. The output can be generated in two different fashions. One is averaging the segmentation of all branches, the other is picking the best branch, and only return that segmentation as output. This is called deep supervision and describes whenever we look deep within a model to find an output opposed to just look in the last layer. This technology was proposed for use on medical imagery, and is designed to reduce the semantic gap between the encoder and the decoder in a traditional Unet such that the segmentation can notice minor details and provide more exact segmentation. This promise of a more exact segmentation is what makes this a technology attractive to the forest species segmentation task.

Attention Gated Unet[34] is another variant of the Unet. This variant is intended to focus in on specifically interesting areas of the data, and tries to pass on only the relevant activations between convolution blocks. The attention gates are the novel technology, and is where the performance gain will lie. Attention gates allow the model to suppress less relevant areas of data, and be more attentive for salient areas.

The Feature Pyramid Network technology is developed by Facebooks AI Research

(FAIR), along with Cornell University and Cornell Tech[35] The convolutional network is a general use architecture for any common computer vision task. The structure (see Fig. 2.10) is a capable technology for object detection, region proposal, and image segmentation. In this paper we will only be using the image segmentation adaptation of the Feature Pyramid Network.

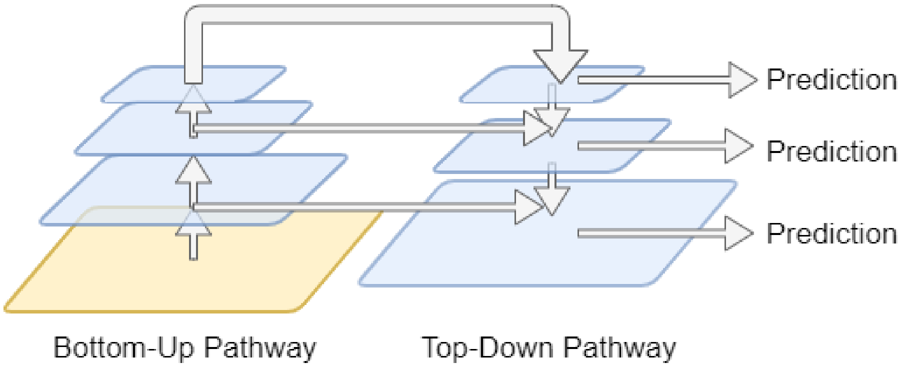


Figure 2.10: Feature Pyramid Network Architecture

The network consists of stages or pyramid levels with a bottom-up pathway and a top-down pathway with lateral connections. Any layer which produces output maps of the same size is considered to be in the same or corresponding stage. The bottom-up pathway acts like the encoder of the architecture, and the top-down pathway acts like the decoder, as the bottom-up uses max pooling, and the top-down uses upsampling, both by the same factor. The feature map output of each of these stages will be used as a reference set in the corresponding layer of the decoder. The top-down pathway uses upsampling, usually by a factor of 2, and merges the feature map with the reference features of the same spatial resolution from the bottom-up path. Every merged map is then carried into a 3x3 convolution to prevent aliasing, and at the end all outputs from the top-down layers are merged into a full prediction.

Residual Feature Pyramid Network only differentiates from the basic FPN by the use of residual blocks [36]. It is a further development of the FPN described by FAIR, and its purpose is to enable deeper learning. In FAIRs paper they prioritized simplicity in their design, but did experiments using a residual backbone for their architecture. Their results pointed in a direction that there are performance gains to be found, but the findings are inconclusive.

Residual blocks are the most important building block of the popular ResNet architecture. It is designed to make deep learning ”even deeper” by adding skip connections (see Fig. 2.11) that can carry the gradient further into the network.[37] This makes it possible to make bigger and deeper networks for complex tasks without major vanishing gradient descent issues. This is called the degradation problem. The degradation problem is a term for the inability of deep algorithms to learn identity functions. Counter-intuitively by adding more layers to the algorithm the accuracy of the model will start to saturate. Thus shallower networks might perform better than their deeper counterparts.

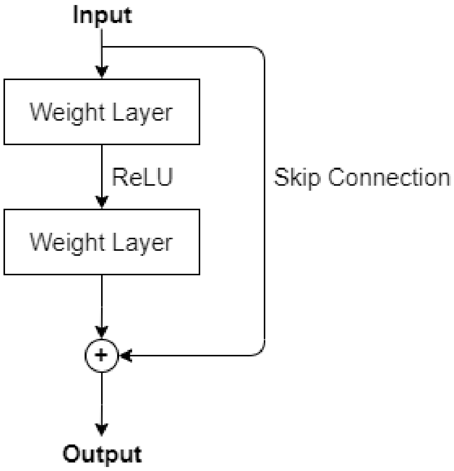


Figure 2.11: A residual block with a skip connection across activation layers.

Residual blocks introduces the ability to skip these layers where there are no performance gain to be found. With skip connections the algorithm will skip over some layers depending on how the error propagates back. This means during training, different parts of the algorithm will change since some are connected and some are skipped. The gradient will now have more momentum when propagating into the layers, and should travel deeper without vanishing. Also the number of layers is an important hyperparameter in any machine learning model, as it determines the complexity and amount of trainable parameters within. Using residual blocks will make the model dynamic, and give it the ability to learn itself how many layers it in effect should use, and how many it should skip. All this without any supervised guidance. The result of this is an architecture that will guarantee to at least equal the performance of a shallow network, with a possibility to learn even more complex relations.

Lastly a recurrent version of Unet with residual blocks has been proposed by earlier works.[38] The recurrent residual variant of Unet shows superior results in their benchmarks when compared to an original Unet. That said it does not guarantee better results in all use cases. A premeditated concern with this specific work is that this technology has mostly been tested on near perfectly labelled dataset with low amounts of noisy labels. The question still stands if a recurrent residual Unet would be robust enough to tackle less refined datasets.

A recurrent residual Unet is designed to be as similar as possible to a classic Unet, but with the addition of using recurrent residual blocks instead of plain convolutions. The strategy is quite simple; by concatenating the previous segmentation mask to the current image, and recurrently feeding this to the network. Theoretically this allows to propagate higher level information throughout the network [39].

Chapter 3

Use Case

This chapter will put the problem, the area of interest for the research, the available data, and its use in focus.

Spatial tree location, and species inventory data is fundamental for a wide range of vegetation management operations. Detailed information about the distribution and species of local vegetation is invaluable for ecology surveillance and conservation [40], wildlife habitat mapping [41], sustainable land use management and planning [42], [43] as well as infrastructure monitoring and risk assessment. [44]

With automated surveillance systems using up to date satellite images, and an occasionally updated LiDAR database, the need for patrols like that will be significantly reduced. The possible use cases of a well-trained deep learning model includes, but are not limited to, tree detection, tree height estimation or species recognition. Vegetation is a common issue when it comes to managing infrastructure, therefore data on growth, height, species, and other spatial information on vegetation is crucial for operations like planning tree trimmings, vegetation risk assessment, or estimating the amount of encroaching vegetation.

This thesis proposes an applied use of remote sensing technology to automate and make the aforementioned procedures more effective and less expensive. Such a tool will be able to locate trees, and classify their species using local data.

3.1 Area of Interest

This research will be conducted on a small set of geographical areas, and are chosen based on the availability of data. For example project GridEyeS is done in collaboration with StormGeo (A norwegian weather intelligence company) which serves several Norwegian clients, therefore analysing geographical data from Norway is preferred. There are specific types of data which are significant for the success of this thesis, such as satellite imagery. The satellite data presented in this thesis is located in Askvoll, Norway. (Fig. 3.1) The choice of area is not completely arbitrary as both recent LiDAR measurements and forest inventories are present in there.

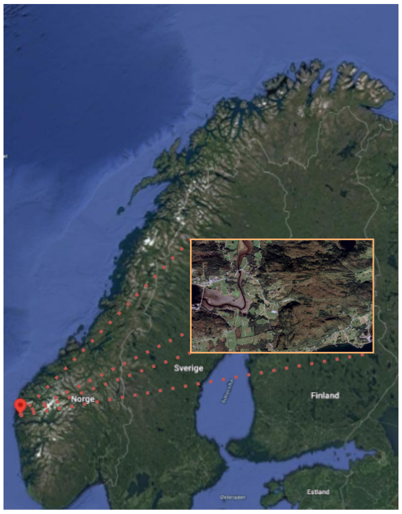


Figure 3.1: Location of Askvoll, Norway

3.2 Available Data

In any development of machine learning algorithms and use cases, the data is the most important part of it all. Collected for the purpose of investigating remote sensing capabilities in vegetation surveillance are 4-band satellite imagery, LiDAR measurements, and forest inventory, all covering Askvoll.

