Norwegian University
of Life Sciences

Master's Thesis 2023 30 ECTS
Faculty of Science and Technology

Predicting Grain Yield by Utilizing Multispectral Images and Convolutional Neural Network

Sara Elisabeth Idris
Data Science

Preface

As I complete my studies at the Norwegian University of Life Sciences, I present this thesis as the culmination of my academic journey. During my time as a student at NMBU, I have gained valuable academic and social skills that will aid me as I face new challenges in the future. The topic for this thesis was suggested by Ingunn Burud, an associate professor in Physics at NMBU, and my work on it began in January 2023. Looking back, this semester has been both challenging and rewarding, as I encountered some obstacles but also made new friendships in the master hall. Most importantly, I have learned a great deal over the past few months.

I would like to express my gratitude to several individuals who have supported me throughout the writing process. I am especially grateful to my supervisor, Ingunn Burud, and my co-supervisors Kristian Hovde Liland and Sahameh Shafiee, for their knowledge, advice, and time throughout the entire process. Their guidance and our discussions have been motivating and immensely helpful in producing this thesis. Finally, I extend my thanks to all of the fellow students with whom I have shared my everyday experiences.

Ås, May 2023

Sara Elisabeth Idris

Abstract

According to the UN's sustainability goals, hunger should have been eradicated by 2030, but the number is going the wrong way, with around 800 million people who suffer from undernourishment in 2022. Therefore, there is a pressing need to produce more food with fewer resources while minimizing crop losses. A combination of remote sensing and deep learning can contribute to more efficient phenotyping of plants. Aerial photography is an excellent alternative to traditional manual or mechanical plant health assessments. By applying remote sensing technologies, drones are now economically feasible to capture multispectral images of fields. The use of deep learning algorithms, such as Convolutional Neural Networks (CNNs), allows for the simultaneous analysis of large areas, significantly saving time. This research examines multispectral unmanned aerial vehicle imagery to extract essential features for wheat yield prediction in wheat breeding programs in NMBU at Ås. The thesis dataset comprises two fields with crops, each with eight multispectral images for each field, with five bands each (red, green, blue, near-infrared, and red-edge).

In this thesis, two simple CNN models with different architectures were utilized. The different combinations of datasets were trained using the two CNN architectures. The most effective model was the model, which included additional variables, such as days to heading and fertilization level, has enhanced the model's predictive accuracy. This thesis employs a CNN to predict grain yield by utilizing all spectra recorded for each plot. The results suggest that this approach is satisfactory when predicting wheat grain yield.

Unlike similar studies, this thesis takes a different approach by utilizing the entire plot as a multispectral image, which allows for extracting all spectra recorded for each plot. Previous studies have used the median value of the plots to make predictions and have not incorporated CNN as part of their methodology. The resultant CNN model achieved an R^2 score of 0.885.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AdaGrad **Adaptive Gradient Algorithm** - *Optimization algorithm that adapts learning rates based on gradient history*

Adam **Adaptive Moment Estimation** - *Deep learning optimization algorithm with adaptive learning rates*

ANN **Artificial Neural Network** - *Machine learning model inspired by biological neural networks*

CNN **Convolutional Neural Network** - *A deep learning network for image classification that learns hierarchical features through convolutional layers*

DH **Days to Heading** - *Measure in plant science for time to first seed head appearance, used to study plant growth and select crop varieties*

EM **Electromagnetic** - *Relating to the electric and magnetic fields generated by moving charges, underlying natural phenomena such as light, radio waves, and X-rays*

EVI **Enhanced Vegetation Index** - *Remote sensing index that assesses vegetation health and density while considering atmospheric and background effects.*

GIS **Geographic Information Systems** - *Computer-based systems used to capture, store, analyze, and display geographical data*

HSI **Hyperspectral Imaging** - *Imaging technique that captures and processes electromagnetic radiation across the spectrum*

MAE **Mean Absolute Error** - *Metric used to measure prediction accuracy in machine learning, calculated as the average absolute difference between predicted and actual values*

MAPE **Mean Absolute Percentage Error** - *a metric commonly used in machine learning to evaluate forecast accuracy, calculated as the average percentage difference between predicted and actual values*

MBE **Mean Bias Error** - *A measure used in statistics to quantify average deviation of estimates from their true values, providing information on direction and magnitude of bias*

MSE **Mean Squared Error** - *Metric used to measure prediction accuracy in machine learning, calculated as the average of the squared differences between predicted and actual values*

MSI **Multispectral Imaging** - *Capturing and processing images using multiple spectral bands beyond human vision*

MTCI **MERIS terrestrial chlorophyll index** - *A vegetation index derived from satellite observations used to estimate vegetation chlorophyll content, monitor health and productivity*

NaN **Not a Number** - *A data type in programming representing an undefined value*

NDVI **Noramlized Difference Vegetation Index** - *Remote sensing index that assesses vegetation health and density using red and near-infrared bands of the electromagnetic spectrum*

NIR **Near Infrared** - *Band of the electromagnetic spectrum used to detect vegetation and other materials in remote sensing*

NN **Neural Networks** - *AI algorithm modeled after the human brain, learning and recognizing patterns in data*

ReLU **Rectified Linear Units** - *Activation function in neural networks that sets negative values to zero and keeps positive values, improving accuracy and training speed*

RGB **Red Green Blue** - *Primary colors used in digital displays to create a wide range of colors*

RMSProp **Root Mean Square Propagation** - *Deep learning optimization algorithm that adapts learning rates based on the moving average of squared gradients*

SD **Standard Deviation** - *A statistical measure that indicates the amount of variation or dispersion of a set of data values relative to their mean or expected value*

SGD **Stochastic Gradient Descent** - *Stochastic gradient descent is an iterative optimization algorithm commonly used for training neural networks*

UAV **Unmanned Aerial Vehicle** - *A type of aircraft that operates without a human pilot onboard*

Chapter 1

Introduction

1.1 Background and Motivation

According to the UN's sustainability goals, hunger should have been eradicated by 2030, but the numbers are going the wrong way. It is around 800 million people who suffer from undernourishment in 2022, which is 150 million more than before the covid-19 pandemic [1]. In addition to more people suffering from hunger, the projections are that the world's population will exceed 10 billion in 2050 [2]. There is a pressing need to produce more food with fewer resources while minimizing crop losses. Wheat, the most widely used crop providing approximately 20% of the world's food calories and proteins, is essential in this regard [3]. However, farmers face increasing challenges due to extreme climate changes, which necessitates the development of wheat varieties that can thrive in a broader range of environments, including warmer and more humid conditions [4]. By providing detailed information about wheat phenotypes under different conditions, aerial photograph-based can contribute to the development of more resilient and productive wheat varieties, ultimately helping to address global food security challenges.

Phenotyping is the comprehensive characterization of an organism's observable traits from genetic and environmental interactions. Recent advances in remote sensing technologies have enabled scientists to use aerial photographs to perform high-throughput phenotyping of crops such as wheat [5]. By extracting detailed morphological and physiological information from these images, researchers can gain insights into plant growth and development, ultimately leading to improved crop management practices and increased yields. Following the forecast for 2050, the demand for grain will increase from 2.1 billion tonnes in 2009 to around 3 billion tonnes [6]. It is recommended to increase cereal output by over one billion tonnes annually. It would also be essential to produce the foods necessary to provide nutrition security and adequately feed the world's population [4]. Traditional phenotyping methods involve manual observation of the crops, which may harm the crops and soil. A great alternative to manual or mechanical plant health assessments is aerial photography. Using aerial imagery will significantly decrease the amount of time it takes to scan a single grain of wheat and also decrease the number of crops that are damaged. Aerial images provide an opportunity to create and test new technologies for image pro-

cessing and genomic production, enabling a more accurate selection of new plant cultivars [7].

Since remote sensing technologies have advanced so quickly over the years, they are now economically feasible, making it easier for more individuals to utilize, i.e., drones to take multispectral images (MSI) of their fields [8, 9]. The use of deep learning models to quickly analyze MSI has also been a notable advancement in machine learning. There are various methods for spectral image analysis. One approach is to utilize a Convolutional Neural Network (CNN) as a method, where an image is processed to identify and capture the significant characteristics for predicting grain yield and other related factors.

Deep learning offers a powerful tool for quickly analyzing large data volumes, for example, large areas of crops. Combining remote sensing and deep learning can save time and reduce the need for destructive manual or mechanical assessments. Because it is possible to analyze large areas at once. As mentioned earlier, to make agricultural and phenotyping more efficient, new methods are required, i.e., to check larger fields for diseases without destroying the fields.

Deep learning and spectral images make it possible to phenotype wheat securely. Initially, photographs are captured of the fields, which are then used to create a model and train on labeled data to detect patterns in the plants. In order to accomplish this task, a deep-learning algorithm called CNN is employed, specifically designed for processing digital images. With the help of MSI, CNN can extract information about chemical materials and use them to calculate Normalized Difference Vegetation Index (NDVI), Enhanced Vegetation Index (EVI), MERIS Terrestrial Chlorophyll Index (MTCI) or other valuable indexes to estimate the plants' health.

This thesis is a part of the Phenocrop project, funded by the Research Council of Norway. One of the project's goals is to speed up plant breeding by utilizing high-throughput phenotyping. This thesis aims to predict the grain yield for the plots grown at Ås, Vollebekk. The deep learning technique of CNN will be utilized to solve this problem. In 2019, the Phenocrop project replaced the vPheno (virtual phenomics) project, initiated in 2017. It took over its objectives. One particular purpose in acquiring image data from wheat fields is to estimate the grain yield for various types of wheat grown in diverse environments, which is particularly crucial as wheat is an essential staple crop. Over the past few years, multispectral images of wheat fields have been collected. These images give farmers valuable information regarding grain yield and their fields' maturity days. Such information aids farmers in making informed management decisions at the farm level [10].

Since 2017, the Phenocrop project has been collecting data to explore various techniques for phenotyping. Shafiee et al. [11] and Ijaz et al. [12] have conducted phenotyping using mean values for the spectra of each plot. In their study, Shafiee et al.[11] compared the outcomes using LASSO regressor with an internal feature selector and Support Vector Regression (SVR) in combination with Sequential Forward Selection (SFS) to predict grain yield. By combining models based on all indices and dates, up to 92% of the variance in grain yield of the test set could be explained.

The crops used in this thesis were previously studied by both Shafiee et al. [11] and Ijaz

et al. [12], who made predictions about their grain yield. Ijaz also included weather data to see if it helps the machine learning model to predict the amount of grain yield.

This thesis differs from previous methods as it utilizes a CNN to forecast grain yield by utilizing all spectra obtained for each plot, unlike the previous method. Additionally, the images used in this thesis were captured in 2022, while the previous research relied on images taken in 2018 and 2019.

1.2 Objectives

This thesis aims to address the feasibility of predicting grain yield for wheat plots from multispectral images by developing a CNN regression model. The objective of this thesis can be summarized in two questions:

1. Does utilize all spectra recorded for each plot instead of the median value enhance the effectiveness of a CNN for gaining deeper insights into a data set?
2. How does incorporating days to heading (DH) and fertilization level as variables affect the performance of models when predicting grain yield?

The findings of this research will contribute to the understanding of the feasibility and effectiveness of utilizing multispectral images and CNN regression for grain yield prediction and have the potential to inform decision-making in agriculture.

1.3 Structure

The structure of this thesis is outlined as follows: Section 2 will provide a comprehensive overview of the theory and background of spectral imaging and machine learning models. Section 3 will detail the selection and description of the multispectral image datasets, as well as the implementation and training of the models. The results and a short discussion for each model will be presented in Section 4, while Section 5 and 6 will discuss and summarize the findings. The appendix contains a table of program and Python modules utilized in this thesis. The source code for this thesis are listed in Appendix A.

Chapter 2

Theory

This chapter will provide the fundamental theory of plants, which is crucial for a comprehensive understanding of the project presented in this thesis. Additionally, the theory behind the machine and deep learning techniques employed for spectral image analysis will be explored.

2.1 Remote Sensing

Remote sensing is obtaining information about objects on the earth's surface without physical contact, as shown in Figure 2.1.1. Remote sensing is done by analyzing signals, like optical, acoustical, or microwave, that travel between the object and the vehicle used for remote sensing [13]. There are a lot of different vehicles used for remote sensing. Satellites orbiting the earth at various altitudes can capture images and other data using a range of sensors, including optical and radar [14]. Airplanes with sensors and cameras can cover large areas in a single flight, making them useful for mapping and monitoring purposes [15]. Drones, also known as unmanned aerial vehicles (UAVs), utilized in this thesis' investigation, can fly at lower altitudes than airplanes and capture high-resolution images and data over small areas [16]. In this particular investigation, the focus is on examining signals in the visible, near-infrared (NIR), and red-edge spectra to investigate different types of wheat in a field [13].

Figure 2.1.1: *Illustration of remote sensing done by a UAV where it obtains information about the wheat trials.*

2.1.1 Electromagentic Radiation

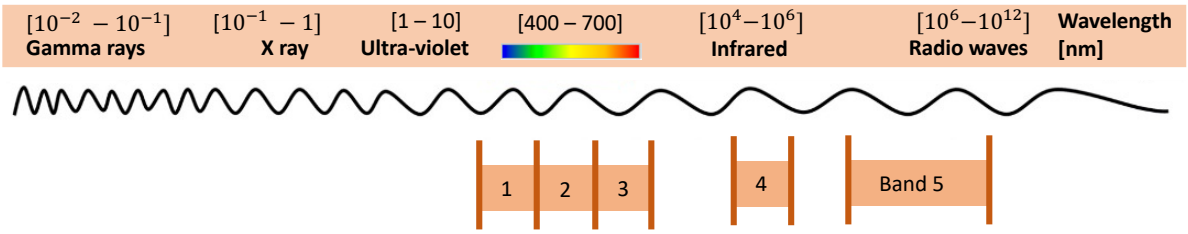


Figure 2.1.2: *Representation of the Electromagnetic spectrum, and a multispectral representation of five bands, red, green, blue, NIR and red-edge.*

Electromagnetic (EM) radiation refers to the propagation of energy through the movement of particles that exhibit wave-like and particle-like properties at the speed of light. Figure 2.1.2 shows the range of the EM spectrum, which encompasses various types of waves, including radio, microwave, infrared, visible, ultraviolet, x-ray, and gamma-ray [17]. The visible spectrum is a small portion of this spectrum that can be detected by the human eye, ranging from 400-700 nanometers [18].

Investigating the possibility of utilizing a more significant portion of the EM spectrum for machine learning models may be particularly relevant in plant phenotyping. The amount of reflected near-infrared radiation can provide information about the chlorophyll present in plants, indicating their overall health. By examining the reflected radiation beyond the visible spectrum, it may be possible to gain insights into the health of plants and soil

rather than solely focusing on how the plants look with the blue eye [19].

2.2 Phenotyping of Plants

Due to significant climate changes, large parts of the crops are destroyed, leading to farmers expanding the fields, deforestation, and increased climate change [20]. In order to produce grain efficiently without increasing the area, plant breeding is the solution. With the help of phenotyping of crops, it becomes possible and practical to check which type of crop is suitable for which type of climate. A quantitative description of a plant's morphological, ontogenetical, physiological, and biochemical characteristics is called *plant phenotyping*. Phenotyping must also happen without destroying the soil. Therefore, different approaches are being developed to stop the destruction of the soil. One of the ways to do this is by utilizing image analysis [21].

2.3 Vegetation Indices

Vegetation indices are mathematical formulas that utilize information from remote sensing units to say something about the physical properties of soil, water, and vegetation over a specific area. These indices are created by comparing how plants reflect or absorb different wavelengths of light, such as red, blue, and NIR light. There are various indices, such as the normalized difference vegetation index (NDVI), enhanced vegetation index (EVI), and MERIS terrestrial index (MTCI).

