Bachelor's Degree Project

Applying Machine Learning to detect historical remains in Swedish forestry using LIDAR data

Tillämpning av maskininlärning för att upptäcka historiska lämningar inom svenskt skogsbruk med hjälp av LIDAR-data

Author: Ruslan Abdulin
Supervisor: prof. Welf Löwe
Semester: VT/HT 2021
Subject: Computer Science

Abstract

Historical remains in Swedish forests are at risk of being damaged by heavy machinery during regular soil preparation, scarification, and regeneration activities. The reason for this is that the exact locations of these remains are often unknown or their records are inaccurate. Some of the most vulnerable historical remains are the traces left after years of charcoal production. In this thesis, we design and implement a computer vision artificial intelligent model capable of identifying these traces using two accessible visualizations of Light Detection and Ranging (LIDAR) data. The model we used was the ResNet34 Convolutional Neural Network pre-trained on the ImageNet dataset. The model took advantage of the image segmentation approach and required only a small number of annotations distributed on original images for training. During the process of data preparation, the original images were heavily augmented, which bolstered the training dataset. Results showed that the model can detect charcoal burners sites and mark them on both types of LIDAR visualizations. Being implemented on modern frameworks and featured with state-of-art machine learning techniques, the model may reduce the costs of surveys of this type of historical remains and thereby help save cultural heritage.

Keywords:
machine learning, neural networks, artificial intelligence, LIDAR, Swedish forestry, image segmentation, historical remains

Contents

1 Introduction

This thesis proposes a method for detecting historical remains in Swedish forests that is based on machine learning (ML) and uses Light Detection and Ranging (LIDAR) data.

Detecting historical remains protects cultural heritage from being destroyed by heavy machinery performing routine maintenance in the woods. Applying ML-algorithms to that detection and survey will save costs and conserve the manpower of archaeology experts.

The practical part of this thesis was performed at Softwerk AB in collaboration with the Archaeology Department of Linnaeus University and the Södra forest industry group.

1.1 Background

Sweden is mostly covered by forests [1]. One of the issues in Swedish forestry is damaging historical remains when heavy machinery is used for scarification, soil preparation, and regeneration felling activities. In a recently held nationwide survey performed by The Swedish Forestry Agency (SKS), it was stated that 4 out of 10 cultural heritage sites in the forest were damaged by regeneration felling activities [2]. Moreover, serious damage was caused by soil scarification, forest machines, logging residue, and wind-felled trees [3].

The Register of Ancient Monuments (FMIS) in Sweden has a database of all the recorded ancient remains in the country [4]. Even though the database provides information of about 1.7 million items, it does not show a complete picture of all the historical remains to be found in Sweden [4]. It has been estimated by SKS that 75 % of all ancient remains in Sweden are located in the forest and that 75 % of the forest area is not fully surveyed [5]. Performing the surveys is expensive as fieldwork time is costly and requires archaeology experts to be involved [6].

One type of historical remains to be found in Sweden is charcoal burner sites, known in Swedish as *Kolbotten*. These constitute the most common type of historical remains situated in the forest, which makes them also the most frequently damaged type [7]. The charcoal fuel industry has been vital for Swedish iron production from the Middle Ages all the way up to the 19th century, when blast furnaces and railroads came into use [7]. Despite the large number of charcoal-related historical remains, knowledge of Swedish charcoal production is still rather limited [7].

Charcoal burner sites were not systematically registered during two rounds of historical remains inventorying. Instead they were mostly transferred into FMIS from isolated surveys or forest history project's records [7]. Therefore, identifying the locations of charcoal burner sites is crucial for their preservation and availability as an archaeological source [7]. The charcoal burner sites create an important link between scientific archeological research and local cultural history, which benefits the promotion of cultural heritage [7].

As long as maps, images, and other geographic information (geodata) of Swedish landscape including LIDAR telemetry, are publicly available, [8] it seems possible to apply artificial intelligence (AI) to the information and create an algorithm for finding these historical remains. The existence of such an algorithm may benefit both Swedish forestry and Swedish cultural heritage.

1.2 Related Work

Attempts to locate different historical remains using AI have been made before.

In 2016, scientists from the Norwegian Computer Center represented their approach to detecting charcoal kilns in Norway using Convolutional Neural Networks for feature extraction and then applying the features to ML classifiers. Their approach aimed to improve the previously used Semi-Automatic technique, which is based on pattern recognition. One limitation of their approach was the lack of the exact locations of the kilns. The reason for this limitation was the use of a linear classifier, which does not show an exact position but only says whether the kilns are present on an image. However, their approach showed high prediction accuracy, allowing for the detection of 84.5 % of the known kilns in the training dataset [9].

In 2020, a group of scientists also applied ML algorithms to LIDAR data in order to reveal ancient artifacts. They used LIDAR data images from the ancient city of Angamuco, Mexico, for which archaeologists visually identified and mapped local features. Those images were fed to deep-learning classifiers in order to train a labeling model. As the result, they trained more than 30 classifiers and showed promising accuracy figures. However, their approach required manual annotations of the objects. These annotations have been collected during several surveys by multiple archaeologists on the order of years [10].

Another group of scientists applied multitemporal ML to remote sensing big data for the detection of archaeological mounds in Pakistan. They used a classifier algorithm against a large-scale collection of synthetic-aperture radar and multi-spectral images, resulting in an accurate probability map for mound-like signatures. The results showed that the area contains many more archaeological mounds than previously found. To achieve their goals they used publicly available multi-spectral satellite images. Even though this information is publicly available for non-commercial use, the coverage of the data is poor while temporal and spatial resolution is limited [11].

Finally, a group of scientists from Slovakia used a dataset from the Pacunam LIDAR Initiative survey of the lowland Maya region in Guatemala to identify Mayan structures using ML models. They explored and compared two models, U-Net and Mask R-CNN, for semantic segmentation. In the study, they proved that a U-Net-based model performed better and was capable of correctly identifying 60–66% of all objects. To achieve these results, they also used manually labeled annotations made by archaeology experts, which can be considered as a limitation of their work since it requires the presence of archaeology experts to apply this approach to another domain [12].

1.3 Target Group

This thesis is aimed mostly at data scientists and computer scientists studying machine learning, as well as at geologists, archaeologists, and other scientists interested in geodata analysis. It may also be of use to companies responsible for forest maintenance and representatives of government regulation agencies whose concerns lie in saving Swedish cultural heritage.

1.4 Problem Formulation and Motivation

Finding historical remains on LIDAR data is a tedious routine task yet it requires a certain level of knowledge in archaeology. As a result, markings of these objects in geoinformation systems (GIS) appear to be incomplete, and updating this information is expensive [6][4]. Related works show that computer vision leveraged by modern AI algorithms can detect historical remains using LIDAR data. However, there seems to be a knowledge gap on applying computer vision-based AI to different visual representations of LIDAR telemetry available for Sweden. Moreover, related studies use custom ML architectures and pay significant attention to either annotation and data preparations [10][12] or feature extraction [9]. If we could design a similar solution based on a publicly available, cutting-edge, off-the-shelf model trained on little, sparse, and biased data, and if that solution could take advantage of different types of available LIDAR visual representations, it would potentially help to reduce the costs of surveys and save Swedish historical remains from being destroyed by adding them to GIS and FMIS.

Even though there have been attempts to identify historical remains using Machine Learning [10][11][12] it seems that the area of detection of historical remains specific to the Nordic region is left uncovered in the Swedish landscape.

This research aims to advance the application of state-of-the-art and off-the-shelf ML models in both archaeology and geodata analysis. The results intend to address two important areas: automatic detection of historical remains using geodata and saving human resources from doing field surveys.

This research also aims to prevent historical remains from being destroyed. Finding and marking them in GIS may help parties responsible for forest maintenance be aware of their existence and take measures to avoid potential damage.

Finally, finding a practically functional and efficient algorithm for historical remains detection can serve as proof of concept for a bigger project in digitizing Swedish cultural heritage, which in turn may potentially have an impact on multi-disciplinary research in computer science, forestry and archaeology. To address the identified problem the following research questions must be answered:

1. Can publicly available modern ML models be used to accurately perform marking of historical remains when only single-point coordinates are available as annotations for training?

2. Can publicly available ML models detect specific historical remains (charcoal production sites) using LIDAR geodata from Sweden while only little, sparse, and biased training data is available?

3. Are there significant differences in historical remains detection over different types of LIDAR geodata visualizations ?

.

1.5 Scope/Limitation

The scope of the thesis has the following restrictions:

- Only standard computer vision neural networks are researched. The focus is on CNN U-Net architecture, which turned out to be successful in image segmentation tasks [12].
- Only two types of high-resolution LIDAR-data are used as geodata.
- For the dataset, the geodata of the Attsjö region in Sweden is used.
- Historical remains are restricted to only one type, which is known to be found in these geographical areas. This type is known in Sweden as *kolbotten*—a mark left after years of charcoal production [7].

1.6 Outline

This thesis is organized into the following chapters. In Chapter 2 we discuss the methodology and tools to be applied to answering the research questions. In Chapter 3, we introduce the required theoretical background knowledge in the field of computer science, ML and archaeology. In Chapter 4, we discuss design and implementation. Chapter 5 is dedicated to applying the model on testing data and the results of that application. In Chapter 6 we perform analysis of those results. In Chapter 7, we discuss the results and report whether the research questions were answered. In Chapter 8, we make a conclusion and suggest potential future work.

2 Methods

In order to answer the research question and cover the identified knowledge gap, we have created a research project. This chapter describes the research project as well as the research methods chosen and the motivation behind that choices.

