Masters Program in **Geospatial Technologies**

Deep Learning for UAV Imagery Segmentation: The Detection of Elymus Athericus Spread in the Hallig Nordstrandischmoor

Mohamed Elmokadem

Dissertation submitted in partial fulfilment of the requirements
for the Degree of *Master of Science in Geospatial Technologies*

NOVA IMS Information Management School
Universitat Jaume I
Westfälische Wilhelms-Universität Münster

Deep Learning for UAV Imagery Segmentation:

The Detection of Elymus Athericus Spread in The Hallig Nordstrandischmoor

Supervised by:

Dr. Christian Knoth, University of Münster

Prof. Dr. Hanna Meyer, University of Münster

Prof. Dr. Jorge Mateu, Universitat Jaume I, Castellon, Spain

- March 2022 -

DECLARATION OF ORIGINALITY

I declare that the work described in this document is my own and not from someone else. All the assistance I have received from other people is duly acknowledged and all the sources (published or not published) are referenced.

This work has not been previously evaluated or submitted to NOVA Information Management School or elsewhere.

Germany, 25.03.2022

Mohamed Elmokadem

ACKNOWLEDGEMENTS

My sincere gratitude goes out to everyone who has provided me with assistance and support during the writing of this thesis. Obviously, I would like to extend a very special thank you to my supervisors Prof. Dr. Hanna Meyer, and Prof. Dr. Jorge Mateu, and especially for my main supervisor DR. Christian Knoth for his time and his excellent guidance throughout this process. Also, I would like to thank Dr. Jan Lehmann for all the support and guidance he provided throughout the thesis, as well as assistance in using the ILÖK - Institute of Landscape Ecology workstation. Lastly, I would like to thank Prof. Marco Painho for his support throughout the master's program.

Deep Learning for UAV Imagery Segmentation: The Detection of Elymus Athericus Spread in The Hallig Nordstrandischmoor

Abstract

It has been observed that Athyrics Elymus engulfs smaller species in low marsh habitats in different areas of Europe. To obtain automated segmentation of the Athurics Elymus in Hallig Nordstrandischmoor, a deep learning model based on the transfer learning technique presented in the VGG16 was implemented with a U-Net deep learning architecture and the data augmentation algorithm. In conclusion, the final results were good, and had been grouped into three levels of accuracy. These groups are also characterized by varying levels of diversity in their environments. The wildrye covers the majority of these single images, with a low percentage of other habitats in the first group with an accuracy greater than 96%. In the second group, the accuracy ranged between 91-95%, which included more elements of variation; however, wildrye was still included. Moreover, the third group, with an accuracy rate between 80-90%, did not seem to include wildrye very often, while at the same time including elements not found in the training dataset, such as surface water and dirt roads. Due to this, time and effort are significantly reduced while high accuracy is achieved.

Table of Contents

List of Figures

CHAPTER 1: INTRODUCTION AND RESEARCH STRUCTURE

1- Introduction and research structure

This chapter aims to present a general overview of this research by identifying the research background, research problem, research aims, research questions, and research structure.

1-1 Research background

Saltmarsh ecosystems are highly productive and act as an ecotone between terrestrial and marine environments, thus being highly valued as protected areas and functioning ecosystems (Pétillon et al., 2005). It has become increasingly apparent in Europe that native clonal grass Elymus athericus is replacing salt marsh plant communities (Valéry et al., 2004). Natural sedimentation facilitates the establishment of this species, which is characteristic of a high salt marsh. Additionally, this species has been observed to invade a low marsh habitat in central Europe by engulfing smaller species (Oldeland et al., 2021).

An invasion of this type can affect biological diversity, ecosystem properties, and functions, as well as the level of the conservation value of infested areas (Pétillon et al., 2005). Accordingly, it is imperative that plant species that negatively affect the environment be monitored and managed.

Since 1970, biotype mapping has been carried out in various locations in Germany to develop a landscape plan (Oldeland et al., 2021). Additionally, at the same time, a similar survey was initiated in the UK, under the name of the "Phase 1 Habitat Survey". This survey was conducted in all UK cities. The mapping of biotope types, which is now considered an important subject for every country, is becoming increasingly important, not only for urban planning but also for ecosystem assessments (Johnson, 2011; Qiu et al., 2010).

Over the last decade, machine learning algorithms have been widely employed for remote sensing, and these have ranged from basic algorithms to more sophisticated ones, including SVM, decision trees, Random Forests, and artificial neural networks (Helin et al., 2022). In the field of artificial intelligence, machine learning contributes to the ability of software applications to make more accurate predictions without explicitly programming them to do so. Moreover, machine learning algorithms can incorporate historical data as inputs when predicting new outputs (Khanzode & D. Sarode, 2020).

The application of machine learning methods (ML) to the interpretation of remote sensing data has enabled the use of its benefits in earth observation applications. The topic "Elymus

athericus mapping" provides a fantastic example of how Machine Learning can be utilized. Furthermore, listing the advantages of employing machine learning in remote sensing illustrates how this may be a useful (Bruzzone & Persello, 2010; Maxwell et al., 2018).

In this context, Machine Learning has several advantages. Firstly, it can handle large volumes of data in an efficient, precise, and robust way, as data consider the main feed for any Machine Learning model (Maxwell et al., 2018). Secondly, it can handle all types of data starting from tabular data to multidimensional ones. Thirdly, machine learning considers an effective tool for enhancing the decision-making process. For example, in producing detection maps, machine-learning preprocessing techniques and how the data is handled leads to improved precision and accuracy in the mapping, which in turn leads to taking many right development decisions (Khanzode & D. Sarode, 2020).

Finally, one of the greatest advantages of machine learning is its ability to automate any process. As a result, the algorithm handles most of the work, saving a considerable amount of time and effort. Nowadays, almost everything is automated. Consequently, allowing us to be more creative (Bruzzone & Persello, 2010). Additionally, the reliability of machine learning is determined by several factors. A key factor is that by using machine learning, large data sets can be processed and analyzed, which is not possible to do with standard systems (Khanzode & D. Sarode, 2020).

1-2 Research problem

The first research problem focused on the detrimental effects of Elymus's invasion on the ecosystems of different environments. Elymus has the key effect of causing a mass loss of species, which affects the biodiversity of the ecosystem. The second research challenge emerges from the case study of Hallig Nordstrandischmoor in Germany, where Elymus spread in the sea marshes and impacted the ecosystems. Thirdly, this research attempted to identify a novel method of detecting Elymus using deep learning and UAV imagery.

1-3 Research aims/objectives

The purpose of this study is to develop a Deep Learning model for segmenting Elymus spreads and to evaluate the model's performance and the results.

To accomplish these goals, two objectives must be met. As a first objective, this study will identify the main challenges associated with developing a Deep Learning model for the automated segmentation of Elymus spread in UAV imagery and determine whether these challenges can be

overcome by using techniques such as data augmentation and transfer learning. Second, we wish to explain the differences in accuracy between different subgroups within the study area. Lastly, a comparison is needed to identify the reasons for these differences.

1-4 Research questions

In light of the purpose and objectives of the study, the following three research questions have been formulated. Initially, what are the main challenges associated with the development of a Deep Learning model for the automated segmentation of Elymus spread in UAV imagery? The second question is: can these main challenges be overcome by using techniques such as data augmentation and transfer learning? As a final question, will the Deep Learning model develop improve the accuracy of the results when compared with the object-based image analysis (OBIA) model?

1-5 Research structure

This dissertation consisted of five chapters. The first chapter show the introduction and research workflow, where the research background, problem, aims, and questions were presented. The second chapter is the literature review which introduced two main points; deep learning uses as a detection tool in different science fields and deep learning & image segmentation. The third chapter discussed the materials and methods which consisted of three main sub-topics; study area, data collection, and research methods. The fourth chapter presented the results and discussions. Finally, the conclusion and main recommendations are presented in chapter five.

CHAPTER 2: LITERATURE REVIEW

2- Literature review

This chapter focus on presenting the available literature review that generally discussed the use of machine learning and deep learning in the dedication field. Unlike the traditional and standard methods of introducing and presenting literature reviews and due to the limited scholarly resources, using machine learning models in spatial and geographical detection of natural phenomena that harm the environmental ecosystems, this chapter presented the results of the collected literature review rather than the literature review itself, which surely will be presented in the context of presenting these results.

Because of the different keywords covered by the proposed topic, the literature review has been divided into two phases. Each phase covers some keywords related to one subject. Many databases have been used to conduct a literature review for the proposed topic (e.g., Scopus, IEEE Xplore, and Google Scholar).

The first phase of the literature review focuses on highlighting the recent and most important uses of machine learning and deep learning in the detection of different elements in various science fields directions. The second phase discusses the major deep learning concepts that are relevant to remote sensing. Numerous publications have been reviewed and analyzed in this section, most of which have been published within the last four years.

2-1 Deep learning uses as a detection tool in different science fields

Since the appearance of machine learning/deep learning and its algorithms, it has been used in many fields to detect various elements, which has helped science to find solutions to many problems and challenges in different scientific fields. For example, in the medical field, different deep learning models were used in identifying and detecting many diseases. Shen et al. in their 2019 article explained how deep learning algorithm helped in improving the detection of Breast Cancer through transferring the classifier to whole images after it was small patches, as well as building a modern CNN using a combination of convolutional layers and fully connected (FC) ones (Shen et al., 2019). Additionally, Scebba et al. confirmed the importance of the deep learning approach in automating the wounds images segmentation (Scebba et al., 2022).

In the spatial analysis field, machine learning and deep learning are used in different forms. For example, Abozeid et al. used a deep learning model to detect the Olive trees allocation using satellite imagery, as these trees represent significant economic values in some geographical areas such as KSA (Abozeid et al., 2022). Another example of using a deep learning algorithm to

detect spatial phenomena was using it in the detection of forest fire. Rahul et al. highlighted the importance of deep learning techniques in the decline of forest fire as one of the most serious threats to our ecosystems and environment (Rahul et al., 2020). They depend on Convolutional Neural Networks (CNN) to early detect these forest fires. Additionally, some researchers used machine and deep learning approaches to detect special elements in agricultural lands. For instance, Wilfried Wöber et al. used machine learning in plant science by investigating the explanatory factors for plant classification using Convolutional neuronal networks (CNNs) (Wöber et al., 2021). There are many recent uses of the machine and deep learning approaches in spatial analysis science such as using them in analyzing bicycle level of service (X. Liang et al., 2021), ship detection (Song et al., 2020), traffic congestion detection (Kamble & Kounte, 2020), urban growth detection (Gómez et al., 2019), flood detection (Hashi et al., 2021), etc.

The dissertation focused in add more scholarly outputs in the use of machine learning and deep learning in the detection of different spatial phenomena, especially the environmental ones such as Saltmarsh and Elymus athericus ecosystems.

2-2 Deep learning and image segmentation

Deep learning algorithms have become increasingly popular over the past few years for analyzing remote sensing images. This section of the literature review introduces several commonly used DL models in remote sensing, including the supervised CNN, recurrent neural network (RNN) models, deep belief network (DBN) models, as well as the recently popular generative adversarial network (GAN) (Helin et al., 2022). An exhaustive review has been conducted to describe/discuss how DL has been applied to remote sensing image analysis tasks, for example, image classification, object detection, segmentation, and object-based image analysis (OBIA). From preprocessing to mapping, the review discusses a variety of applications and technologies related to remote sensing.

At the beginning of the 1990s, backpropagation-based training of deep neural networks with multiple layers became a formal research topic, however, it was largely ignored by the machine-learning community (Ma et al., 2019). For quite some time, deep neural networks (DNNs) have not received much attention from the computer vision and remote sensing communities, due in part to difficulties associated with training deep neural networks (DNNs). DNNs have recently received wide attention, largely because they have outperformed competitive machine-learning algorithms in numerous application areas (LeCun et al., 2015). With the advent of unsupervised learning, deep learning became practical to some degree with

the help of deep feedforward networks. As one early breakthrough, the CNN-based AlexNet architecture won in 2012 by a wide margin the popular ImageNet contest. It achieved this success mainly due to its efficient use of graphics processing units (GPU), rectified linear units (ReLUs), and a large number of training examples. Following this contest, DL received much more attention in different subfields of computer vision, and it was found to be effective for many applications in the remote sensing (Ma et al., 2019).