The satellite imagery are high resolution images captured by satellites orbiting over a geographical area (see Fig. 3.2). Images that have been collected for this thesis are multi-spectral images with a resolution of 0.5m per pixel provided by Pleiades-1. A Normalized Difference Vegetation Index (NDVI) band has also been calculated for these images.

$$NDVI = \frac{NIR - Red}{NIR + Red}$$

The calculation which requires the presence of Near Infra-Red(NIR) and visible red light is an approach to visually estimate live green vegetation, and have been proven to be a valuable feature for vegetation detection. [45]

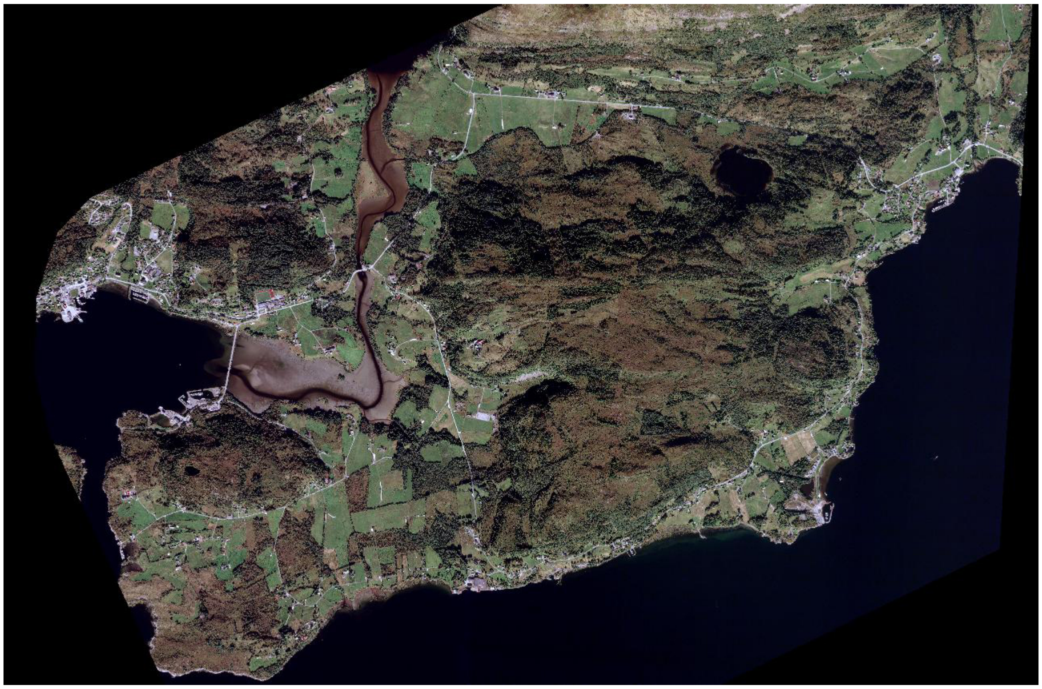


Figure 3.2: High resolution satellite Image of Askvoll area

LiDAR is a geographical measurement that generates 3D point clouds of the environments. It works by sending out laser signals towards the ground, and measures the time and strength of the returning signal. LiDAR can with this technique map vegetation height data over a given area with very high resolution and precision. Such surveillance is done by some aerial vehicle with a mounted LiDAR device, but this process can be very expensive. It also can not cover great areas at the time, so the mapping is usually done over specific areas of interest. These measurements are a much used tool to survey power grids, and therefore the only data available will be around the power grid itself (see Fig. 3.3), not necessarily dense forests. For this thesis such LiDAR data have been procured. Since the goal with this data is not to estimate height, but to locate the trees as one entity, the data have been turned into a binary mask. This binary masks marks where ever there exists vegetation 2.5m above the ground. So this leaves out any shrubbery or ground level vegetation, and provides a focused data set on tall vegetation.

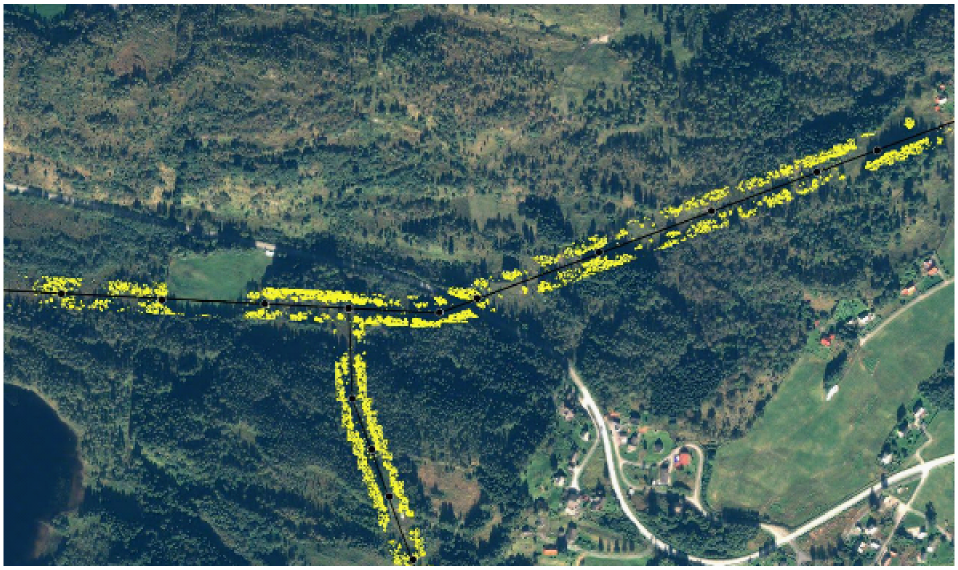


Figure 3.3: 2.5m LiDAR map projected over Askvoll. This shows the small scale of the LiDAR measurements, and how it is only collected around the power lines.

NIBIO provides a free forest inventory dataset covering almost all of Norway (see Fig. 3.4). It can be viewed by anyone on their website, and it is from here the species dataset used in this paper's approaches is collected from. In NIBIOs very diverse datasets, one can find several different data classes, but for species segmentation we extracted their forest inventory masks. These inventories include Pine, Spruce, Deciduous, Coniferous, and Mixture. For a more specific definition of how the masks are calculated, here are the descriptions given by NIBIO.

- *Spruce dominated*: Where the volume of spruce constitutes more than 50% of the total volume.

- *Pine dominated*: Where the volume of pine constitutes more than 50% of the total volume.

- *Deciduous dominated*: Where the volume of deciduous trees constitutes more than 50% of the total volume.
- *Coniferous mixed*: Where the volume of spruce and pine makes up more than 75% of the total volume.
- *Mixture*: Where the volume of spruce, pine, and deciduous are all less than 50%, and the volume of spruce and pine is less than 75%.
- *Not wooded*: Where the volume of spruce, pine and leaves is equal to zero.

These 5 categories of trees define the target classes in some of the methods mentioned in the methodology chapter of this thesis.

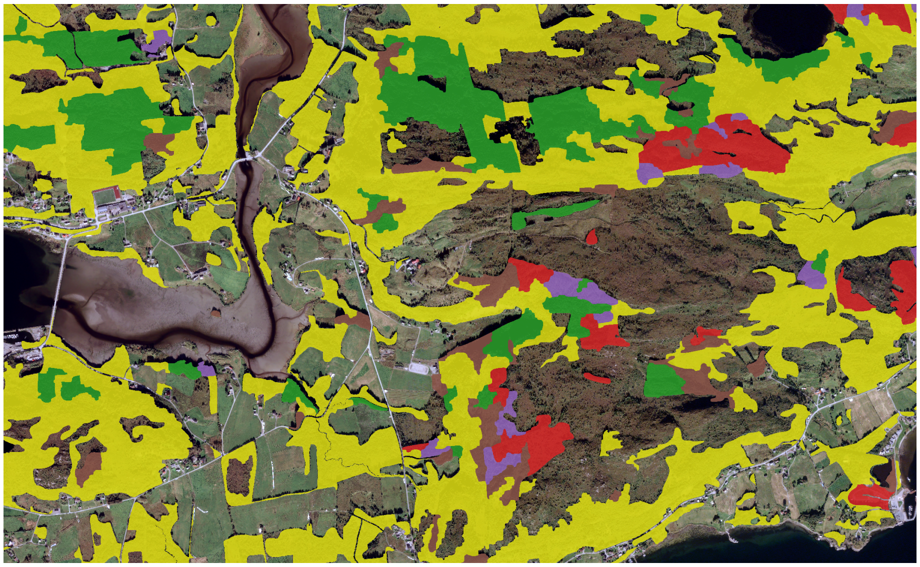


Figure 3.4: NIBIO forest inventory data visualized in QGIS over Askvoll area. Yellow = Deciduous, Green = Spruce, Red = Pine, Brown = Mixture, and Purple = Coniferous.

With all data comes some downsides, this is why carefully choosing what data to use is a very important part of putting together a dataset that machine learning algorithms can perform well on. In this case there is not a lot of carefully produced and processed data available to us, so there will be some flaws and some struggles to overcome with the dataset used in our method. Luckily high resolution satellite imagery has become such a well-defined service, so getting the perfect satellite images for the Askvoll area with useful analytic bands is not a problem in these experiments. Nevertheless, the LiDAR and the Forest inventory comes with some less attractive characteristics. LiDAR is in this case only provided in spatial squares of around 20m x 20m. This limits the research and any eventual solution to a very small scale monitoring. The data is also gathered over the power lines, so any LiDAR data on unaltered nature is hard to come by. For the forest inventory of which improvements will be put forward, come in a very coarse form. It only provides a rough estimate of what tree species are dominating specific areas. The inventory also only covers dense forests, and leaves out any single trees outside of any clearly visible forest. This characteristic, along with the classes containing a mixture of species, can represent some significant noise in the dataset.

Chapter 4

Methodology

The method developed in this thesis consists of three major blocks as illustrated in Fig. 4.1. The first block (A) is a lightweight, small-scale binary tree segmentation Unet algorithm. This will input 4-band satellite imagery paired with LiDAR labels to produce tree location masks. The second block (B) inputs the raw NIBIO species data, performs a series of refinement processes, and merges the refined data with the Unet's locations masks (A). In the third block (C) the merged labels is used to train a multi-class species segmentation model based on the Residual Feature Pyramid Network architecture. This final block is then able to generate an estimated mask of species present in a satellite image.