2.1 Research Project

The Södra forest industry group, in collaboration with Softwerk AB and The Department of Archaeology at Linnaeus University initiated this research project. Södra was interested in an AI model that could help to identify historical remains in Swedish Forests using the sparse available geodata. In order to prove its reliability, the model should show at least 50% accuracy in these findings. To develop such a model, we adopted the design science approach. In this chapter, we describe the details of that approach as well as reasons behind selecting it. To comply with the research questions, the model should be publicly available, state-of-the-art, and working out of the box without the need to customize it.

2.2 Research Methods

One of the most comprehensive articles dedicated to design science [13] highlights the following guidelines:

- Design as an artifact: Design-science research aims to produce an artifact in the form of a model, a method, or a construction. According to the article, the artifact is an innovation that defines ideas, practices, or technical capabilities [13].
- Problem relevance: The objective of design-science research is to acquire knowledge and apply it to important but yet unsolved relevant business problems [13].
- Design evaluation: Developed artifacts should be evaluated on constant basis in terms of functionality, completeness, consistency, performance, reliability, fit with the organization, or other relevant quality attributes[13].
- Research contributions: Effective design-science research ought to provide clear contributions to the areas of the design artifact, including construction or evaluation knowledge [13].
- Research rigor: Application of rigorous methods in both the construction and evaluation of the design artifact is vital for design-science research [13].
- Design as a search process: The search for an optimised and efficient solution to reach objectives is an essential part of the design process [13].
- Communication of research: The results of the research must be presented not only to technology-oriented audiences but to management-oriented audiences as well. It is also crucial for the audiences to understand the processes which the artifact's construction and evaluation rely on [13].

Bearing these guidelines in mind, we planned to start by defining relevant problems as well as objectives for a solution. After that, we designed and implemented an AI model

as an artifact. During the implementation, we searched for an optimized and efficient way for the model to reach the objectives. Finally, we demonstrated the efficiency of that model by conducting a careful evaluation procedure against the data the company provided. The evaluation was performed together with technical and management stakeholders at the company.

Figure 2.1 illustrates the steps for following the chosen method. All the steps except *define goals* and *define objectives* are iterative and must be repeated until the validation shows that all the objectives are met.

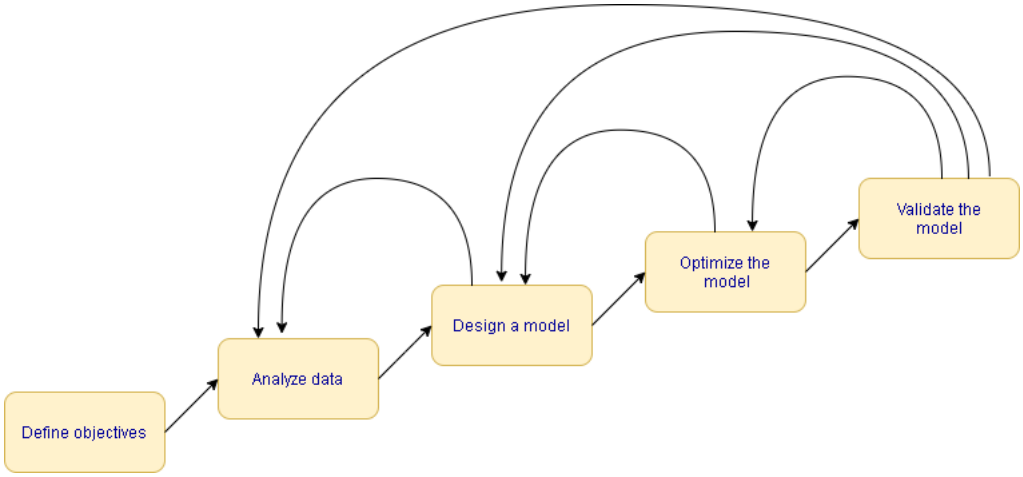


Figure 2.1: Iterative steps of the design science method.

Although the search for a working design in design science is an iterative process [13] the problem remains unchanged. Therefore, our objectives are the research questions formulated in the Problem Formulation and Motivation section.

2.2.1 Data analysis

Before we start searching for an appropriate model, it is important to understand what the data we are dealing with look like. Depending on the type of data, we should either select a model which can take that type as an input or we should change the data so the chosen model can process it. We should also consider the amount of data we receive as it also may affect the choice of the model. Being iterative, the analysis might be carried out many times for different models and/or optimizations.

2.2.2 Model design and optimization

Once the data is analyzed and prepared, we can start looking for a model that can process the data. We consider modern state-of-the-art AI models as a backbone for further optimizations. If a model shows results somehow different from a random distribution, we try to optimize the model. We will use related works [10][11][12] as guidelines for searching, optimization, and data preparation. If a model cannot reach the objectives after a considerable amount of efforts, we decline the model and continue the search. Picture 2.2 shows the model search process. The search is iterative and bounces in between studying related papers and trying to apply a similar approach in practice.

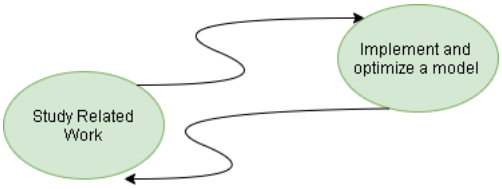


Figure 2.2: Model search and optimization.

2.2.3 Model validation

To understand the effectiveness of a model a robust and trustworthy validation will be applied. To validate the model, we will use the original geodata, for which we know what is true and false. Then we test the model against the geodata and derive the efficiency as metrics. Comparing the metrics for different models we should be able to say which one works better and how the optimizations helped to improve the model.

2.3 Reliability and Validity

As long as we plan to use a modern off-the-shelf model, we assume it to be reliable out of the box. However, analyzing and preparing geodata for that model can introduce errors and inaccuracy. Moreover, the validation process will be relying on the same data analysis hence can be vulnerable for the same errors. In order to mitigate errors accumulation, we plan to implement data visualization at the very early stage of the data analysis. Additionally, we plan to implement and run sanity checks on a constant basis. Finally, we will run automatic unit tests against the most critical parts of the implementation.

2.4 Ethical Considerations

It is worth mentioning that even though the geodata of Sweden is considered to be public, the annotation dataset used to conduct this report is private property which belongs to the Södra Company and is protected by copyright. This means that this thesis cannot reveal the protected data but only the results, from which the original data cannot be derived.

3 Theoretical Background

In order to understand the matters described further in this paper, the related theoretical background must be established. This chapter provides a brief overview of the concepts and matters this thesis is dealing with.

3.1 Computer Science

This thesis aims to advance the application of ML. Hence, knowledge of the basics of ML, as well as techniques for preparing data for ML, are required for further reading. Additionally, basic knowledge of LIDAR technology is also needed for clear understanding of the concepts described in the thesis.

3.1.1 Machine learning

ML is becoming more and more common in computer science. It has been proven to solve a wide range of problems not only in computer vision [14], but also in other fields, such as data classification, knowledge extraction, or even speech recognition [15].

In general, ML is a process of optimizing a model. A model is a set of step-by-step instruction defined up to some parameters, which aims to solve a given task. Being an algorithm, a model can transform given input into desirable output with some level of efficiency. In order to increase the efficiency of the model, its parameters need to be optimized. The optimization process is called *training* and it is performed against relevant training data. The training process runs in steps called *epochs*. After each epoch, the model can be validated in order to keep track of the training process. The validation is done by providing the model with data, which the model has not seen during the training. Analysing the validation outcome defines efficiency of the model [15].

Figure 3.3 represents a model with an algorithm, input and output data. The algorithm is defined with three parameters: param1, param2, param3. These are equivalent to X, Y and Z respectively. The reference data is marked as *reference* on the picture and represents known desirable results. The ML training then is finding such X, Y, and Z such that the output data is as close as possible to the reference data.

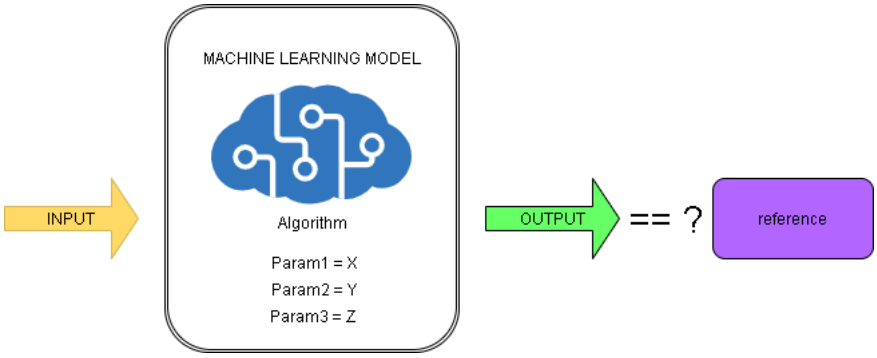


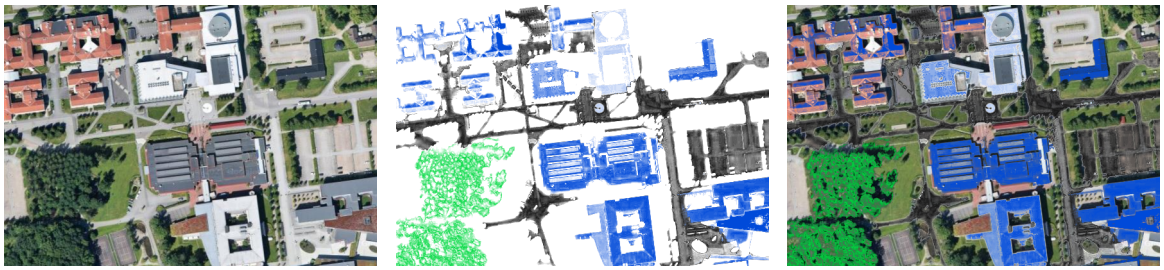
Figure 3.3: Machine learning

3.1.2 Image segmentation

Image segmentation is a process of dividing elements in an image into a set of classes so that all elements in each class share a common property [16]. Image segmentation can be visualized as a process of coloring elements on a picture. Applying image segmentation to an image means classifying every single pixel of the image into one or more classes which, in turn, will split the whole image into segments of different classes.