2-2-1 Convolutional neural networks

As one of the most widely used distributed learning algorithms, CNN was designed to process data on multiple arrays (LeCun et al., 2015; Litjens et al., 2017). Due to this characteristic, it is particularly ideal for processing image data derived from remote sensing of multiband signals in which pixels are arranged regularly. To be precise, CNNs use three different types of hierarchical structures: convolution layers, pooling layers, and fully connected layers. Every layer is altered by adding biases γ = b1, · · ·, bK and K kernels W = W 1, W 2, · · ·, W K , resulting in a new feature map Xk.A nonlinear transform σ(·) is performed upon these features, and this process is repeated with each subsequent convolutional layer l: X kl = σW kl − 1 ∗ X l − 1 + bkl − 1 (Litjens et al., 2017). In contrast to traditional multi-layer perceptrons (MLPs), CNNs aggregate the values of pixels within a neighbourhood of a certain size using an invariant permutation function, usually the max or average operation. The network usually ends with fully connected layers (i.e., regular neural-network layers) at the end of the convolutional stream, but weights are no longer shared (Helin et al., 2022).

2-2-1-1 Convolutional layers

In order to understand how CNNs work, it is essential to consider how the convolutional layer operates. Convolutional layer parameters revolve around utilizing learnable kernels. A kernel of this type has relatively small dimensions, but it covers the entire depth of the input. Convolutional layers combine each filter across each spatial dimensionality of the input to generate a 2-dimensional activation map (LeCun et al., 2015; Litjens et al., 2017). Figure [1] illustrates this. Scalar products are calculated for each value in the input as we progress through it. Using this data, the network will learn to generate kernels that “fire” when they view a specific feature present at a particular spatial position. Such kernels are commonly known as activations (O’Shea & Nash, 2015).

For each kernel, an activation map will be generated, which is stacked along the depth dimension to create the total output volume from the convolutional layer. The problem with

training ANNs on inputs such as images is that the resulting models are often too large to be effectively trained. In order to mitigate this issue, each ANNs neuron of a convolutional layer is only connected to a small portion of the input space in order to mitigate this issue. Generally, this region's size is referred to as the neuron's receptive field (Litjens et al., 2017; O'Shea & Nash, 2015).

In plenty of cases, the connectivity is determined by the depth, which is almost always the same as the input depth. Assuming that the network has a size of 64 * 64 * 3 (that is, an RGB image with 64 * 64 dimensions) and that the receptive fields are 6 * 6, we will have 108 weights on each neuron in the convolution layer. As a point of comparison, a standard neuron found in other forms of ANNs would contain 12, 288 weights per unit. The use of convolutional layers reduces the complexity of the model by optimizing the model's output. This is performed by optimizing three parameters: depth, stride, and zero padding (Litjens et al., 2017; O'Shea & Nash, 2015).

It is possible to specify the depth of the output volume produced by the convolutional layers manually by selecting the number of neurons in each layer corresponding to the same region of the input. Other types of ANNs resemble this in the sense that all neurons in the hidden layer are connected to every single neuron beforehand. Reducing this parameter can significantly reduce the number of neurons in the network, but it can also reduce the model's ability to recognize patterns (H. Liang & Li, 2016; Ma et al., 2019).

In order to place the receptive field, it is also possible to define the stride in which we set the depth surrounding the spatial dimensionality of the input. As an example, we would experience extremely large activations if astride were set to 1. While a higher stride setting will reduce the amount of overlap and result in a lower spatial dimension (Navab et al., 2015).

The method of zero-padding involves padding the border of the input and is an effective method of controlling the output volume dimensions. Using these techniques, enable us to change the spatial dimension of the output of convolutional layers. In order to do so, the following formula could be used:

$$\frac{(V\ R) + 2Z\ S}{+\ 1}$$

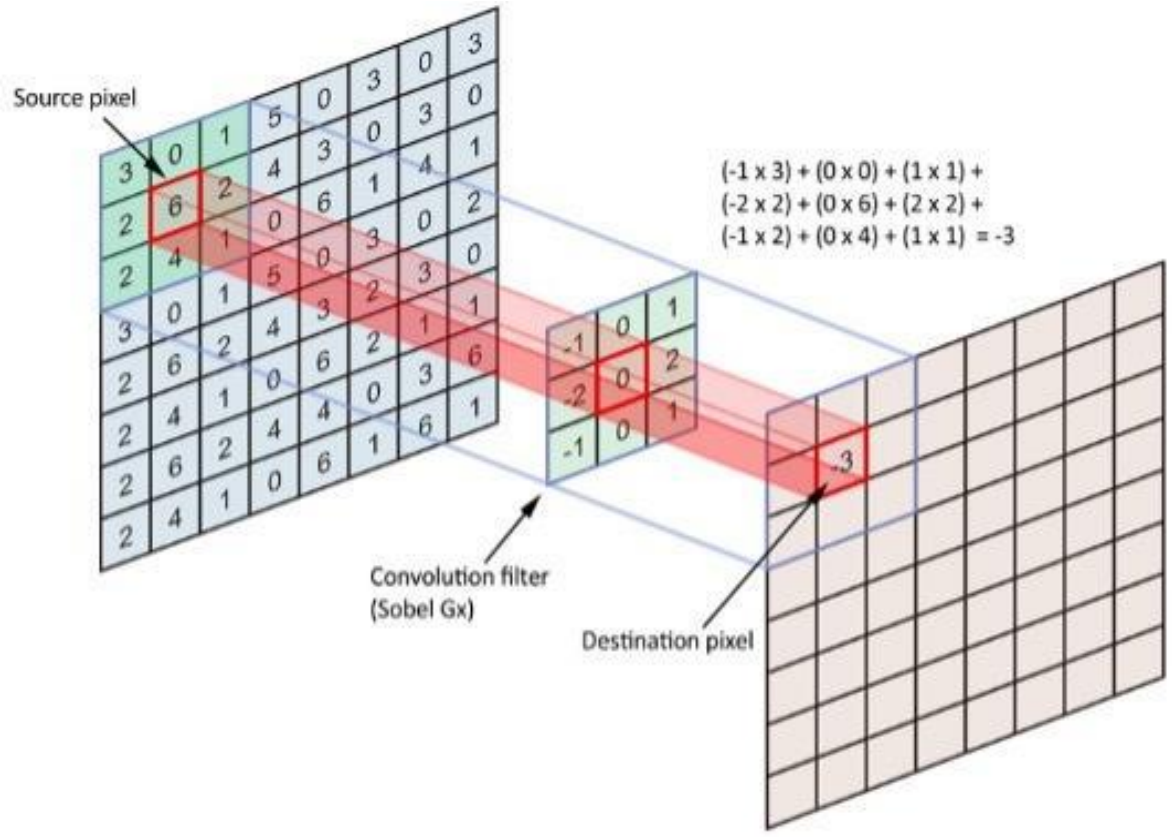
There are four components in this equation: "V" is the input volume size (height*width*depth), "R" is the receptive field size, "Z" is the amount of padding set to zero, and "S" corresponds to the stride. In the event that the calculated result was not a whole integer,

the stride would not have been set correctly, as the neurons would not be able to span the given input (Ma et al., 2019; Navab et al., 2015).

The deep learning models would still be very large if we used an image input with any real dimensionality, even if, we have attempted our utmost to make our models more compact. Although, methods have been developed that greatly reduce the overall number of parameters within the convolutional layer (H. Liang & Li, 2016).

In practice, parameter sharing assumes that if a region feature is useful for computing in one spatial region, then it will likely be useful in another region. The number of parameters produced by the convolutional layer will be reduced by a great deal if each activation map within the output volume is constrained to have the same weights and bias (Pitaloka et al., 2017). In this regard, as the backpropagation stage occurs, each neuron within the output will represent the overall gradient, which is totalized across the depth - thus we only must update one set of weights and not every individual one (H. Liang & Li, 2016).

Figure (1): Convolutional layers way of work



Source: (Arden Dertat, 2017)

2-2-1-2 Pooling layers

The purpose of layering is to reduce the number of dimensions in the representation, which further reduces the computational complexity of the model (Vives-Boix & Ruiz-Fernández, 2021). The pooling layer is applied over each activation map in the input, allowing it to scale in dimensions using the "MAX" function. Generally, CNNs employ max-pooling layers that apply kernels of dimensionality 2*2 and a stride of 2 along the input dimensions. By doing so, we reduce the activation map to 25% of the original size, while maintaining the depth volume at its normal size (Ma et al., 2019).

Max-pooling is commonly observed in two general methods due to the destructive nature of the pooling layer. Pooling layers typically have a stride and filters that are set to 2*2, which will allow the layer to extend across the entire spatial dimension of the input. A third option would be to utilize overlapping pooling, where the stride is set to 2 while the kernel size is set to 3 (Litjens et al., 2017; Ma et al., 2019). As a result of the destructive nature of pooling, a kernel size greater than 3 will usually significantly decrease the model's performance (H. Liang & Li, 2016).

A CNN architecture may contain general-pooling as well as max-pooling. Normally, general pooling layers are made up of neurons that can perform a number of common operations such as average pooling, and normalization (Litjens et al., 2017; Navab et al., 2015).

2-2-1-3 Fully connected layers

There are neurons located in the fully connected layer which are directly connected to those in the two adjacent layers, without being linked to any neurons within those layers. The way neurons are arranged in traditional ANNs is analogous to the way they are arranged in traditional ANNs (Montavon et al., 2012).

2-2-2 Residual network (ResNet)

As found in many literature articles, the use of VGG or other deep convolutional blocks causes two main problems (vanishing gradient problem, and over-fitting) vanishing gradient problem. "Meaning your weights are updated incrementally" and if that increment goes very small okay then your weights are not updated at all as the change in the weight is very insignificant. So why will it go very small, this is backpropagation as it is going from the layer to the one before then one before. Stacking Convolutional layers causes overfitting and effect the generalization ability of the network. which maybe will have when we apply our model to other

subsets in the study area. to overcome this problem many scholars used ResNet as the backbone of the U-Net.

The residual network will just make a skip connection from the first layer to the last layer so not the whole weights in the first layer go in the Res network block but just part of it and the residual added at the end of the block to the last layer weight.

Because when working with a very deep convolutional network for example VGG is a great architecture but if you keep increasing the number of layers, you will run to something called a vanishing gradient problem. “Meaning your weights are updated incrementally” and if that increment goes very small okay then your weights are not updated at all as the change in the weight is very insignificant.

So why will it go very small, this is backpropagation as it goes from the layer to the one before then one before. Stacking Convolutional layers causes overfitting and effect the generalization ability of the network.

2-2-3 The U-net architecture

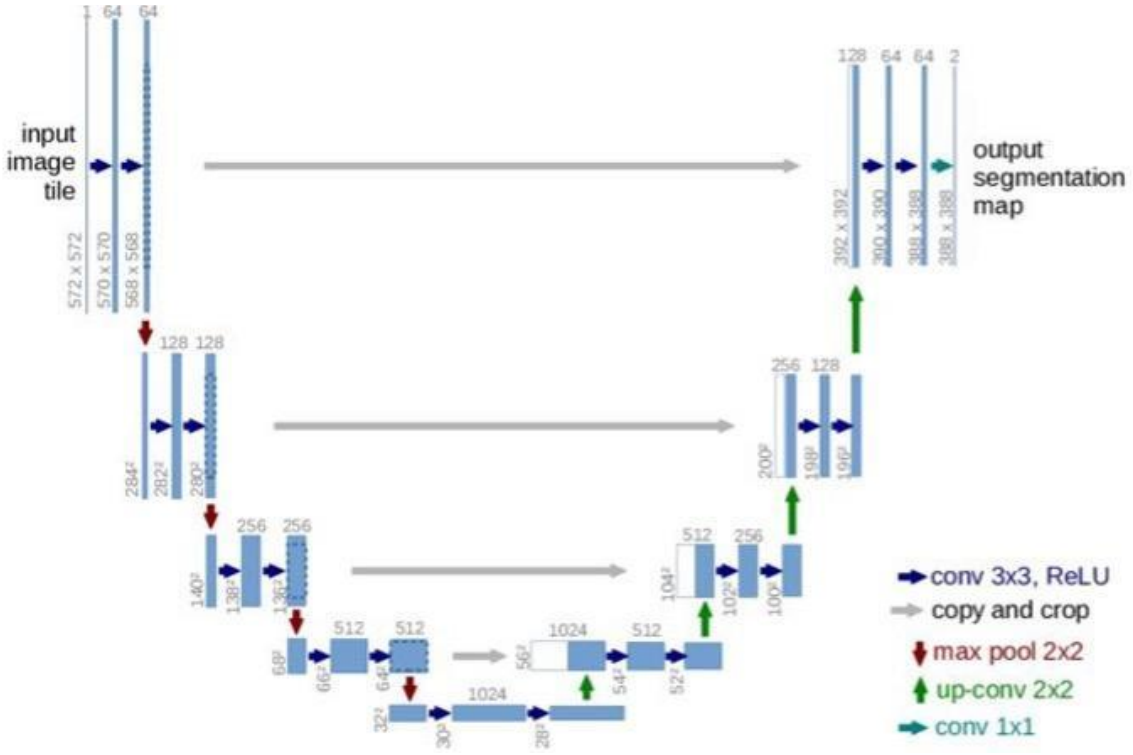
U-net has established itself as a gold standard for comparing the quality and accuracy of semantic segmentation. U-net is a convolutional autoencoder consisting of additional interconnections between the encoder and the decoder parts. Such neural networks are referred to as hourglass architectures. In many spatial applications, U-net or neural networks utilizing the same approach produce highly accurate results (Ronneberger et al., 2015).