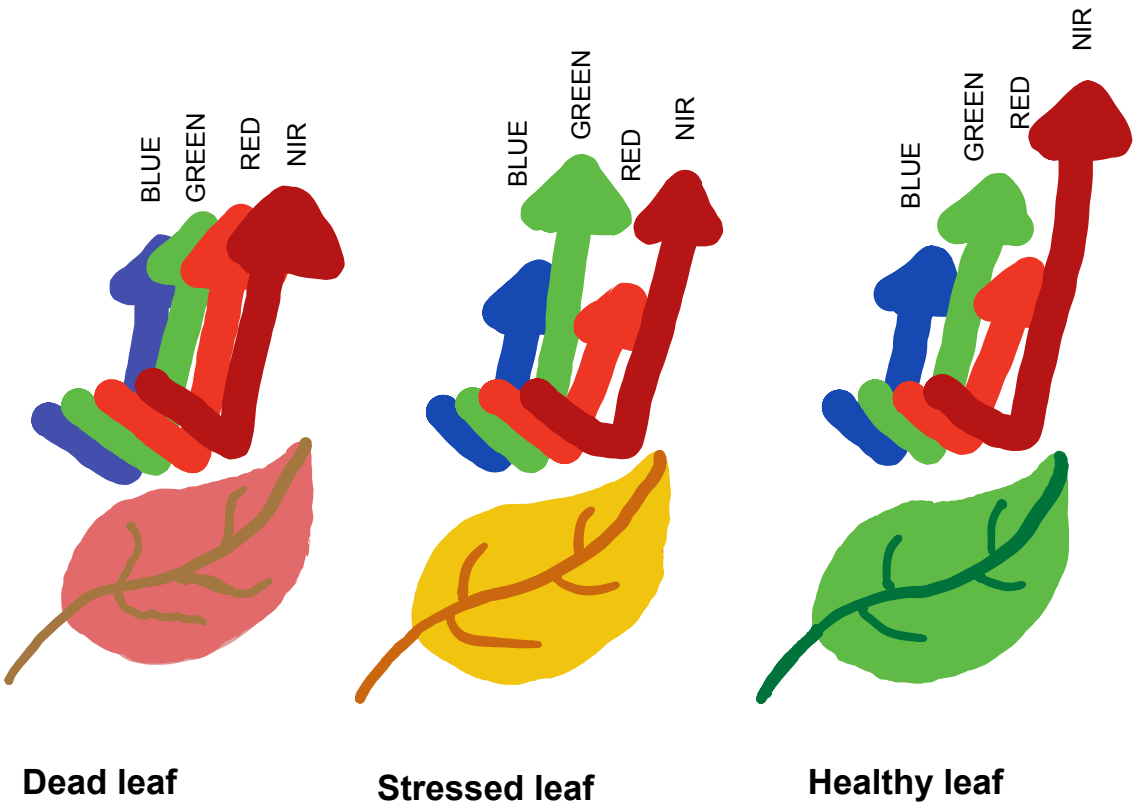


Figure 2.3.1: *The different stages of a plant, including dead, stressed, and healthy states. A dead plant does not exhibit significant reflectance in green and NIR radiation. A stressed plant reflects slightly higher levels of green and NIR radiation than blue and red. On the other hand, a healthy plant shows higher reflectance in green and significantly higher reflectance in NIR radiation compared to red and blue wavelengths. The figure is inspired by [22].*

Normalized Difference Vegetation Index is the most common vegetation index. The NDVI compares the reflected NIR with the absorbed red reflection, shown in Equation 2.1. As Figure 2.3.1 shows, the more reflected NIR and the more absorbed red, the more chlorophyll the plant contains. Hence the healthier the plant is. The information extracted from the NDVI can be used to monitor changes in vegetation over time, identify areas of stress or damage, and inform decisions related to land management and agriculture.

$$NDVI = \frac{NIR - Red}{NIR + Red} \tag{2.1}$$

Enhanced Vegetation Index is considered an improvement over the NDVI by Matsushita et al. [23]. EVI incorporates the blue band and the red-edge region of the electromagnetic spectrum in addition to the red band, and the calculation is shown in Equation 2.2. The NDVI equation, which does not consider feedback or adjustments based on changing atmospheric and canopy background conditions, can result in significant errors and uncertainties in estimating vegetation health. In comparison to NDVI, EVI was found to be more linearly correlated with green leaf area index (LAI) in crop fields [24], less

prone to saturation in temperate and tropical forests [25, 26], and minimally sensitive to residual aerosol contamination from extensive fires in the Amazon and Northern Asia [27, 28].

$$EVI = \frac{2.5NIR - RED}{NIR + 6RED - 7.5BLUE + 1} \tag{2.2}$$

The MERIS Terrestrial Chlorophyll Index is another mesure for chlorophyll content. The computation is shown in Equation 2.3. The MTCI considers the red-edge reflection, the area where a significant change in reflectance occurs between wavelengths 690 and 750 nm. This region indicates the transition from chlorophyll absorption to leaf scattering [29].

$$MTCI = \frac{NIR - REDEDGE}{REDEDGE - RED} \tag{2.3}$$

2.4 Spectral Image Acquisition

Conventional remote sensing techniques may pose challenges in capturing spatial and spectral information for food supplies. Examples of such techniques include physically sampling crops for laboratory analysis or using handheld spectrometers to measure the reflectance of individual leaves or plants directly. Hence, these techniques require direct contact with crops, which can cause damage or disturbance to the fields. However, newer remote sensing techniques that utilize non-destructive methods, such as MSI, may be able to capture more detailed information about the crops without harming them. Since 1972, when NASA launched Landsat 1 and introduced aerial MSI, the cost of equipment has significantly decreased, making multispectral sensors more widely available. Today, drones, planes, and satellites can all carry these sensors. With the advancements in both machine learning and spectral imaging, utilizing airborne photos has become an efficient method for phenotyping wheat and other tasks [30].

Over the years, there has been considerable development in the field of MSI due to the advances in sensing different materials. The fundamental components of an MSI system are a light source (i.e., a lighting system), a suitable objective lens, a wavelength dispersion tool, and a camera with a 2D detector. Independent of the kind of radiation, combining conventional imaging and conventional single-point spectroscopy in the MSI procedures [31].

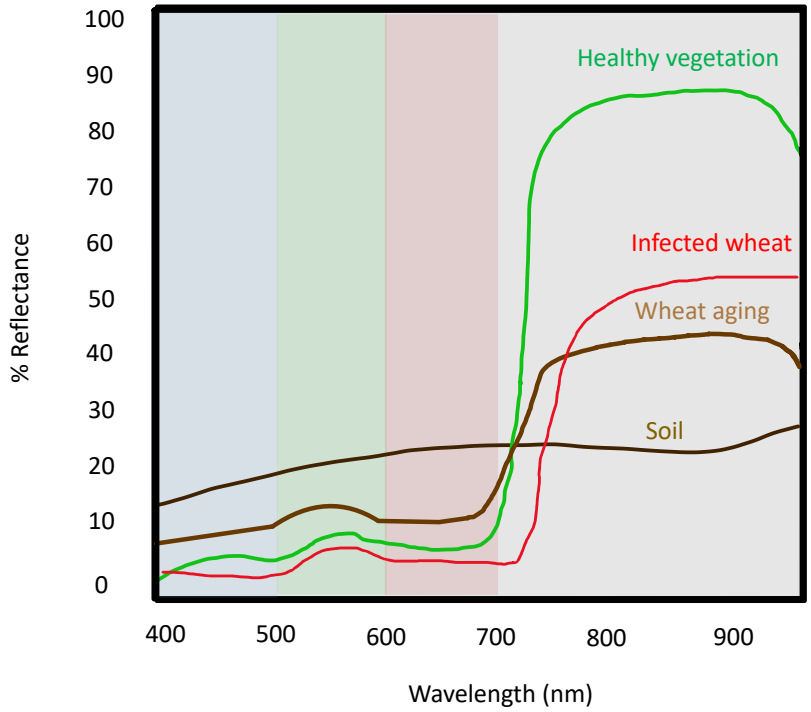


Figure 2.4.1: *Illustration of how the spectrum for different stages of wheat changes. Based on the signature of the interaction between matter and energy expressed in the spectrum, these spectra are utilized to retrieve information about the chemicals of wheat.*

Imaging spectroscopy involves capturing images in which the spectrum of energy entering the sensor is measured for every spatial resolution unit. As depicted in Figure 2.4.1, the various growth stages of wheat reflect different amounts of different parts of the spectrum. By analyzing the signature of the interaction between matter and energy expressed in the spectrum, this information is then utilized to extract various data, such as computing NDVI, EVI, or MTCI as explained earlier in this chapter [32].

2.5 Machine Learning

Machine learning is similar to how the human brain acquires information and insight in that it relies on input. However, while the brain searches for domains and entities and makes connections between them, machine learning algorithms use graphs and datasets to identify patterns to build a model based on sample data, known as training data, and make assumptions and predictions without being particularly programmed to do so [33].

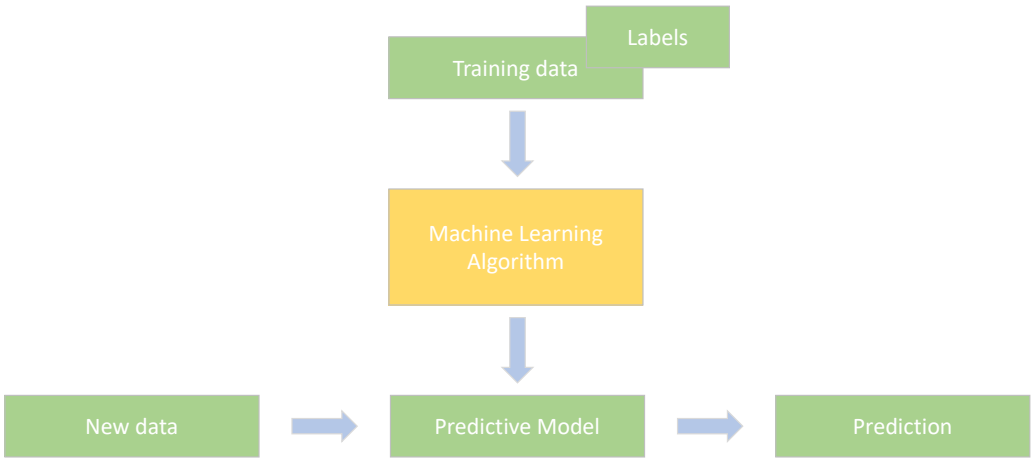


Figure 2.5.1: *A supervised machine learning algorithm is trained using labeled data, where the input data used for training is already associated with known output values. The algorithm identifies patterns from labeled data and uses them to predict new unseen data.*

In supervised learning, the model is trained on labeled data, which means that the input data used for training is already associated with known output values. The algorithm identifies patterns from existing labeled data and utilizes them to estimate target values for datasets without ground truth information; this workflow is depicted in Figure 2.5.1. After that, evaluating the performance of the chosen algorithm by looking at its accuracy and loss can reveal whether or not it can make accurate predictions based on unobserved data. Applying machine learning algorithms, such as regression or classification, can address various problems [34]. However, it is essential to note that machine learning has no one-size-fits-all solution; as indicated by the *No free lunch theorem*, there is no universal optimization algorithm that performs best for all possible problem instances. [35]. The choice of algorithm depends on the specific problem at hand, and experimenting with different methods is essential to achieve optimal accuracy.

Machine learning has wide-ranging applications, including urban land-use mapping, self-driving cars, and plant phenotyping [36, 37, 38]. It is a rapidly evolving field with diverse use cases and potential for solving various tasks.

2.5.1 Training and Validation

While training a model, it is possible to encounter overfitting, where the model becomes excessively tailored to the training data, including random variations. Splitting the dataset into a training set and a validation set is standard practice to address this problem. The model is trained on the training set, which contains input features and target values, and then evaluated on the validation set, which comprises new, unseen data. However, if the dataset is small, there may still be a risk of overfitting, as the training and validation sets may be too similar.

Another problem that can arise is underfitting, where the model needs to be more complex and have lower training accuracy than validation accuracy. In order to prevent underfitting, the model can be made more complex, or regularization can be decreased, depending on the specific situation. It is essential to carefully balance the model complexity and regularization to optimize model performance during training.

2.5.2 Deep Learning

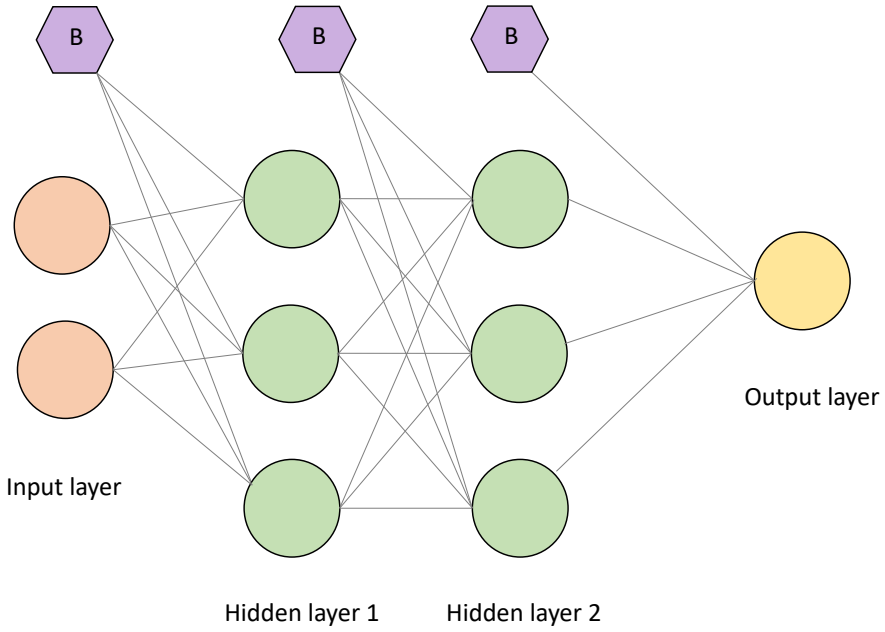


Figure 2.5.2: *Architecture of a simple ANN. The input layer is the initial layer that receives information from different sources. This information then passes through multiple layers and units within the network, transforming the way until the output unit eventually processes it for making predictions.*

An Artificial Neural Network (ANN) is created to imitate the human brain's process of acquiring knowledge and decision-making. These networks are essentially traditional computers programmed to function like interconnected brain cells. The network uses multiple layers of mathematical processing to interpret the information provided to it. A network is considered *deep* when it has more than one layer, and each layer may contain numerous artificial neurons or units. The input layer is the first layer and receives information from various sources. Most neural networks (NNs) are fully connected, meaning that each unit in one layer is linked to every unit in the next layer, as shown in Figure 2.5.2. However, convolutional layers are used instead of fully connected layers when dealing with image recognition. This type of network is called a Convolutional Neural Network (CNN), which will be discussed in greater detail in Section 2.5.2, and helps identify patterns in images to classify objects, classes, and categories. CNNs have three key features: they reduce the number of input nodes, tolerate small shifts in the pixel's location in the image, and take advantage of the correlation observed in complex images. The connections among the

units in the network possess varying weights, and the strength of these weights determines the level of influence a connection exerts. The information travels through all the layers and units within the network, transforming the way until it is ultimately utilized by the output unit to make a prediction [34].

There is a variety of machine learning techniques, including deep learning, which enables a computer to learn by doing as humans do naturally: by observation. In deep learning, machines train to do various tasks using various datasets, such as photos, text, or sound. The NN algorithm plus a sizable collection of labeled data allow the model to solve complicated problems. The main difference between a deep learning algorithm and a machine learning algorithm is that the feature extraction has to happen manually when utilizing a machine learning algorithm. In contrast, when utilizing a deep learning algorithm, the relevant features are extracted from the image automatically [39].

Backpropagation

Deep learning models utilize a loss function and an optimization strategy. In NNs, the process of generating output values is known as forward propagation. During training, the input data is propagated through the network, and the resulting output is compared to the ground truth, representing the measured grain yield in this case. Based on this comparison, the network weights are adjusted to minimize the error predicted by the model. Backpropagation, introduced by Rumelhart et al. [40] in 1986, is the most commonly used technique for updating the weights in a NN. Backpropagation involves iteratively modifying the weights in the network to minimize the error predicted by the model. Backpropagation is done by calculating the gradient of the loss function with respect to the weights and biases in the network using the chain rule. Equation 2.4 states that the derivative of F with respect to x can be computed by first finding the derivative of the outermost function F with respect to its direct input g, then multiplying it by the derivative of g with respect to its input h, and so on until we reach the innermost function v, whose derivative with respect to x gives the final term in the product. Thus, in deep networks with multiple layers and activation functions, small derivatives can lead to vanishing gradients, which can result in minimal weight updates for the earlier layers during backpropagation. In contrast, the later layers receive significant updates and pose a challenge in training deep NNs [34].

$$\frac{dF}{dx} = \frac{d}{dx}F(x) = \frac{d}{dx}f(g(h(u(v(x))))) = \frac{df}{dg} \cdot \frac{dg}{dh} \cdot \frac{dh}{du} \cdot \frac{du}{dv} \cdot \frac{dv}{dx} \tag{2.4}$$

In order to prevent this issue, the Rectified Linear Units (ReLU) activation function can be employed as it helps alleviate the vanishing gradient problem by maintaining a gradient of 1 for positive input values, allowing for more effective weight updates in the earlier layers, leading to improved training performance and better learning of complex patterns in the data [34].

Activation Functions

Activation functions are mathematical functions used in a NN to introduce non-linearity into the output of a neuron. The non-linearity is essential because it allows the network to learn complex patterns and relationships. In Figure 2.5.3 two of the most common activation functions are depicted, respectively, ReLU and Sigmoid.