Image segmentation in ML differs in that the segmentation is performed by an ML model optimized by the ML process [17].

Figure 3.4 demonstrates image segmentation performed on a Google Maps area of Linnaeus University, Växjö. Image (a) is the original image, image (b) is a colored segmentation mask for the original image, and image (c) is an overlay of the original image with the segmentation mask.



(a) Original image (b) Segmentation mask (c) Overlay image

Figure 3.4: Image segmentation performed on a Google Maps image.

3.1.3 Transfer learning

Transfer learning (TL) has been attracting more and more attention in both research and industry for building ML models capable of solving problems using the data from related problems [18][19]. The key idea of TL is that a model pre-trained on a larger number of training data for a source problem can be reused to solve target problems with limited data available [19].

3.1.4 Image augmentation

In some cases, the original data can be sparse and insufficient for training a model even applying the TL technique. For those cases the dataset can be drastically extended using image augmentation. This augmentation is done by applying various transformations to the original images such as cropping, flipping, rotation, shifting, scaling, etc., making augmented images to be added to the original dataset. Those transformations as well as their various combinations can improve the robustness of the model to be trained [20].

Figure 3.5 shows an example of image augmentation performed on a Google Maps image. For example (a) is an original image, (b) is the original image vertically flipped, and (c) is the original image horizontally flipped.

(a) Original image (b) Vertical flip (c) Horizontal flip

Figure 3.5: Image augmentation performed on a Google Maps image.

3.1.5 Convolutional neural network

In recent years convolutional neural networks (CNN) have demonstrated great performance in visual recognition problems [21]. The CNN model is a class of ML models commonly applied to analyze images [22]. CNN were inspired by biological processes and are used for pattern recognition such as hand-writing analysis, face recognition [23], and many other visual recognition tasks [24]. Moreover, CNN models pre-trained on ImageNet image database can be successfully applied to image segmentation tasks [24].

3.1.6 LIDAR

In recent years, LIDAR technology has had a major impact on surface analysis in geomorphology [6]. LIDAR systems allow for measuring distance by sending a Laser beam to the object and measuring the time before the reflected beam comes back. Usually installed on a plane, LIDAR systems allow for scanning the ground beneath the aircraft to create accurate topographical maps. Their ability to detect objects even through massive vegetation has proven to be effective and practical for archaeologists [25].

3.2 Archaeology

Since this thesis touches matters outside of computer science, more precisely archaeological traces of historical remains, a bit of theoretical knowledge about types of historical remains to be found is also required for further reading.

Charcoal burners sites are locations where archaeological traces have been left in burners such as charcoal piles or temporary wooden kilns [7]. The traces can be rectangular or round depending on the type of the burner they used to serve. It is known that charcoal burners sites were used and reused, possibly for centuries [7]. In his research, Hesse shows that this type of historical remains can be identified using LIDAR data [26]. Figure 3.6 shows what the sites, marked with yellow circles, look like on LIDAR images.

3.2.1 Historical charcoal production

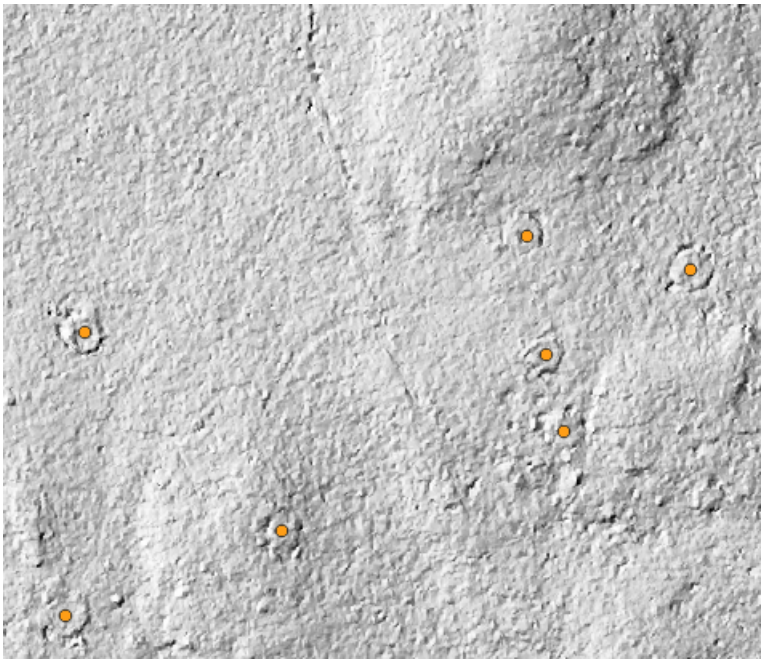


Figure 3.6: Charcoal production burning sites marked on a Lidar image

4 Research Project Design and Implementation

This chapter is about the design decision we adopted doing the research, the motivations behind the decisions and actual implementation of the design into a functional model.

4.1 Design

As was stated in 2.Methods chapter, finding the best design for that project is an iterative process. Here we describe the design decisions which appeared to work better than others for reaching objectives declared in 2.Methods. Also, we highlight decisions and techniques that had limited success but impacted the development of the design.

4.1.1 Data analysis and data preparation

The practical part of this thesis is based on processing data provided by Södra company. The format of the data is the following:

- High resolution geoinformation images of the Attsjö area given in GeoTiff format [27] representing LIDAR data. Each image has 256 x 256 pixels dimensions. The images come in two different types types, known as *hillshade* and *inclination*. Figure 4.7 shows examples of the two types available as geodata.

(a) Hillshade

(b) Inclination

Figure 4.7: Examples of the two types of geoinformational images.

- Annotation in CSV format [28] with coordinates of historical charcoal production sites. Each site is annotated as single point coordinates.

The fact that we are dealing with images forces us to search for a model that accesses images. On the other hand, we need to fulfill the output objective, which requires clear markings of the objects found. Those two factors limits us to one of two general approaches:

- Image segmentation: The output image must be segmented and the objects found are overlaid as a segmentation mask.
- Object detection: The output image must have a frame that clearly borders the objects found.

Due to the fact that the related studies use image segmentation as a general approach [10][11][12], we made a decision to keep it as a primary paradigm for our project as well. Since objects are annotated as a single point rather than an image segmentation mask, as was done in related studies [10][11][12], we decided to create annotation masks for the

images ourselves.

Annotation masks are 3x3 squares of pixels where the central pixel is exactly the single point coordinates provided by the annotation. Before we came to this design, we tried bigger square areas as well as round areas. We also tried to apply filters in order to derive better annotations, but practice showed that 3x3 squares work better for our case.

Annotation mask creation is a cornerstone of the project since all the future model optimizations and model validations are based on the combination of an original image plus a corresponding annotation mask for that image. Hence, it is vital that the annotation masks are created correctly. Keeping in mind that the historical charcoal production sites are mostly visible on LIDAR images[26], we perform regular sanity checks by visualizing the masks and performing visual analysis. This is done by overlaying a bunch of original images with corresponding masks and visually verifying that the masks are located in the middle of charcoal production sites on the images.

Figure 4.8 shows an example of an image overlaid by an annotation mask for a sanity check. It is clearly visible that the red 3x3 pixels mask sits in the middle of the U-shaped structure.

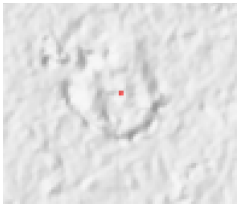


Figure 4.8: Annotation mask sanity check example.

Another important part of the dataset preparation is image augmentation. For better model training (optimization), we need a set of images that is significantly larger than what we have. To extend the set, we add randomly augmented copies of original images. By adding augmented images, we increase the diversity of unique images to be given to the model. We perform random combinations of the following augmentations:

- horizontal flip
- vertical flip
- rotation on a random angle in the range 0°– 360°

The motivation behind the choice of augmentations is based on practical experience during the design search.

In order to prepare a complete cycle of model training, we divide the available dataset into three parts: training, validation, and testing. We exclude the testing part from the dataset before expanding the dataset with augmentation. Both training and validation we use to train the model. Validation for each epoch of training involves randomly choosing 10% of the dataset, testing part excluded. Each validation part is used to validate the respective epoch of training. Testing part is used to evaluate the model after all the epochs of the training process. To keep the evaluation reliable, we do not show the testing part to the

model during the training.

Figure 4.9 illustrates an example of a dataset split for 3 epochs of training. The red circle represents the images excluded from the dataset before it was extended by augmentation. The blue part illustrates images available for training. The yellow zone shows images randomly chosen for epoch validation.