U-net utilizes convolutional layers generally for semantic segmentation. In comparison to the normal CNN U-net, it builds a very large receptive field and is considerably less expensive. In an asymmetric network, spatial features are extracted from the image by the encoder and used by the decoder to construct a segmentation map. The typical pattern of an encoder is similar to that of a convolutional neural network (Ronneberger et al., 2015).

Following two convolutions of size 3*3, a pooling operation is performed with a pooling size of 2*2 and a stride of 2. Similarly, the process is repeated four times, and the number of filters in each convolutional layer is doubled after each downsampling. In order to connect the encoder to the decoder, two 3*3 convolutions are performed. By comparison, the Decoder first increases the sample size of the feature map through a 2*2 transposed convolution operation, which reduces the channel size by half. Following this, two 3x3 convolutions are performed (Zeiler et al., 2010).

Similar to the Encoder, this sequence of one upsampling operation and two convolution operations is repeated four times, thereby halving the number of filters at each stage. It is necessary to perform a 1*1 convolution operation in order to generate the final segmentation map. All convolutional layers except the final one utilize the ReLU (Recognized Linear Unit) activation function; the final layer uses the Sigmoid activation function (LeCun et al., 2015).

Figure (2): U-net architecture



Source: (Ronneberger et al., 2015)

2-2-4 Transfer Learning

For the optimal performance of deep convolutional neural networks, large amounts of labelled data are typically required. Transfer learning relates to the idea of applying the knowledge gained from a task with many training samples to a task with relatively few, which increases the probability of gradient propagation during training, and reduces the limitations of the data on performance. As labelled data is extremely expensive, it is important to optimize the use of existing datasets.

The model does not need to learn from scratch as with traditional networks because some low-level features, such as edges and shapes, are useful and can be shared by transferring parameters. It is therefore the optimization objective of the target task to extract abstract and

sophisticated high-level features. Transfer learning is typically accomplished by tuning a pretrained network model on a target dataset. Grigorick et al. pre-trained a convolutional neural network on ImageNet before ne-tuning all network parameters for a task (detection) where enough data is unavailable. In their study, Long et al. fine-tuned only the parameters of the last few layers and concluded that specific features from these layers tailored to a particular task cannot provide a successful solution to the domain discrepancy. To solve different classification tasks, Sharif Razavian et al. used a pre-trained model to extract features with a support vector machine (SVM) (Razavian et al., 2014).

2-3 Deep learning mechanism

In machine learning and deep learning algorithms, data preprocessing plays a critical role, and proper data preprocessing is essential for improving the detection performance (Vives-Boix & Ruiz-Fernández, 2021). According to (Maxwell et al., 2018), data preprocessing includes data cleansing, normalization, transformation, feature extraction, and selection. There are some well-known algorithms that are presented in their study for each step of data preprocessing. In order to perform detection tasks utilizing data obtained from UAV images, a segmentation operation is essential, since raw data vary greatly over time (Maxwell et al., 2018; Pitaloka et al., 2017).

2-3-1 Data augmentation

Having too few samples to learn from results in overfitting, as you cannot train a model that can generalize to new data. In the case of infinite data, your model would be exposed to every possible aspect of the data distribution: you would never overfit the model. It is possible to generate more training data from existing training samples by randomly transforming the samples in order to generate believable-looking images by augmentation. We want to ensure that your model will never see the same picture twice during training. As a result, the model is exposed to more data aspects and can generalize more effectively.

2-4 Deep learning conceptual kernel

In the hope of identifying potential areas for improvement, [Ibtehaz and Rahman, 2020] analyzed the U-Net architecture carefully. They observed some discrepancies between the features propagating through the encoder network and the features propagating through the decoder network. Their solution involves adding additional processing to make the two feature maps more homogeneous, to reconcile these two sets of incompatible features. As a result, they have proposed MultiRes blocks to enhance U-Net by enabling multi-resolution analysis. Based on

the Inception building blocks, they developed a compact analogous structure, which is relatively lightweight and requires less memory. The MultiResUNet architecture incorporates these modifications.

Based on their research, U-Net can perform segmentation with remarkable accuracy when dealing with perfect or near-perfect images. In these instances, their proposed architecture performs only slightly better than U-Net. Nevertheless, for intricate images suffering from noise, perturbations, unclear boundary lines, etc., MultiResUNet increases performance significantly. MultiResUNet has resulted in a relative improvement of 10.15%, 5.07%, 2.63%, 1.41%, and 0.62% for the five datasets they used. As well as achieving a higher evaluation metric score, the segmentations produced by MultiResUNet also correspond visually to the ground truth. Furthermore, on the very challenging images, U-Net tends to over segment, under-segment, make false predictions and even miss the objects completely. The experiments conducted by these researchers have shown that MultiResUNet is more reliable and robust. Despite its subtlety and sensitivity in detecting boundaries, MultiResUNet has been resilient in segmenting images with a lot of perturbations and has been able to reject outliers as well. Moreover, MultiResUNet is able to capture fine details even in segmenting the majority class, where the U-Net tends to over-segment. Furthermore, MultiResUNet's simple 3D adaptation has performed better than the 3D U-Net, which has not simply been a 3-dimensional adaptation of the U-Net but has been enhanced and improved. Even though the segmentations generated by MultiResUNet are not perfect, they outperform the classical U-Net by a moderate margin in most cases (Ibtehaz & Rahman, 2020).

CHAPTER 3: MATERIALS AND METHODS

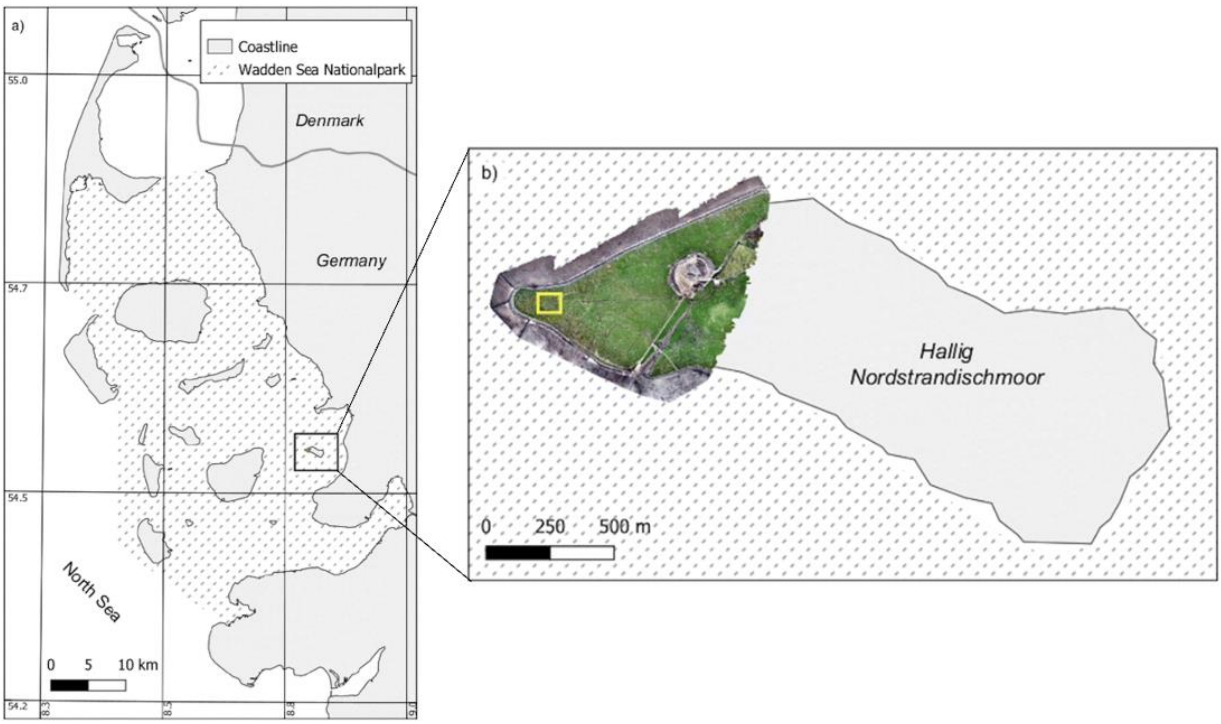
3- Material and methods

This chapter is divided into three main sub-sections, namely: study area, data collection, and research method. These sub-sections will be presented as follows:

3-1 Study area/Case study

Hallig Nordstrandischmoor is the subject of the study. In the Wadden Sea National Park, the Hallig is located at the northern tip of Germany's North Sea coast. In 2019, UAV imagery was collected on 18 ha of the western tip of the Hallig. Figure [3] presented the location of the research study area “Hallig Nordstrandischmoor”.

Figure (3): The study area of Hallig Nordstrandischmoor



Source: (Oldeland et al., 2021)

It is important to note that Hallig Nordstrandischmoor is one of the places under close monitoring for the growth of wildrye. This species has been observed as an invasive species. Mapping these types of biotopes is very important especially if it is such an invasive species in order to assess and monitor ecosystems. It is not feasible to collect the required parameters from the field in an extensive manner, so remote sensing (primarily high-resolution RGB and NIR data) is usually used.

3-2 Data collection

These data are usually analyzed manually by a trained analyst. As the objective of this project is to develop deep learning models for automated segmentation of wildrye, a high-resolution RGB image of the area was required.

This database is composed of high-resolution remote sensing images 10 cm, in RGB colour. From the research conducted by (Oldeland et al., 2021), a comprehensive dataset of manually digitized biotope types is available as a guide for creating masks for training and validation. Moreover, the author has created a complementary dataset of visually digitized wildrye or non-wildrye in order to ensure that the minimum number of points required to calculate accuracy is 30 per single image.

3-3 Research method

This section of the methodology is based on a tutorial written by (Christian Knoth, 2020). Six steps constitute the research methodology used in this thesis. Data preparation is the first step. The second step focuses on building the model using U-Net. The third step involves training the model. The fourth step involves predicting using the model. The fifth step transfers the model from subsets to single images. The last step is to develop the Confusion Matrix for each single image. These steps will be discussed in more detail in the following paragraphs.

3-3-1 Step I: Preparing the data

The deep learning model expects very small images as an example some models require images of 128x128 pixels or 448x448 pixels. Nevertheless, in remote sensing, it is common to make predictions based on larger images and expect maps rather than subsets of maps as the end result. In order to feed larger images into a CNN, we need to split the larger images into smaller subsets. (Christian Knoth, 2020)

3-3-1-1 Transfer from single images to subsets

In this step, I used a function to convert the RGB UAV imagery acquired into single images and then to convert the single images into subsets for use in training, validation, and prediction. The function used requires the input image (as a raster) and a target subset size parameters. Consequently, it crops the input image to a size that is exactly a multiple of the desired subset size, so that all subsets are of the same size. After this, the image is divided into subsets, and the subsets are written to disk.