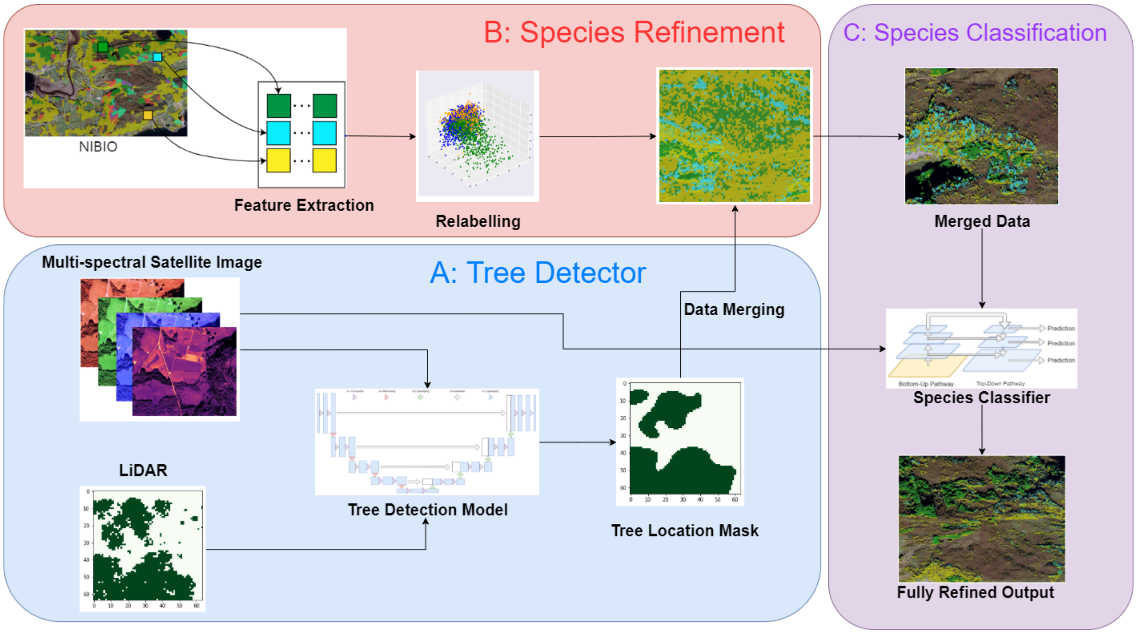


Figure 4.1: Visualization of the full pipeline including A: tree detection algorithm, B: feature extraction and relabelling, and C: species classification. The output of A and B is merged to create a refined map inputted to C.

4.1 Coding Setup

The coding, development, and testing of these technologies is done in Python with some help from other frameworks and softwares. Following is a shortlist of the most important libraries and tools used in these experiments:

- Programming language Python 3.7
- Integrated Development Environment Spyder 4.2.0
- Geographical Information System QGIS 3.12
- Essential Libraries: Gdal, rasterio, numpy, matplotlib, seaborn, pandas, keras, and tensorflow

- The training is done on a 8 GB Vram Nvidia GTX 1080 Graphical Processor Unit.

4.2 Binary Tree Segmentation Block

Segmentation in the machine learning field is a technique of categorizing an image into several areas of different classes. A deep learning model can be trained to identify parts of those images and put them into classes like trees, houses, shrubbery. Provided new data it should be able to make an educated guess of where in the image certain elements are, like a tree or other vegetation. With this first approach we are trying to identify class labels of single pixels in a satellite image. Our dataset comprises 2600 64x64 pixels satellite images with a 0.5m spatial resolution, and a matching target dataset of LiDAR masks depicting pixel-specific true or false labels of whether there is a tree above 2.5m here or not. The LiDAR used is actually a more dynamic measurement of estimated canopy height, but given the goal of predicting location, this data has been thresholded at 2.5 meters. The resulting mask is then a binary map of any vegetation that rises 2.5m above ground or beyond. Below are examples of this data (Fig. 4.2). This model will take the two data types as input; the RGB-NDVI satellite image, and the target LiDAR array of true or false labels describing if there is tall vegetation at this pixel or not.

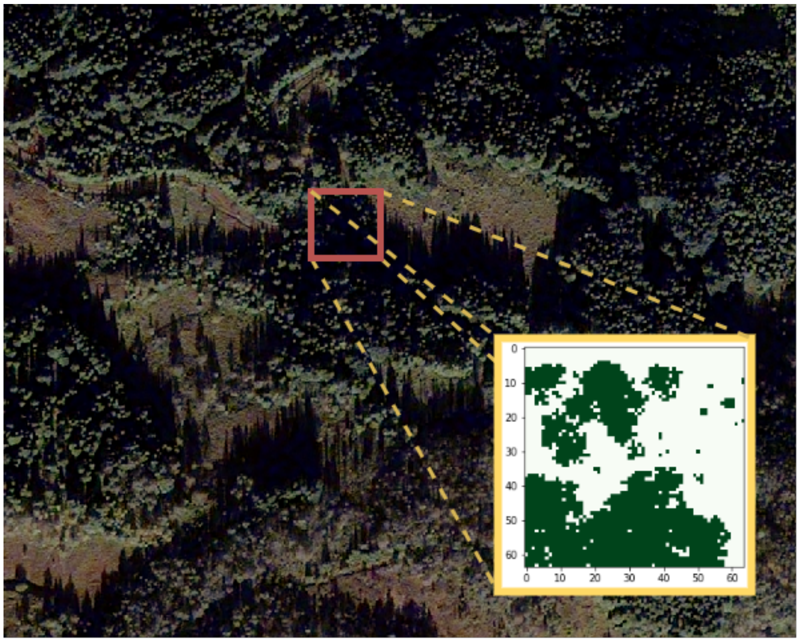


Figure 4.2: 64x64 LiDAR square viewed over the entire satellite image.

Deep learning algorithms usually perform better on bigger data and bigger images, but in our case we only have small scale squares of LiDAR data. This forces a downscaling of the images to avoid falsely labelling pixels. If the image size is increased the data will provide an increasing amount of false negatives, or in other words pixels labeled as ”no tree” when in fact there is a tree there, and would confuse the learning algorithm. This is due to the LiDAR data being collected only over the power lines with a certain width and height, which in this case is a little above 64 pixels. For this reason the chosen architecture is the smallest architecture proposed in this thesis, the Unet (see Fig. 4.3).

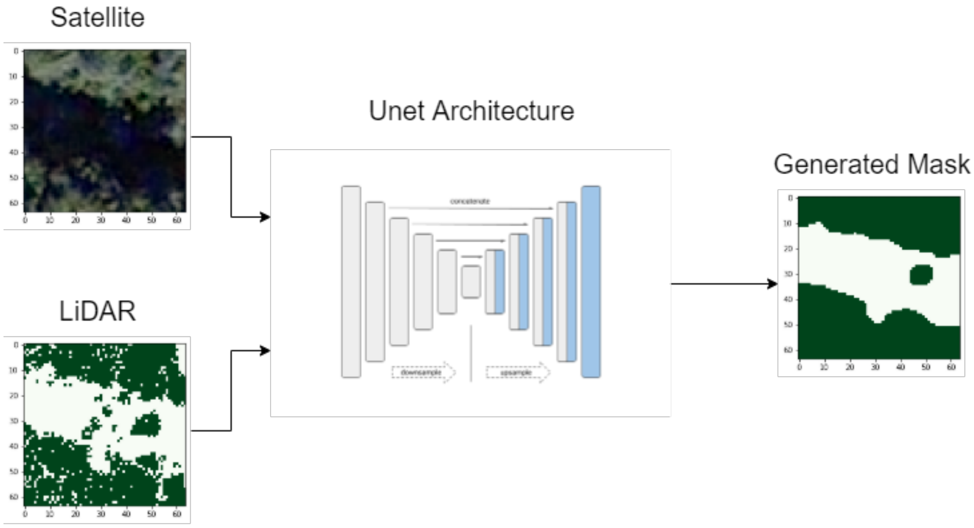


Figure 4.3: Satellite images paired with LiDAR measurements are used to train a Unet model. Afterwards the model can with the input of only a satellite image, and without the need for any paired LiDAR measurements generate masks of forests.

The basic functionality of the Unet architecture is explained in background, but this section will go more in depth of the implementation. Keras, an open-source API for artificial intelligence, is the main framework for the implementation of all the algorithms in this thesis.

The first layer of the Unet is an input layer with the addition of a normalization layer, and is connected to the first encoder block. Each encoder block in the Unet consists of a convolution layer with a number of filters according to the layer number, a kernel size of (3,3), a stride of 1, ”same” padding, l2-regularization with a small weight decay of 0.0001, a dropout layer dropping 10% of nodes while training, and a Rectified Linear Unit activation. Then a batch normalization layer, another convolution layer, and lastly a maxpooling layer. Every encoder block is connected to the next encoder block, except the last which is connected to the first decoder block, and all encoder blocks are concatenated with the corresponding decoder block. Every decoder block consists of a deconvolution layer with the same parameters as a convolution layer in the encoder blocks, the only difference being a stride of (2,2). The output of the deconvolution is concatenated with the corresponding encoder block output, and is followed by a convolution layer, batch normalization, a dropout layer, another convolution layer, and is finally connected to the next decoder block. The last encoder block is connected to a softmax activation function mapping it to the final output layer.

Path Point	**Output Shape**
Input	(64, 64, 4)
Encoder Part 1	(32, 32, 32)
Encoder Part 2	(16, 16, 64)
Encoder Part 3	(8, 8, 128)
Encoder Part 4	(4, 4, 256)
Decoder Part 1	(8, 8, 128)
Decoder Part 2	(16, 16, 64)
Decoder Part 3	(32, 32, 32)
Decoder Part 4	(64, 64, 16)
Output	(64, 64, 2)

Table 4.1: Each part of the Unets encoder pathway and decoder pathway with its output shape

All of these layers add up to 1,944,178 trainable parameters, and fully trained the Unet can produce some very accurate tree location masks. Below is one example of the output of this Unet model (Fig. 4.4).

