LIÈGE université Library

https://lib.uliege.be

https://matheo.uliege.be

Mémoire

Auteur : Baudhuin, Alice
Promoteur(s) : 20595; Jonard, François
Faculté : Faculté des Sciences
Diplôme : Master en sciences spatiales, à finalité spécialisée
Année académique : 2022-2023
URI/URL : https://arcg.is/1aqK0v0; http://hdl.handle.net/2268.2/17638

Avertissement à l'attention des usagers :

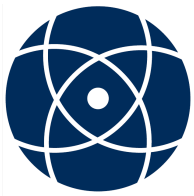
Tous les documents placés en accès ouvert sur le site le site MatheO sont protégés par le droit d'auteur. Conformément aux principes énoncés par la "Budapest Open Access Initiative"(BOAI, 2002), l'utilisateur du site peut lire, télécharger, copier, transmettre, imprimer, chercher ou faire un lien vers le texte intégral de ces documents, les disséquer pour les indexer, s'en servir de données pour un logiciel, ou s'en servir à toute autre fin légale (ou prévue par la réglementation relative au droit d'auteur). Toute utilisation du document à des fins commerciales est strictement interdite.

Par ailleurs, l'utilisateur s'engage à respecter les droits moraux de l'auteur, principalement le droit à l'intégrité de l'oeuvre et le droit de paternité et ce dans toute utilisation que l'utilisateur entreprend. Ainsi, à titre d'exemple, lorsqu'il reproduira un document par extrait ou dans son intégralité, l'utilisateur citera de manière complète les sources telles que mentionnées ci-dessus. Toute utilisation non explicitement autorisée ci-avant (telle que par exemple, la modification du document ou son résumé) nécessite l'autorisation préalable et expresse des auteurs ou de leurs ayants droit.

University of Liège

Faculty of Science

Master in Space Sciences - Professional Focus



ZRC SAZU

LIÈGE université

Deep Learning in ArcGIS Pro using Lidar data for automatic detection of archaeological structures

Alice Baudhuin

Academic year 2022-2023

Master Thesis

Supervisors : Žiga Kokalj
François Jonard

Readers : Andrea Nascetti
Pierre Hallot

Contents

Acknowledgments

First and foremost, I would like to thank Žiga Kokalj for his insightful mentoring during my internship at ZRC SAZU. I am grateful for his enthusiasm, ideas and advice for this Master's thesis. I would also like to thank all the other staff members at ZRC SAZU for welcoming me to their institute. I am very thankful for their help and time.

I would also like to thank Prof. François Jonard from the University of Liège for agreeing to supervise my Master's thesis. Moreover, I would like to express my gratitude to Michaël De Becker for his advice during my two years in the Master of Space Sciences.

Finally, I would like to express my sincere gratitude to my family for their constant support. I am grateful for their interest and curiosity in my studies.

Abstract

The application of deep learning to airborne laser scanning data (ALS, lidar) is now very useful for archaeological purposes. ALS data has already proven to be extremely relevant in archaeology, as archaeological features are much more visible. Advances in artificial intelligence now also make it possible to expand areas of interest in an automated way. This has a potential to lead to significant time savings in the labelling of archaeological features.

The state-of-the-art neural network architecture for this type of computer vision task is a convolutional neural network. In this work, both object detection and semantic segmentation were investigated, using a workflow in ArcGIS Pro. The area considered was the ancient Maya urban center of Chactún, which contains many labelled archaeological features for training the models. Several neural network architectures were trained on this area, including the ResNet-18, ResNet-34 and ResNet-50 backbones. Furthermore, two visualisations were used to train the deep learning models, namely the one-band and three-band visualisations for archaeological topography (VAT), and the digital elevation model. The models were tested on terrain similar and different from the training area.

The results show that the most suitable computer vision task for the available dataset is semantic segmentation. Furthermore, the best performing backbone architecture depends on the visualisation used for training. However, the model with the highest overall performance proved to be the one using the one-band visualisation for archaeological topography and ResNet-34 backbone. We also found that if there are overlapping objects in the training set, they should be considered separately by training one model per feature class. In this case, the best model differs depending on the feature class considered. Finally, the important parameter of patch size was investigated and a size of 256×256 pixels (with a pixel size of 0.5 m) was found to be best for the scale of the considered objects.

List of Figures

List of Tables

1 Introduction

Airborne laser scanning data (ALS, lidar) has become indispensable in archaeology. Before embarking on a field study, it is interesting to look at the area from a distance. ALS data acquisition, which became possible with the advent of lasers in the 1960s, is a way to do this. A notable achievement of ALS technology was the measurement of altitude during the flight of the Ingenuity helicopter on Mars, which illustrates the importance of this technology [1]. Mapping the Earth with ALS is of particular interest to archaeology. ALS-derived imagery provides a high-resolution, detailed view of an area where archaeological structures can be seen clearly. Not to mention that vegetation cover is not an issue with ALS, with the exception of very dense forests, whereas it is with photogrammetric imagery. Structures that might be missed in a ground survey due to dense vegetation can be visible in a ALS image. In this way, many unknown remains of the ancient Maya civilisation buried under a dense forest are being uncovered. The implications for understanding the Maya civilisation are great [2]. The visualisations obtained from ALS allow archaeological features to be identified either visually or using deep learning algorithms.

Improvements in computing power allow us to use deep learning in ways that were not possible in the 1990s [3]. Artificial Intelligence is crucial today as we face the Big Data tsunami. Indeed, current technological advances are resulting in an enormous amount of data that continues to grow. Although this information is undoubtedly useful for people around the world, analysing such a large amount of data is becoming a problem. Artificial intelligence is one solution to this issue. The more specific field of deep learning is becoming increasingly popular for the analysis of remote sensing imagery. In particular, convolutional neural networks are nowadays the most popular algorithm for image analysis. These neural networks narrow the gap between humans and computers, allowing computers to "see" as we do [4]. Deep learning is therefore an undeniable added value for the field of archaeology. Large data sets can be processed and potential sites automatically identified. With deep learning, manual visual inspection is no longer the only method for recognising archaeological features.

This paper investigates the two computer vision tasks of object detection and semantic segmentation. The corresponding deep learning models, RetinaNet for object detection and U-Net for semantic segmentation, are trained on the Maya centre of Chactún. An extensive dataset with labelled archaeological features is available for this region, which makes deep learning possible. The workflow in ArcGIS Pro to train a deep learning model is described in detail in this thesis. We also explain how the trained model can be applied to a new area to detect new structures. The impact of the model's architecture on its performance and the influence of the chosen visualisation of the area are examined. Two different visualisations of the ALS data were investigated. The models are also tested on several new areas to investigate the transferability of the model to different terrain types and different data quality. Finally, we test the influence of the size of the images being fed to the convolutional neural network.

2 The Chactún area

Chactún is an ancient urban centre of the Maya civilisation that has only recently been discovered. It is located in Campeche, Mexico. More precisely in the Calakmul Biosphere Reserve in the central lowlands of the Yucatán Peninsula. Very recently, a large number of archaeological structures were discovered in this area thanks to ALS data. This discovery sheds new light on the Maya civilisation, which has a population larger than assumed, and makes it very interesting to study. Evidence suggests that this Maya centre flourished during the Late Classic period, which corresponds to about 600 to 1000 AD. Remains of structures can be found all over the site delineated in Figure 1. Professor Šprajc's team discovered three ancient cities within this region in 2013 : Chactún, Tamchén and Lagunita. These three residential areas, which can be seen in Figure 1, form the Chactún area considered in this study. They are located on the terrain with the highest elevation. Smaller clusters of buildings are found all around these three urban centres. Chactún has the largest architectural volume. The relief of the entire area is characterised by low hills with surrounding bajos (seasonally flooded flat areas) [5].

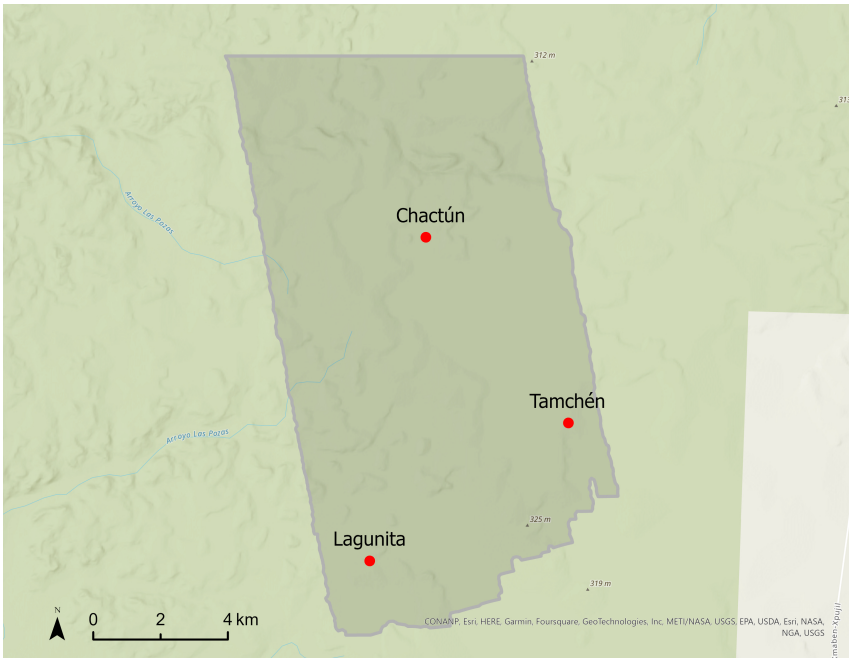


Figure 1: The Chactún area where the three largest cities are indicated.

The urban core of Chactún can be seen in Figure 2. The area includes several complexes, which are described in detail in the article by Šprajc et al. (2022) [5]. First of all, platforms are found all over Chactún. These are artificial, flat surfaces that are elevated with respect to the surroundings and on which remains of buildings are often found. It can be noted that platforms that do not appear to support a building may have supported perishable structures. The second and most numerous component of the region are buildings of various sizes, either standing on platforms or not. Most of them were residential. They can be found in groups or isolated. In some cases, only remains of walls are found, which suggests thatched roofs. Most, however, indicate vaulted rooms with remnants of roofs. An example of remains of a building can be found in Figure 3a. Finally, several aguadas

(artificial water reservoirs) have been identified. They are characterised by a lower terrain compared to the surrounding area, with slightly raised edges. Their existence stems from the fact that there is no permanent water on the surface. A large aguada can also be seen in Figure 3b. In figure 2, two ball courts are visible, as well as remains of buildings such as that of the western complex. These building structures stand on a platform. At the top left is the largest aguada in the Chactún region, labelled as "reservoir". Causeways, up to 30 m wide, connect the various complexes together. Other archaeological features found in the Chactún region include temple pyramids, sculpted monuments (such as stelae and altars), terraces, plazas, ridges and lime kilns. The elevation of the area ranges from 220 to 295 metres, rising by a few metres if the buildings are considered [6]. The terrain is a mixture of flat and hilly areas. Based on recent discoveries, the Chactún area can be classified as one of the largest Maya centres in the central Yucatán lowlands discovered to date. For this reason, it is worth exploring in detail. However, Chactún is covered by a dense tropical forest with trees up to 20 m high [5]. It is then necessary to penetrate this semi-deciduous forest in order to see the archaeological features. ALS data offers a solution to this problem. ALS data for the region was acquired in 2016 and supplemented by pedestrian field surveys in 2017 and 2018. The data can provide information on water management and agriculture through terrain changes. Insights can also be derived about the socio-political organisation in which Chactún played an important role [5].

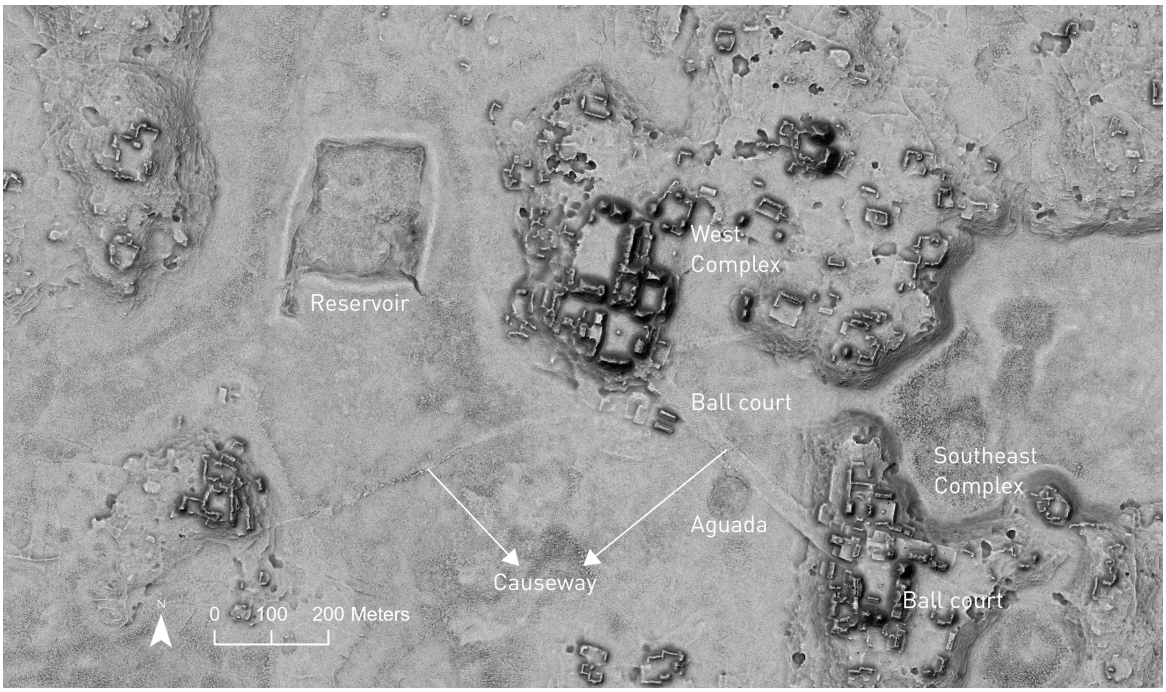


Figure 2: Annotated visualisation of Chactún core.

(a)

(b)

Figure 3: Picture of (a) a building and (b) an aguada (photo by Žiga Kokalj).

3 Deep learning

3.1 A brief introduction to machine learning

Deep learning as a branch of machine learning, which is itself part of the broader field of artificial intelligence. A few words deserve to be said about machine learning, as deep learning is only one specific case of this field. The reason this field was developed in the first place was to find a way to use computers to solve problems that were unsolvable for humans. One example is the navigation of the Mars rover. The great distance to the red planet makes it difficult for humans to navigate a vehicle from Earth. One can point out the time delay between the transmission and reception of a signal between the rover and Earth. As well as the limited amount of data that can be transmitted. Autonomous rovers are a key to overcoming these issues. Machine learning is able to provide this necessary autonomy [7]. Another classic example of machine learning is the recommendation algorithms of music streaming services which suggest songs you might like based on previously played songs [8]. A final example is the recognition of handwritten characters [4].

The discipline has continued to evolve and has become an essential tool for making predictions and turning difficult and time-consuming tasks into automated ones. Machine learning focuses on algorithms that allow computers to learn from a given set of data, called a training set. During the learning process, the model will improve its performance. Once the model is fully trained, it can be applied to new data and provide valid knowledge using what it has learned.

There are two types of algorithms depending on the initial dataset provided to this algorithm. The two types correspond to two different learning methods. The first is supervised learning, where the training data contains a target. In other words, the output that the algorithm should provide is known and the computer can adjust its parameters to find the desired output. In contrast, one speaks of unsupervised learning when the training data does not contain an expected output and the algorithm has to figure it out on its own [8]. This paper focuses on supervised machine learning. The ultimate goal of the algorithm is then to find a function of the inputs (X1, X2, X3) which provides the best possible approximation of the given output: $\hat{Y} = f(X1, X2, X3)$ [9]. This equation is the machine learning model. The inputs are called learning or training samples. Finding the best function parameters corresponds to achieving the highest performance of the model.

This is accomplished by minimising a loss function. This function returns the error of the model, i.e. the difference between the target output and the predicted output. This loss is often calculated as the root mean square error for a regression problem (with a numerical output):

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{1}$$

where N is the total number of learning samples, y is the target output (called ground truth) and $\hat{y}$ is the predicted output. In a classification problem, where the output is a class associated with a probability, the loss function is often the binary cross-entropy loss [10]:

$$CE = -\frac{1}{N}\sum_{i=1}^{N}(y_i \; log(p_i) + (1 - y_i) \; log(1 - p_i)) \tag{2}$$

where p_i is the probability of belonging to a class. A small loss value indicates a prediction which is close to reality. The parameters that minimise the loss function are found by computing the first derivative of the loss function with respect to that parameter. The function with the smallest loss is kept by the model [8].

The dataset is divided into several parts. One part is the training set from which the algorithm learns and minimises the loss function. A second part is the testing set which consists of new data that the algorithm has never seen before and allows the performance of the model to be evaluated. Therefore, one error is associated with the training set and another with the testing set. These errors can be seen in Figure 4, which illustrates an important concept in machine learning: overfitting and underfitting. Overfitting occurs when the model "sticks" too much to the training data and includes noise. The model is then too complex and the error on the testing set increases. Underfitting happens when the model is too simple to represent the phenomenon. In this case, the error is high for both the training and testing sets. The ability of the model to make predictions on new data is called the generalisation of the model. This generalisation is evaluated through the testing set and the best model minimises the test error [8].

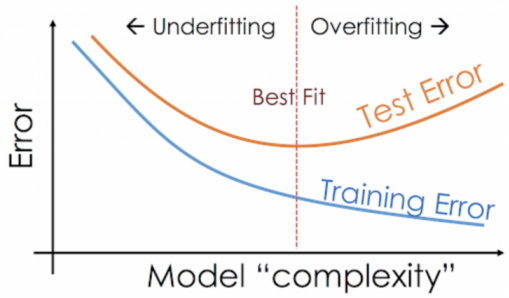


Figure 4: Error for the training and testing sets as a function of the model complexity [11].

A third part of the dataset is often used for training the model. It is called the validation sample. Like the training sample, it is used during the learning phase. It

allows to assess the performance of the model created with the training set. Even though both assess the performance of the model, the validation sample and the test sample are different. The validation sample allows the model to adapt in order to improve its performance and avoid overfitting. To do so, the hyperparameters of the model can be modified. These hyperparameters determine the structure of the model and control the learning process [12]. In contrast, the test sample is used at the very end to evaluate the generalisation performance of the final model and was never seen by the model during training [13].

3.2 Basis of deep learning

Deep learning is a subfield of machine learning. The main difference between the two is that machine learning requires some pre-treatment of the data. Namely, the features (also called variables) have to be engineered manually. In other words, the raw data must be modified to create the variables needed to train the model [14]. On the contrary, deep learning doesn't require such pre-treatment. Feature engineering is learned by the model starting from the raw data. For example, if a machine learning model is presented with an image of a car, the user must tell the model that this is a car. A deep learning model will recognise the car on its own. This feature engineering aspect will be illustrated further. Deep learning is also more performant than machine learning when the dataset is large.

Deep learning uses Artificial Neural Networks (ANN). The idea behind it is to replicate the processes that take place in the biological neurons of our brain. Just like the biological neurons, the artificial neurons take in information, process it and produce an output. An individual artificial neuron, called a perceptron, is shown in Figure 5. The mathematical equation of a neuron is as follows [15]:

$$\hat{y} = f(\sum_i w_i x_i + b) \tag{3}$$

The neuron is provided with a set of input values x_i. An input is an attribute and can be anything from the surface of a flat to the value of a pixel of an image. One requirement is that the input must be numeric, so an image is represented by its pixel values. Each value is assigned a weight w_i. The b corresponds to the bias. These weights and the bias are the parameters that the model will change in order to learn and improve its performance, by minimising the loss function. $\hat{y}$ is the output of the neuron and f is a non-linear function, called the activation function. This function decides whether the neuron is activated or not. The non-linearity allows the model to learn more complex processes and approximate almost anything.

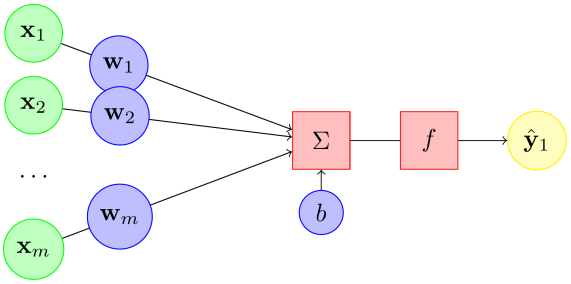


Figure 5: Computational graph of a perceptron [15].

These artificial neurons are connected to each other to form a layer. Several layers are in turn connected to form what is called a neural network. The most common form of a neural network, referred to as Multi-Layer Perceptron, is shown in Figure 6. In a neural network, the output of one neuron becomes the input of another. The network can vary in size, with varying numbers of layers and artificial neurons, depending on the application. If there is more than one hidden layer, it is called a deep neural network. Each circle (green, orange or blue) corresponds to a neuron (or perceptron) and can therefore be replaced by the Figure 5. Like any machine learning algorithm, neural networks need to be trained.

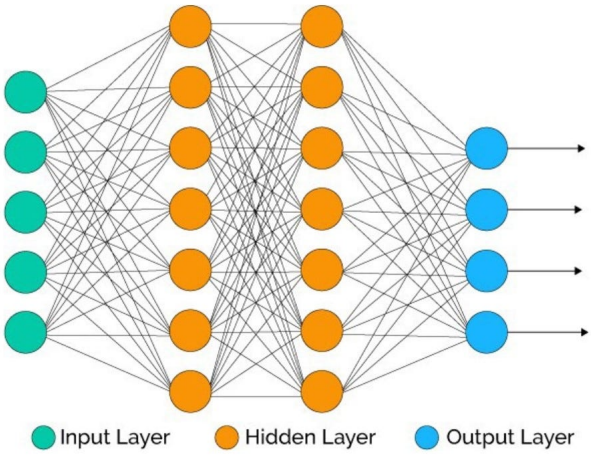


Figure 6: Illustration of a usual neural network [16].

Minimising the loss function across the neural network is achieved through a process called gradient descent or backpropagation. This is an optimisation algorithm where the gradient of the loss function with respect to the weights (which must be minimised) is calculated for the very last layer of the network and is then propagated back towards the very first layer. During the process, the values of the weights are updated to reach the minimum loss. The goal of the training process is to achieve generalisation, which is basically defined as how well the model is able to make predictions for new data [3].

A word should be said about transfer learning since it will be used later on. It involves using an already trained deep learning model as the basis for the structure of a new model. It helps to improve the performance and speed of the training process of the new model. The weights are in fact initialised with the values of the pre-trained model and are therefore already closer to the optimal values (minimising the loss). They are then updated through backpropagation. A simplified deep learning workflow can be found in Appendix A.

3.3 Convolutional Neural Networks

Convolutional Neural Networks, abbreviated CNNs, belong to the field of computer vision. This field is concerned with the processing of images and videos. CNNs thus try to replicate the visual system of the human brain [4]. Since images will be used here in the case of archaeology, these networks are suitable. A CNN is a particular neural network

architecture, illustrated in Figure 7, whose individual elements are explained below. The aim of the model is to determine what the object in the image is.

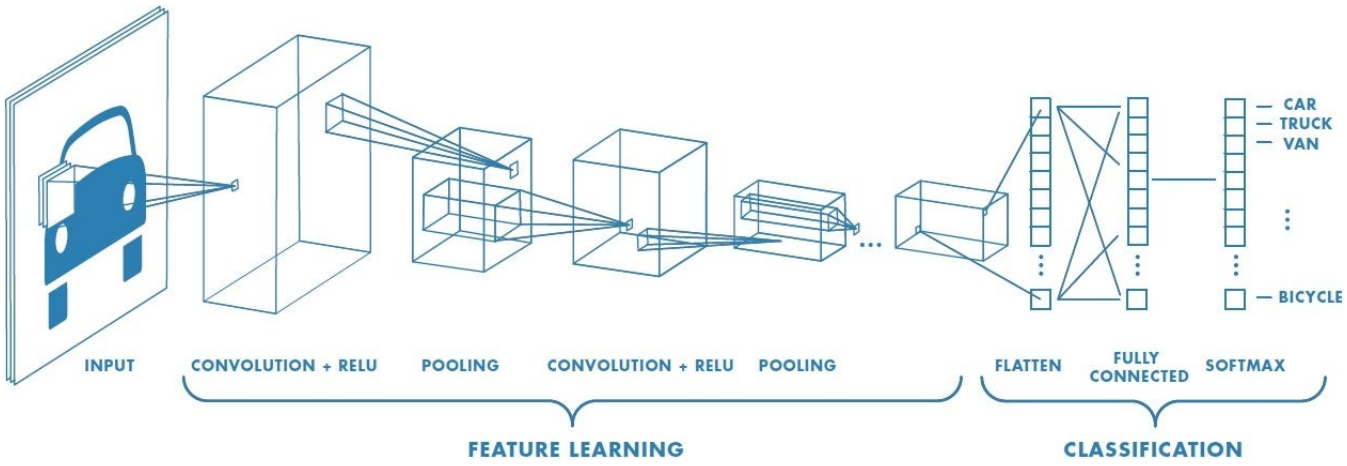


Figure 7: Illustration of a convolutional neural network [17].

The network uses filters (also called kernels) to identify specific features in an image. They thus perform feature extraction (or feature engineering) themselves, as mentioned above. These filters form the convolutional layers of the network. An example of a filter is shown in Figure 8. The filter slides over the image and produces a numerical value for each convolution, through linear matrix multiplications. When the filter has been applied to the entire image, the result is a feature map. During the learning process, the parameters (weights) of the filters are modified through the gradient descent method to improve the performance of the network. One may notice that a boundary line of zeros has been added to better analyse the edges of the input image. This addition is called padding.

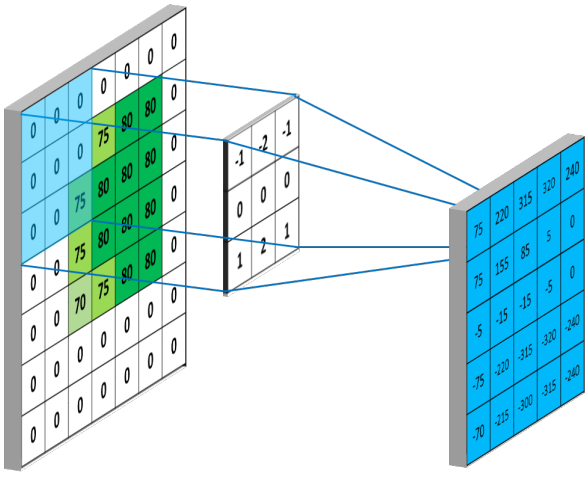


Figure 8: Illustration of a convolutional layer with the input image on the left, the filter in the middle and the output feature map on the right [18].

After the feature map is created, an activation function is applied to it. The aim is to increase the non-linearity of the final output, to better fit the results. The activation

function commonly used in CNNs is the ReLU function (Rectified Linear Unit), shown in Figure 9. It indicates whether a neuron is activated or not, hence introducing a non-linearity. The neuron is inactive if the input is negative [3]. However, a CNN doesn't only consist of an individual convolutional layer, as illustrated in Figure 7. A convolution and a ReLU function are followed by a pooling layer. This layer reduces the dimension of the feature map. A so-called maximum pooling layer, shown in Figure 9, takes the maximum value of the part of the feature map it is examining. It keeps only the important information and disregards the noise, leading to higher accuracy and speed. A succession of convolutional layers and pooling layers are available in the network for feature extraction. The first layers learn low-level features such as edges, lines or colours, while the last layers extract high-level features such as a tire. The final feature map of the feature learning process is then flattened into a vector and connected to a fully-connected layer. As the name suggests, each neuron in the flattened layer is connected to each neuron of this fully-connected layer. This layer performs the classification process. It makes a decision by learning possible combinations of high-level features. The output of the fully-connected layer is the probability that the feature belongs to a class. In our example of archaeology, one neuron contains the probability that the object belongs to the class of buildings, another to the class of platforms, and a third one to the class of aguadas. The last layer of the model contains a Softmax activation function that normalises the probabilities between 0 and 1. The label corresponding to the highest probability is retained. This completes the classification process [12]. One can note that in Figure 7, the layers are represented as rectangles with a certain width. This comes from the fact that several filters are used (one filter for one neuron), resulting in several feature maps. Each feature map contains a different characteristic of the input image. In other words, the parameters (weights and biases) of a filter are adjusted to search for and identify a particular aspect of the image. For example, one filter may be searching for vertical edges and another for horizontal edges. Once the learning process is complete, new images can be provided to the network which will be able to identify what is on them.

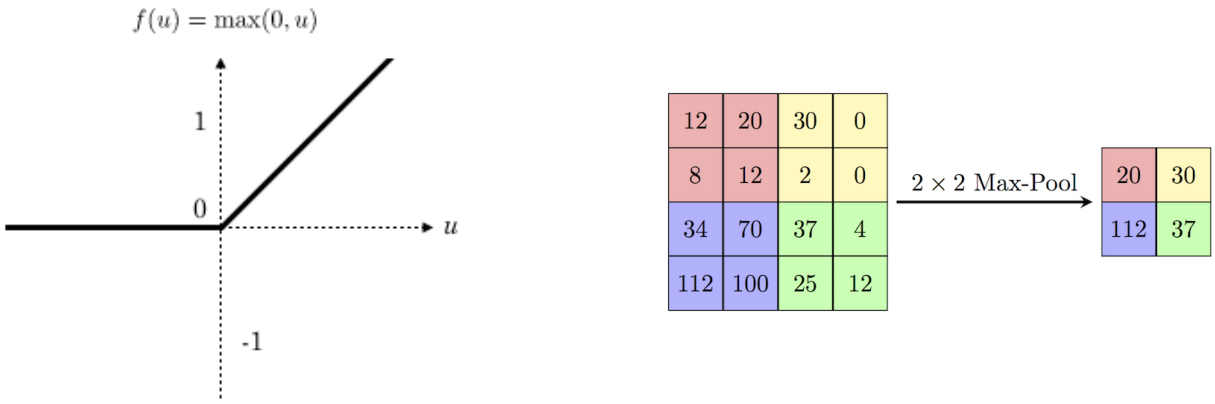


Figure 9: ReLU activation function on the left [19] and example of a max pooling layer on the right [20].

3.4 RetinaNet model for object detection

In this work, two different computer vision tasks using CNNs were tested: Object Detection and Semantic Segmentation. An object detection deep learning model identifies and localises objects in the image. Object detection is based on bounding boxes containing the objects. The training data contains ground truth bounding boxes (also called annotations) that encompass the known objects. The model is then trained to define a prediction bounding box as close as possible to this ground truth box. The idea behind object detection is the following: The model creates a set of anchor boxes within the image that have a predefined shape and size, as shown in Figure 10. The various anchor boxes are then examined to determine whether or not they contain an object. To do this, they are compared with the ground truth bounding boxes. The network calculates a probability that an object will be found and a IOU (Intersection Over Union). This IOU indicates how much of the anchor box is overlapped by the ground truth box. The definition is provided in Figure 10 and shows that a high IOU means that the prediction box predicts the inside of the ground truth box well (numerator), while not overflowing it (denominator). The value of the IOU goes from 0 to 1. The anchor box closest to the ground truth box, with the highest IOU and the highest probability, is retained. It is eventually adjusted to better fit the object and thus becomes the final prediction bounding box. This is the difference between bounding boxes and anchor boxes, because in the latter the size and shape remain unchanged, while the former are adjusted to better fit the object. This is illustrated in Figure 10, where the yellow boxes are the prediction boxes defined relative to an anchor box. An anchor box with a low IOU relative to the ground truth box is assigned as background [21].

Figure 10: Illustration of a set of anchor boxes on the left and illustration of the Intersection Over Union on the right [21].

The deep learning model used in this thesis for object detection is the RetinaNet model. It is a one-stage algorithm for object detection. One-stage means that the algorithm directly uses a CNN to identify objects. In contrast, two-stage algorithms first identify the regions where objects could be found, and then focus only on those regions by using a CNN to detect objects [22]. The RetinaNet model is well suited for the detection of dense and small-scale features. It consists of two main components: Feature Pyramid Network

(FPN) and Focal Loss. The architecture of the model is shown in Figure 11. The figure shows four different parts of the model: (a) a bottom-up and (b) a top-down pathway, (c) a classification and (d) a regression subnetwork [23].

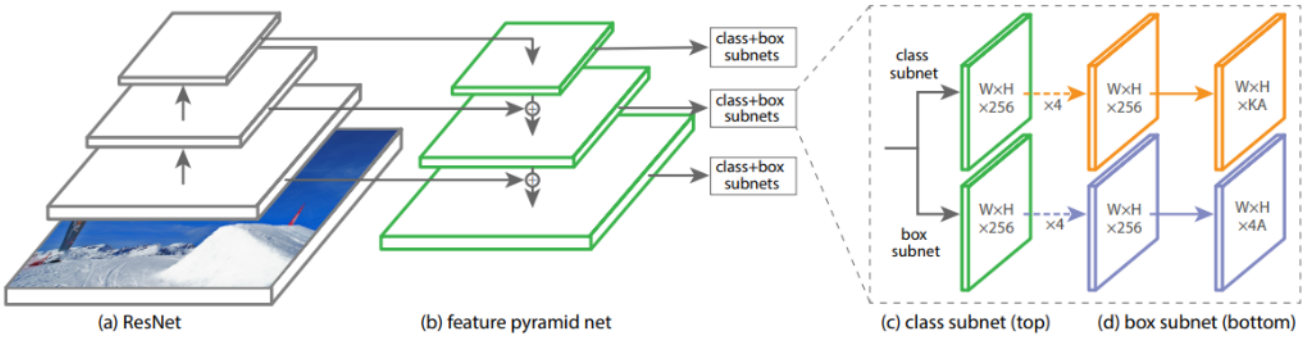


Figure 11: Architecture of the RetinaNet model [23].

A feature pyramid network, which here is the left part of Figure 11, combines low- and high-resolution data. The bottom-up part of the architecture goes from high-resolution maps to low-resolution ones. In other words, this part of the architecture computes feature maps with varying scales, just like any CNN. Objects with different scales can then be identified. The top-down part upsamples the low-resolution feature maps and combines them with the bottom-up layers via lateral connections. This combines low-level features (edges, lines, colours) and high-level features (object, scene). These come respectively from the high-resolution layers and the low-resolution layers, as illustrated in Figure 12. This creates a stronger representation that combines good spatial information and good semantic information. Further in the network, the classification subnetwork then analyses the anchor boxes associated with each point on the feature maps and determines the probability for an object to be found inside. The regression subnetwork reduces the difference between the anchor box and the ground truth box, if there is one, and thus provides the prediction box [23].

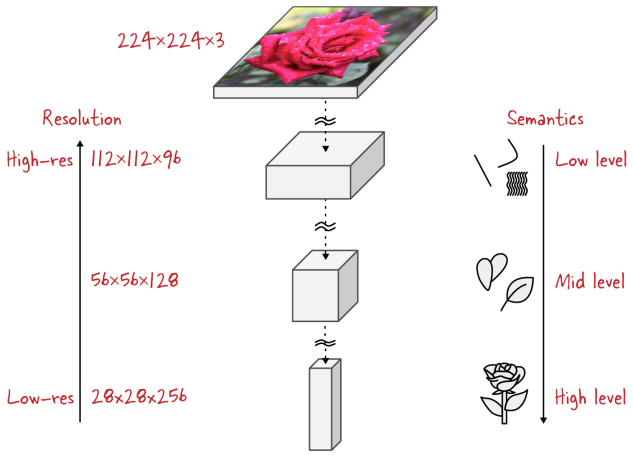


Figure 12: Illustration of feature maps at several layers of a CNN [21].