The files are numbered and enumerated, so that they can be read in the correct order during prediction and can later be reassembled according to their order. Additionally, the function returns a cropped input raster, which serves as a reference when reassembling the prediction results. (Christian Knoth, 2020)

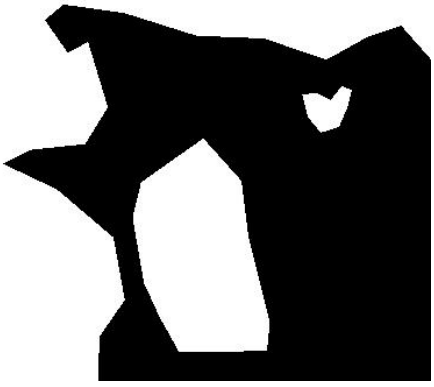
3-3-1-2 Generating Image Segmentation Masks

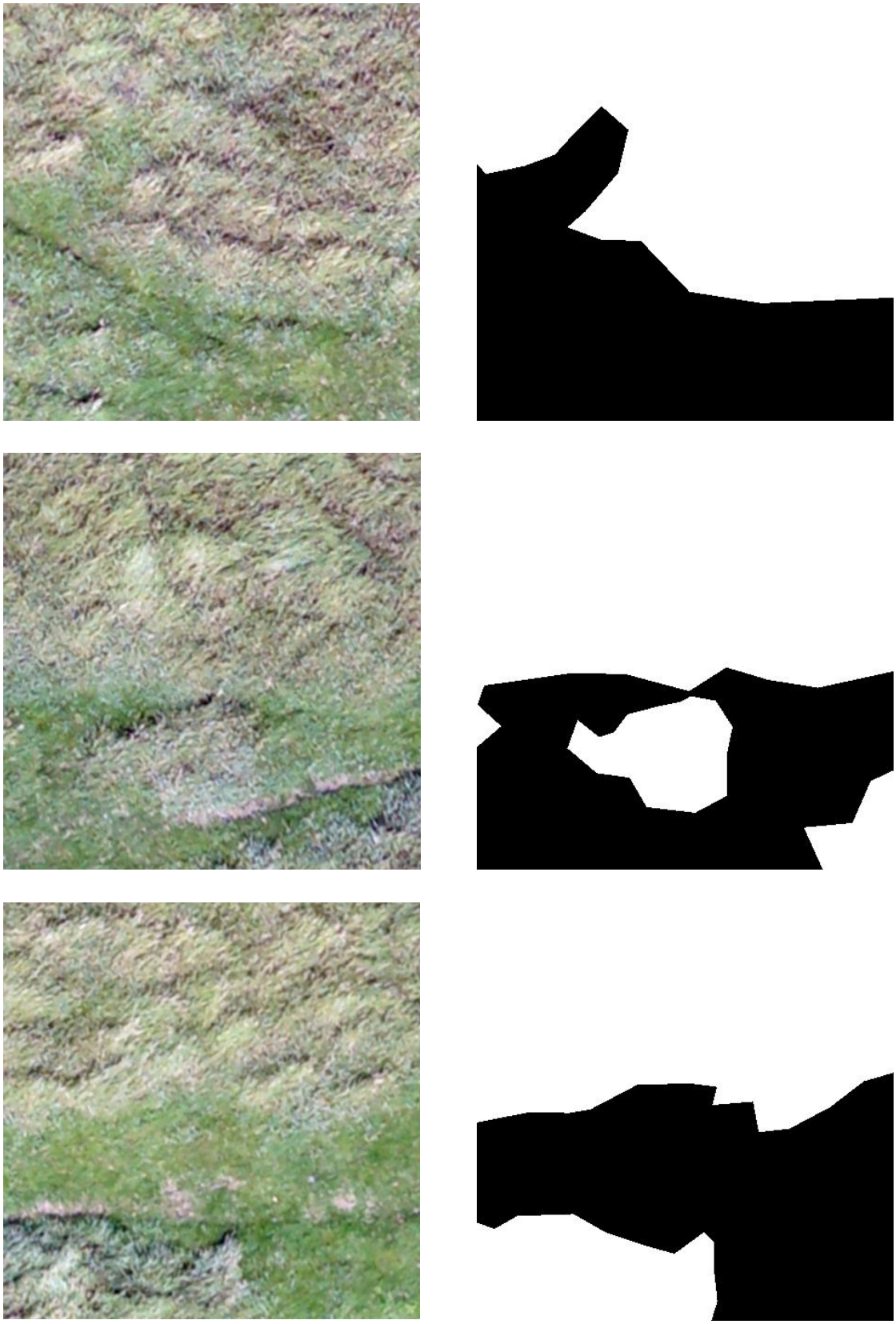
QGIS was used to create the training data, creating masks for the subsets (448x448 pixels). To start with, 28 subsets were to be used as a training sample. Figure (4) presented some samples of the training data used. Which represents 0.1 % of the total subsets. while 80 images were the number of subsets used to get the final results, which represented 0.3% of the total number of the subsets. The purpose of this increase was to represent other elements not included in the first training dataset and to enhance the diversity of the training dataset.

Figure (4): Samples of the used subsets and their segmentation masks

(a) Subsets samples

(b) Segmentation masks





Source: author, 2022

3-3-1-3: Data augmentation

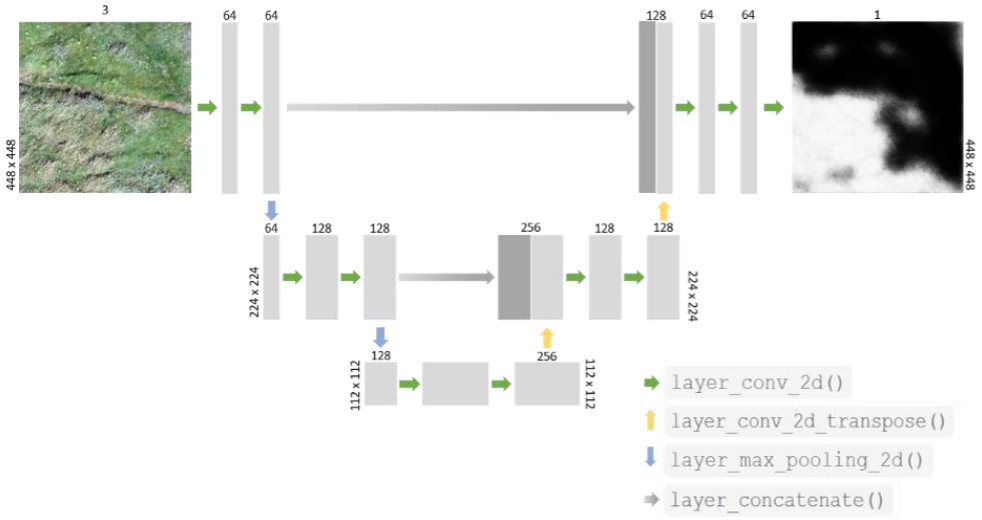
Utilizing data augmentation is one of the most common methods of increasing the amount of training data. In this case, image transformations and manipulations are applied to the training data to produce new synthetic images for learning. In my methodology, the TensorFlow module was utilized, which contains functions that can be used for data augmentation. Geometric manipulations had been performed such as flipping and rotating images in addition to spectral manipulations to simulate different lighting conditions. (Christian Knoth, 2020)

Three different augmentations are applied to the input data to prepare the training data in the function used: Flip the images (and masks) left to right, and up and down Combine left-right flip and up-down flip on images (and masks). The function also applies spectral augmentation to the images during each of these augmentations (not the masks) to increase the differences between the original images and simulate different lighting conditions.

3-3-2 Step II: Building the model using U-net

Essentially, the model consists of a series of layers sequentially transforming the input data into a number of representations informative of that data. In this chain, the prediction result comes from the layer at the end. To build such a model I used the model architecture of a U-net, adopted by (Christian Knoth, 2020), and proposed by (Ma et al., 2019; Navab et al., 2015; Ronneberger et al., 2015).

Figure (5): Building the model using U-net



Source: (Christian Knoth, 2020)

This network begins with a series of convolutional layers and a maximum pooling layer. The following sequence identifies features and spatial patterns that increase complexity (increasing the number of convolutions/maximum pooling blocks). It is also known as the contraction path of the network. A loss of accurate location information occurs when increasingly complex and abstract features are extracted at the expense of reduced resolution.

Adding one or two flat and dense layers after the last max-pooling layer is sufficient for CNN to perform image classification. Unlike a normal convolutional network, a U-net architecture adds another part (the second half of the letter "U"), also known as the extended path. In this extended path, the feature map is resampled to the same resolution as the input image. A series of upsampling and convolution layers are used to accomplish this. An important point to note here is that the feature map is not simply unsampled but combined with the output of the corresponding layer (in terms of resolution) in the shrinking path.

This contains the high-resolution features extracted there (gray arrows). This is achieved by concatenating the layers. Thus, the continuous convolutional layer incorporates not only the information about complex spatial patterns aggregated throughout the contraction path but also the high-resolution features extracted from the middle layer of the contraction path. As a result, you can find the target in the image subset with greater accuracy.

3-3-2-1 Using pre-trained networks in the U-net

Deep learning has the advantage of being portable. For the DL model, the first layers of a CNN are responsible for extracting generic features and patterns (edges of various forms and directions, patterns of pixel values). Despite being trained on a different dataset, these feature extractors may still prove useful for our purpose. Here, I used a CNN that was trained on the "ImageNet" dataset called "VGG16" (Simonyan & Zisserman, 2015).

There are animals and objects such as cats, bicycles, balloons, and so on in this set of data. It is important to note, however, that we can also extract features from our images using the first layers, which allows us to distinguish between targets and non-targets. To conduct the actual classification, we used the first 15 layers as a basis and then add "our own" dense layers. (Christian Knoth, 2020)

3-3-3 Step III: Training the model

As a first step, the model must be configured or prepared for training. The compile function is used to accomplish this. It is possible to specify some parameters for the training of

the model using the function that is used to train the model. There is the optimizer (including the learning rate lr), the loss function, and one or more accuracy metrics for measuring/visualizing the model's performance (metrics are not used for training).

On a randomly selected batch of training data, the neural network calculates the outputs and predictions. Using the defined loss function, the output is compared to the corresponding label and the loss (how far off our predictions are) is calculated. Using the optimizer and the learning rate lr, the parameters of the layers are then adjusted a little bit to reduce the loss on the current batch.

Upon completion of this step, the network will process the next batch of data with the new parameters. Depending on the fit method, this step is repeated as many times as necessary.

This process is known as (mini-batch) stochastic gradient descent (mini-batch SGD) and there are a number of variants. The optimizer decides which one to use. In this section, we will briefly look at the arguments that we specify when we call compile function:

- Loss function: Measure the differences between prediction and label using a loss function that specifies how the loss of the network will be computed. The binary cross-entropy is a commonly used loss function in binary classification problems and that is why I used it in my model.

- Optimizer: An optimizer is an algorithm used to adjust the parameters during each iteration of training in order to reduce the loss. An optimizer specifies how the parameters will be adjusted in order to reduce the loss. In the model, we used a rmsprop optimizer.

- Learning rate: The learning rate specifies the extent to which parameters are adapted during each iteration, and in other words how strongly the weights are being varied. When the learning rate is greater, training can be completed more quickly, but if the rate is too high, training can become unstable.

- Metric: Our model's accuracy on the validation data is indicated by this metric after each epoch. Using the accuracy method, we calculate the fraction of samples that are correctly classified into 0 or 1 (by applying a threshold of 0.5 for a classification of 0 or 1 which is used in our model).