Sigmoid Sigmoid is the classical when it comes to activation functions in a NN. The sigmoid activation function is described in Equation 2.5, x is the input and the function restricts it to a value between 0 and 1. During backpropagation, utilizing the chain rule, the derivative of the activation function is calculated. Backpropagation and the chain rule will be further discussed in Section 2.5.2. The sigmoid function's derivative is between 0 and 0.25, implying that a NN that employs only the sigmoid activation function is susceptible to vanishing gradient descent.

$$\phi_{sigmoid} = \frac{1}{1 + e^{-x}} \tag{2.5}$$

Rectified Linear Units Rectified linear units (ReLU) is a non-linear function suitable for training deep learning models. Because ReLU is resistant to vanishing gradient, in comparison to the sigmoid function, which manifests when the derivative of the activation function is extremely small, thus ReLU is an activation function that is often used in NN. The ReLU function is described in Equation 2.6, and the derivative is always one for positive numbers. Therefore, the vanishing gradient problem is solved [34].

$$\phi_{ReLU} = max(0, z) \tag{2.6}$$

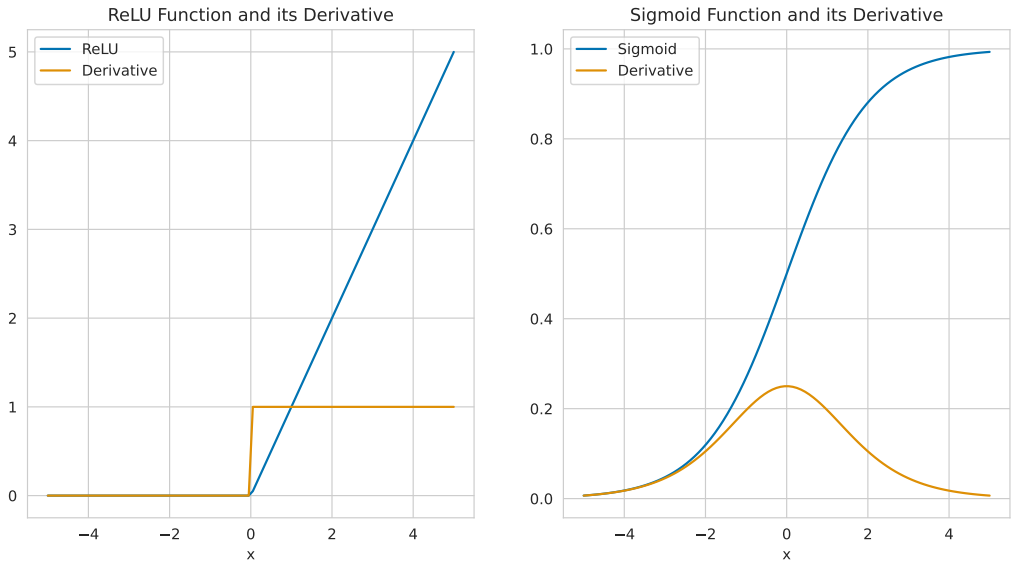


Figure 2.5.3: *Sigmoid and ReLU activation functions and their corresponding derivatives.*

Convolutional Neural Network

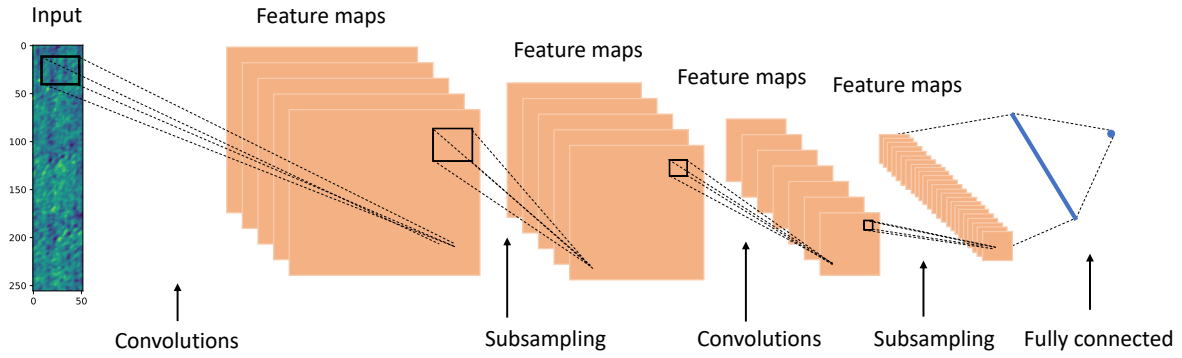


Figure 2.5.4: *Illustration of a CNN architecture including convolutional layers, subsampling layers, and a fully connected layer for prediction, where the convolutional layers extract high-level features such as edges or patterns, and the subsampling layers downsample the feature maps. The kernel and the feature map undergo element-wise multiplication, and the results are summed with the bias term to generate a one-depth channel feature map. This process is repeated in multiple layers to learn hierarchical representations of the input data. The final layers of the CNN consist of a dense layer with one node.*

Pixels in digital images are arranged in a binary representation of a grid-like structure, each of which indicates the brightness and color it should have. A network with expertise in handling such data representation is called a CNN. A CNN is a feed-forward NN generally used to analyze visual images by processing data with a grid-like topology. The earlier layers of a CNN are considered feature extraction since they can extract useful features from raw data [34]. Figure 2.5.4 depicts a conventional CNN model comprising convolutional and alternate subsampling layers. The final layers of the network are fully connected layers, enabling it to generate predictions as output. The convolutional layer performs a discrete convolution, also called convolution, which can be considered a kernel that slides over the image with a particular stride. The kernel and the feature map undergo element-wise multiplication, followed by summation. All the results are summed with the bias to give a squashed one-depth channel feature map. The convolution operation in a CNN is designed to recognize and extract low-level features, such as edges, corners, textures, color, and gradient orientation, from the input image. These low-level features are captured in the early layers of the CNN through convolutional and pooling layers. They serve as essential visual attributes that can be directly extracted from the input image. These low-level features can then be combined in the deeper layers of the network to form higher-level features, such as object boundaries, object parts, or object shapes, which are more complex and abstract representations learned by the network. These top-level characteristics can create more intricate shapes, such as the general curves of an object. In order to prevent feature extraction from solely relying on the center pixels of an image, padding is a technique that can be employed.

As illustrated in Figure 2.5.4, a CNN calculates feature maps by processing local patches from the previous feature maps. Each element in the feature maps is computed from a corresponding local patch in the previous feature maps.

Padding is a method that can shrink, expand, or keep the same shape for the input and output images. In general, there are three types of padding; *valid*, *same*, and *full* padding. *Valid* padding refers to the case when there is no padding, $p = 0$, and the output image shrinks. *Same* padding is usually used to ensure that the input and output are of the same shape. In this case, the filter size and the requirement that the input and output sizes be the same are used to calculate the padding parameter, p. *Full* padding is the case where $p = kernel\_size - 1$ increases the dimensions of the output; thus, it is rarely used in CNN architectures. The most commonly utilized padding in CNN is *same* padding, which is the case for this model's thesis [34].

Pooling Layers

In practice, it is advised to maintain spatial size using *same* padding for convolutional layers and reduce spatial size using pooling layers as an alternative. The pooling layers provide an approach to downsample the feature maps by summarizing the presence of features in patches of the feature map. The pooling layers in a CNN model are typically applied in two different forms; *max-pooling* and *mean-pooling*. *Max-pooling* keeps the maximum value inside the chosen pooling size, and *mean-pooling* computes the average number out of all the numbers inside the pooling size [34].

Fully Connected Layers

Standard CNN architectures are often used for classification tasks because they include fully connected layers in the output layer, also known as dense layers, which determine the size and format of the network's output. In classification tasks, the last dense layer typically has one output node per class, allowing the network to assign a probability to each class. For example, if a CNN is used to identify three different types of wheat, the last dense layer would have three output nodes, one for each type. The network would then assign a probability to each output node indicating the likelihood that the input belongs to that class.

For regression problems, a fully connected layer is typically placed before the regression layer at the network's end. In this case, the network outputs a numerical value instead of a probability, which represents the predicted continuous variable for the input. Overall, CNNs can be adapted to both classification and regression tasks, with their architecture and dense layers tailored to the specific problem at hand [34].

2.5.3 Regression

Conducting a regression analysis to forecast the grain yield for each crop in the dataset is crucial. This type of analysis is aimed at identifying a correlation between multiple predictor variables, referred to as features in machine learning, and a continuous response variable, also known as the target variable [34].

Mean Squared Error

Mean squared error (MSE) is a loss function commonly used in statistical models to evaluate accuracy. It measures the average of the squared differences between predicted and actual values, as shown in Equation 2.7. When the model's prediction is perfect, and there is no error, the MSE value will be zero. However, as the model's error increases, the MSE value also increases accordingly. As depicted in Figure 2.5.5, a more accurate model will have data points closer to the regression line.

$$MSE = \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n} \tag{2.7}$$

Here, y_i is the i^{th} observed value, $\hat{y}_i$ is the corresponding predicted value, and n is the number of observations [41].

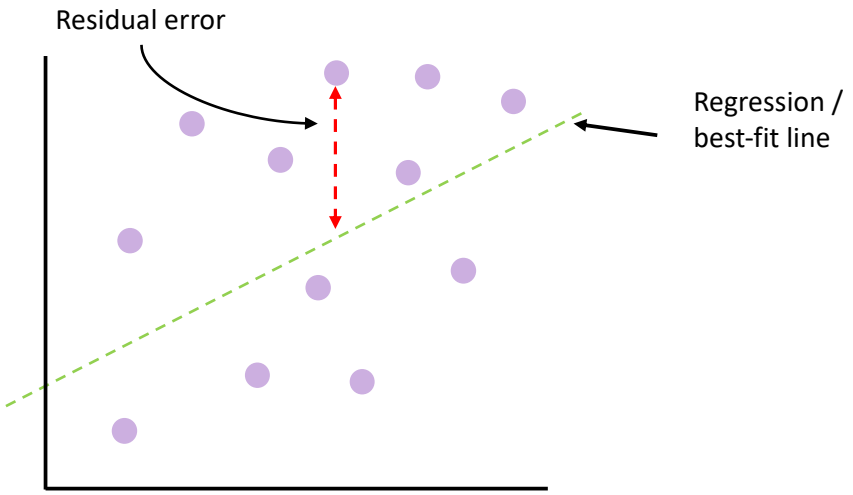


Figure 2.5.5: *The figure showcases the residuals computed for the MSE in a graphical form. The green diagonal line in the figure represents the regression line; here, 100% correct predictions are. The purple circles represent the residuals. Residuals are the differences between the actual values and the corresponding predicted values obtained from a model. In the case of MSE, a commonly used loss function for regression tasks, the residuals are squared and averaged to calculate the error.*

Dropout

Choosing the optimal size of a NN can be challenging, involving decisions such as the number of layers and the size of the weight matrix. If the network is not complex enough, it may not be able to capture the complexity of the data and may result in underfitting. On the other hand, if the network is too extensive and complex, it may overfit the training data, leading to good performance on training data but poor performance on test data.

One approach to address this problem is to build a network with a slightly larger capacity than needed to perform well on the training data and then use dropout for regularization to prevent overfitting. Dropout works by randomly dropping out a fraction of hidden units during each iteration of the training phase, with a probability of $p_{keep} = 1 - p_{drop}$. When dropping out random hidden units, the weights associated with the remaining units are rescaled to account for the dropped units [34].

The effect of this random dropout is that it forces the network to learn redundant representations of the data. The network cannot rely heavily on any specific activation of a set of hidden units, as they may be dropped during different parts of the training process. The dropout process forces the network to learn more general and robust patterns from the data [34].

Batches and Epochs

Deep learning models are trained by repeatedly feeding the ANN with batches of training data. By splitting a large dataset into smaller batches, it becomes possible to work with larger datasets by training the model incrementally on each batch. Convergence occurs when the model's performance stabilizes and stops improving significantly. Smaller batches can lead to faster convergence. However, in some cases, larger batches can help the model to learn more representative statistics about the data distribution and perform better on unseen data. The process where batches are fed into the network continues until all the training data has been used, constituting one epoch of training. After each batch, the model's predictions are compared to the expected output, and the algorithm is updated to improve the model's performance. The iterative process of forward and backward propagation helps the model improve its accuracy and ability to make accurate predictions [34].

2.5.4 Optimizer strategies

Optimizers are algorithms or techniques used to minimize the error function, also known as the loss function, in machine learning and NN. They operate on the learnable parameters of the model, such as weights and biases, to determine how to adjust them and the learning rate to minimize the losses. These functions play a crucial role in optimizing the

performance of a NN by finding the best set of parameters that result in accurate predictions. Various optimization algorithms are available, each with strengths and suitability for different problems. In this thesis, two optimization algorithms, Adaptive Moment Estimation (Adam) and Root Mean Square Propagation (RMSProp), have been tested on a regression problem to determine their effectiveness in improving model performance.

Adaptive Gradient Algorithm

Adaptive Gradient Algorithm (AdaGrad) adjusts the learning rate η for each parameter θ_i based on its past gradients, shown in 2.8. The adjustment is made at each iteration t.

$$\theta_{t+1,i} = \theta_{t,i} - \frac{\eta}{\sqrt{G_{t,i} + \epsilon} \cdot g_{t,i}} \tag{2.8}$$

Here, $g_{t,i}$ is the gradient of the loss function concerning the parameter θ_i at iteration t, $G_{t,i}$ is a diagonal matrix where each diagonal element i is the sum of the squares of the gradient concerning θ, and ϵ is added to avoid division by zero [42].

Root Mean Squared Propagation

RMSProp restricts the fluctuation in the vertical direction with respect to the gradient. Hence, the algorithm can increase the learning rate and take more significant steps in the horizontal direction of the gradient leading to speed up convergence. The RMSProp calculations are shown in the equations below.

$$v_{dw} = \beta \cdot v_{dw} + (1 - \beta) \cdot dw^2 \tag{2.9}$$

$$v_{db} = \beta \cdot v_{dw} + (1 - \beta) \cdot db^2 \tag{2.10}$$

$$W = W - \alpha \cdot \frac{dw}{\sqrt{v_{dw}} + \epsilon} \tag{2.11}$$

$$b = b - \alpha \cdot \frac{db}{\sqrt{v_{db}} + \epsilon} \tag{2.12}$$

The first two Equations (2.9 and 2.10) compute the exponentially weighted moving average of the squared gradients for the weights (v_{dw}) and biases (v_{db}), respectively. Here, β is a hyperparameter that controls the decay rate of the moving average and is typically set

to 0.9. *dw* and *db* are the gradients of the loss function with respect to the weights and biases, respectively.

The following two Equations (2.11 and 2.12) update the weights and biases using the computed moving averages of the squared gradients. Here, W and b represent the weight and bias matrices, respectively, while α is the learning rate, and ϵ is a small constant to avoid division by zero.

The denominator term in each of these equations is known as the RMS (root mean square) of the gradients, which normalizes the updates by a running estimate of the variance of the gradients. Thus, preventing the updates from becoming too large or too small, especially in the presence of sparse gradients or noisy data [43].

Adaptive Moment Estimation

Adaptive moment estimation (Adam) is an optimizer that combines the AdaGrad with RMSProp extensions of stochastic gradient descent (SGD). SGD approximates the cost from a single training sample (online learning) or a small subset of training examples (mini-batch learning). It makes learning faster because of the more frequent weight updates than gradient descent and is suitable for solving problems with non-linear activation functions. Since non-linear functions do not have a convex cost function, SGD's noisy nature helps escape local minima. The Adam optimizer provides the advantages of both AdaGrad and RMSProp. It employs the average of the second moments of the gradients and the average of the first moments, which is how RMSProp adjusts the parameter learning rates (the uncentered variance). The parameters β_1 and β_2 determine the decay rates of these moving averages, and the method creates an exponential moving average of the gradient and the squared gradient [44].

Chapter 3

Methods

This section presents a detailed account of the research process, from the data collection to analysis and interpretation. In this thesis, the methodology section focuses on applying machine learning models to multispectral imaging data.

The first part of this section describes the datasets used in the study and the process of selecting the appropriate data for analysis, including a description of the data pre-processing steps to ensure the data is clean, normalized, and properly formatted for the chosen machine learning models. Next, the section presents the architecture of the machine learning models used in the study, including a detailed explanation of the NN models, loss functions, and optimization algorithms. Following this, the section describes the validation and evaluation of the models, including the metrics used to assess the error and generalization of the models.

3.1 Image Acquisition

The images were captured with a DJI Phantom 4 multispectral (P4M) drone, which has an integrated camera, and the camera specifications are listed in Table 3.1.1. The multispectral images were obtained between May and August. P4M UAV is specifically designed for agricultural use, featuring a fully integrated narrow-band multispectral camera, which provides more detailed and accurate information by capturing precise measurements of specific wavelengths, which allows for early detection of health issues [45].

Table 3.1.1: *P4M Camera Specifications*

Specification	**Details**
Bands	Wavelengths
Red	650 ± 16nm
Green	560 ± 16nm
Blue	450 ± 16nm
NIR	840 ± 26nm
Red edge	730 ± 16nm
Focal length	5.74mm
Sensor size	4.87mm × 3.96mm
Image size	1600 × 1300
Sunshine sensor	Yes
Calibration panel	SphereOptics, Diffuse Reflectance Target-53%R
UAV platform	Has its own platform

3.1.1 Time Series for Image Acquisition

Table 3.1.2: *The table displays the dates when the multispectral images were captured, organized by month. It appears that the images were taken at random times, possibly due to weather conditions or other limitations that made it challenging to capture images consistently.*

date-times				
May	June	July	August	September
16.05.2022	28.06.2022	21.07.2022	01.08.2022	01.09.2022
20.05.2022		28.07.2022	05.08.2022	
31.05.2022			05.08.2022	
			30.08.2022	

Collecting multiple images at different times of the day, under varying lighting and weather conditions, can provide essential data for training a CNN model to forecast grain yield. However, the consistency of the images captured is crucial for determining the usefulness of a time series approach. Table 3.1.2 demonstrates that the MSI was captured inconsistently, making it difficult to determine if the time series approach is advantageous. The table reveals that only one day had two images captured, and the time intervals between the images were significantly different. This limited dataset may not fully capture the temporal dynamics of the crops, and the lack of a more uniform time series could adversely affect the model's accuracy. Nevertheless, capturing images multiple times a day allows for the analysis of the crops' spectral response under varying lighting and weather conditions, leading to a more comprehensive time series dataset, which could result in more accurate predictions and insights into crop growth and maturity dynamics.