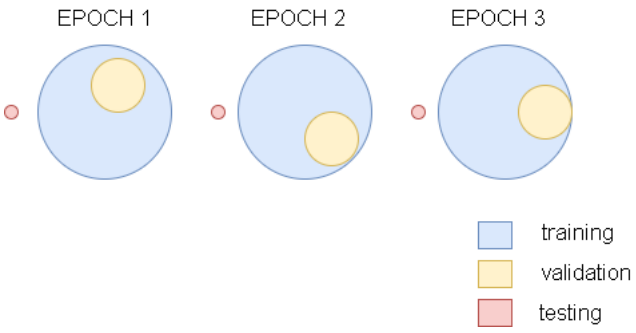


Figure 4.9: Example of a dataset split.

The split design is motivated by best practices in ML [29].

4.1.2 Model

The model to be trained is the key element in ML [15]. Related studies suggest using the U-Net type Convolutional neural network [30] as a model architecture [10][11][12]. During the search for an optimal design, we tried a number of U-Net models starting with a trained-from-scratch U-Net backbone and ending up with publicly available pre-trained models including but not limited to Xception [31], Inception [32], ResNet [33] and others. We found out that the ResNet models, which feature U-Net architecture and were pre-trained on the ImageNet dataset then post-trained according to TL technique performs better than others. ResNet models significantly improve performance for image segmentation tasks due to deep trainable structure [34].

4.1.3 Training

In ML the training variables that are related to the training process rather than to a model are called *hyperparameters* [15]. For our design, the following hyperparameters are also found to be the best:

- Number of epochs: 20
  Number of epochs is the number of iterations for the model to be trained on. During each epoch, the model goes through the whole dataset trying to optimize its parameters.
- Batch size: 32
  Batch size is the number of images to be trained on at once.
- Validation split: 0.1
  Validation split is the amount of randomly chosen images saved for current epoch's validation.

- Loss function: dice loss [35]
  Loss function is a function which tells us how well a model performs [36].
- Optimizer: Adaptive Moment Estimation (Adam) [37].
  Optimizer is a function that tweaks the model's parameters in response to the output of its loss function [37].

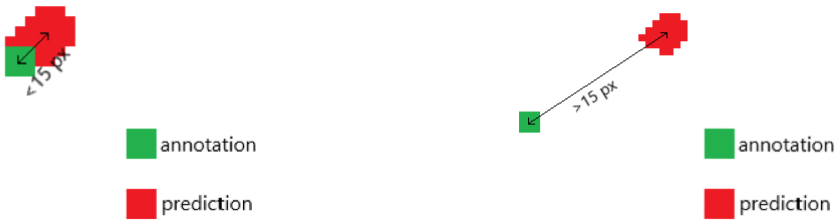
During the training, we show images from our dataset in batches to the model. This is absolutely vital for each image-mask pair to be shown to the model at the same time as the model learns by comparing an image with its mask. In order to make sure that the pairs are tied together, we run yet another sanity check by randomly choosing image-mask pairs and visually checking that when overlaid with the image, the mask highlights the areas we are interested in.

4.1.4 Validation

After being trained a model requires validation. One way to validate the model is to calculate its prediction accuracy. In order to perform validation, we use the original images and the mask we created for the images. Then we compare two masks for the same image—the mask the model returns as output and the mask we generated ourselves.

As we are dealing with a single point coordinates resulting in a single pixel, a standard approach of pixel by pixel comparison of given and predicted masks may not be correct. The reason for this is that the object on an image can be significantly bigger then the mask we created, so even though a prediction hits the object, it still may be very different from the mask and have no common pixels at all.

Because of this, we have chosen another way to calculate whether a prediction is correct. First, we group all the connected pixels on the predicted mask and call this a predicted object. Next, we calculate a central pixel of that object and measure the distance between the pixel and the center of the annotation. If the distance is within 15 pixels, we consider it to be a hit—if greater, it is a miss. Figure 4.10 shows examples of hit and miss in a prediction. 4.10a is a hit because the distance between centers of the annotation and the object is less than 15 pixels. 4.10b is a miss because centers of the predicted object and the ground truth annotation are more than 15 pixels apart.



(a) Hit (b) Miss

Figure 4.10: Hit and miss in a prediction mask.

Now we can apply a formula to calculate accuracy:

$$prediction\ accuracy = correctly\ predicted\ objects / all\ predicted\ objects \times 100\%$$

In order to derive additional metrics, we split all the annotated objects into 3 categories:

True-positive (TP)—existing objects, found by a model.

False-positive (FP)—not existing objects, wrongly predicted by a model.

False-negative (FN)—existing objects, not found by a model.

The additional metrics we were interested in are *Precision*, *Recall*, and *F-score*. Precision is a fraction of true-positive objects among all the predicted objects, recall is a fraction of true-positive objects among all the existing object, and F-score is a harmonic mean of those two.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{FP + FN}$$

$$\text{F-score} = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

4.2 Implementation

In order to answer, the research questions the design should be properly implemented. We use the Python programming language to implement the design because it performs well on ML tasks and is suitable for programmers of all levels [38]. The implementation is split into three parts: data, model, and validation.

4.2.1 Data

The cornerstone of data preparation is correct annotation mapping. We designed a function that takes in annotated coordinates, the smallest coordinates of an image, the biggest coordinates of the image, the smallest pixel on the image, and the biggest pixel on the image. The function then returns the exact pixel where the center of an annotation mask should be.

The annotation dataset consisted of 143 coordinates' annotations. Applying the function on 110 available images, we created 110 annotation masks for the images. We took the first 10 pairs (image + mask) and saved them as testing data. All the other images we use for training.

The dataset of 100 pairs we extended up to 2000 pairs using the image augmentation technique. To implement image augmentation, we used the ImageDataGenerator class of the Keras framework [39]. We applied the following random transformations: horizontal flip, vertical flip, and rotation to a random angle in range 0°—360°. We passed the same random seed to ImageDataGenerators to guarantee that random augmentations are applied simultaneously to images and masks.

4.2.2 Model

In order to leverage TL we used the segmentation models framework available on GitHub [40]. In the Model subsection of Design section, we motivated the use of ResNet models. For the implementation, we have taken the ResNet34 model pre-trained on the ImageNet dataset. The motivation behind using the model is the balanced ratio in between the speed of work and quality [41], which impacts training time and allows for bigger batch size. We used the dice-loss function from the framework since dice loss performs well learning from classes with lesser representation in an image [42]. Since the model was pre-trained on millions of general images (ImageNet dataset) [43] but we use it for LIDAR images only, this type of transfer learning can be considered to be parameter-based domain adaptation. We trained the model on real data for 20 epochs. We saved logs of the training and drew a plot using the matplotlib framework [44] to evaluate the training process.

4.2.3 Validation

Implementing the validation process is vital for model evaluation. Our model validation relies on accuracy. The key part in calculating accuracy is determining whether a predicted object is a hit or miss depending on how far it landed from the annotation mask.

The implementation of the distance measurement is rather difficult. We used the OpenCV computer vision library, which was created to analyze visual content [45]. With the help of the library we detected predicted objects, found their borders, and used them to detect the object's central point. Then we compared the distances between the objects and masks, finding closest objects for each mask and measuring the distance between central points. Finally, we used our "hit or miss" approach to decide whether the predicted object was correct.

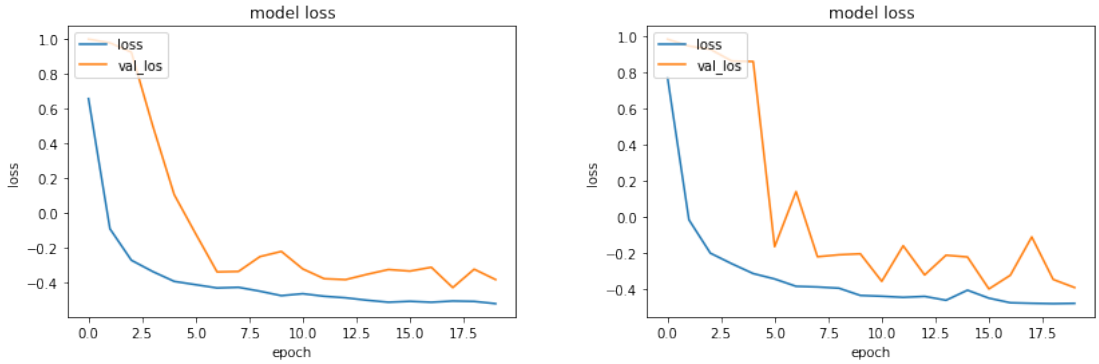
In order to visualize the results, we drew plots of 256x256 pixels representing a mask and put pixel coordinates for both known and predicted masks on the plots. For visualization purposes, we have also shown overlays of original images with predicted masks.

5 Results and Discussion

In this chapter, we present the results of the training process as well as the results of the model validations. We also perform analysis of the results and discuss our findings.

5.1 Training

To visualize the training process we used training logs and created plots representing loss over training epochs. Fig 5.11 shows two training plots for two models. We put two lines on the plot—blue is training loss and yellow is validation loss. Model (a) is training on the hillshade dataset and model (b) is training on the inclination dataset.



(a) hillshade (b) inclination

Figure 5.11: Training process.

In order to evaluate training error and training accuracy all the images of the training datasets were given to the models. Table 5.1 summarizes the prediction results. *Objects* in the table refers to the total number of objects in a corresponding dataset. *Predictions* describes the total number of predicted objects the model has found. *Correct* shows how many predictions were considered to be hits. *False* shows how many predictions were misses. *Missed* indicates how many existing objects the model could not find. The accuracy formula in section 4.1.4 was used to determine accuracy.

Table 5.1: Predictions—training dataset

	Objects	Predictions	Correct (TP)	False (FP)	Missed (FN)	Accuracy
Hillshade	125	124	107	17	18	86.2%
Inclination	125	224	105	119	20	46.8%

The additional metrics for the training dataset are presented in the table 5.2. These metrics were derived according to formulae in section 4.1.4 .