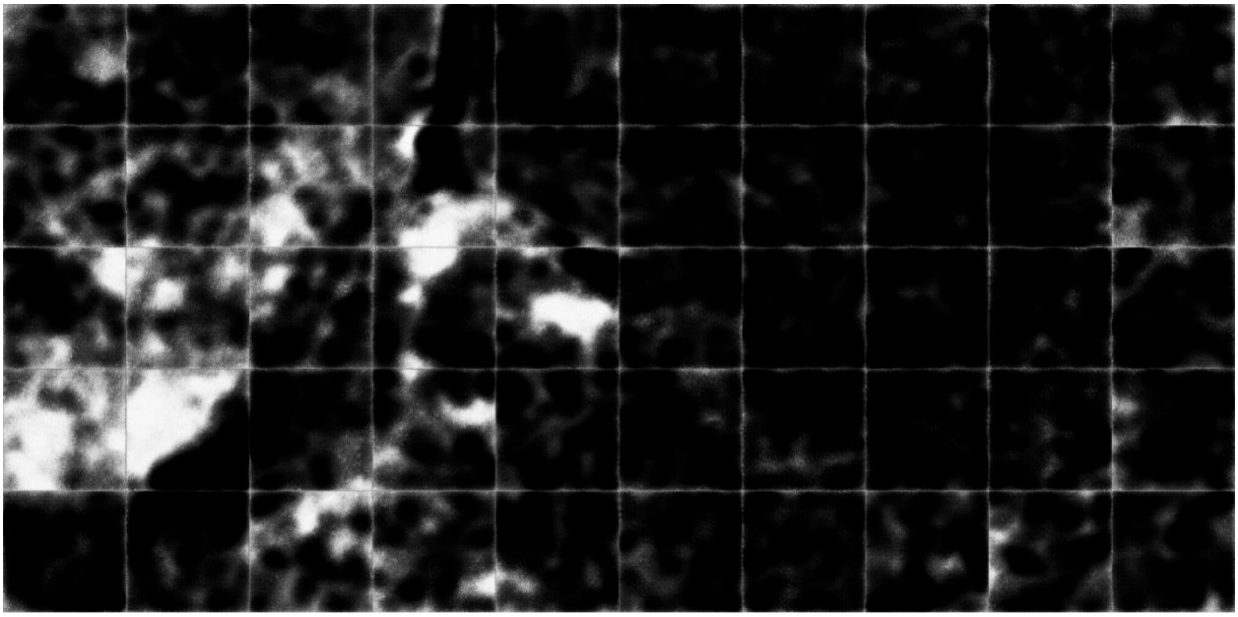
3-3-4 Step IV: Predicting using the model

After completing the training process, the trained model was ready to be used to predict new data. The predict function had been utilized for this purpose. This function requires as arguments the model to use for prediction and the dataset on which it is to be applied.

3-3-5 Step V: Transfer back from subsets to single images

This step consisted of transferring the prediction from multiple subsets to individual images using a function. As inputs to this function, prediction subsets, the desired output path, and the cropped reference raster that was a result of the transferring step are required. Below is a sample from the prediction subsets and the single images resulting from the function.

Figure (6): The result of transferring back the subsets to a single image



Source: author, 2022

3-3-6 Step VI: Confusion Matrix for the single images

Using (Oldeland et al., 2021) comprehensive dataset and the complementary dataset, a confusion matrix was applied on the model to every single image. while the final step was classifying the results into three groups to allow more analysis. It is worth noting that the spiral nature of the methodology allowed for a degree of accuracy to be attained by moving back and forth until the required level of accuracy was achieved.

CHAPTER 4: RESULTS AND DISCUSSION

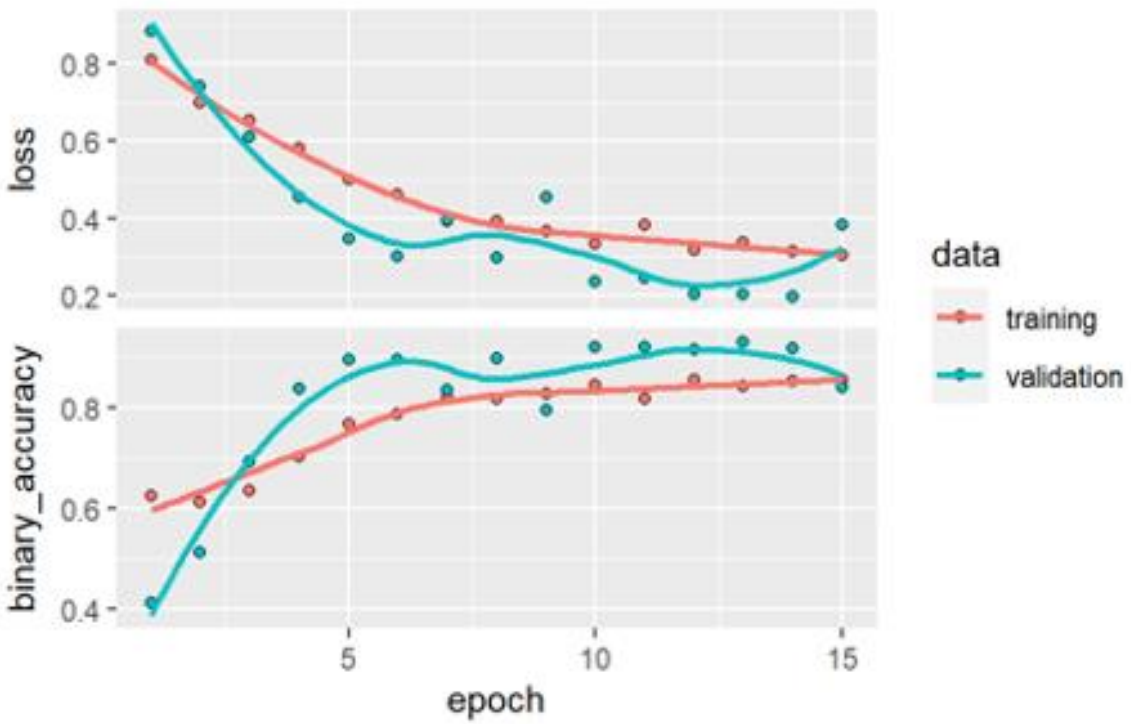
4- Results and discussion

This chapter will present the results of conducting the deep learning model for detection of Elymus spread in Hallig Nordstrandischmoor, Germany. The chapter is divided into three sections: the preliminary results will be presented at the beginning, followed by a summary of the final results. Finally, the discussion and interpretation of the results will be presented.

4-1 Preliminary results

As can be seen from the following plot, the preliminary results for the first trained model were good. This plot shows the training curves which were stored as a variable during compilation. However, it should be noted that the curve is not flattened and is prone to ups and downs.

Figure (7): The training curve for the first main model



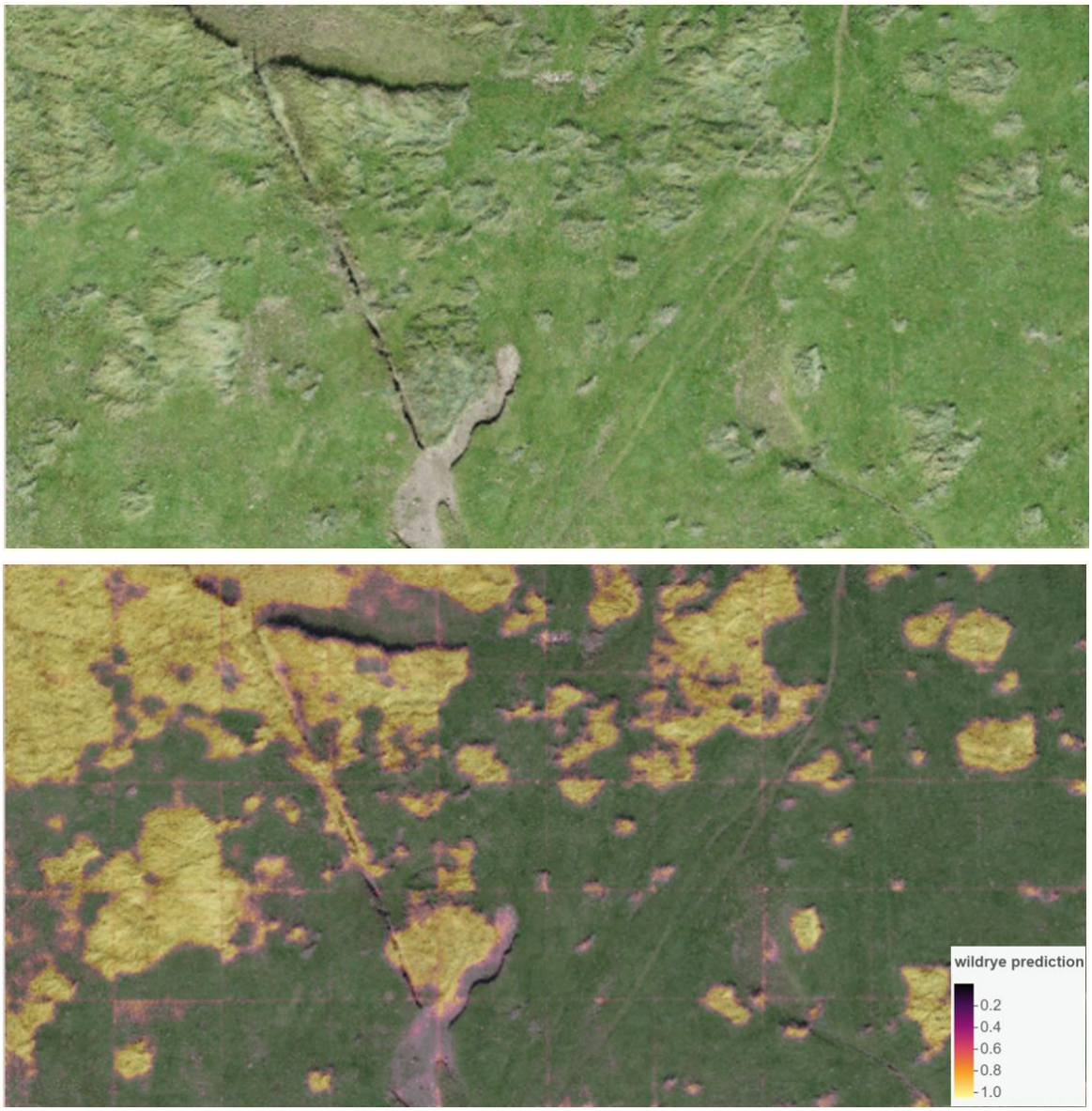
Source: author, 2022.

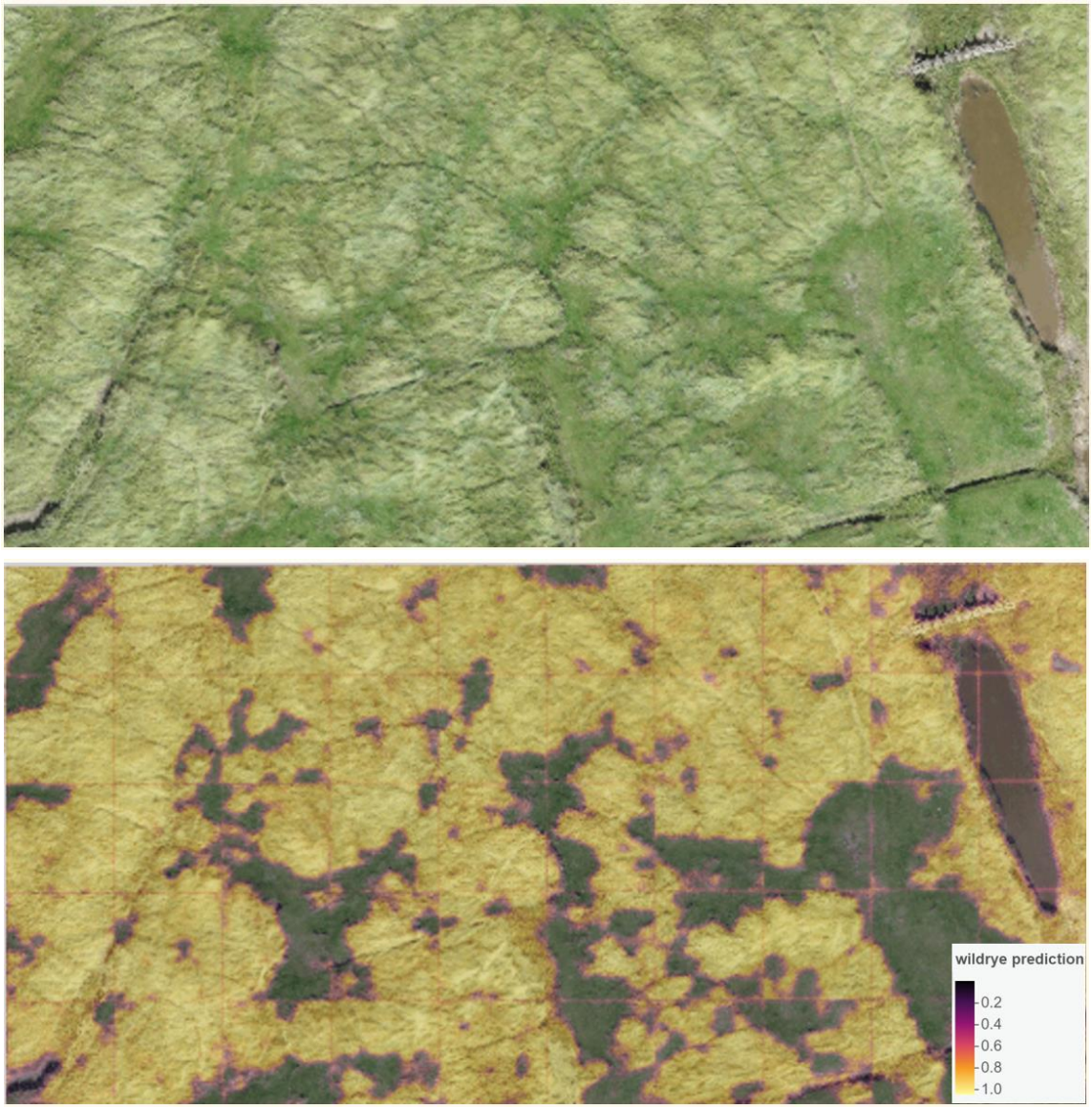
On the basis of that and in order to comprehend the various aspects of the results, a visual interpretation of the predictions combined with an analysis of its accuracy proved to be one of the best tools. The following are examples of the preliminary results maps organized into three