3.2 Test Site

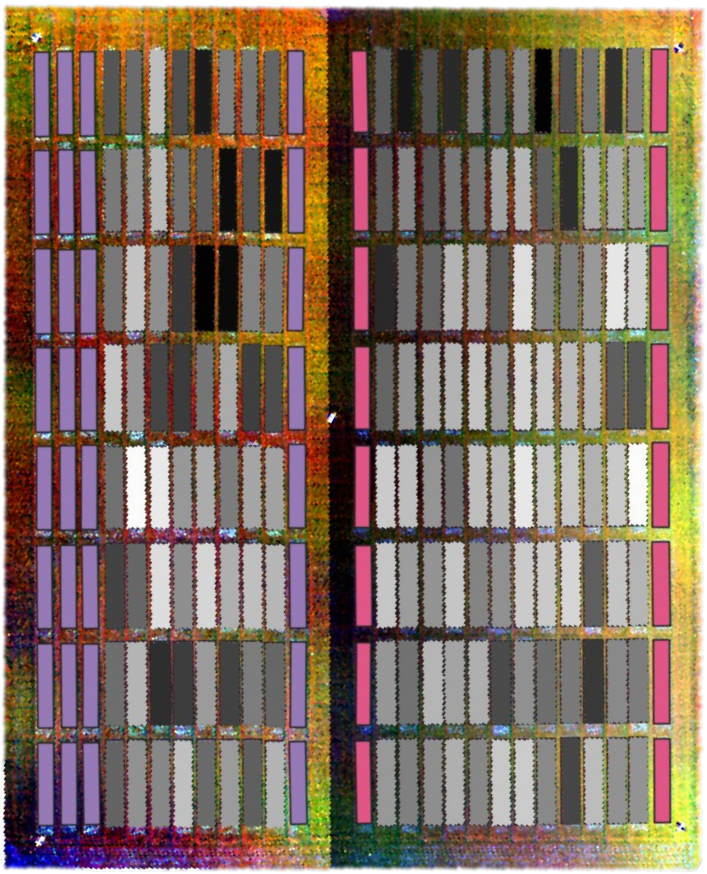


Figure 3.2.1: *The figure displays both the 22MLNOBAL (left side) and the 22MLROBOT (right side). Each plot is represented by a grayscale value indicating the measured grain yield. However, the pink and purple plots are not included in the dataset. The background of the figure represents the NIR radiation band captured by the multispectral camera.*

During the summer of 2022, multispectral images of wheat trials were captured at Vollebekk Research Station in Ås, southeastern Norway ($59°39'N, 10°45'E$), near the Norwegian University of Life Sciences (NMBU). The test site consisted of several fields with various wheat varieties. Two of these fields, 22MLNOBAL and 22MLROBOT, were used in this study. Both fields are shown in Figure 3.2.1 with different gray levels representing grain yield amounts. However, the pink and purple plots are not included in the dataset. 22MLROBOT had 24 historical wheat cultivars grown at two different fertilization levels of 75kg N/ha and 150kg N/ha, arranged in a split-plot design with two replicates, resulting in a total of 96 plots (1.5m x7.5m). 22MLROBOT was planted at the beginning of May 2022. As the plot's background reveals, the plots are extracted from the NIR band.

The 22MLNOBAL field is similar to the 22MLROBOT field in that both received the same fertilization rates of approximately 75kg N/ha, with half of the plots receiving this amount and the other half receiving 150kg N/ha. The 22MLNOBAL consists of 16

different cultivars; each plot was replicated twice, resulting in a total of 64 plots. Both fields are located adjacent to each other, and the image capturing was done simultaneously under the same weather and light conditions with the same imaging parameters and flight height (50m).

3.2.1 Grain Yield and Days to Heading (DH)

After allowing all the plants to reach full maturity, field experiments were conducted. The harvested grains were stored in netting bags until they reached a moisture content of 14%, at which point their weight was measured in grams per square meter and converted to tonnes per hectare (t/ha) to serve as ground truth for the images. In order to further aid the model in predicting the amount of grain yield, the heading day was manually measured and used in conjunction with the images. This additional information on DH provides a more comprehensive dataset for further analysis and modeling, potentially leading to more accurate predictions and a deeper understanding of crop growth and maturity dynamics.

3.3 Pre-Processing

Machine learning requires pre-processing, which is a crucial step. Various approaches were utilized for this thesis. Orthophoto of all fields has been created in the Phenocrop project using the Pix4D software (pix4d.com). For further details on image processing, please refer to [46] and [11]. Ensuring only plots were used for prediction, *rioxarray's rio.clip* extracted them using coordinates from the GeoJson file [47]. The *rio.clip* function is used to clip a raster dataset to a specific spatial extent, which means to subset the data only to include the area that falls within a specified geometry or bounding box, in this case, the plots with cultivars. The clipping process can be helpful when working with large raster datasets since it is possible to focus on smaller areas of interest and reduce the amount of data needed to process. The thesis dataset comprises 16 multispectral images, one for date-time point for each field, with five bands each (red, green, blue, NIR, and red-edge).

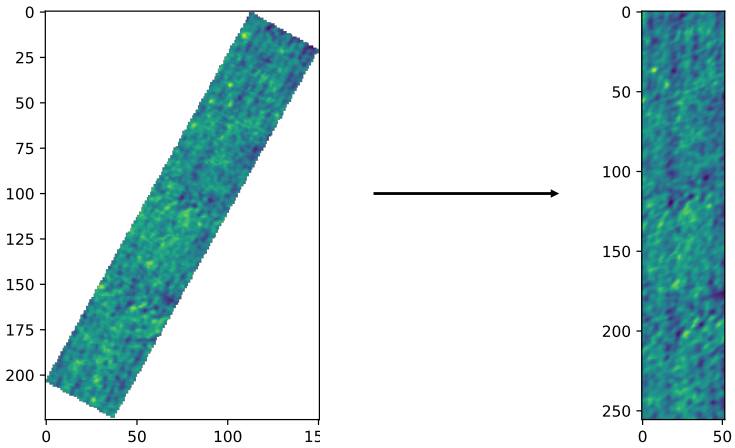


Figure 3.3.1: *Each image was rotated and cropped to focus only on the plots as part of the data preparation process.*

The orthophoto follows the image's diagonal, with each cropped image being one plot along the diagonal and only not a number (NaN) values in the background, presenting a challenge for the regression model, as it relies on measuring the distance between predicted and target values. When the target values are all NaN, the machine learning algorithm perceives an infinite gap between the predicted and target values, which makes it impossible to update the weights and bring the predicted values closer to the target values. Hence, the NaN values were replaced with zeros, which led to a biased model. Each image was rotated and cropped to focus solely on the plots, depicted in Figure 3.3.1, to solve the bias problem. Each image now has a shape of (256, 52, 5), and predicted values are obtained for each image. Images with their corresponding ground truth are saved in an HDF5 (Hierarchical Data Format version 5) file and used for training the model.

3.3.1 Normalization

Data normalization standardizes the raw data by converting them into a specific range using a linear transformation. There are various normalization techniques; in this thesis, the Min-Max Normalization is utilized. In order to ensure that the pixel values for each image are in the same range, Equation 3.1 was performed pixel-wise per channel on every sample before training and testing the model. The technique keeps a relationship among original data; it is a simple technique and can, if specified, fit the data in a pre-defined boundary. In the case of this thesis, no boundary is specified, meaning that the range will be between zero and one [48].

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{3.1}$$

where X is the original value, X_{min} is the minimum value of the dataset, X_{max} is the maximum value of the dataset, and X_{norm} is the normalized value.

3.3.2 Data Augmentation

Since the dataset for this thesis is small, data augmentation can help to predict more accurate results. Data augmentation is a method for increasing the variety of the training set through random (yet realistic) modifications such as image rotation [49]. In this thesis, every image in the dataset has undergone vertical and horizontal mirroring, thereby tripling the original amount of data.

3.3.3 Missing Values

The issue of missing values is common in real-world datasets used for machine learning. Such missing values can result from data corruption, failure to record data, or removal based on high residuals. One approach to dealing with missing values is to remove these data points, which can improve the overall fit of the model by reducing the influence of these outliers and decreasing the error of the predictions. However, it is crucial to consider the reasons for the high residuals and ensure that the data is not being improperly manipulated or biased by removing specific observations. It is also important to note that the dataset cannot contain NaN values. In order to prevent this, the mean grain yield value across all the other plots has been assigned to each plot with missing values to fill in the gaps.

3.4 Data Analysis

Table 3.4.1: *The table displays the dataset-model combinations with their respective Model IDs, which will be utilized throughout the thesis. It also comprises a short description of each model.*

Dataset-model combinations	**Model ID**	**Description**
SingleDateTimeRobot	Model 1	One date-time from the 22MLROBOT
AllDateTimeRobot	Model 2	All date-times from the 22MLROBOT
MixedData	Model 3	All date-times from both 22MLNOBAL and 22MLROBOT
SeparatedDataTestRobot	Model 4	All date-times for 22MLNOBAL are used for training, and All the date-times for 22MLROBOT are used for testing
DateTimeConcatenationRobot	Model 5	All date-times for 22MLROBOT are used; fertilization level is added as an extra feature

In Table 3.4.1, all the dataset-model combinations utilized for this thesis are listed with a given Model ID and a short description. For simplicity and ease of reference, the Model IDs will be used throughout all subsequent discussions and analyses of the models. Initially, the only interest is in the 22MLROBOT, but the data is limited since that field is small. Therefore, 22MLNOBAL is being utilized when training some of the models. In total, there are five datasets-model combinations; SingleDateTimeRobot, AllDateTimeRobot, MixedData, SeparatedDataTestRobot and DateTimeConcatenationRobot.

All the dataset-model pairs contain an additional feature called DH which is added after the CNN body. This feature informs the model about the number of days until or after the heading date for each plot. Model 5, however, includes an extra feature, the fertilization level for each plot, added at the same position.

3.4.1 Model 1

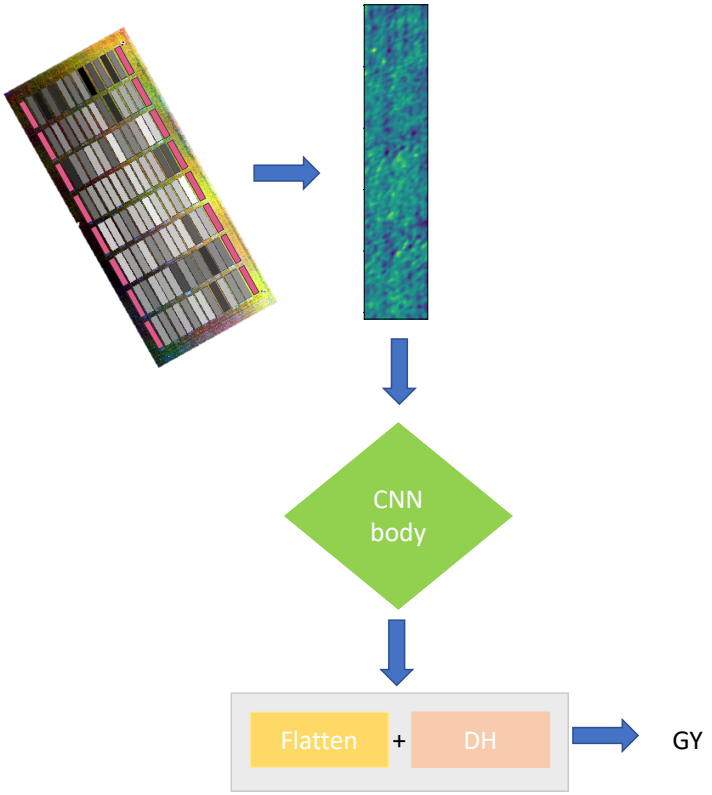


Figure 3.4.1: *Flowchart where 22MLROBOT datasets were cropped and propagated through the CNN. The body of the CNN consists of two convolutional layers and one max-pooling layer, repeated three times. The concatenation layer combines the flattened layer and the DH vector.*

As previously stated, the dataset for Model 1 comprises a single date-time, precisely the date 1.08.2022, which is approximately one month after the average heading date. The dataset consists of 96 original images, along with 96 images that were vertically rotated and 96 images that were horizontally rotated, resulting in a total of 288 images. The workflow of Model 1 is shown in 3.4.1.

3.4.2 Model 2

The dataset for Model 2 is derived from the 22MLROBOT dataset and includes all the date-times available in the original dataset, except the first and the two last dates. Each image in this dataset has a shape of (256, 52, 5), where the first two numbers represent the spatial axis, and the last number represents the number of bands. The original dataset contained eleven date-times, but only eight were extracted for training as the removed layers provided noise. Additionally, all images in this dataset are vertically and horizontally rotated, resulting in 2304 images.

3.4.3 Model 3

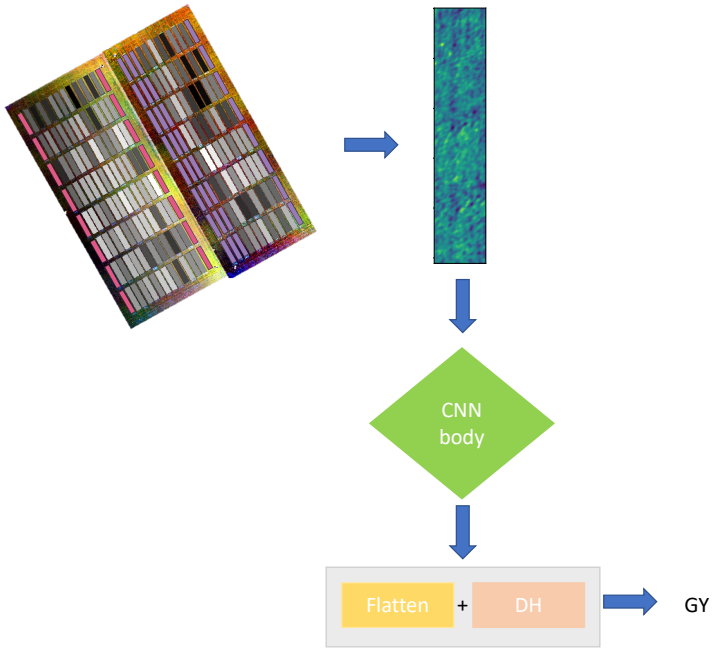


Figure 3.4.2: *Flowchart where 22MLROBOT and 22MLNOBAL datasets were cropped and propagated through the CNN. The body of the CNN consists of two convolutional layers and one max-pooling layer, repeated three times. The concatenation layer combines the flattened layer and the DH vector.*

The dataset for Model 3 is created by combining training data from the 22MLNOBAL and 22MLROBOT datasets. The 22MLROBOT dataset consists of 96 plots, while the 22MLNOBAL dataset comprises 64 plots. For each plot, the corresponding DH value is extended in a new list, and each image is also extended in a list, resulting in two lists with all the plots from all date-times and all the corresponding DH values. The list with DH is later concatenated with the flattened layer from the CNN body, as shown in Figure 3.4.2. Each image now has the resulting shape (256, 52, 5), where the first two axes represent spatial pixels, and the last axis represents the number of bands. Data augmentation techniques, such as vertical and horizontal rotation, are applied to all images in both datasets, resulting in a total of 3840 images.

3.4.4 Model 4

The dataset for Model 4 follows a similar approach as the dataset for Model 3 but with a different train-test-split strategy. In this case, only the 22MLNOBAL dataset is used for training, and the 22MLROBOT dataset is exclusively used for testing.