An additional word should be said about the bottom-up part of the architecture. This part is called the backbone model and is chosen here as the ResNet model. ResNet is a neural network architecture that enables transfer learning, i.e. it has already been trained before. In particular, it has been trained for a classification task on the ImageNet dataset, which contains more than one million images [24]. The number following the name ResNet indicates the number of layers in this backbone network. This already trained model is used as the base architecture for the training process of the new model, improving its performance.

Focal Loss is used in the RetinaNet model and makes it possible to deal with the problem of class imbalance. This relates to the fact that a sample may contain many more data samples in one class than in other classes. An algorithm will then assign more weight to the majority. This class imbalance issue occurs in one-stage models because of the sampling performed through the anchor boxes. In fact, each point in an input image is assigned a certain number of anchor boxes (nine for RetinaNet). Each box is then investigated during the training process and either associated to an object or not. If there is no object from the training data within the box, it is classified as background. Since the training dataset contains only a small number of objects compared to the total number of anchor boxes, many boxes are classified as background. The small losses associated with these backgrounds will overwhelm the model [21]. Focal loss reduces the contribution of these background boxes to the loss by slightly altering the cross-entropy loss function [23]. This is illustrated in Figure 13, which compares the formulas for cross-entropy (CE) and focal loss (FL). Note that sum and indices have been omitted for simplicity. In RetinaNet, a value of $\gamma = 2$ is used [25].

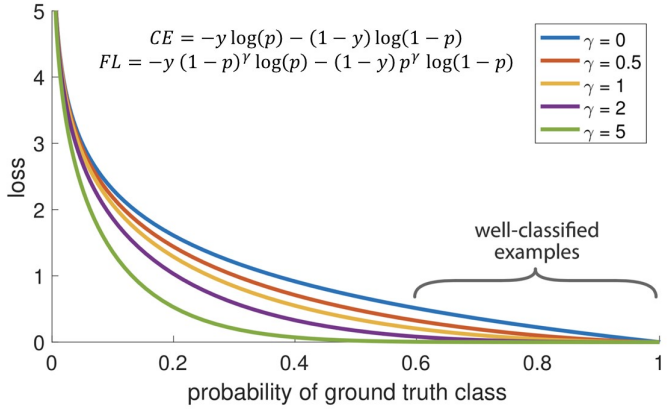


Figure 13: Illustration of the effect of focal loss [25].

The parameters of the RetinaNet model are listed in Table 1 with the default values. The scale corresponds to the size of the anchor boxes. It is set depending on the expected size of the objects. Aspect ratio is the ratio of height to width of an anchor box. It indicates the shape of the object. The values are chosen depending on the expected shape. Nine types of anchor boxes are available with RetinaNet, through the use of three different scales and three aspect ratios. These nine anchor boxes are built at each point of a feature map, as explained above. The chip size parameter is related to the fact that

an input image is divided into a set of sub-images called image chips or patches. This is explained in more detail in Section 6.1. The valid_loss setting for the monitor parameter indicates that the validation loss (difference between prediction and ground truth) is used to monitor the model. Training will continue until this loss is minimised.

Scales	1, 0.79, 0.63
Ratios	0.5, 1, 2
Chip size	256
Monitor	Valid_loss

Table 1: Model arguments of the RetinaNet model with the default values.

3.5 U-Net model for semantic segmentation

An alternative to object detection is semantic segmentation. This technique doesn't use bounding boxes, but classifies each pixel of an image (see Figure 14). A pixel is then assigned either to an object class or to the background. As opposed to simple image classification, where the entire image is given a class label, here the output of the model is a class label with information about its localisation. There are two names for this technique that can be used interchangeably: Semantic Segmentation and Pixel Classification.

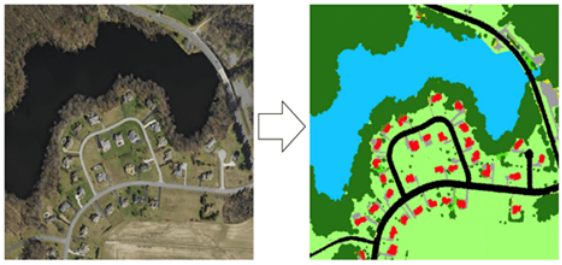


Figure 14: Illustration of semantic segmentation with the input image on the left and the output of the segmentation on the right [26].

The U-Net model is the most used model in semantic segmentation. It was originally developed for biomedical purposes [27]. The architecture is shown in Figure 15. It is based on an encoder and a decoder. The encoder compresses the information into a smaller dimension, reducing the resolution while providing high-level features. This part of the model architecture is similar to the usual architecture of a CNN, with convolutional layers, ReLU activation functions and pooling layers. It is usually a pre-trained ResNet model. The decoder, on the other hand, decompresses the information by upsampling, thus increasing the resolution back to the initial value. This part allows the projection of the previously learned high-level features into the higher-resolution pixel space. For better learning, connections are also made to the encoder part, as in the RetinaNet model [26]. The main advantage of this technique is that it allows to detect high-level features (contextual information) thanks to the encoder path, but also to localise features using the higher resolution maps obtained by the upsampling path. The output is then more spatially accurate [27].

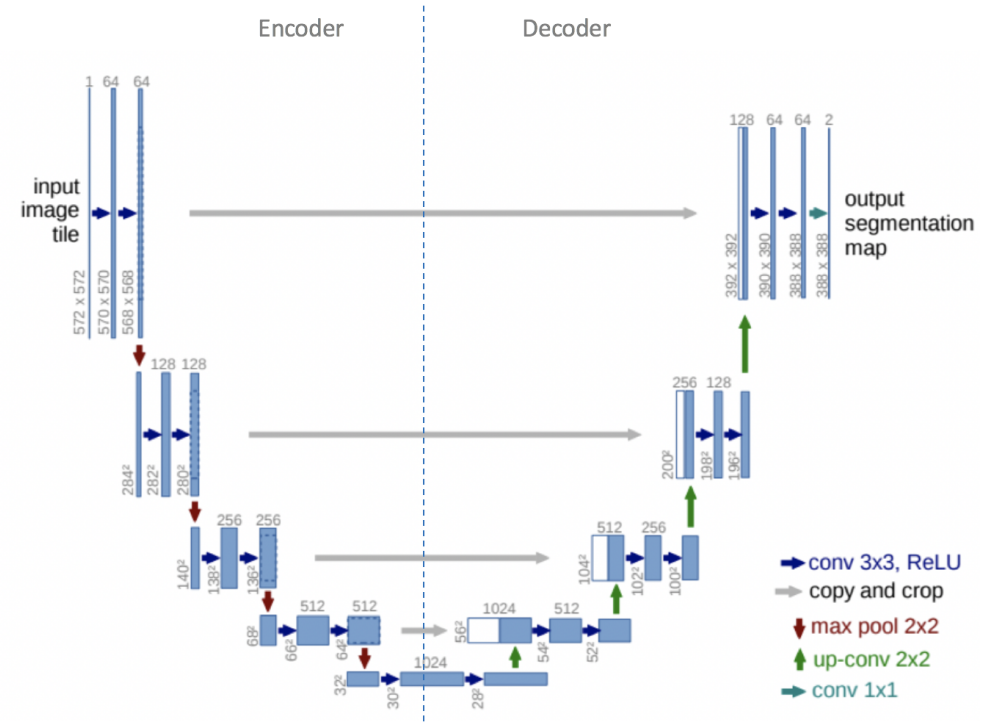


Figure 15: Architecture of the U-Net model. The blue rectangles are mutli-channel feature maps with the number of channels written at the top. Their size is written at their lower left [26].

The numbers above the layers (blue rectangles) on Figure 15 represent the number of channels and is related to the number of kernels applied to the input image. A feature map is then created for each kernel. The number 64 for the first set of feature maps means that 64 kernels have been applied to the image. As the network progresses, more and more feature maps are created by applying more kernels. This illustrates why it is impossible for humans to understand what is happening in a neural network and why it makes a particular decision. Neural networks are a black box of which we can only see the inputs and the outputs.

The model arguments are listed in Table 2 along with the default values. The class balancing and focal loss arguments are related to the class imbalance problem mentioned earlier. It is related to the fact that one class can have more training samples than another. Typically there are more background than target objects. Class Balancing and Focal Loss can solve the issue by reducing the contribution of classes with higher frequency to the loss. Class balancing focuses on smaller classes (with a smaller number of samples) and ensures that they are considered as important as other larger classes. Focal loss focuses on decreasing the number of misclassified examples by concentrating on pixels with a low probability. Mixup can be set to True if the number of training samples is low and data augmentation is needed. Chip size is the size of the sub-images that will be provided as input to the network. Finally, monitor provides the loss that will be used to monitor the performance of the model. These two last parameters are the same as those used with RetinaNet.

Class balancing	False
Mixup	False
Focal loss	False
Chip size	256
Monitor	Valid_loss

Table 2: Model arguments of the U-Net model with the default values.

3.6 Performance metrics

Several performance metrics are used in object detection and semantic segmentation to assess the performance of a deep learning model. The performance metric most commonly used for object detection is the accuracy. The accuracy indicates the percentage of data that was well predicted and is defined as follows:

$$\begin{aligned} Accuracy &= \frac{Number\ of\ correct\ predictions}{Number\ of\ predictions} \\ &= \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative} \end{aligned} \tag{4}$$

In the case of semantic segmentation, additional metrics are used to evaluate the performance: precision, recall and F1. They are defined below [28].

$$Presicion = \frac{True\ Positive}{True\ Positive + False\ Positive} \tag{5}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{6}$$

$$F1\ score = 2 \times \frac{Precision\ \times\ Recall}{Precision + Recall} \tag{7}$$

The precision metric is quite straightforward. A high value means that almost all predictions are ground truths. The recall metric measures the ratio of true positive predictions to the total number of ground truths. False negative predictions are in fact objects that have been incorrectly classified as background. A high recall score means that all ground truths were predicted. Finally, the F1 score provides the weighted average of the precision and recall metrics. Another metric of the model, mentioned above for object detection, is the accuracy. It is defined by the ratio of accurate predictions to the total amount of predictions (see Equation (4)). Thus, accuracy is the number of pixels that were correctly classified. While the accuracy evaluates the model in general (including the background), the precision and recall metrics focus only on the performance of the archaeological classes.

To assess how well the model performed on the test set, the IOU metric is commonly used for semantic segmentation. This metric gives the overlap between the ground truth and prediction regions in terms of pixels. Note that although this metric is also defined for object detection (see Figure 10), it is more important in semantic segmentation for performance evaluation. Namely, since each pixel is classified, it is no longer possible to count the number of false positives and true positives in terms of objects. A true or false

positive is now one pixel. However, an amount can be obtained for these pixels so that the IOU can be calculated by the following formula:

$$IOU = \frac{True\ Positive}{True\ Positive + False\ Positive + False\ Negative} \tag{8}$$

The values for the amount of TP, FP and FN are obtained using ArcGIS Pro's "Combine" tool, which compares the values of multiple rasters. The output is an attribute table with all possible combinations of raster values and the number of times they occur (see Appendix B). A pixel that has a value of 1 in both the ground truth and prediction rasters is a true positive. On the other hand, if a pixel has a value of 0 in the ground truth and 1 in the prediction, it is a false positive. The processing chain to obtain the values for calculating the IOU is shown in Figure 16. The "Test_..." vector files are the ground truths for the test area, while the "Classified_pixels" raster is the result of the segmentation.

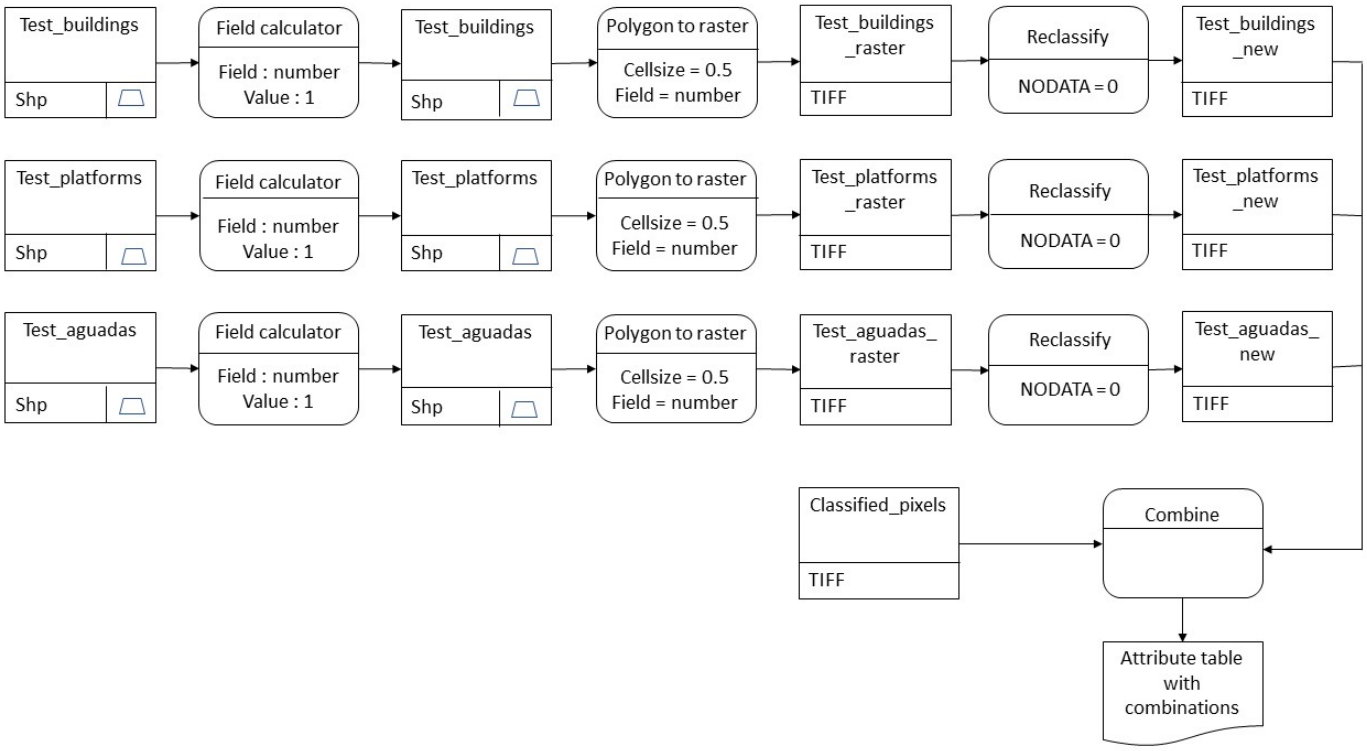


Figure 16: Processing chain for the IOU computation.

Another important metric is the dice coefficient. It is related to the previously mentioned IOU. To calculate the dice coefficient, we need to compute the number of pixels associated with a class (object or background) in the ground truth image and in the predicted image. The calculation to compute the dice coefficient is shown in Figure 17.

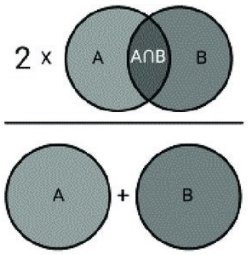


Figure 17: Illustration of the dice coefficient with A the prediction and B the ground truth [29].

The formula of the dice coefficient is the following:

$$Dice = \frac{2 \times True\ Positive}{(True\ Positive + False\ Positive) + (True\ Positive + False\ Negative)} \quad (9)$$

As with the IOU, the value is between 0 and 1, and a good model has a high dice value. One can note that the final dice value is the average of the dice calculated for both the class and the background.

4 Lidar remote sensing

Lidar, or ALS, is a recent field of remote sensing, with an emergence in the 1960s. The technology has continued to develop, especially in the last decade, and its applications keep growing. Today, for example, it is often used to create Digital Elevation Models (DEM) [30]. As radar, it is an active remote sensing method. This means that the sensor sends its own electromagnetic energy and does not rely on solar illumination. Lidar stands for "Light Detection and Ranging" and is based on measurements of distances. A laser pulse is sent to a target and part of its energy is scattered back to the sensor. The distance measurement can be achieved through two different methods: Time-of-Flight and phase shift.

This thesis uses data from airborne lidar surveys, but lidar data can also be acquired from the ground (ground-based), from space (spaceborne) or from drones (UAV-based). Lidar measurements from an airborne platform are also known as Airborne Laser Scanning (ALS). An ALS system, represented in Figure 18, consists of two main elements: a laser range finder (laser scanner and ranging unit) and a positioning system (GNSS and IMU) to georeference the points. The laser range finder provides the distance between the sensor and the target, called the range. The positioning system is necessary because the coordinates of the target can only be determined if the location and orientation of the ALS sensor are known. The location is provided by the Global Positioning System (GNSS or Differential GNSS with ground-based stations), while the orientation (defined by yaw, pitch and roll) is determined by the Inertial Measurement Unit (IMU). From the information provided by the GNSS, IMU, ranging unit and the scan angles, coordinates (x, y, z) can be assigned to a laser return point (in the WGS84 coordinate system). The Figure 18 illustrates three types of laser scanning: zigzag, parallel and elliptical.

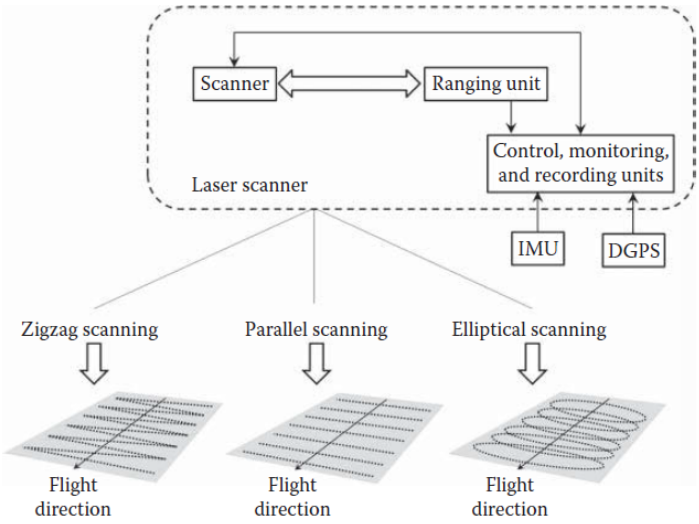


Figure 18: Representation of an ALS system [31].

Two particularities of ALS are that data collection can take place both during the day and during the night (since the system is active and thus independent of solar radiation) and that the laser signal can "see" through vegetation cover (thanks to gaps). The latter feature is particularly useful for creating a DEM under a forest canopy. One pulse is associated with several returns coming from different parts of a tree, for example. This comes from the fact that the emitted laser beam has a certain width and is therefore characterised by an instantaneous laser footprint, which is almost circular at nadir. Part of the beam can reach the ground if it is not reflected by vegetation (i.e. if gaps are present). Multiple returns from both vegetation and soil can then be recorded and distinguished thanks to the time difference between the returns [31]. Note that if the vegetation is too dense and no gaps are present, the beam cannot reach the ground. The soil will then not be seen by the ALS system.

An ALS acquisition, illustrated in Figure 19, has several characteristics that must be taken into account when planning the mission: the altitude and speed of the airborne system, the scan angle, the pulse repetition frequency (PRF which is the number of pulses per second), the swath width, the field of view and the overlap of the flight lines. All these parameters are selected according to the application. Laser pulses can be emitted at a very high cadence (1 million pulses per second for the state-of-the-art system). The point density can be derived from the first four parameters (altitude, speed, scan angle and PRF). A scanning mirror is used to orient the laser pulses across track (perpendicular to the line of flight). Dense forests, like the one over Chactún, require an ALS acquisition with a smaller field of view, so that the beam does not have to cross too much vegetation to reach the ground, and a higher point density (between 10 and 50 $points/m^2$) [30].

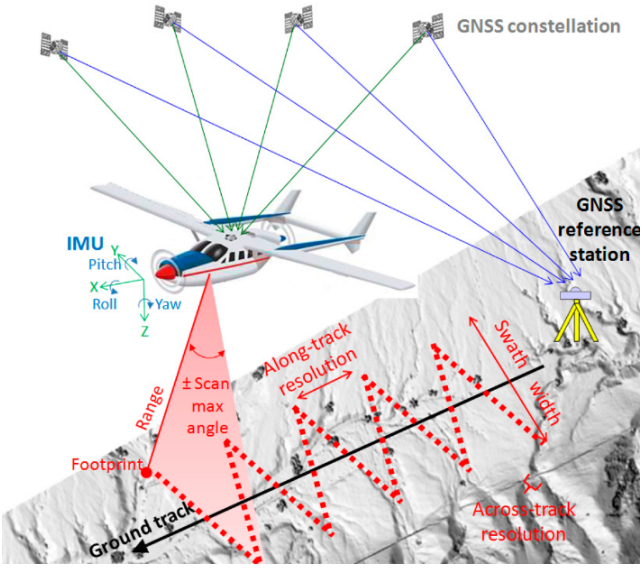


Figure 19: Illustration of an ALS acquisition [32].

There are two types of recording of returns: discrete or full-waveform. They are illustrated in Figure 20. With discrete recording, a few measurements are recorded per pulse emitted. With full-waveform, on the other hand, many more recordings are taken, typically at time intervals of 1 ns. Such a time interval corresponds roughly to a distance of 15 cm between two points. This allows a detailed vertical profile of the surface to be created with an almost continuous backscatter signal. ALS is therefore characterised by fine spatial resolution. Common wavelengths for lidar (airborne and ground-based) are between 500 and 1600 nm [31].

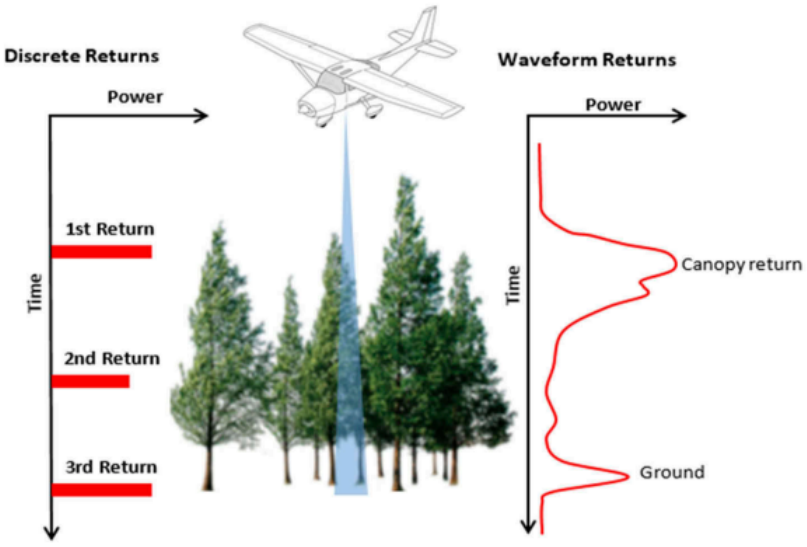


Figure 20: Illustration of discrete return on the left and full-waveform on the right [33].

The result of an ALS acquisition is a 3D point cloud (showing the geometry of the scene with additional attributes). From there, two main steps of data processing are required.

The first step is the classification of the points. Specifically, ground points need to be identified and distinguished from return points from the vegetation and other above-ground objects. This is called point filtering and can be achieved using algorithms. The next step is interpolation, which makes it possible to go from a point cloud to a continuous raster surface. A high-resolution digital elevation model raster can then be created from the ground points. Other classes can be used, for example to characterise the vegetation [31]. A lidar-derived image interpretable by archaeologists is an enhanced visualisation of the DEM obtained from the ALS point cloud. By optimising the algorithms, the vegetation points can be removed while the points associated with archaeological structures are preserved. This creates a DEM of great interest to archaeologists.

5 Description of the data

5.1 Chactún dataset

ALS data of the Chactún area was acquired by the National Centre for Airborne Laser Mapping (NCALM) at the University of Houston. The data was collected in May 2016 and covers an area of 230 km^2 around Chactún. The airborne platform used for the acquisition was a fixed-wing type. The sensor used is a multispectral airborne lidar sensor (Optech Titan). Three wavelengths were used for the laser beam: 1550 nm, 1064 nm and 532 nm (infrared, near infrared and green). The platform flew at an altitude of between 800 and 900 m. The swath width was 600 m. An overlap of 50% between the flight lines was chosen [34]. This overlap increases the point density and avoids data gaps that can occur, for example, with steep surfaces. Full-waveform recording was achieved.

The NCALM also performed the first step of data processing and created a point cloud from the full-waveform. The centre also performed ground classification and removed the vegetation cover. The last part of processing was achived by ZRC SAZU prior to the internship. It consisted of an additional ground classification and the visualisation of the data. To obtain this final visualisation, the point cloud first had to be interpolated into a raster digital elevation model, from which the visualisation is then calculated. It should be noted that although the vegetation was removed, the remains of human activities were not. The final point cloud then includes parts of the ruined buildings and water reservoirs. The mean density of the point cloud, with the three laser channels combined, is 12.8 $ground\ points/m^2$. The spatial resolution of the resulting DEM is 0.5 m [34]. It has been shown that these orders of magnitude for point density and resolution provide a large amount of archaeological information [35]. When artificial and natural structures are part of the model, it is referred to as a Digital Surface Model (DSM). A model of the bare Earth's surface, on the other hand, is called a Digital Terrain Model (DTM) [35]. In the present case, only the vegetation was removed. Archaeological features such as walls and roads are still present in the model. The interpolation is then neither a true DSM nor a true DTM, and the more general term of digital elevation model is used. The name Digital Feature Model (DFM) is sometimes also used for such an archaeology-specific model [35].

The visualisation of the DEM used comes from the work of Kokalj and Somrak (2019) [34] (ZRC SAZU). In their paper, they offer a new type of visualisation for ALS data, which is referred to as the Visualisation for Archaeological Topography (VAT), and which is a

combination of several common visualisations. The idea behind this is to define what could be a standardised visualisation for archaeological purposes. The common visualisations are the following: hillshade, slope gradient, sky-view factor and positive openness. They are shown in Figure 21, where the last two visualisations are the ones proposed by ZRC SAZU and are explained below.

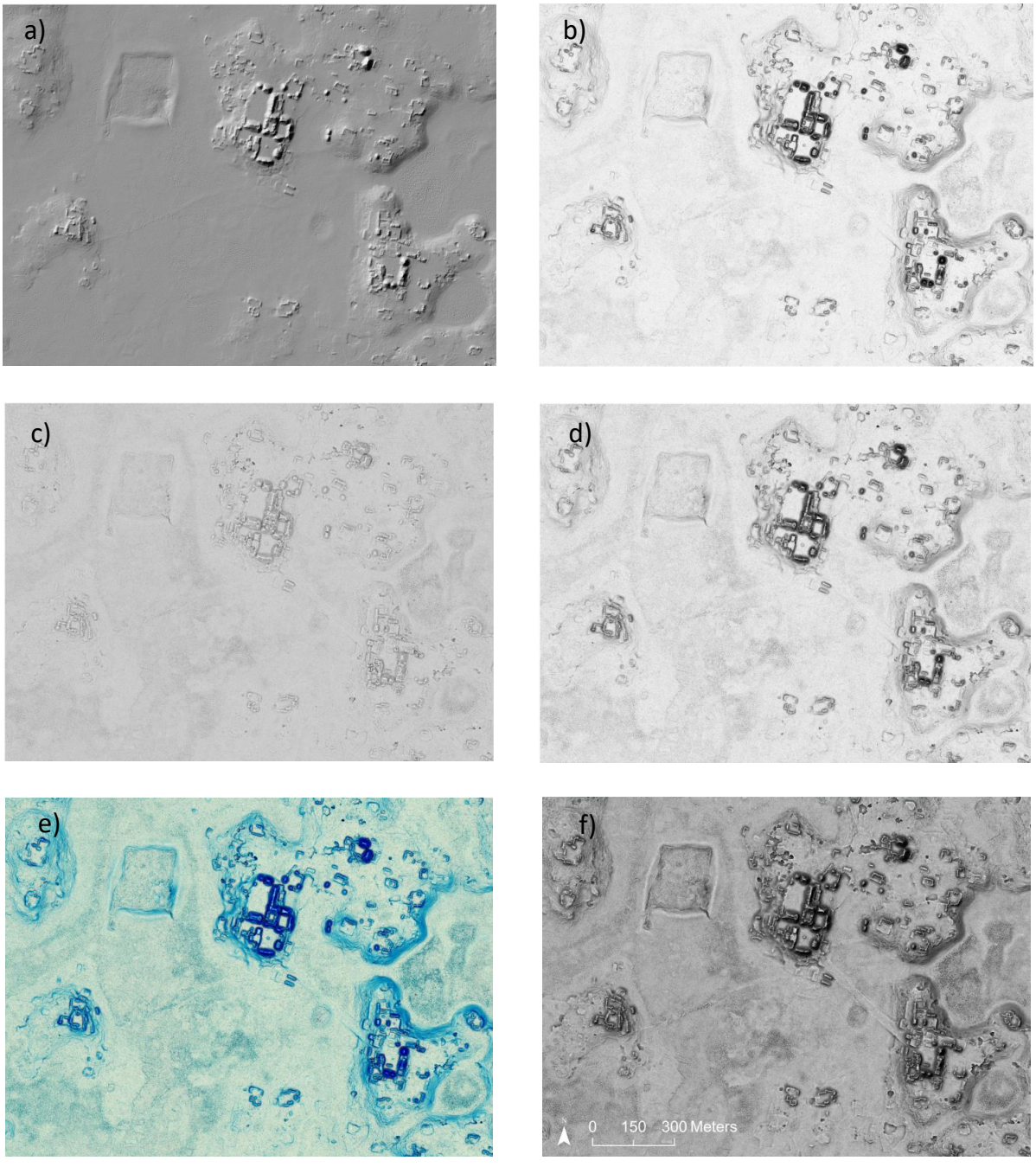


Figure 21: Visualisations of Chactún obtained from the Relief Visualisation Toolbox [34]. (a) Hillshading, (b) slope gradient, (c) positive openness, (d) sky viewing factor, (e) three-band VAT and (f) one-band VAT.

The most common visualisation technique for archaeological purposes has long been the hillshade. It provides a view of the shadows and bright areas that are created when the Sun is simulated in a particular direction. The slope gradient visualisation provides information about the steepness of the terrain. It is obtained by calculating the first derivative of the DEM. Sky-view factor provides the portion of the sky that is visible from a given location. Positive openness is obtained by taking the mean of the zenith angles of the lines tangent to the surface. This is illustrated in Figure 22. In other words, the openness indicates how a location can be seen by an external observer. If nadir angles are considered, the negative openness is obtained [36].

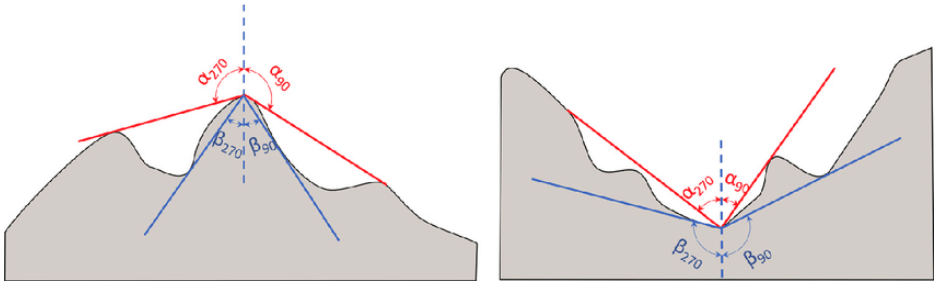


Figure 22: Illustration of positive openness with the α angles and negative openness with the β angles [37].

The visualisation proposed by ZRC SAZU is called the Visualisation for Archaeological Topography (VAT). It combines the different visualisations mentioned above through the use of blending modes. These allow two images to be merged by applying an equation to the two matching pixels when the images are superimposed. The blending modes used to obtain the VAT are Multiply, Overlay and Luminosity. Multiply applies a multiplication between the luminance values of the bottom and top pixels. This makes the image darker. Overlay provides an enhanced contrast (better visibility of bright and dark areas). The Luminosity mode keeps only the luminance of the top layer and the colours of the bottom layer. For each mode, an opacity value must be set for the top image. Otherwise, the bottom image will just disappear entirely [34].

The Table 3 contains information about the creation of the VAT, which was achieved prior to this work. The blending was made starting from a hillshading layer, which was blended with a slope visualisation using the luminosity method. The resulting layer was then merged with the positive openness through overlay. Finally, the resulting layer was blended with the sky-view factor through the multiply mode. The result of these steps is the combined VAT image. Two VAT were actually created, one with one band (greyscale) and one with three bands. The one-band VAT was created using the workflow of Table 3, which provides a greyscale image. The three-band VAT is created by combining slope gradient, sky-view factor and positive openness, using each visualisation as one band. The reason why hillshading was not used for this VAT is that hillshading depends on the orientation, which makes it less usable for data augmentation. The VAT image shows all the positive characteristics of the raster visualisations from which it was created. It improves the recognition of objects and makes the interpretation of the image more straightforward. One can conclude that the proposed VAT meets the criteria of a good visualisation. These

criteria include discrimination of small features, ease of interpretation, invariance of terrain type and object shape (which makes it possible to obtain information about the size of features), absence of artefacts and so more [34].

Visualisations	Blending type	Opacity
Sky-view factor	Multiply	25%
Positive openness	Overlay	50%
Slope	Luminosity	50%
Hillshading - 315° azimuth	Base layer	

Table 3: Visualisations combined to give the VAT along with the blending mode used and opacity [34].

The visualisation for archaeological topography of the training area (Chactún) is the first element needed for the training of the deep learning model for the automatic detection of archaeological features. This VAT is referred to as the source image in ArcGIS Pro. Note that the source image can also be any other visualisation. The DEM of Chactún will also be used later on to investigate the impact of the visualisation. To train the model, a second dataset is needed that contains a set of manually labelled objects (archaeological features) that the model can train and learn with. These objects are stored as polygons in a vector file and are called ground truths or annotations. The distribution of objects over the Chactún area is shown in Figure 23. Three types of archaeological features have been identified at Chactún area: buildings, platforms and aguadas. The labelled dataset contains 8,658 buildings, 2,016 platforms and 52 aguadas. Hence a total of 10,726 Maya archaeological features. Each class is stored in its own layer.