groups according to their accuracy. In each group are two or three single images as a representative sample of variations in the environment of the individual images.

The average accuracy of the first group was 86.5%. The first two single images (see Fig.8) demonstrate a sample from the first group while the accuracy ranged between 85-88%.

Figure (8): figures (UAV imagery followed with its segmented results) illustrating samples of the prediction of the wildrye from the first group with accuracy ranging between 85-88%

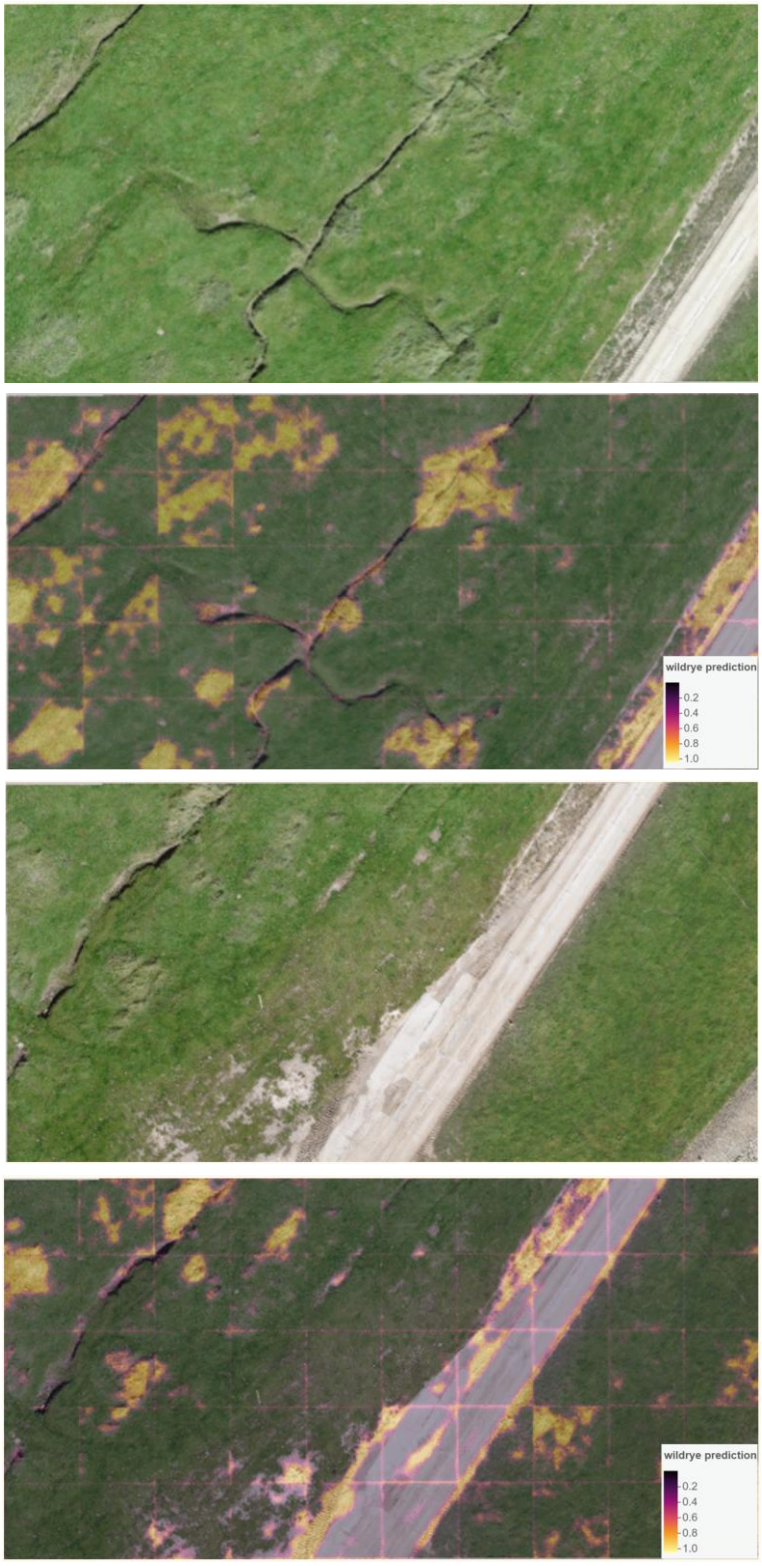




Source: author, 2022.

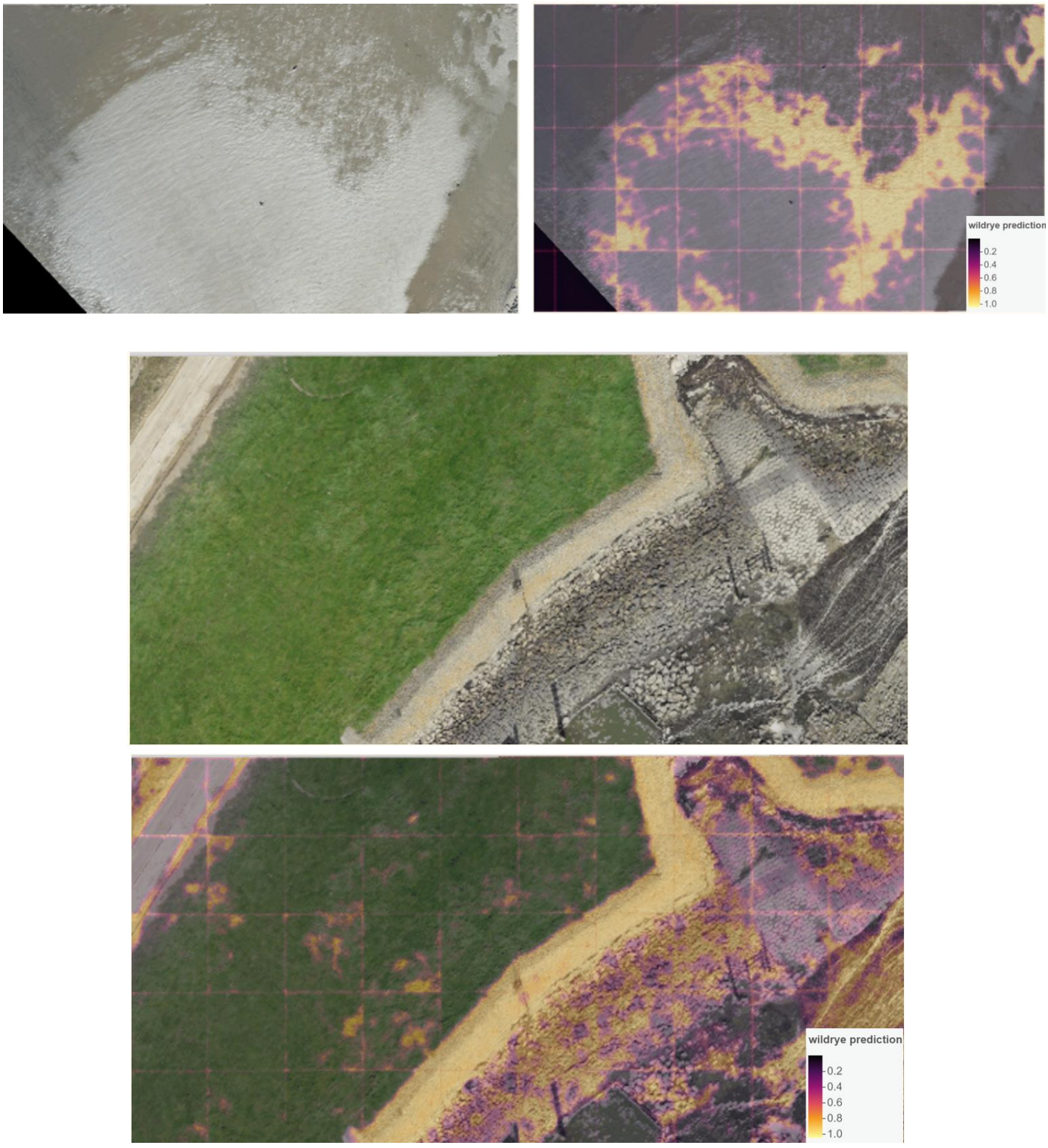
Furthermore, in the second group demonstrated in the following two single images (Fig.9) the accuracy decreased by 5% but the results are still satisfactory while the accuracy does not fall below 76%. Additionally, results from the third group of single images indicate a significant drop in accuracy, reaching up to 55%, while this can be interpreted visually in the last two single images (Fig.10).

Figure (9): figures (UAV imagery followed with its segmented results) illustrating samples of the prediction of the wildrye from the second group with accuracy ranging between 76-84%



Source: author, 2022.

Figure (10): figures (UAV imagery followed with its segmented results) illustrating samples of the prediction of the wildrye from the second group with accuracy ranging between 55-75%

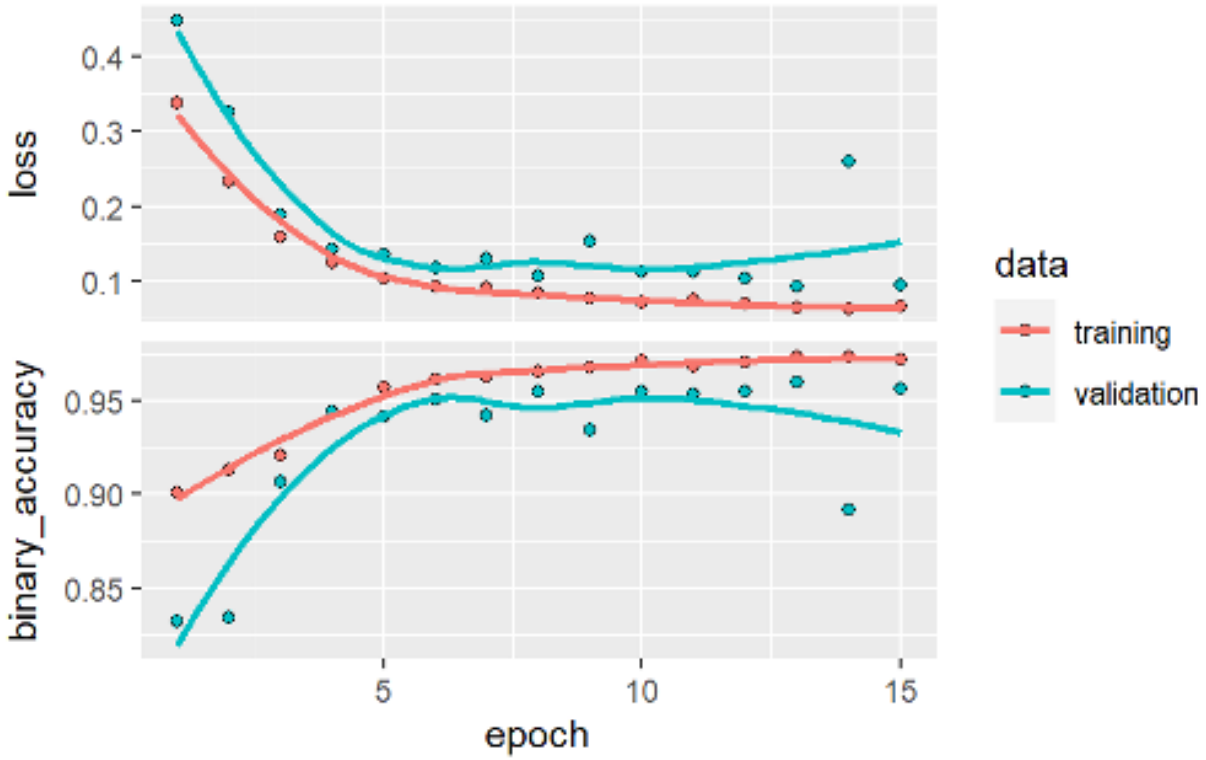


Source: author, 2022.