3.4.5 Model 5

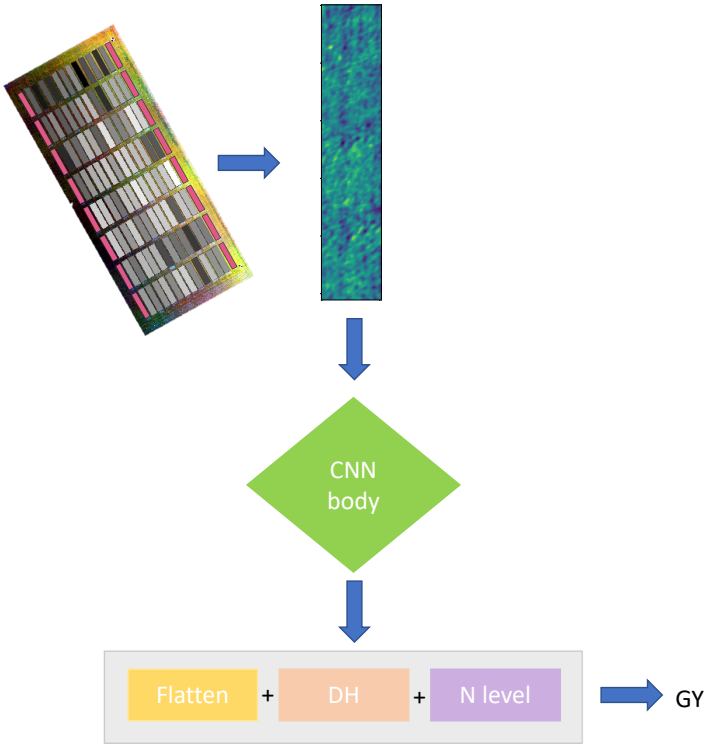


Figure 3.4.3: *Flowchart where 22MLROBOT datasets were cropped and propagated through the CNN. The body of the CNN consists of two convolutional layers and one max-pooling layer, repeated three times. The concatenation layer combines the flattened layer, the DH vector, and the fertilization level vector.*

Model 5 follows a similar approach to the Model 2 dataset in pre-processing the dataset. However, during the training phase, there is a difference in the CNN model architecture. In Model 5, the CNN model concatenates the flattened layer of the image data with two additional vectors: DH and another representing the fertilization level, shown in Figure 3.4.3. These two vectors are utilized as additional information to improve grain yield prediction.

3.4.6 Workflow

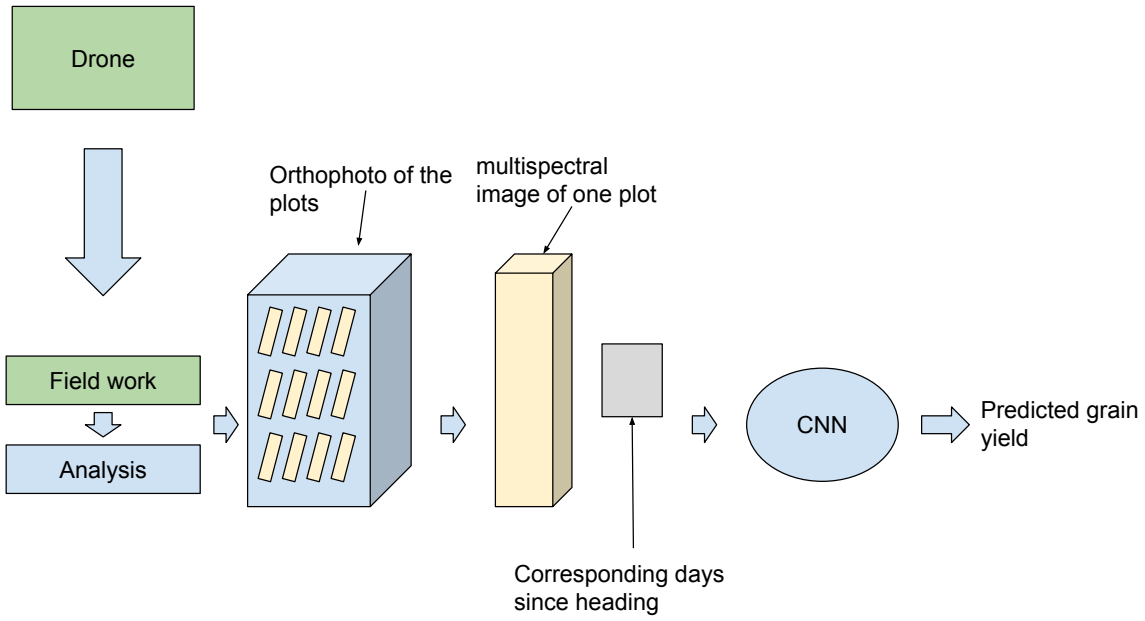


Figure 3.4.4: *Illustration of a flowchart of how the grain yield is predicted from start to end. Initially, the UAV to capture images. Next, the fieldwork is conducted, followed by the clipping of orthophotos into multiple plots and the creation of a corresponding vector of DH. Last, the CNN model is trained and used to make predictions.*

The whole workflow of yield prediction using UAV imagery is presented in Figure 3.4.4. Predicting grain yield begins with using the UAV to capture field images. Next, the fieldwork is conducted; as earlier mentioned, two different fields have been captured, 22MLNOBAL and 22MLROBOT; the orthophotos are clipped into plots and then followed by clipping orthophotos into multiple plots and saved in individual files named after their respective field, with a number from zero to eleven based on the date-time. All plots from a single day are consolidated into a single file. A corresponding vector of DH is created for each channel per image, and the CNN model is then trained and used to make predictions. All the various files for the 22MLROBOT field were examined, and the one with the lowest loss is discussed in this thesis.

3.5 Software, Hardware, and Memory

In this thesis, Python is mainly utilized for data processing and analysis. Multispectral data was visualized and explored using the open-source Geographic Information System GIS tool, Quantum Geographic Information System (QGIS). *TensorFlow* is the primary framework employed for machine learning. In order to manipulate images, the *Python Image Library (PIL)* is utilized. Both images and graphs are visualized and plotted using *Matplotlib* and *Seaborn*.

Table 3.5.1: *System specifications for the standard laptop and Google Colab Pro+.*

System	CPU	RAM	GPU name	GPU RAM
MacBook Pro	2.9 GHz	8.00 GB	-	-
Google Colab Pro+	2.2 GHz	25.7 GB	NVIDIA A100-SXM4	40 GB

Working with photos, particularly spectral images with more bands than standard RGB images, necessitates a significant amount of processing gear and Memory. Increasingly computationally intensive calculations need improved hardware and a powerful GPU. As a result, the NNs were trained on high-capacity external computers. The machine learning experimentation was done in Google's Colaboratory, an IPython Kernel in Google's cloud service. This virtual environment has the GPU as well as the majority of the Python 3 environments required for the study. Colaboratory's Pro+ version increased computation time on GPUs and RAM. Table 3.5.1 shows the system specifications for the standard laptop and Google Colab Pro +, which was utilized to pre-process the datasets and train the models.

Machine Learning Libraries and Platforms

Scikit-learn is an open-source Python library that can be used commercially for implementing machine learning models. The version employed in this thesis is scikit-learn 1.2.0 [50]. On the other hand, TensorFlow is a comprehensive machine learning platform, and the version used is TensorFlow 2.11, which allows building and training models using the high-level Keras API. A complete overview of programs and Python packages utilized in this thesis is included in Appendix B.

3.6 Model Specification

For this thesis, two different architectures for CNN models were utilized. The first model is a relatively regular CNN model with minor specifications. The second model has the same architecture as the first but also concatenates the fertilization level for each plot.

Both models start with alternating double convolutional layers and one max-pooling layer, which comprises the body of the CNN. After the last max-pooling layer, the feature maps are flattened. The first model concatenates the DH vector with the flattened feature maps, whereas the second model concatenates both the DH vector and fertilization level for each plot. In order to address the regression problem, MSE was chosen as the loss function, and there was no activation function in the output layer.

3.6.1 Parameter Optimization

Table 3.6.1: *Hyperparameters used consistently across all CNN models in this thesis, determined through simple model tuning*

Hyperparameters	**Details**
Number of filters	128
Kernel	3
Pooling size	2
Dropout rate	0.2
Epochs	1000
Loss function	Adam
Learning rate	0.00001
Batch size	16

Machine learning problems are varied and complex, and no single model can provide an effective solution to all of them. As a result, trial and error methods are often employed to determine the most suitable model and its parameters for a specific problem. In this study, a parameter grid dictionary consisting of various hyperparameters was run in a for-loop. All combinations were fitted, and the combination with the smallest loss value was identified as the optimal hyperparameter combination and printed. Table 3.6.1 presents the best hyperparameters obtained after tuning the model. The number of epochs was manually selected, and early stopping was applied.

3.6.2 Early Stopping

Early stopping is a technique used to prevent overfitting a NN during training. It works by monitoring the validation loss of the model during training and stopping the training process when the validation loss starts to increase. Preventing the model from continuing to learn from the training data and memorizing it would result in overfitting.

In Keras, early stopping can be implemented using the *EarlyStopping* callback function. The function requires various parameters, such as the metric to monitor, the minimum change required to consider it as an improvement, and the number of epochs to wait before stopping if no improvement is observed, called patience. In this particular thesis, the parameter settings for *EarlyStopping* were monitor='loss' and patience=50. Early

stopping is a simple and effective way to prevent overfitting and improve the generalization performance of NNs. Models can train more efficiently and with better results by utilizing early stopping in Keras [51].

3.6.3 Train, Validation, Test Split

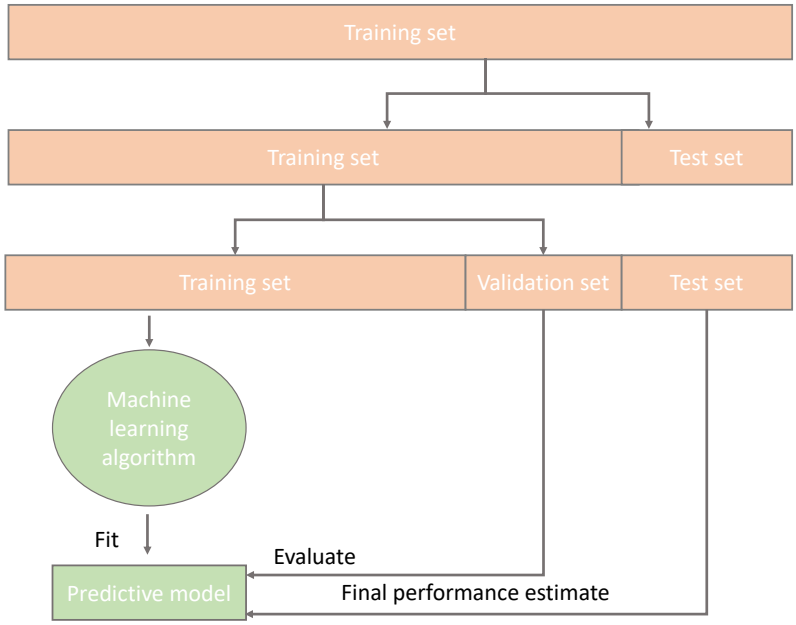


Figure 3.6.1: *Illustration of dividing the data into training and testing involves isolating the test set from the machine learning model until the time for making predictions.*

The data were divided into training and testing sets using the scikit-learn library's *train_test_split* function, with a test size of 0.2. Hence, 20% of the data was reserved for testing, and the remaining 80% was used for training the model, as shown in Table 3.6.2. The test set was kept isolated from the machine learning algorithm until the final predictions were made, as shown in Figure 3.6.1. Additionally, the training set was further divided into a training and validation set, with a ratio of 0.8 for training and 0.2 for validation, which was used to assess the model's performance during training. Once the model finished training, the test set was employed to predict grain yield.

3.6.4 Evaluation Metric

Section 2.5.3 presented the MSE metric as a loss function, which can also function as a metric. However, multiple evaluation metrics can be employed to assess the outcomes of a model after it has undergone training. This subsection will introduce some of those metrics.

Table 3.6.2: *The distribution of data for the training, validation, test sets, and the respective sample shapes.*

22MLROBOT	**Shape of X data**	**Shape of y data**
Train set	(1467, 256, 32, 5)	(1467, 256, 32, 1)
Validation set	(367, 256, 32, 5)	(367 , 256, 32, 1)
Test set	(461, 256, 32, 5)	(461, 256, 32, 1)
22MLNOBAL and 22MLROBOT	**Shape of X data**	**Shape of y data**
Train set	(2462, 256, 32, 5)	(2462, 256, 32, 1)
Validation set	(616, 256, 32, 5)	(616, 256, 32, 1)
Test set	(768, 256, 32, 5)	(768 , 256, 32, 1)

Mean Absolute Error

Mean absolute error (MAE) is a metric closely related to MSE. However, instead of squaring the differences between predicted and observed values, it takes the absolute value of the difference. Equation 3.2 calculates the average of the absolute differences between predicted and observed values, where y_i represents the i^{th} observed value, $\hat{y}_i$ represents the corresponding predicted value, and n represents the total number of observations [52].

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|(y_i - \hat{y}_i)| \tag{3.2}$$

Mean Absolute Percentage Error

A prominent metric for assessing a forecasting model's performance in machine learning and statistical modeling is mean absolute percentage error (MAPE). The average percentage difference between the expected and actual values is what is measured. A lower MAPE implies a better match between the model and the data. The MAPE is represented as a percentage. Taking the absolute difference between the anticipated and actual values, dividing it by the actual value, and averaging these percentages overall observations are the steps in the computation, as shown in Equation 3.3. The fact that MAPE is scale-independent makes it advantageous for assessing the effectiveness of models that predict various units of measurement.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\% \tag{3.3}$$

where y_i represents the i^{th} observed value, $\hat{y}_i$ represents the corresponding predicted value, and n represents the total number of observations [53].

Mean Bias Error

The Mean Bias Error (MBE) measures the difference between predicted ($\hat{y}_i$ and actual (y_i) values as stated in Equation 3.4. The closer the value is to zero, the more accurate the predictions are. The MBE can be positive, indicating that the model's predictions, on average, are higher than the actual values, or negative, indicating that the model's predictions, on average, are lower than the actual values [54].

$$MBE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i) \tag{3.4}$$

Coefficient of Determination

The coefficient of determination denoted R^2, is a statistical measure used in regression analysis to assess how well the independent variable explains the variance in the dependent variable. As shown in Equation 3.5, it ranges from zero to one, where higher values indicate a better fit. A value of one indicates a perfect prediction, while zero indicates that the model's predictions are no better than simply using the mean of the observed values. Negative values indicate that the model performs worse than simply using the mean value. However, R^2 alone does not determine the correctness of the model. It is essential to consider other variables in the statistical model, such as scatterplots, MBE, and residual plots, to draw accurate conclusions about the model's performance [55].

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{3.5}$$

where y_i is the i^{th} observed value, $\hat{y}_i$ is the corresponding predicted value, $\bar{y}$ is the mean of the observed values, and n is the number of observations.

Scatter Plots

Scatter plots are a visual aid that can compare predicted and actual values. They typically plot predicted values on one axis and actual values on another, allowing for easy visual assessment of how well they agree. By analyzing a scatter plot, it is possible to gain insights into the accuracy, precision, and consistency of a model's predictions. A high-quality model will usually exhibit most of its predicted values close to the diagonal line, which suggests a good level of agreement between predicted and actual values.

Residual Plots

A regression model's residual plot represents the imbalance between the observed and fitted response values. The ideal residual plot, the null residual plot, shows a uniform scattering of points that create a band of consistent width around the horizontal axis. However, it can be challenging to pinpoint specific issues using the residual plot alone in models with numerous terms. A non-null residual plot suggests that there may be problems with the model, but it may not necessarily provide insights into the exact nature of these issues [56].

Chapter 4

Results and Discussion

The Results section presents the findings from analyzing the collected data on grain yield prediction using a time series approach. The data includes multiple images captured over the crops during the growing season, which were analyzed to identify patterns and trends in crop growth. The Results section presents a comprehensive summary of the findings obtained through the analysis of the MSI. Additionally, it includes a brief discussion of the implications and significance of the results. The results are presented using tables and graphs, visually representing the patterns and trends observed in the data. Through a comprehensive analysis of the results, this section sheds light on the effectiveness of including agronomic data, such as DH, and field management data, such as fertilization level, for grain yield prediction.

4.1 Data Analysis

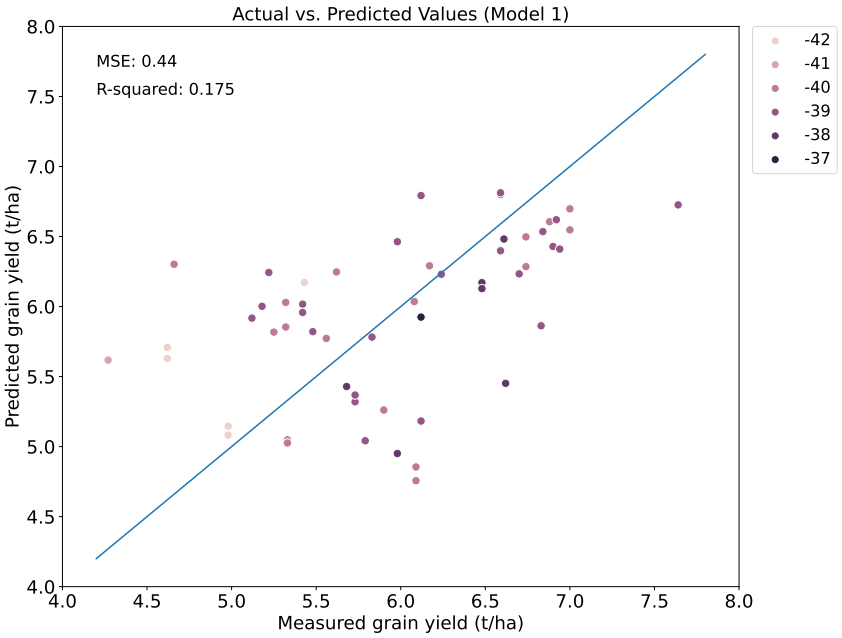
As Table 4.1.1 shows, the MBE values are all close to zero, indicating that the model's average performance is unbiased. However, the MBE for Model 1, Model 3, and Model 4 is negative, suggesting that the model may consistently predict lower values than the actual values. For Model 2 and Model 5, the model consistently predicts higher values than the actual values.