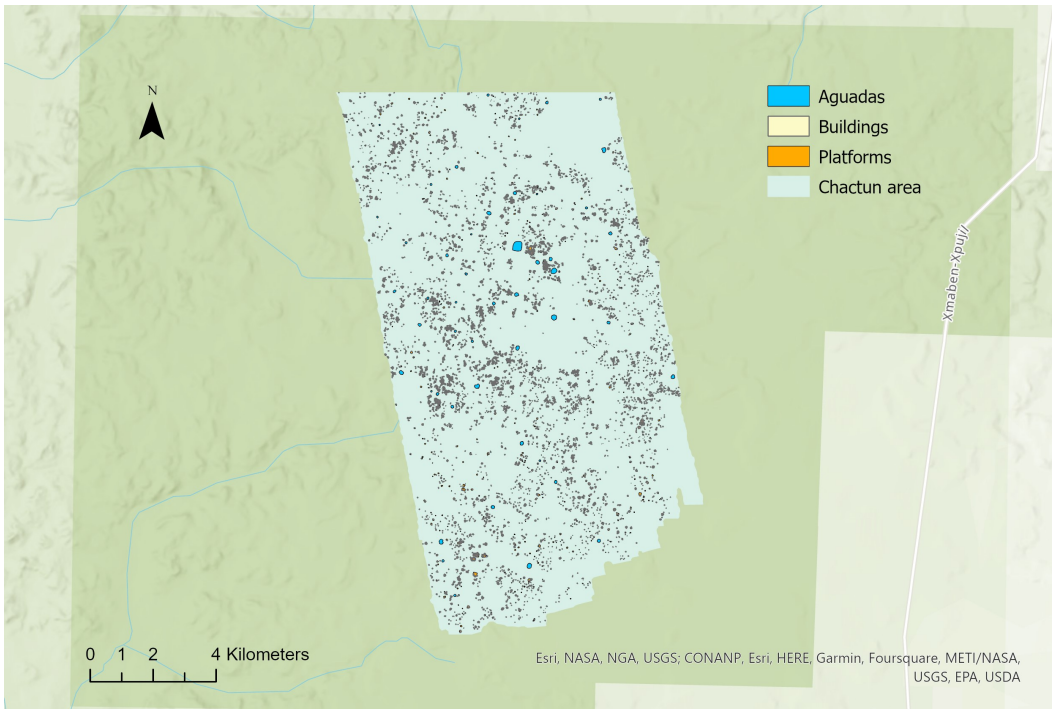


Figure 23: Map of the polygons of archaeological features of the Chactún area.

A particularity of this vector data is that the annotations linked to the objects follow their contours (see Figure 24). This enables semantic segmentation which is not possible with centroids or bounding boxes. An enlargement of the different features, which are not easily distinguishable on Figure 23, can be seen in Figure 24. It can be viewed that buildings often stand on platforms, although it is not always so. The reverse is also true, as not all platforms have a building on top. One should bear in mind that the building polygons actually correspond to parts of a building, e.g. the walls. Pyramids and ball-courts are also labelled as buildings. Hence one polygon corresponds to one structure, or several structures if the boundaries were too difficult to determine [5].

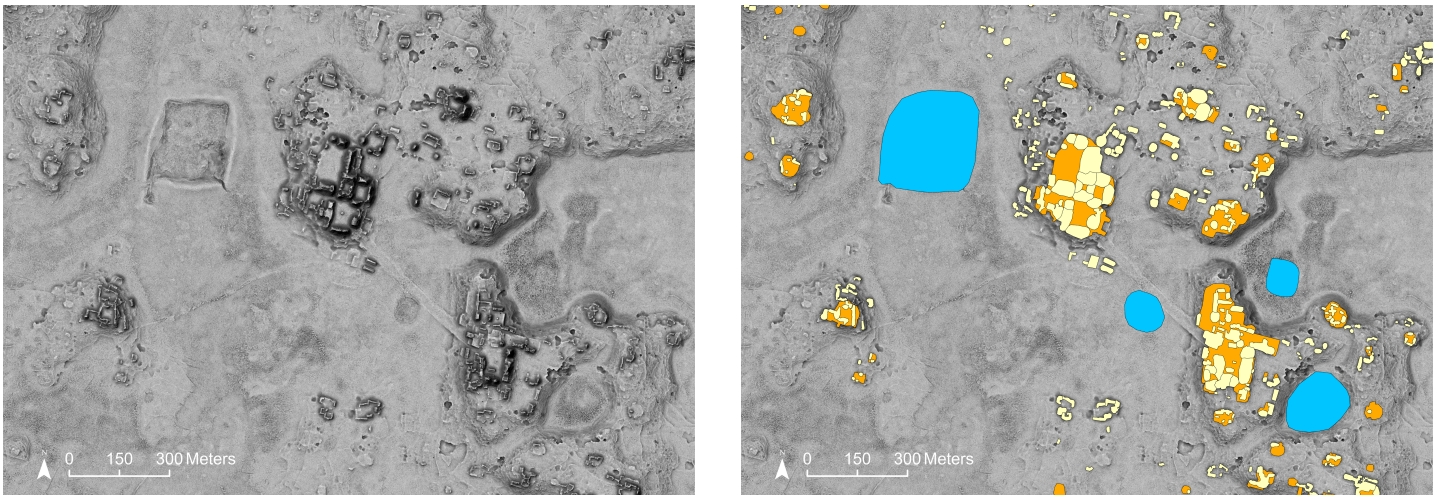


Figure 24: VAT of Chactún core on the left and ground truth archaeological features superimposed to the VAT on the right. Aguadas are in blue, buildings in yellow and platforms in orange.

The raster visualisations for the Chactún area considered in this thesis have the properties shown in Table 4.

Raster name	Bit depth	Range	Number of bands	Pixel size
One_band_VAT	8-bit	0-255	1	0.5m
Three_bands_VAT_32bit	32-bit	0-1	3	0.5m
Three_bands_VAT_8bit	8-bit	0-255	3	0.5m

Table 4: Properties of the raster visualisations of the Chactún area.

5.2 G-LiHT dataset

Other datasets were used to test the deep learning models. It is important to consider an area that the model has not seen during training to avoid bias in evaluating the model's performance. The first dataset used is referred to as G-LiHT, which stands for Goddard-Lidar, Hyperspectral and Thermal imager [38]. Even though G-LiHT refers to the instrument used for the data acquisition (multi-sensor airborne imaging system of NASA's Goddard Space Flight Center), the name G-LiHT will be used here to refer to the test

area. The dataset is also located in Campeche in the Yucatan Peninsula of southern Mexico and was collected in 2013. As for Chactún, the area is covered by tropical forest. The system's lidar sensor includes a 1550 nm laser. The flying height was 335 m. Only a small sample of the total acquired dataset is used in this thesis. The location of the sample area is visible in Figure 25. The central stripe is the VAT of the investigated area. A zoom is shown in Figure 26. The VAT resulting from the ALS point cloud has a resolution of 1 m.

Figure 25: G-LiHT dataset for the testing of the model. The central stripe is the VAT of the area considered.

An important aspect of the data acquisition is that it was achieved by environmental scientists. The consequence is that the ALS data was not collected with archaeological research in mind. The main objective of the data collection was indeed to measure the forest carbon stocks [38]. The scale and location of the sample, as well as the lidar acquisition parameters (PRF, beam footprint, etc) are therefore not optimised for an archaeological objective. For example, the data was acquired with a single pass of the lidar sensor. This may have lead to a too small amount of ground points to resolve archaeological features the most optimal way. This might impact the performance of the deep learning model when applied to this data. One can already notice with Figure 26 that the terrain of G-LiHT is quite different from that of the Chactún area. Due to the rugged terrain of G-LiHT, the archaeological structures are less distinguishable. The question now is whether or not this affects the performance of the deep learning model.

The G-LiHT dataset contains anthropogenic features, which were documented by Witschey and Brown (2010) [39]. These ground truth objects, however, might not have been verified and could lack reliability and precision [38]. The polygons of the known archaeological sites (ground truths) can be seen on the right side of Figure 26. The dataset contains 60 platforms, 795 buildings and 3 aguadas. In total there are 858 objects.

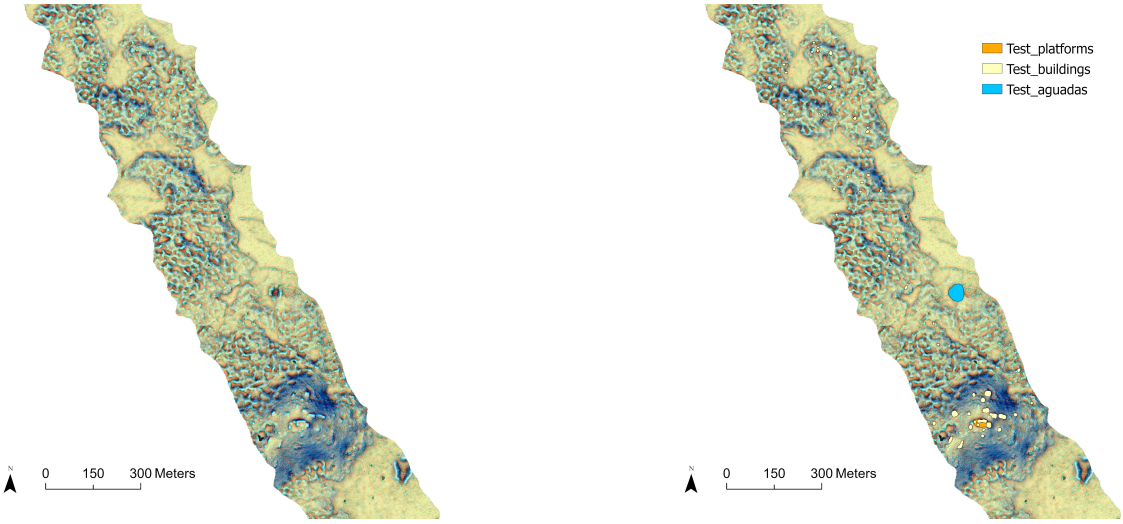


Figure 26: Three-band VAT on the left and ground truths superimposed to the VAT on the right.

5.3 Holmul dataset

Another testing set was available to test the performance of the models and their generalisation (also referred to as transferability). The data cover a 14 km^2 area of the Holmul Maya archaeological site in Petén, Guatemala. The area can be seen in Figure 27. This data is a sample of the dataset acquired in 2016 during the Pacunam Lidar Initiative (PLI). During this survey, $2,144$ km^2 of the Maya Biosphere Reserve in Guatemala were mapped. The data collection was achieved by scanning the terrain at six different view angles. The flying height was 650 m. The resolution of the DEM obtained from the point cloud is 1 m [40].

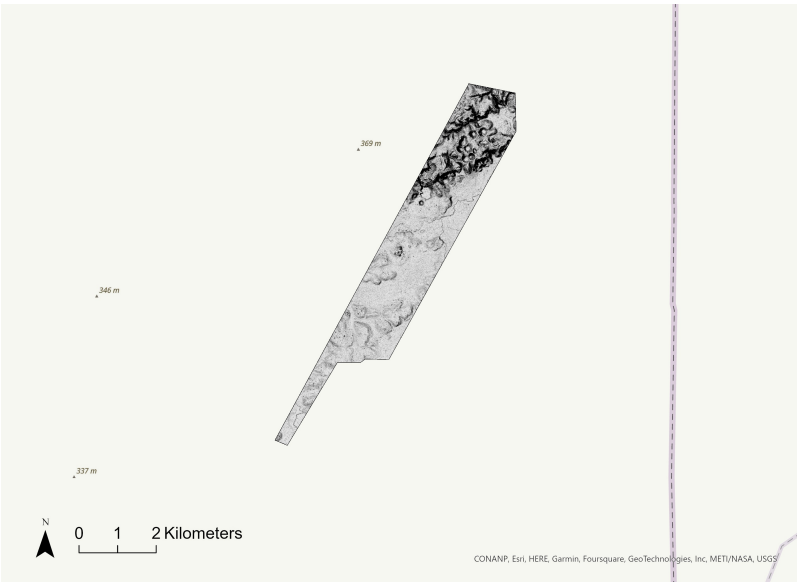


Figure 27: One-band VAT of Holmul testing area.

The Holmul area contains remains of archaeological settlement. Agricultural features, as well as defensive features and causeways are found on the site. Again, ground truth polygons were available to compare the predictions with reality. These ground truths were verified by pedestrian surveys during the year 2017 [40]. The test set over the Holmul region contains only 472 buildings (no platforms and aguadas). However, the ground truth objects for this test area were digitised in a different way. Indeed, the previous polygons followed the boundaries of the buildings and platforms in a precise way. However, the Holmul ground truth objects are polygons with a more rectangular shape, that define the buildings with their original, unruined shape. This will be seen later in Section 11. The data and ground truth polygons were kindly provided by Francisco Estrada-Belli of the Middle America Research Institute at Tulane University.

6 Deep learning in ArcGIS Pro

The ArcGIS Pro software offers a range of tools to apply deep learning to images. Both object detection and semantic segmentation can be performed. In order to detect archaeological remains in a new area, the deep learning model needs to be trained on a known area where remains have already been identified, labelled and stored. A set of tools is available to go from the data described in the previous section to automated predictions for a new area. This is achieved in three steps:

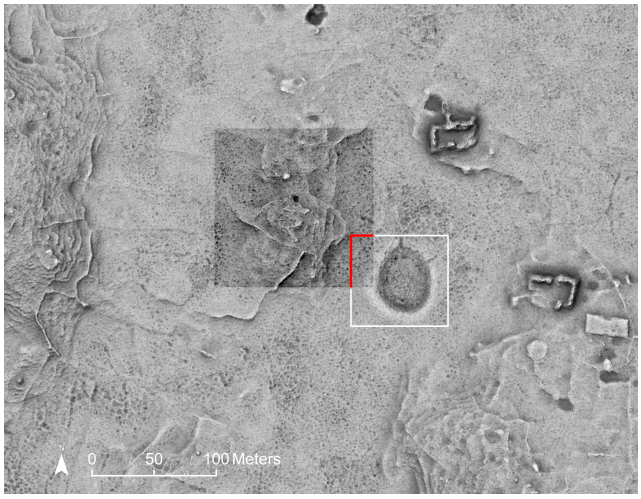
- Creating the training data
- Training the model on that data
- Testing the trained model on a new dataset

The tools available for this are described in the following sections: "Export Training Data for Deep Learning", "Train Deep Learning Model", "Detect Objects Using Deep Learning" and "Classify Pixels Using Deep Learning". Depending on whether the goal is object detection or semantic segmentation, only one of the last two tools needs to be used.

6.1 Creation of the training data

First, the creation of the training data in the case of object detection is described. However, it is very similar to the case of semantic segmentation described later. The training data is created from the two available datasets: the VAT and the vector file with the archaeological ground truth features. ArcGIS Pro provides a tool to create the training data from these two datasets: "Export Training Data For Deep Learning". The tool is used in batch mode when multiple layers are used as input. Here, one layer is assigned to one archaeological class (buildings, platforms or aguadas). Therefore, the tool is used in batch mode when all three feature classes are considered simultaneously. The tool provides as output a set of patches, also called image chips or tiles, and the objects contained in these sub-images (objects refer to archaeological features). A patch is shown in Figure 28 (darker square). Each patch is assigned to one or several bounding boxes that contain the objects. Hence, along with the patch there is a label file that contains metadata about that patch and the object(s) it contains. This information about the object is provided by information about the bounding box. The Figure 28 shows the information contained in the label file. The position of the bounding box is calculated from the top left corner of the patch.

Here only one bounding box has been defined in the patch (actually part of a bounding box), but there can be multiple bounding boxes within an individual patch. Using the values given in the label file, the bounding box was roughly drawn on the image (red line). Only part of the bounding box is found within the patch and the object contained within the bounding box is an aguada. The white square is an estimated representation of the whole box for illustration. Note that the depth is equal to the number of channels in the input data. The value is then 1 for the one-band VAT and 3 for the three-band VAT.



Size of chip	Width	256
	Height	256
	Depth	1
Object bounding box	xmin	231.65
	ymin	194.85
	xmax	256.00
	ymax	256.00

Figure 28: Illustration of a patch (image chip) with the one-band VAT on the left. The patch corresponds to the darker square and the white square is the bounding box of the aguada. Table with information about the patch on the right. The bounding box coordinates correspond to the red part of the square.

The Export Training Data tool depends on several parameters which must be entered by the user. They are listed in Table 5, together with the initial settings chosen for the present work. The *Input Raster* is the VAT of Chactún. The *Output Folder* specifies the location where the training data is stored after the operation. The *Batch Input Feature Class* corresponds to the vector files containing the polygons. The *Image Format* is selected as TIFF. The *Tile Size* is the size of the patches in pixels. The default value is 256×256 pixels, or 128×128 metres. The *Stride* corresponds to the overlap between two adjacent tiles. It has been set to half the *Tile Size* to give a standard overlap of 50%. This choice increases the number of patches, and thus the training data. The parameter *Output No Feature Tiles* allows to export only those patches that contain a feature, thus reducing the data volume. The *Metadata Format* depends on the deep learning model to be used. For the RetinaNet model, the PASCAL Visual Object Classes must be selected as the metadata format. The *Class Value Field* is specified when several object classes are considered. We did not use data augmentation, but the *Rotation Angle* can be interesting if the amount of data is insufficient. If one selects a value of 90° the data will be multiplied by a factor of four. *Reference System* corresponds to the coordinate system, which can be either the image space or the map space. *Buffer Radius* will add a buffer of a certain width around the patches. *Crop Mode* specifies if the patches are cropped to a same size

(Fixed_size mode) or if they are cropped to match the bounding boxes (Bounding_box mode) [41].

Input Raster	Path to file
Output Folder	Path to folder
Batch Input Feature Class	Aguadas;Buildings;Platforms
Image Format	TIFF
Tile Size X	256
Tile Size Y	256
Stride X	128
Stride Y	128
Output No Feature Tiles	ONLY_TILES_WITH_FEATURES
Metadata Format	PASCAL_VOC_rectangles
Class Value Field	-
Rotation angle	0
Reference System	MAP_SPACE
Processing Mode	PROCESS_AS_MOSAICKED_IMAGE
Buffer Radius	0
Crop Mode	FIXED_SIZE

Table 5: Parameters of the Export Training Data tool [41].

During the creation of the training data, four other files are created in addition to the patches and the label files. They contain some statistics about the data that are useful for the training of the deep learning model. Some of this statistical information is listed in Table 6. Note that the patch is called a tile in the table.

Number of tiles	24,634	
Number of classes	1	
Number of features	61,629	
Number of features per tile	Min	1
	Max	26
	Mean	2.5018
Feature area	Min	6.1e-05
	Max	16,384
	Mean	389.3750
	Sum	23,996,790.82

Table 6: Statistical information from the Export Training Data tool for object detection for the object class archaeological features.

As mentioned earlier, the output training data is a set of patches. They can be seen in the Appendix C. One might wonder why the image is divided in this way and not provided to the deep learning model as a whole. This has to do with computational efficiency, but also with the definition of the training samples. For CNNs, a training sample is in fact not one polygon but one patch. The more training samples available, the better the performance of the model and the lower the overfitting. The kernels are applied to each of

these patches, which are the input training images. The input is then not just one image, but a set of thousands of smaller images.

The creation of the training data for semantic segmentation follows the same process as for object detection. The difference with object detection resides in the metadata format, which is set to "classified tiles". The output of the export training data tool also contains an additional element. For each patch, a label mask (also called a segmentation mask) is assigned instead of a text label file. If an object is located at a certain position in the patch, the value of the mask pixel is equal to 1. If the pixel belongs to the background, the value is 0. An example can be seen in Figure 29, where the white pixels correspond to the aguada previously seen in Figure 28. The set of patches and their corresponding masks are fed to the deep learning model.

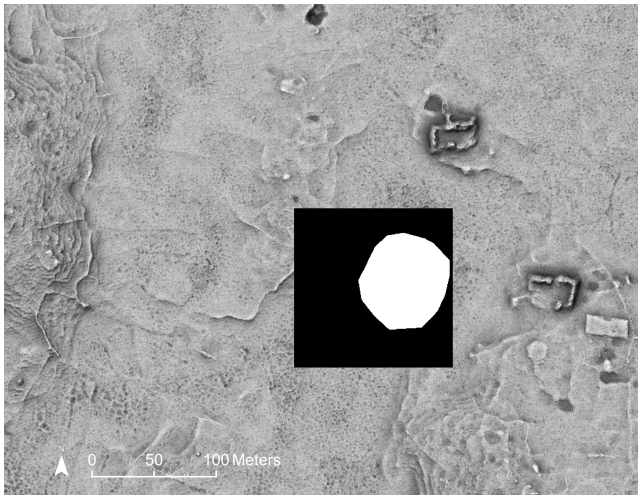


Figure 29: Mask associated to one patch, obtained during the export of the training data.

6.2 Training of the deep learning model

Once the training data is ready, the deep learning model can be trained with another tool available in ArcGIS Pro: "Train Deep Learning Model". The parameters are listed in Table 7 for the case of object detection. The settings given in the table are the first ones chosen to train a model. *Input Training Data* refers to the training sample data obtained previously. *Max Epochs* corresponds to the maximum number of times the entire dataset is processed. The model is in fact trained several times until it achieves good enough performance. The entire dataset is used during one epoch. The batch size indicates how many training samples, i.e. how many patches, are processed at once. The gradient descent is then applied to each batch and we speak of mini-batch gradient descent. The model parameters are updated after each batch gradient descent. *Model Arguments* are parameters that are required for RetinaNet to work. The four arguments have already been explained in Section 3.4. If the *chip_size* is smaller than the *Tile Size*, the images are cropped for training. The *Learning Rate* defines the pace at which the weights are updated. No prior value is assigned in order to let the model choose the optimal learning rate to achieve the minimum loss. *Validation %* is the percentage of the input training data that is used for validation. The usual value is 20%. The model has not been frozen, which means that the

backbone parameters can be changed to better fit the new model. Finally, the *Parallel Processing Factor* indicates that the operations are divided into several processes in the graphics card.

Input Training Data	Path to folder
Output Model	Path to folder
Max Epochs	20
Model Type	RetinaNet
Batch Size	64
Model Arguments	scales '1, 0.79, 0.63';ratios '0.5, 1, 2'; chip_size 256;monitor valid_loss
Learning Rate	-
Backbone Model	ResNet34
Pre-trained Model	-
Validation %	20
Stop when model stops improving	STOP_TRAINING
Freeze Model	UNFREEZE_MODEL
Parallel Processing Factor	100

Table 7: Parameters of the Train Deep Learning Model tool [42].

Another parameter worth mentioning, over which the user has no control in the ArcGIS Pro Graphic User Interface, is the IOU threshold. This Intersection Over Union threshold is automatically set to 0.1. It is used to compare a ground truth box with a prediction box. The threshold defines when a prediction box is considered a true positive. So a bounding box with an IOU of more than 0.1 is considered a true positive here, while a smaller IOU indicates a false positive.

The Table 7 shows the parameters chosen for the first model investigated. Most of the parameters remain the same for the different models. The only ones that change are the backbone model and the batch size. The batch size has to be reduced for some models because the GPU memory is too small (8 GB).

For semantic segmentation, only the model type and model arguments differ. The model type is then U-Net and its arguments are those mentioned in Section 3.5 (class balancing, mixup, focal loss, chip size and monitor).

6.3 Testing of the deep learning model

In the case of object detection, to test the previously created deep learning model, one can use the tool "Detect Objects Using Deep Learning". The parameters are listed below with the usual values.

Input Raster	Path to file
Output Detected Objects	Path to file
Model Definition	Path to file
Arguments	padding 64; threshold 0.5; nms_overlap 0.1; batch_size 64; exclude_pad_detections True
Non Maximum Suppression	NMS
Confidence Score Field	Confidence
Class Value Field	-
Processing Mode	PROCESS_AS_MOSAICKED_IMAGE
Parallel Processing Factor	100

Table 8: Parameters of the Detect Objects tool [43].

Input Raster provides the image of the new area where one wants to detect objects. *Output Detected Objects* is the path to the vector file with the detected objects, in particular their bounding boxes. *Model Definition* specifies the location of the deep learning model (.dlpk file) to be used. *Arguments* are selected to improve the detection. *Padding* specifies the area added to the edge of the image for better analysis. *Threshold* is the minimum level of confidence allowed in the detections. *Nms_overlap* indicates the maximum overlap at which the bounding boxes should be merged. *Exclude_pad_detections* removes detections that come exclusively from the padding zone. The *Non Maximum Suppression* parameter allows duplicate objects to be removed. That is, a feature described by two bounding boxes. The box with the smallest IOU and the lower confidence is removed. *Confidence Score Field* specifies the name of the field in which the confidence values are to be stored. *Class Value Field* is again specified if there is more than one class of features. The output of the object detection tool is a feature layer that contains the objects that the model has detected. More specifically, it contains the bounding boxes associated with the objects [43].

To test the deep learning model for semantic segmentation, another tool must be used. The new tool to be run is "Classify Pixels Using Deep Learning". It classifies an input raster by applying the U-Net deep learning model to it. Each pixel is assigned a class label. The parameters are similar to the previous tool "Detect Objects" and are summarised in the following table with the usual values. The argument *test_time_augmentation* can be set to True to increase the amount of data for testing. Several additional test patches will then be created through rotation and cropping. The final prediction will be the average of the predictions from the different versions of the patch image [44].

Input Raster	Path to file
Output Classifier Raster	Name of output raster
Model Definition	Path to folder
Arguments	padding 64; batch_size 10; predict_background True; test_time_augmentation False; tile_size 256
Processing Mode	PROCESS_AS_MOSAICKED_IMAGE
Parallel Processing Factor	100

Table 9: Parameters of the Classify Pixels tool [45].

7 Object detection with the G-LiHT test set

We trained a series of deep learning models for object detection and semantic segmentation using the data described in Section 5.1. Several parameters were modified to determine their impact on the performance of the models. In addition, different visualisations were used for training. Finally, different test sets were used to evaluate the final performances and to investigate the generalisation of the models. The different experiments achieved will be described in the following sections. The workflow diagram of each experiment is provided. Note that the experiments are numbered in the order they were achieved. The general workflow diagrams that summarise all the computations for object detection and semantic segmentation are shown in Appendix D. The following sections of this thesis are divided according to the dataset used to test the models. The test set is here the one referred to as G-LiHT dataset.

7.1 One-band VAT

Deep learning models were initially created from the one-band VAT. The Figure 30 shows the workflow of Experiment 1. As always, the first step is to create the training data using the Export Training Data tool. This tool was first used in batch mode, considering that all archaeological objects belong to the same class (thus the different labels of buildings, platforms and aguadas are not taken into account). This decision was made because of the overlap between the different objects. The Train Deep Learning Model tool was then used to train the model. The RetinaNet model is used for object detection. This model was trained with different backbones during Experiment 1: ResNet-18, ResNet-34 and ResNet-50.

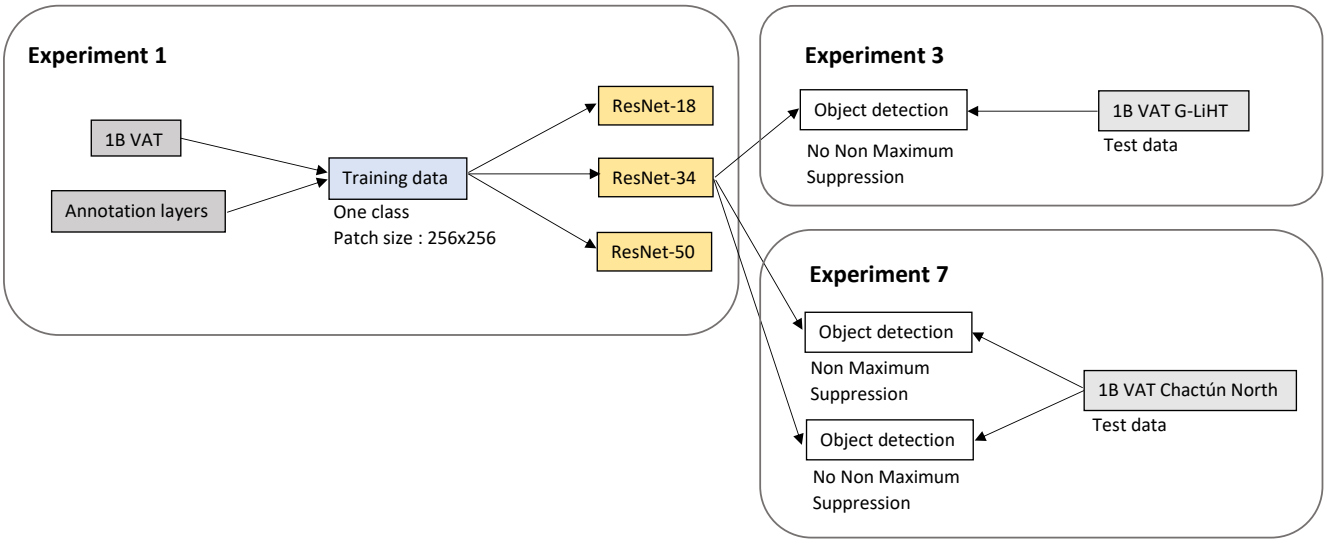


Figure 30: Workflow of experiments 1, 3 and 7.

The first model used ResNet-34 and a batch size of 64. The results of the training are detailed hereafter. The minimum and maximum learning rates were 4.79e-05 and 4.79e-04,

respectively. The accuracy of the ResNet-34 model was 32.9%. This value is quite low and other model architectures should therefore be investigated. The training and validation losses can be seen in Figure 31. The decrease and stabilisation of the losses over time is a requirement for a good model. Indeed, the decrease of the training curve denotes the learning process and the decrease of the validation curve indicates the generalisation [46]. However, the gap between the validation and training losses seen here is too large for the model to be optimal. In fact, an ideal model has no gap between the two losses. The higher validation loss indicates overfitting. That is, the model "sticks" too much to the training examples and does not generalise well on unseen data.

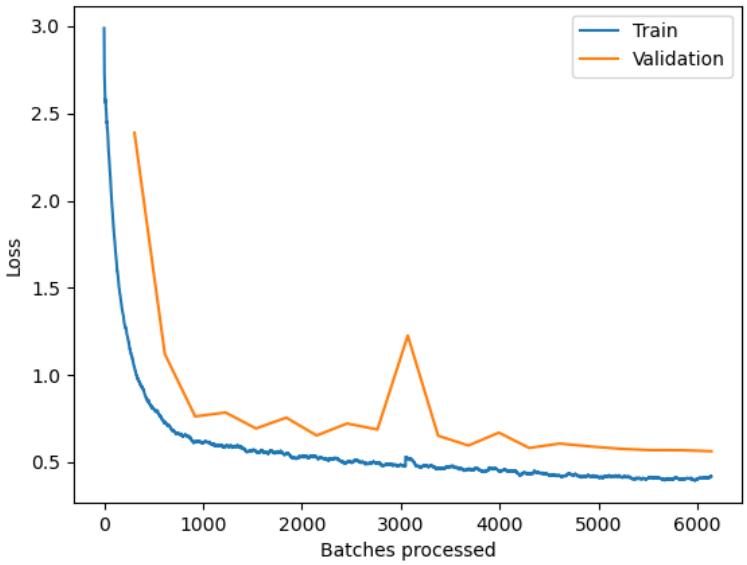


Figure 31: Training and validation losses computed with the ResNet-34 backbone model.

A comparison between the ground truths and the predictions of the model for the validation set is shown in Figure 32. Five different sample areas (randomly selected patches) are considered. All ground truths were predicted by the model. In other words, each bounding box on the left images is recovered on the right images. However, the sizes of the bounding boxes differ slightly. On the third image, additional objects were predicted that weren't included in the ground truth data. Some of them could be new buildings while the majority are false positives. One can notice that the boxes are labelled "Undefined" because no class name was assigned to the training data. In other words, the algorithm gave the data this default class name.

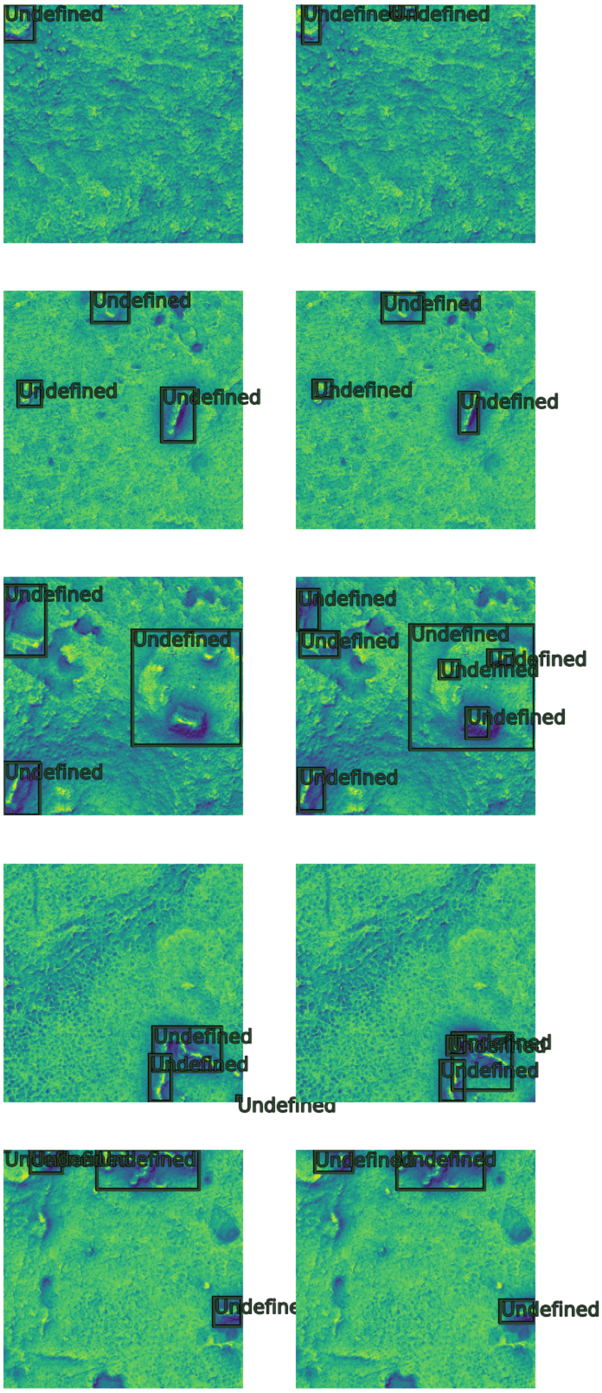


Figure 32: Ground truth objects on the left and predictions on the right for ResNet-34.

The ResNet-50 backbone model has also been trained. It contains more convolutional layers than ResNet-34 and is therefore more complex. The batch size for this model had to be reduced to a value of 30 because the GPU memory was not sufficient. The results were less satisfactory than with ResNet-34, with an accuracy of 30.8%. As can be seen in Figure 33, the validation loss is higher and more variable, indicating a worse

overfitting than ResNet-34. Two sample of the results can also be seen in Figure 33. The first sample shows that the model failed to detect an archaeological feature. The second sample displays shapes of the predicted bounding boxes which are quite different from the shapes of the ground truth boxes compared to ResNet-34. These results, together with the training and validation losses, reveal that the model has fitted the training data too well. The predictions for the validation data are then less accurate. A backbone with 50 layers is too complex and has too many parameters for the number of training data at hand.

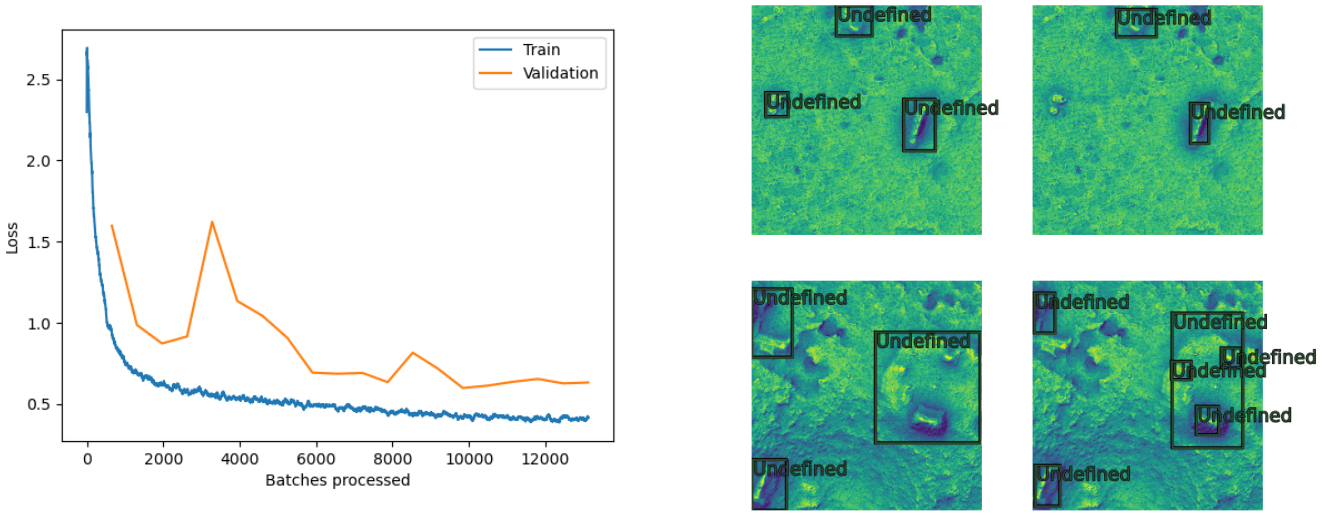


Figure 33: Training and validation losses for the ResNet-50 backbone model on the left. Sample results of the model on the right, with ground truths on the left and predictions on the right.