4-2 Final results

Here we illustrate the training curves that were stored as variables during the compilation of the final model. We can observe that the curve has become more flatten especially after the 5th epoch which means that the model is more stable. Furthermore, the accuracy percentage is 10% higher.

Figure (11): The training curve for the final model

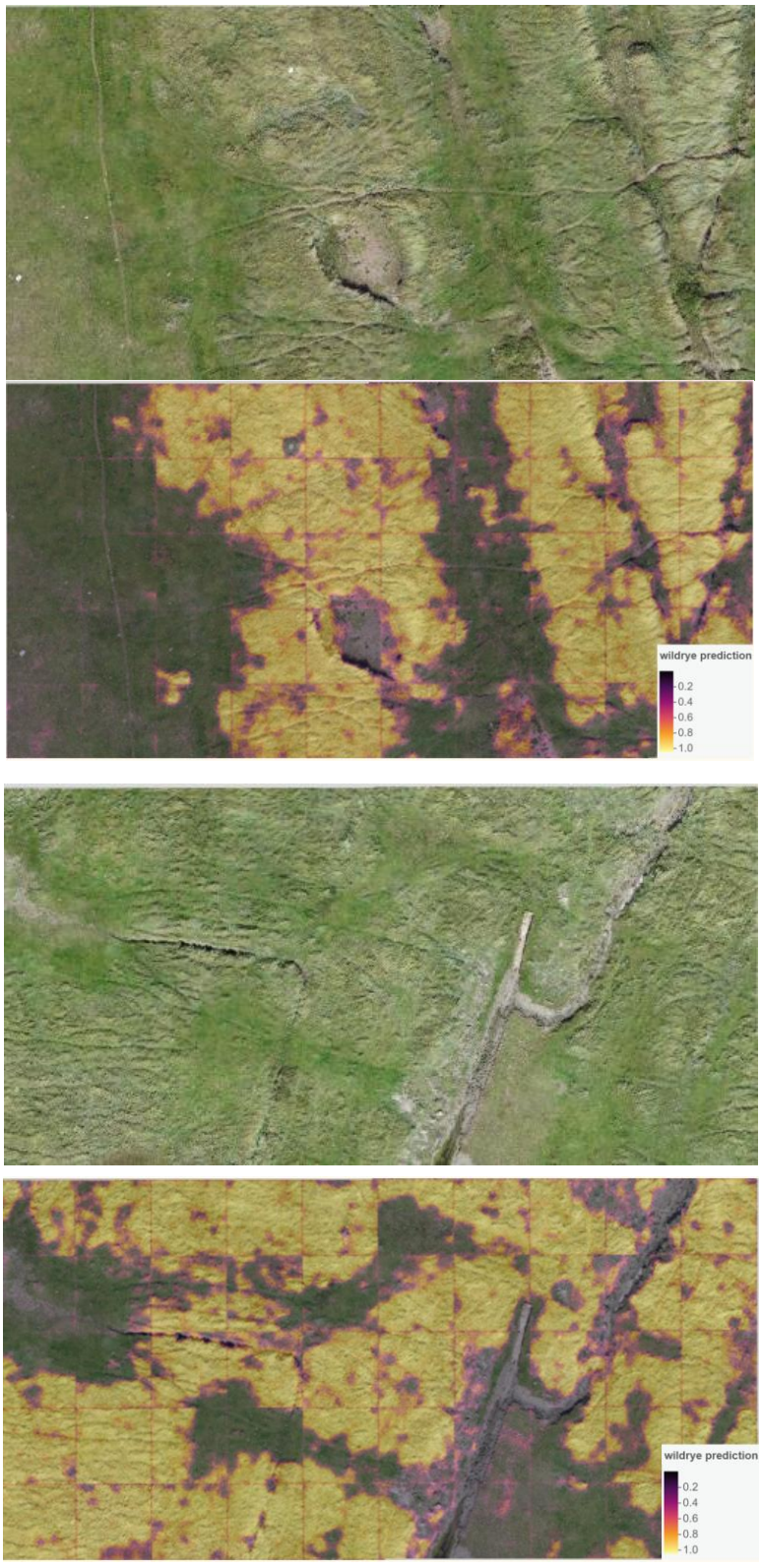


Source: author, 2022.

Furthermore, the same methodology applied to the preliminary results was applied on to the final results. Below you can view some examples of single images from the final results organized into three groups based on the same criteria as before.

The final results indicate that the average accuracy of the first group is not less than 96%. This is illustrated in the first two single images (Fig.12). However, when considering the second group as shown in the two single images (Fig.13), the accuracy between single images decreased by 5%, yet the results are still excellent as the accuracy falls between 91-95%.

Figure (12): figures (UAV imagery followed with its segmented results) illustrating samples of the prediction of the wildrye from the third group with accuracy not less than 96%



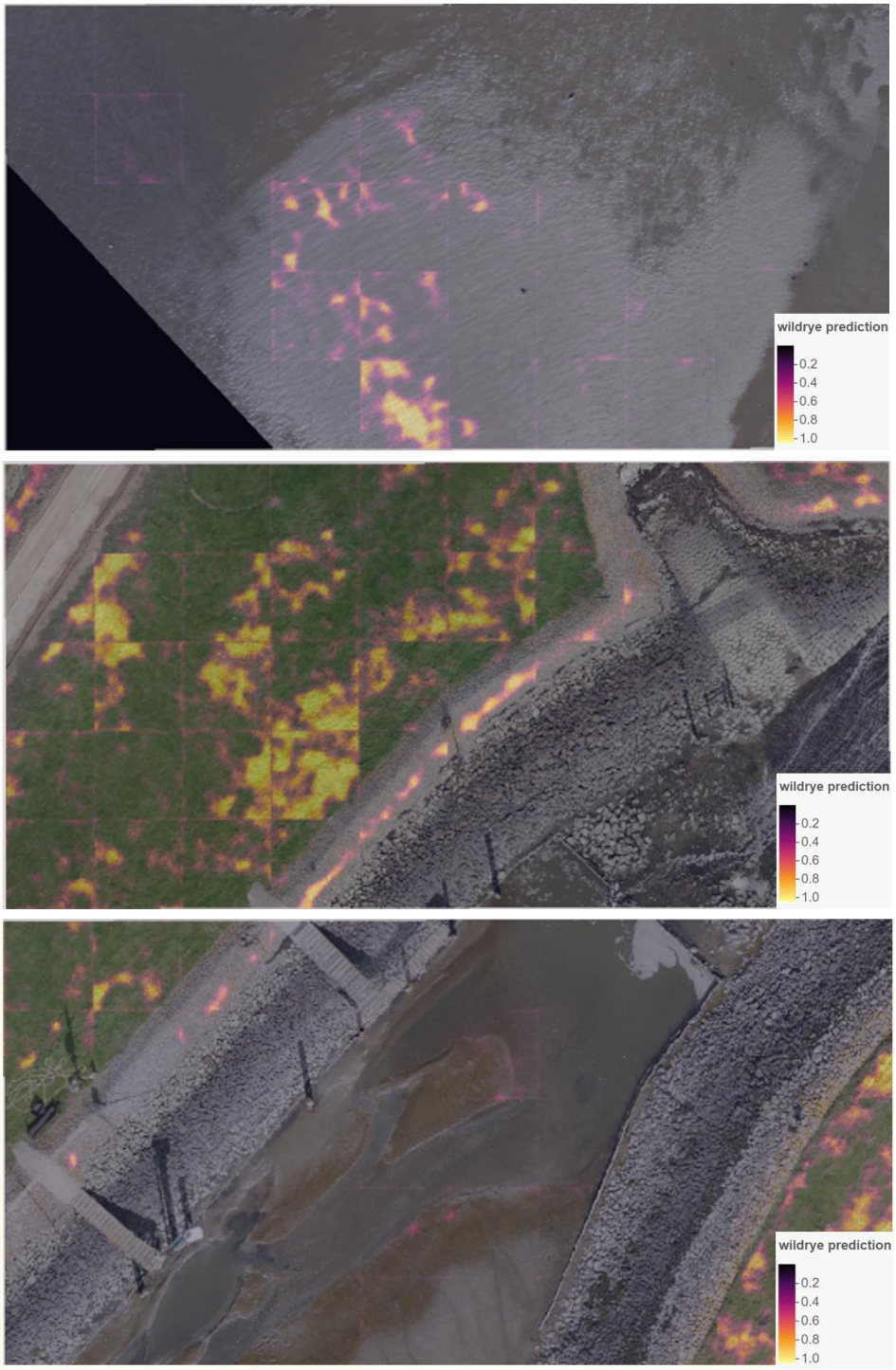
Source: author, 2022.

Figure (13): figures (UAV imagery followed with its segmented results) illustrating samples of the prediction of the wildrye from the third group with accuracy ranged between 91-95%

Source: author, 2022.

However, the results of the third group of the following three single images (fig.14) indicate a high decrease in accuracy with 10% reaching 80% and still can be interpreted visually.

Figure (14): figures illustrating samples of the prediction of the wildrye from the third group with accuracy ranged between 80-90%



Source: author, 2022.

4-3 Discussion

The results of the analysis, as can be seen, were highly positive; however, there was some variation in accuracy depending on the environment of the single images. Upon analyzing the different groups of results, one major factor was identified as the primary reason for this.

In all groups, an apparent increase in accuracy was evident between the preliminary and final results. The increase was made by increasing the training sample from 0.1% to 0.3% of the total number of subsets. While this represents an increase of 33%, 0.3% of the total number of subsets is still considered a low percentage.

Upon reading the differences between the preliminary and final results of the three experimental groups, it could be concluded that when the subset contains the same environmental elements that were contained in the training dataset, the accuracy was on the maximum side, while it was on the minimum side when the environment contained elements that were not contained in the training dataset.

In order to test this hypothesis, we applied the confusion matrix to the individual images of the different groups and compared it with the training dataset. In the first group, wildrye covers most of these single images with the appearance of other habitats that most of it included in the training datasets. While the second includes more variation of elements, it still includes wildrye, but some of the environment elements of this group were not included in the training dataset. Furthermore, the third group appeared to only occasionally include wildrye, but at the same time it included elements that were not largely present in the training dataset, such as surface water and dirt roads.

In the non-trained environment, such as surface water bodies and physical structures, accuracy was at its lowest. However, the accuracy reached a maximum of 75% with few wildrye areas appearing in such datasets in the preliminary results, whereas with the inclusion of more of these elements in the training sample the accuracy reached a maximum of 90%.

As the diversity of the elements in the training dataset was increased, the accuracy (first group) increased by 10%, while the accuracy (second group) increased approximately by 15% when the variation of the environmental elements was increased and an increase in the not wildrye areas was introduced.