Table 5.2: Additional metrics—training dataset

	Precision	Recall	F-score
Hillshade	0.862	0.856	0.859
Inclination	0.468	0.84	0.601

5.2 Validation

Model validation is performed by giving the test hillshade images and test inclination images to the hillmodel and inclination models, respectively. We split this section into two parts: validation of the hillshade model and validation of the inclination model.

5.2.1 Hillshade model

Figure 5.12 shows the application of the test hillshade data to the hillshade model (model trained on hillshade dataset). For each test image, a pairs plot overlay was generated. The known annotation is marked with green on the plot whereas predictions made by the model are marked with red on both plot and overlay.

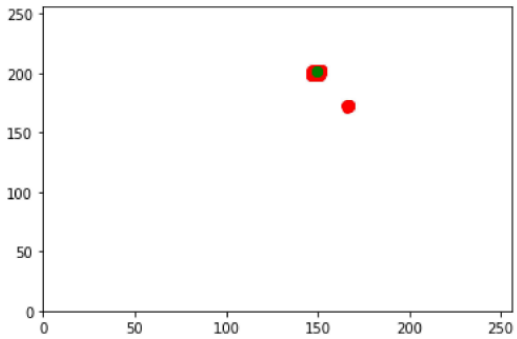


image 1 - plot

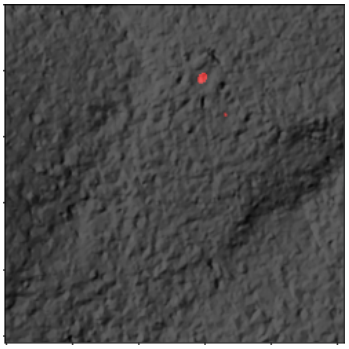


image 1 - overlay

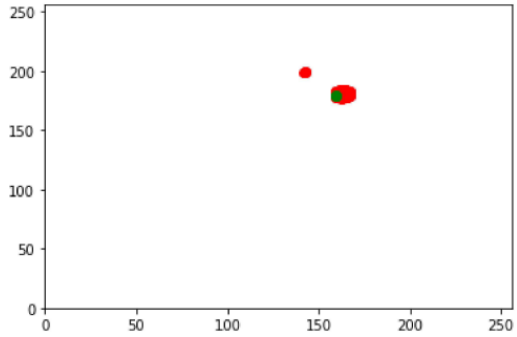


image 2 - plot

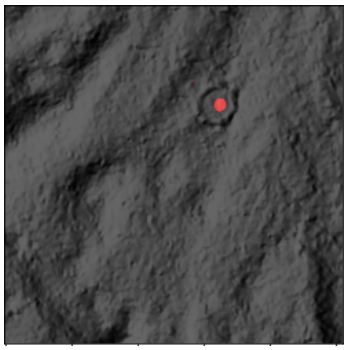


image 2 - overlay

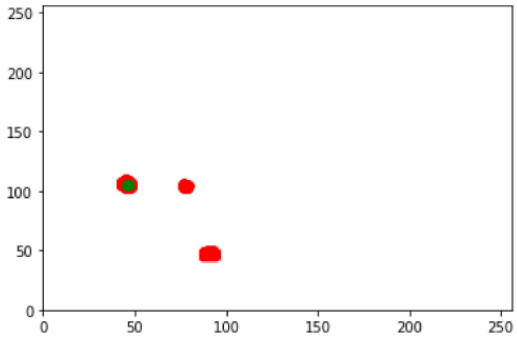


image 3 - plot

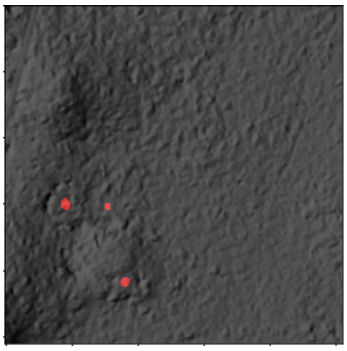


image 3 - overlay

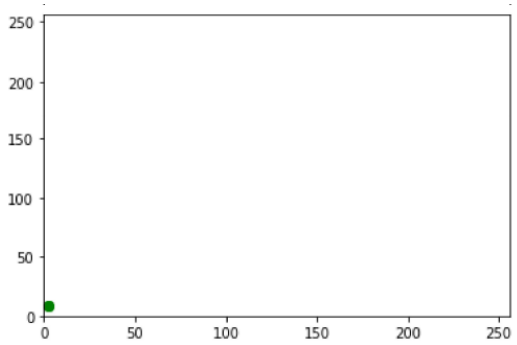


image 4 - plot

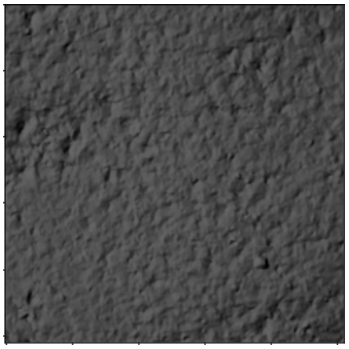


image 4 - overlay

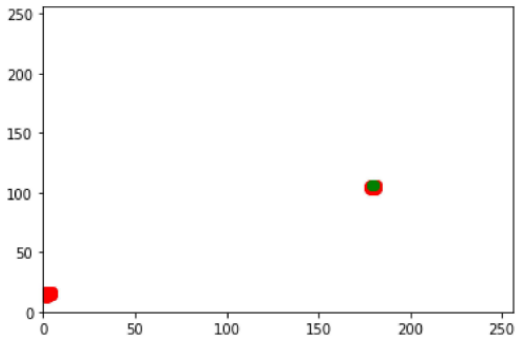


image 5 - plot

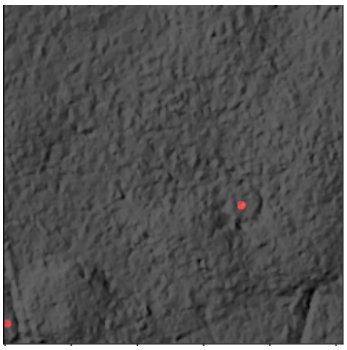


image 5 - overlay

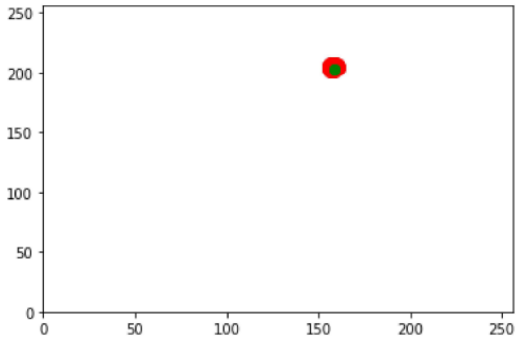


image 6 - plot

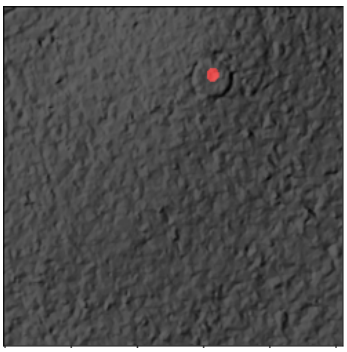


image 6 - overlay

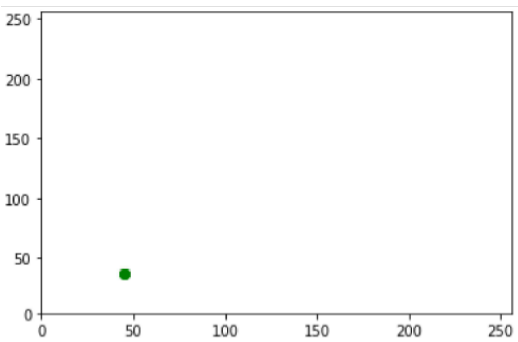


image 7 - plot

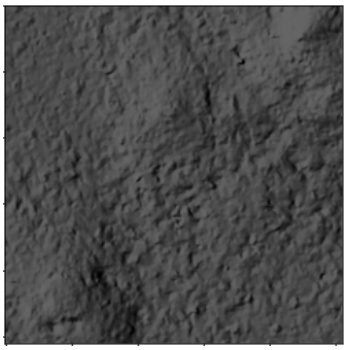


image 7 - overlay

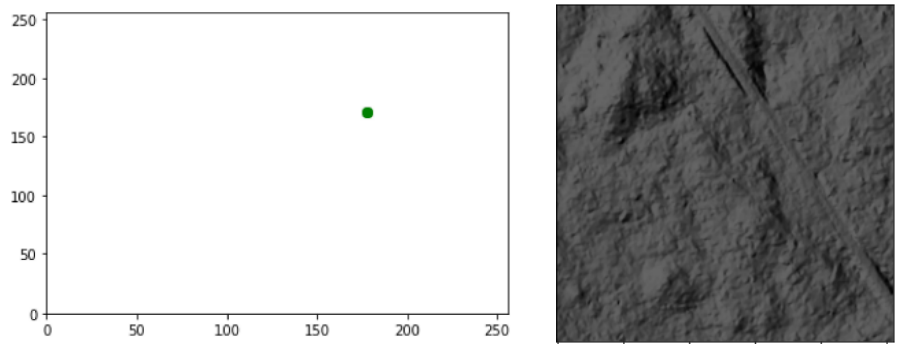


image 8 - plot | image 8 - overlay

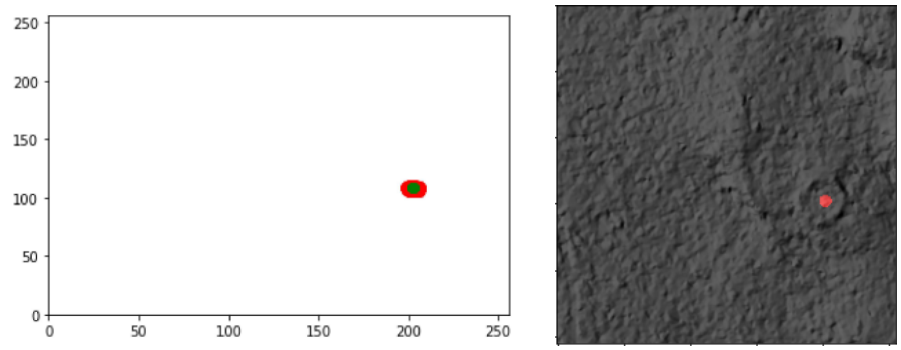


image 9 - plot | image 9 - overlay

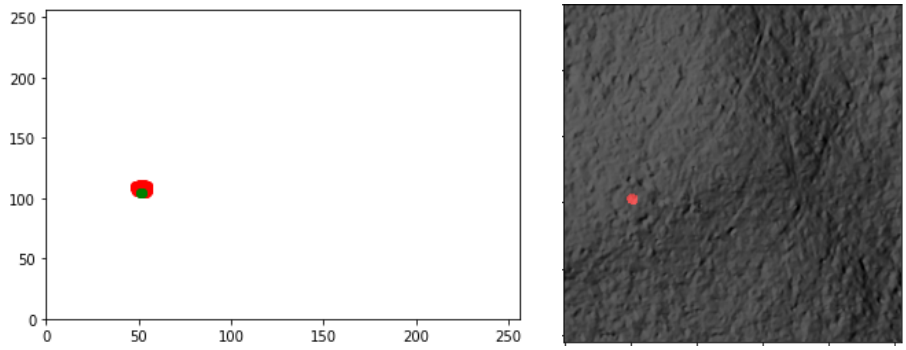


image 10 - plot | image 10 - overlay

Figure 5.12: Prediction results for hillshade test images.

5.2.2 Inclination model

Figure 5.13 shows the application of the test inclination data to the inclination model (model trained on inclination dataset). For each test image a pairs plot overlay was generated. The known annotation is marked with green on the plot whereas predictions made by the model are marked with red on both plot and overlay.