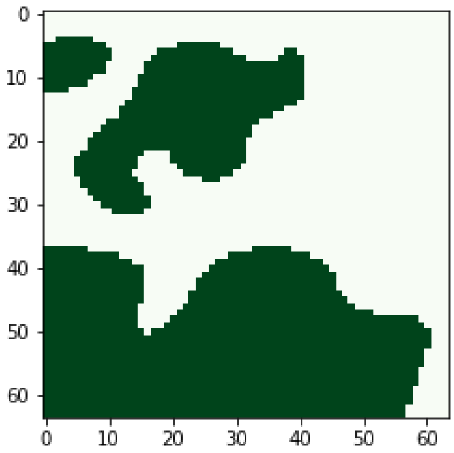


Figure 4.4: An example of a tree mask generated by the fully trained Unet model.

4.3 Tree Species Dataset Improvement Block

One of the goals of this thesis is to broaden the arsenal of vegetation monitoring tools. Until this point this thesis has provided a tree detector algorithm able to segment small scale patches of forests, and given only a satellite image will generate a mask of the tree locations. In order to expand on this, let us look towards another aspect of vegetation, namely species. In available data (3.2) another dataset provided by NIBIO is described which gives a rough estimate of what tree species are dominating in specific regions. The perfect machine learning algorithm will always just be as good as the data it is provided, and with this data there are some non-idealities. The non-ideal characteristics include coarse labels only regarding dense forests, some very populated classes, some equally underpopulated classes, and classes with diffuse purposes. All of these contribute negatively in a machine learning context. Improvements need to be made. To do so this thesis proposes two things; merging species data with location data, and clustering classes based on their visible features instead of the species data. Firstly, there is some motivation to keep the feature vectors for the dataset as big as possible, so simply merging species data with LiDAR data does not necessarily benefit the algorithm, as this will shorten the dataset. Since there has already been developed a tree segmentation model for this area, the idea of combining the binary output of this model with the species data seems much more prosperous, and can cover a vastly larger area. This way the dataset will retain all its species data and also provide much more accurate masks in terms of tree locations.

For a closer look at the non-ideal characteristics in the species data, consider these examples. Since the data set is so coarse, large areas of actual trees remain unlabeled, and will only provide noise in the data, and hinder any machine learning algorithm to recognize the patterns of species. As one can tell from the examples, only the densest forests are labeled with species data (see Fig. 4.5), but slightly sparser forests are excluded.

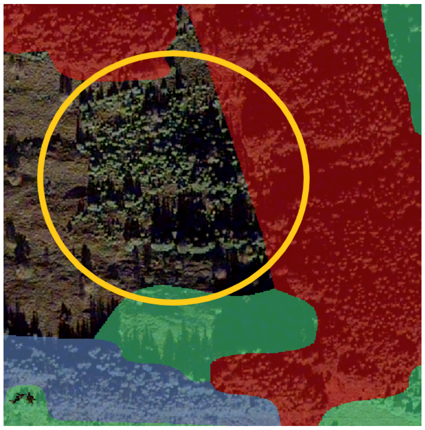


Figure 4.5: One example in the species set where great areas of trees remain unlabeled.

It is not only sparse forests that are introducing noise into the data, there are also many occasions where the data labels large areas of no trees as a specific species.

The example below shows (Fig. 4.6) a big area almost completely empty of live growing vegetation, yet is still labelled as if there is a forest there.

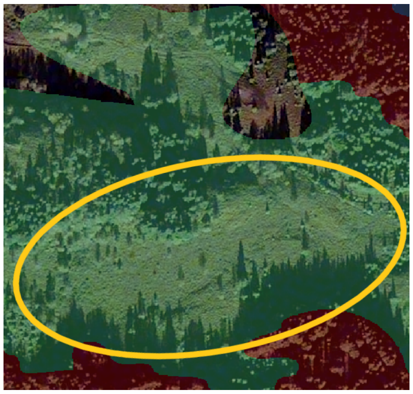


Figure 4.6: One example in the species set where great areas of no trees are labelled as trees.

Lastly the species data labels forests with unrealistically clear borders. In nature a forest does not contain only one species of trees, and are also not defined with such clear borders between each other. NIBIO points this out by defining classes of mixed species, and also defines that the labels suggests that in the area of a certain class there is a majority of trees of this species, not 100%. In machine learning, any diffuse class will end up inhibiting a model from learning properly. An example of this false labeling is when there are areas labelled as one species, but actually contains multiple. Some of these cases are clearly visible on satellite images. Below is an area populated by several species of trees (see Fig. 4.7), but is entirely labelled as a single species in the data set.

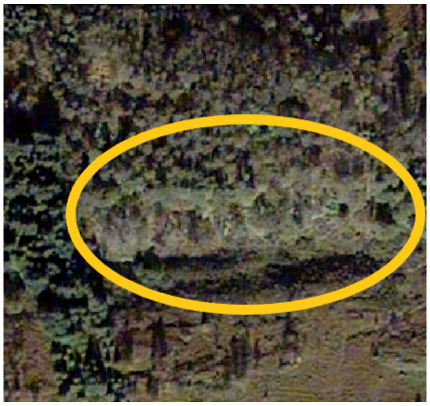


Figure 4.7: An area of the data set where every pixel is labelled as one species, but there exists a clear visible discrepancy between the forest within the circle and the forest outside it.

To solve the multiple issues with the species data, this thesis proposes a more sophisticated solution to refine our data. Firstly, an aspect of tree location needs to be injected into the species data. The masks are so coarse that a huge quantity of useful information is lost. By merging the species masks with tree locations generated by a binary tree segmentation model, the output is much more fine-grained, and can provide tree-specific data. This is done by generating tiles of tree location data, combining these together to produce a location map over Askvoll, then merge it with the species data. The process will then remove any false labels in the data set, and add any tree location not covered by the original species data.

So far the approach does not solve the issue of any wooded areas not covered by the original species data, as the data produced by the tree location refinement would then technically be of an unknown species. By this logic another refinement is needed to complete the data, a refinement on species.

We assume that any trees that share similar characteristics also share the same species (see Fig. 4.8, regardless of their pre-assigned label. We also assume that the NIBIO data is correct in saying that the majority of the trees within each labelled area is true. By calculating what features are most prevalent within each label we can expand on this, and see if the same feature is found in other places of the data. If so, that data can then be re-labelled based on their true characteristics rather than their majority label. (see Fig. 4.8)

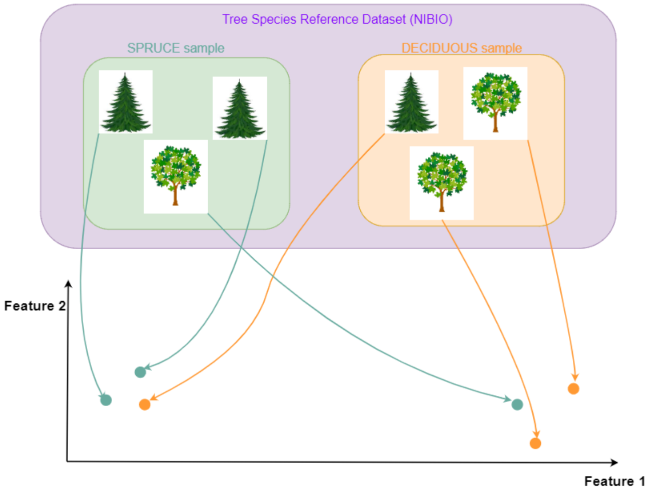


Figure 4.8: A visualization of the logic behind the relabelling method.

To implement this, tiles are collected from areas of each of the main NIBIO classes, these include pine, spruce, and deciduous. Mixed and coniferous contains multiple different species and would only introduce noise to the data. The contents and the features of these tiles are thus associated with each of their classes. For the features extracted, this approach looks for color, texture, and vegetation index, and assumes that trees with similar values of these features are of the same species.

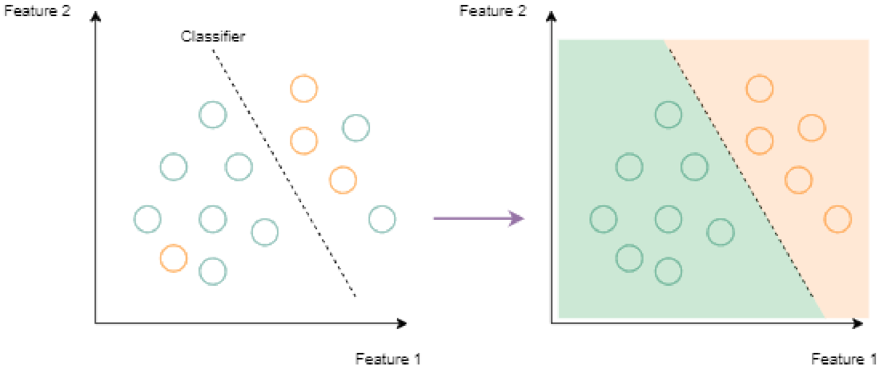


Figure 4.9: If a classifier is trained to recognize features in small patches, it can relabel each patch into their true label.

Considering this as an entirely new dataset, containing color, contrast, texture features, and NDVI, these features are paired with the original label, and used to train a machine learning classifier (see Fig. 4.9). Several machine learning algorithms are suitable for doing this traditional task, like support vector machines or decision trees. When fully trained, this classifier has learned what features correlate to which class. Said classifier can then be fed patches of satellite images and the features of its contents, and relabel the species within. The resulting species mask can be viewed below (Fig. 4.10).