Therefore, the identification of most, if not all, the elements of the study area's distinct environment and its representation in the training dataset is considered one of the main

challenges involved with developing a Deep Learning model for the automated segmentation of Elymus spread in UAV imagery.

As a result of combining the transfer learning technique and Data Augmentation, the training dataset we have been using has been enhanced and the results were optimized and optimized with only 0.3% as a percentage of the training dataset used. However, the diversity of the training dataset and the inclusion of the elements was still required to ensure the maximum level of accuracy of the model in different environments.

The other major challenge was the issue of computational power, as the process of implementing Deep Learning was computationally very expensive, requiring large amounts of memory and computational resources, and it is not easily transferred to other problems as deep learning requires expensive graphics cards. Thus, the first two research questions have been answered.

In order to answer the third and final research question, even with imagery that included only RGB data, the final results were also compared to the object-based image analysis (OBIA) model results, which used imagery with more multispectral information from (Oldeland et al., 2021) good accuracy (kappa = 0.937). In general, the results of both models were close to one another. However, as explained previously, the average accuracy of the deep learning model varies with the type of environment in which it is used, and therefore further analysis is warranted to compare the two results using the same method used in our study between single images.

CHAPTER 5: CONCLUSION, RECOMMENDATIONS, AND FUTURE WORK

5- Conclusion, recommendations, and future work

In this chapter two distinct sections are presented, the first section represents the conclusion of the study, and the second section is intended to provide some recommendations and potential future directions.

5-1 Conclusion

In different areas of Europe, Athyrics Elymus has been observed engulfing smaller species in low marsh habitats. Machine learning/deep learning and its algorithms have been used to detect various elements in many fields, which has helped scientists find solutions for many problems and challenges in various scientific fields. In the study area of Hallig Nordstrandischmoor, a deep learning model has been applied to RGB 10 cm UAV imagery for the automated segmentation of Athurics Elymus.

The final results were good and can be classified among three levels of accuracy. Furthermore, these groups are also characterized by different levels of diversity in their environments. In the first group with an accuracy greater than 96%, the wildrye covers the majority of these single images with a low percentage of other habitats. The second group with the highest accuracy ranged between 91-95%, which includes more elements of variation, but still included wildrye. Further, the third group, with an accuracy range between 80-90%, did not appear to include wildrye very often, but at the same time included elements that were not largely present in the training dataset, such as surface water and dirt roads.

The use of the transfer learning technique shown in the VG16 as the backbone for U-net deep learning architecture, coupled with the use of the data augmentation algorithm, was one of the primary advantages of using deep learning. While features are automatically derived and optimally tuned to achieve desired results. As a result, time and effort are significantly reduced and high accuracy is achieved. However, it was not enough to ensure high accuracy in all areas of the study area as it was necessary to include most or all of the different elements within the study area in the training dataset. Another limitation was the fact that deep learning required a large amount of memory and computational resources, as deep learning requires expensive GPUs. This resulted in an extremely high computational cost.

5-2 Recommendations and future work

The diversity of the training data is more important than the amount of the training data set, and therefore, it is recommended to train the same deep learning model again by including the missing elements from the study area to the training dataset in order to optimize the model and achieve the highest level of accuracy.

Further analysis is needed in the comparison between the OBIA model and the final results of the deep learning model, in order to have a more detailed answer to the third research question. However, it is evident that the deep learning model is still capable of achieving greater results.

In the future, it may be possible to use the same trained model to segment Athyrics Elymus in the same study area by using newer imagery. The deep learning model integrating with its associated techniques gives the advantage of being able to use the same model to predict in different Imageries if it includes the same environments. This increases the probability of obtaining good results, particularly during bad weather conditions. Moreover, the integration of freely available medium resolution satellite data such as Sentinel is also possible. The benefit of this technique is that it will allow for the monitoring of habitats in near-real-time, which will also save a tremendous amount of money and time.

REFERENCES

References

Abozeid, A., Alanazi, R., Elhadad, A., Taloba, A. I., & Abd El-Aziz, R. M. (2022). A Large-Scale Dataset and Deep Learning Model for Detecting and Counting Olive Trees in Satellite Imagery. Computational Intelligence and Neuroscience, 2022, 1–8. https://doi.org/10.1155/2022/1549842

Bruzzone, L., & Persello, C. (2010). Recent trends in classification of remote sensing data: Active and semisupervised machine learning paradigms. 2010 IEEE International Geoscience and Remote Sensing Symposium, 3720–3723. https://doi.org/10.1109/IGARSS.2010.5651236

Gómez, J. A., Patiño, J. E., Duque, J. C., & Passos, S. (2019). Spatiotemporal Modeling of Urban Growth Using Machine Learning. Remote Sensing, 12(1), 109. https://doi.org/10.3390/rs12010109

Hashi, A. O., Abdirahman, A. A., Elmi, M. A., Hashi, S. Z. M., & Rodriguez, O. E. R. (2021). A Real-Time Flood Detection System Based on Machine Learning Algorithms with Emphasis on Deep Learning. International Journal of Engineering Trends and Technology, 69(5), 249–256. https://doi.org/10.14445/22315381/IJETT-V69I5P232

Helin, R., Indahl, U. G., Tomic, O., & Liland, K. H. (2022). On the possible benefits of deep learning for spectral preprocessing. Journal of Chemometrics, 36(2). https://doi.org/10.1002/cem.3374

Ibtehaz, N., & Rahman, M. S. (2020). MultiResUNet: Rethinking the U-Net architecture for multimodal biomedical image segmentation. Neural Networks, 121, 74–87. https://doi.org/10.1016/j.neunet.2019.08.025

Johnson, S. A. (2011). Challenges in Health and Development. Springer Netherlands. https://doi.org/10.1007/978-90-481-9953-2

Kamble, S. J., & Kounte, M. R. (2020). Machine Learning Approach on Traffic Congestion Monitoring System in Internet of Vehicles. Procedia Computer Science, 171, 2235–2241. https://doi.org/10.1016/j.procs.2020.04.241

Khanzode, K. C. A., & D. Sarode, R. (2020). ADVANTAGES AND DISADVANTAGES OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING: A LITERATURE REVIEW. International Journal of Library & Information Science, 9(1), 30–36. https://iaeme.com/Home/article_id/IJLIS_09_01_004

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. Nature, 521(7553), 436–444. https://doi.org/10.1038/nature14539

Liang, H., & Li, Q. (2016). Hyperspectral Imagery Classification Using Sparse Representations of Convolutional Neural Network Features. Remote Sensing, 8(2), 99. https://doi.org/10.3390/rs8020099

Liang, X., Zhang, T., Xie, M., & Jia, X. (2021). Analyzing bicycle level of service using virtual reality and deep learning technologies. Transportation Research Part A: Policy and Practice, 153, 115–129. https://doi.org/10.1016/j.tra.2021.09.003

Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A. W. M., van Ginneken, B., & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. Medical Image Analysis, 42, 60–88. https://doi.org/10.1016/j.media.2017.07.005

Ma, L., Liu, Y., Zhang, X., Ye, Y., Yin, G., & Johnson, B. A. (2019). Deep learning in remote sensing applications: A meta-analysis and review. ISPRS Journal of Photogrammetry and Remote Sensing, 152, 166–177. https://doi.org/10.1016/j.isprsjprs.2019.04.015

Maxwell, A. E., Warner, T. A., & Fang, F. (2018). Implementation of machine-learning classification in remote sensing: An applied review. International Journal of Remote Sensing, 39(9), 2784–2817. https://doi.org/10.1080/01431161.2018.1433343

Montavon, G., Orr, G. B., & Müller, K.-R. (Eds.). (2012). Neural Networks: Tricks of the Trade: Second Edition (Vol. 7700). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-35289-8

Navab, N., Hornegger, J., Wells, W. M., & Frangi, A. (Eds.). (2015). Medical Image Computing and Computer-Assisted Intervention -- MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part II (Vol. 9350). Springer International Publishing. https://doi.org/10.1007/978-3-319-24571-3

Oldeland, J., Revermann, R., Luther-Mosebach, J., Buttschardt, T., & Lehmann, J. R. K. (2021). New tools for old problems—Comparing drone- and field-based assessments of a problematic plant species. Environmental Monitoring and Assessment, 193(2), 90. https://doi.org/10.1007/s10661-021-08852-2

O'Shea, K., & Nash, R. (2015). An Introduction to Convolutional Neural Networks. ArXiv:1511.08458 [Cs]. http://arxiv.org/abs/1511.08458

Pétillon, J., Ysnel, F., Canard, A., & Lefeuvre, J.-C. (2005). Impact of an invasive plant (Elymus athericus) on the conservation value of tidal salt marshes in western France and implications for management: Responses of spider populations. Biological Conservation, 126(1), 103–117. https://doi.org/10.1016/j.biocon.2005.05.003

Pitaloka, D. A., Wulandari, A., Basaruddin, T., & Liliana, D. Y. (2017). Enhancing CNN with Preprocessing Stage in Automatic Emotion Recognition. Procedia Computer Science, 116, 523–529. https://doi.org/10.1016/j.procs.2017.10.038

Qiu, L., Gao, T., Gunnarsson, A., Hammer, M., & von Bothmer, R. (2010). A methodological study of biotope mapping in nature conservation. Urban Forestry & Urban Greening, 9(2), 161–166. https://doi.org/10.1016/j.ufug.2010.01.003

Rahul, M., Shiva Saketh, K., Sanjeet, A., & Srinivas Naik, N. (2020). Early Detection of Forest Fire using Deep Learning. 2020 IEEE REGION 10 CONFERENCE (TENCON), 1136–1140. https://doi.org/10.1109/TENCON50793.2020.929372

Razavian, A. S., Azizpour, H., Sullivan, J., & Carlsson, S. (2014). CNN Features off-the-shelf: An Astounding Baseline for Recognition. ArXiv:1403.6382 [Cs]. http://arxiv.org/abs/1403.6382

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. ArXiv:1505.04597 [Cs]. http://arxiv.org/abs/1505.04597

Scebba, G., Zhang, J., Catanzaro, S., Mihai, C., Distler, O., Berli, M., & Karlen, W. (2022). Detect-and-segment: A deep learning approach to automate wound image segmentation. Informatics in Medicine Unlocked, 29, 100884. https://doi.org/10.1016/j.imu.2022.100884

Shen, L., Margolies, L. R., Rothstein, J. H., Fluder, E., McBride, R., & Sieh, W. (2019). Deep Learning to Improve Breast Cancer Detection on Screening Mammography. Scientific Reports, 9(1), 12495. https://doi.org/10.1038/s41598-019-48995-4

Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. ArXiv:1409.1556 [Cs]. http://arxiv.org/abs/1409.1556

Song, J., Kim, D., & Kang, K. (2020). Automated Procurement of Training Data for Machine Learning Algorithm on Ship Detection Using AIS Information. Remote Sensing, 12(9), 1443. https://doi.org/10.3390/rs12091443

Valéry, L., Bouchard, V., & Lefeuvre, J.-C. (2004). Impact of the invasive native species Elymus athericus on carbon pools in a salt marsh. Wetlands, 24(2), 268–276. https://doi.org/10.1672/0277-5212(2004)024[0268:IOTINS]2.0.CO;2

Vives-Boix, V., & Ruiz-Fernández, D. (2021). Synaptic metaplasticity for image processing enhancement in convolutional neural networks. Neurocomputing, 462, 534–543. https://doi.org/10.1016/j.neucom.2021.08.021

Wöber, W., Mehnen, L., Sykacek, P., & Meimberg, H. (2021). Investigating Explanatory Factors of Machine Learning Models for Plant Classification. Plants, 10(12), 2674. https://doi.org/10.3390/plants10122674

Zeiler, M. D., Krishnan, D., Taylor, G. W., & Fergus, R. (2010). Deconvolutional networks. 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2528–2535. https://doi.org/10.1109/CVPR.2010.5539957

Arden Dertat, 2017. Applied Deep Learning - Part 4: Convolutional Neural Networks https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

Christian Knoth, 2020. Introduction to Deep Learning in R for the Analysis of UAV-based Remote Sensing Data OpenGeoHub summer school 2020. https://dachro.github.io/ogh_summer_school_2020/Tutorial_DL_UAV.html#preparing_your_data