The model was also trained with the backbone ResNet-18. The results, which can be seen in Figure 34, were better than ResNet-50, but slightly less good than ResNet-34, with an accuracy value of 32.1%. A comparison of the three models can be found in Table 10. From the values, it can be seen that the ResNet-34 model is the best for the one-band VAT. One could argue that the training loss is lower for ResNet-50, however this is not what we are looking for in a deep learning model. In fact, the most important loss is the validation loss, as one wants to have a generalised model and detect objects in a new area. A very low training loss only indicates overfitting. One can also notice that the values of the losses obtained for the three models are quite high. This will be particularly evident when examining the semantic segmentation later on.

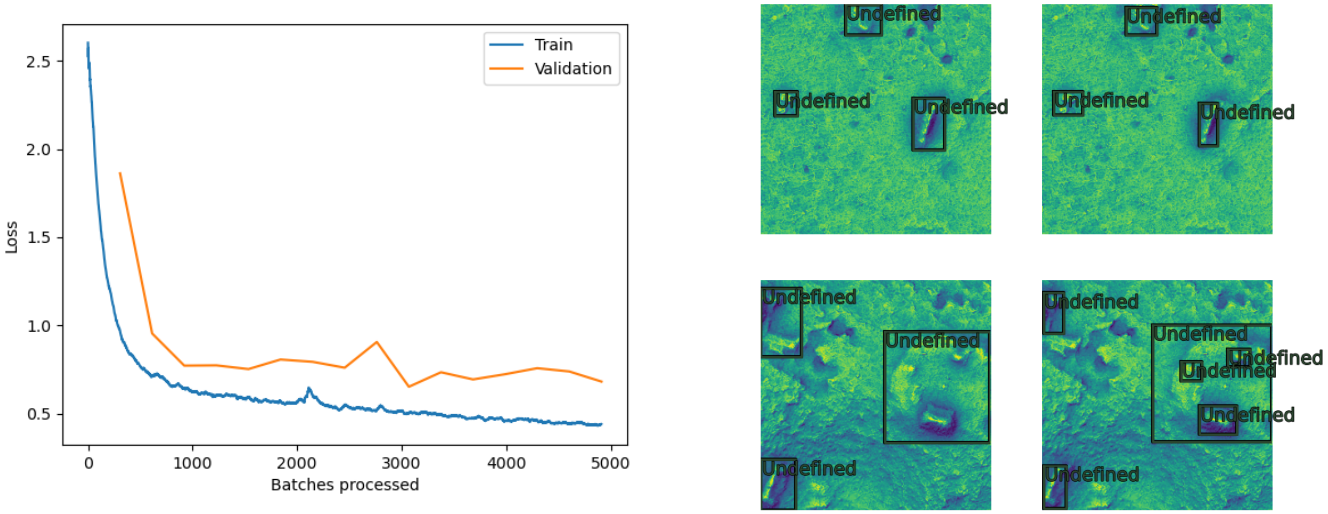


Figure 34: Training and validation losses for the ResNet-18 backbone model on the left. Sample results of the model on the right, with ground truths on the left and predictions on the right.

Backbone architecture	Training loss	Validation loss	Accuracy
ResNet-18	0.6702	0.6819	32.1%
ResNet-34	0.5660	**0.5626**	**32.9%**
ResNet-50	**0.5121**	0.6322	30.8%

Table 10: Comparison of the results for the one-band VAT. The lowest losses and highest accuracy are in bold.

During Experiment 2, for which the workflow can be seen in Figure 35, another set of training data was created. This set is made for multiclass object detection with a "classvalue" attribute (1 for aguadas, 2 for buildings and 3 for platforms), to account for the three different classes in one model. The "Class Value Field" parameter in the ArcGIS Pro tools was therefore defined as "classvalue". In this way, the aguadas, buildings and platforms would be treated as different objects. Since ResNet-34 turned out to be the best backbone, we chose to train this model.

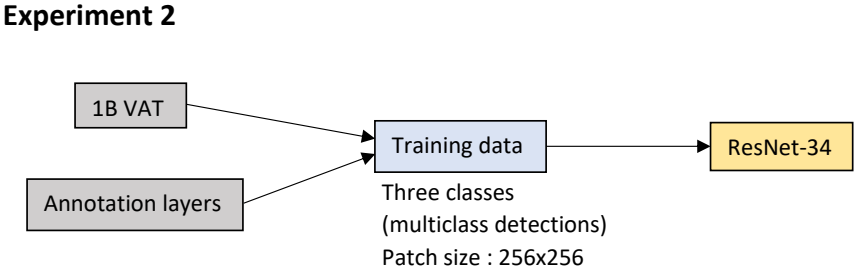


Figure 35: Workflow of experiment 2.

The results of the training, summarised in Table 11, were less satisfactory than with a single class for all objects. The accuracies for buildings and platforms are indeed very low. This is certainly because these two feature classes often overlap. The sample of the results in Figure 36 shows a platform that was wrongly classified as a building. One can notice that the main difference between the previous models and this model lies in the number of neurons in the output layer. For only one class, there is only one output neuron. It provides the probability that the object is an archaeological feature. For three classes, on the other hand, there are three neurons in the output layer, which indicate the probability that the feature belongs to one of the three classes. The highest probability gives the output.

	Aguadas	Buildings	Platforms
Accuracy	50.0%	18.1%	15.0%
Training loss	0.6889		
Validation loss	0.6476		

Table 11: Accuracies, training and validation losses for the ResNet-34 model with three classes.

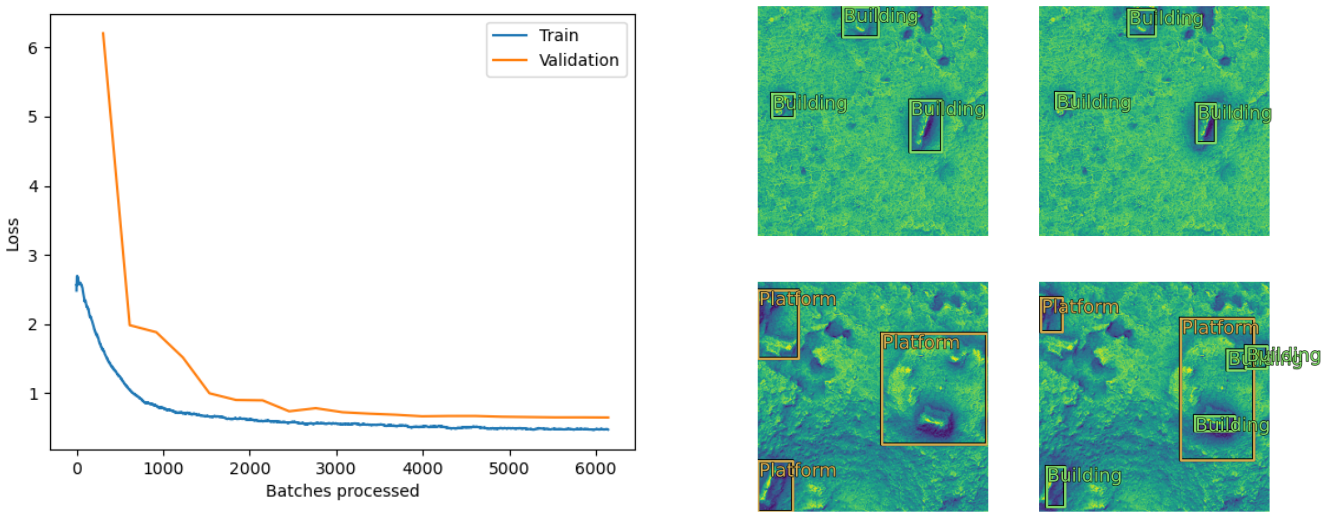


Figure 36: Training and validation losses for the ResNet-34 backbone model with three classes on the left. Sample results of the model on the right, with ground truths on the left and predictions on the right.

Even though the results of the training for the four models trained so far are not very satisfactory, one can investigate how the best model generalises to a new dataset. To assess the final performance of the best model (ResNet-34 with one class), it was applied to the G-LiHT testing dataset. This is Experiment 3 in Figure 30. As a reminder, the G-LiHT dataset has never been seen by the model. The geographical division of the training and testing data avoids possible overlaps that could arise from a random division and that would bias the results. Since the model was trained on the one-band VAT of Chactún,

the visualisation of the testing area must of course also be a one-band VAT. It first had to be resampled to a resolution of 0.5 m, to match the resolution of the raster used for training. In fact, the original resolution was 1 m. To do so, the Resample tool of the Data Management toolbox in ArcGIS Pro was used with the bilinear resampling technique. This method performs an interpolation by considering the four nearest neighbours of a pixel. The new pixel value is computed from the average of the four neighbouring pixels [47]. A deep learning model performs better when the testing and training sets share the same resolution. The Detect Objects tool, which applies the model to the G-LiHT area, detected 75 objects. These include 52 true positives (11 platforms and 41 buildings) and 18 false positives, some of which could be newly identified buildings. The number of true positives is very small compared to the total number of ground truths, which is 858. A large number of objects have been overlooked, as can be seen in the sample of Figure 37. The reason for this low detection value is threefold. First, the G-LiHT testing terrain is quite different from the training terrain of Chactún. Second, the training set with the polygons that precisely delineate the objects may be more suitable for semantic segmentation than for object detection. Third, the ground truth objects of G-LiHT may not have been precisely delineated in the first place. Hence the data quality is different between the training and testing datasets. We can also note that the data density is different for G-LiHT compared to Chactún.

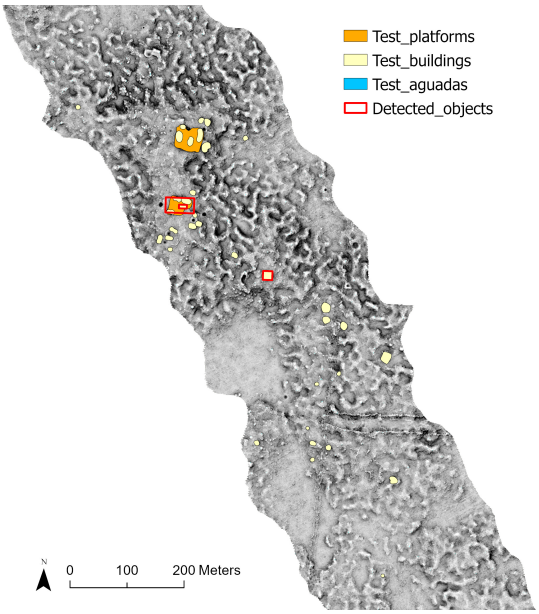


Figure 37: Sample of the results of the object detection for the model with ResNet-34 and one-band VAT.

7.2 Three-band VAT

One might wonder what happens when the three-band VAT is considered for the deep learning model. Are the detections on the G-LiHT dataset more accurate for such a visualisation? Since the raster is made of three bands, the inputs of the deep learning model consist of three arrays of pixels. Experiment 4, which follows the workflow of Figure 38, investigates two backbones when using the three-band VAT of 32bit.

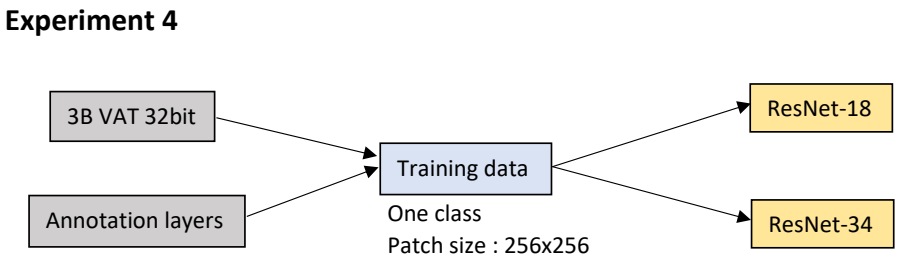


Figure 38: Workflow of experiment 4.

The RetinaNet model with ResNet-34 was trained first with this three-band raster file. The results of the training are displayed in Figure 39. Looking at the graph of the losses, the first thing to notice is the very high factor of 1e33 on the vertical scale. Since this result suggests overfitting (although even with overfitting this high value is unlikely), the model was simplified by reducing the number of layers in the backbone. ResNet-18 was chosen, which contains the smallest number of layers available. The disturbing factor 1e33 did not occur in this model. However, something strange still happened with the losses defined as NaN for several epochs. The graphs of the losses and samples of the results can be found in the Appendix E (Table 76).

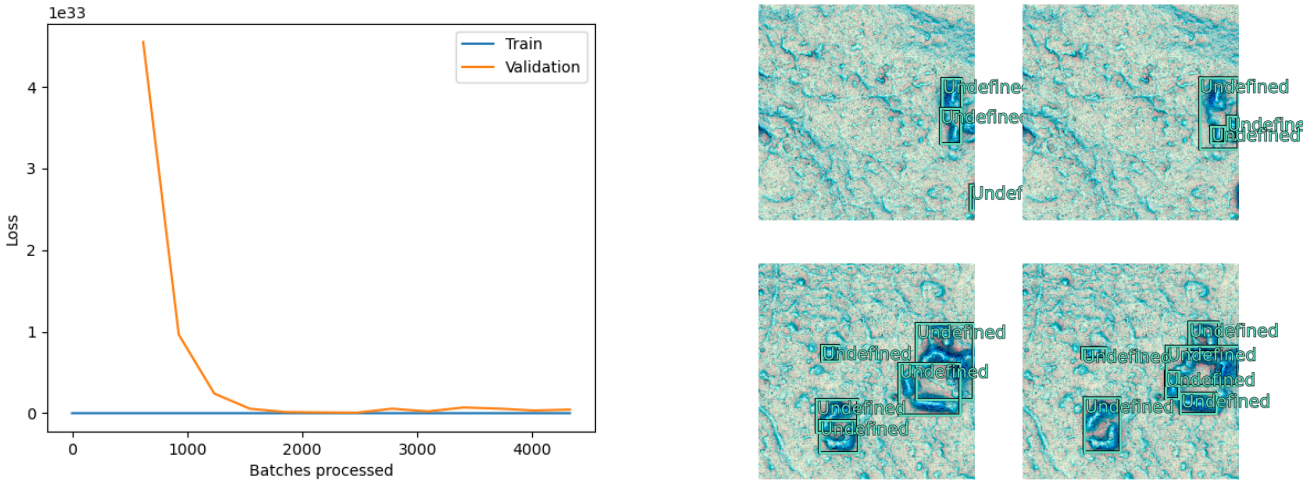


Figure 39: Losses for the ResNet-34 backbone model on the left. Sample results of the model on the right, with ground truths on the left and predictions on the right.

One possible explanation for these bad results was that the raster had a pixel depth of 32bit (float) and not 8bit (unsigned integer) as for the one-band VAT. This explanation was investigated in Experiment 5, for which the workflow can be found in Figure 40.

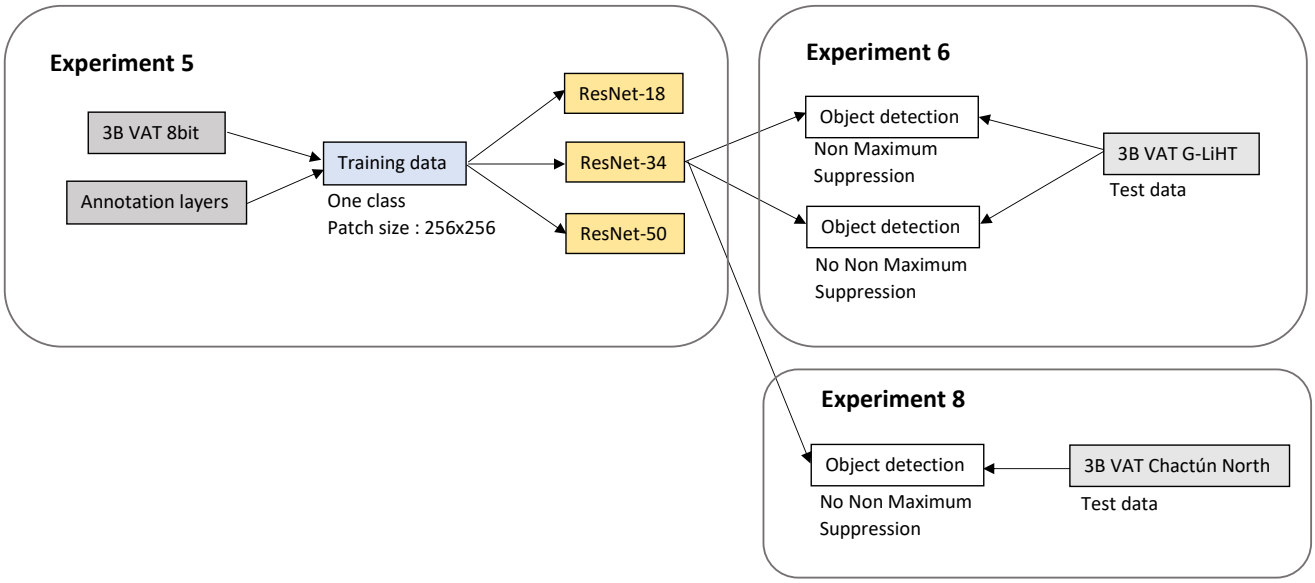


Figure 40: Workflow of experiments 5, 6 and 8.

The three-band VAT was exported as 8bit. The training data was also recreated with this new 8bit raster. The model with ResNet-34 was run with the new data and gave the results shown in Figure 41, with an accuracy of 45.6%. The losses are more plausible than those obtained with the 32bit raster. The accuracy is also higher than the one of the previous models. The samples of the results on the right-hand side of Figure 41 show prediction bounding boxes that differ quite significantly from the ground truths. The model predicted smaller instances (buildings) that were not included in the training set. Therefore, in the bottom right image one sees smaller bounding boxes within a larger one (indicating a platform).

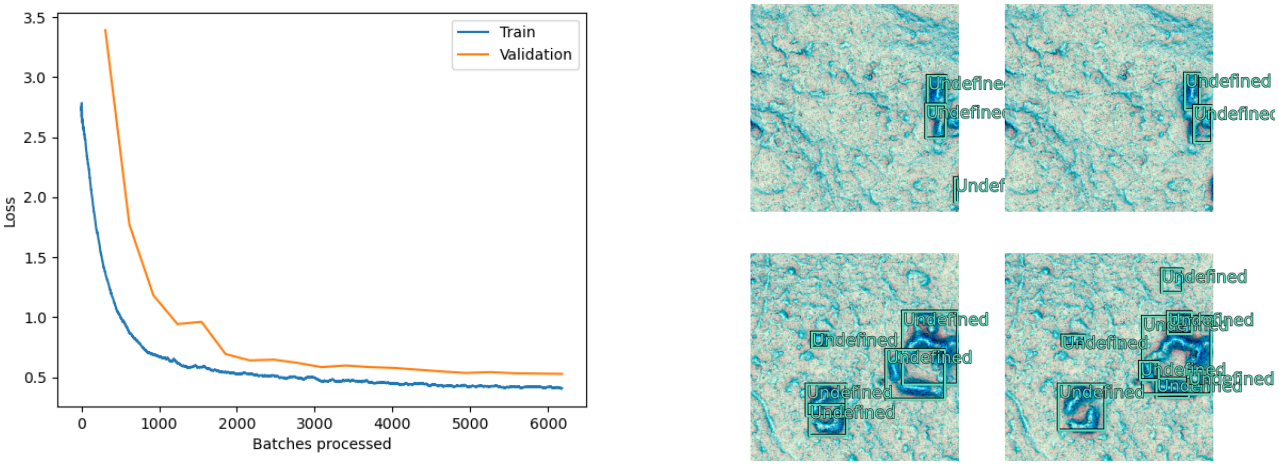


Figure 41: Losses for the ResNet-34 backbone model and three-band VAT 8bit on the left. Sample results of the model on the right, with ground truths on the left and predictions on the right.

ResNet-18 was also examined, reducing the complexity of the model. As well as ResNet-50. Loss graphs and samples of the results for these two backbones can be found in Appendix E (Table 77 and 78). The results for the three backbones are summarised in Table 12. The high accuracy and low validation loss for ResNet-34 suggest that this backbone is more suitable for the case study.

Backbone architecture	Training loss	Validation loss	Accuracy
ResNet-18	**0.4947**	0.5371	43.6%
ResNet-34	0.5123	**0.5271**	**45.6%**
ResNet-50	0.9372	0.8051	44.2%

Table 12: Comparison of the results for the three-band VAT. Smallest losses and highest accuracy are in bold.

Experiment 6 (see Figure 40) allowed to test the ResNet-34 model on the G-LiHT data while investigating the influence of the Non Maximum Suppression (NMS) parameter. The raster of the G-LiHT area also had to be exported from 32bit to 8bit (this is called byte scaling) and resampled to match the 0.5 m resolution of the training raster. However, the object detection results, when using NMS, were not very good as only 77 objects were detected. Among the detected objects there are 57 true positives, 20 false positives and thus 801 false negatives (objects that were not detected). A sample of the results is shown below in Figure 42. The upper platform, which the model failed to detect with the one-band VAT, was predicted with the three-band VAT.

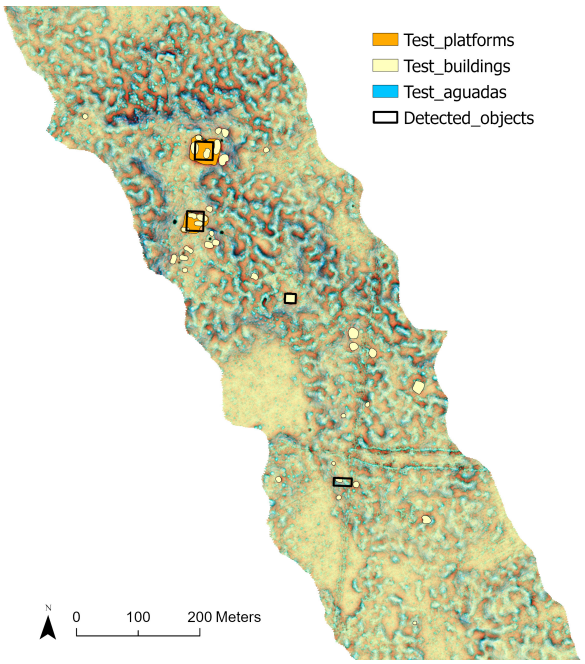


Figure 42: Sample of the results of the object detection for the model with ResNet-34 and three-band VAT 8bit.

Another testing was made without using the NMS parameter. Since buildings and platforms often overlap, using the NMS might remove bounding boxes that are thought

to belong to the same object but actually belong to distinct ones. More objects were detected when overlap was allowed (i.e. the NMS was not used), 86 in total. With the same number of false positives (20), 62 true positives were detected. This is five more than previously. As can be seen in Figure 43, buildings on platforms were detected, which was not the case before. However, three cases of two bounding boxes for the same object were registered (a platform), as shown in Figure 43. Thus, by removing the NMS, only two more ground truth objects could be detected.

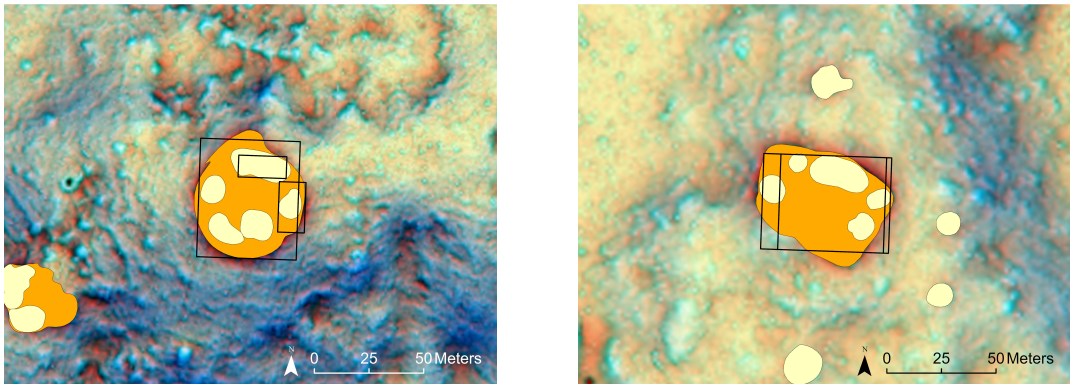


Figure 43: Prediction of buildings on a platform on the left and double detection on the right. Buildings are in yellow and platforms in orange.

7.3 Comparison of the two VAT

The results for the two different VAT are summarised in Table 13. A comparison of the three architectures shows that the ResNet-34 backbone model is best for both the one-band VAT and the three-band VAT. However, the results obtained with the three-band VAT indicate that this visualisation is better suited for the RetinaNet model. The accuracies are indeed higher by more than 10% for all backbones. The table shows that the best performing model is ResNet-34 with the three-band VAT.

Visualisation	Backbone architecture	Training loss	Validation loss	Accuracy
One-band VAT	ResNet-18	0.6702	0.6819	32.1%
	ResNet-34	0.5660	0.5626	32.9%
	ResNet-50	0.5121	0.6322	30.8%
Three-band VAT 8bit	ResNet-18	**0.4947**	0.5371	43.6%
	ResNet-34	0.5123	**0.5271**	**45.6%**
	ResNet-50	0.9372	0.8051	44.2%

Table 13: Comparison of the results for the one-band and three-band VAT. Lowest losses and highest accuracy are in bold character.

Comparison of the visualisations on the G-LiHT test data shows that the one-band VAT gave 52 true positives and 18 false positives, while the three-band VAT gave 57 true positives and 20 false positives. Although the latter model lead to two more false detections, five more ground truths were predicted. In general, we look for a model which

detects a maximum of ground truths and a minimum of false positives. However, in archaeology, we can consider that if a choice needs to be made, it is more interesting to have a model that detects a maximum of ground truths rather than a minimum of false predictions. Hence the testings would suggest that the three-band VAT is more appropriate. However, using the three-band VAT instead of the one-band did not drastically improve the performance of the models. The terrain difference, as well as data difference, thus remains a problem for the transferability of the model in object detection.

8 Object detection with the Chactún test set

8.1 Data description

As mentioned in the previous section, the trained deep learning model did not perform well on the G-LiHT dataset. To find out if this model transferability issue came from the terrain and data of G-LiHT, the northern part of Chactún was also used for testing. However, one cannot take the previously trained model and apply it to the north of Chactún. The reason for this is that the models were trained on the entire area of Chactún, including the northern part. The testing must be achieved on an area that the model has never seen. The training only has to be achieved on the south of Chactún before doing the testing on the north.

Therefore, the area of Chactún has been divided into two parts (training in the south and testing in the north) using ArcGIS Pro's "Clip" tool. The area selected for testing makes up about $1/5^{th}$ of the Chactún area. The division is shown in Figure 44. To avoid overlap between training and testing, a band of 323 meters between the two areas was not considered. The upper part was also omitted as no objects were labelled in this area.

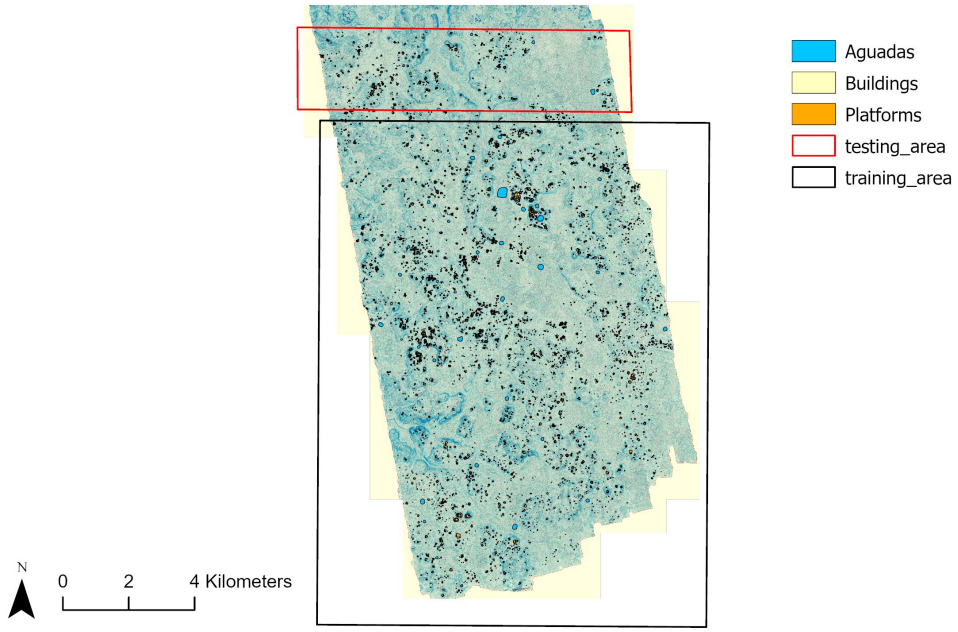


Figure 44: Separation of the Chactún area for training and testing. The base map is the three-band VAT.

The process of exporting the training tiles had to be carried out once again on the southern part. This part of Chactún counts 1,780 platforms, 7,443 buildings and 43 aguadas. For the export of the training data, both the VAT (one-band and three-band) and the ground truth objects within the southern region were used. The statistics for this new training data are summarised in the table below. One class of archaeological features was again considered, which includes buildings, platforms and aguadas.

Number of tiles	21,441	
Number of classes	1	
Number of features	53,782	
Number of features per tile	Min	1
	Max	27
	Mean	2.5084
Feature area	Min	1.3e-05
	Max	16,384
	Mean	390.6319
	Sum	21,008,966.29

Table 14: Statistical information from the Export Training Data tool.

8.2 One-band and three-band VAT models

For object detection, the best model for both the one-band and three-band VAT was the one with ResNet-34. Therefore, this backbone architecture was used to train the model with the new Chactún data. The results of the two models can be seen in Table 15. The accuracy is higher for the one-band VAT. However, the losses are lower for the three-band. The loss graphs and samples of the results can be found in the Appendix E (Table 79 and 80).

	Training loss	Validation loss	Accuracy
One-band ResNet-34	0.6755	0.6844	**51.4%**
Three-band ResNet-34	**0.5238**	**0.5473**	44.6%

Table 15: Results of the training of the models on Chactún South. Best values are in bold.

As each model has its own positive points (either accuracy or losses), both were tested on the northern area of Chactún. This area includes 204 platforms, 1,045 buildings and 7 aguadas. Hence a total of 1,256 archaeological features. The details of the number of objects for this area can be found in Table 16.

	Buildings	Platforms	Aguadas
Total number	1,045	204	7
Individual	409	26	-
On platforms	636	-	-
With buildings	-	178	-

Table 16: Amounts of testing archaeological features. Individual buildings and platforms are found alone and not in a building/platform pair.

Experiment 7 (see Figure 30) investigated the impact of the NMS parameter for the testing on Chactún North. The first object detection algorithm was then run with Non Maximum Suppression (NMS) and with the one-band VAT. The model detected 911 features. The false and true positives were manually examined. 634 true positives and 277 false positives were identified. This leaves about half of the objects that were missed by the model (false negatives). Compared to the results obtained with the G-LiHT test set, the number of true positives is much closer to the actual amount of ground truths. However, Non Maximum Suppression was used with this first object detection. The effect of this parameter on the data is to remove duplicate detections, but also the detection of buildings on platforms. As a consequence, when a platform was predicted the potential buildings on it were not, and inversely. The vast majority of false negatives were due to this trade-off. Another model was then trained without this parameter. The influence of the NMS can be seen in the sample of the results shown in Figure 45.

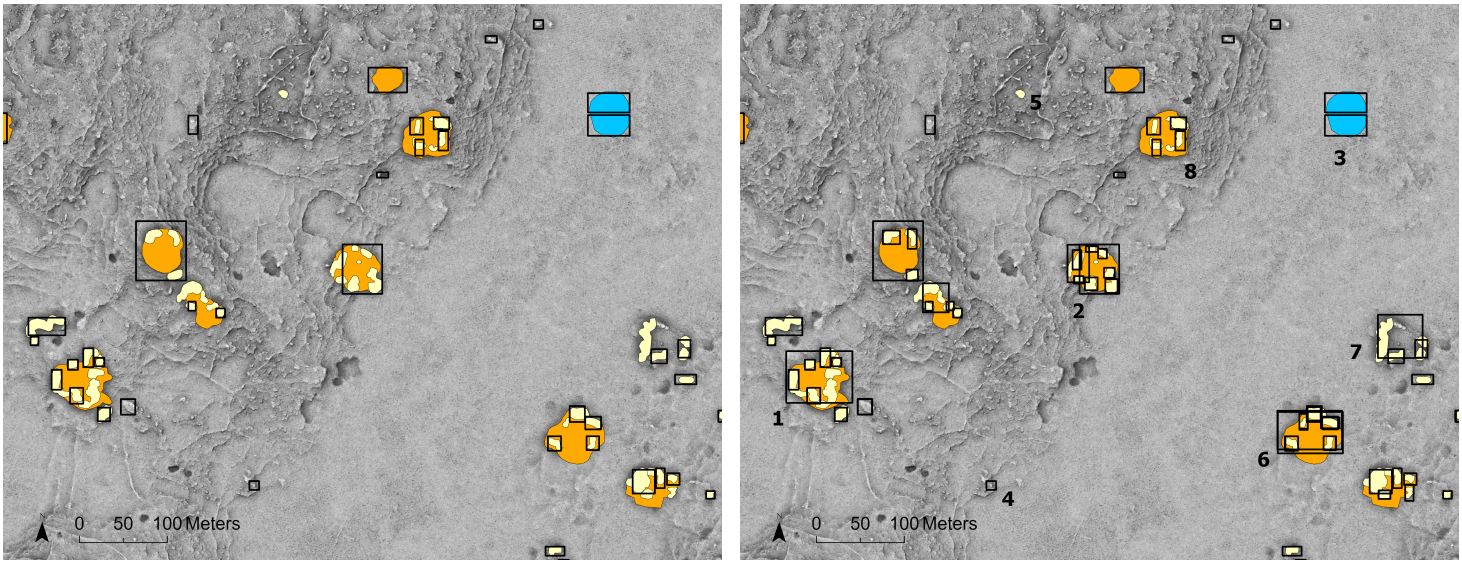


Figure 45: Sample of the results of the object detection with NMS active on the left and inactive on the right. Platforms are in orange, aguadas in blue, buildings in yellow and the predicted objects are the black rectangles.