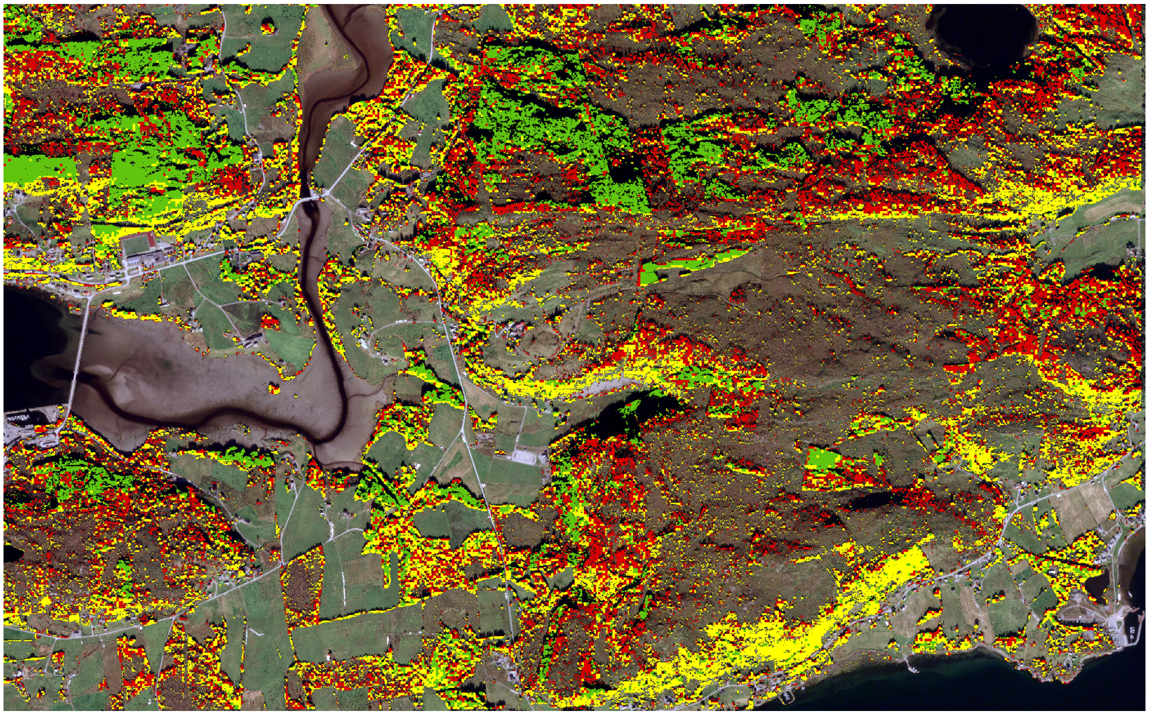


Figure 4.10: Species data and area after data enhancement, Yellow = Deciduous, Green = Spruce, and Red = Pine.

This data is the last iteration of data in this masters thesis, and will be used to manufacture the final machine learning algorithms for species classification. In the result section, the results of all fully trained segmentation models will be revealed, as well as an in depth analysis of any findings in discussion.

4.4 Multi-class Trees Species Segmentation Block

The proposed method is to use the refined species dataset as a ground truth label for training multi-class segmentation algorithms, not just binary ”tree or no tree” segmentation. The species dataset covers all of Askvoll, and this opens up some distinct advantages. Without the dimensional constraints of the LiDAR data, a segmentation algorithm on species can cover any dimension. For this thesis the chosen dimension is 192 x 192 pixels, and is mainly based on practicality. The training of these architectures is done on a limited amount of GPU power and memory, so this effectively lowers the possible resolution of the images. The dimension of 192 is still the largest dimension considered trainable on this setup, yet small enough to include a diverse and relatively large amount of data. For all the models trained in this section, the data is split into 3 non-overlapping sets. One for training, one for validation, and one for testing. The distribution of these are roughly 70% training data, 20% validation data, and 10% test data. This is in accordance with traditional machine learning methods. A little over 2000 images have been used for training, a batch size of 32, Adam as an optimizer algorithm, and the learning rate have varied between 0.0001 and 0.00001 with a decay rate of the initial learning rate divided by the current epoch number.

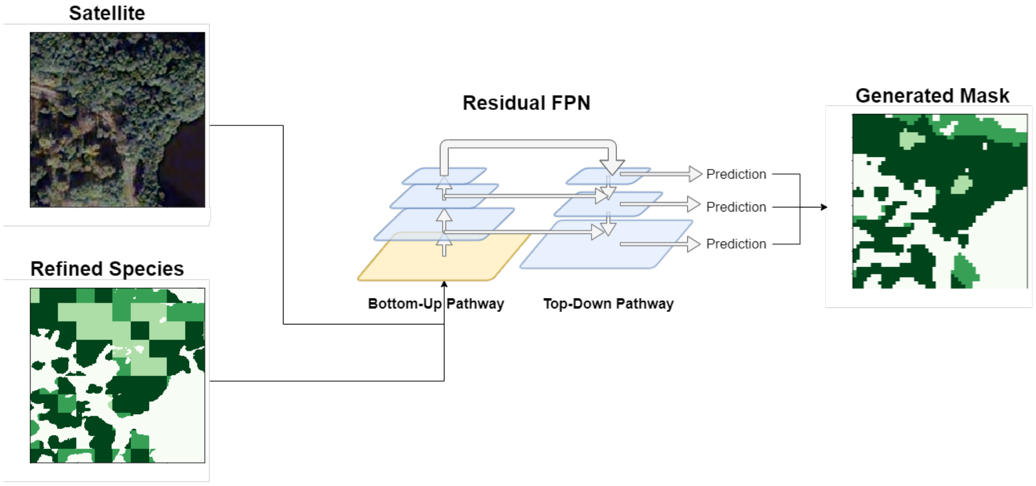


Figure 4.11: A Residual Feature Pyramid Network model is trained on multi-spectral satellite imagery and corresponding species data.

The full pipeline (Fig. 4.11) of this approach is described as follows. Using the trained binary tree segmentation model, a mask covering all of Askvoll's tree locations is generated using satellite images and LiDAR as input. The species mask is clustered using a texture-based feature extraction approach to refine the classes, and also remove some troublesome labels like mixed and coniferous. The tree location mask and the relabelled species mask is then merged, such that any location of trees in the species set which is not included in the tree location mask is removed. Also any tree location outside the original species data gets assigned an estimated class. Thus resulting in a fully refined dataset both in spatial features and species data. This refined data paired with satellite images is used to train a multi class segmentation model in order to provide accurate species masks over satellite imagery.

For this particular case of classification we propose using a feature pyramid network architecture, preferably with the inclusion of residual blocks. This is based on the findings of the comparative research done in this thesis. The proposed architecture is built like a FPN with the backbone of a ResNext-50 structure.[46] The architecture is similar to the one used in ”Residual Bi-Fusion Feature Pyramid Network for Accurate Single-shot Object Detection”. [36]

All layers of the residual FPN are fitted with residual blocks and skip connections, to further architecture complexity (see Fig. 2.11). Every encoder layer is concatenated with the corresponding decoder layer through a lateral connection, and connected with the next layer in the same path. Each layer in the decoder path is generating a prediction, connected to the next, and finally all predictions of their respective dimension generates the output mask. The encoder layers include a 2-dimensional convolution, batch normalization, and a ReLU activation. Training this model showed no symptoms of overfitting, so this architecture does not include any kernel regularizers or dropout. Given the immense size of this architecture with ~ 28 million parameters and 50 layers, a full summary of the model will not be included here.

Chapter 5

Results and Discussion

Following are the results of every fully trained model fitted for this use case. For each model and task is a short repetition of its purpose, the specifics of the model, and the resulting output. The findings will be discussed and analyzed here as well.

5.1 Binary Tree Detection

The Unet Tree detectors main purpose is to locate trees. This classifier model deals with the task of classifying each pixel in a high resolution satellite image to either True/Tree or False/No Tree. Structure-wise it is based on a basic Unet architecture without many modifications. The satellite and LiDAR images are extracted from .tif files covering the Askvoll area. 2600 windows have been extracted along the power lines of Askvoll, resulting in a training set of 2200 windows, and a validation data set of 400 windows. For context the image size is set by 64 x 64 pixels. By the end of the training phase, the fully trained model produced these measurements over the validation set. An example prediction generated by the Unet model is also provided (see Fig. 5.1).

- Dice Loss: 0.1737
- Accuracy: 87.47%

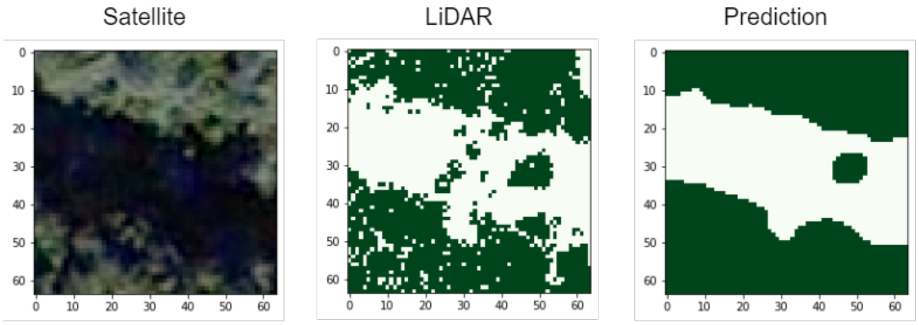


Figure 5.1: A comparison of the inputted and outputted data. From the left: A, Satellite image previously unseen by the model. B, the corresponding LiDAR to the data A. C, the models prediction of how the LiDAR of A should look without having seen the data B beforehand.