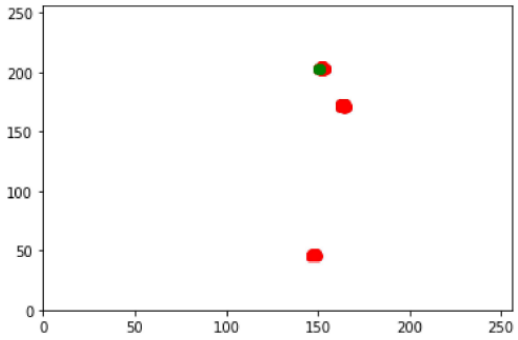


image 1 - plot

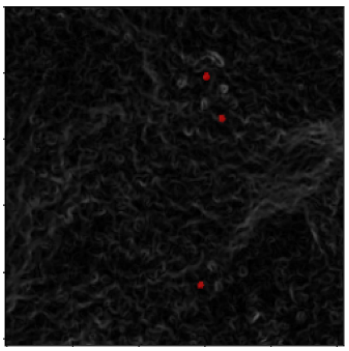


image 1 - overlay

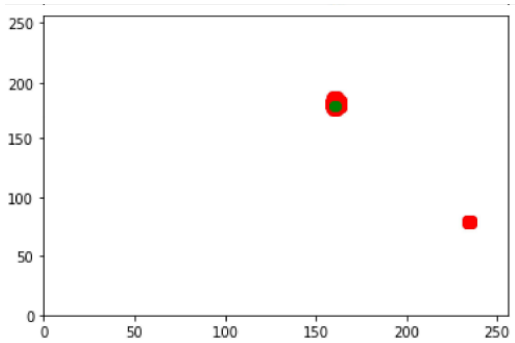


image 2 - plot

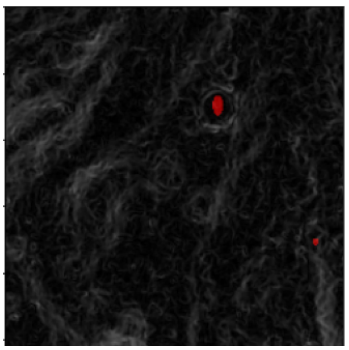


image 2 - overlay

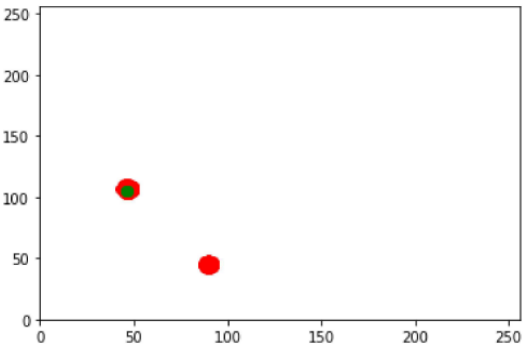


image 3 - plot

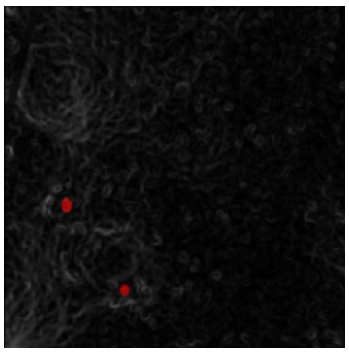


image 3 - overlay

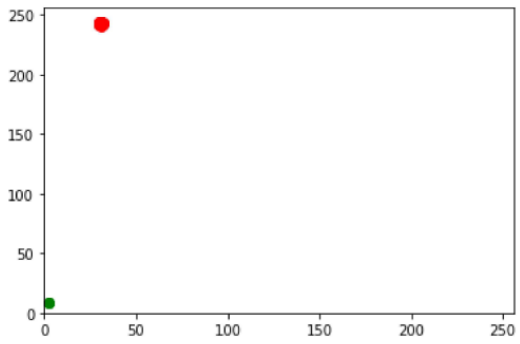


image 4 - plot

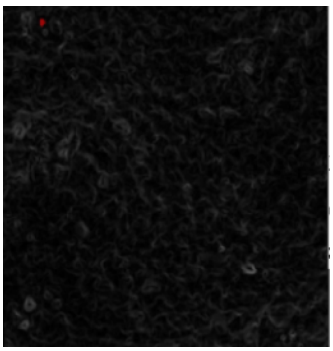


image 4 - overlay

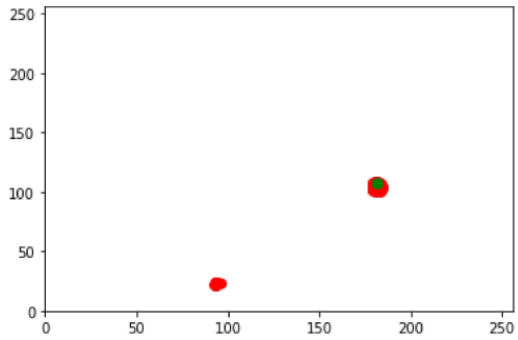


image 5 - plot

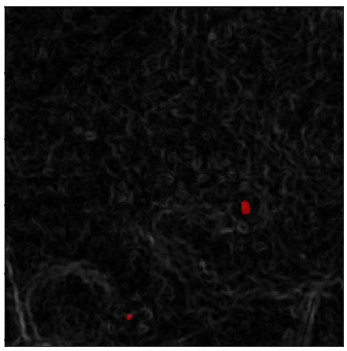


image 5 - overlay

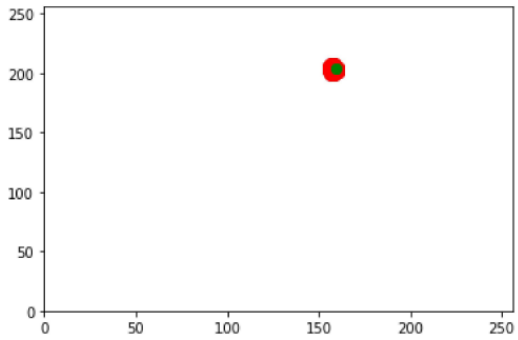


image 6 - plot

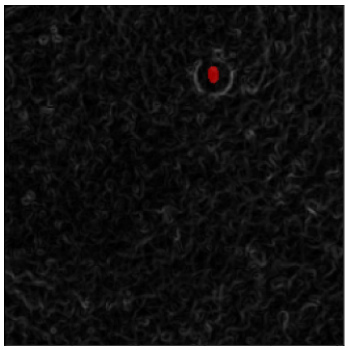


image 6 - overlay

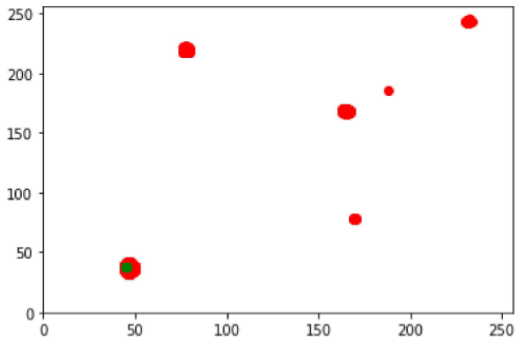


image 7 - plot

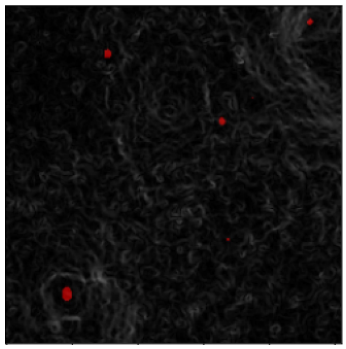


image 7 - overlay

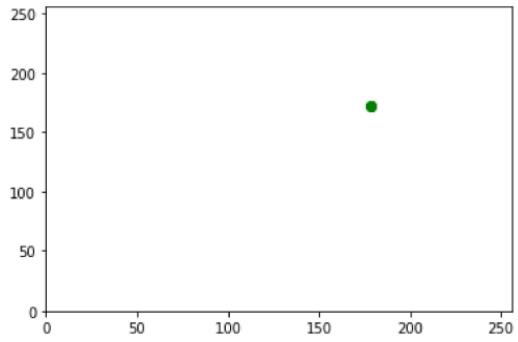


image 8 - plot

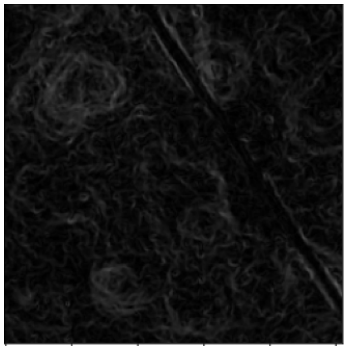


image 8 - overlay

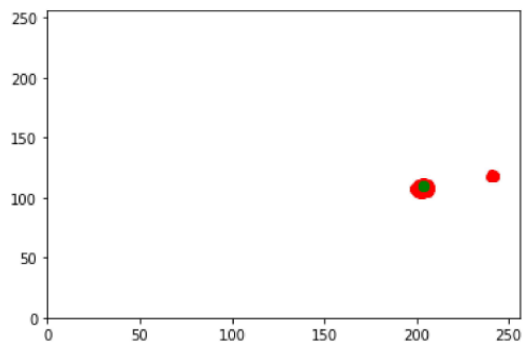


image 9 - plot

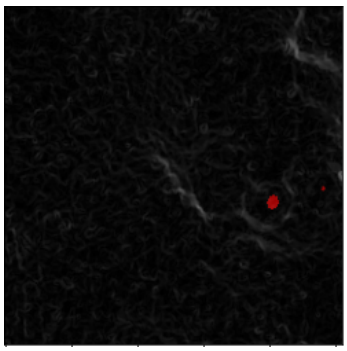


image 9 - overlay

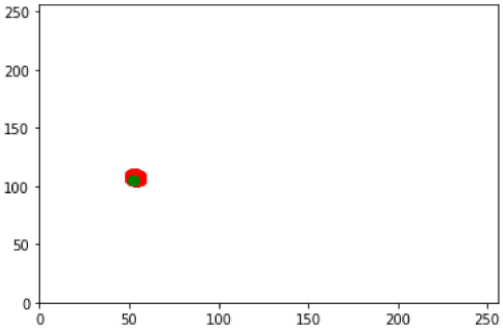


image 10 - plot

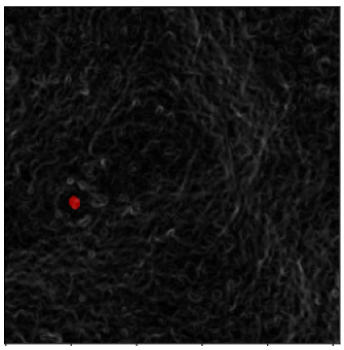


image 10 - overlay

Figure 5.13: Prediction results for inclination test images.

5.2.3 Statistics

The results for validation datasets are presented in the table 5.5. *Objects* in the table refers to the total number of objects in a corresponding dataset. *Predictions* describes the total number of predicted objects the model has found. *Correct* shows how many predictions were considered to be hits. *False* shows how many predictions were misses. *Missed* indicates how many existing objects the model could not find. The accuracy formula in section 4.1.4 was used to determine accuracy.

Table 5.5: Predictions—testing dataset

	Objects	Predictions	Correct (TP)	False (TN)	Missed (FN)	Accuracy
Hillshade	10	12	7	5	3	58.3%
Inclination	10	20	8	12	2	40%

The additional metrics for the testing dataset are presented in the table 5.6. These metrics were derived according to formulae in section 4.1.4 .

Table 5.6: Additional metrics—testing dataset

	Precision	Recall	F-score
Hillshade	0.583	0.7	0.636
Inclination	0.4	0.8	0.533

5.3 Analysis

To understand the results, we started by analyzing the training processes. Training graphs illustrated in figure 5.11 showed that both models had high initial loss values and the values dropped significantly in the beginning of training processes. After approximately 17 epochs, both loss graphs reached the state where they did not tend to go down much further. We also should notice that the validation loss graph for the inclination model showed bigger fluctuations than the corresponding graph for the hillshade model.