If predictions are to be obtained for building/platform pairs, the NMS must be inactive (see for example number 1 in Figure 45). However, the predictions must then be further investigated to discard any double detections that may occur. The result of this object detection without NMS shows 1,373 predicted features. Among them are 933 true positives and 353 false positives. We can therefore assume that 87 bounding boxes were assigned twice to the same object (see numbers 2 and 6). In summary, by removing the NMS, 311 more true objects could be detected. In this case study, it is therefore more interesting not to use NMS. The high value of 933 out of 1,256 archaeological features was predicted by the one-band VAT model. It can also be noted that only 1 aguada, 2 small individual platforms and 24 individual buildings were not predicted (see number 5). No specific pattern was identified in these missing detections (false negatives). The remainder and the majority of the false negatives were from building/platform pairs, where the

boundaries of the structures are less easy to identify. The unpredicted aguada shows a lower contrast compared to the predicted aguadas, with a more shallow and flat shape. False positives arise mainly on rugged terrain or small, isolated hills on flat terrain. Basically, discontinuities in the ground are sometimes predicted as an archaeological feature (see number 4). Clusters of buildings were often predicted to have a platform (see number 7).

The three-band VAT ResNet-34 model was also applied to the test area (the NMS was not activated). This is Experiment 8 (see Figure 40) which investigates the transferability of the model on Chactún North. The sample of the results of the three-band model can be found in Figure 46. One can already see that the aguada was overlooked and fewer buildings were predicted than with the previous model. In general, a lower number of objects were predicted, with a value of 948. 683 true positives and 125 false positives were registered. There are therefore 140 double detections. It can be noted that 40 individual buildings were missed, as well as 1 individual platform and 3 aguadas. Note that individual objects refer to buildings and platforms which are not found in a building/platform pair. A comparison between the one-band and the three-band model is depicted in Table 17. The table shows that for archaeological detections, the one-band model (without NMS) is better. Indeed, for archaeological purposes, one looks for a model with the highest number of true positives and the lowest number of false negatives. In summary, the best model for object detection on a new dataset uses the one-band VAT and the backbone ResNet-34 without NMS. This also shows that higher accuracy is more important for a model (in the sense that it leads to better results) than lower loss, as shown in Table 15.

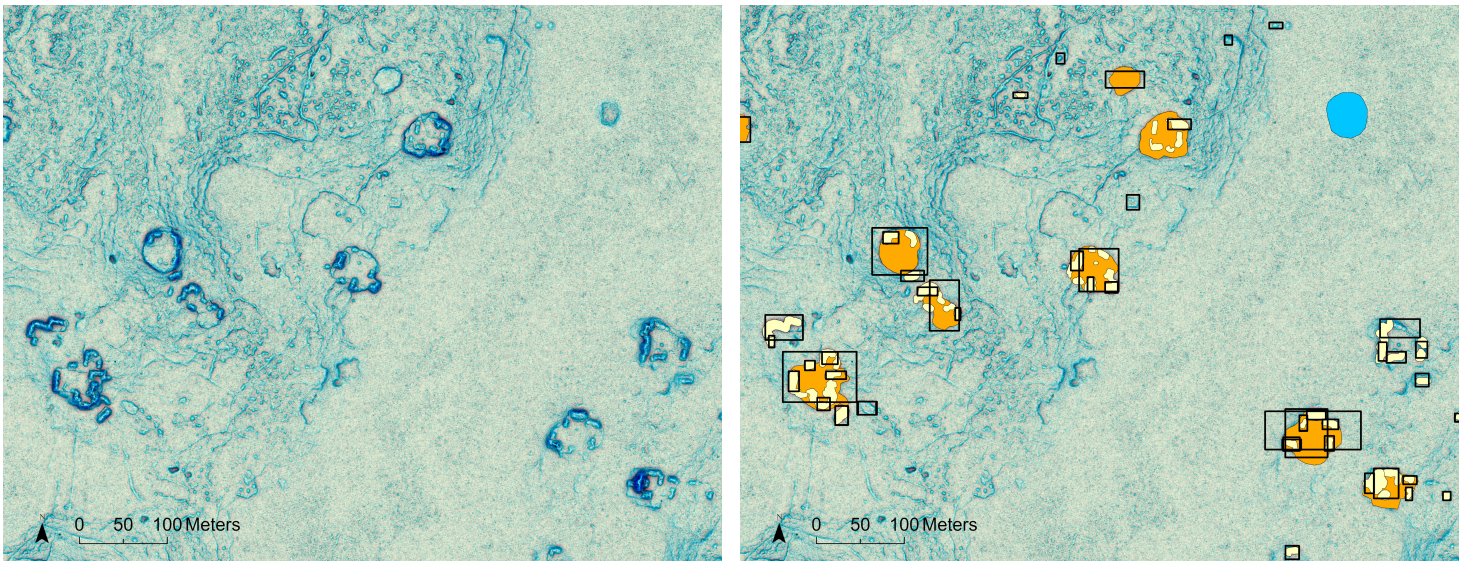


Figure 46: Input VAT on the left and sample of the result of the object detection on the right. Buildings are in yellow, platforms in orange, aguadas in blue and the predictions are the black rectangles.

Deep learning model	One-band ResNet-34 NMS	One-band ResNet-34 No NMS	Three-band ResNet-34 No NMS
Total number of predictions	911	1,373	948
True positives	634	933	683
False positives	277	353	125
False negatives	622	323	573

Table 17: Results of the object detection for the one-band VAT and three-band VAT models.

8.3 Separation of the three classes

Because of the overlap between buildings and platforms, it was not possible to consider the three classes separately within a single model. Another possibility exists to take into account the different types of archaeological features. It consists of training three different models, one for each class. Experiment 9, for which the workflow is visible in Figure 47, investigates this separation of the classes.

Experiment 9

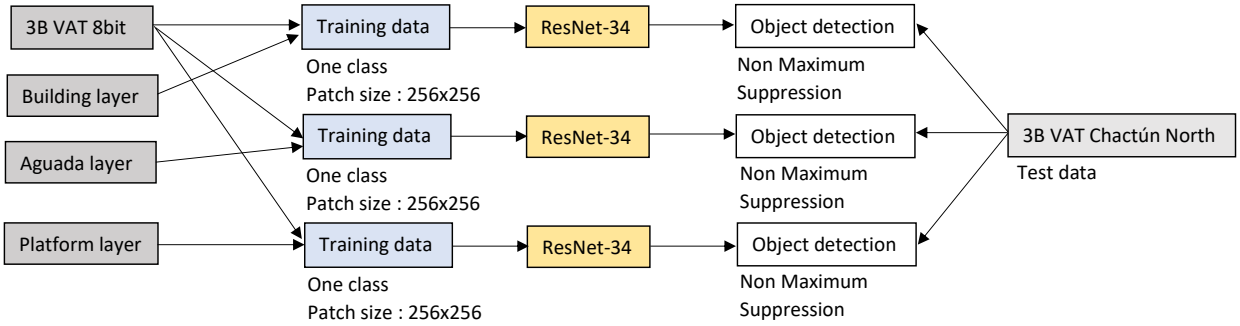


Figure 47: Workflow of experiment 9.

To see if this improves the results, the three-band VAT was used for this class separation. The patches were exported with the PASCAL metadata type for each class. The RetinaNet model was then trained on each class separately, using the ResNet-34 architecture. The results of the training are summarised in Table 18. The losses are higher for aguadas because fewer training examples are available. The model then overfits the examples and cannot generalise well. The predictions are therefore further away from reality in the case of aguadas. However, for the three features, the accuracy is much higher than if a single class of combined archaeological objects was considered.

Feature class	Training loss	Validation loss	Accuracy
Buildings	0.5393	0.6163	72.8%
Platforms	0.4867	0.4540	67.6%
Aguadas	1.7627	1.7621	62.2%

Table 18: Results of the training of the RetinaNet models with the three-band VAT and ResNet-34 architecture for the three classes.

The three resulting models were then tested on the northern part of Chactún. Non Maximum Suppression was enabled as there were now no objects overlapping within one model. A sample of the results can be found in Figure 48.

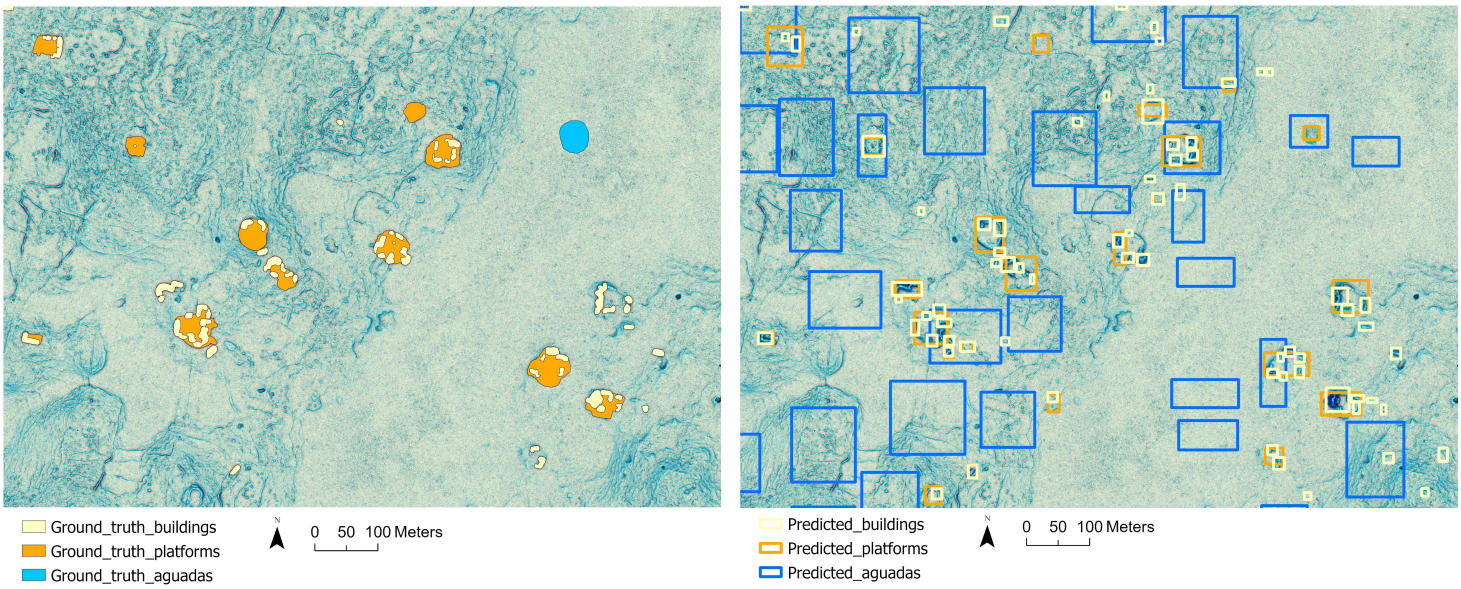


Figure 48: Sample of the results of the object detection for the RetinaNet models with the three-band VAT. Predictions for each class were achieved with a different model.

The results of the visual inspection are summarised in Table 19. For the aguadas, the model detected a very high number of false positives. These false positives occurred at various locations in the area and did not show a clear pattern in the detections. It can be concluded that the number of examples used for training was too small to obtain a performant model. Indeed, the usual amount required to train a deep learning model is of the order of at least several thousand. In contrast, only 43 aguadas were used for training here. One way to increase the amount of training examples is to use a data augmentation technique, such as sample rotation. Note that then only the three-band VAT could be used, as it is independent of the direction. As for the buildings, most of them (74%) were predicted. Those that were not detected are part of a cluster and hence the boundaries of the individual buildings are harder to identify. Many false positives were also predicted (though much fewer than for the aguadas), including 14 predictions that were actually platforms. Other false positives include small, elevated hills. Finally, 90% of the platforms were predicted. However, some natural flattened surfaces were also misclassified as such, as were some buildings. All clusters of buildings were predicted to have a platform, even if none was recorded. This confirms that the model learned well from the training data. It is indeed very plausible to find a platform under a building cluster. Ultimately, more false positive than true positive predictions were made. In the Table 19 one can find the number of missed ground truths (false negatives), which is quite small compared to the total number. This is mainly what one looks for in the case of archaeology (to minimise the number of false negatives). A maximum of known structures should be detected. From the results, it can be concluded that the models predict the archaeological features quite well, but add a lot of false positives. Increasing the minimum

confidence value allowed could be a way to reduce the number of false positives, however the confidence value varies a lot among those false predictions, going from 50% to 88%. For object detection, considering one model for each class then leads to a high amount of false positive predictions.

Feature class	Buildings	Platforms	Aguadas
Number of ground truths	1,045	204	7
Number of predictions	1,227	480	498
True positives	774	184	7
False positives	453	296	491
False negatives	271	20	0

Table 19: Results of the object detection with the RetinaNet models for the three classes.

9 Semantic segmentation with the G-LiHT test set

The training data available for the Chactún area enable the semantic segmentation method.

9.1 One-band VAT

The first visualisation considered was the one-band VAT. Again, the export (using the Export Training Data tool) only worked for buildings, platforms and aguadas considered as a single class, as objects from different classes overlapped in the image. Therefore, we initially assumed that the aguadas, buildings and platforms belonged to one class, namely the archaeological features. For the export, the option "classified tiles" has to be selected. For the training of the model, we used the U-Net model which belongs to the "pixel classification" group. It is most commonly used for semantic segmentation. Two backbones were investigated during Experiment 10, for which the workflow can be seen in Figure 49. This experiment also investigates the impact of the focal loss.

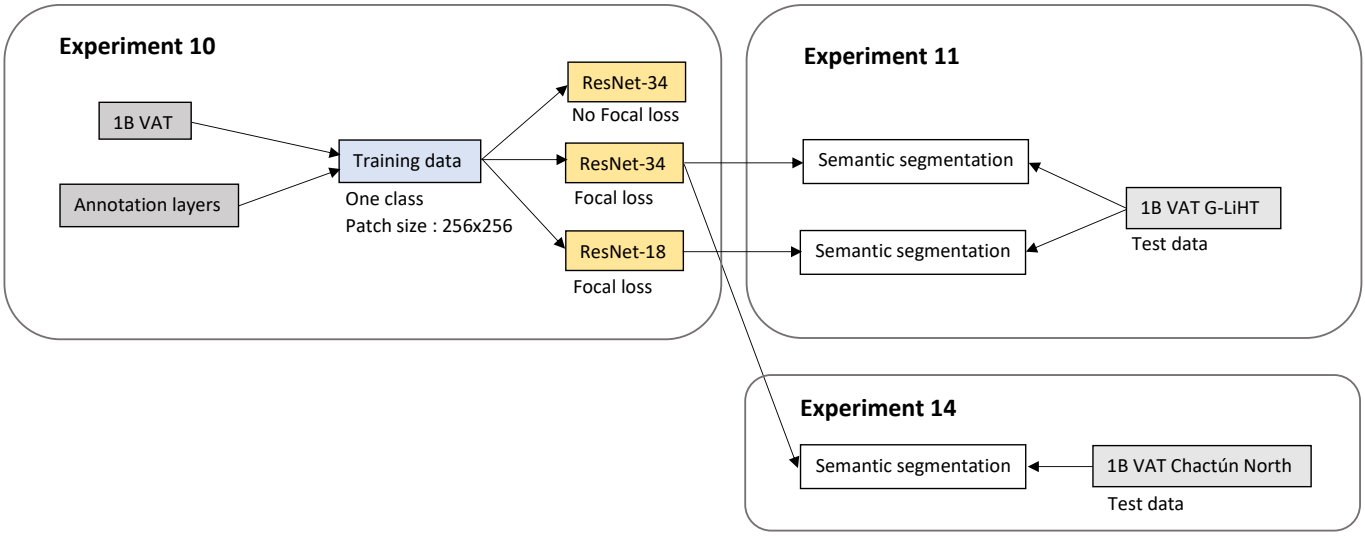
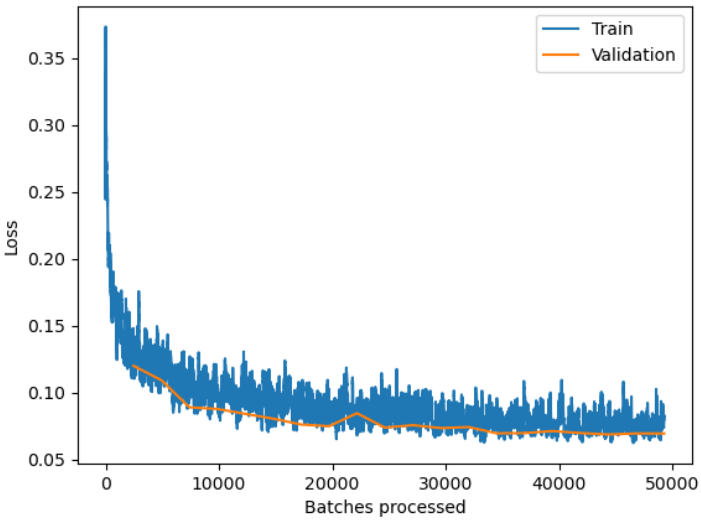


Figure 49: Workflow of experiments 10, 11 and 14.

The model was first trained with the ResNet-34 backbone and without using the focal loss option. In general, for semantic segmentation, the batch size had to be reduced compared to object detection. For ResNet-34, a batch size of 8 was used. Note that the processing time for segmentation is much longer, taking about 12 hours. For comparison, the processing time for object detection was 6 hours on average. The results of the model are shown in Figure 50.



	Min	Max
Learning rate	4.36e-06	4.36e-05

	Background	Class
Precision	98.2%	74.5%
Recall	98.5%	71.8%
F1 score	98.4%	73.2%

Figure 50: Training and validation losses for the U-Net model with ResNet-34 on the left and model metrics on the right.

The large fluctuations in the training loss are due to the small batch size. Despite these fluctuations, however, the loss is much lower than with object detection using bounding boxes. The table on the right of Figure 50 contains the performance metrics of the model. One can see that the values of the performance metrics for the background are very high. The background was correctly classified as such in 98% of the cases. The values for the archaeological features (referred to as "class" in the table) are more meaningful and are of the order of 70%, which is slightly lower but still very good, especially when compared to the accuracies achieved for object detection, which were of the order of 40%.

Samples of the results can be found in Figure 51. It clearly shows the difference between object detection (bounding boxes) and semantic segmentation (pixel classification). The predictions follow quite well the shape defined by the ground truths. Only the platform in the third sample was not predicted. From the results it can be concluded that the way the training dataset was prepared - with polygons following the contours of the objects - is well suited for semantic segmentation. Object detection works less well with this dataset.

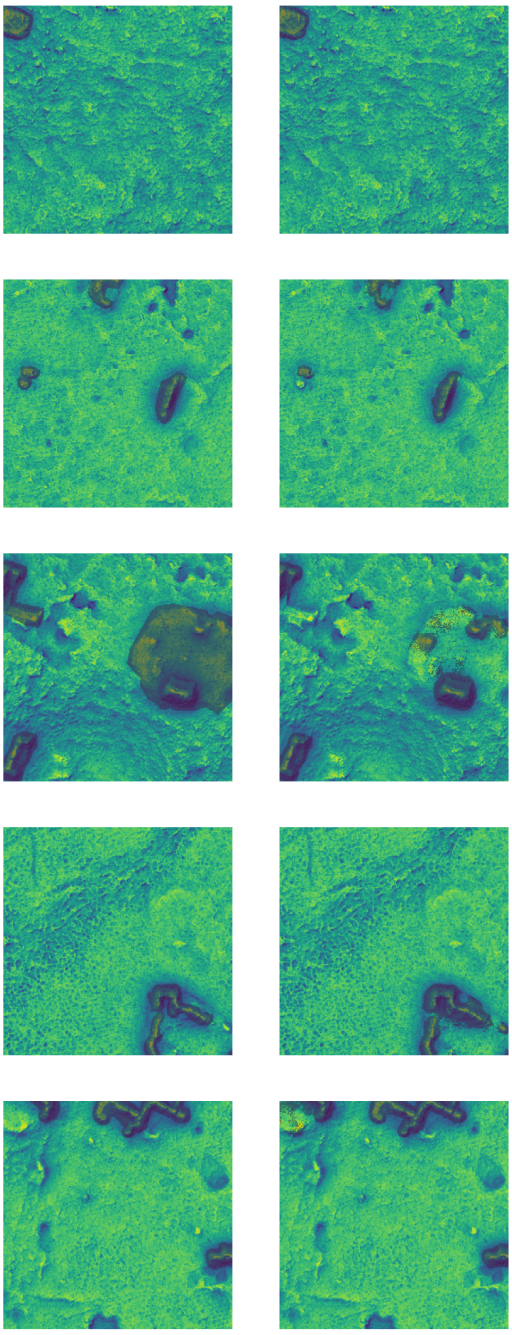


Figure 51: Result samples for the U-Net model trained with ResNet-34. Ground truths on the left and predictions on the right.

Another model was trained in the Experiment 10 by setting the arguments class_balancing and focal_loss to True. Previously, they had in fact been left at the default value of False. In this way it was possible to check whether or not the dataset contained a class imbalance problem. Here the negative data (background) is the majority group. The idea was to see if this class had a strong impact on the accuracy of the model. The accuracy of the two models was very similar: 96.9% when the class imbalance was not corrected versus

96.5% when it was corrected. This indicates that the dataset does not depict a strong class imbalance. To remain cautious, the following models were nevertheless trained with the class balancing and focal loss parameters. These parameters were not available in the object detection model, because the RetinaNet model inherently corrects the class imbalance problem through the focal loss function.

The ResNet-50 backbone was not implemented because the increased number of layers and thus complexity, required a reduction of the batch size to a value of 2. However, such a small value would lead to an erratic and long training, with a loss function that might not reach the minimum [48]. Consequently, the model would be less performant. As a result, the ResNet-18 backbone architecture was trained. The results are shown below in Figure 52. Note that the loss fluctuations are less pronounced as the batch size could be increased to a value of 20. The training is faster than for ResNet-34, which is evident from the rapid decrease of the training loss. The higher the number of layers in the model, the longer the training takes. As can be seen in Table 20, the losses are higher than for ResNet-34, suggesting that the latter is a better fit. However, one could argue that the performance metrics for ResNet-18 are slightly higher. To verify this, both models were tested on the G-LiHT dataset in Experiment 11 (see Figure 49).

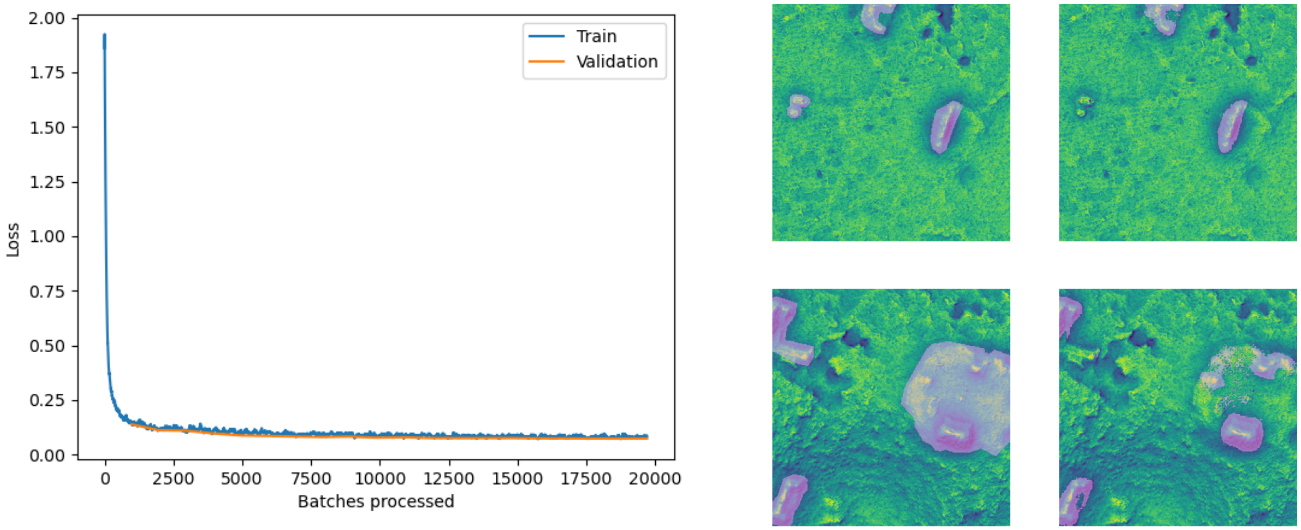


Figure 52: Training and validation losses for the ResNet-18 backbone model on the left. Sample results of the model on the right, with ground truths on the left and predictions on the right.

Backbone	Training loss	Validation loss	Accuracy	Precision	Recall	F1	Dice
ResNet-18	0.0317	0.0726	**96.8%**	**73.9%**	**70.7%**	**72.3%**	61.7%
ResNet-34	**2.530e-05**	**0.0006**	96.5%	71.6%	69.4%	70.5%	**63.5%**

Table 20: Comparison of the results for the one-band VAT. The best values are in bold.

In Experiment 11 the two models for the one-band VAT were tested on G-LiHT using the Classify Pixels tool. A sample of the results for ResNet-34 is provided in Figure 53.

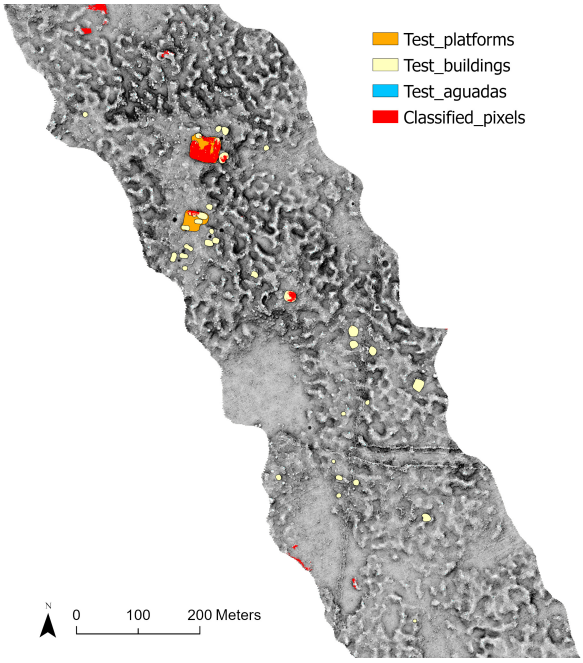


Figure 53: Results of the classify pixels tool with the U-Net model ResNet-34 for the one-band VAT.

As mentioned previously, the IOU metric is often used to evaluate the model performance. The pixel quantities determined through the IOU computation (see Figure 16) for ResNet-18 and ResNet-34 are provided in the Appendix F (Table 31). Equation (8) gives a IOU of 14.2% for ResNet-18 and 17.7% for ResNet-34. Thus the latter is slightly better and will be examined in more detail. Note, however, that such a low IOU is not satisfactory and is most likely due to the difference in terrain and data quality between the training and testing sets. Due to the rugged terrain, many false positives occur, as can be seen in Figure 54. Some of these false positives could be new buildings that were not spotted when the ground truths were defined. An example of this can also be seen in the figure. However, the model failed to detect most of the known buildings. The visual inspection actually revealed that 167 buildings were correctly predicted and 628 were missed. This gives a total of 21% of predicted buildings. Note that an object is considered missed if less than 30% of its pixels are classified correctly. Of the three aguadas, only one was correctly predicted. Platforms are a special case. If a building is found on a platform, the pixels of the building can be predicted. However, the rest of the platform is not detected. This is illustrated in Figure 55, where the centre of the platforms is never classified. One can conclude that the overlap between buildings and platforms prevents the detection of both objects when only one class is considered.

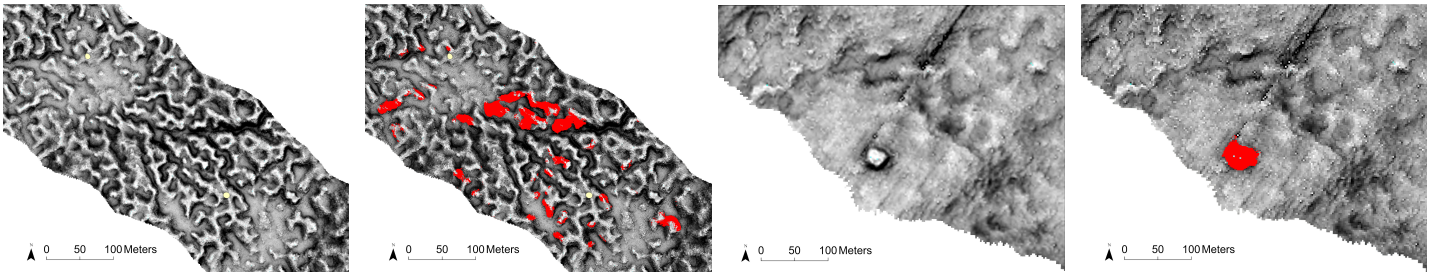
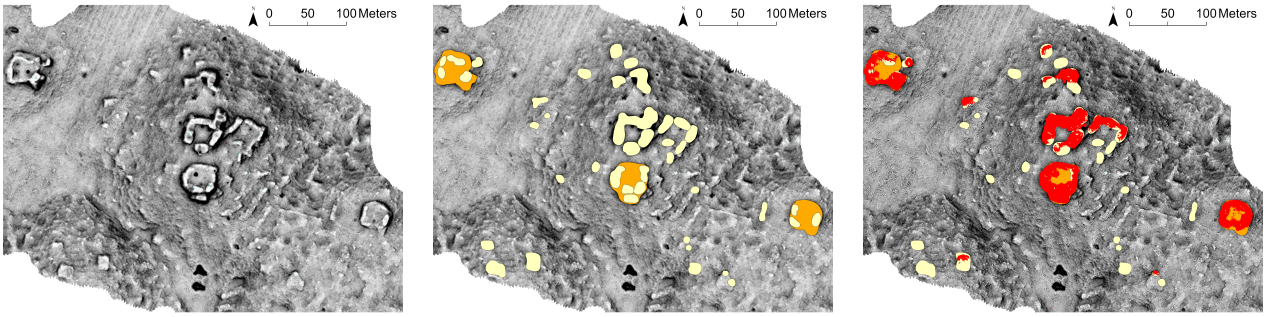


Figure 54: False positives on rugged terrain on the left images and possible new building on the right images.



(a) Input VAT (b) Ground truths (c) Predictions

Figure 55: Detection of buildings on platforms with (a) the input VAT, (b) ground truths and (c) classified pixels. Buildings are in yellow, platforms in orange and classified pixels in red.

The dice coefficient can also be computed, although it is similar to the IOU. Equation (9) gives a value of 30.1%.

9.2 Three-band VAT

The training data was also exported as classified tiles using the three-band raster visualisation. This is Experiment 12 (see Figure 56) during which two backbones were investigated.

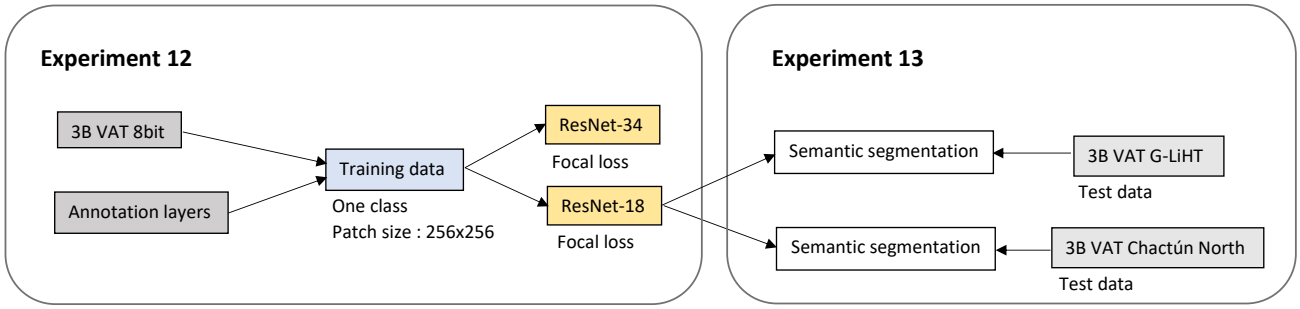


Figure 56: Workflow of experiments 12 and 13.

For the training of the model, the batch size was set to 15, as the algorithm ran out of GPU memory at a higher value. The ResNet-18 and ResNet-34 backbones were trained and provided accuracies of 97.0% and 95.7% respectively. The ResNet-18 backbone is therefore better suited for archaeological detections. The training and validation losses of this model, shown in Figure 57, are very close to each other, indicating that overfitting is not occurring. The sample of the results on the right side of the figure shows that the pixels in the validation set were overall well classified.

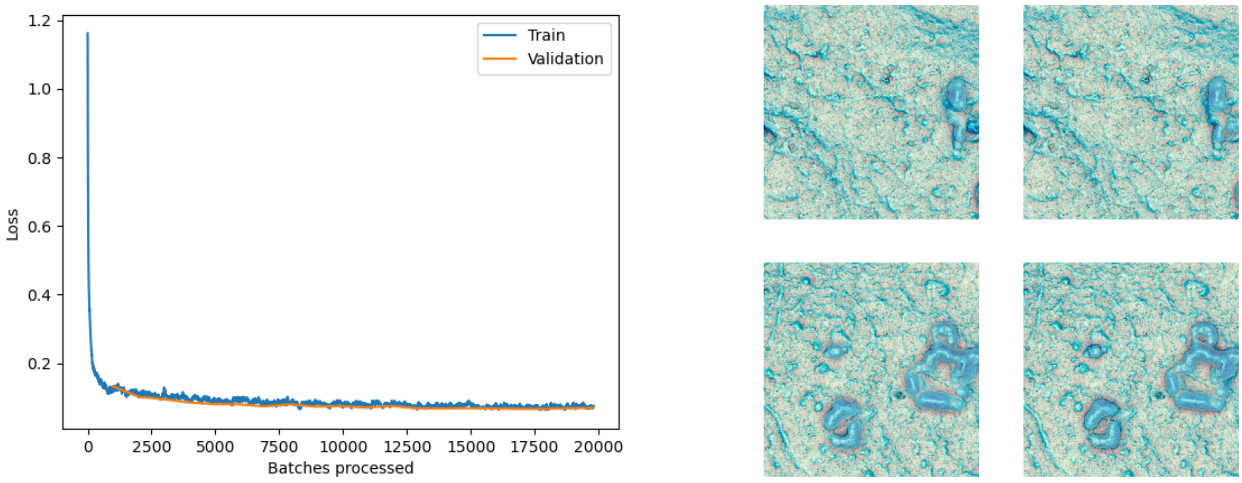


Figure 57: Losses for the ResNet-18 backbone model on the left. Sample results of the model on the right, with ground truths on the left and predictions on the right.

This ResNet-18 model has been tested on G-LiHT in Experiment 13 (see Figure 56). A sample of the results can be found in Figure 58. The IOU computation (see processing chain Figure 16) shows a value of 2,775 pixels corresponding to false positives, 12,003 true positives and 83,467 false negatives. From the IOU formula, the value for the IOU is 12.2%, for the archaeological pixel class. The dice coefficient itself has a value of 24.9%. This result is again not satisfactory for the testing set. Note that the 3 aguadas were missed. Only 50 archaeological features were well classified as such. These included 6 platforms and 44 buildings (out of 60 and 795 respectively). False positives again result from the fact that the test area has a very rugged and variable terrain. The tops of hills or mounds were sometimes mistaken for buildings and more extensive valleys for platforms.