This thesis has shown that using LiDAR data, even in its small scale form can produce very accurate tree location algorithms. Although LiDAR is not the most accessible resource, and this thesis have not been able to test whether the results found in the fully trained Unet segmentation model is robust enough to keep its location accuracy high enough over longer periods of time without retraining, the results shows that using deep learning algorithms can do this form of vegetation surveillance.

Also worth noting is that this Unet model is developed over the "natural" distribution of trees in the local area, meaning that no heavily interfering or complex preprocessing is needed to develop the "perfect" dataset. Even though the dataset was very skewed towards the class of "Tree" opposed to "No Tree" (see Fig. 5.2 and Fig. 5.3), using a relevant loss function and some regularization techniques limited the impact of this, and in turn the model was able to produce accurate masks nonetheless.

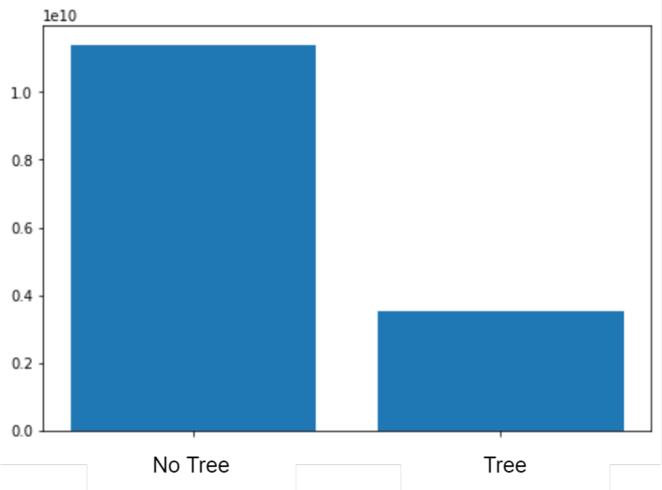


Figure 5.2: Frequency table of the data used to train the Binary Unet tree detector segmentation model.

By reviewing the confusion matrix of the same model, the result seems to be featured there as well. With no significant amount of false positives or negatives in the produced masks.

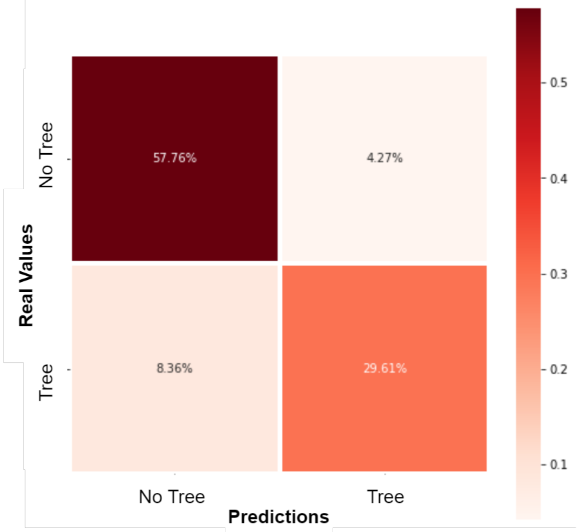


Figure 5.3: Confusion matrix comparing the Unets predicted values with the actual values of the LiDAR data

This approach should before any commercial implementations be tested on multiple areas, on multiple data bands provided in satellite measurements, and any temporal advantages or disadvantages should be investigated. Beyond that the findings in this thesis are pointing in the direction of automated remote sensing being a powerful and efficient tool in vegetation detection.

5.2 Tree Species Classification

For species classification several technologies were tested on the available dataset in order to see which one would serve the purpose best. With an input of 192 x 192 pixels images over 4 bands, Red,Green,Blue, and NDVI, the data is matched with a species mask provided by NIBIO improved with location data generated by a binary tree segmentation model and a feature extraction approach. The model's purpose is to classify what species of trees are located at each pixel. Below is a table of the test scores and prediction masks for this comparison (scores in Table 5.1 and masks in Fig. 5.4).

Model	**Dice**	**Acc**	**Prec**	**Recall**
Unet	0,4662	0,8151	0,8208	0,8102
Unet++	0,7793	0,6990	0,8072	0,6202
AttUnet	0,3457	0,7284	0,7334	0,7116
ResFPN	**0,2943**	**0,8261**	**0,8286**	**0,8248**
RecUnet	0,3305	0,7909	0,7917	0,7903
FPN	0,4389	0,7888	0,7889	0,7889

Table 5.1: Test Results of all architectures. Leftmost column is the architecture, followed by the dice loss, categorical accuracy, precision, and recall of the specified architecture instance.

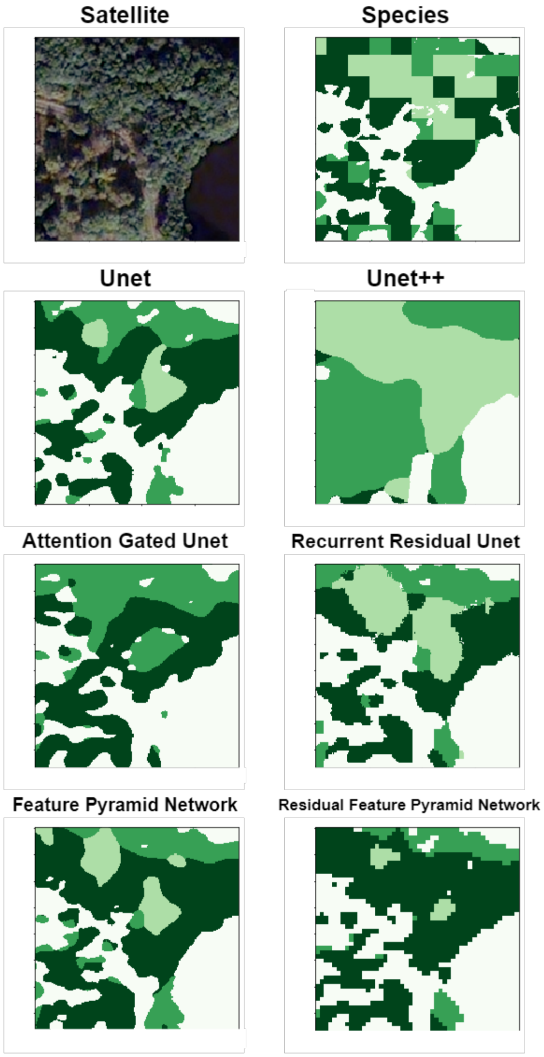


Figure 5.4: Top-left is a satellite image which has been inputted to fully trained instances of each architecture, top-right is the corresponding species mask to the inputted satellite. The following are a collection of all generated masks outputted from fully trained instances of each architecture provided the inputted satellite image.

The species dataset is several times more complex than the tree location, with images being bigger and having multiple classes to segment. Logic infers that a more complex architecture should fit this task better than the binary tree segmentation, but this is not easily measured with simple scores and metrics. The generated masks of all the trained models can only tell us so much. Looking at small areas at the time only using our human intuition will never get us anywhere. Let us take a deeper look at the individual models to see what mistakes they make, and where their strengths lie. But before analysing the models, an important note in the dataset is its representation of classes. It has been described earlier that the datasets have been skewed towards some classes, and this is important to keep in mind when analysing further. Below is a frequency table of all classes in the training data (Fig. 5.5).

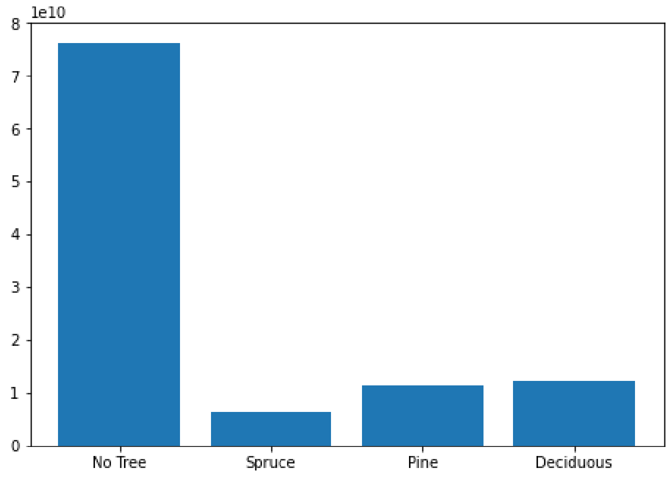


Figure 5.5: Frequency table for the species training data.

From here one can tell that any model trained on this data should be perfectly able to predict any areas where there exists no trees, as there is simply so much of this data. For this reason when plotting the confusion matrices for analysis, only predictions made on actual species are part of the validation. A confusion matrix is a simple plot that compares the predicted labels of the model to the actual values of those predictions. By analysing confusion matrices (see Fig. 5.6) it is easy to tell where the model shines, and where it fails in terms of class predictions. On the x-axis are the real values, and on the y-axis are the predicted ones. The diagonal is then ”True Positives”, or in layman terms; all the correct predictions on actual species. These numbers comes from the pixel values in every image of the validation set and their generated predictions. They are then added together and divided on the data length to provide a percentage in the matrix rather than a number.