Table 5.5 shows that both models have made false predictions. Trying to understand whether there were similarities in the false predictions, we put testing images side by side for comparison. Figure 5.14 shows two images where both models were confused by the same objects.

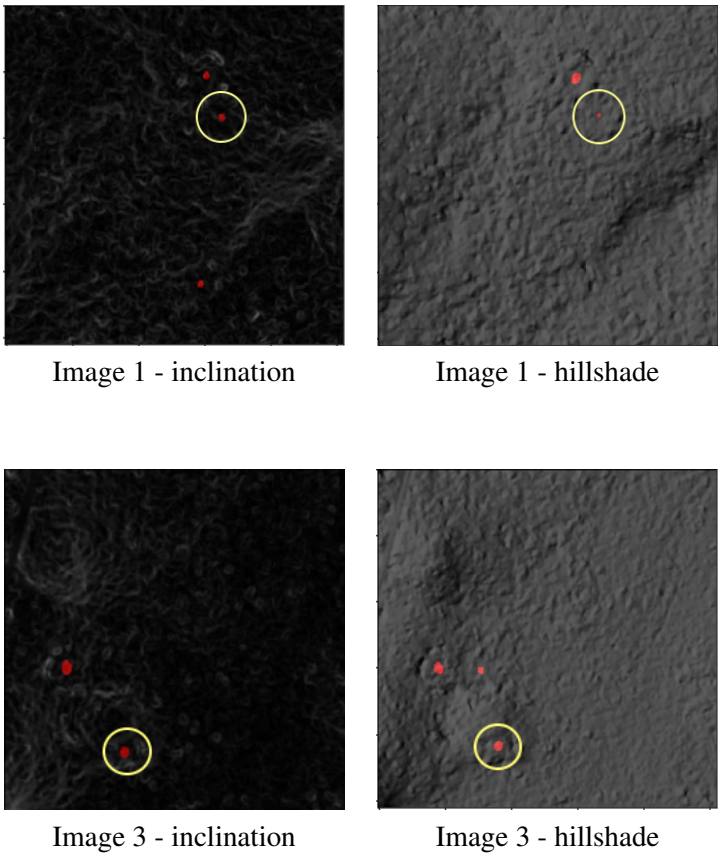


Image 1 - inclination | Image 1 - hillshade

Image 3 - inclination | Image 3 - hillshade

Figure 5.14: Prediction comparison.

During the side-by-side comparison, we found one case where the hillshade model could not detect an object but the inclination model clearly marked it in the left bottom corner. Figure 5.15 illustrates this case.

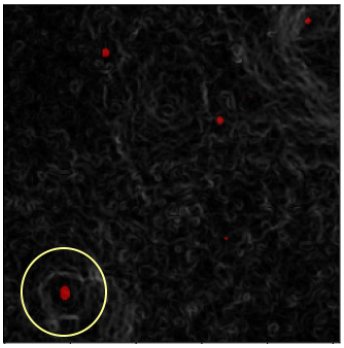


Image 7 - inclination

Image 7 - hillshade

Figure 5.15: Prediction Differences

It is also important to notice that in 100% of cases where predictions were correct predicted masks overlapped annotation masks.

5.4 Discussion

To address the research questions, we need first to link back to the design and implementation section. The implementation was based on a state-of-the-art ML model called ResNet34. The model's architecture was ready off-the-shelf and was not customized during the implementation. The original datasets were equally small and sparse but extensive data augmentation allowed them to expand dramatically.

The Results section showed promising results even though the number of images available for testing was limited to 10. We understand that being performed on a bigger set of testing images, the results might be very different. Unfortunately, we could not afford to exclude more annotated images and sacrifice them for testing, especially considering how small the original datasets were.

Analysis showed that historical charcoal production sites can be marked on both visualizations of LIDAR data. Moreover, both models show rather high the recall values for both datasets which indicates their ability to detect a high percentage of all the hidden objects. However, the accuracy and the precision for the hillshade model was higher than the accuracy and the precision for the inclination model due to the amount of false-positive predictions. This resulted in a higher F-score mean for the hillshade model. Also, the inclination model found one object on the testing dataset, that hillshade model missed.

It is also important to mention that there were false-positive objects both models agreed on. Keeping in mind that not all the sites of historical charcoal production are registered, those objects might theoretically be unsurveyed kilns, which would potentially increase the prediction accuracy up to 75% for the hillshade model and up to 50% for the inclination model.

The precision of marking is another topic for discussion. All the correctly located markings for both models landed right in the middle of the objects and overlapped with annotation masks. This highlights the ability of the models to return exact coordinates of discovered objects.