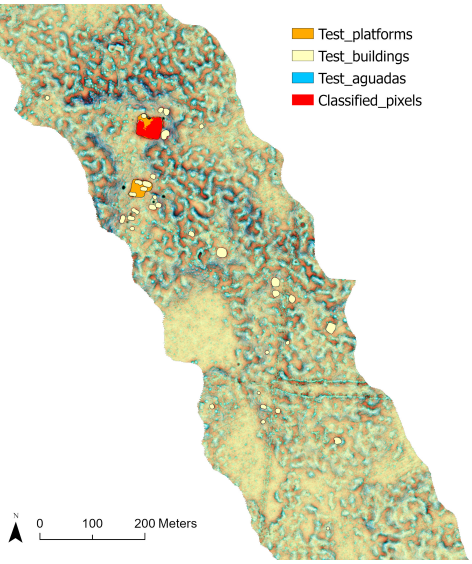


Figure 58: Classified pixels with the U-Net model ResNet-18 for the three-band VAT.

9.3 Comparison of the two VAT

For a comparison of the different models, see Table 21. The lowest losses and the highest values of the metrics are in bold. From these, one can conclude that the best model, with the highest performance, is the ResNet-18 model obtained from the three-band VAT. The loss values are lower for the ResNet-34 and one-band VAT model. Looking at the results of the testing on the G-LiHT data, the IOU for the one-band model is 17.7%, while it drops to 12.2% for the three-band model. Thus both models have positive and negative aspects. However, from the low IOUs, one can conclude that the models do not perform very well on this testing raster, regardless of the visualisation for archaeological topography.

Visualisation	Backbone	Training loss	Validation loss	Accuracy	Precision	Recall	F1	Dice
One-band VAT	ResNet-18	0.0317	0.0726	96.8%	73.9%	**70.7%**	72.3%	61.7%
	ResNet-34	**2.530e-05**	**0.0006**	96.5%	71.6%	69.4%	70.5%	63.5%
Three-band VAT	ResNet-18	0.0775	0.0695	**97.0%**	**78.2%**	**70.7%**	**74.2%**	**65.1%**
	ResNet-34	0.0042	0.0022	95.7%	69.4%	53.2%	60.2%	47.6%

Table 21: Comparison of one-band VAT and three-band VAT U-Net models. Best values are in bold.

10 Semantic segmentation with the Chactún test set

10.1 One-band and three-band VAT models

Semantic segmentation was also applied on the northern Chactún data for testing. The training samples from the southern region were exported as classified tiles for the one-band and three-band VAT. The best model previously identified for the entire Chactún region was ResNet-34 for the one-band VAT and ResNet-18 for the three-band VAT. These two

backbones were then used to train the new models. The training and testing of the three-band VAT model compose Experiment 13 (see Figure 56) which investigates the difference between G-LiHT and Chactún North. While the training and testing of the one-band VAT model constitute Experiment 14 (see Figure 49), which again compares the two test sets. The results of the training are summarised in Table 22. Loss graphs and samples of the results can be found in Appendix E (Figure 81 and 82). The performance metrics, as well as the losses, are very close for the two models. However, the one-band VAT model generally shows more performant values. For example, the dice value is higher for the one-band VAT model. This suggests that a greater proportion of the total ground truth and prediction areas overlap. Since in archaeology it can be considered more important to detect a maximum number of ground truths (minimising false negatives) than to minimise the number of false positives, the recall metric can be considered the most important. A high recall value is therefore preferable. Note however that the number of false positives should still not reach a too high value which would make the results unpractical. The recall metric value in Table 22 also indicates that the best model is the one-band VAT ResNet-34.

Deep learning model	Training loss	Validation loss	Accuracy	Precision	Recall	F1	Dice
One-band VAT ResNet-34	**0.0012**	**0.0007**	**96.6%**	71.4%	**69.5%**	**70.4%**	**62.0%**
Three-band VAT ResNet-18	0.0014	**0.0007**	96.5%	**72.4%**	66.5%	69.3%	59.4%

Table 22: Results of the training of the one-band VAT and three-band VAT models for semantic segmentation. Best values are in bold.

The models were tested on the northern region of Chactún. A sample of the results can be found in Figure 59. Visual inspection already shows that the three-band VAT model seems to have predicted fewer ground truths than the one-band VAT model.

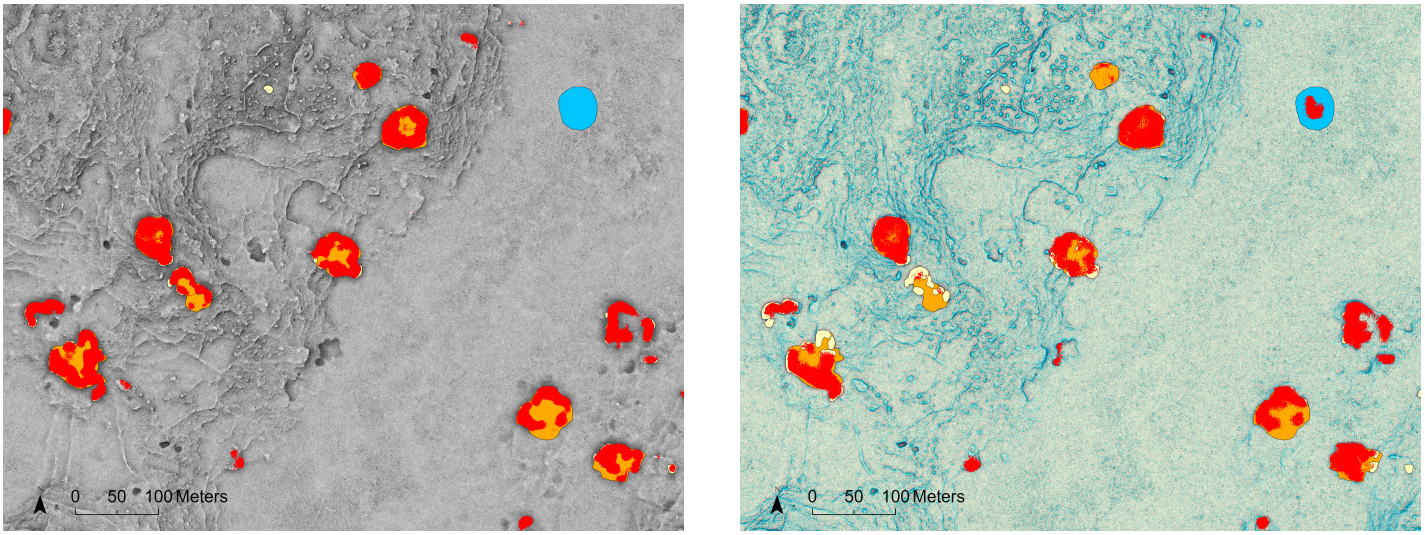


Figure 59: Sample of the results of the semantic segmentation for one-band VAT on the left and three-band VAT on the right. Platforms are in orange, aguadas in blue, buildings in yellow and classified pixels in red.

To evaluate the performance of the models, we count the number of true and false positives pixels by pixels, as each pixel has been classified as either a background or an archaeological object. We did this using the method described in Figure 16. For the one-band VAT model, the IOU was calculated using the following values:

$$IOU = \frac{TP}{TP + FP + FN} = \frac{1,038,073}{1,038,073 + 127,357 + 563,390} = 60.0\%, \tag{10}$$

IOU shows that the model performs better on data with the same properties compared to the previous G-LiHT test data. During visual inspection, the first thing that stands out is that one aguada was completely classified as background by the model (see Figure 59). In contrast to the object detection, this aguada is deeper than the other ones.

A clear advantage of the one-band VAT model trained with ResNet-34 is that the pixels corresponding to the buildings on platforms were all well classified. Only four insignificant exceptions were spotted for very small buildings. However, these perfect classifications for buildings on platforms came at the expense of detecting the platforms underneath. In fact, all platforms supporting buildings were classified as background, except for small platforms that are almost completely hidden by buildings. This aspect can be seen in Figure 59. 55 individual buildings on bare ground were missed (with classified pixels on less than 30% of the building). Four individual platforms were also not predicted. Two of these failures stemmed from a building being misidentified on them. Finally, background pixels were sometimes misclassified as archaeological features. This occurs on hills and bulges of a few metres high that are interpreted as buildings. However, these areas could actually be areas of interest as they potentially represent true buildings. In summary, for 1,018 objects, more than 30% of the pixels were correctly classified as archaeological features. With a total number of 1,256 features, this figure is very satisfactory. By comparison, the best object detection model delivered 933 true positives.

The IOU for the three-band VAT model trained with ResNet-18 is:

$$IOU = \frac{846,789}{846,789 + 128,707 + 754,674} = 48.9\%, \tag{11}$$

The IOU shows that the performance is less good than for the one-band VAT model. Again, the buildings that were on a platform were detected well, while the platform was missed. However, more pixels were classified towards the centre of the platform when it is surrounded by buildings. This can be seen in Figure 59. Unlike the previous model, not all buildings on the platforms were well classified. Visual inspection revealed 40 overlooked buildings. 197 individual buildings and 14 platforms were also missed. All aguadas contained well classified pixels, but only a small amount (less than 10%). This is especially true for the shallowest aguada, where the raised edges are less pronounced. Finally, 17 building/platform ensembles were misclassified as background, which never occurred for the one-band model. These ensembles consisted of a small platform with a small building. To sum up, the best semantic segmentation model uses the one-band VAT and the ResNet-34 backbone.

10.2 Separation of the three classes

Experiment 15 investigates the separation of the three classes. The workflow is visible in Figure 60. The best model for semantic segmentation just defined (one-band VAT with ResNet-34) was used in the case of three separate classes, as was done earlier for object detection. Three models are then trained, one for each class.

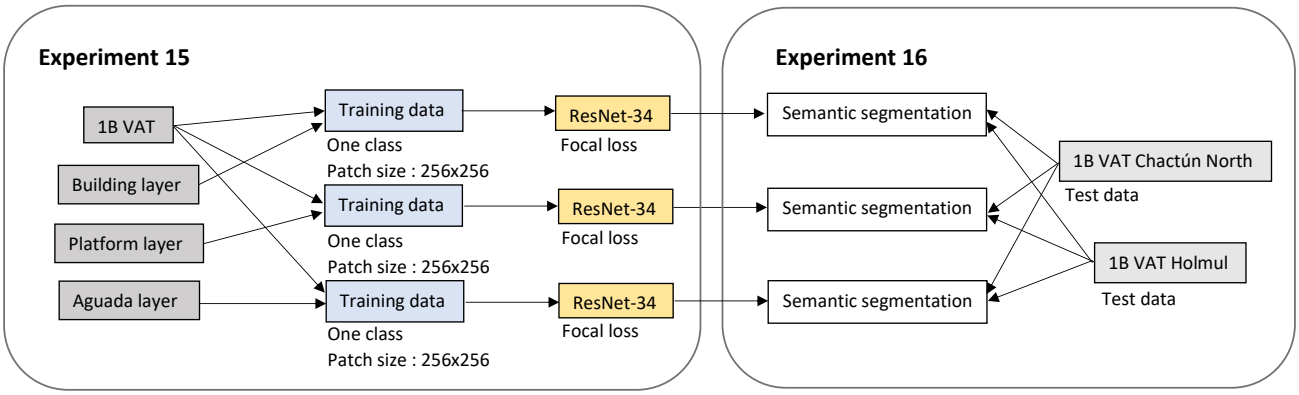


Figure 60: Workflow of experiments 15 and 16.

The results of the training for the three models can be found in Table 23. The results are particularly good for buildings and platforms, with a high accuracy, recall and dice coefficient. The dice value is lower for aguadas, again due to the small number of training examples. The recall values are higher when considering each class separately. Indeed, the previous model gave a value of 69.5%.

Feature class	Training loss	Validation loss	Accuracy	Recall	Dice
Buildings	4.1054e-06	8.1507e-05	98.3%	77.7%	65.1%
Platforms	0.0005	0.0009	96.7%	77.9%	67.3%
Aguadas	0.0020	0.0104	91.6%	75.5%	42.1%

Table 23: Results of the training of the U-Net models with the one-band VAT and ResNet-34 architecture for the three classes.

The Classify Pixels tool was run to test the models on the northern part of Chactún (see Experiment 16 in Figure 60). The results are shown in Figure 61. Concerning the model to detect aguadas, as with object detection, many false positives were predicted. A sample of these false positives can be seen in Figure 62, where no ground truth aguada is present. This is again due to the small number of training examples available. However, visual inspection shows that the results are still better for semantic segmentation, with fewer false positives compared to object detection. This could be due to the architecture of the U-Net model, which is better adapted to cases where only a small number of training examples are available [27]. Of the seven aguadas that were present in the testing area, only one was missed. This is again the deepest aguada, which was also missed when using the one-band VAT. False positives occur mainly on small flat areas that show a lower elevation compared to the surrounding area. However, pixels in rough terrain were also

frequently classified as aguadas. So the model does not detect any new possible aguadas, but mainly false positives.

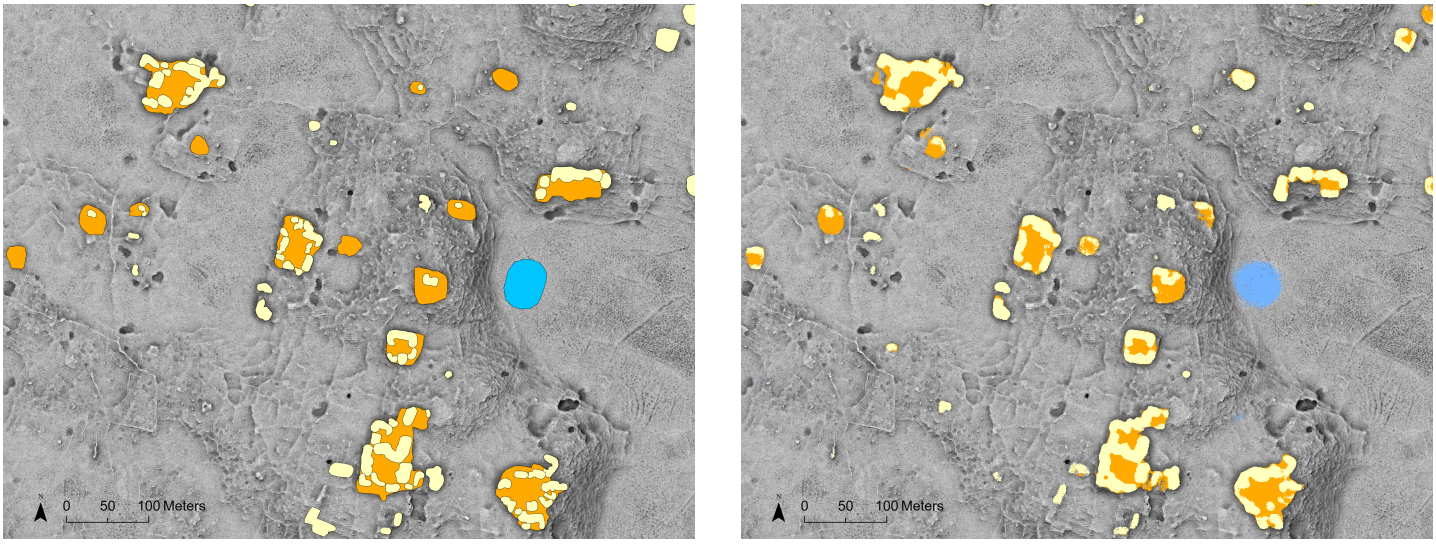


Figure 61: Ground truths on the left and results of the semantic segmentation on the right. Buildings are in yellow, platforms in orange and aguadas in blue. Each feature segmentation layer was obtained from a separate model.

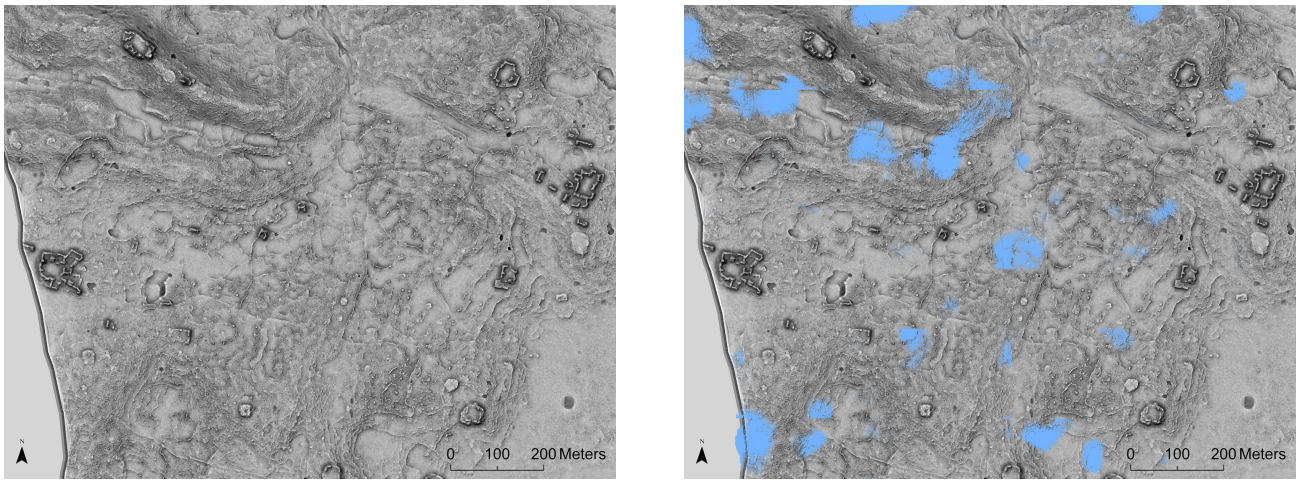


Figure 62: Sample of the false predictions of the aguada model with the input VAT on the left and the classified pixels on the right. No true aguada exists in the area chosen.

The IOU for the aguada model is again calculated using the number of TP, FP and FN pixels.

$$IOU = \frac{148,647}{148,647 + 738,844 + 15,850} = 16.4\%, \tag{12}$$

One might wonder how this rather poor result comes about, considering that the accuracy for the aguada model was 91.6%. This comes from the definition of accuracy, which also takes into account the true negatives, i.e. the background. More representative of the performance of the model is then the IOU coefficient, which was calculated here for the class objects only, without taking the true negatives into account. The dice coefficient of the model, provided in Table 23, is also more representative. However, the value provided by ArcGIS Pro is an average value between the dice coefficient of the feature class and the background class. This explains the higher value compared to the dice calculated for the aguada class, which is 28.3%.

As for the building class model, the results were very good. This is confirmed by the IOU:

$$IOU = \frac{760,923}{760,923 + 235,804 + 167,889} = 65.3\%, \tag{13}$$

The few missed buildings are small remains with little height discontinuity with respect to the surrounding area. The few false positives are small bulge formations that could be interpreted as buildings. Buildings standing on platforms were all correctly classified. They are the most visible and show a large discontinuity in the ground.

Finally, the model for platform detection turns out to also work well. All platforms are correctly predicted. Some false positives arise mainly for clusters of buildings or naturally flat terrain with a slightly higher elevation. The two shallowest aguadas were also misclassified as platforms.

$$IOU = \frac{1,272,823}{1,272,823 + 516,163 + 190,060} = 64.3\%, \tag{14}$$

Since the performance metrics for the three-band VAT were of the same order of magnitude as for the one-band (see Table 22), this visualisation was also trained by separating the three classes. This is Experiment 17 which can be seen in Figure 63.

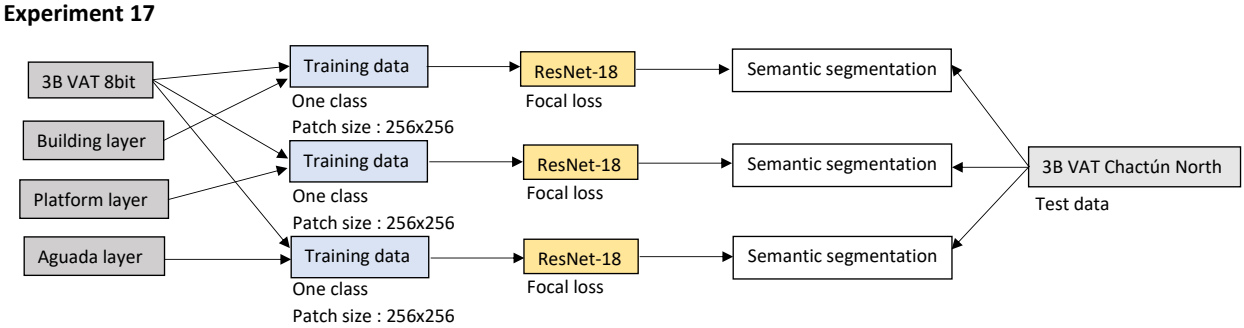


Figure 63: Workflow of experiment 17.

The tiles were exported again for this visualisation using the Export Training Data tool, with one training set for each class. Then the U-Net model was trained on these tiles using ResNet-18, resulting in one model per class. The performance metrics are provided in the table below. The metrics are slightly lower than for the one-band VAT.

Feature class	Training loss	Validation loss	Accuracy	Recall	Dice
Buildings	2.3211e-05	0.0001	98.2%	71.5%	62.4%
Platforms	0.0002	0.0005	97.1%	72.5%	66.3%
Aguadas	0.0054	0.0101	91.6%	66.8%	38.1%

Table 24: Results of the training of the U-Net model with the ResNet-18 architecture for the three classes with the three-band VAT.

The three models were then tested on the north of Chactún using the Classify Pixels tool. A sample of the results, which is the same sample area as previously, can be seen in Figure 64.

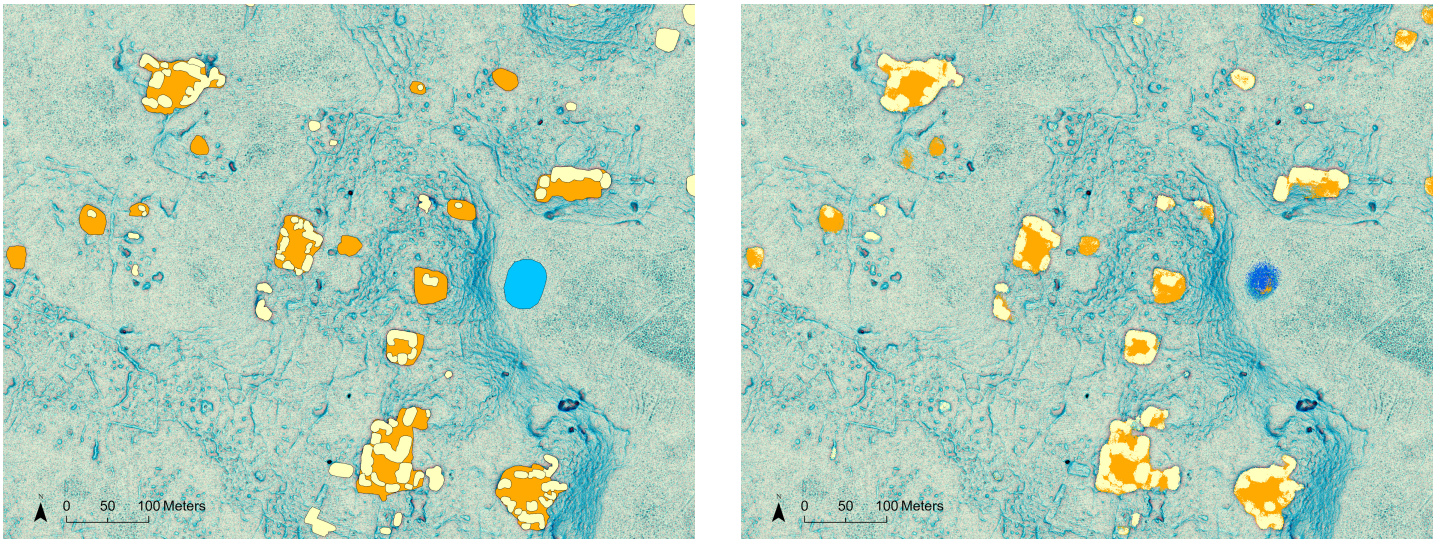


Figure 64: Ground truths on the left and sample of the results of the semantic segmentation on the right. Aguadas are in blue, buildings in yellow and platforms in orange.

The model for the aguadas gave better results than the one using the one-band VAT. Indeed, the IOU has a value of 28.3% (see Appendix F Table 32 for the values used in the calculation). Visual inspection showed that fewer false positives occurred, and those that did occur could actually be unidentified aguadas, except on rough terrain. As for the platforms, the vast majority were well predicted, with a IOU of 64.7%. As with the one-band VAT, a platform was always predicted for the clusters of buildings, even if it was not included in the ground truth dataset. Finally, the IOU value for buildings is 63.7%. These buildings were always well classified if they were on a platform. Overall, the results are quite similar to the results of the one-band VAT. However, the IOU values will determine the best model. A comparison table is provided below.

	Buildings	Platforms	Aguadas
One-band VAT ResNet-34	65.3%	64.3%	16.4%
Three-band VAT ResNet-18	63.7%	64.7%	28.3%

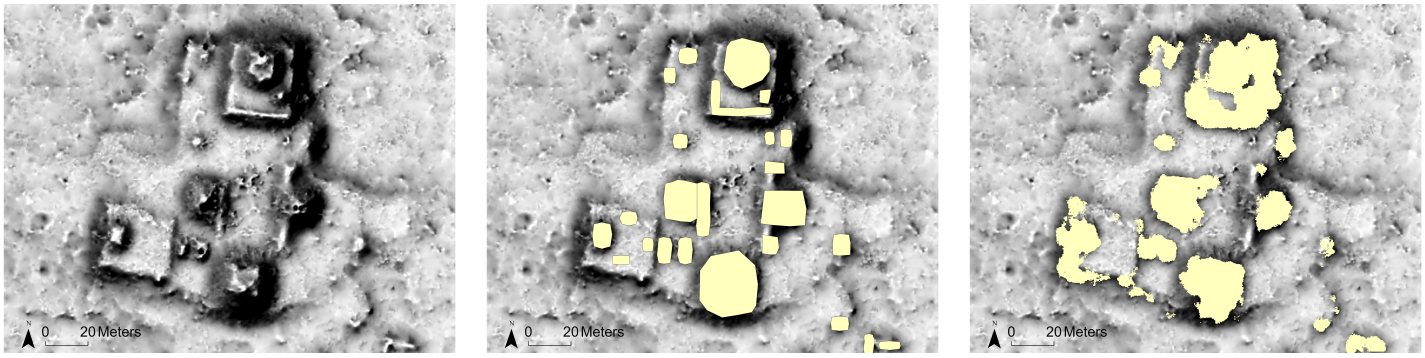
Table 25: IOU values of the U-Net model for the one-band and three-band VAT.

For the platforms, the values are similar for the two models. The building predictions are slightly better for the one-band VAT model. The main difference, however, concerns the aguadas, whose IOU has almost doubled for the three-band VAT model compared to the one-band VAT model. In conclusion, the most appropriate visualisation for a deep learning model depends on the feature class. If one wants to use the same model for the three classes, the three-band VAT with ResNet-18 can be considered the most suitable model.

11 Semantic segmentation with the Holmul test set

This section covers Experiment 16 (see Figure 60) in which we achieved the testing of the best deep learning model with the Holmul test set described earlier in Section 5.3. This allows to investigate further the transferability of the model. The Holmul data contains 472 ground truths, digitised following the borders of the original buildings and not the borders of the ruined buildings as was the case for Chactún and G-LiHT. This is visible in Figure 65, where the central image shows the ground truths. Because of this different digitisation, the IOU may no longer be a good performance metric.

The one-band VAT of the area was used. The model tested therefore had to be one that had been trained on the one-band VAT of Chactún. The model with the best performance metrics previously identified for semantic segmentation was the model using ResNet-34. This model was then tested on the area. Since this area contains only buildings, the model trained on buildings was used. A sample of the result can be found in Figure 65.



(a) Input VAT (b) Ground truths (c) Predictions

Figure 65: (a) Input VAT of Holmul, (b) ground truths and (c) predictions.

Visual inspection of the result of the Classify Pixels tool was achieved by searching for ground truths predicted with at least 30% of their pixels. Using this criterion, 253 buildings were predicted, while 219 were missed. Considering that 21% of the buildings in the G-LiHT dataset were well predicted, the Holmul dataset gives better results with a value of 53.6%. This higher value comes from the fact that Holmul's terrain is closer to that of Chactún. However, part of the terrain consists of steep slopes, which leads to a large number of false positives. This is visible in Figure 66. As with the other testing areas, false positives also arise on bulges which could be buildings.

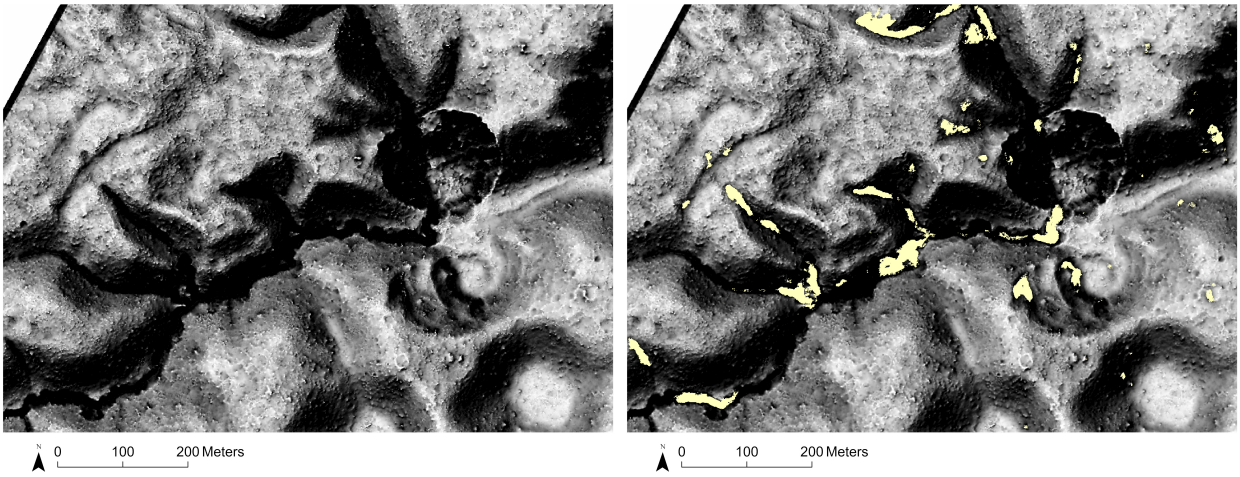


Figure 66: Sample of the results of the semantic segmentation for the U-Net model. Predictions on the right are false positive buildings.

The IOU was calculated for this dataset, keeping in mind that ground truth objects are less precise than they were for training. The number of pixels obtained during the IOU computation can be seen in the Appendix F (Table 33). From the IOU formula, we get a value of 22.2% for the IOU. Thus, even with miss-shaped polygons for the ground truths, the results are better than for the G-LiHT test set, which had an IOU of 17.7%. One can conclude that the most important aspect to achieve a good performance for a model is the similarity between the training and testing terrains.

12 DEM as source layer

An attempt was made to use the DEM instead of the VAT for both training and testing of the deep learning model. The main goal was to see if the G-LiHT area gave better results than previously. Experiment 18, for which the workflow can be seen in Figure 67, investigated two backbones. The training data was exported using the DEM of Chactún South as the source layer. The patches created with this export were normalised to have values ranging from 0 to 255. They were also converted from 32bit to 8bit, since deep learning works better in this case. Semantic segmentation was performed on the DEM, as this computer vision task gave better accuracy than object detection. Again, the three classes were investigated in three different models.

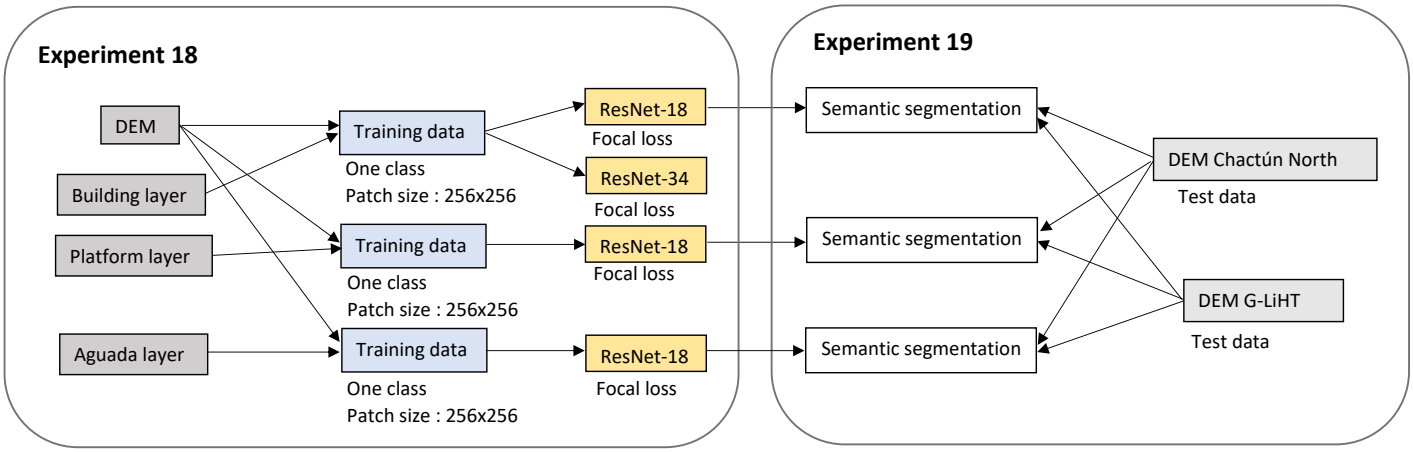


Figure 67: Workflow of experiment 18 and 19.

For semantic segmentation, the backbone architecture which performed the best varied with the visualisation. Both ResNet-18 and ResNet-34 were therefore tested with the DEM model for the building class. The results, which can be seen in Table 26, show that ResNet-18 performs better in the present case. Although the accuracies of the two architectures are very similar with a value of 98%, the recall metric and dice coefficient are higher for ResNet-18. The platform and aguada models were then trained with this backbone. One can already see lower dice coefficients for platforms and aguadas.

Feature class	Backbone	Training loss	Validation loss	Accuracy	Recall	Dice
Buildings	ResNet-18	0.0037	0.0001	98.3%	76.4%	64.8%
	ResNet-34	0.0068	0.0002	98.2%	71.6%	56.8%
Platforms	ResNet-18	2.3739e-05	0.0013	95.8%	69.3%	44.9%
Aguadas	ResNet-18	0.0037	0.0073	93.5%	77.9%	46.3%

Table 26: Results of the training of the U-Net model with the DEM for the three classes.

Experiment 19 (see Figure 67) investigates the impact of the choice of the testing set. The models were first tested on the northern part of Chactún by running the pixel classification tool. A sample of the results is provided in Figure 68. The results are generally less good when using the DEM than when using the VAT. This is especially true for aguadas, of which only one aguada was partially classified as such. However, only one area was incorrectly classified as an aguada. Therefore, the number of false positive predictions is much lower than with the VAT. The IOU has a value of only 1.6% (see Appendix F Table 34 for the details). Concerning the predictions for the buildings, they agree quite well with the ground truths, especially for the buildings on the platforms, which are the most visible. The only missed buildings are individual buildings without a platform. The IOU then has a good value of 57.5%. Finally, more platforms were overlooked compared to buildings, as can be seen in Figure 68. The IOU is then slightly lower at 50.3%. Using the VAT for archaeological detection is therefore a better choice than using the DEM, resulting in higher IOU values. The DEM raster only allows the detection of visible objects with a high elevation discontinuity. The only argument that

could be held against the VAT is the detection of more false positives. However, these may be new discoveries and areas of interest.