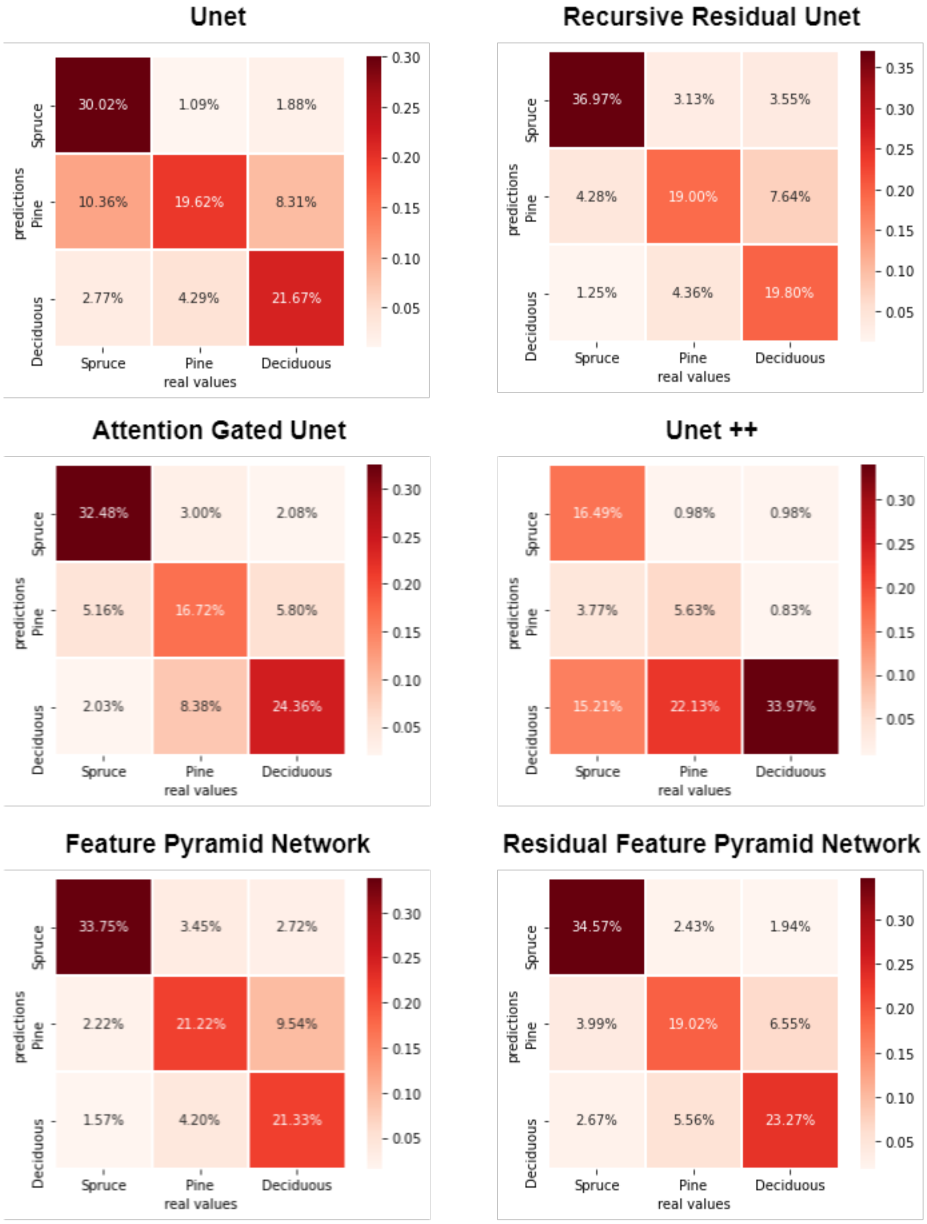


Figure 5.6: Confusion matrices of all tested architectures. On the x-axis are the predicted values, on the y-axis are the actual values. The diagonal squares represent when the model is correct in its prediction.

Both the confusion matrices and the correct predictions are calculated on only actual species and leaves out the ”No Tree” class, as the importance of this use case is species recognition.

Reviewing the numbers and the matrices (see Fig. 5.6) there are some clear indications for which models provide higher performance. The classic variant of Unet, although it shows excellence in spruce and deciduous recognition, does have a high amount of false positives for pine. Unet++ entirely neglects the pine class, and shows a high error rate on deciduous predictions. Attention gated Unet and the classic feature pyramid network have some minor struggles with certain classes (deciduous in AttUnet, and pine in FPN), but proves to be robust predictors for the other two classes. The last two architectures have one thing in common, other than providing the best predictions, they contain the highest number of trainable parameters above every other architecture tested in this thesis, and both make good use out of residual blocks. Recurrent residual Unet consists of $\sim$ 24 million trainable parameters, and the residual feature pyramid network consists of $\sim$ 28 million trainable parameters. The third biggest architecture being the classic feature pyramid network consisting of $\sim$ 18 million trainable parameters. It seems clear that more complex neural network structures provide better predictions, so given the scalability of neural networks in general, there is definitely room for improvements beyond these results.

As a final question in this discussion; are these solutions feasible for improving vegetation monitoring? The binary tree segmentation model shows great results with a simple and not too heavy weight(referencing parameters and digital size) architecture. It provides exact tree masks showing where any vegetation taller than 2.5 meters is located. This tool can be used anywhere LiDAR and satellite data is available, and provide automatic tree location estimation. It is a machine learning tool in its simplest form, as it does not take into account anything other than if there exists a tree here or not. For many problems in the real world, this can be enough, but most likely more sophisticated solutions with more in depth information is preferred. This thesis attempts species classification using remote sensing. It shows that this field of classification is very much possible to do with remote sensing techniques. The comparison done in this thesis proves the validity of remote sensing in large scale satellite image species segmentation. Bigger architectures like residual feature pyramid network and recurrent residual Unet provide the best prediction abilities. The results also suggest that the use of residual blocks can be a credible choice when doing vegetation classification, but concluding on any of these hypotheses requires further research.

Chapter 6

Conclusion

This section provides a summary of the findings, and tries to answer the previously set research questions. Any possible future work and improvements which have been found feasible during this research is also briefly mentioned.

6.1 Answers to research questions

Research Question I: *How can today's available satellite image data be effectively utilized to monitor vegetation?*

To answer this questions, one has to take into account the ease of such a solution. The several services providing satellite imagery of almost any area around the globe up to several times a day, makes satellites an easily accessible commodity for any agent wanting to do vegetation management. Manual inspections will of course always be accessible to any companies, but in this era one should look towards automated and smarter solutions to save both the environment and local resources. This thesis have found that tree location masks using relatively lightweight neural net segmentation algorithms do provide accurate masks of tall vegetation. The characteristics of the tree location masks can be specified per scenario, but in this case the segmentation was based on vegetation 2.5m or taller above the ground, and provided accurate masks of forests both dense and sparse.

Forest inventories like the ones used in this thesis should cover even more requirements and allow automated species recognition. We have in this thesis proof that even though the raw data available is too coarse and unrefined, there are ways of processing species data to the point that it can indeed produce well performing segmentation algorithms. These algorithms are very much capable of recognizing a set of tree species, and hold the potential to do even further recognition when data becomes available.

Research Question II: *What are some of the most effective satellite image analysis techniques when applied to vegetation monitoring?*

To test the capabilities of remote sensing algorithms, a handful of technologies varying in complexity have been implemented and tested on this use case. The research has shown that Unet, given vegetation location data and satellite imagery with an NDVI band, can automatically produce accurate masks of trees and forests. It has uncovered that architectures with a higher amount of trainable parameters, like Residual FPN and recurrent residual Unet performs best on species classification. Also present in the research is the possible significance of residual blocks, since they are a part of the two best performing technologies.

6.2 Contribution

This thesis has tried to find efficient automated solutions for vegetation surveillance aimed to help monitor live vegetation in Norway. Such a smart and cost-efficient approach to monitor vegetation can benefit society as a whole, as it would lessen the resources poured into manned surveillance patrols, and can help to plan smarter solutions for vegetation management. A binary tree detector model using satellite images and LiDAR height data provided a strong and precise classifier. Using a simple Unet architecture proved that the task of segmenting vegetation in satellite images, even with a limited amount of LiDAR data is very much possible, and can be a resourceful tool.

Pixel-wise species prediction based on NIBIOs forest inventories is a starting point for developing a robust species classifier. Although local data can be imprecise and impractical for machine learning, this thesis provides a different angle to the problem. By utilizing several artificial intelligent solutions on non-ideal data it is possible to process and refine such data enough to warrant accurate model predictions. This thesis has shown that segmentation algorithms do fit naturally into this niche of surveillance, but that they do rely on what datasets are provided and available. Nevertheless this thesis should provide enough evidence to show that the processing of local data which is not accurate enough for training machine learning models can improve and refine it to the point where AI models can learn to solve complex tasks.

6.3 Future Work

For future work we propose extending the data analysis into new territories. This includes more labels, more sophisticated measurements, more satellite bands, digital elevation models, and adding a temporal feature. The research conducted in this thesis covers satellite images captured in the summer, but to further improve the usability of this method as a tool, images from different times of year should be included in the dataset. As vegetation has a yearly life cycle and thus visibly changes over time, a temporal addition to the dataset would extend its feasibility remarkably.

Bibliography

[1] ESA-Business-Applications. *Smart Grid Eye, From Space To Sky.* URL: https://business.esa.int/projects/grideyes. (accessed: 01.05.2020).

[2] Michele Gazzea et al. "Automated Satellite-based Assessment of Hurricane Impacts on Roadways". In: *IEEE Transactions on Industrial Informatics* (2021), pp. 1–1. DOI: 10.1109/TII.2021.3082906.

[3] Mingyang Chen et al. "Developing City-Wide Hurricane Impact Maps using Real-Life Data on Infrastructure, Vegetation and Weather". In: *Transportation Research Record Journal of the Transportation Research Board* 2675 (Dec. 2020). DOI: 10.1177/0361198120972714.