The results also revealed limitations of the design. In the design, bigger pictures are cut into 256 x 256 pixel-sized areas to be given to the model. If a historical site spans over two or more images being cut by the edges the model has no chance of putting the images together and might have a hard time trying to recognize the object from pieces.

Finally, the results we obtained correlate with the results of related research, proving that modern CNN are powerful tool for computer vision tasks. Even though our work highlights that this tool can be used off-the-shelf, basic knowledge of ML and data preparation, as well as a foundation of knowledge in Computer Science are still required.

6 Conclusions and Future Work

Trying to answer the research question, we applied a design science methodology to design and implement an AI algorithm capable of finding and marking historical charcoal production sites using available geodata of the Swedish landscape. The solution shows acceptable performance and meets other requirements of the project while the implementation part serves as a step-by-step manual on how to replicate the design.

Even without further improvements, our solution can be applied for practical use in semi-automatic mode where the found objects have to be verified by archeology experts. Even though such a usage scenario still requires an expert's attention, it is instrumental in saving time and efforts of highly qualified specialists.

The solution we designed has wide potential for future work. First and foremost is the application of a similar approach to detection of other types of historical remains. As long as only little amount of annotated data is needed and no additional annotations except geocoordinates are required, it seems possible to replicate the success for other types of historical remains hidden in Swedish forests.

It is important to understand that our solution is more of a working prototype than an end-user application. Thus, it can be improved at least by allowing for the combination and/or combining prediction results from the two types of geodata described in the thesis. Another improvement that can be done is to mitigate the limitation of a potential object being cut by images' edges. The model training process can also be done in a slightly different way using other optimization functions or with different amounts of images trained at the same time—those adjustments can potentially improve the overall performance.

Finally, there are other types of visual representations of LIDAR telemetry to be tried against the same model, as well as others types of models to be tried against the same data.

References

[1] C. Helander, "Forests and forestry in sweden," https://www.skogsstyrelsen.se/globalassets/in-english/forests-and-forestry-in-sweden_2015.pdf, 2015, accessed: February 13, 2021.

[2] "Four out of ten cultural heritage sites are damaged by regeneration felling activities," https://www.skogsstyrelsen.se/en/news/four-out-of-ten-cultural-heritage-sites-are-damaged-in-regeneration-felling/, December 8, 2015, accessed: February 13, 2021.

[3] "Survey of environmental consideration to cultural heritage," https://www.skogsstyrelsen.se/en/statistics/subject-areas/survey-of-environmental-consideration-to-cultural, August 8, 2020, accessed: February 13, 2021.

[4] "About Fornsök," https://www.raa.se/in-english/digital-services/about-fornsok/, accessed: February 13, 2021.

[5] "Thesis projects at softwerk 2021 - södra," 2021, unpublished document.

[6] M. D. Johnson, O. Fredin, A. E. Ojala, and G. Peterson, "Unraveling scandinavian geomorphology: the lidar revolution," *GFF*, vol. 137, no. 4, pp. 245–251, 2015.

[7] A. Hennius, "Spår av kolning : arkeologiskt kunskapsunderlag och forskningsöversikt," Tech. Rep., 2019. [Online]. Available: http://creativecommons.org/licenses/by/4.0/deed.sv

[8] "Maps and geographical information lantmäteriet," https://www.lantmateriet.se/sv/Kartor-och-geografisk-information/, accessed: February 13, 2021.

[9] Øivind Due Trier, A.-B. Salberg, and L. H. Pilø, "Semi-automatic mapping of charcoal kilns from airborne laser scanning data using deep learning," in *Oceans of Data*, M. Matsumoto and E. Uleberg, Eds. Archeaeopress Publishing LDT, Mar. 2016, pp. 219–231.

[10] C. M. Albrecht, C. Fisher, M. Freitag, H. F. Hamann, S. Pankanti, F. Pezzutti, and F. Rossi, "Learning and recognizing archeological features from lidar data," in *2019 IEEE International Conference on Big Data (Big Data)*, Los Angeles, CA, USA, 2019, pp. 5630–5636.

[11] H. A. Orengo, F. C. Conesa, A. Garcia-Molsosa, A. Lobo, A. S. Green, M. Madella, and C. A. Petrie, "Automated detection of archaeological mounds using machine-learning classification of multisensor and multitemporal satellite data," *Proceedings of the National Academy of Sciences*, vol. 117, no. 31, pp. 18 240–18 250, 2020. [Online]. Available: https://www.pnas.org/content/117/31/18240

[12] M. Bundzel, M. Jaščur, M. Kováč, T. Lieskovský, P. Sinčák, and T. Tkáčik, "Semantic segmentation of airborne lidar data in maya archaeology," *Remote Sens*, no. 12(22), 2020.

[13] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS quarterly*, vol. 28, no. 1, pp. 75–105, 2004.

[14] N. Sebe, *Machine Learning in Computer Vision*, 1st ed., ser. Computational Imaging and Vision, 29. Dordrecht: Springer Netherlands : Imprint: Springer, 2005.

[15] E. Alpaydin, *Introduction to machine learning*, third edition.. ed., ser. Adaptive computation and machine learning. Cambridge, Massachusetts ; London, England: The MIT Press, 2014.

[16] E. Smistad, T. L. Falch, M. Bozorgi, A. C. Elster, and F. Lindseth, "Medical image segmentation on gpus – a comprehensive review," *Medical image analysis*, vol. 20, no. 1, pp. 1–18, 2015.

[17] *Computer vision in medical imaging*, ser. Series in computer vision ; v. 2. Singapore ; Hackensack, New Jersey: World Scientific Publishing Company, 2014.

[18] Y. Ma, G. Luo, X. Zeng, and A. Chen, "Transfer learning for cross-company software defect prediction," *Information and software technology*, vol. 54, no. 3, pp. 248–256, 2012.

[19] Y. Gao and K. M. Mosalam, "Deep transfer learning for image-based structural damage recognition," *Computer-aided civil and infrastructure engineering*, vol. 33, no. 9, pp. 748–768, 2018.

[20] Y. Wang, L. Mo, H. Ma, and J. Yuan, "Occgan: Semantic image augmentation for driving scenes," *Pattern recognition letters*, vol. 136, pp. 257–263, 2020.

[21] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.

[22] M. Valueva, N. Nagornov, P. Lyakhov, G. Valuev, and N. Chervyakov, "Application of the residue number system to reduce hardware costs of the convolutional neural network implementation," *Mathematics and computers in simulation*, vol. 177, pp. 232–243, 2020.

[23] M. Matsugu, K. Mori, Y. Mitari, and Y. Kaneda, "Subject independent facial expression recognition with robust face detection using a convolutional neural network," *Neural networks*, vol. 16, no. 5, pp. 555–559, 2003.

[24] F. Liu, G. Lin, and C. Shen, "Crf learning with cnn features for image segmentation," *Pattern recognition*, vol. 48, no. 10, pp. 2983–2992, 2015.

[25] Lidar. [Online]. Available: academic-eb-com.proxy.lnu.se/levels/collegiate/article/lidar/544321

[26] R. Hesse, "Lidar-derived local relief models - a new tool for archaeological prospection," *Archaeological prospection*, vol. 17, no. 2, pp. 67–72, 2010.

[27] Ogc geotiff standard. [Online]. Available: https://www.ogc.org/standards/geotiff

[28] Common format and mime type for comma-separated values (csv) files. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc4180

[29] Splitting into train, dev and test sets. [Online]. Available: https://cs230.stanford.edu/blog/split/

[30] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.

[31] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," 2017.

[32] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," 2015.

[33] H. Choi, S. Ryu, and H. Kim, "Short-term load forecasting based on resnet and lstm," 2018.

[34] Q. Zhang, Z. Cui, X. Niu, S. Geng, and Y. Qiao, "Image segmentation with pyramid dilated convolution based on resnet and u-net," in *Neural Information Processing*, D. Liu, S. Xie, Y. Li, D. Zhao, and E.-S. M. El-Alfy, Eds. Cham: Springer International Publishing, 2017, pp. 364–372.

[35] Q. Huang, J. Sun, H. Ding, X. Wang, and G. Wang, "Robust liver vessel extraction using 3d u-net with variant dice loss function," *Computers in Biology and Medicine*, vol. 101, pp. 153–162, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0010482518302385

[36] Concepts – ml glossary documentation. [Online]. Available: https://ml-cheatsheet.readthedocs.io/en/latest/nn_concepts.html?highlight=loss#loss-functions

[37] Optimizers – ml glossary documentation. [Online]. Available: https://ml-cheatsheet.readthedocs.io/en/latest/optimizers.html#adam

[38] W.-M. Lee, *Python machine learning*, 1st ed. Indianapolis, Indiana: Wiley, 2019.

[39] Image data processing. [Online]. Available: https://keras.io/api/preprocessing/image/

[40] Segmentation models. [Online]. Available: https://github.com/qubvel/segmentation_models

[41] Resnet34 (imagenet) - model - supervisely. [Online]. Available: https://supervise.ly/explore/models/res-net-34-image-net-4225/overview

[42] An overview of semantic image segmentation. [Online]. Available: https://www.jeremyjordan.me/semantic-segmentation/

[43] Imagenet. [Online]. Available: https://www.image-net.org/

[44] Matplotlib: Python ploting. [Online]. Available: https://matplotlib.org/

[45] *OpenCV essentials : acquire, process, and analyze visual content to build full-fledged imaging applications using OpenCV*, ser. Community experience distilled. Birmingham, England: Packt Publishing, 2014.