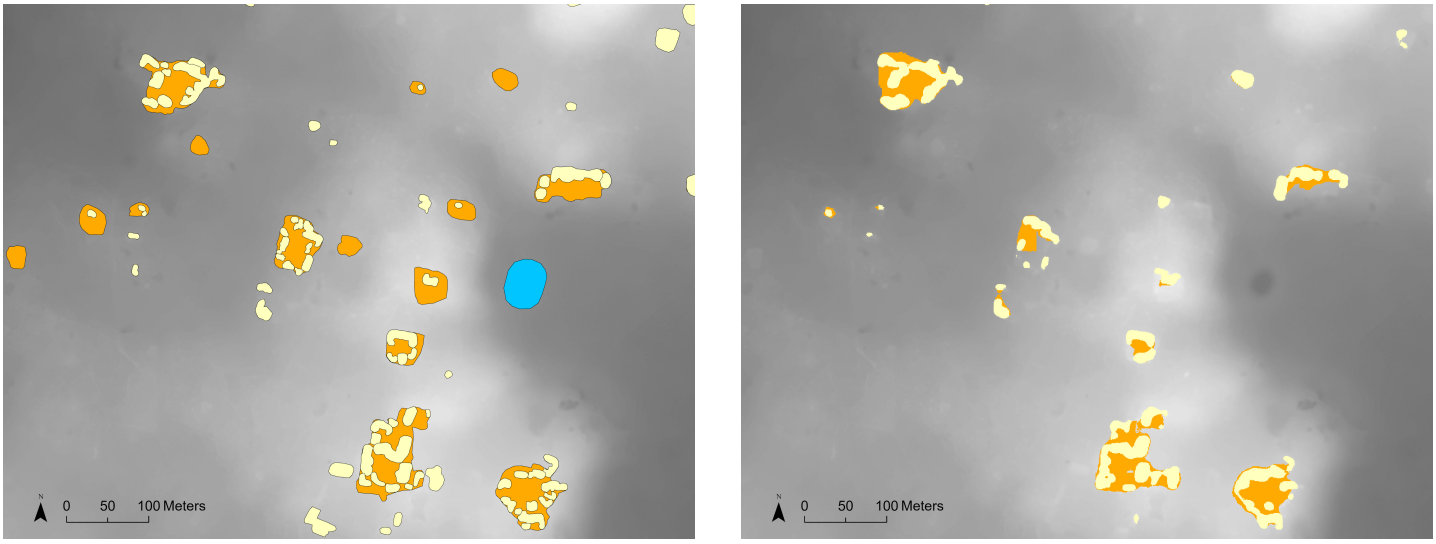


Figure 68: Ground truths on the left and results of the pixel classification on the right. Aguadas are in blue, buildings in yellow and platforms in orange.

The Classify Pixels tool was also applied to the G-LiHT area in Experiment 19 to determine if the cause of the previously poor results was the VAT. Showing a sample of the results is not very useful, as only a few pixels were classified over the entire area. Nevertheless, the interested reader can still refer to the Appendix G for a sample of the results. The aguadas were not detected at all, while the borders of the DEM were misclassified as such. Since no aguadas were detected, no true positives exist - the IOU has a value of 0. As for buildings, the model resulted in very few detections. With only 3 known buildings that were well classified. The number of false positives was lower than when using the VAT, with about 10 false positives for the DEM. The IOU value for buildings is 0.73% (see Appendix F Table 34 for the details). Finally, only 3 platforms had some of their pixels well classified. The IOU for platforms gave a value of 1.7%. Hence, using the DEM does not solve the transferability issue of the model. The results are even worse than with the VAT. It can be concluded that the origin of the bad results when applying the trained model to this test area is the different terrain and data. This is a rather limiting constraint on the transferability of the deep learning model.

13 Influence of the patch size

The patch size has been referred to by various names. In fact, a patch also corresponds to a tile and an image chip. When using ArcGIS Pro, the Export Training Data tool divides an input image into a series of patches, and each patch is an input image to the CNN. The larger the patches, the more contextual information can be derived, while smaller patches better define smaller features. One can also point out that smaller patches are more numerous and therefore require less initial data. With a higher number of patches,

overfitting can also be avoided. However, too small a patch size can lead to a low accuracy, as the model is not submitted to broader contextual information. In contrast, more performant models can be achieved with larger patches, which lead to a reduction of the noise. However, a larger size also requires more computing power [49]. The localisation accuracy also decreases with the patch size [27].

Note that the U-Net network has a predefined size for the input image. This can be seen in the architecture of Figure 15 where the input patch size is 572×572. As a consequence, each patch will be resampled to that size when being fed to the deep learning model.

Patch size is an interesting parameter to play with. Experiment 20, visible in Figure 69, compares six models created with different patch sizes. Note that for simplicity the three feature layers are injected into the training data following the same arrow. However, in reality, one set of training data (and hence one model) was created for each feature layer. One can refer to the workflow of Experiment 17 (Figure 63) where the layers were well separated.

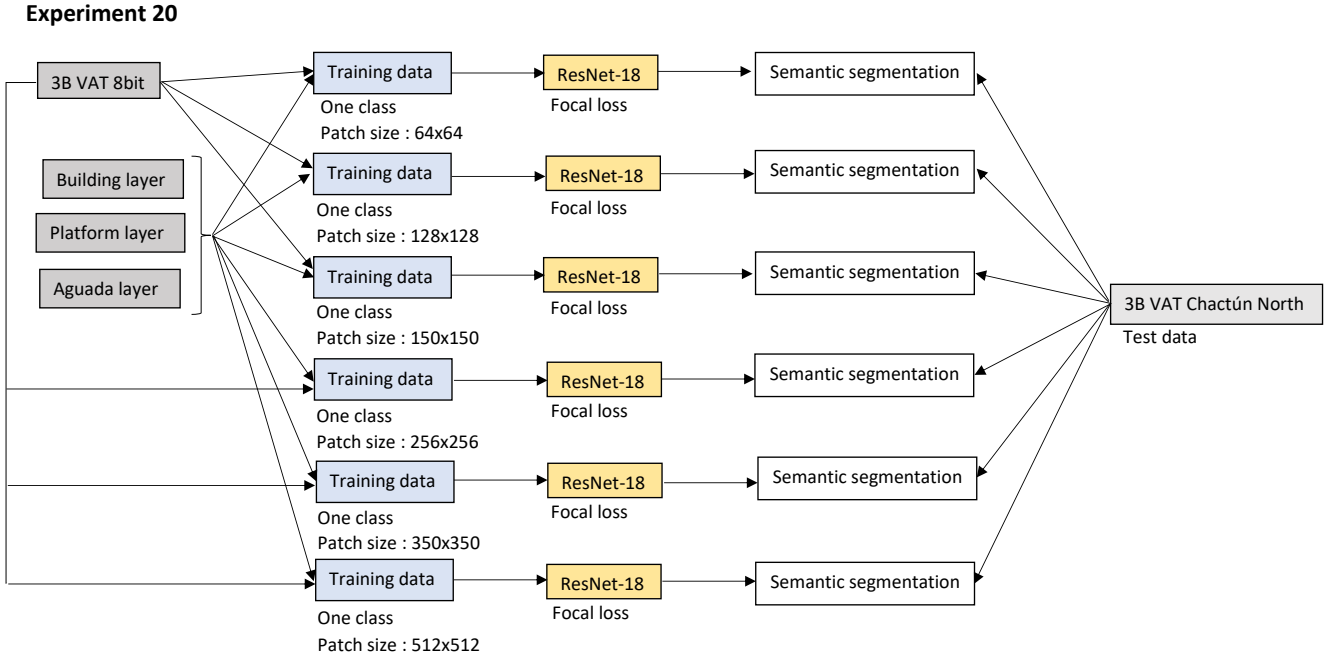


Figure 69: Workflow of experiment 20.

The model and visualisation that previously gave the best results were considered. The computer vision task that gave the best performance metrics was semantic segmentation. Better recall and IOU values were obtained by considering one model for one class of objects, which also allowed both buildings and platforms to be detected when they overlapped. However, the model for the aguadas gave poor results with many false positives. Changing the patch size could have an impact on these false detections. As for the best model architecture, it varied with the visualisation. However, the best results

were obtained with the three-band VAT and ResNet-18 backbone when three classes were considered (see Section 10.2). This model was then trained on each class. The patch size was initially reduced to a value of 150 pixels (the previous value was 256 pixels). Changing the patch size requires to re-export the patches using the Export Training Data tool. The tile size has been set to 150 × 150 pixels, while the stride size has been set to 75 × 75 pixels, in order to keep it as half of the tile. The batch size was set to 15. The results of the training of the three models are summarised in Table 27. The performance metrics are again less good for aguadas.

Feature class	Training loss	Validation loss	Accuracy	Recall	Dice
Buildings	0.0007	0.0004	97.4%	75.3%	69.2%
Platforms	0.0026	0.0025	95.2%	86.6%	71.1%
Aguadas	0.0937	0.0371	85.0%	69.8%	47.5%

Table 27: Results of the training of the U-Net models with the 150 × 150 patch size.

Pixel classification was then carried out using these models to test them on the north of Chactún. Due to the smaller tile size, the processing time is much shorter. The aguada model leads to many more false positives compared to the previous patch size of 256. This can be seen in Figure 70. These false positives do not seem to correspond to any area of interest. In fact, most of the misclassified areas seem to follow the square shape of the tile. Ground truth aguadas were detected except for the deepest one. The IOU amounts to 13.4% (see Appendix F Table 35). The building model is quite good, as can be seen in Figure 71. The IOU value is actually 64.4%. Again, the best predictions are for buildings on platforms. The classified pixels do not follow the boundaries of the tiles, as was the case previously with the aguadas. As for platforms, visual inspection showed that most were detected. Only 14 platforms showed less than 30% of classified pixels.

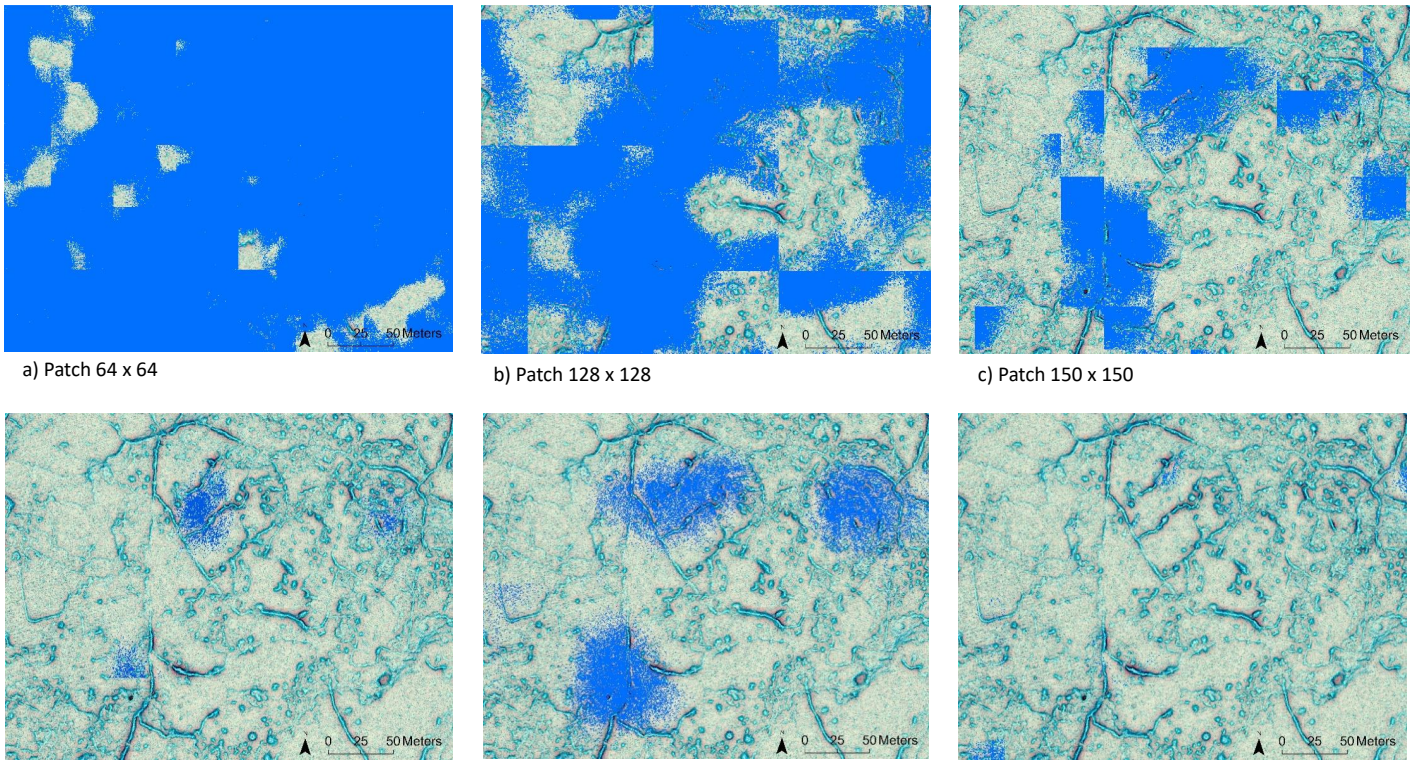
The patch size was then increased to 350 × 350. The tiles were then re-exported with the new size and a stride of 175 × 175. To train the models, the batch size had to be decreased to a value of 10. The result of the training is summarised in the table below. The aguada model has a better accuracy compared to the 150 patch size. However, the dice and recall values are lower. This counterintuitive variation is related to the fact that fewer false positives occur, which leads to a better classification of the background. This increases the overall accuracy. For buildings and platforms, the performance metrics were generally higher with the smaller patch size.

Feature class	Training loss	Validation loss	Accuracy	Recall	Dice
Buildings	1.6194e-05	0.0002	97.9%	64.7%	61.0%
Platforms	5.0723e-05	0.0005	97.2%	64.8%	58.4%
Aguadas	0.0719	0.0088	92.7%	59.7%	37.2%

Table 28: Results of the training of the U-Net models with the 350 × 350 patch size.

Testing the models on Chactún North resulted in IOUs of 16.8% for aguadas, 62.2% for buildings and 64.3% for platforms. The results can be seen in Figure 71. Visual inspection of the classified pixels shows a lower number of false positive aguadas compared to the patch size of 150. All ground truth aguadas were predicted.

Other patch sizes were investigated: 64×64 pixels, 128×128 pixels and 512×512 pixels. These are more common patch sizes as they are of the order of 2^x pixels. The different training results (losses and performance metrics) are provided in Appendix H. A comparison of the results of the different patch sizes for the aguadas detection is provided in Figure 70. The three smallest sizes lead to classified pixels which follow the patches. The smaller the patch size, the higher the amount of false positives. These three sizes are then too small for the aguada model. As for the larger patch sizes, the one leading to the smallest number of false positives on the sample of Figure 70 is 512×512. However, further visual investigation of the classified pixels reveal that this patch size leads to predictions at the edges and outside of the VAT. For that reason, visual inspection suggests that the optimal patch size for detecting aguadas is 256×256. This can be further verified with the IOU values which are summarized in Table 29. The 256×256 patch indeed gives the highest IOU for aguadas. As for buildings and platforms, the classified pixels for the different patches can be seen in Figure 71. This figure, as well as the IOU values, show that buildings are better predicted with the smaller patch size of 128×128. As for platforms, the best IOU is obtained for 150×150. However, the platform detections are very similar for many patch sizes (with very close IOUs for 150, 256, 350 and 512). Only 64×64 is too small for the model to detect well platforms. If one wants to use the same patch size for the three objects (buildings, platforms and aguadas), the best patch size is 256×256 pixels, which gives a good model performance for the three objects.



a) Patch 64 x 64 b) Patch 128 x 128 c) Patch 150 x 150

d) Patch 256 x 256 e) Patch 350 x 350 f) Patch 512 x 512

Figure 70: Aguada predictions for a patch size of (a) 64×64, (b) 128×128, (c) 150×150, (d) 256×256, (e) 350×350 and (f) 512×512. All predictions are false positives.

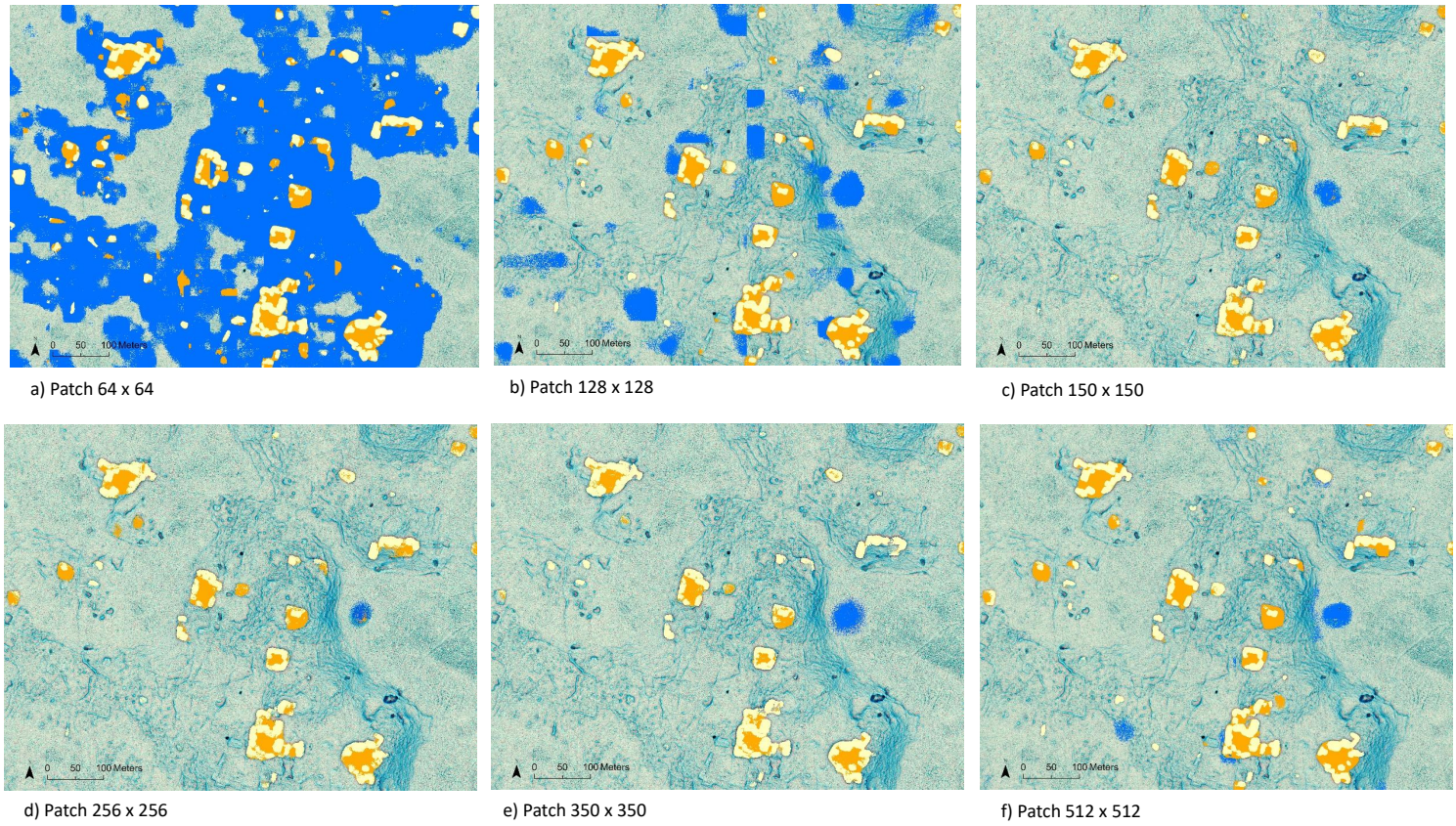


Figure 71: Sample of the results of the semantic segmentation for the U-Net model with a patch size of (a) 64×64, (b) 128×128, (c) 150×150, (d) 256×256, (e) 350×350 and (f) 512×512. Aguadas are in blue, buildings in yellow and platforms in orange.

	64×64	128×128	150×150	256×256	350×350	512×512
Buildings	52.5%	**68.0%**	64.4%	63.7%	62.2%	66.9%
Platforms	41.2%	63.0%	**64.8%**	64.7%	64.3%	64.2%
Aguadas	0.53%	2.6%	13.4%	**28.3%**	16.8%	13.9%

Table 29: Comparison of the IOU values for the six patch sizes. Best values are in bold.

We can then conclude that for larger objects such as aguadas, a larger patch size is more appropriate. However, when the patch size becomes too large, the IOU metric decreases. This could be due to the fact that more false detections occur when the patch size is too large. On the other hand, for smaller features, such as buildings, a smaller patch provides better performance. Platforms have a range of sizes that allows them to be detected well in most patch sizes.

14 ArcGIS StoryMap

The primary aim of this thesis was to create a StoryMap for archaeologists. The idea was to show archaeologists how to implement a deep learning model in ArcGIS Pro that would help them to detect new archaeological features in an automated manner. A nice way to write about the ArcGIS Pro workflow is to use a StoryMap. The ArcGIS StoryMaps platform offers the possibility to create a story and make it interactive for the reader, e.g. through the use of personalised maps. The StoryMap created can be found at *https://arcg.is/1aqK0v0*. The different steps required in ArcGIS Pro to use deep learning are described in detail, from creating the training data, training the model, to testing this model on a new area. For the StoryMap, the case of semantic segmentation was chosen because of the suitability of the dataset for this type of computer vision task.

15 Conclusion

Areas of interest for archaeologists can nowadays be investigated by more than just a field survey. Remote sensing enables archaeologists to look at an area from a distance. In particular, airborne laser scanning (ALS, lidar) can provide visualisations of the area which serve archaeological needs. Such a visualisation is the visualisation for archaeological topography (VAT), which provides a clear view of archaeological remains. A second field can be used to help archaeologists in their work, which is artificial intelligence. Deep learning, especially semantic segmentation, was shown to be efficient in annotating archaeological features over a new area acquired by ALS. This automatic prediction of objects can facilitate the work of archaeologists. The inspection of an area of interest, which is necessary before a field expedition, could be greatly accelerated by the use of deep learning. The goal of this study was to investigate several deep learning models and their performance in finding new archaeological features when applied to lidar-derived images. Both object detection and semantic segmentation were investigated, as well as several backbone architectures and patch sizes. Two visualisations for archaeological topography were studied, as well as the digital elevation model. Three testing datasets were used to assess the performance of the models.

The trained deep learning models show recall values generally above 70%. The major result that emerges from all the processing of these models is that their transferability is limited. Indeed, a model trained on a particular dataset does not generalise well to other datasets if the terrain or data are too different. One solution that could be investigated, is to use transfer learning to fine-tune the given model on the new data. However, this would here require some data augmentation to increase the number of training samples of the G-LiHT dataset.

The best deep learning model depends on the nature of the object annotations. Indeed, objects annotated for the training set by following the edges are better suited for semantic segmentation, i.e. for the U-Net deep learning model. Object detection can also be used, although the accuracy is lower for this task. This comes also from the overlapping between different types of features, such as buildings and platforms. This requires to train one model for each class of objects. When all classes are considered in the same model, the Non Maximum Suppression parameter should not be used to detect building/platform pairs.

The separation of the three classes is also necessary for semantic segmentation. Otherwise, platforms under a set of buildings cannot be predicted. Better accuracies are achieved with feature class separation. However, the amount of training examples of aguadas is too small to obtain a good model performance for this feature class. Using data augmentation techniques to increase this number of examples is a possibility that could be further explored.

The best backbone architecture differs for different types of visualisations. For semantic segmentation, the best performing backbone was ResNet-18 for the three-band VAT and ResNet-34 for the one-band VAT. Another interesting result is that deep learning in ArcGIS Pro only works for visualisations with a pixel depth of 8bit. No single visuali-

sation was identified as the best one in any case. In fact, it depends on both the task performed and the feature class. When considering one class of archaeological features and a similar terrain for testing and training, the one-band VAT (using ResNet-34) gave the best results in both object detection and semantic segmentation. While when considering one deep learning model per class, the best visualisation varied depending on the objects. The comparison between the VAT and the DEM showed that the VAT leads to a more performant deep learning model.

Natural formations and other man-made structures that resemble considered classes, e.g. rock piles, outcrops etc, are often misclassified by the models. One should bear in mind that a false positive prediction on the testing set could be an unidentified object that was overlooked in the definition of ground truths. However, rugged terrain that does not seem to show any possible objects is sometimes misclassified. Another important result is that the models predict a platform for each cluster of buildings. This shows that the models learned well since it makes sense to find a platform under a group of buildings, even though it was not identified during annotation of the ground truths.

The results show that the patch size of 256×256 is appropriate for the studied objects, especially for aguadas. This is mainly due to the size of the archaeological features available. A smaller patch size of 128×128 also leads to a good intersection over union (IOU) value for buildings, as these objects have a smaller size.

We can conclude that manual image inspection could be enhanced by the application of a deep learning model. Improvements can still be made by fine-tuning the parameters to find the model with the very best performance. However, this will always depend on the type of data used and the application. The models studied here have already made it possible to identify archaeological features on a new area in a promising way. It could be argued that the time saved by using a deep learning model to detect structures is compensated by the need for manual verification. However, this verification would take much less time if the structures have already been delineated automatically. Finally, the ArcGIS StoryMap created in this study demonstrates the use of deep learning with ALS data for archaeological detections. An ArcGIS Pro workflow is provided for archaeologists.

A Appendix : Deep learning workflow

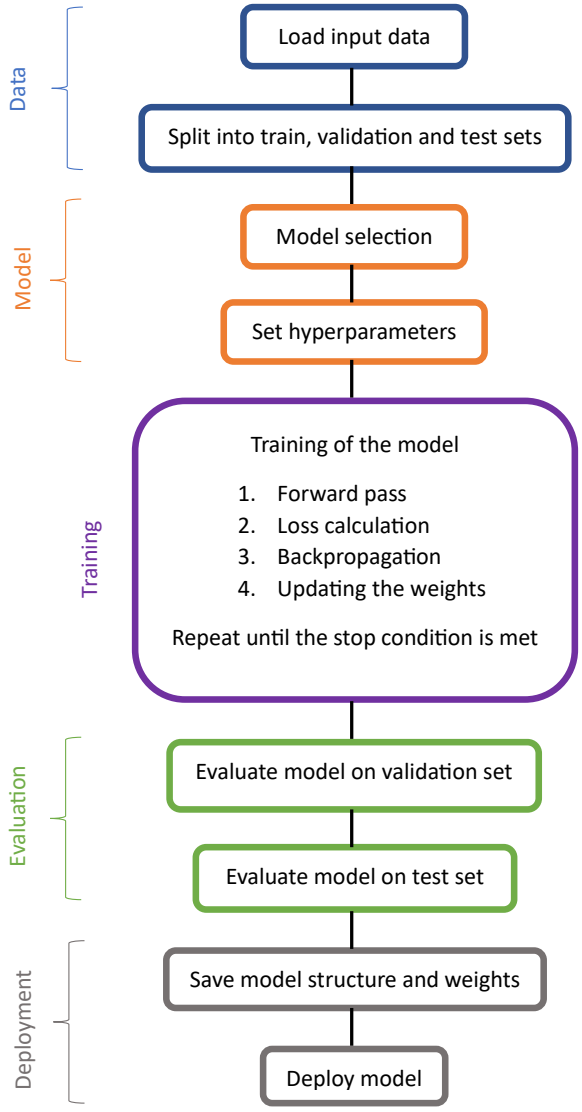


Figure 72: Simplified deep learning workflow. Inspired from a diagram of Maya Somrak (ZRC SAZU).

B Appendix : Combine tool attribute table

OBJECTID	Count	Classified _pixels	Test_buildings _new	Test_platforms _new	Test_aguadas _new
1	61,400,195	0	0	0	0
2	19,500	0	0	1	0
3	52,407	0	1	0	0
4	2,775	1	0	0	0
5	1,382	1	1	0	0
6	7,675	0	0	0	1
7	5,810	1	0	0	1
8	4,811	1	1	0	1
9	3,885	0	1	0	1

Table 30: Attribute table of the result of the Combine tool of ArcGIS Pro.

C Appendix : Feature tiles

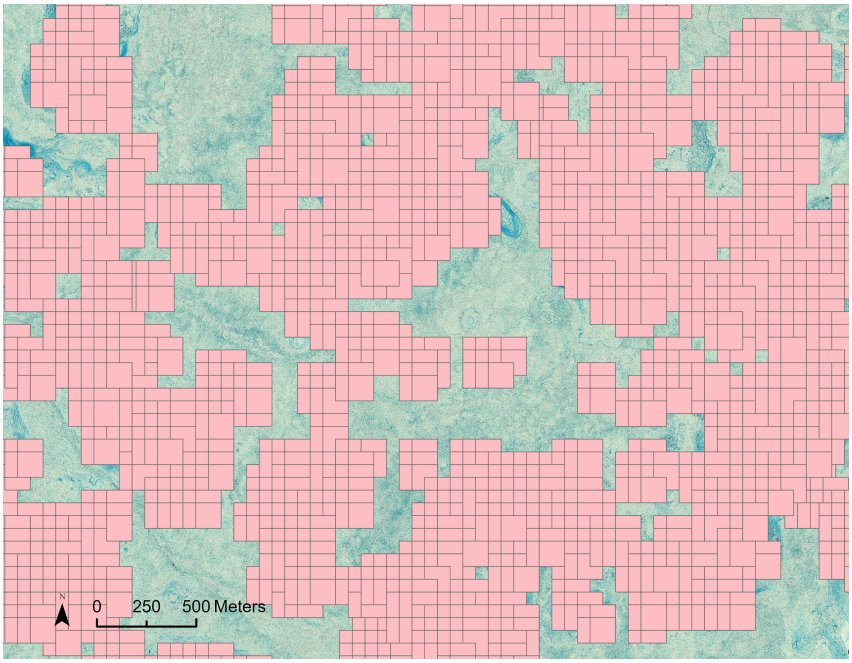


Figure 73: Tiles containing archaeological features above the three-band VAT. One can see the 50% overlap between two tiles and the meaning of exporting only tiles with features.

D Appendix : Workflow diagrams

In the workflow diagrams, when "annotation layers" are injected into the training data, all three feature classes are considered as one (the archaeological feature class) and injected into a single model. In contrast, in the rightmost part of the workflows, each layer is considered separately and a separate model is trained for each class. Note that, for semantic

segmentation, part of the experiment where different patch sizes were investigated could not be added on the diagram because of a lack of space. The three patch sizes of 64×64, 128×128 and 512×512 are missing.

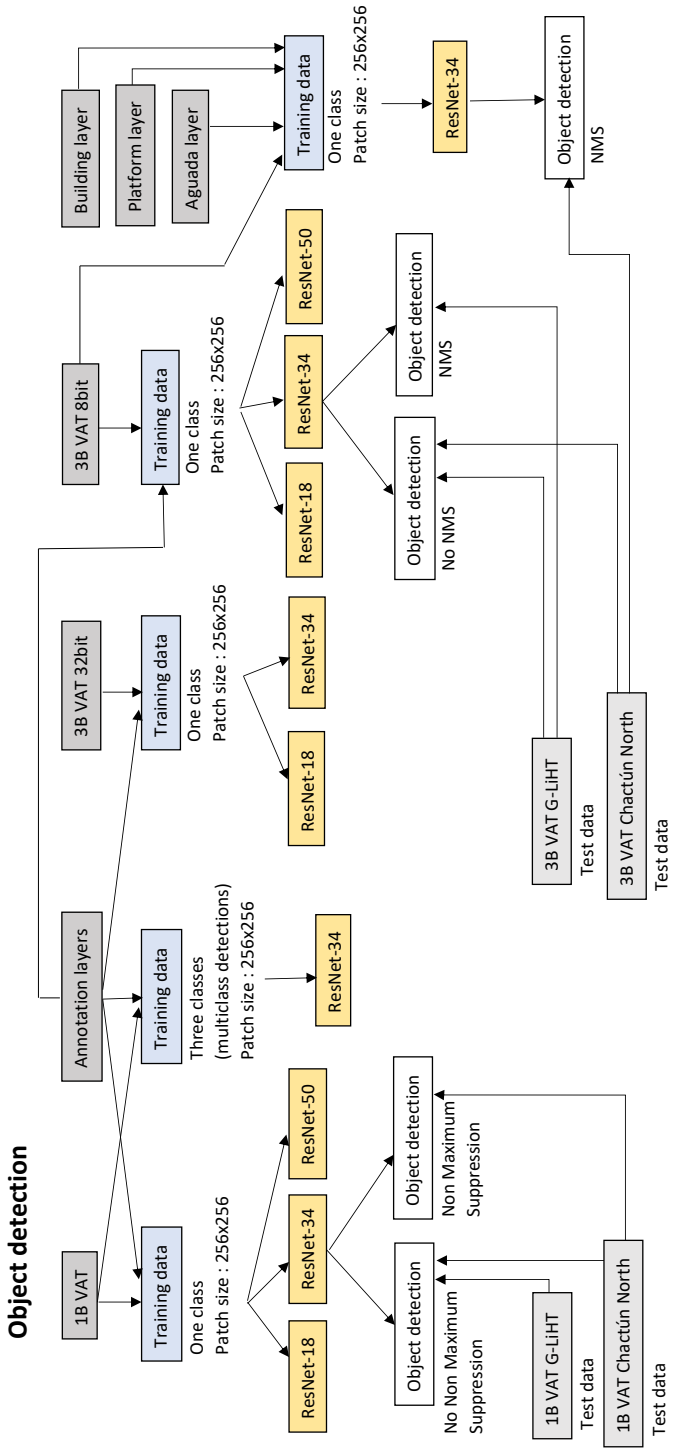


Figure 74: Workflow diagram of the object detection experiments. The order of experiments is as they are described in the text.