Table 4.1.1: *The outcomes for each model-dataset combination after forecasting the quantity of grain yield is given by the R^2, MBE, MSE, and MAPE. The best result is highlighted in bold.*

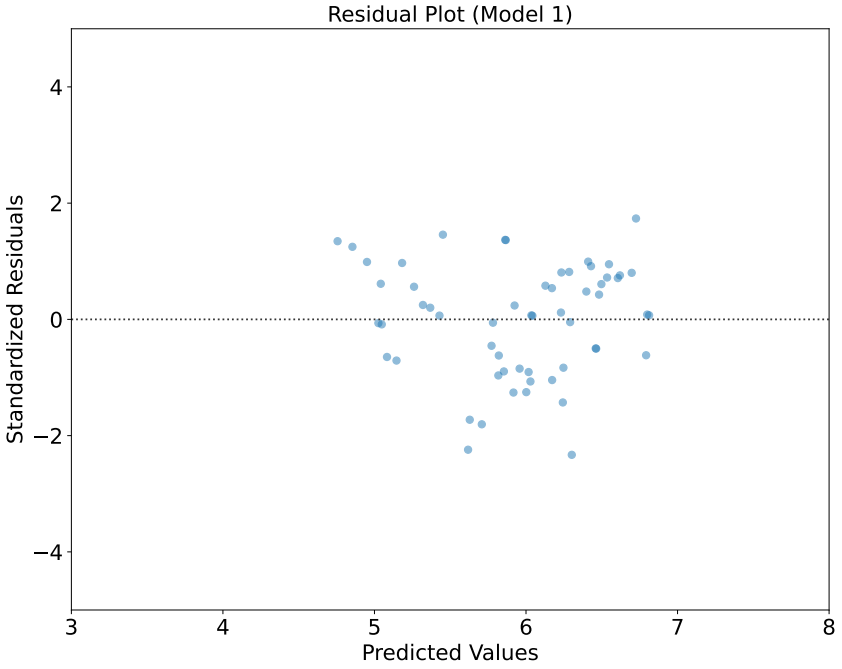
Model ID	R^2	MBE	MSE	MAPE
Model 1	0.175	-0.028	0.44	9.57
Model 2	0.851	**0.027**	0.09	3.62
Model 3	0.776	-0.050	0.15	4.45
Model 4	-0.209	-0.205	0.68	11.17
Model 5	**0.885**	-0.056	**0.06**	**3.01**

The MSE is a commonly used performance metric in regression tasks, measuring the average squared difference between predicted and actual values. The lower the MSE, the better the model's performance, as it indicates more minor errors in the predicted values compared to the actual values. The results of the MSE analysis provide insights into the effectiveness of the different dataset-model combinations and contribute to the overall evaluation of the model's performance in predicting grain yield. From Table 4.1.1, the smallest MSE was when the 22MLROBOT data with concatenation for both DH and fertilization level were included. The range for grain yield goes from around four t/ha to eight t/ha. As previously noted in Section 3.6.4, the MAPE is a metric that evaluates the average percentage difference between predicted and actual values. Referring to Table 4.1.1, it is observed that the MAPE values for Model 5, Model 3, and Model 2 are relatively low. However, for Model 1 and Model 4, the MAPE scores are significantly higher.

4.1.1 Results for Model 1



(a)



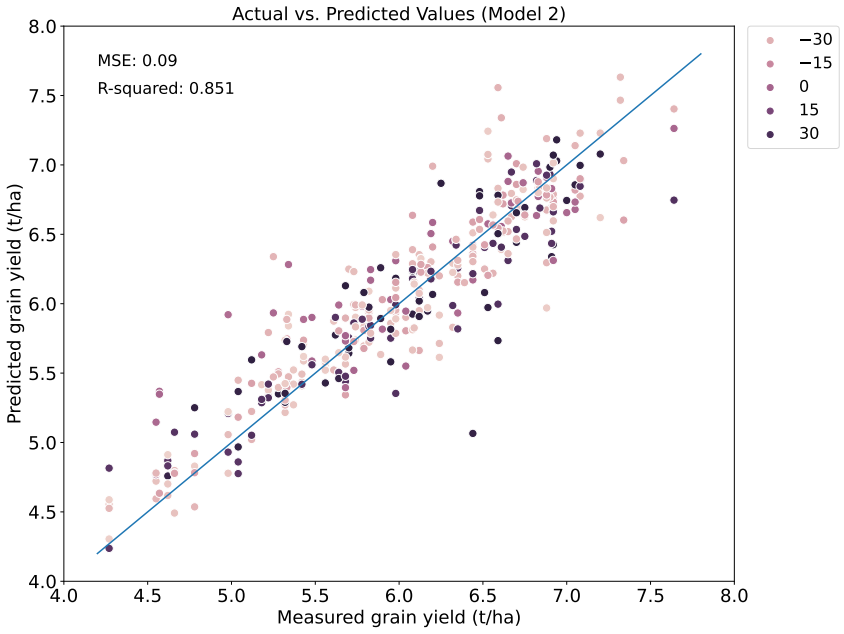
(b)

Figure 4.1.1: *(a) Scatterplot for Model 1, the grain yield is expressed in t/ha. The colors indicate how many days between the image being captured and the heading date. A Negative value indicates days after heading, and a positive value indicates days before heading. (b) Residual plot displaying the prediction error on the vertical axis against the predicted values on the horizontal axis.*

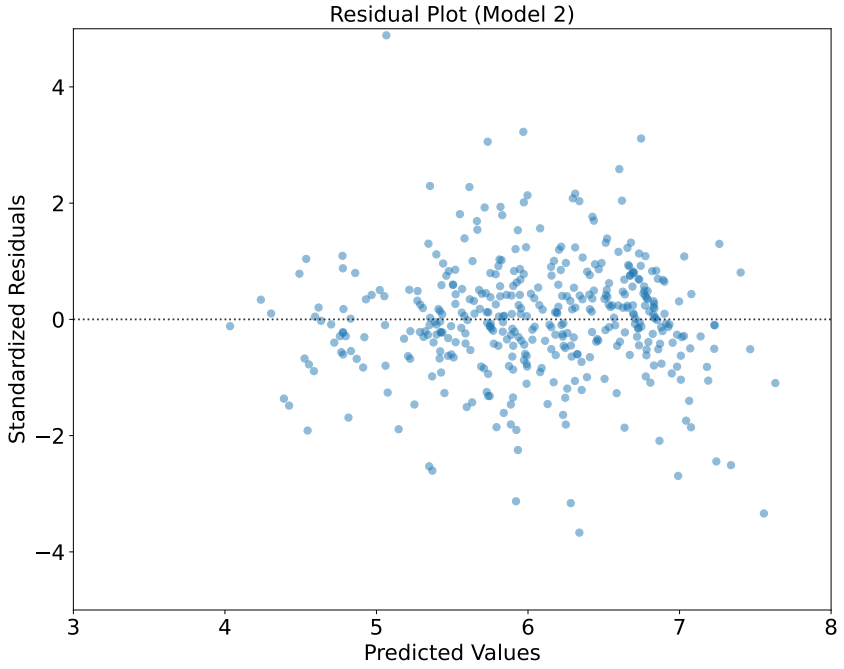
The scatter plot for Model 1 is presented in Figure 4.1.1a. Most of the points are not following the identity line (blue line), and most of the predictions are lower than the actual value of grain yield, matching the MBE from Table 4.1.1 well, since it is negative, which indicates underprediction. The MBE is not significantly high, which means that the overall predictions are not biased. The R^2 value, also presented in Table 4.1.1, is poor, indicating that the dataset does not fit the model well. An R^2 value of 0.175 means that only 18% of the variability in the dependent variable is explained by the response variable. A low R^2 value means that only a tiny percentage of the variation in the dependent variable is accounted for by the model's independent variable(s), and the remaining variation is attributed to other factors not included in the model. A low R^2 score is expected from a small dataset where the regression model may not have enough data to accurately capture the variability in the outcome variable. A small dataset can result in a poor model fit, which can be reflected in the scatter plot. The labels on the right-hand side of the model indicate the number of days between the heading day and the image capture. The negative values imply that the images were taken after the heading date. The labels highlight minor variations in the heading day for each plot since Model 1 includes only one date-time, resulting in all points having the exact date for capture but different heading days for each plot.

Figure 4.1.1b exhibits the prediction errors along the vertical axis plotted against the residuals along the horizontal axis. Ideally, the residuals should exhibit a random scatter around the horizontal line at y=0, indicating that the model's predictions are unbiased and possess consistent variance. Although the plot does not show a distinct cluster, the residuals are widely dispersed. The residuals should be around zero to indicate a good model, but that is not the case here; most of the points are inside two standard deviations (SDs) from zero. It is hard to analyze results with so few points; when the sample size is small, the residual plot may appear more scattered due to the limited data. In this case, it can be challenging to determine whether the scattered residuals are due to model misspecification or simply due to the small sample size.

4.1.2 Results for Model 2



(a)



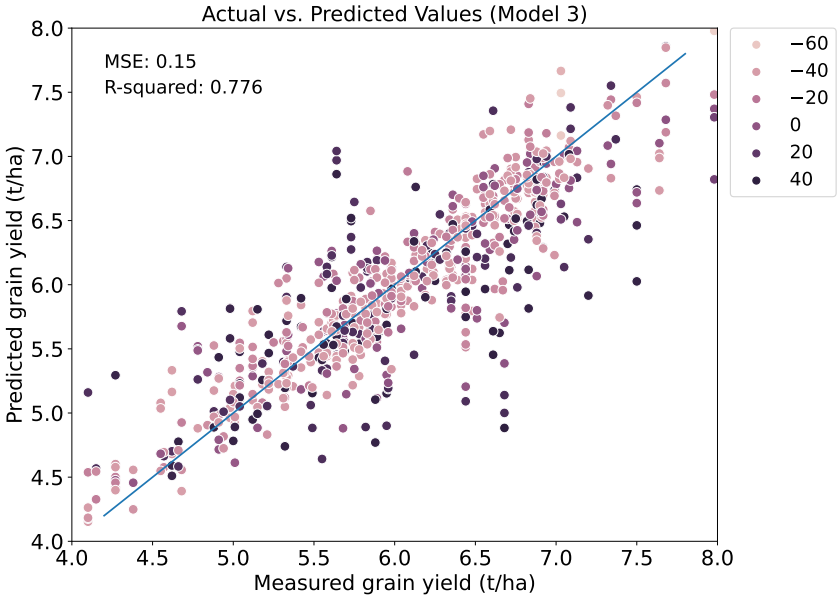
(b)

Figure 4.1.2: *(a) Scatterplot for Model 2, the grain yield is expressed in t/ha. The colors indicate how many days between the image being captured and the heading date. A Negative value indicates days after heading, and a positive value indicates days before heading. (b) Residual plot displaying how many SDs from 0 on the vertical axis against the predicted values on the horizontal axis.*

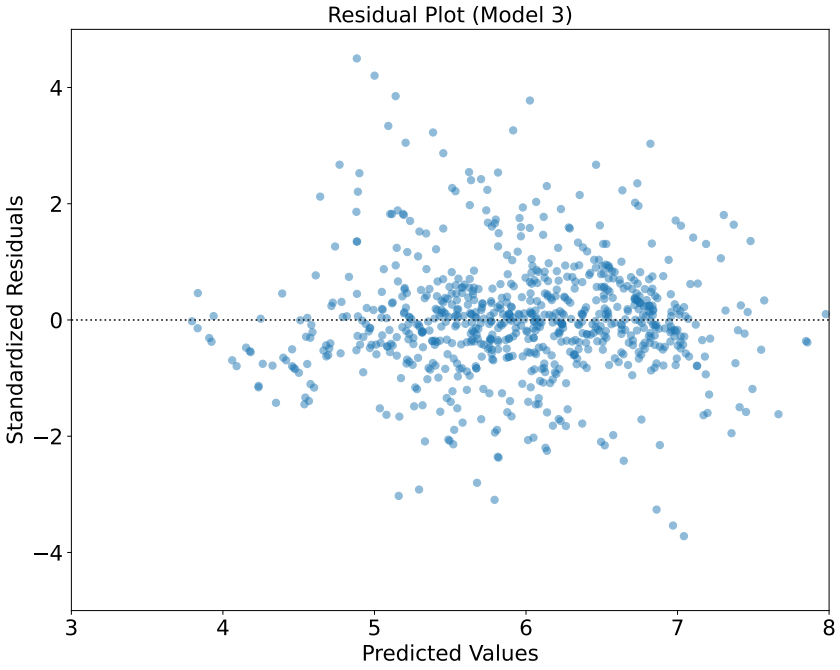
The scatter plot in Figure 4.1.2a displays a positive correlation between the predicted and actual values, with most points clustering around the identity line. The colors represent the time between the image capture and the heading day of the crops. The predicted values are slightly higher than the actual values, as indicated by the low positive value of the MBE, 0.027, from Table 4.1.1. Here the days between the image capture and the heading date are more spread out, with a range from -30 to 30, approximately one month before and after the day of heading.

Figure 4.1.2b shows the residual plot, which supports the assumption from the scatter plot. The residuals are randomly scattered around the vertical axis and do not follow any discernible pattern. The residuals are distributed across a range of three SDs from zero, indicating that they are widely dispersed. There are no visible patterns, such as a curve or a trend. The R^2 value from Table 4.1.1 is 0.851, indicating a good fit between the model and the data, supporting the earlier assumptions.

4.1.3 Results for Model 3



(a)



(b)

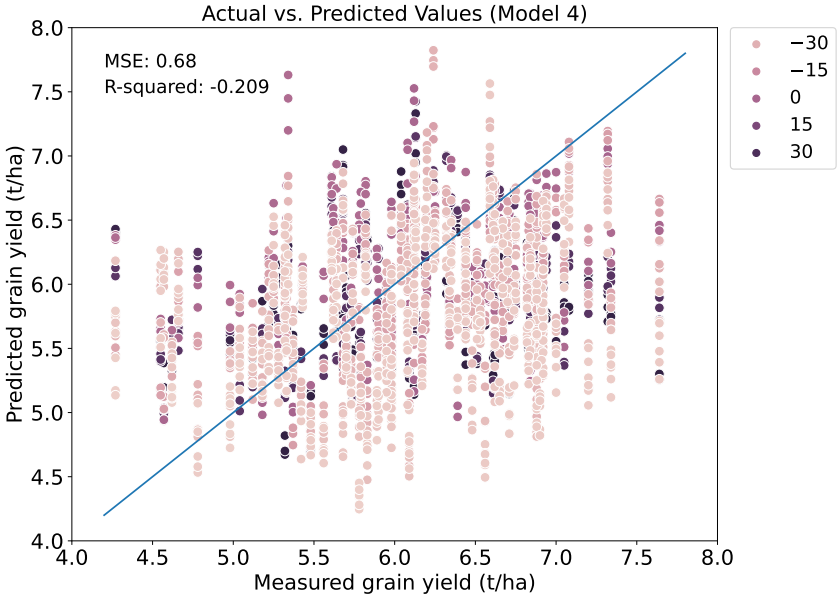
Figure 4.1.3: *(a) Scatterplot for Model 3, the grain yield is expressed in t/ha. The colors indicate how many days between the image being captured and the heading date. A Negative value indicates days after heading, and a positive value indicates days before heading. (b). Residual plot displaying the prediction error on the vertical axis against the predicted values on the horizontal axis.*

The scatterplot shown in Figure 4.1.3a represents Model 3. After analyzing the plot, it is evident that most of the data points are clustered around the identity line, which

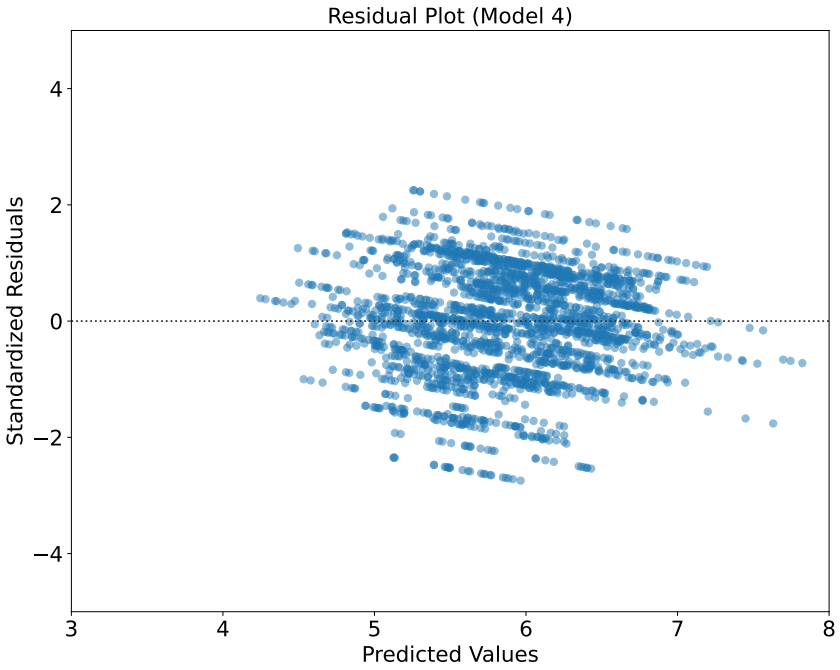
suggests a positive correlation between the predicted and measured grain yield. This finding indicates that the prediction model is effective and that the predicted grain yield is related to the measured grain yield. However, there are a few outliers in the data, and as the measured grain yield increases, the model's prediction becomes more challenging. Table 4.1.1 shows a relatively low MSE value of 0.15, indicating that the predicted values are close to the actual values with minor prediction errors.