[4] M. Gazzea et al. "Automated Power Lines Vegetation Monitoring using High-Resolution Satellite Imagery". In: *IEEE Transactions on Power Delivery* (2021), pp. 1–1. DOI: 10.1109/TPWRD.2021.3059307.

[5] Michele Gazzea et al. "Post-Hurricanes Roadway Closure Detection using Satellite Imagery and Semi-Supervised Ensemble Learning". In: *Transportation Research Board 100th Annual MeetingTransportation Research Board* (2021).

[6] Alican Karaer et al. "Leveraging Remote Sensing Indices for Hurricane-induced Vegetative Debris Assessment: A GIS-based Case Study for Hurricane

Michael". In: *Transportation Research Board 100th Annual MeetingTransportation Research Board* (2021).

[7] Ken Peffers et al. *Design Science Research Process: A Model for Producing and Presenting Information Systems Research.* 2020. arXiv: `2006.02763 [cs.SE]`.

[8] Yoshitaka Kumagai, John Bliss, and Steven Daniels. "Research on Causal Attribution of Wildfire: An Exploratory Multiple-Methods Approach". In: *SSWA Faculty Publications* 17 (Feb. 2004). DOI: `10.1080/08941920490261249`.

[9] Tatjana Dokic, Po-Chen Chen, and M Kezunovic. "Risk Analysis for Assessment of Vegetation Impact on Outages in Electric Power Systems". In: Oct. 2016.

[10] Aamir Malik, Likun Xia, and Nadia Ashikin. "Vegetation encroachment monitoring for transmission lines right-of-ways: A survey". In: *Electric Power Systems Research* 95 (Feb. 2013), pp. 339–352. DOI: `10.1016/j.epsr.2012.07.015`.

[11] Zisis Petrou et al. "Estimation of vegetation height through satellite image texture analysis". In: vol. XXXIX-B8. Aug. 2012. DOI: `10.5194/isprsarchives-XXXIX-B8-321-2012`.

[12] Leena Matikainen et al. "Remote sensing methods for power line corridor surveys". English. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 119 (Sept. 2016), pp. 10–31. ISSN: 0924-2716. DOI: `10.1016/j.isprsjprs.2016.04.011`.

[13] Dooahn Kwak et al. “Detection of individual trees and estimation of tree height using LiDAR data”. In: *Journal of Forest Research* 12 (Jan. 2007), pp. 425–434. DOI: 10.1007/s10310-007-0041-9.

[14] J. Kim and J.-P Muller. “3D reconstruction from very high resolution satellite stereo and its application to object identification”. In: *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 34 (Jan. 2002).

[15] S. Solberg, S. Puliti, and Osama Youssif. “Kartlegging av skogskader og hogst langs kraftlinjer”. In: 2018.

[16] D Poli, Fabio Remondino, and C Dolci. “Use of satellite imagery for DEM extraction, landscape modeling and GIS application”. In: (May 2012).

[17] Lake Singh et al. “Low cost satellite constellations for nearly continuous global coverage”. In: *Nature Communications* 11 (Jan. 2020). DOI: 10.1038/s41467-019-13865-0.

[18] Zhilin Pan. “Urban Vegetation Type Analysis Method Based on High Resolution Satellite Images”. In: Nov. 2016, pp. 613–616. DOI: 10.1109/ICSCSE.2016.0165.

[19] Robert McLaughlin. “Extracting Transmission Lines From Airborne LIDAR Data”. In: *Geoscience and Remote Sensing Letters, IEEE* 3 (May 2006), pp. 222–226. DOI: 10.1109/LGRS.2005.863390.

[20] Conor Mcmahon. “Remote sensing pipeline for tree segmentation and classification in a mixed softwood and hardwood system”. In: *PeerJ* 6 (Feb. 2019), e5837. DOI: 10.7717/peerj.5837.

[21] E Dmitriev et al. "Automatic detection of constructions using binary image segmentation algorithms". In: *Information Technology and Nanotechnology* (Jan. 2019), pp. 264–268. DOI: 10.18287/1613-0073-2019-2391-264-268.

[22] Tara H. Abraham. "(Physio)logical circuits: The intellectual origins of the McCulloch–Pitts neural networks". In: *Journal of the History of the Behavioral Sciences* 38.1 (2002), pp. 3–25. DOI: https://doi.org/10.1002/jhbs.1094. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jhbs.1094. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/jhbs.1094.

[23] Stefan Maetschke et al. *Understanding in Artificial Intelligence*. 2021. arXiv: 2101.06573 [cs.AI].

[24] Zewen Li et al. *A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects*. 2020. arXiv: 2004.02806 [cs.CV].

[25] Hiromu Yakura et al. "Malware Analysis of Imaged Binary Samples by Convolutional Neural Network with Attention Mechanism". In: Mar. 2018, pp. 127–134. DOI: 10.1145/3176258.3176335.

[26] Harsh Panwar et al. "A Deep Learning and Grad-CAM based Color Visualization Approach for Fast Detection of COVID-19 Cases using Chest X-ray and CT-Scan Images". In: *Chaos Solitons Fractals* 110190 (Aug. 2020), p. 110190. DOI: 10.1016/j.chaos.2020.110190.

[27] Petra Bosilj et al. "Transfer learning between crop types for semantic segmentation of crops versus weeds in precision agriculture". In: *Journal of Field Robotics* (Mar. 2019). DOI: 10.1002/rob.21869.

[28] Chigozie Nwankpa et al. *Activation Functions: Comparison of trends in Practice and Research for Deep Learning*. 2018. arXiv: 1811.03378 [cs.LG].

[29] Diederik P. Kingma and Jimmy Ba. *Adam: A Method for Stochastic Optimization*. 2017. arXiv: 1412.6980 [cs.LG].

[30] Carole H. Sudre et al. "Generalised Dice Overlap as a Deep Learning Loss Function for Highly Unbalanced Segmentations". In: *Lecture Notes in Computer Science* (2017), pp. 240–248. ISSN: 1611-3349. DOI: 10.1007/978-3-319-67558-9_28. URL: http://dx.doi.org/10.1007/978-3-319-67558-9_28.

[31] Geoffrey E. Hinton et al. *Improving neural networks by preventing co-adaptation of feature detectors*. 2012. arXiv: 1207.0580 [cs.NE].

[32] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. arXiv: 1505.04597 [cs.CV].

[33] Zongwei Zhou et al. *UNet++: A Nested U-Net Architecture for Medical Image Segmentation*. 2018. arXiv: 1807.10165 [cs.CV].

[34] Ozan Oktay et al. *Attention U-Net: Learning Where to Look for the Pancreas*. 2018. arXiv: 1804.03999 [cs.CV].

[35] Tsung-Yi Lin et al. *Feature Pyramid Networks for Object Detection*. 2017. arXiv: 1612.03144 [cs.CV].

[36] Ping-Yang Chen et al. *Residual Bi-Fusion Feature Pyramid Network for Accurate Single-shot Object Detection*. 2019. arXiv: 1911.12051 [cs.CV].

[37] Kaiming He et al. *Deep Residual Learning for Image Recognition.* 2015. arXiv: 1512.03385 [cs.CV].

[38] Md Zahangir Alom et al. *Recurrent Residual Convolutional Neural Network based on U-Net (R2U-Net) for Medical Image Segmentation.* 2018. arXiv: 1802.06955 [cs.CV].

[39] Wei Wang et al. *Recurrent U-Net for Resource-Constrained Segmentation.* 2019. arXiv: 1906.04913 [cs.CV].

[40] Michael A. Wulder et al. "High Spatial Resolution Remotely Sensed Data for Ecosystem Characterization". In: *BioScience* 54.6 (June 2004), pp. 511–521. ISSN: 0006-3568. DOI: 10.1641/0006-3568(2004)054[0511:HSRRSD]2.0.CO;2. eprint: https://academic.oup.com/bioscience/article-pdf/54/6/511/26895719/54-6-511.pdf. URL: https://doi.org/10.1641/0006-3568(2004)054[0511:HSRRSD]2.0.CO;2.

[41] G. Jansson and P. Angelstam. "Threshold levels of habitat composition for the presence of the long-tailed tit (Aegithalos caudatus) in a boreal landscape". In: *Landscape Ecology* 14 (2004), pp. 283–290.

[42] A. Barbati, P. Corona, and M. Marchetti. "A forest typology for monitoring sustainable forest management: The case of European Forest Types". In: *Plant Biosystems - An International Journal Dealing with all Aspects of Plant Biology* 141 (2007), pp. 103–93.

[43] Alex M. Lechner, Giles M. Foody, and Doreen S. Boyd. "Applications in Remote Sensing to Forest Ecology and Management". In: *One Earth* 2.5 (2020), pp. 405–412. ISSN: 2590-3322. DOI: https://doi.org/10.1016/j.oneear.

2020.05.001. URL: https://www.sciencedirect.com/science/article/pii/S2590332220302062.

[44] Ryan Klein et al. “Risk Assessment and Risk Perception of Trees: A Review of Literature Relating to Arboriculture and Urban Forestry”. In: *Arboriculture & Urban Forestry* 45 (Jan. 2019), pp. 26–38. DOI: 10.48044/jauf.2019.003.

[45] G. Meera Gandhi et al. “Ndvi: Vegetation Change Detection Using Remote Sensing and Gis – A Case Study of Vellore District”. In: *Procedia Computer Science* 57 (2015). 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015), pp. 1199–1210. ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2015.07.415. URL: https://www.sciencedirect.com/science/article/pii/S1877050915019444.

[46] Saining Xie et al. *Aggregated Residual Transformations for Deep Neural Networks*. 2017. arXiv: 1611.05431 [cs.CV].