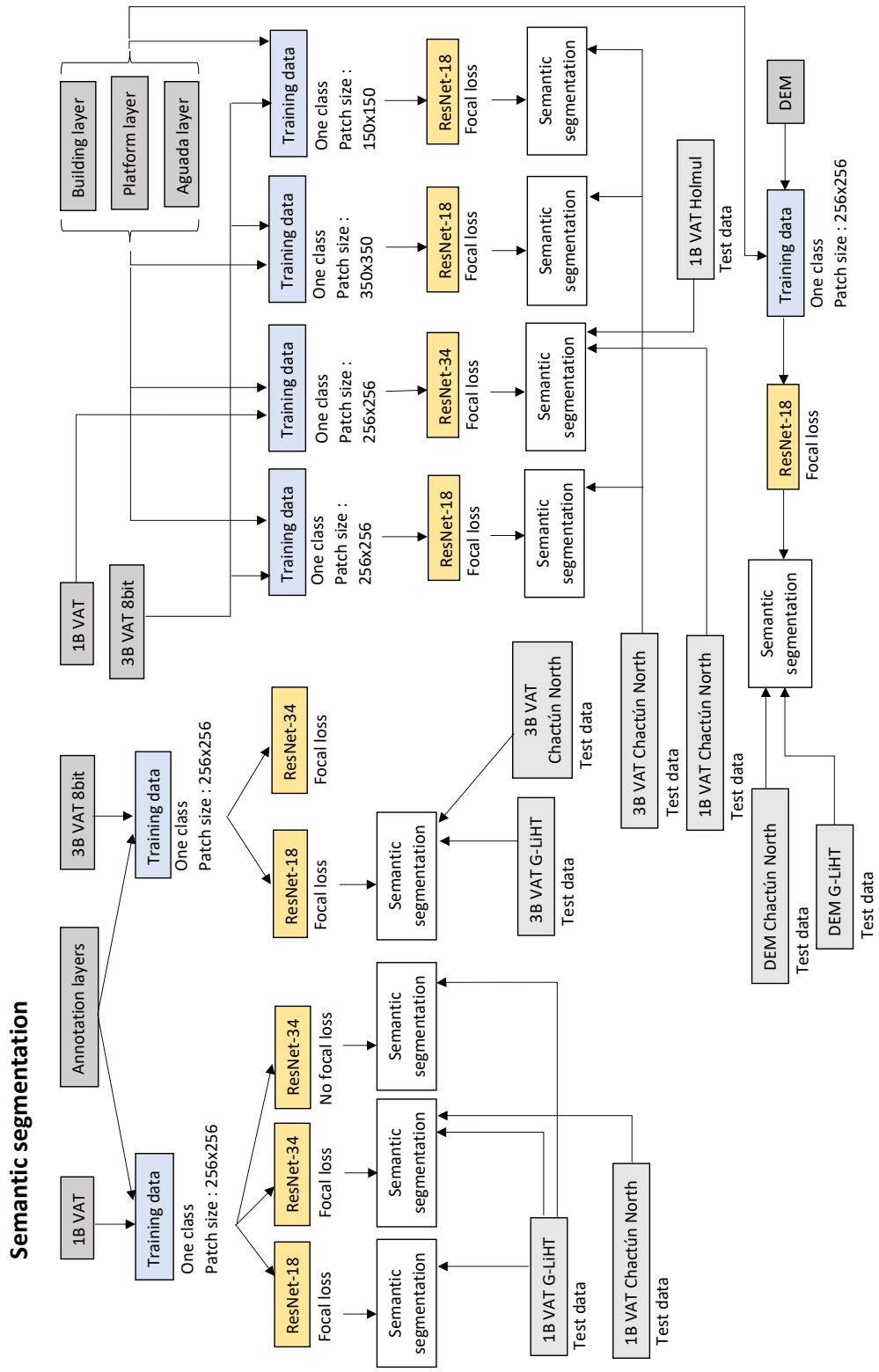


Figure 75: Workflow diagram of the semantic segmentation experiments.

E Appendix : Loss graphs and sample of the results

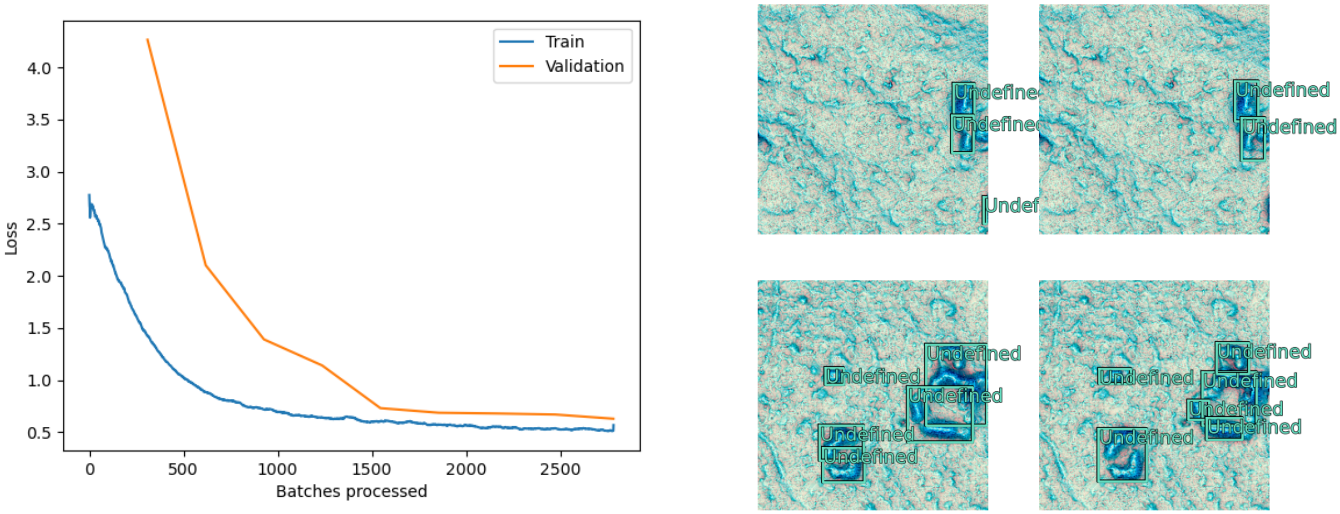


Figure 76: Loss graph for the ResNet-18 model with the three-band VAT 32bit on the left. Sample results of the model on the right, with ground truths on the left and predictions on the right.

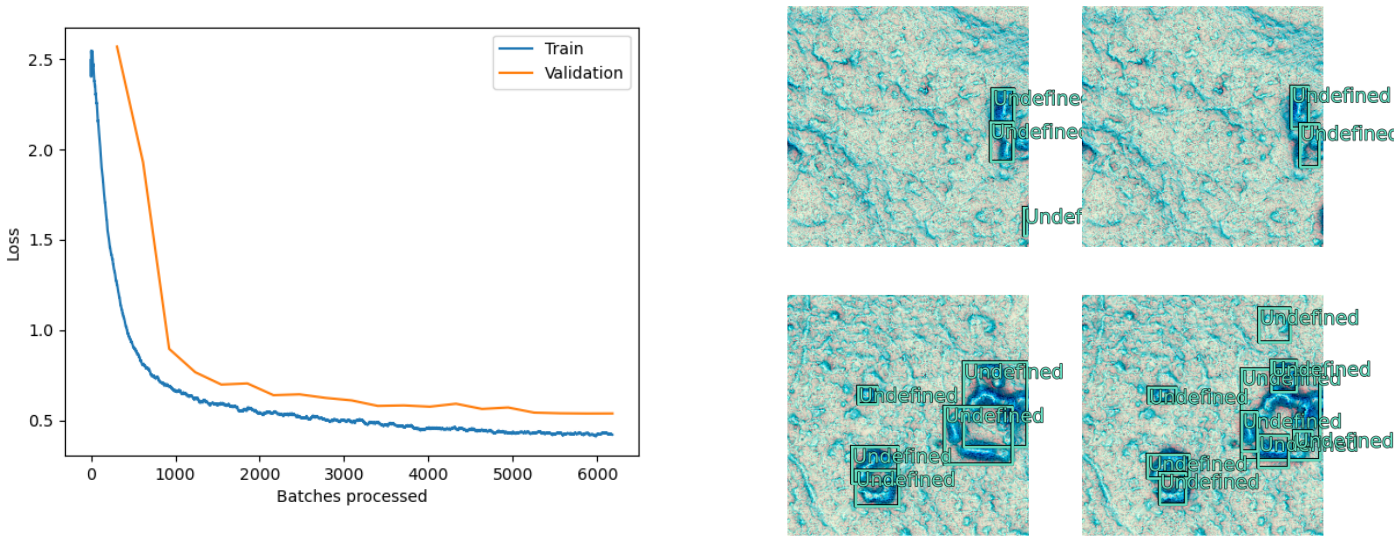


Figure 77: Loss graphs of RetinaNet ResNet-18 with three-band VAT 8bit on the left. Sample results of the model on the right, with ground truths on the left and predictions on the right.

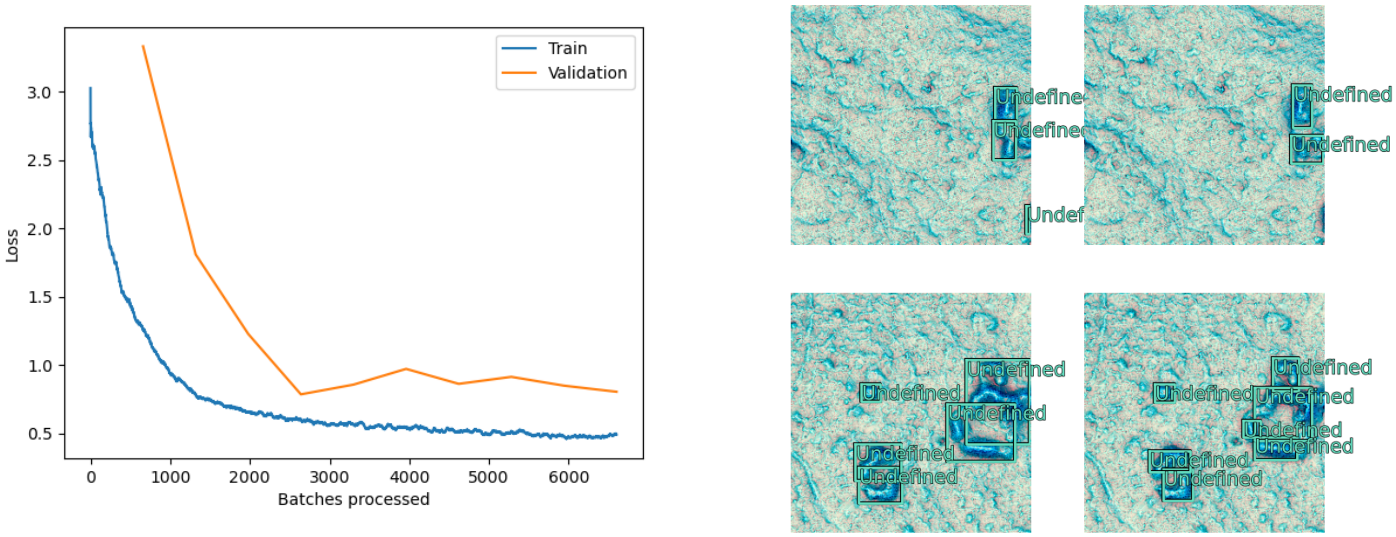


Figure 78: Loss graphs of RetinaNet ResNet-50 with three-band VAT 8bit on the left. Sample results of the model on the right, with ground truths on the left and predictions on the right.

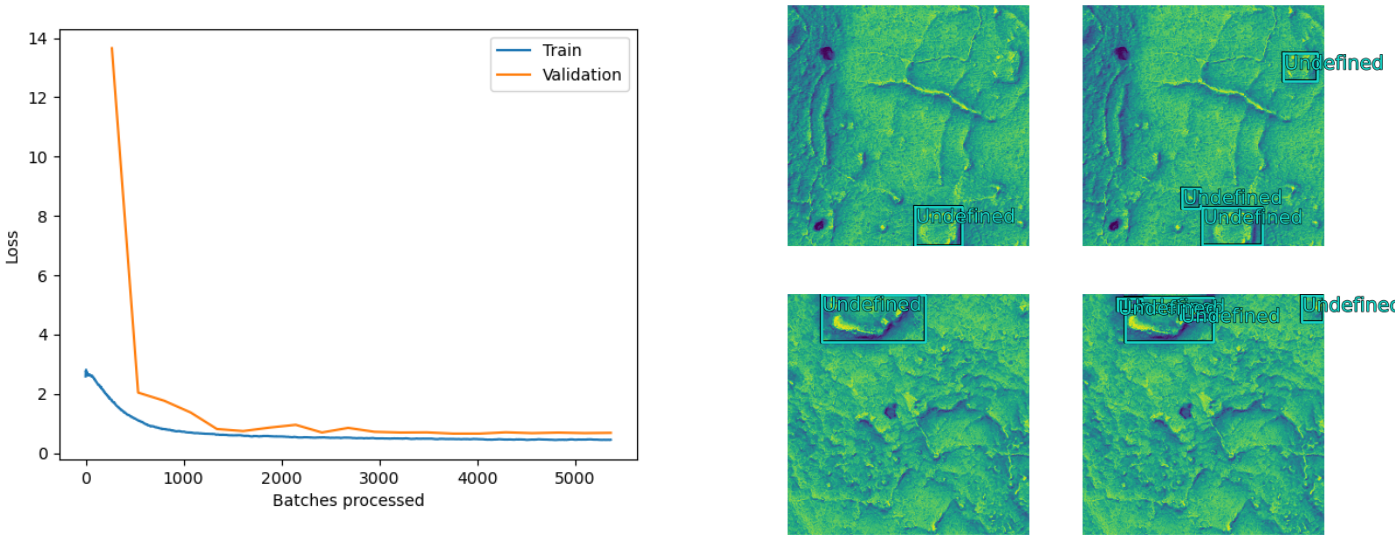


Figure 79: Loss graphs of RetinaNet ResNet-34 with the one-band VAT trained on Chactún South on the left. Sample results of the model on the right, with ground truths on the left and predictions on the right.

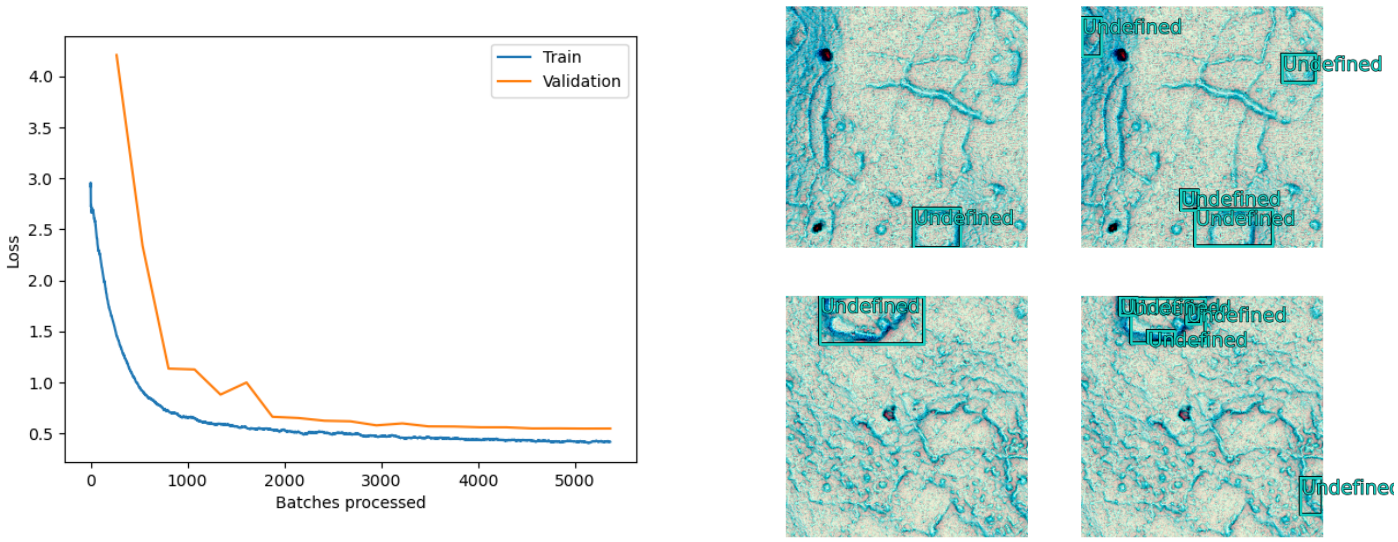


Figure 80: Loss graphs of RetinaNet ResNet-34 with the three-band VAT trained on Chactún South on the left. Sample results of the model on the right, with ground truths on the left and predictions on the right.

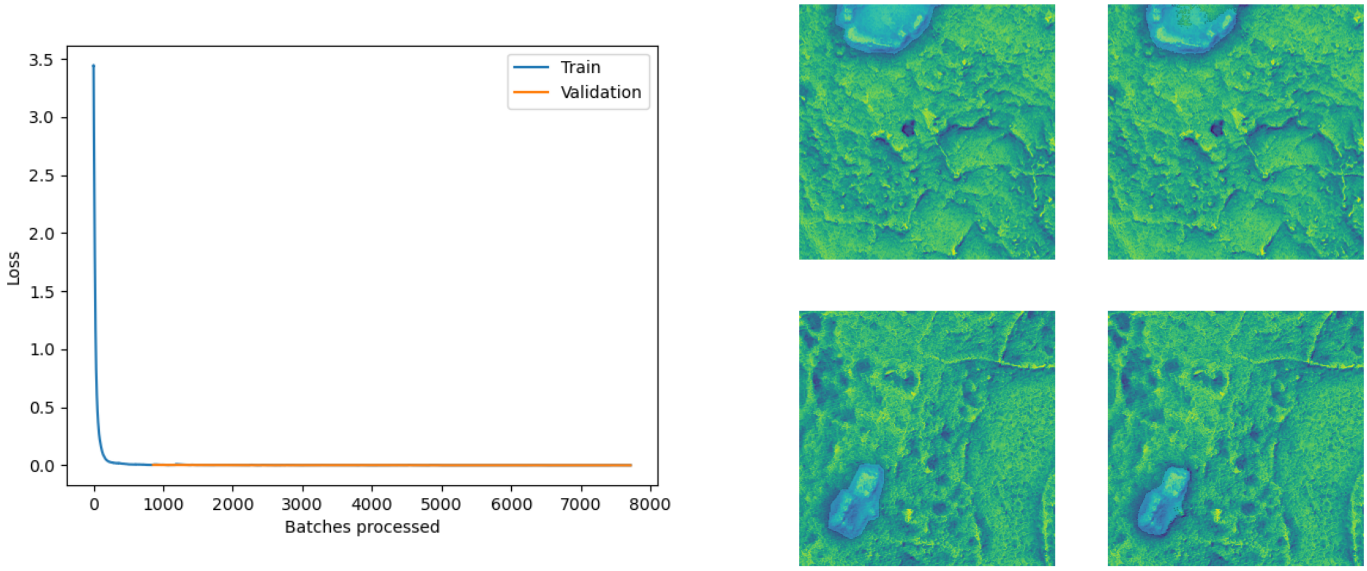


Figure 81: Loss graphs of U-Net ResNet-34 with one-band VAT trained on Chactún South on the left. Sample results of the model on the right, with ground truths on the left and predictions on the right.

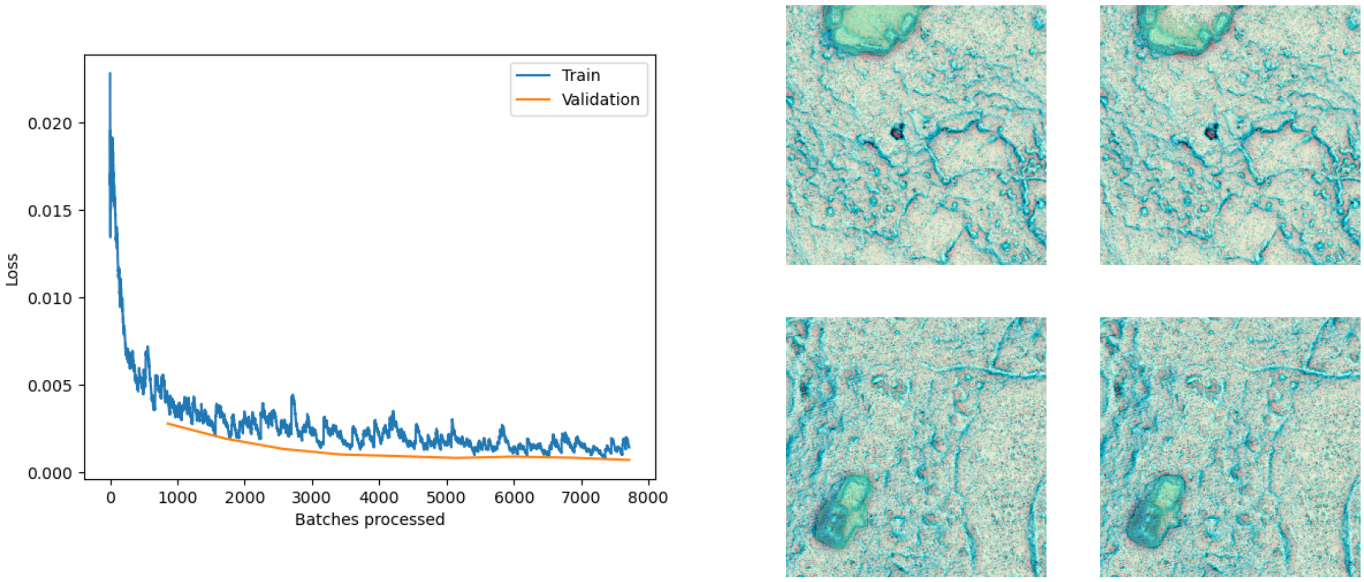


Figure 82: Loss graphs of U-Net ResNet-18 with three-band VAT trained on Chactún South on the left. Sample results of the model on the right, with ground truths on the left and predictions on the right.

F Appendix : Results of Combine tool

Model	Amount of true positives	Amount of false positives	Amount of false negatives
U-Net ResNet-18	14,648	8,179	80,111
U-Net ResNet-34	22,395	31,807	72,364

Table 31: Amounts of pixels obtained from the Combine tool for G-LiHT.

Model	Feature class	Amount of true positives	Amount of false positives	Amount of false negatives
Three-band U-Net ResNet-18	Buildings	688,270	150,860	240,542
	Platforms	1,198,736	388,471	264,147
	Aguadas	140,880	333,506	23,617

Table 32: Amounts of pixels obtained from the Combine tool for Chactún North.

Model	Feature class	Amount of true positives	Amount of false positives	Amount of false negatives
U-Net ResNet-34	Buildings	144,407	332,261	174,218

Table 33: Amounts of pixels obtained from the Combine tool for Holmul.

Model	Testing area	Feature class	Amount of true positives	Amount of false positives	Amount of false negatives
DEM U-Net ResNet-18	Chactún North	Buildings	583,119	85,923	345,693
		Platforms	876,815	278,845	586,068
		Aguadas	2,856	11,058	161,641
	G-LiHT	Buildings	2,514	21,104	320,518
		Platforms	4,496	62,942	190,658

Table 34: Amounts of pixels obtained from the Combine tool for the DEM.

Model	Patch size	Feature class	Amount of true positives	Amount of false positives	Amount of false negatives
U-Net ResNet-18	64×64	Buildings	790,064	576,857	138,748
		Platforms	1,319,009	1,741,677	143,874
		Aguadas	157,565	29,627,752	6,932
	128×128	Buildings	751,185	175,718	177,627
		Platforms	1,337,325	660,599	125,558
		Aguadas	155,832	5,841,095	8,665
	150×150	Buildings	688,133	139,488	240,679
		Platforms	1,190,034	373,557	272,849
		Aguadas	143,755	904,309	20,742
	350×350	Buildings	713,907	219,225	214,905
		Platforms	1,077,272	212,431	385,611
		Aguadas	147,073	709,042	17,424
	512×512	Buildings	738,218	174,789	190,594
		Platforms	1,331,195	609,914	131,688
		Aguadas	126,970	749,203	37,527

Table 35: Amounts of pixels obtained from the Combine tool for the different patch sizes.

G Appendix : Classified pixels on G-LiHT with the DEM

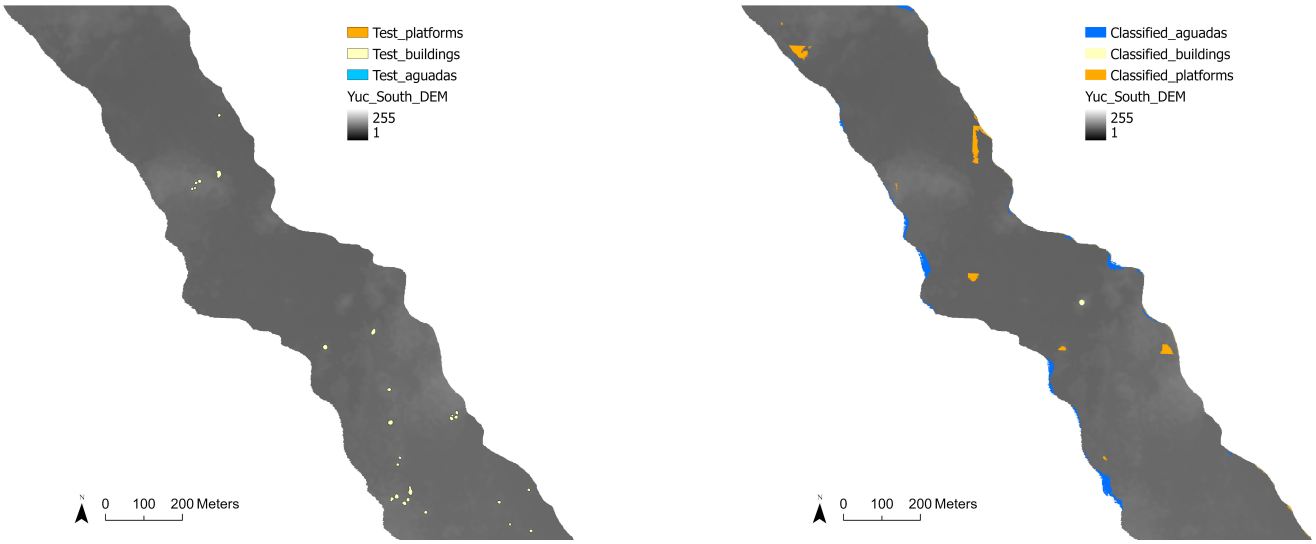


Figure 83: Sample of the results of the pixel classification for the U-Net models of the three classes with the DEM as source layer. Ground truths are on the left and predictions on the right.

H Appendix : Training results for the patch sizes 64×64, 128×128 and 512×512

Feature class	Training loss	Validation loss	Accuracy	Recall	Dice
Buildings	0.0021	0.0032	94.9%	81.6%	73.8%
Platforms	0.0144	0.0120	91.8%	86.1%	74.7%
Aguadas	0.1153	0.0874	74.7%	74.2%	48.5%

Table 36: Results of the training of the U-Net models with the 64×64 patch size.

Feature class	Training loss	Validation loss	Accuracy	Recall	Dice
Buildings	0.0002	0.0003	97.3%	78.1%	72.7%
Platforms	6.3618e-05	0.0018	95.7%	87.7%	73.8%
Aguadas	0.0384	0.0514	81.4%	69.4%	42.9%

Table 37: Results of the training of the U-Net models with the 128×128 patch size.

Feature class	Training loss	Validation loss	Accuracy	Recall	Dice
Buildings	0.0001	3.58e-05	98.9%	72.8%	65.4%
Platforms	0.0118	0.0034	95.7%	7.2%	10.2%
Aguadas	0.0101	0.0367	89.3%	2.9%	4.0%

Table 38: Results of the training of the U-Net models with the 512×512 patch size.

References

[1] GARMIN. Garmin on Mars. https://www.garmin.com/en-US/blog/general/garmin-on-mars/. [Online; accessed 29-03-2023].

[2] Steve Snow. Lidar Images Show Mayan Civilization in a New Light. https://www.esri.com/about/newsroom/blog/lidar-images-reveal-mayan-civilization/. [Online; accessed 06-04-2023].

[3] Maja Somrak. Deep Learning. *ZRC SAZU*, 2023.

[4] Gilles Louppe. Deep Learning. *Deep Learning course INFO8010*. University of Liège, Belgium.

[5] Šprajc I, Marsetič A, Štajdohar J, Dzul Góngora S, Ball JW, Esparza Olguín O, and et al. Archaeological landscape, settlement dynamics, and sociopolitical organization in the Chactún area of the central Maya Lowlands. *PLoS ONE 17(1)*, (e0262921), 2022.

[6] Maja Somrak, Sašo Džeroski, and Žiga Kokalj. Learning to Classify Structures in ALS-Derived Visualizations of Anciant Maya Settlements with CNN. *Remote Sensing*, (10.3390/rs12142215), 2020.

[7] Ali Momennasab. Machine Learning for Mars Exploration. https://arxiv.org/pdf/2111.11537.pdf. [Online; accessed 15-02-2023].

[8] Yannis Chaouche. Initiez-vous au Machine Learning. https://openclassrooms.com/en/courses/4011851-initiez-vous-au-machine-learning. [Online; accessed 01-02-2023].

[9] Pierre Geurts. Introduction. *Introduction to Machine Learning course ELEN062-1*. University of Liège, Belgium.

[10] Knowledge Transfer. How to choose cross-entropy loss function in Keras? https://androidkt.com/choose-cross-entropy-loss-function-in-keras/. [Online; accessed 23-03-2023].

[11] Ajitesh Kumar. Overfitting Underfitting in Machine Learning. https://vitalflux.com/overfitting-underfitting-concepts-interview-questions/. [Online; accessed 01-03-2023].

[12] Gustau Camps-Valls, Devis Tuia, Xiao Xiang Zhu, and Markus Reichstein. *Deep Learning for the Earth Sciences*. Wiley, 2021.

[13] Ajitesh Kumar. Machine Learning – Training, Validation Test Data Set. https://vitalflux.com/machine-learning-training-validation-test-data-set/. [Online; accessed 22-02-2023].

[14] Forecast Global. Feature Engineering. https://corporatefinanceinstitute.com/resources/data-science/feature-engineering/. [Online; accessed 01-03-2023].

[15] Romain Herault and Clement Chatelain. Initiez-vous au Deep Learning. https://openclassrooms.com/en/courses/5801891-initiez-vous-au-deep-learning. [Online; accessed 10-02-2023].

[16] Roza Dastres and Mohsen Soori. Artificial Neural Network Systems. *International Journal of Imaging and Robotics (IJIR).*

[17] MathWorks. What Is a Convolutional Neural Network? https://www.mathworks.com/discovery/convolutional-neural-network-matlab.html. [Online; accessed 07-03-2023].

[18] MLNotebook. Convolutional Neural Networks - Basics. https://mlnotebook.github.io/post/CNN1/. [Online; accessed 07-03-2023].

[19] Leo Pauly, Harriet Peel, Shan Luo, David Hogg, and Raul Fuentes. Deeper Networks for Pavement Crack Detection. https://www.researchgate.net/publication/319235847_Deeper_Networks_for_Pavement_Crack_Detection. [Online; accessed 07-03-2023].

[20] Papers with Code. Max Pooling. https://paperswithcode.com/method/max-pooling. [Online; accessed 27-03-2023].

[21] O'REILLY. Chapter 4. Object Detection and Image Segmentation. https://www.oreilly.com/library/view/practical-machine-learning/9781098102357/ch04.html#complete_view_of_the_retinanet_architect. [Online; accessed 27-02-2023].

[22] ArcGIS Developers. How single-shot detector (SSD) works? https://developers.arcgis.com/python/guide/how-ssd-works/. [Online; accessed 23-02-2023].

[23] ArcGIS Developers. How RetinaNet works? https://developers.arcgis.com/python/guide/how-retinanet-works/. [Online; accessed 23-02-2023].

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. *Cornell University*, (arXiv:1512.03385), 2015.

[25] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal Loss for Dense Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42:318–327, 2017.

[26] ArcGIS Developers. How U-net works? https://developers.arcgis.com/python/guide/how-unet-works/. [Online; accessed 07-03-2023].

[27] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *Computer Science Department and BIOSS Centre for Biological Signalling Studies, University of Freiburg, Germany*, 2015.

[28] ArcGIS Pro. How Compute Accuracy For Object Detection works. https://pro.arcgis.com/en/pro-app/latest/tool-reference/image-analyst/how-compute-accuracy-for-object-detection-works.htm. [Online; accessed 07-03-2023].

[29] Konovalenko Ihor, Maruschak Pavlo, Brezinová Janette, Prentkovskis Olegas, and Brezina Jakub. Research of u-net-based cnn architectures for metal surface defect detection. *Machines*, (10.3390/machines10050327), 2022.

[30] Lillesand, Kiefer, and Chipman. *Remote Sensing and Image Interpretation.* Wiley, United States, 2015.

[31] Pinliang Dong and Qi Chen. LiDAR Remote Sensing and Applications. *Taylor Francis Group*, (978-1-4822-4301-7), 2018.

[32] Juan Carlos Fernandez-Diaz, William E. Carter, Ramesh L. Shrestha, and Craig L. Glennie. Now you see it... now you don't: Understanding airborne mapping lidar collection and data product generation for archaeological research in mesoamerica. *Remote Sensing*, (10.3390/rs6109951), 2014.

[33] Eric Ariel L Salas. Waveform LiDAR concepts and applications for potential vegetation phenology monitoring and modeling: a comprehensive review. *Geo-spatial Information Science*, (24:2, 179-200), 2021.

[34] Žiga Kokalj and Maja Somrak. Why Not a Single Image? Combining Visualizations to Facilitate Fieldwork and On-Screen Mapping. *Remote sensing*, (10.3390/rs11070747), 2019.

[35] Benjamin Štular, Edisa Lozić, and Stefan Eichert. Airborne LiDAR-Derived Digital Elevation Model for Archaeology. *Remote Sensing*, (10.3390/rs13091855), 2021.

[36] Takeshi Inomata, Flory Pinzón, José Luis Ranchos, Tsuyoshi Haraguchi, Hiroo Nasu, Juan Carlos Fernandez-Diaz, and et al. Archaeological Application of Airborne LiDAR with Object-Based Vegetation Classification and Visualization Techniques at the Lowland Maya Site of Ceibal, Guatemala. *Remote Sensing*, (10.3390/rs9060563), 2017.

[37] Xia Li and Gregory W. McCarty. Application of Topographic Analyses for Mapping Spatial Patterns of Soil Properties. *Earth Observation and Geospatial Analyses*, 2019.

[38] Charles Golden, Timothy Murtha, Bruce Cook, Derek S. Shaffer, and Whittaker Schroder et al. Reanalyzing environmental lidar data for archaeology : Mesoamerican applications and implications. *Journal of Archaeological Science: Reports*, 2016.

[39] Walter Witschey and Clifford Brown. The Electronic Atlas of Ancient Maya Sites. 2023.

[40] Marcello A. Canuto, Francisco Estrada-Belli, Thomas G. Garrison, Stephen D. Houston, and Mary Jane Acuña et al. Ancient lowland maya complexity as revealed by airborne laser scanning of northern Guatemala. *Science*, 2018.

[41] ArcGIS Pro. Export Training Data For Deep Learning (Image Analyst). `https://pro.arcgis.com/en/pro-app/latest/tool-reference/image-analyst/export-training-data-for-deep-learning.htm`. [Online; accessed 15-02-2023].

[42] ArcGIS Pro. Train Deep Learning Model (Image Analyst). https://pro.arcgis.com/en/pro-app/latest/tool-reference/image-analyst/train-deep-learning-model.htm. [Online; accessed 15-02-2023].

[43] ArcGIS Pro. Detect Objects Using Deep Learning (Image Analyst). https://pro.arcgis.com/en/pro-app/latest/tool-reference/image-analyst/detect-objects-using-deep-learning.htm. [Online; accessed 15-02-2023].

[44] Nikita Kozodoi. Test-Time Augmentation for Tabular Data. https://kozodoi.me/blog/20210908/tta-tabular. [Online; accessed 05-05-2023].

[45] ArcGIS Pro. Classify Pixels Using Deep Learning (Image Analyst). https://pro.arcgis.com/en/pro-app/latest/tool-reference/image-analyst/classify-pixels-using-deep-learning.htm. [Online; accessed 08-03-2023].

[46] ArcGIS Pro. Assess point cloud training results. https://pro.arcgis.com/en/pro-app/latest/help/data/las-dataset/assessing-point-cloud-training-results.htm. [Online; accessed 24-02-2023].

[47] ArcGIS Pro. Resample (Data Management). https://pro.arcgis.com/en/pro-app/latest/tool-reference/data-management/resample.htm. [Online; accessed 02-03-2023].

[48] Panagiotis Antoniadis. Differences Between Epoch, Batch, and Mini-batch. https://www.baeldung.com/cs/epoch-vs-batch-vs-mini-batch. [Online; accessed 15-03-2023].

[49] The AI Blog. What is patch size in deep learning? https://aiblog.co.za/ai-faq/what-is-patch-size-in-deep-learning#The_Role_of_Patch_Sizes_in_Deep_Learning. [Online; accessed 12-04-2023].