Figure 4.1.3b presents a residual plot for Model 3, revealing evenly distributed residuals around zero with no discernible trend or pattern, suggesting that the model's assumptions are met, and it is a good fit for the data with constant variance and no bias in the residuals. The results from Table 4.1.1 show that the MBE is -0.050, indicating a slight possibility of underpredictions but insignificant. The MBE and the good residual plot suggest low bias in the model. Model 3 shows an R^2 score of 0.776, indicating that about 78% of the response variable's variability is explained by the predictors. However, it should be noted that the interpretation of the R^2 score depends on the context of the data and research question. In this case, the model was trained and tested on fields with different cultivars, which can make prediction challenging and affect the R^2 score.

4.1.4 Results for Model 4



(a)



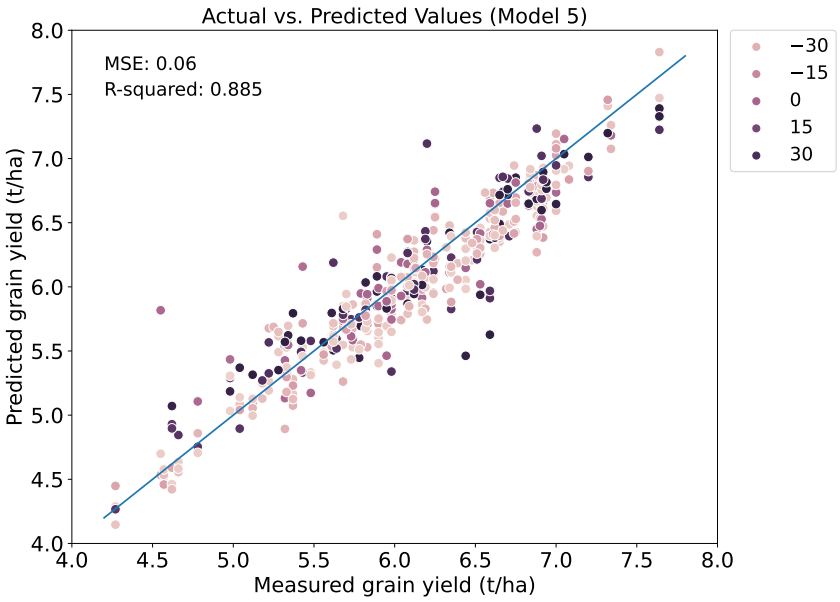
(b)

Figure 4.1.4: *(a) Scatterplot for Model 4, the grain yield is expressed in t/ha. The colors indicate how many days between the image being captured and the heading date. A Negative value indicates days after heading, and a positive value indicates days before heading. (b) Residual plot displaying the prediction error on the vertical axis against the predicted values on the horizontal axis.*

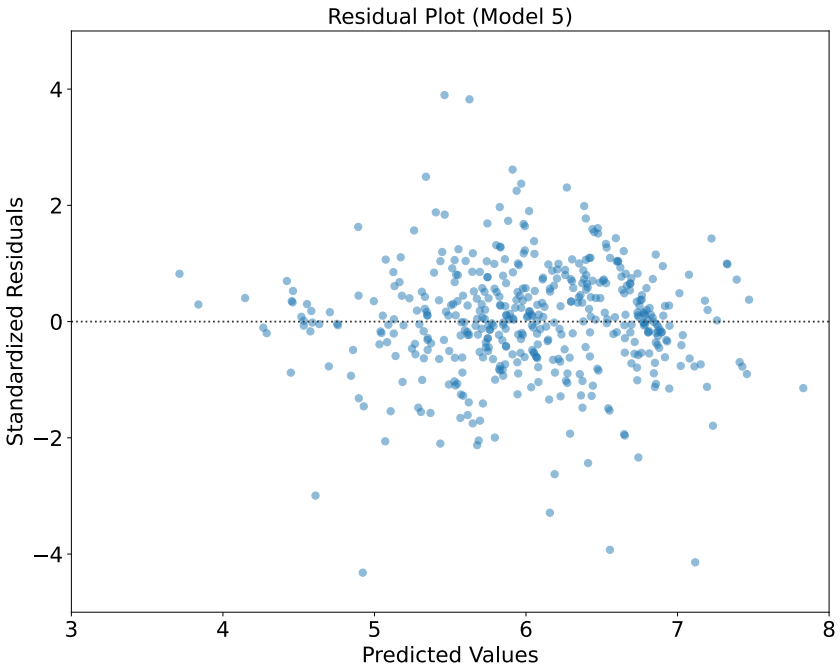
Figure 4.1.4a shows that the predicted values are widely scattered and do not follow the identity line, indicating that the model is not accurately capturing the relationship between the predictor and the outcome variable. Although the MSE value of 0.74 does not seem very high, if considering the range of actual grain yield values, the error is relatively large. The negative R^2 value further confirms that the model does not fit the data well. One possible explanation for the stripes in the scatter plot is the numerous repetitions of the same plot due to data augmentations, resulting in identical yield values. Using separate fields for training and prediction, each with different cultivars may have made it difficult for the model to identify the correct patterns, leading to overfitting.

The residual plot displayed in Figure 4.1.4b further reinforces the conclusion that the model does not adequately fit the data. The residuals are clustered around a range of two SD and are not centered around zero on the vertical axis, forming a dense cluster of points. The MBE value of -0.205 indicates that the model underpredicts the grain yield to a certain extent. Based on these observations, the model needs to be revised, or more data needs to be collected to enhance the predictions. In summary, all the evidence, including the scatterplot, residual plot, and MBE value, suggest that the model does not fit the data well. Additional research is required to determine the cause and to develop methods for improving the model's performance.

4.1.5 Results for Model 5



(a)



(b)

Figure 4.1.5: *(a) Scatterplot for Model 5, the grain yield is expressed in t/ha. The colors indicate how many days between the image being captured and the heading date. A Negative value indicates days after heading, and a positive value indicates days before heading. (b) Residual plot displaying the prediction error on the vertical axis against the predicted values on the horizontal axis.*

Figure 4.1.5a shows the predicted values compared to the actual values for grain yield, expressed in t/ha. The points in the scatterplot follow the identity line pretty well but slightly underpredict, in general, a bit. That matches the MBE showed in the same scatterplot, which has a value of -0.056.

According to the residual plot shown in Figure 4.1.5b, there are no discernible patterns or trends in the residuals, indicating that the model has no systematic errors or varying variances in its predictions. The points are widely dispersed, ranging up to four SD, which suggests that the model has captured most of the trends in the data.

Table 4.1.1 displays the results for MSE, MBE, and R^2 score. With a value of 0.06, the MSE is remarkably low, considering the grain yield range is between four to eight t/ha. The low MSE indicates that the model predicts a minimal amount of incorrect values. The R^2 value of 0.885 indicates that the model can account for roughly 89% of the variance.

4.1.6 MSE for All Date-Times

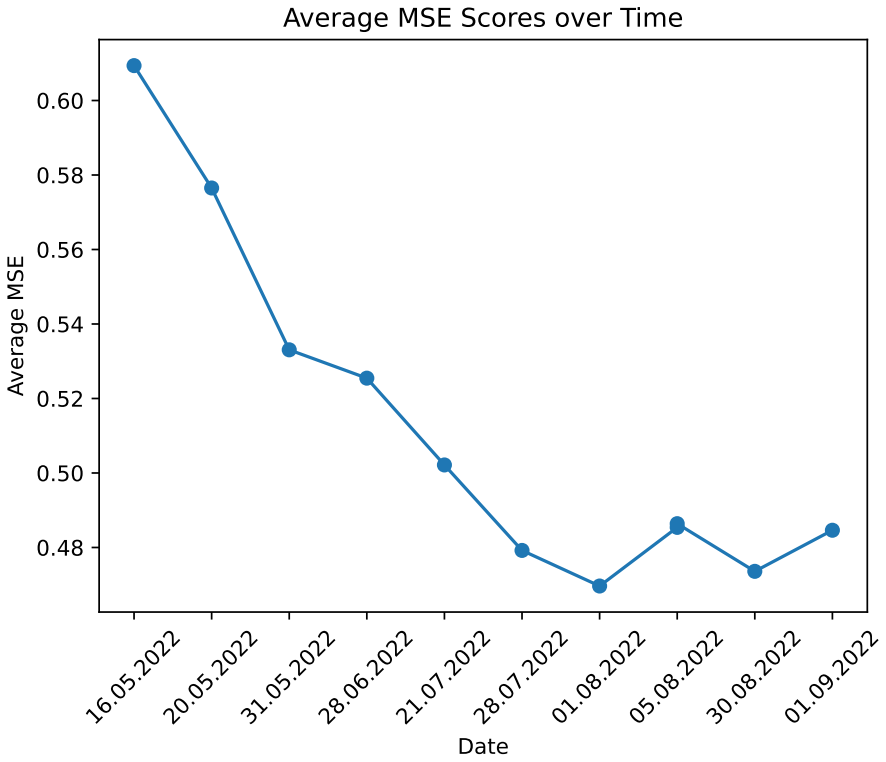


Figure 4.1.6: *Plot displaying when the smallest MSE score occurs for Model 2.*

The model's performance in predicting grain yield for each date-time was exclusively evaluated for the 22MLROBOT field by comparing the predicted values with the actual target values. The predictions were made without any agronomic data or field management data. The purpose of this analysis was to determine the accuracy of the model's predictions for each date-time and to identify the earliest date-time at which the predictions are reliable. The results are presented in Figure 4.1.6, which shows that the loss decreases steadily from the beginning, and the predictions become more accurate around the end of July. Additionally, the graph indicates that the smallest MSE is observed on 01.08.2022.

Chapter 5

General Discussion

This part of the thesis aims to analyze and interpret the findings obtained from the study on predicting grain yield using multispectral images of wheat trials. The results obtained from the experiments conducted at Vollebekk Research Station, Ås, Norway, shed light on the effectiveness of multispectral images for predicting grain yield. This section includes a comprehensive analysis of the findings, a discussion of the implications of the results, and highlights of the significance of the study in the broader context of crop cultivation, phenotyping, and climate variability. Furthermore, the study's limitations will be identified, and propose areas for future research.

5.1 Climate Conditions

Climate plays a significant role in cultivating various cultivars and characterizing plant traits. However, if there are considerable fluctuations in climatic conditions from year to year, developing a model that performs well in diverse conditions can be challenging. Temperature is a critical factor affecting crop growth and development, with warmer temperatures accelerating crop growth while cooler temperatures slowing it down. Extreme temperatures, such as heat waves, can also adversely affect crops, reducing yields and quality [57].

Water availability is another significant factor that can influence crop growth and development. Water is scarce in hot and dry climates, making it challenging to maintain adequate soil moisture levels, resulting in reduced crop growth, yield, and quality. In contrast, excessive rainfall in wet climates can lead to waterlogging, damaging crops. Climate variability also impacts soil conditions, with dry climates having low soil moisture levels leading to reduced nutrient availability. In contrast, wet climates can cause soil erosion and nutrient loss, reducing soil quality [58].

Climate variability can also influence pests and diseases, which affects crop growth and yield. Favorable pest and disease outbreak conditions are created in warm and humid

climates, while cold and dry climates can reduce their prevalence. All these factors can cause significant variations in grain yield from year to year, making it difficult to predict. Thus, data over several years and diverse environments are crucial to predict grain yield accurately [59].

5.2 Model Generalization

In this study, two fields, namely 22MLNOBAL and 22MLROBOT, were utilized for training the model. These fields were imaged during the same year and, thus, subjecting to similar climatic conditions, which may have resulted in a less generalizable model. It is challenging to determine if the trained model will perform well only on the fields used for testing and training or if it will generalize effectively to entirely different fields. Model 4 did not have good results when comparing the measured grain yield with the predicted, which shows that there is difficult to train on one field and then predict on another, even if the climate is the same since the cultivars are different. In the case of this study, the two fields had different kinds of cultivars, so there may be better results if there are two different fields with the same cultivars. The dataset for both fields where also small, so more data from various environments and over several years may improve the model. The influence of climate on cultivar performance and phenotyping is a crucial factor to consider, as variations in climate can significantly impact crop growth and yield, as mentioned in Section 5.1. Therefore, the transferability and generalizability of the model to other fields or climatic conditions need to be thoroughly examined and validated in future studies.

5.3 Comparison with Similar Studies

It is important to note that the approach used in this thesis differs from that used by Shafiee et al. [11] in their study for predicting grain yield in 2021. In their study, the plot values for each band and index were calculated as the median value of the pixels in each plot, and they did not consider the differences internally in the plot. In contrast, this thesis considers all the pixels in the plot and tries to predict the average grain yield for the whole plot.

Table 4.1.1 displays the findings that the most favorable outcome for this research was accomplished when the features fertilization and days to heading were incorporated. The R^2 score and MSE are 0.885 and 0.06, respectively. Although the R^2 score is marginally lower than Shafiee et al.'s study, the MSE in this research is lower. In Shafiee et al.'s study, their best model had an R^2 score of 0.92 and an MSE of 0.11. However, as the MSE and R^2 scores indicate different patterns, it suggests that the grain yield distribution may vary, and a direct comparison between the two scores may not be appropriate. Nonetheless,

the results show minimal differences overall, suggesting that the model's predictions are equally satisfactory.

The lower R^2 value in this study may be attributed to the different approach utilized compared to the previous study. Utilizing all the pixels in the prediction can result in more noise and variability in the data, which may affect the model's performance. However, this approach should enhance the model's robustness by incorporating more information about the field. In addition, differences in the R^2 values could be due to variations in the dataset, such as different crop, weather, and soil conditions, which can significantly influence the accuracy of the model's predictions. Furthermore, the imbalance in the R^2 values may also be due to differences in the model architecture, hyperparameters, pre-processing, and feature selection steps.

By utilizing the data for different fields in 2019 and 2020, Ijaz et al. [12] utilized various models for predicting grain yield. When comparing those outcomes with the outcomes of this thesis, it can be observed that the best model in this thesis has an R^2 score of 0.885, which is significantly higher than the optimal model in [12], which had an R^2 score of 0.725. The top model in that study was a Random Forest Regressor that used spectral bands, indices, and weather features and only utilized the median value.

In contrast to the model utilized by Ijaz et al. [12], the model in this thesis does not incorporate any weather data. Despite that, it performs remarkably well in predicting the grain yield. The model uses only two features, including fertilization and days to heading, to predict the grain yield. Despite the absence of weather data, the model achieves an R^2 score of 0.885 and an MSE of 0.06, indicating that it captures a significant portion of the variability in the data and makes accurate predictions. These results suggest that the model's simplicity does not hinder its predictive power. Furthermore, the promising performance of this basic model implies that by incorporating weather data, such as temperature, precipitation, and solar radiation, the model's predictive ability could be further enhanced.

As stated in Section 2.1.1, studying the reflected radiation beyond the visible spectrum may provide information about the health of plants and soil, rather than just relying on their visual appearance. In a study by Nevavuori et al. [60], a CNN was used to predict grain yield based on RGB images, and the best-performing model achieved a mean absolute error of 8.8% for within-field yield prediction. However, in this thesis, the best-performing model achieved a significantly better MAPE of 3.01%. Table 4.1.1 indicates that three out of the five model-dataset combinations evaluated in this thesis had a lower error than when using only RGB images.

5.4 Future Work

The results of this study show that utilizing the whole image for the prediction of grain yield is possible but need some modifications. However, several opportunities for the data foundation and model architecture for future research can give better results.

In terms of future directions, it may be valuable to extend this investigation by incorporating a more extensive and diverse dataset, as the current sample size is restricted to eight multispectral images from each field. Given that the pre-processing steps are relatively time-consuming for these multispectral images, a small dataset where utilized; therefore, it would be advantageous to acquire additional data to decrease the prediction error of the models. In retrospect, it would also have been beneficial to rotate the images horizontally and vertically during the thesis work to increase the amount of data by four times rather than three.

Furthermore, utilizing a multi-year dataset would be preferable to relying solely on the 2022 dataset used in this study. One potential avenue would be to train the model on data from prior years and subsequently apply it to forecast the grain yield of the current crop.

A compelling option to consider is to perform semantic segmentation. The model can classify each pixel individually and estimate the grain yield for each rather than for the entire plot. Such an approach would likely be more precise for farmers and offer even more practical insights for agriculture management.

Exploring the potential of hyperspectral images (HSI) for enhancing grain yield prediction models is a promising avenue for future research. HSI has the potential to provide valuable supplementary information due to its ability to capture more spectral bands than MSI. However, incorporating HSI data into grain yield prediction models may require careful consideration of various factors such as pre-processing, feature selection, and normalization techniques.

In addition, comparing the effectiveness of 2D-CNNs versus 3D-CNNs could be a valuable area of exploration. While 2D-CNNs are commonly used for image processing tasks, 3D-CNNs can simultaneously capture the spatial and spectral dimensions of the HSI data, potentially improving the accuracy of grain yield predictions. Other deep learning models, such as recurrent neural networks or graph neural networks, could also be considered for this task. Further investigation is needed to determine the most effective modeling approach for incorporating HSI data into grain yield prediction models and to identify the potential limitations and challenges associated with this approach.

Another approach that could have been done is to stack the date-time images so that the shapes of the images are (256, 52, 8, 5). When stacking the date-time may contribute to more accurate predictions. If the images are taken over a period of time, stacking them can provide temporal information that can be useful for predicting future events. It can help the model learn how the wheat crops evolve, making more accurate predictions for later years when the growing pattern is already learned. However, it is essential to keep in mind that stacking too many images can also introduce noise or make the data too complex for the model to learn from effectively. Additionally, it is essential to consider other factors that may affect crop growth, such as weather conditions, soil quality, and pests, when making predictions based on stacked images of crop fields.

According to the findings of Ijaz et al. [12]'s research in the spring of 2021, the relationship between the variables became more robust by including extra weather data, and especially the temperature was a vital weather factor for predicting grain yield. Hence,

the utilization of such data may also reduce the margin of error in the prediction of the current study.

Chapter 6

Conclusion

The primary goal of this thesis was to assess the feasibility of predicting grain yield from multispectral images by developing a CNN regression model. Several factors have motivated the choice of the objectives presented in Section 1.2.

Despite using a simple CNN model with minimal tuning, this study achieved satisfactory results despite the relatively small dataset. Various combinations of datasets and CNN architectures were trained, resulting in a wide range of outcomes. The most effective combination was Model 5, which used the 22MLROBOT dataset and included DH and fertilization level as additional variables to improve the model's predictive accuracy. Including additional variables, such as DH and fertilization level, effectively enhanced the correlation between the predicted and measured grain yield. Specifically, from Model 4 to Model 5 there where an improvement of the R^2 and the MSE score. The R^2 score improved from 0.851 and an MSE of 0.09 to an R^2 score of 0.885 and an MSE of 0.06. The inclusion of agronomic and management information as a concatenated layer further improved the prediction results by approximately 4%. Overall, adding DH and fertilization level to the model helped to better understand how these factors impact grain yield production.

Therefore, the approach used in this thesis shows promising results, especially considering the differences. Through rigorous analysis and experimentation, the results suggest that utilizing the entire plot as a sample yields richer information and thus can be a useful approach for data analysis. The performance of the combinations has been compared to each other and to two earlier studies, Shafiee et al. [11], and Ijaz et al. [12], where the median value and not images have been utilized for prediction.

Based on the comparison with the results of the study by Ijaz et al. [12], it can be concluded that the best-performing model developed in this thesis, Model 5, with an R^2 score of 0.885, is significantly better than the best-performing model in that study, which utilized a Random Forest Regressor with spectral bands, spectral indices, and weather features and achieved an R^2 score of 0.725.

Bibliography

[1] United Nations Association of Norway. Utrydde sult. URL: https://www.fn.no/om-fn/fns-baerekraftsmaal/utrydde-sult.

[2] Hernán G Villarraga. Population and environment. *Villarraga, Hernán G*, 2022.

[3] CGIAR Wheat - Nutrition Research and Impacts. URL: https://wheat.org/nutrition/.

[4] FAO. *The State of the World's Land and Water Resources for Food and Agriculture – Systems at breaking point (SOLAW 2021): Synthesis report 2021*. FAO, Rome, Italy, 2021. doi:10.4060/cb7654en.

[5] Giorgio Impollonia, Michele Croci, Andrea Ferrarini, Jason Brook, Enrico Martani, Henri Blandinières, Andrea Marcone, Danny Awty-Carroll, Chris Ashman, Jason Kam, Andreas Kiesel, Luisa M. Trindade, Mirco Boschetti, John Clifton-Brown, and Stefano Amaducci. UAV Remote Sensing for High-Throughput Phenotyping and for Yield Prediction of Miscanthus by Machine Learning Techniques. *Remote Sensing*, 14(12):2927, January 2022. Number: 12 Publisher: Multidisciplinary Digital Publishing Institute. doi:10.3390/rs14122927.

[6] Global agriculture towards 2050. URL: https://www.fao.org/fileadmin/templates/wsfs/docs/Issues_papers/HLEF2050_Global_Agriculture.pdf.

[7] Feltdag på NMBU med droner, robot og ny genetikk, August 2021. URL: https://www.nmbu.no/aktuelt/node/43702.

[8] Xiaofei Yang, Yunming Ye, Xutao Li, Raymond YK Lau, Xiaofeng Zhang, and Xiaohui Huang. Hyperspectral image classification with deep learning models. *IEEE Transactions on Geoscience and Remote Sensing*, 56(9):5408–5423, 2018.

[9] Bing Lu, Phuong D Dao, Jiangui Liu, Yuhong He, and Jiali Shang. Recent advances of hyperspectral imaging technology and applications in agriculture. *Remote Sensing*, 12(16):2659, 2020.

[10] Zvi Hochman, Dean Holzworth, and JR Hunt. Potential to improve on-farm wheat yield and wue in australia. *Crop and Pasture Science*, 60(8):708–716, 2009.

[11] Sahameh Shafiee, Lars Martin Bøe Lied, Ingunn Burud, Jon Arne Dieseth, Muath K. Alsheikh, and Morten Lillemo. Sequential forward selection and support vector regression in comparison to LASSO regression for spring wheat yield prediction based on UAV imagery. *9*, 2021. doi:10.1016/j.compag.2021.106036.

[12] Muhammad Fahad Ijaz. Spring wheat trait prediction using combined multi-environment, weather and multispectral timeseries UAV data. Master's thesis, Norwegian University of Life Sciences, Ås, 2021.

[13] Robert A. Schowengerdt. *Remote Sensing: Models and Methods for Image Processing*. Elsevier, August 2006.

[14] Lingli Zhu, Juha Suomalainen, Jingbin Liu, Juha Hyyppä, Harri Kaartinen, Henrik Haggren, et al. A review: Remote sensing sensors. *Multi-purposeful application of geospatial data*, pages 19–42, 2018.

[15] Toni Mastelic, Josip Lorincz, Ivan Ivandic, and Matea Boban. Aerial imagery based on commercial flights as remote sensing platform. *Sensors*, 20(6):1658, 2020.

[16] Ken Whitehead and Chris H Hugenholtz. Remote sensing of the environment with small unmanned aircraft systems (uass), part 1: A review of progress and challenges. *Journal of Unmanned Vehicle Systems*, 2(3):69–85, 2014.

[17] Electromagnetic Radiation - an overview | ScienceDirect Topics. URL: https://www.sciencedirect.com/topics/neuroscience/electromagnetic-radiation.

[18] Electromagnetic Spectrum - Introduction. URL: https://imagine.gsfc.nasa.gov/science/toolbox/emspectrum1.html.

[19] Reflected Near-Infrared Waves | Science Mission Directorate. URL: https://science.nasa.gov/ems/08_nearinfraredwaves.

[20] G. Velu and Ravi Prakash. *Phenotyping in Wheat Breeding*. January 2014. Journal Abbreviation: Phenotyping for Plant Breeding: Applications of Phenotyping Methods for Crop Improvement Publication Title: Phenotyping for Plant Breeding: Applications of Phenotyping Methods for Crop Improvement. doi:10.1007/978-1-4614-8320-5_2.

[21] Achim Walter, Frank Liebisch, and Andreas Hund. Plant phenotyping: from bean weighing to image analysis. *Plant Methods*, 11(1):14, March 2015. doi:10.1186/s13007-015-0056-8.

[22] How to track crop growth using MSAVI, NDVI and NDRE | Regrow Ag. URL: https://www.regrow.ag/post/how-to-track-crop-growth-using-msavi-ndvi-and-ndre.

[23] Bunkei Matsushita, Wei Yang, Jin Chen, Yuyichi Onda, and Guoyu Qiu. Sensitivity of the Enhanced Vegetation Index (EVI) and Normalized Difference Vegetation Index (NDVI) to Topographic Effects: A Case Study in High-density Cypress Forest. *Sensors*, 7(11):2636–2651, November 2007. Number: 11 Publisher: Molecular Diversity Preservation International. doi:10.3390/s7112636.

[24] Eva Boegh, Henrik Soegaard, N Broge, CB Hasager, NO Jensen, Kirsten Schelde, and Anton Thomsen. Airborne multispectral data for quantifying leaf area index, nitrogen concentration, and photosynthetic efficiency in agriculture. *Remote sensing of Environment*, 81(2-3):179–193, 2002.

[25] Alfredo R Huete, Kamel Didan, Yosio E Shimabukuro, Piyachat Ratana, Scott R Saleska, Lucy R Hutyra, Wenze Yang, Ramakrishna R Nemani, and Ranga Myneni. Amazon rainforests green-up with sunlight in dry season. *Geophysical research letters*, 33(6), 2006.

[26] Xiangming Xiao, David Hollinger, John Aber, Mike Goltz, Eric A Davidson, Qingyuan Zhang, and Berrien Moore III. Satellite-based modeling of gross primary production in an evergreen needleleaf forest. *Remote sensing of environment*, 89(4):519–534, 2004.

[27] T Miura, AR Huete, WJD Van Leeuwen, and K Didan. Vegetation detection through smoke-filled aviris images: An assessment using modis band passes. *Journal of Geophysical Research: Atmospheres*, 103(D24):32001–32011, 1998.

[28] Xiangming Xiao, Bobby Braswell, Qingyuan Zhang, Stephen Boles, Stephen Frolking, and Berrien Moore III. Sensitivity of vegetation indices to atmospheric aerosols: continental-scale observations in northern asia. *Remote sensing of Environment*, 84(3):385–392, 2003.

[29] Jadunandan Dash and PJ Curran. The meris terrestrial chlorophyll index. 2004.

[30] Dr Alexander F. H. Goetz. Measuring the Earth from Above: 30 years (and Counting) of Hyperspectral Imaging. URL: `https://www.photonics.com/Articles/Measuring_the_Earth_from_Above_30_years_and/a47298`.

[31] José Manuel Amigo and Silvia Grassi. Chapter 1.2 - Configuration of hyperspectral and multispectral imaging systems. In José Manuel Amigo, editor, *Data Handling in Science and Technology*, volume 32 of *Hyperspectral Imaging*, pages 17–34. Elsevier, January 2019. `doi:10.1016/B978-0-444-63977-6.00002-X`.

[32] AVIRIS - Airborne Visible / Infrared Imaging Spectrometer - Imaging Spectroscopy. URL: `https://aviris.jpl.nasa.gov/aviris/imaging_spectroscopy.html`.

[33] Jay Selig. What Is Machine Learning? A Definition., March 2022. URL: `https://www.expert.ai/blog/machine-learning-definition/`.

[34] Sebastian Raschka and Vahid Mirjalili. *Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2*. Packt Publishing Ltd, 2019.

[35] Stavros P Adam, Stamatios-Aggelos N Alexandropoulos, Panos M Pardalos, and Michael N Vrahatis. No free lunch theorem: A review. *Approximation and Optimization: Algorithms, Complexity and Applications*, pages 57–82, 2019.

[36] Wanliu Mao, Debin Lu, Li Hou, Xue Liu, and Wenze Yue. Comparison of Machine-Learning Methods for Urban Land-Use Mapping in Hangzhou City, China. *Remote Sensing*, 12(17):2817, January 2020. Number: 17 Publisher: Multidisciplinary Digital Publishing Institute. `doi:10.3390/rs12172817`.

[37] Jack Stilgoe. Machine learning, social learning and the governance of self-driving cars. *Social Studies of Science*, 48(1):25–56, February 2018. Publisher: SAGE Publications Ltd. doi:10.1177/0306312717741687.

[38] Unseok Lee, Sungyul Chang, Gian Anantrio Putra, Hyoungseok Kim, and Dong Hwan Kim. An automated, high-throughput plant phenotyping system using machine learning-based plant segmentation and image analysis. *PLOS ONE*, 13(4):e0196615, April 2018. Publisher: Public Library of Science. doi:10.1371/journal.pone.0196615.

[39] What Is Deep Learning? | How It Works, Techniques & Applications. URL: https://se.mathworks.com/discovery/deep-learning.html.

[40] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, October 1986. doi:10.1038/323533a0.

[41] Keras Team. Keras documentation: Regression losses. URL: https://keras.io/api/losses/regression_losses/.

[42] Asmelash Teka Hadgu, Aastha Nigam, and Ernesto Diaz-Aviles. Large-scale learning with AdaGrad on Spark. In *2015 IEEE International Conference on Big Data (Big Data)*, pages 2828–2830, Santa Clara, CA, USA, October 2015. IEEE. doi:10.1109/BigData.2015.7364091.

[43] Keras Team. Keras documentation: RMSprop. URL: https://keras.io/api/optimizers/rmsprop/.

[44] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization, January 2017. arXiv:1412.6980 [cs]. URL: http://arxiv.org/abs/1412.6980.

[45] Xingbo Wang, Jean-Baptiste Thomas, Jon Yngve Hardeberg, and Pierre Gouton. Multispectral imaging: narrow or wide band filters? 2014.

[46] Lars Martin Bøe Lied. Multispectral image analysis of spring wheat using UAV and machine learning. Master's thesis, Norwegian University of Life Sciences, Ås, 2019.

[47] Example - Clip — rioxarray 0.14.0 documentation. URL: https://corteva.github.io/rioxarray/stable/examples/clip_geom.html.

[48] SGOPAL Patro and Kishore Kumar Sahu. Normalization: A preprocessing stage. *arXiv preprint arXiv:1503.06462*, 2015.

[49] Data augmentation | TensorFlow Core. URL: https://www.tensorflow.org/tutorials/images/data_augmentation.

[50] scikit-learn: machine learning in Python — scikit-learn 1.2.0 documentation. URL: https://scikit-learn.org/stable/index.html.

[51] Keras Team. Keras documentation: EarlyStopping. URL: https://keras.io/api/callbacks/early_stopping/.

[52] Tianfeng Chai and Roland R Draxler. Root mean square error (rmse) or mean absolute error (mae). *Geoscientific model development discussions*, 7(1):1525–1534, 2014.

[53] Arnaud De Myttenaere, Boris Golden, Bénédicte Le Grand, and Fabrice Rossi. Mean absolute percentage error for regression models. *Neurocomputing*, 192:38–48, 2016.

[54] Padhma M. End-to-End Introduction to Evaluating Regression Models, October 2021.

[55] Alessandro Di Bucchianico. Coefficient of determination (r 2). *Encyclopedia of statistics in quality and reliability*, 2008.

[56] Residual plot > Linear fit > Fit model > Statistical Reference Guide | Analyse-it® 6.10 documentation. URL: `https://analyse-it.com/docs/user-guide/fit-model/linear/residual-plot`.

[57] Qunying Luo. Temperature thresholds and crop production: a review. *Climatic change*, 109(3-4):583–598, 2011.

[58] Yinhong Kang, Shahbaz Khan, and Xiaoyi Ma. Climate change impacts on crop yield, crop water productivity and food security–a review. *Progress in natural Science*, 19(12):1665–1674, 2009.

[59] Alain Ratnadass, Paula Fernandes, Jacques Avelino, and Robert Habib. Plant species diversity for sustainable management of crop pests and diseases in agroecosystems: a review. *Agronomy for sustainable development*, 32:273–303, 2012.

[60] Petteri Nevavuori, Nathaniel Narra, and Tarmo Lipping. Crop yield prediction with deep convolutional neural networks. *Computers and Electronics in Agriculture*, 163:104859, August 2019. `doi:10.1016/j.compag.2019.104859`.

Appendix A

GitHub links

Table A.0.1: *Source code links for GitHub*

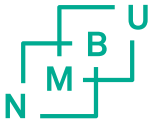
Title	**Link**	**Commit hash**
data_preparation.ipynb	GitHub:data_preparation	32afed4
ml_pipeline.ipynb	GitHub:ml_pipeline	bd80391

Appendix B

Program and Python models

Table B.0.1: *Program and Python modules and their appliance used in this study.*

Module / Program name	**Appliance**	**Version**
Python	High level programming language	3.9.16
QGIS	GIS program	3.22.11
Tensorflow	Deep learning framework	2.12.0
Keras	Deep learning API	2.12.0
Spectral	Multispectral image	0.23.1
Matplotlib	Plotting graphic	3.7.1
Seaborn	Plotting graphic	0.12.2
Numpy	Mathematical functions	1.22.4
Pandas	Data analysis	1.5.3
GeoPandas	Geographic pandas extensions	0.12.2
xarray	labeled arrays and datasets	2022.12.0
Rioxarray	geospatial xarray extension	0.14.1
netCDF4	Data storage format type	1.6.3
HDF5	File format for data	3.8.0
PIL/Pillow	Python Imaging Library	8.4.0
Scikit-Learn	Python library for machine learning	1.2.0



Norges miljø- og biovitenskapelige universitet
Noregs miljø- og biovitskapelege universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway