Tel Hai College

Department of Water Sciences

Developing models for soil texture classification from ancillary data using machine learning

by Ido Dan

Supervisors:
Michael (Iggy) Litaor
Ofer M. Shir

September 2021

Student signature: Date:
Supervisor signature: Date: 24/01/2022
Supervisor signature: Date:

This work was done under the guidance of:

Prof. Michael (Iggy) Litaor - Department of Water Sciences, Tel-Hai college and MIGAL.
Prof. Ofer M. Shir - Department of Computer Science, Tel-Hai college and MIGAL.

Acknowledgements

I would like to express my gratitude to my teachers and mentors, Prof. Iggy Litaor and Prof. Ofer Shir, for their guidance and support along the way. Special thanks to Asaf Israeli who was always happy to assist when I needed and to the hydro-geochemistry lab members at MIGAL who helped me to acquire the data for this project and I enjoyed their company, including Adi Zarka, Nimrod Meitav and Dr. Oren Reichmann. Special thanks to Tel Hai college that has been a second home to me for the past 5 years and an even greater thanks to Limor Turgeman who was a second mom in Tel Hai.

Finally, I would like to thank my dear wife, Lior, for her support and love.

Abstract

Traditional methods for measuring soil texture (sand, silt and clay relative percentages) are laborious and time consuming. Studies have shown that soil texture can be linked to sensory data that is much easier and cheaper to acquire. The current study proposes a methodological approach for constructing machine learning models for predicting soil texture when given ancillary and imagery data as input. This study also provides a detailed description of evaluation measures to assess the models' success-rate, by considering a random model as a control group, as well as an explicit formulation of the objective function – sum root mean square error (SRMSE) – to be minimized during the learning process. In the case study presented here, 63 soil samples were analyzed by two different instrumental measurements, namely the Hydro-meter and the Laser Diffraction, and various models were trained over the data in order to predict results from both. In addition, computational methods for assessing the suitability of different types of ancillary data were proposed and various techniques to improve machine learning models were examined. Evidently, better prediction rates in this study were obtained per the data measured by the Laser Diffraction. Especially, the Random Forest and Neural Network with ancillary data were the most successful models.

Key words: soil texture, machine learning, precision agriculture, hydro meter, laser diffraction.

Contents

List of Figures

List of Tables

List of Acronyms

PA - Precision agriculture
HM - Hydro meter
LDS - Laser diffraction system
MS - Master sizer
RMSE - Root mean square error
SRMSE - Sum root mean square error
RF - Random forest
MLR - Multiple linear regression
NN - Neural network
CNN - Convolutional neural network
EA - Evolution algorithm
GA - Genetic algorithm
ECa - Apparent electrical conductivity
MSa - Apparent magnetic susceptibility
NDVI - Normalized difference vegetation index
NIR - Near infra red
TRS - Thermal remote sensing
JPG - Joint photographic group
ROI - Region of interest
CEC - Cation exchange capacity
PSD - Particle size distribution
RPD - Residual prediction deviation
RPIQ - Ratio of performance to interquartile distance
DTM - Digital terrain model

Chapter 1

Introduction

The agricultural revolution, starting around 10,000 B.C., marked the transition from small hunter-gatherers communities to larger agricultural settlements and the beginning of civilization. Since than, human life and the constant growth in population goes hand-in-hand with agricultural advances.[1] For most of history, the growth in population and the need to increase yields was usually met by expanding the cultivated area. In the 19th century in what is called the *Green Revolution* new technological advancements like inorganic fertilizers, chemical pesticides and the use of industrial agricultural practices managed to produce much more crops in the same area. Until the middle of the 20th century most developed countries managed to achieve sustained food surpluses and many developing countries were able to close this gap in the following decades [1]. In the beginning of the 21th we have reached a point in which roughly one-third of the food produced worldwide is wasted[2], and where the obesity problem is considered more severe than the hunger problem. In addition, the agricultural industry is one of the main contributors to climate change with excessive irrigation, over-use of fertilizers and pesticides that permeable to water bodies and contaminate soils, deforestation and it is one of the main contributors of greenhouse gases to Earth atmosphere. The difficulty to estimate within-field spatial variation of soil characteristics leads to prevailing uniform application methods that causes sub-optimal conditions in some areas while others are over irrigated or fertilized and cause waste of resources that finally reach nature and causes environmental problems [2].

Precision Agriculture (PA) are a relatively new concept developed in the mid-1980th that wishes to answer some of the challenges listed above and others, to feed the growing population but to minimize the anthropogenic affect on the environment, produce more with less resources. According to Pierce et al. [3] PA provides the possibility to do the right thing, in the right place, at the right time and in the right way. It aims to constantly monitor every point in the field, to understand the current status and to make real-time and research based decisions.

[1] History of Agriculture - `https://www.crestcapital.com/tax/history_of_agriculture`

[2] urlhttps://www.unep.org/thinkeatsave/get-informed/worldwide-food-waste

Recent technological advancements for water conservation and optimal use of resources use different types of sensors for mapping the field and build an optimal strategy based on the field characteristics [4]. The term soil texture refers to the relative content of particles of various sizes, (1) sand (2 to 0.02 mm particles) (2) silt (0.02 to 0.002 mm particles) and (3) clay ($\leq$0.002 mm particles), which is represented by a trinary vector of the percentage of each one of them (that sums to 100% - [30% sand, 30% silt, 40% clay] for example). Soil texture is usually constant over time and it is associated with soil porosity, which in turn regulates the water holding capacity, gas diffusion, water infiltration and it affects the entire microbial population in the ground and in 2009 Katreji et al. [5] showed the connection between soil texture and water use efficiency. It dictates water infiltration and the amount of available water for plants roots intake which makes it a valuable parameter for agronomists and soil researches.

Traditional methods for measuring soil texture require manual field measurements that must be analyzed in a lab which is expensive in time and labor which makes it not feasible for large areas. There is more than one method to measure soil texture, and in this research, two were used: The first is the traditional and more common, the Hydro Meter (HM) method that is based on Stoke's law and the calculation is done under the assumption that soil particles are spherical [6]. The second method is the laser diffraction system (LDS) that was done using the Mastersizer (MS) v3.5-3000 (Malvern Panalytical Ltd. Malvern, U.K.). This method is based on measuring the scattered laser beam sent on desired soil sample [7].

The results from the two methods do not agree with each other and the subject of matching textural results acquired from one method to the other has been discussed in previously published literature but there is not a clear answer whether the results are comparable. Al-Hashemi et al. [8] and Fea et al. [9] encourage researchers to use the LDS and claim that with the appropriate pre-treatment the results can be matched to traditional methods results but Eshel et al. [10, 11] claim that there is a potential in using the LDS but as an independent measurement and that it cannot be compared to other methods. Taking side in this discussion is beyond the scope of the work presented here so the results from both were used but separately as almost two independent projects.

Previous work have tried to link soil texture to various features in the field which can be measured much easily. Carroll et al. [12], Kelley et al. [4] and Heil et al. [13] mapped clay content with electromagnetic induction. Riese et al. [14] and Casa et al. [15] classified soil texture based on hyper-spectral data. Khanal et al. [16], Muller et al. [17] and Wang et al. [18] used thermal remote sensing for estimating soil texture and Greve et al. [19] used environmental parameters to predict soil texture. In addition, attempts to predict soil texture based on soil images were made by Qi et al. [20] Morais et al. [21] and Swetha et al. [22] with some successes. The studies listed above focused on a single parameter for predicting soil texture and an obvious question is what will be predictions performance if more than one parameter will be used?

Creating a prediction model for multiple input parameters requires sometimes computational methods that are more sophisticated than traditional statistical tools like multiple linear regression. Machine learning (ML) is a branch in computer science that comes to help with this challenge. It specializes in learning the mathematical connections between features and it is widely used in agricultural production systems like: (a) crop management, creating applications for yield prediction, disease and weed detection and quality tests; (b) livestock management; (c) water and fertilizers management; and (d) soil management [23]. Soil texture is an unusual continues value, since it is has three values that sums to 100%, so a normalization layer had to be implemented on top of the ML prediction:

$$sand\% = \frac{sand}{sand + silt + clay}, \quad silt\% = \frac{silt}{sand + silt + clay}, \\ clay\% = \frac{clay}{sand + silt + clay} \tag{1.1}$$

Where $sand, silt$ and $clay$ are the predicted values and $sand\%, silt\%$ and $clay\%$ are the normalized to 100% values.

A key concept in ML is the use of large data sets for constructing prediction models hence the natural connection to PA that uses many sensors that produce a lot of data. The use of large data sets require building ML tools that are automatic and with minimum manual operations that waste valuable time. The more generalized an ML project is, the more it could used for similar tasks in the future. The task of predicting soil texture in one field is of course almost identical to predicting soil texture in a different field, therefor the structure of the ML project must be designed in a flexible way that allows adding more data sources easily or adding more input features. The code written for this project is published via `github`[3] and is publicly available for other researches.

Objective and Research Question

The study presented here is a test-case of predicting soil texture from ancillary and image data in a specific area but its goal is to serve as a blue-print and a proof of concept for a more generalized research that will use data from more than one field.

The current study targets the following research question:

> Which features are most suitable for predicting soil texture from ancillary data and which machine learning model should be used?

[3] `https://github.com/idodan1/thesis.git`

Contribution and Structure

The contribution of this project is,

1. Proposes statistical methods for evaluating machine learning models performance.
2. Suggests ways to deal with small data sets in the task of predicting soil texture.
3. Creates machine learning model that uses multiple ancillary data channels with image data.

This thesis has the following structure: Chapter 2 provides a short introduction to the computational concepts that guided the approach for the task of predicting soil texture and why they are relevant for this task. Chapter 3 explains in detail the types of data used in this project - why they were chosen for this project and how they were collected. Chapter 4 outlines the computational steps that were taken for fitting the computer science theory to the real world problems and real world data in here. The practical results of the case study are reported in Chapter 5 where it is also discussed which methods show better performance. Finally, project findings are summarized in Chapter 6, and suggestions for future work are presented.

Chapter 2

Computational Methodology

In this chapter the general history of Machine Learning (ML) and its use in Precision Agriculture (PA) is presented, as well as the concrete ML algorithms used throughout this work. Then the general concepts of Evolutionary Algorithms (EAs) are outlined, and particularly the Genetic Algorithm (GA); in addition, the main implementation principles of this work are described.

2.1 Machine Learning: A Brief Background

The original use of the term *Computer* is for "one who calculates, one whose occupation is to make arithmetical calculations"[1] and it stems from the verb *to compute*. In its early days, in the middle of the 19th century, the role of the computer was to replace human calculators, to do basic mathematical operations rapidly and without errors.[2] Only later in history the idea that a computer can mimic the human ability to learn was presented [24]. Human learning is based on experience, a human observes a phenomenon in the outside world and learns from it. The field of ML was constructed on the same concepts. The main idea for a learning algorithm is to use input data to achieve a desired task without literally programmed. The input data is accompanied with desired outcome and the algorithm alters its parameters to fit to it as much as possible, in a process called training. The training is the "learning" part, in which the computer discovers by itself what is the best way to approach the desired task. A baby can distinguish between a dog and a cat even before being fully able to speak,[3] but coding a computer program that will do the same task is apparently quite difficult. Instead of trying to hard code all the different representations of every animal it is easier to feed the computer with many annotated examples of dogs and cats and let it find the best way to identify which one is it. This type of learning is called supervised learning, in which every data point in the input is accompanied

[1] Online etymology dictionary:`https://www.etymonline.com/word/computer`

[2] `https://www.wikipedia.org`

[3] `https://www.babycenter.com`

with a label. The algorithmic goal is to learn the relationships between them and to be able to predict the label of unseen data later. A second type of ML is unsupervised learning, which uses unlabeled data, and are useful when a complex processing is needed. This type of algorithms are usually used in different types of clustering tasks. The term of unlabeled data describes data that does not have a specific attribute that defines it. In the example above, every image has a label *animal-type* (cat or dog) that defines it. If the task was to partition the cats' images to two clusters having the strongest self-similarity within, without being given specific attributes to this end, the learning algorithm would have to find the best way to do it by itself [25].

2.2 Machine Learning in Precision Agriculture

The never-ending growth in population, as well as the need to feed it with decreasing farming areas, force farmers and agronomists to search for better ways to perform farming, to produce more yield with less resources [23]. With the help of accurate sensors of different types, e.g., drones and satellites, it became possible to generate big data sets with many characteristics describing field and crops in a fast and relatively cheap way [23]. To use all these data for better predictions, for constructing data-based strategies and for real time decisions based on field current, status it is crucial to use methods that are big-data-oriented [26]. For those reasons, it is only natural to use ML in PA, which is suited to large data sets. Liakos et al. [23] and Dimitriadis et al. [26] describe in detail the fields in agriculture that adopts ML techniques and explain why it is so beneficial. It is clear that because of the large number of variables that affect field status and the constant change in them, it is too complicated for a human to take all the information into consideration. Thus, the potential for an Artificial Intelligence system to process all this information is much greater.

2.3 ML Models

All the ML models in this work are supervised learning models, i.e., learning through annotated examples; in addition, a *random model* plays a role of a control/reference group to the ML models. The 4 ML models that were used: (1) Random Forest (2) Multiple Linear Regression (3) Neural Network and (4) Convolutional Neural Network [23]. In this work, the data set is constructed from 63 sampling points, the data on each sampling point was split to a feature vector (explanatory variables such as ECa - apparent electrical conductivity or NDVI - normalized difference vegetation index) and to the associated label (soil texture values). When using guided learning models it is important to split the data set to train and test sets. The train set (54 points $\approx 85\%$ of all sampling points) is used for model calibration and the test set (9 points $\approx 15\%$) is used for evaluation [23].

All the models were implemented in `python 3.6` environment (Python software foundation, DE, USA) [27].

2.3.1 Random Model

The goal of this model is to serve as a control group for all the other models. The random model is based on the training data but there is no real learning occurring when it is constructed. A single random model is created based on the mean and standard deviation of the 3 textural classes (sand, silt, clay) in the training set. A prediction is defined to be 3 random values, one for each textural class, based on the normal distribution around the class mean and standard deviation,

$$sand = \mathcal{N}(\mu_{sand}, \sigma_{sand}), \quad silt = \mathcal{N}(\mu_{silt}, \sigma_{silt}), \quad clay = \mathcal{N}(\mu_{clay}, \sigma_{clay})$$

where $\mathcal{N}$ denotes the normal distribution, $\mu_{sand}, \mu_{silt}, \mu_{clay}$ are the mean values of the textural classes, and $\sigma_{sand}, \sigma_{silt}, \sigma_{clay}$ are their standard deviations, respectively. The 3 values are then normalized to 100$\%$ based on equation (1.1) and this serves as a baseline prediction to be compared to a test set data point. The Root Mean Square Error (RMSE) between the measured and predicted of the 3 textural classes is calculated and summed to a single value, referred to as Sum Root Mean Square Error (SRMSE), and the mean SRMSE over all data points serves as the final result of the random model. In order to get a full picture of the possible range of the SRMSE that a random model can achieve, a histogram plot, which depicts the normal distribution of the random model, was generated out of 100,000 random models that predicted the test set, their SRMSE was calculated.

2.3.2 Random Forest (RF) - Multiple decision trees

A decision tree is a model built from decision nodes and leaves (Figure 2.1) and it is used for classification and regression problems [28]. The decision process for a single data point is made by “traversing” along the tree, starting in the root, and in every junction, one turns either left or right based on a prescribed condition per this junction that tests a concrete coordinate within the feature vector (e.g., if ECa is larger than 5 turn left, else turn right). One proceeds until reaching a leaf, where the decision concerning the data point is made. Building a decision tree serves as the learning process in which nodes are iteratively added to the tree by considering a certain property from the feature vector and a value that splits the training set in the best way. The process is terminated when the improvement as a result from adding another node is smaller than a threshold value and that node turned into a decision leaf. The Random Forest (RF) model is a set of multiple decision trees, where in every tree a subset of properties from the

feature vector is chosen randomly and the final decision is made by summing the results from all the trees. Every tree by itself could be less accurate than a tree with all the properties included but there is a high probability that the summation over all the different trees will account for the influence of all the properties and will obtain better results. Overall, this process improves the algorithmic outcome because the summation helps to neutralize noise and to avoid over-fitting [23].

Our implementation of the RF model is done with the code package `sklearn` [29] which allows implementing different architectures for the forest.

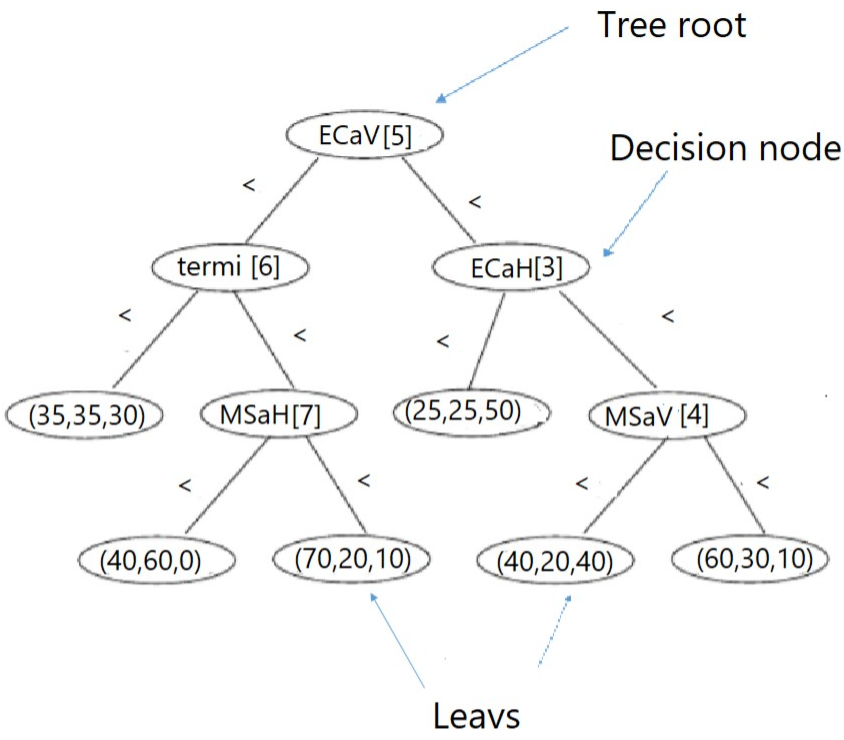


Figure 2.1: Decision tree visualization. When classifying a data point: start in the root and go down the tree until reaching a leaf. In the decision nodes there is one feature (ECah for example) and a decision value (if larger than 3 turn right otherwise turn left). In the decision leaves there is a prediction of the trinary soil texture value that fits the route from the root. An RF is built from multiple decision trees. (ECah - Apparent Electrical Conductivity horizontal, MSav - Apparent Magnetic Susceptibility vertical).

2.3.3 Multiple Linear Regression (MLR)

MLR is a statistical technique that uses several explanatory variables to predict target variable [30]. The goal is to model the linear relationship between the explanatory variables (independent) to the target variable (dependent). The outcome of this model is a set of coefficients and

a bias that describe the relationship between the independent to dependent variables.

$$y = b + a_1x_1 + a_2x_2 + a_3x_3 + a_4x_4 + ... + a_kx_k,$$

where b is the bias (the intercept with y-axis), a_i are the coefficients, x_i are the explanatory variables and y is the target value. The learning process is to find the plane that fits the data best and it is done by minimizing the root mean square between measured and calculated values [31].

This model is considered relatively simple but, in many cases, it is good enough and it is easy to implement in `python` using the `sklearn` code package [29].

2.3.4 Neural Network (NN)

This algorithm imitates the behavior of biological neural networks in animals, with computer simulations of mathematical functions, and it is usually used for classification or regression problems [32]. The human brain is built from billions of neurons connected to each other, every neuron by itself is a relatively simple computation unit but their connectivity, which is made with electro-chemical signals, increase their computing power, and allows us to do complex tasks such as: image recognition (our vision), mathematical operations, abstract thinking and more. The neural network receives input from the outside world through our senses, transfers it through the net, process them and outputs a decision in the form of action. Similarly, NN is an abstract model that represents a biological network but much smaller (Figure 2.2). Every virtual neuron receives an input, performs a simple mathematical operation on it and outputs the result to the next neuron down the net. The neurons in the net are divided into 3 types of layers:

1. Input layer – receives the input for the model, the number of neurons in it is identical to the dimension of the feature vector.

2. Hidden layers – one or more layers that do nonlinear transformation of the inputs.

3. Output layer – the final layer in which the decision about the input is made.

The connection between two neurons is called a weight (a real number) and it represents the relation between them. The NN procedure for calculating the output of a single input is called “Feed Forward” and it is done as follows: the input layer starts with the feature vector, the first neurons apply a nonlinear transformation and feed the next layer with the results after they are multiplied by the appropriate weights. This procedure repeat itself in the hidden layers until reaching the output layer. The learning phase is done using the labeled samples in the training set, with them the model calibrates the weights among the neurons. In the beginning, the weights are initialized with random values, the samples are fed to the model and the results in

the output layer are compared with the labels. The distance between them is calculated and it is used in a "back propagation" manner to update the weights in a way that minimizes the distance. After several iterations, the weights are expected to converge to values representing the connections among the different variables (assuming that there is a connection) and the net can be used for predicting unseen data [23, 33, 34].

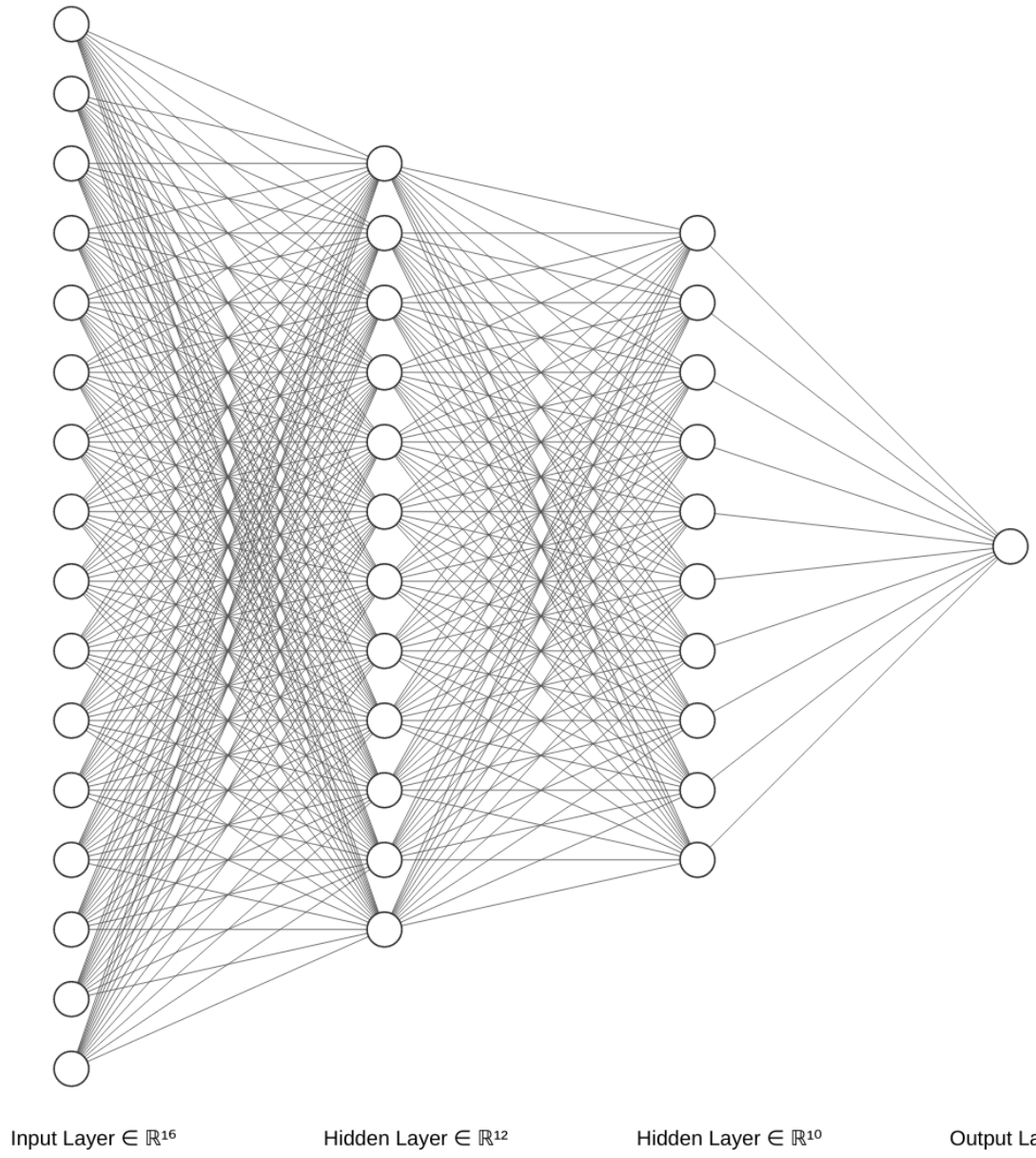


Figure 2.2: NN illustration. Here the input layer has 10 neurons, and the hidden layers have 8 and 6 neurons each. The output layer has 3 neurons. Created in `http://alexlenail.me/NN-SVG/index.html`

2.3.5 Convolutional Neural Network (CNN)

The name of this model stems from the mathematical linear operation between matrices called *convolution*. A CNN is built from layers similarly to NN, but it has other types of layers such as convolutional layers, pooling layers, and fully non-linearity layers (Figure 2.3) [35]. CNNs exhibit excellent performance in ML tasks, especially in applications that deal with imagery data [36]. Convolutional layers specialize in pattern recognition that is not spatially dependent and it is commonly used in agricultural learning tasks that require analysis of the field's shape [35, 37]. In this research two types of CNNs were implemented. The first is the type of net that receives as input only imagery data (Figure 2.3) – `ConvImg`. The second is a type of net that receives both imagery data and numeric data (Figure 2.4) – `Conv`. The second type of net is built from two separate tensors that process the data (image or numeric) and then connects (Figure 2.4) into one tensor and continue as a regular net. The advantage of this type of net is that it analyses each data channel by itself and then uses the two outputs together for better predictions. On the other hand, the learning process of the two channels can interfere with each other, and this high complexity could sometimes become a burden when compared to a simpler model that relies on a single channel [38].

In this research both NN, CNN with only imagery data (`ConvImg`) and CNN with image + numeric data (`Conv`) were implemented. All of them were implemented in `python` using the renowned Tensor-Flow [39] and Keras [40] code packages.

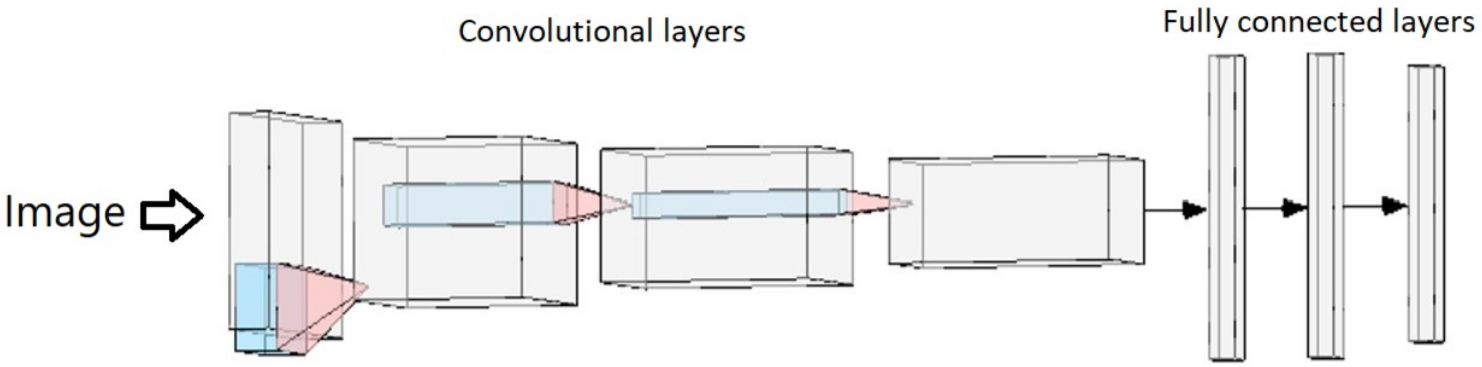


Figure 2.3: CNN illustration. In a CNN, usually the first layers will be the convolutional layers and the final layers will be fully connected layers (like NN layers). The net in this figure is image only type of net. Created `http://alexlenail.me/NN-SVG/index.html`

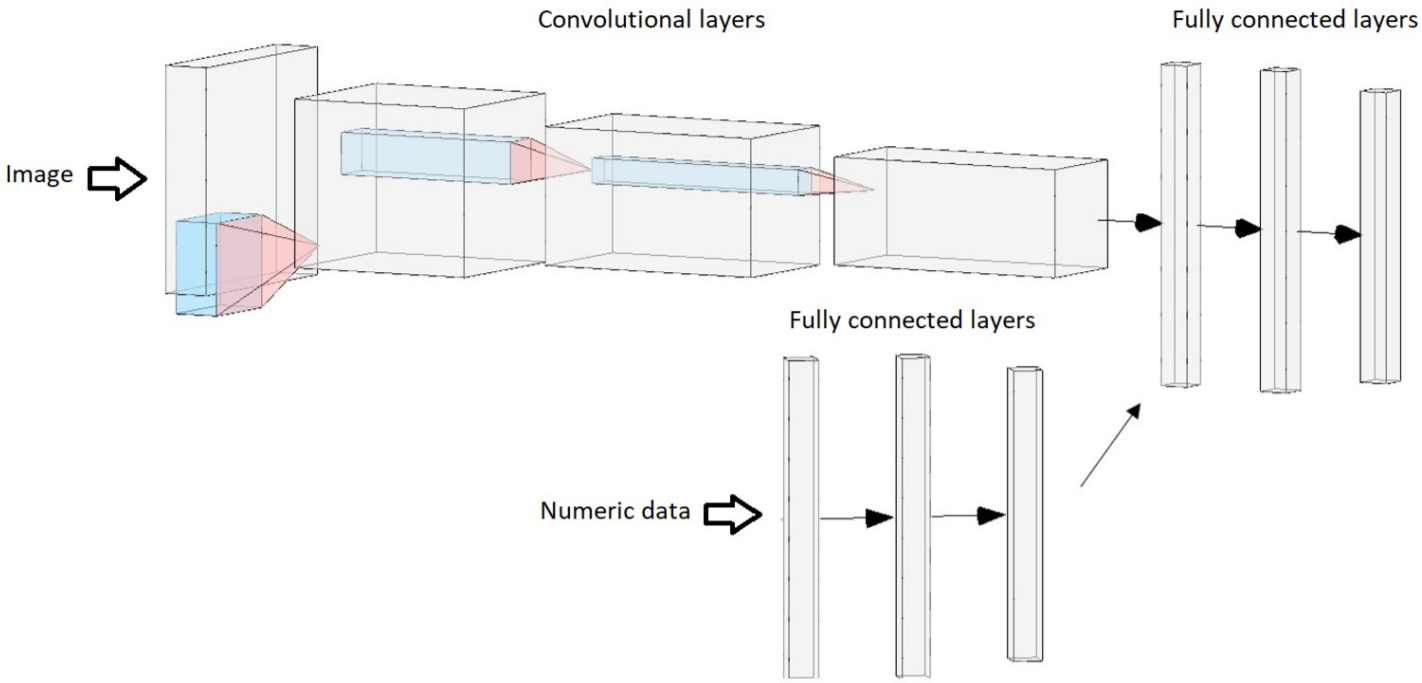


Figure 2.4: CNN illustration. A net that receives two types of data, image data and numeric data. Created in `http://alexlenail.me/NN-SVG/index.html`

2.4 The Genetic Algorithm (GA)

In 1859 Charles Darwin introduced the concept of Evolution, a biological process that performs as a problem-solver to the hard task of surviving in nature. In the twentieth century this idea was reinforced by molecular biologists that revealed the existence of DNA, a complex molecule that encompass the hereditary traits that are passed on from parent to children. The parts in the DNA that code these traits are called genes and the entire set of genes is called the genome. The main mechanism in Evolution is "The Survival of the Fittest", which says that an individual that is more fitted to its environment has higher probability to survive and to pass on its genes to the next generation. A tiger with an albino mutation will have a hard time hiding in the jungle and will probably die without having any offspring. However, if an ice age will occur at the same time, it will probably be the fittest tiger and will have many offspring that will carry on his genes. When it comes to genes, the process of evolution operates via two primary mechanisms. In the first, recombination, the genes from two parents mix for creating a new genome that is similar to both parents but not identical to either of them. This step saves the good traits from fitted parents to offspring. In the second step, mutation, random changes occur in the new genome and the result can alter the gene function, prevent its function, or have no effect. This step adds variation to the population which is needed in a changing environment.

In mathematics, the procedure called optimization refers to the process of finding the minimum (or maximum) of an objective function by adjusting decision variables while satisfying a set of constraints [41].

GAs simulate the evolutionary process on a computer for finding good solutions in an optimization problem. First, a population of candidate solutions is generated (usually randomly) and according to their rate of successes in minimizing the objective function, the fittest solutions are selected; in a sequential process, which imitates the recombination and mutation operations in nature, new solutions are generated and evaluated again. This process is repeated until reaching a suitable solution or for a predefined number of iterations and the best solution is chosen (Algorithm 1) [42].

Algorithm 1: Generalized GA

Result: most_fitted_population

$t \leftarrow 0$ (t: generation)

$P(t) \leftarrow initialize\_random()$ (P: population)

$fitness \leftarrow calculate\_fitness(P(t))$

while *termination condition not met* **do**

  $new\_P \leftarrow Recombination(P(t), fitness)$

  $new\_P \leftarrow Mutate(new\_P)$

  $fitness \leftarrow calculate\_fitness(new\_P)$

  $P(t) \leftarrow new\_P$

  $t \leftarrow t + 1$

return most_fitted

2.5 Implementation by Generic Programming

The term Generic Programming refers to developing source code that even though it was created for a specific task, it is written in a general way that will require only minor changes for using it in a similar problem. It aims to increase the flexibility of use and to enable the possibility to modify or to improve specific parts of the project with no effect on other parts [43]. The code for this project is written in a way that enables changes and improvements in an easy way:

1. Adding data - future work that will include additional data.
2. Adding data types - new sources of data, such as satellite data, which could improve the learning.
3. Changing the target variable - future work that aims to predict other observables/properties in the field.

4. Changing the optimization algorithm - there are competing optimization algorithms (e.g., Ant Colony Optimization [44]) that might be more suitable for the current task - future work could test them.

5. Add more learning algorithms - different problems might require different algorithms, it will be easy to add them.

The concrete objective of this study is to explores ML techniques for predicting soil properties. The main rationale behind our generic implementation is to enable future work, potentially by other researchers, for expanding or improving this research.

Chapter 3

Data Collection

In this chapter the 3 different types of data that were used in this work are described, (1) Ancillary data (2) Image data (3) Texture data (the first two serve as input features and the third is the target variable). This is the first, and probably the most important, step for building an ML project, its successes rely on the assumption that the data describes a phenomenon in the real world. A model that uses inconsistent or inaccurate data (e.g., due to excessive noise) will not be able to learn the connections between the data types and it can lead to false conclusions. The ancillary data and the soil samples were taken from Newe Ya'ar, the northern research farm of the agricultural department in Israel, located in the boundary between Yezreel valley and the lower Galilee. The model farm in Newe Ya'ar size is 35 ha, out of which 10 ha were used for this research.

3.1 Ancillary Data: Data Types and Collection

3.1.1 Soil Conductivity

The geophysical survey was done using the EM38-MK2 conductivity meter (Geonics Ltd., Canada) (Figure 3.1-a). The device measures both the Apparent Electrical Conductivity (ECa) and the Apparent Magnetic Susceptibility (MSa) in two positions (vertical and horizontal). The device measures in 14.6 kHz frequency and it has one transmitter and two receiver coils that are in different distances from the transmitter (0.5 and 1 meter). The two distances allow multiple measuring depth, in the vertical mode the measuring depth is between 0.75 and 1.5 meter and in the horizontal mode it is between 0.4 and 0.75 meter. This method allows a continuous measurement to root depth without direct contact or violating the ground. It is a simple, fast and relatively cheap method that is easily synchronized with satellite location and for those reasons it has become the most popular method for mapping soil electrical conductivity that helps in predicting other soil properties that are important for agricultural use [45, 46, 13]. The mea-

surements were done: (1) manually, only in data points location and (2) sequentially over the entire field with a wooden slide connected to a vehicle (Figure 3.1.b, c). The measurements were done few days after rain events because it is vital to measure when the soil moisture is not lower than 70% field capacity for accurate output [13].

Figure 3.1: **a.** EM38-MK2 EMI device and Archer XF101 GPS device. **b.** Special wagon without metal parts for containing the EMI device. **c.** Vehicle dragging the special wagon with the EM38-MK device inside it.

3.1.2 Normalized Difference Vegetation Index - NDVI

Multi-spectral data was acquired with a sensor capturing four spectral bands (green, red, red edge and near infra-red) positioned on a drone (Mavic Pro, DJI, Shenzhen, China) at a spatial resolution of 6x6 cm. The NDVI was calculated by the Red and Near Infra-Red (NIR) bands [47]:

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

3.1.3 Thermal Remote Sensing - TRS

TRS was done using a sensor (FLIR a655SC) located on a drone (DJI Matrice 600 Pro). The measured value is the surface temperature, measuring it in different times allows to identify differences in the thermal inertia, the spatial tendency of the soil to change the surface temperature, that was linked to soil texture [16, 17, 18].

3.1.4 Geographic Information System (GIS) and Light Detection and Ranging (LIDAR) data

Terrain parameters (elevation, slope gradient, slope aspect, flow accumulation) that were acquired with GIS software and LIDAR data were linked to soil texture fractions [19]. This data was available from past research conducted in Newe Ya'ar.

3.2 Image Acquisition

In previously published literature it have been shown that images of grounded soil can be linked to soil texture. Qi et al. [20] and Morais et al. [21] used a microscope for image acquisition, while Swetha et al. [22] used a simpler method with a smartphone camera. The second option was chosen due to its simplicity and the availability of use in the field.

3.2.1 Setup for Image Acquisition

A dark chamber (15 cm x 8cm x 8cm) was built from recycled materials (a cardboard) and a LED strip was assembled on the top for illumination. The chamber was painted in black to avoid light reflectance from the outside. Inside the box a square shaped area was fenced with cardboard strips and the smartphone camera, that was placed on the lid, and centered on the fenced area and the dried and grounded soil samples were spread uniformly across the strip (Figure 3.2**b.**). The purpose of the box was to standardize the image acquisition with consistent variables like light and camera distance. The images were taken using an Iphone 7 camera with 12-megapixel images (Figure 3.2**b.**). The captured images were saved as Joint Photographic Group (JPG) files (3024 X 3024 pixels) total of 63 images [22].

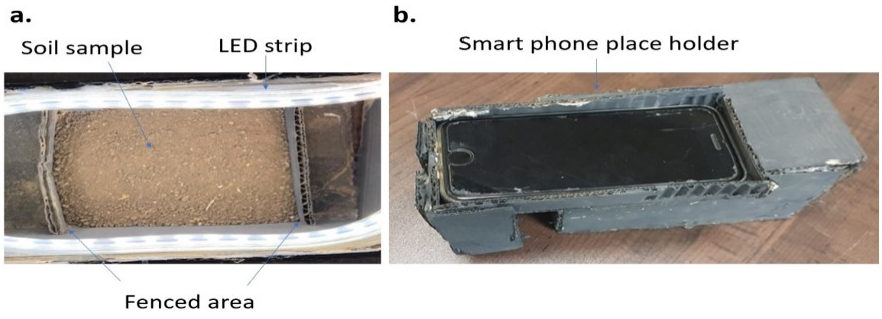


Figure 3.2: Views of **(a)** the dark chamber interior and **(b)** the chamber from the outside.

3.2.2 Image Pre-processing

Image analyses were performed in the `python 3.6` environment (Python software foundation, DE, USA) [27] using the `cv2` code package. The first step was to select the region of interest (ROI), the area in the image that shows only the sample (Fig 3.3). The ROI was chosen to be a square-shaped box of 2000x2000 pixels [22].

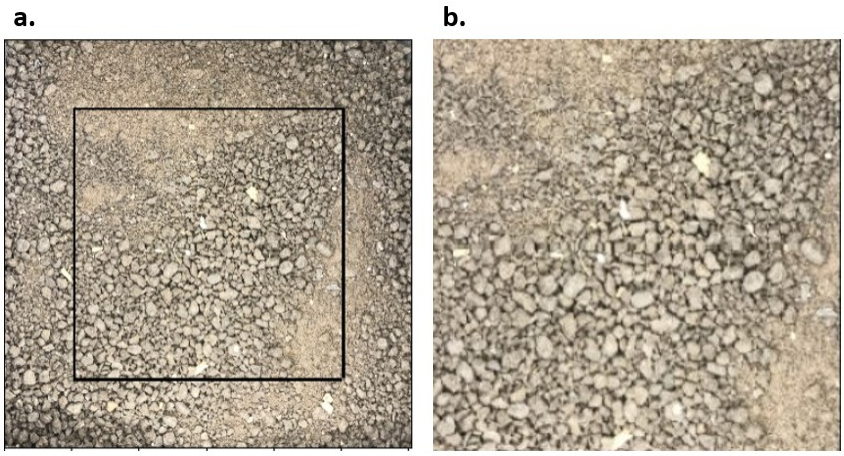


Figure 3.3: **a.** Original image taken with Iphone camera; the black box is the ROI. **b.** New image of the ROI only.

3.3 Soil Sampling and Texture Analysis

3.3.1 Choosing Sampling Points Based on Preliminary Soil Conductivity Surveys

Israeli et al. [48] suggest a method for choosing the optimal number of sampling points and their location based on field spatial variability. The optimization goal is to minimize the number of sampling points while fully representing the soil information spectrum and maximizing the distance between sampling locations. This approach is using an evolutionary multi objective algorithm and information-theory statistics for finding a satisfactory sample-size at which the rate of information gain for every additional sampling point is declining and then construct a sampling plan. Preliminary soil conductivity surveys were done during the years 2018-2020 by Israeli and co-workers [48] and a sampling plan that consists of 63 points was constructed.

3.3.2 Texture Analysis

As mentioned in Chapter 1, soil texture refers to the size of the soil particles and it is composed from three values: (1) sand (2 to 0.02 mm particles) (2) silt (0.02 to 0.002 mm particles) and (3) clay ($\leq$ 0.002 mm particles) that sum up to 100% [14]. Soil texture is an important property when examining soil properties like soil moisture, CEC (Cation exchange capacity), water penetrability and many others [49]. In this research, two measuring methods for soil texture were used:

The first is the traditional and more common, the Hydro Meter (HM) method. This method is based on Stoke's law and the calculation is done under the assumption that soil particles are spherical. Soil samples are floated in collagen solution after oven drying in 65 Celsius, grinding and sieving. The soil particles are shaken for 8 hours and the suspension density is measured in fixed time steps, the density is proportional to the particles concentration and every time step is related to specific particle size [6]. The samples were analyzed in the service lab in Newe Ya'ar.

The second method is the laser diffraction system (LDS) that was done using the Mastersizer (MS) v3.5-3000 (Malvern Panalytical Ltd. Malvern, U.K.). This method is based on measuring the scattered laser beam sent on desired soil sample. The device measures the angle at which the beam has scattered, which is inversely proportional to the soil particle size. Light detectors are spread around the measuring cell and the gleaned data are processed by a specialized software that calculates the particle size distribution (PSD). Methodological aspects of measurement parameters such as dispersion method (sonication) and conversion of light data to PSD (Fraunhofer theory) were chosen based on the work of [7]. According to a protocol developed in MIGAL lab, every sample examination was repeated 3 times, with 8 repetitive measurements. Every measurement provides the relative part of soil particles divided to sand, silt, and clay. For every 8 measurements the median was taken and the mean of the 3 repetitions was calculated, as the final value for that sample.

As explained in Chapter 1, the results from the two methods do not agree with each other and there is not a clear answer how to deal with this inconsistency, therefor ML models were built for predicting both separately and also the predictions performance are presented separately.

Chapter 4

Computational Steps

In this chapter the computational steps are explained in detail, including the adjustments that were made to fit the computer science theory to the real-life questions with the real-world data. Since our data set is small and homogeneous, it was necessary to focus on ways to squeeze as much insight from it as much as possible.

4.1 Objective Function Formulation

Our goal is to predict the trinary vector that represents the soil texture. The 3 values in this vector sum up to 100% (Eq. 1.1), which makes them dependent in each other, and the task of evaluating the model's success-rate becomes nontrivial. In regression problems, the commonly used metrics for success-rate are the Coefficient of Determinant (R^2) and the Root Mean Square Error (RMSE), having the former evaluating how well does the model fit the dependent variable, and having the latter evaluating the goodness of the fit (the mean distance between measured and calculated samples). However, by having 3 output values, rather than 1, the task becomes more complicated. Qi et al. [20] and Swetha et al. [22] decided to separate the performance measurements of the 3 values. They dealt with every texture class (sand, silt, and clay) as an independent variable and ignored the connection between them (sum to 100%). Every texture class was treated as a single regression target and the above metrics were calculated. Swetha et al. [22] added additional regression metrics known as Residual Prediction Deviation (RPD) and Ratio of Performance to Interquartile distance (RPIQ), both of them are used to evaluate the model validity, and the greater they are the better the model's predictive capacity becomes [50]. In their reports, both Qi et al. [20] and Swetha et al. [22] concentrated on the R square values, whereas the RMSE value was reported but less discussed. In this work the same metrics were used but more attention is given to the RMSE value. Since this work is directed toward PA, the RMSE value is much more explanatory on how well the model

performed than the R square value. Explicitly, the R square is calculated as follow:

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}}, \tag{4.1}$$

where SS_{res} is the residual sum of squares ($\sum(y_{predicted} - y_{measured})^2$) and the SS_{tot} is the total sum of squares ($\sum(y_{measured} - y_{mean})^2$). When looking at the above formulation, it is easy to see that a scattered data set will tend to have higher values, when compared to a dense data set, since the SS_{tot} will tend to be larger. The RMSE value tells us explicitly if the calculated is close enough to the measured. When looking at the texture triangle (Figure 4.1) it becomes much clearer what is the magnitude of tolerated error and what is just far off. Since many models were examined and compared it was essential to have a single value for a single model (rather than 3), the trinary RMSE values were summed for a single value Sum Root Mean Square Error (SRMSE) and that was the primary value to minimize during training. Importantly, the targeted objective function, subject to minimization, is the following:

$$minimize_{model} \quad SRMSE. \tag{4.2}$$

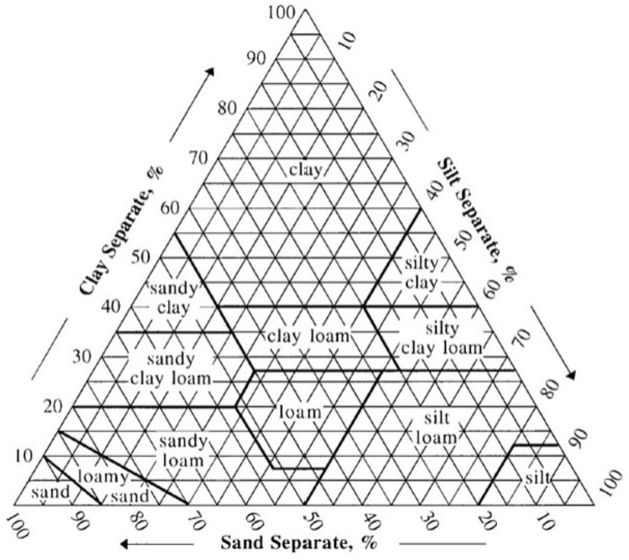


Figure 4.1: Soil texture triangle, a diagram that helps classifying soil texture class based on the percent of sand, silt and clay.

4.2 Techniques for Handling Small Data Sets

4.2.1 GA for Network Architecture

Vafaie et al. [51] explain how GAs can be utilized for improving an ML task. In this work, the GA was used for choosing the network architecture and the hyper-parameters of the learning

process. An individual in a population of candidate solutions is a set of characteristics like: the number of layers, number of neurons per layer, activation function, number of filters, number of epochs and batch size. There are no clear guidelines on how to set these parameters and they are likely to vary in different learning tasks. Usually, a trial-and-error procedure is done for a small number of iterations until obtaining a generalized model that is good enough because the computational costs are high and there is no time for testing many models. This procedure helped us in "squeezing" the most of our data set. The main motivation for using the GA was the relatively small size of our data set, 63 points. On one hand, the size of the data set allowed to create and examine many architectures due to the short training procedure. On the other hand, the size of the data set made the learning task difficult, and it was needed to find the best model architecture that fits this kind of learning [52].

4.2.2 Data Augmentation

Data Augmentation is a practice applied to deep learning models that possess large number of parameters. Training such models must include large data sets, but in real-world problems it is not always possible to collect enough data or it is too expensive and not feasible. The goal of this process is to enrich the data set, based on existing data and without collecting new samples; it is usually applied to imagery data. Manipulation on images, e.g., rotation, adding blur, changing contrast or focusing on part of the image, allow to generate new data instances for the training and therefore enable improvement of the model accuracy. In this study data augmentation was applied only for models that utilize imagery data. Every image was flipped 3 times: (1) horizontally (2) vertically and (3) horizontally + vertically (Figure 4.2), creating a data set 4 times bigger (total of 252 images) for the CNN training [53].

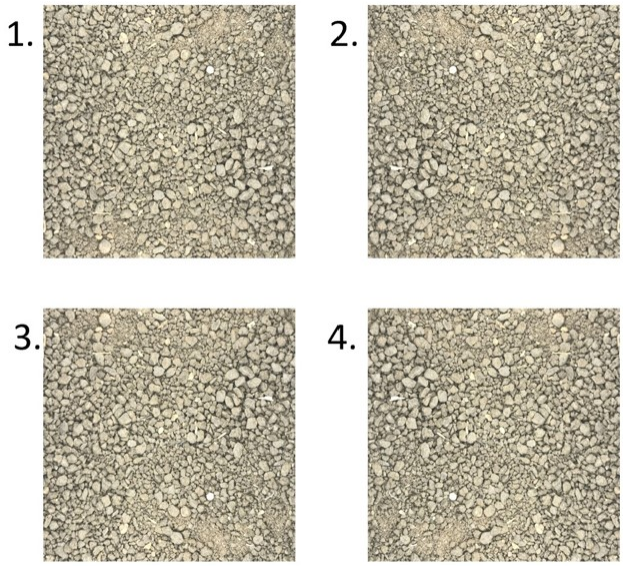


Figure 4.2: Image augmentation, creating 3 new images from the original.
1. The original image. 2. Flipped vertically. 3. Flipped horizontally. 4. Flipped vertically + horizontally.

4.3 Automated Image Partitioning

A CNN that takes as input images with resolution of 2000×2000 pixels requires too much memory and a too long computation on a standard computer. Wu et al. [54] discuss in detail the problems caused by large images and suggest to partition the image into smaller segments, to independently process those segments, to aggregate the results and produce a single prediction. After a trial-and-error procedure it was found that the maximal resolution that allowing computation in a reasonable time on a standard computer is around 150×150; eventually, it was decided to take a resolution of 100×100 since 150 is not a divisor of 2000. Altogether, a single image is partitioned into a set of segments (i.e., mini-images), that are separately fed to the CNN with the same label, and the final prediction is the mean sand, silt and clay of this entire set. Partitioning an image into segments assumes that the texture is uniform across all the segments. Importantly, since texture analysis in MS uses only 25 mg that represent an entire sample, this assumption seems reasonable. In order to validate this assumption, the maximal gap in prediction per a set of segments representing one soil sample was recorded.

4.4 Feature Selection Heuristic

High-dimensional data can be beneficial to an ML task but often times it can lead to over complexity, long computation times and even reduced accuracy [55]. Our data set is composed of different types of sensors and for some of them the measurements were done more than once, hence it is hard to tell which measurement is accurate and will help the learning and which is not. Redundant data will interfere with the learning process, especially in a small data set when the learning process is relatively short. Chandrashekar et al. [56] explain that the optimal way to decide which subset of features is best fitted would be to try the 2^n (n = number of features) subsets and choose the best one but usually this will take too long when n increases. In our work, $n = 10$ yields $2^{10} = 1024$, being too many subsets for generating an NN per each one of them. At the same time, since RF and LR models, which are much faster to build, are also considered in this work, all the 1024 options were tested for finding which features improve the models and which are not. The outcome of this exhaustive procedure was inconclusive: The two models were not in agreement on which subset is better so each feature was examined by itself, every subset without that feature was compared to the same subset containing it. Features that in most of the times have worsen results were excluded and a final subset were chosen and used later for constructing the NN and the CNN. Figure 4.3 summarizes the computational steps for creating neural network models.

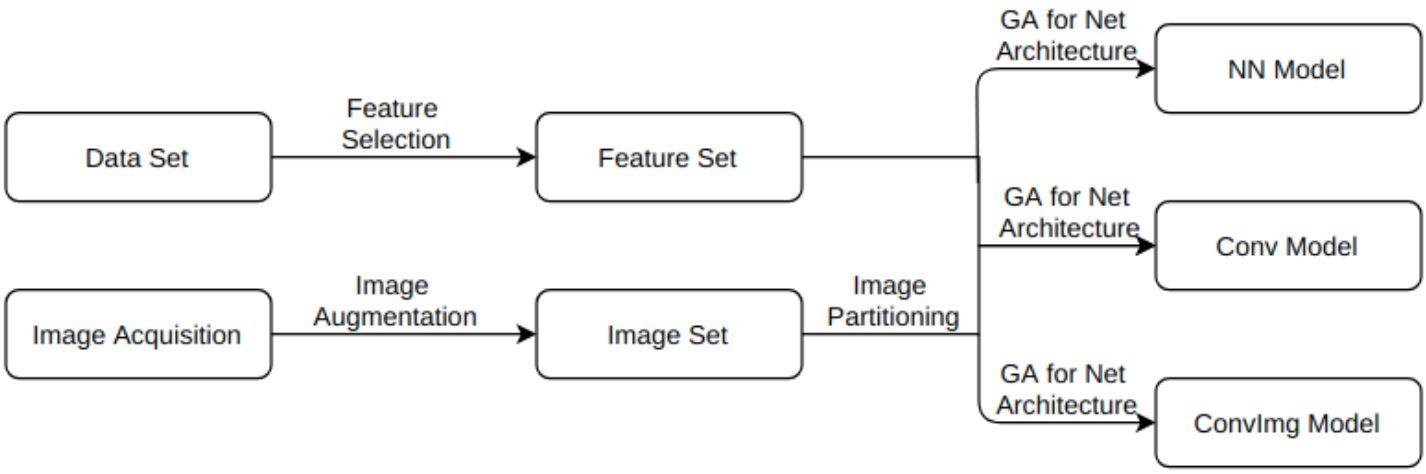


Figure 4.3: Flowchart that summarizes the steps for creating neural network models. The NN model uses only ancillary data features, ConvImg model uses only image data features and Conv model uses both.

Chapter 5

Experimental Results

In this chapter the experimental results of the learning task are presented, step by step. First, the data set statistics and the results of the Random model predictions, which serves as a control group. Then, the outcomes of the computational steps are presented and finally the results of the predictive models.

5.1 Descriptive Statistics

The descriptive statistics of soil properties for 63 soil samples are summarized in Table 5.1. The statistics are separated to MS data and HM data. For MS, sand content ranged from 2.88% to 20.58% with mean 7.16% and a standard deviation of 3.34%, silt content ranged from 57.51% to 65.25% with mean 61.52% and a standard deviation of 1.7% and clay content ranged from 21.41% to 35.49% with mean 31.31% and a standard deviation of 2.81%. According to the soil texture triangle (Figure 4.1), soil texture for 63 soil samples varied from silt loam to silty clay loam. For HM, sand content ranged from 17.5% to 41.2% with mean 25.82% standard deviation of 4.24%, silt content ranged from 12.4% to 28.4% with mean 20.48% standard deviation of 3.29% and clay content ranged from 38.5% to 59.2% with mean 53.72% and standard deviation of 5.09%. Soil texture varied from clay to clay loam.

Table 5.1: Descriptive statistics of soil properties for Master Sizer (MS) and Hydro Meter (HM) data.

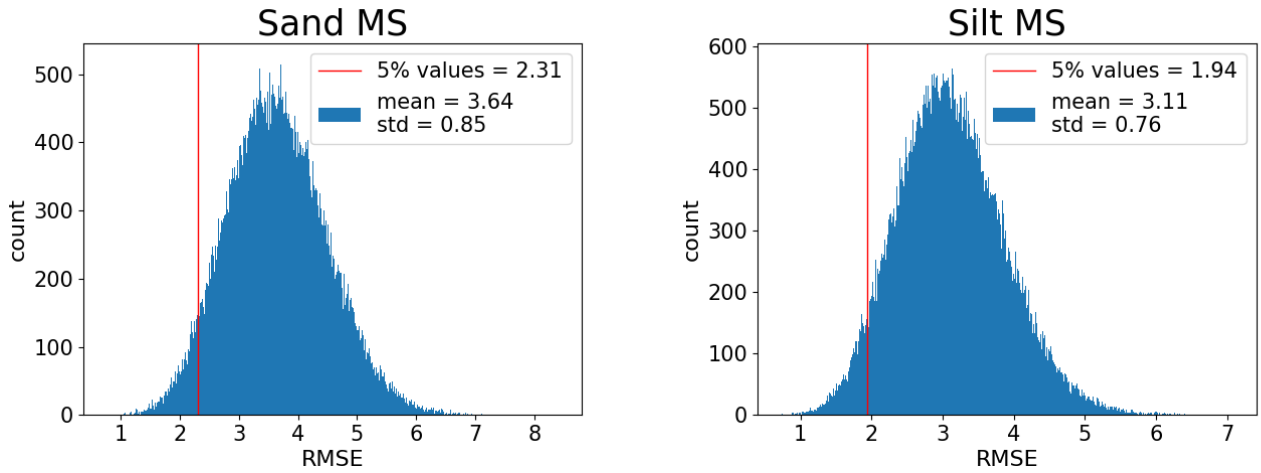
Statistic	Min	Max	Median	Mean	Max-Min	STD
Sand MS	2.88	20.58	6.22	7.16	17.7	3.34
Silt MS	57.51	65.25	61.47	61.52	7.74	1.7
Clay MS	21.41	35.49	31.91	31.31	14.08	2.81
Sand HM	17.5	41.2	25.4	25.82	23.7	4.24
Silt HM	12.4	28.4	20.4	20.48	16	3.29
Clay HM	38.5	59.2	54.5	53.74	20.7	5.09

5.2 Random Model

The Random model, that serves as a control group, was initiated 100,000 times and for every model the RMSE for every soil class and the SRMSE were calculated.

MS Results

The mean SRMSE is 9.72 (std-1.63) and 95% of values are greater then 7.13. The mean RMSE for sand, silt and clay is 3.64, 3.11, and 2.96 respectively (Figure 5.1).



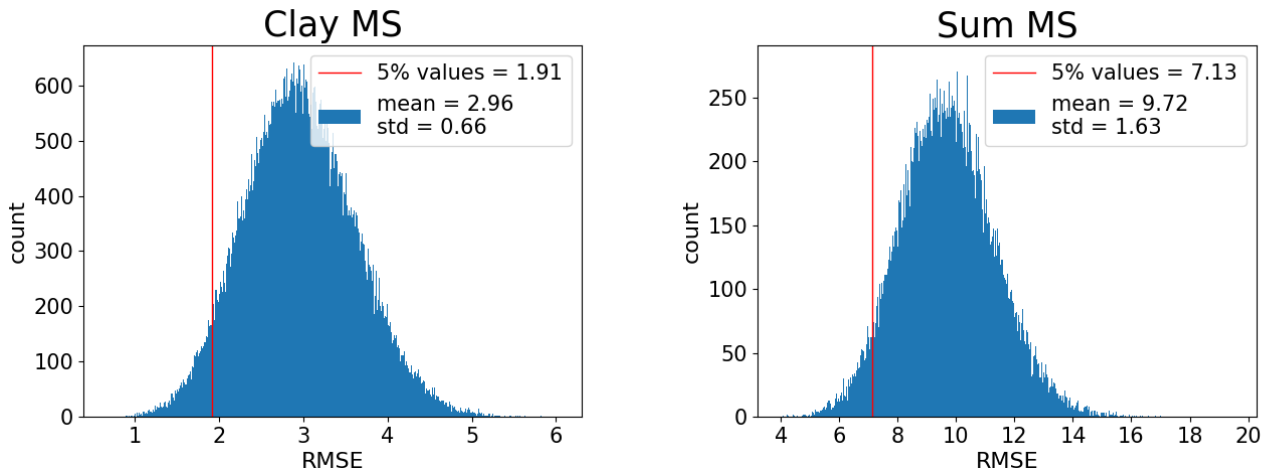
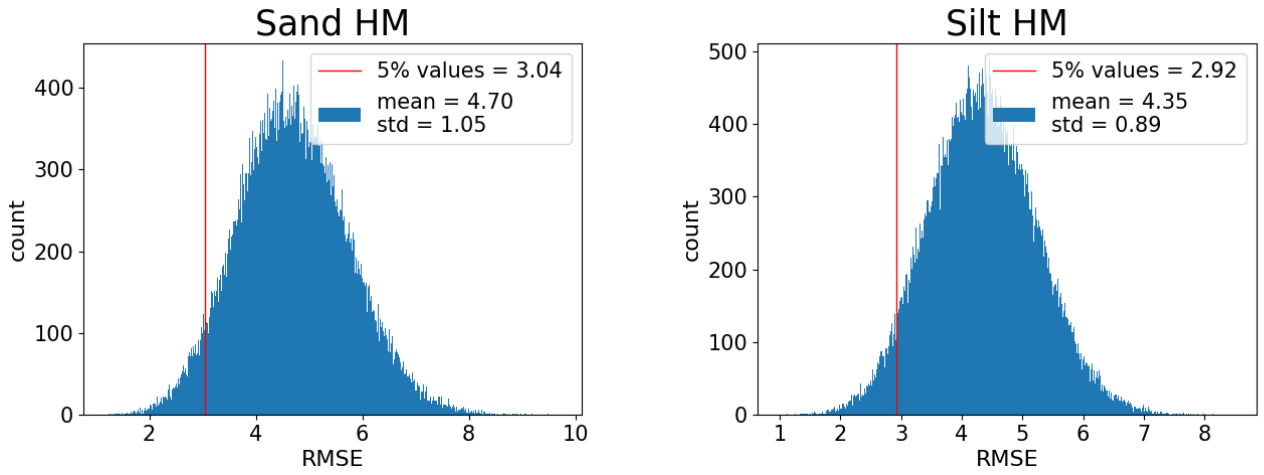


Figure 5.1: Random model mean RMSE histogram for 3 soil classes and mean SRMSE for MS data.

HM Results

The mean SRMSE score for texture data from HM is 13.84 (std-2.15) and 95% of values are greater then 10.42. The mean RMSE for sand, silt and clay in 4.7, 4.35, and 4.79 respectively (Figure 5.2).



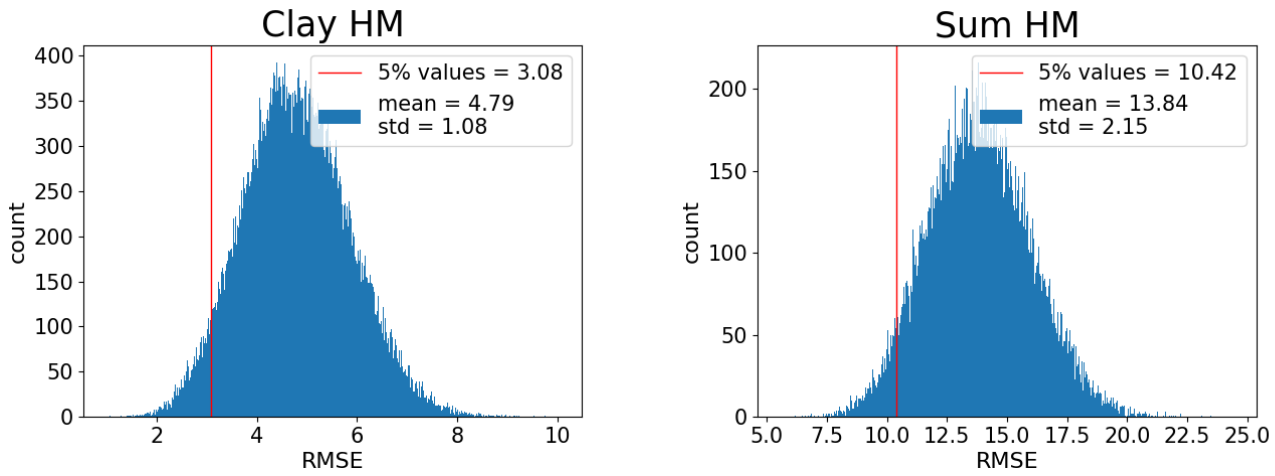


Figure 5.2: Random model mean RMSE histogram for 3 soil classes and mean SRMSE for HM data.

The goal of this step is to verify that the ML models accomplish actual learning. When using an ML model, it is not always clear what is a "good" outcome and of course it depends on the task and what is an acceptable error and what is not. In this study, that serves as a proof of concept, it was hard to define what is a good SRMSE value and it is not always clear how to compare to other studies' results, since the input data and its variability have high impact on the SRMSE. The results in this section are a test that will verify that some kind of learning actually took place. For example, if a model SRMSE score on MS data will turn out to be 8 it will be quiet clear that there was not any learning happening since it is a score that more than 5% of Random models have achieved without using the explanatory variables (the red vertical lines in Figures 5.1 and 5.2).

5.3 Feature Selection Heuristic

As explained in section 4.4, there are 10 candidates for the feature vector: 3 sequential and 1 manual conductivity measurements, Digital Terrain Model from LIDAR (DTM), Slope angle from GIS, 2 TRS measurements and 2 NDVI measurements. The conductivity measurements produce 4 values for every sampling point (ECa vertical and horizontal and MSa vertical and horizontal) but they are highly correlated (Figure 5.3), so only one of them (ECa vertical) was used.

The won bar in figures 5.4 and 5.5 is the number of times that a model with a subset of features **with** the feature beneath it had better results than a model with the same subset **without** it and the loss bar is its complement. For example, if a model that used DTM and slope as features showed **better** performance than a model that used only DTM, the **won** bar

for slope got +1 and and vice versa, if the model that used only slope was better than the first the lost bar for DTM got +1.

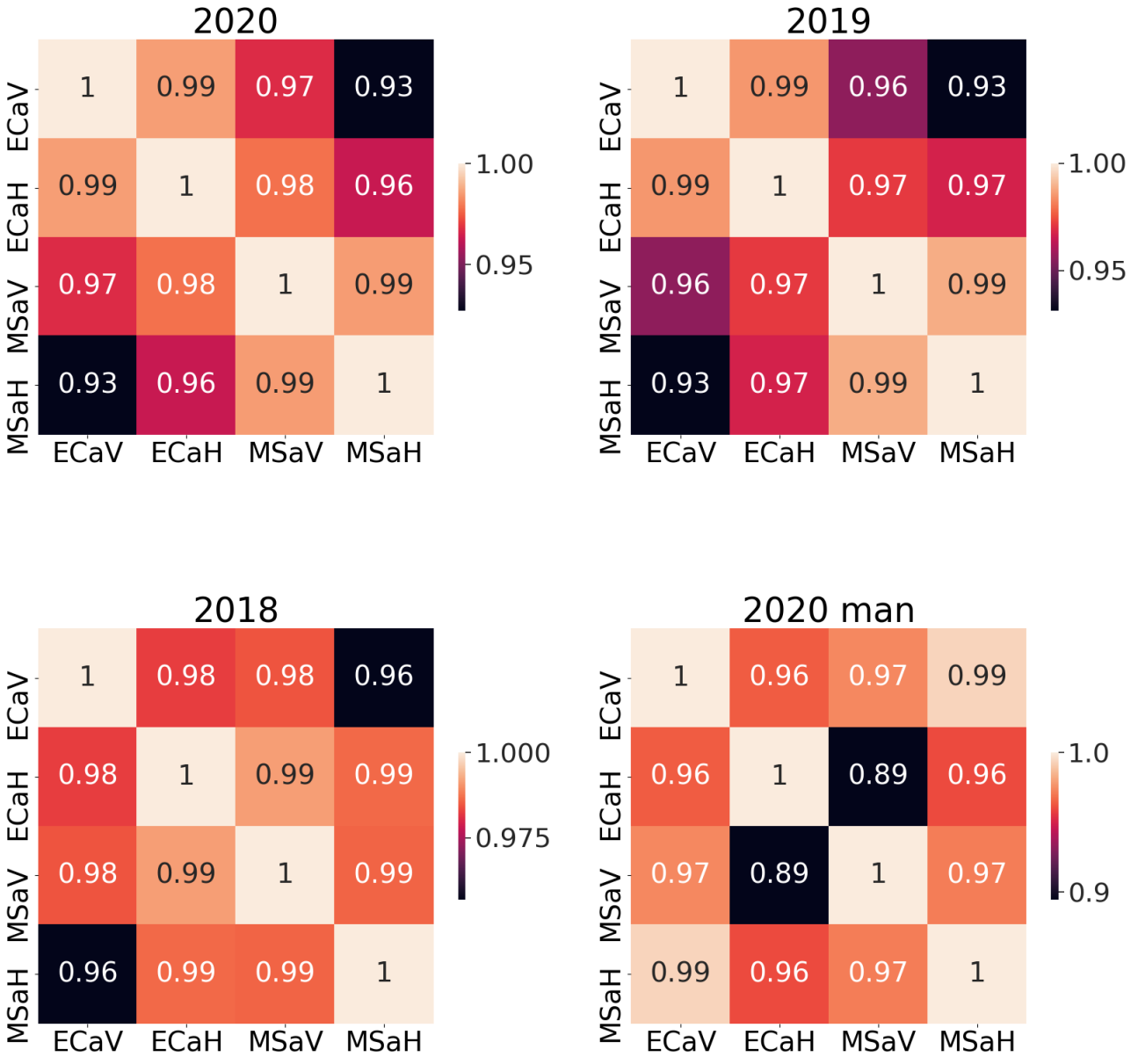


Figure 5.3: Correlations among the 4 values that are generated by each ECa measurement.

MS Features

The features that were chosen for building NN and CNN models for soil texture data acquired with MS are (the features whose *won bar* is higher than their *lost bar* in Figure 5.4): (1) ECa manual (2) ECa measurement from 2020 (3) ECa measurement from 2018 (4) DTM (5) Slope (6) NDVI measurement from 2019 (7) TRS measurement from 2020 and (8) TRS second measurement from 2020.

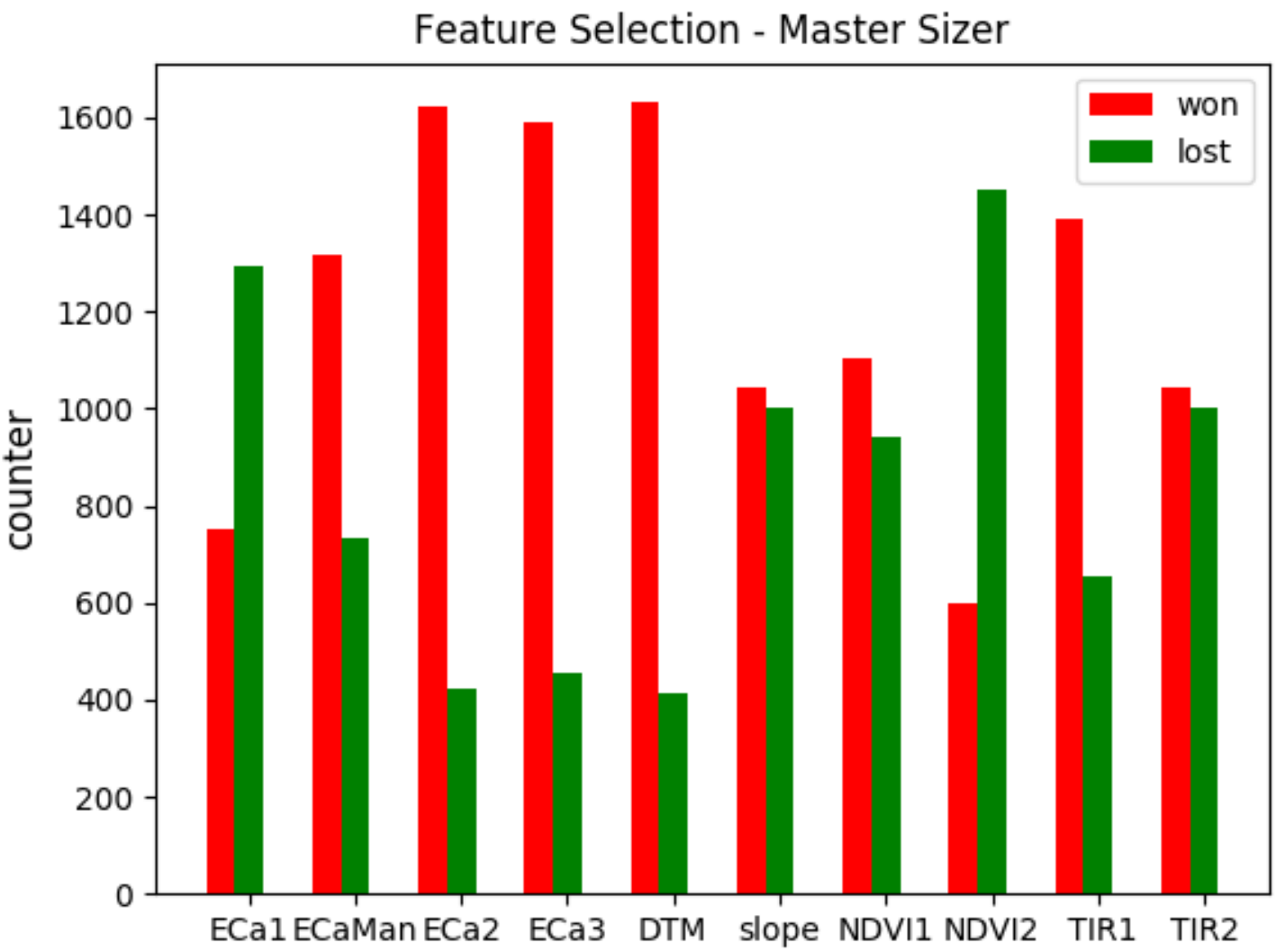


Figure 5.4: Feature selection for MS results, 8 features were selected.

HM Features

The features for soil texture data acquired with HM are (the features whose *won bar* is higher than their *lost bar* in Figure 5.5): (1) ECa measurement from 2019 (2) ECa manual (3) ECa measurement from 2018 (4) DTM (5) Slope and (6) TRS measurement from 2020.

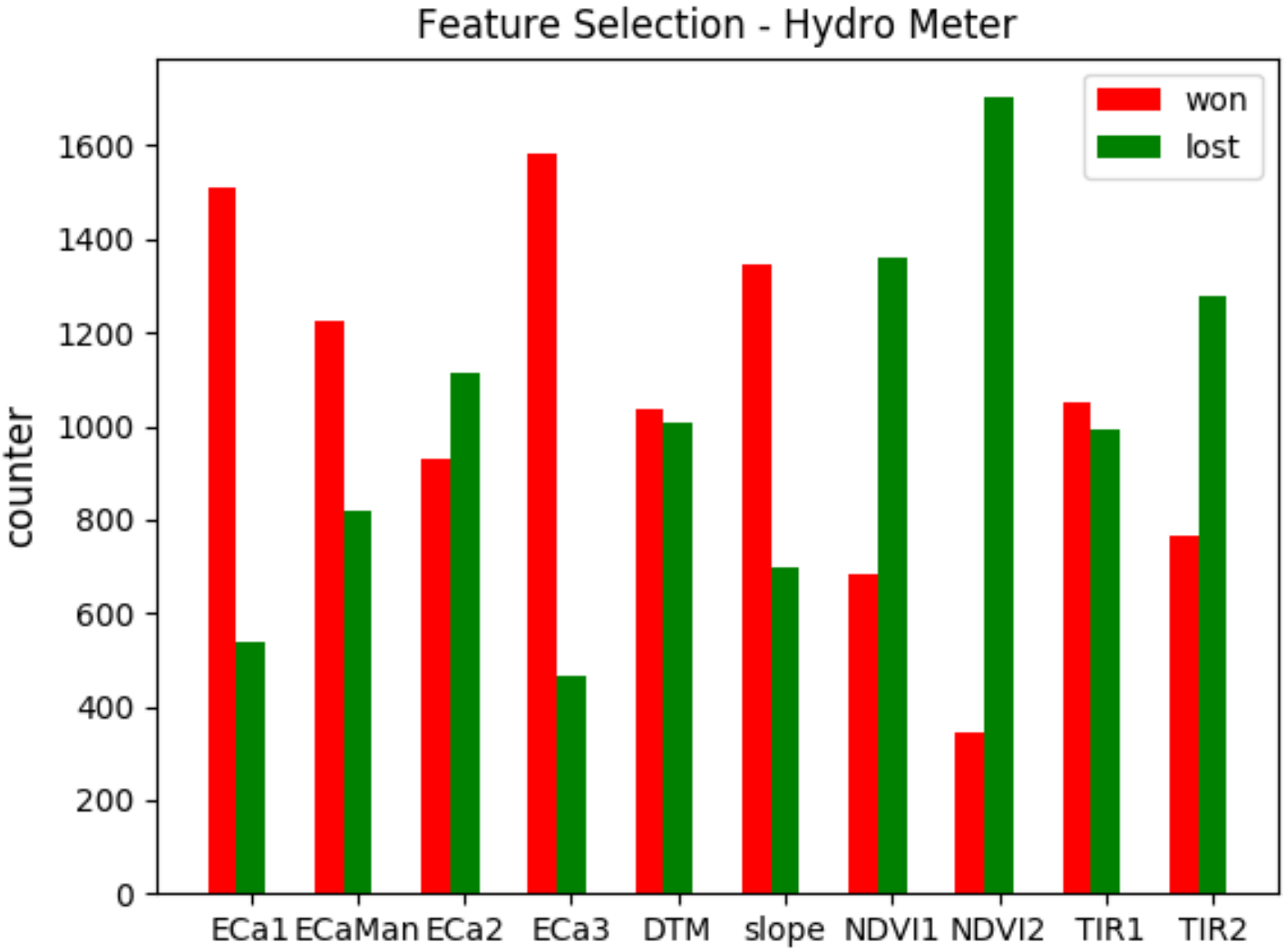


Figure 5.5: Feature selection for HM, 6 features were selected.

The features (1) ECa manual (2) ECa measurement from 2018 (3) DTM (4) Slope and (5) TRS measurement from 2020 were chosen for both MS and HM data which raises their reliability as explanatory features for soil texture predictions. Only one NDVI measurement was chosen and only for a single texture type (MS) – the won bar is only slightly higher than the lost bar, which raises a question whether it suits for the learning task. Since automation of the

entire process is a primary goal of the project the features for every texture type were chosen based on the condition that the won bar is higher than the lost bar regardless of the differences between bars or to their performance in the second texture type.

5.4 GA Results

As explained in section 4.2.1, a GA was used for 3 types of models (NN, CNN-image, CNN-image+numeric). A population of size 20 was initiated and the evolutionary process was done for 15 iterations, where those values were chosen after a trial-and-error procedure that measured the approximate duration for a single iteration (approximately 12 hours for the entire procedure for 1 model and 1 soil texture measuring type). The fitness values (i.e., SRMSE) of the best and the worst members in the population were recorded in every iteration (Table 5.2) and the top 5 members of the entire procedure were also recorded. The results in Table 5.2 show that the possible fitness values corresponding to this search space justify a procedure that is more systematic than trial-and-error but preliminary runs showed that increasing the number of generations did not pay off. Overall, it seems that the GA with this data set did not contribute much and that a simpler search algorithm could have obtained a similar outcome.

Table 5.2: GA fitness results of the best and worst members in the first and last generation.

texture type	model	generation	best	worst
MS	NN	0	4.28	11.62
	NN	14	3.54	7.65
	CNN-image	0	5.11	55.86
	CNN-image	14	4.59	8.75
	CNN-image+numeric	0	4.61	18.24
	CNN-image+numeric	14	3.64	6.67
HM	NN	0	7.88	14.6
	NN	14	7.16	12.41
	CNN-image	0	8.7	44.94
	CNN-image	14	8.23	15.13
	CNN-image+numeric	0	9.22	40.60
	CNN-image+numeric	14	8.34	19.42

5.5 Automated Image Partitioning Validation

The split to mini-images described in section 4.3 is not a trivial step and it requires some assessment, to validate that a trained CNN predicts about the same values to a set of mini-images that represents the same sample. A gap is defined to be the maximum difference in prediction between two images in mini-images set. Table 5.3 summarizes the mean gap in predictions for samples in test and train sets for sand, silt and clay. The Conv model predictions on MS data shows almost identical predictions (max gap is 1.3%) and on HM data the mean gap on clay is a bit high (3.48%) but it is still less than 1 STD (Table 5.1). The ConvImg predictions for MS data are still low for sand (1.15% $<$1 STD) but less homogeneous for silt with 5.26% gap (3.1 STD) and 5.58% gap (1.98 STD) for clay. For HM data the gaps are larger with 3.15% gap (1.34 STD), 6.31% gap (1.9 STD) for silt and 6.89% gap (1.3 STD) for clay. Since Conv model uses numeric data in addition to image data (which is identical to all mini-image in the same set) it is not surprising that the gaps are small, the bigger gaps for ConvImg undermines the validity of the splitting step or at least invites a further investigation on how to use this method better.

Table 5.3: Mean gap in test and train sets predictions on sand, silt and clay in mini-images set. A gap is the maximum difference in prediction between two images in mini-images set.

model	data set	mean gap sand (%)	mean gap silt (%)	mean gap clay (%)
Conv	MS test	0.19	1.15	0.58
	MS train	0.21	1.3	0.67
ConvImg	MS test	1.15	4.46	5.4
	MS train	1.03	5.26	5.58
Conv	HM test	1.85	1.25	3.47
	HM train	1.85	1.3	3.48
ConvImg	HM test	1.14	6.31	6.89
	HM train	3.15	2.85	5.77

5.6 ML Models Results

MS - models results

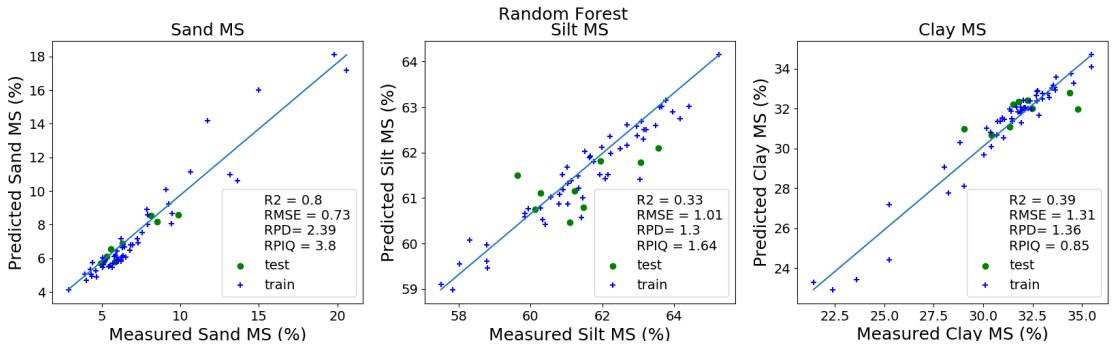
Table 5.4 summarizes Random and 5 ML models test set predictions statistics for silt, sand and clay (green dots in Figure 5.6). Sand predictions showed the best performance with RMSE ranging from 0.73 (RF) to 1.66 (ConvImg) and relatively high R^2, RPD and RPIQ values for RF (0.8, 2.39 and 3.8 respectively) and NN (0.69, 1.92 and 3.05 respectively) models. The performance of clay and silt predictions were not so good and showed higher range of RMSE values,

1.31 (RF) to 1.69 (ConvImg) for clay and 1.01 (RF) to 1.48 (LR) for silt, and higher R^2, RPD and RPIQ values 0.39, 1.36 and 0.85 respectively for clay and 0.33, 1.3 and 1.64 respectively for silt (RF). All the models showed significant improvement compared to the Random model, the SRMSE values of 4.59, 3.06, 3.54, 4.06 and 3.65 for ConvImg, RF, NN, LR and Conv, respectively, fall within the less-than-5%-interval of the random results (Figure 5.1), which indicates that some degree of learning was accomplished by all of them. Figure 5.6 shows regression models for RF and NN, the most successful models.

Comparing the prediction performance for clay, silt and sand presented here to those reported by Qi et al. [20] (RMSE values of 2.98, 6.01 and 5.92 and R^2 values of 0.71, 0.68 and 0.77 for clay, silt and sand respectively) and Swetha et al. [22] (RMSE values of 2.77, 2.94 and 2.9 and R^2 values of 0.98, 0.75 and 0.98 for clay, silt and sand respectively) show the difficulty to evaluate and to compare model performance. While the R^2 values of Qi and Swetha seem much better, the SRMSE values (14.91 for Qi and 8.61 for Swetha) tell a different story and these values might be more important when focusing on finding the right soil class (Figure 4.1). As discussed in section 4.1, the data in those studies is more scattered, the range of values for clay, silt and sand is 45.4, 35, and 28.8 respectively for Qi and 75, 19 and 79 respectively for Swetha whereas in this research it is only 14.08, 7.74 and 17.7 (Table 5.1 Max-Min), which leads to small R^2 values.

Table 5.4: Five ML models and Random model performance for predicting test set points from MS texture data set.

Target	Model	RMSE(%)	R^2	RPD	RPIQ
Silt MS	Random Model	3.11			
	ConvImg	1.23	0	1.05	1.33
	RF	**1.01**	0.33	1.3	1.64
	NN	**1.22**	0.02	1.07	1.35
	Linear Regression	1.48	-0.44	0.88	1.11
	Conv	1.27	-0.05	1.03	1.3
Sand MS	Random Model	3.64			
	ConvImg	1.66	0	1.05	1.68
	RF	**0.73**	0.8	2.39	3.8
	NN	**0.91**	0.69	1.92	3.05
	Linear Regression	1.06	0.59	1.66	2.63
	Conv	0.98	0.64	1.78	2.82
Clay MS	Random Model	2.96			
	ConvImg	1.69	0	1.06	0.66
	RF	**1.31**	0.39	1.36	0.85
	NN	**1.4**	0.3	1.27	0.79
	Linear Regression	1.51	0.19	1.18	0.73
	Conv	1.38	0.32	1.29	0.8
Sum MS	Random Model	9.72			
	ConvImg	4.59			
	RF	**3.06**			
	NN	**3.54**			
	Linear Regression	4.06			
	Conv	3.65			



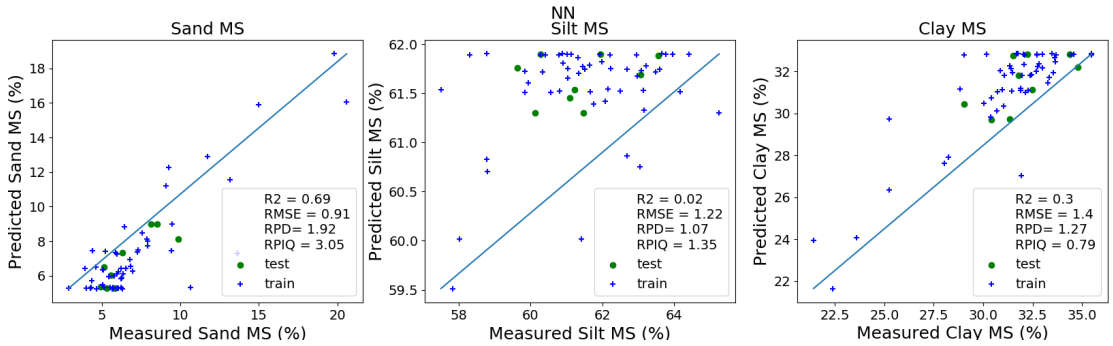


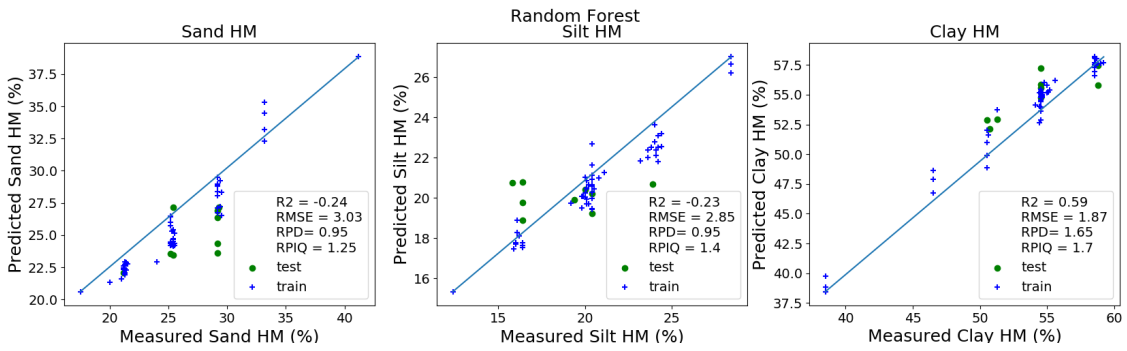
Figure 5.6: RF and NN predicted vs laboratory MS measured values. The green dots are for test set and the blue for train set. The blue line represents the 1:1 line.

HM - models results

Table 5.5 summarizes Random and 5 ML models test set predictions statistics for silt, sand and clay (green dots in Figure 5.7). All the models showed significant improvements when compared to the Random model, the SRMSE values of 8.23, 7.76, 7.16, 8.81 and 8.34 for ConvImg, RF, NN, LR and Conv respectively fall within the less-than-5%-interval of the random results (Figure 5.2), which indicates that some degree of learning was accomplished by all of them. In contrast, the low R^2 values and the fact that the HM data was more scattered than the MS data, ranges of 23.7, 16 and 20.7 against 14.08, 7.74 and 17.7 for sand, silt and clay respectively (table 5.1), tells that it was limited compared to the MS. The best models were NN and RF with RMSE values of 2.36, 2.47 and 1.87 for NN and 2.85, 3.03 and 1.87 for RF for silt, sand and clay respectively (Figure 5.7).

Table 5.5: Five ML models and Random model performance for predicting test set points from HM texture data set.

Target	Model	RMSE(%)	R^2	RPD	RPIQ
Silt HM	Random Model	4.35			
	ConvImg	2.62	-0.04	1.03	1.52
	RF	**2.85**	-0.23	0.95	1.4
	NN	**2.36**	0.14	1.14	1.68
	Linear Regression	2.76	-0.15	0.98	1.44
	Conv	2.48	0.06	1.09	1.6
Sand HM	Random Model	4.7			
	ConvImg	2.72	0	1.05	1.39
	RF	**3.03**	-0.24	0.95	1.25
	NN	**2.47**	0.17	1.16	1.53
	Linear Regression	3.36	-0.53	0.85	1.12
	Conv	2.87	-0.11	1	1.32
Clay HM	Random Model	4.79			
	ConvImg	2.88	0.02	1.07	1.1
	RF	**1.87**	0.59	1.65	1.7
	NN	**2.32**	0.37	1.33	1.37
	Linear Regression	2.68	0.15	1.15	1.19
	Conv	2.98	-0.04	1.03	1.07
Sum HM	Random Model	13.84			
	ConvImg	8.23			
	RF	**7.76**			
	NN	**7.16**			
	Linear Regression	8.81			
	Conv	8.34			



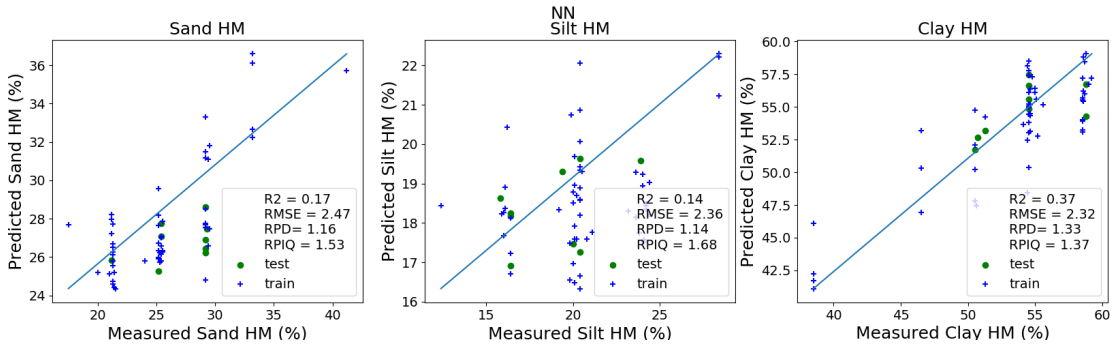


Figure 5.7: RF and NN predicted vs laboratory HM measured values. The green dots are for test set and the blue for train set. The blue line represents the 1:1 line.

Chapter 6

Discussion

In this study the task of predicting soil texture from ancillary data was examined. The main goal was to examine many ancillary data channels and many computer science techniques in order to decide what is the right strategy when approaching this task. Previous studies [4, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22] have shown that the various ancillary data channels are useful for predicting soil texture, but none of them considered combining altogether to exploit their accumulated information. In the case considered here, the small size of the data set and the fact that it was quite homogeneous were a challenge, therefore a substantial part of the work here focuses on methods to address these constraints.

An important part of this study is the discussion on how to measure success-rates of an ML model that predicts soil texture. Comparing to other studies' performance [20, 21, 22] is almost impossible since the data set characteristics have a strong impact on the predictions' quality. A Random model was proposed in Section 2.3.1 as a control group that gives a scale to how well is the model performance; it eventually helped in showing that the ML models have accomplished significant learning.

The ML models RF and NN with ancillary data showed the best performance for both texture measuring methods. It is a bit surprising that the RF achieved similar and even better performance than the NN, which is considered a much more powerful model. This may be because of the data set properties which are less suitable for NN that usually uses larger data sets. The models with imagery data showed less promising results but it is important to add that due to data set homogeneity the color differences were quite small and that images were not taken in high quality. The proposed method here to use CNN with two types of inputs (ancillary and imagery data) can be beneficial since previous work [20, 21, 22] have showed that soil texture can be predicted from soil images and also this study showed a significant improvement in performance compared to a Random model. In general, there is an inherit advantage in using ancillary data over imagery data since all ancillary data sources are sequential (Except for the manual ECa measurement) and therefore a detailed map of predicted soil texture for the entire

field can be created while the proposed method to use image data in this study do not allow it because it requires manual sampling in specific locations.

Notably, the open discussion on the differences between measurement methods of soil texture can not be resolved in this study, but in this work models with MS data exhibited better performance than HM data. It is important to note that the models for both methods showed significant learning, when compared to the Random model, so it can be concluded that both methods measure a soil feature that can be learned from ancillary data but it is not clear if they actually measure the exact same thing.

An important part of this study, which was not fully expressed so far, is the Python code implementation under the hood. The main principal for the code development was to make it accessible for further use, to add data or to add ancillary data channels. It is clear that in order to serve as a real tool for agronomists, this project must include more data sources and it will have to be an ongoing research. The work presented here could serve as a solid ground for this project.

Next, we propose a possible direction of future research.

6.1 Future Work

Future work that would like to expand this project should consider using satellite data which is becoming more accessible and with better resolution, this might allow using color features without the need to sample the soil.

In this work the ML models with imagery data used the raw images without any processing before feeding it to the model. It was done according to a rule of thumb in ML that states that with sufficient learning the model will learn by itself what is the right way to process the data and that it should remain as flexible as possible. The data set size in this work did not allow a long training phase so it might be beneficial to do prepossessing and to extract color features from the images as an alternative route. This step would require more processing work but will lead to simpler and faster models that are likely to require a shorter training time.

Bibliography

[1] Hazell, P. and Wood, S. “Drivers of change in global agriculture”. In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 363.1491 (2008), pp. 495–515. ISSN: 09628452. DOI: 10.1098/rstb.2007.2166.

[2] Kang, M. S. and Banga, S. S. “Global Agriculture and Climate Change”. In: *Journal of Crop Improvement* 27.6 (2013), pp. 667–692. ISSN: 15427528. DOI: 10.1080/15427528.2013.845051.

[3] Pierce, F. J. and Nowak, P. “Aspects of Precision Agriculture”. In: *Advances in Agronomy* 67.C (1999), pp. 1–85. ISSN: 00652113. DOI: 10.1016/S0065-2113(08)60513-1.

[4] Kelley, J., Higgins, C. W., Pahlow, M., and Noller, J. “Mapping soil texture by electromagnetic induction: A case for regional data coordination”. In: *Soil Science Society of America Journal* 81.4 (2017), pp. 923–933. ISSN: 14350661. DOI: 10.2136/sssaj2016.12.0432.

[5] Katerji, N. and Mastrorilli, M. “The effect of soil texture on the water use efficiency of irrigated crops: Results of a multi-year experiment carried out in the Mediterranean region”. In: *European Journal of Agronomy* 30.2 (2009), pp. 95–100. ISSN: 11610301. DOI: 10.1016/j.eja.2008.07.009.

[6] Gee, G. W. and Bauder, J. W. “Particle size analysis by hydrometer a simplified method for routine textural analysis and a sensitivity test of measurement parameters”. In: *Soil Science Society American Journal* 43.5 (1979), pp. 1004–1007. URL: https://www.researchgate.net/publication/250125222_Particle_Size_Analysis_by_Hydrometer_A_Simplified_Method_for_Routine_Textural_Analysis_and_a_Sensitivity_Test_of_Measurement_Parameters1.

[7] Ryzak, M. and Bieganowski, A. “Methodological aspects of determining soil particle-size distribution using the laser diffraction method”. In: *Journal of Plant Nutrition and Soil Science* 174.4 (2011), pp. 624–633. ISSN: 14368730. DOI: 10.1002/jpln.201000255.

[8] Al-Hashemi, H. M., Al-Amoudi, O. S., Yamani, Z. H., Mustafa, Y. M., and Ahmed, H. U. R. “The validity of laser diffraction system to reproduce hydrometer results for grain size analysis in geotechnical applications”. In: *PloS one* 16.1 (2021), e0245452. ISSN: 19326203.

DOI: 10.1371/journal.pone.0245452. URL: http://dx.doi.org/10.1371/journal.pone.0245452.

[9] Faé, G. S., Montes, F., Bazilevskaya, E., Añó, R. M., and Kemanian, A. R. "Making Soil Particle Size Analysis by Laser Diffraction Compatible with Standard Soil Texture Determination Methods". In: *Soil Science Society of America Journal* 83.4 (2019), pp. 1244–1252. ISSN: 0361-5995. DOI: 10.2136/sssaj2018.10.0385.

[10] Eshel, G., Levy, G. J., Mingelgrin, U., and Singer, M. J. "Critical Evaluation of the Use of Laser Diffraction for Particle-Size Distribution Analysis". In: *Soil Science Society of America Journal* 68.3 (2004), pp. 736–743. ISSN: 03615995. DOI: 10.2136/sssaj2004.7360.

[11] Eshel, G., Warrington, D. N., and Levy, G. J. "Comments on "Inherent factors limiting the use of laser diffraction for determining particle size distributions of soil and related samples" by Kowalenko and Babuin (Geoderma 2013; 193-194: 22-28)". In: *Geoderma* 226-227.1 (2014), pp. 418–419. ISSN: 00167061. DOI: 10.1016/j.geoderma.2014.02.024. URL: http://dx.doi.org/10.1016/j.geoderma.2014.02.024.

[12] Carroll, Z. L. and Oliver, M. A. "Exploring the spatial relations between soil physical properties and apparent electrical conductivity". In: *Geoderma* 128.3-4 SPEC. ISS. (2005), pp. 354–374. ISSN: 00167061. DOI: 10.1016/j.geoderma.2005.03.008.

[13] Heil, K. and Schmidhalter, U. "Characterisation of soil texture variability using the apparent soil electrical conductivity at a highly variable site". In: *Computers and Geosciences* 39 (2012), pp. 98–110. ISSN: 00983004. DOI: 10.1016/j.cageo.2011.06.017. URL: http://dx.doi.org/10.1016/j.cageo.2011.06.017.

[14] Riese, F. M. and Keller, S. "Soil texture classification with 1d convolutional neural networks based on hyperspectral data". In: *ISPRS Annals of the Photogrammetry,Remote Sensing and Spatial Information Sciences* 4.2/W5 (2019), pp. 615–621. ISSN: 21949050. DOI: 10.5194/isprs-annals-IV-2-W5-615-2019. arXiv: 1901.04846.

[15] Casa, R., Castaldi, F., Pascucci, S., Palombo, A., and Pignatti, S. "A comparison of sensor resolution and calibration strategies for soil texture estimation from hyperspectral remote sensing". In: *Geoderma* 197-198 (2013), pp. 17–26. ISSN: 00167061. DOI: 10.1016/j.geoderma.2012.12.016. URL: http://dx.doi.org/10.1016/j.geoderma.2012.12.016.

[16] Khanal, S., Fulton, J., and Shearer, S. "An overview of current and potential applications of thermal remote sensing in precision agriculture". In: *Computers and Electronics in Agriculture* 139 (2017), pp. 22–32. ISSN: 01681699. DOI: 10.1016/j.compag.2017.05.001. URL: http://dx.doi.org/10.1016/j.compag.2017.05.001.

[17] Müller, B., Bernhardt, M., Jackisch, C., and Schulz, K. “Estimating spatially distributed soil texture using time series of thermal remote sensing - A case study in central Europe”. In: *Hydrology and Earth System Sciences* 20.9 (2016), pp. 3765–3775. ISSN: 16077938. DOI: 10.5194/hess-20-3765-2016.

[18] Wang, D. C., Zhang, G. L., Zhao, M. S., Pan, X. Z., Zhao, Y. G., Li, D. C., and Macmillan, B. “Retrieval and mapping of soil texture based on land surface diurnal temperature range data from MODIS”. In: *PLoS ONE* 10.6 (2015), pp. 1–14. ISSN: 19326203. DOI: 10.1371/journal.pone.0129977.

[19] Greve, M. H., Kheir, R. B., Greve, M. B., and Bøcher, P. K. “Quantifying the ability of environmental parameters to predict soil texture fractions using regression-tree model with GIS and LIDAR data: The case study of Denmark”. In: *Ecological Indicators* 18 (2012), pp. 1–10. ISSN: 1470160X. DOI: 10.1016/j.ecolind.2011.10.006. URL: http://dx.doi.org/10.1016/j.ecolind.2011.10.006.

[20] Qi, L., Adamchuk, V., Huang, H. H., Leclerc, M., Jiang, Y., and Biswas, A. “Proximal sensing of soil particle sizes using a microscope-based sensor and bag of visual words model”. In: *Geoderma* 351 (Oct. 2019), pp. 144–152. ISSN: 00167061. DOI: 10.1016/j.geoderma.2019.05.020.

[21] Morais, P. A. d. O., Souza, D. M. de, Carvalho, M. T. d. M., Madari, B. E., and Oliveira, A. E. de. “Predicting soil texture using image analysis”. In: *Microchemical Journal* 146.October 2018 (2019), pp. 455–463. ISSN: 0026265X. DOI: 10.1016/j.microc.2019.01.009. URL: https://doi.org/10.1016/j.microc.2019.01.009.

[22] Swetha, R. K., Bende, P., Singh, K., Gorthi, S., Biswas, A., Li, B., Weindorf, D. C., and Chakraborty, S. “Predicting soil texture from smartphone-captured digital images and an application”. In: *Geoderma* 376.June (2020), p. 114562. ISSN: 00167061. DOI: 10.1016/j.geoderma.2020.114562. URL: https://doi.org/10.1016/j.geoderma.2020.114562.

[23] Liakos, K. G., Busato, P., Moshou, D., Pearson, S., and Bochtis, D. “Machine learning in agriculture: A review”. In: *Sensors (Switzerland)* 18.8 (2018), pp. 1–29. ISSN: 14248220. DOI: 10.3390/s18082674.

[24] Fradkov, A. L. “Early history of machine learning”. In: *IFAC-PapersOnLine* 53.2 (2020), pp. 1385–1390. ISSN: 24058963. DOI: 10.1016/j.ifacol.2020.12.1888.

[25] Naqa, I. E. and Murphy, M. J. “Machine Learning in Radiation Oncology”. In: *Machine Learning in Radiation Oncology* (2015), pp. 3–11. DOI: 10.1007/978-3-319-18305-3.

[26] Dimitriadis, S. and Goumopoulos, C. “Applying machine learning to extract new knowledge in precision agriculture applications”. In: *Proceedings - 12th Pan-Hellenic Conference on Informatics, PCI 2008* (2008), pp. 100–104. DOI: 10.1109/PCI.2008.30.

[27] Van Rossum, G. and Drake, F. L. *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009. ISBN: 1441412697.

[28] Myles, A. J., Feudale, R. N., Liu, Y., Woody, N. A., and Brown, S. D. "An introduction to decision tree modeling". In: *Journal of Chemometrics* 18.6 (2004), pp. 275–285. ISSN: 08869383. DOI: 10.1002/cem.873.

[29] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[30] Uyanık, G. K. and Güler, N. "A Study on Multiple Linear Regression Analysis". In: *Procedia - Social and Behavioral Sciences* 106 (2013), pp. 234–240. ISSN: 18770428. DOI: 10.1016/j.sbspro.2013.12.027.

[31] Marill, K. A. "Advanced Statistics: Linear Regression, Part II: Multiple Linear Regression". In: *Academic Emergency Medicine* 11.1 (2004), pp. 94–102. ISSN: 10696563. DOI: 10.1197/j.aem.2003.09.006.

[32] Khan, G. M. "Artificial neural network (ANNs)". In: *Studies in Computational Intelligence* 725 (2018), pp. 39–55. ISSN: 1860949X. DOI: 10.1007/978-3-319-67466-7_4.

[33] Zhang, S., Xia, Y., and Zheng, W. "A complex-valued neural dynamical optimization approach and its stability analysis". In: *Neural Networks* 61 (2015), pp. 59–67. ISSN: 18792782. DOI: 10.1016/j.neunet.2014.10.003. URL: http://dx.doi.org/10.1016/j.neunet.2014.10.003.

[34] Zhu, X. X., Tuia, D., Mou, L., Xia, G.-S., Zhang, L., Xu, F., and Fraundorfer, F. "Deep learning in remote sensing: a review". In: 41501462 (2017), pp. 1–60. DOI: 10.1109/MGRS.2017.2762307. arXiv: 1710.03959. URL: http://arxiv.org/abs/1710.03959%5C%0Ahttp://dx.doi.org/10.1109/MGRS.2017.2762307.

[35] Albawi, S., Mohammed, T. A. M., and Alzawi, S. "Layers of a Convolutional Neural Network". In: *Ieee* (2017).

[36] Liu, Q., Zhang, N., Yang, W., Wang, S., Cui, Z., Chen, X., and Chen, L. "A review of image recognition with deep convolutional neural network". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 10361 LNCS (2017), pp. 69–80. ISSN: 16113349. DOI: 10.1007/978-3-319-63309-1_7.

[37] Kamilaris, A. and Prenafeta-Boldú, F. X. "A review of the use of convolutional neural networks in agriculture". In: *Journal of Agricultural Science* 156.3 (2018), pp. 312–322. ISSN: 14695146. DOI: 10.1017/S0021859618000436.

[38] Ni, C. and Ma, X. “Prediction of wave power generation using a Convolutional Neural Network with multiple inputs”. In: *Energies* 11.8 (2018), pp. 1–18. ISSN: 19961073. DOI: 10.3390/en11082097.

[39] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Man, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Vigas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: https://www.tensorflow.org/.

[40] Chollet, F. et al. *Keras*. 2015. URL: https://github.com/fchollet/keras.

[41] Arora, J. S., Elwakeil, O., Chahande, A., and Hsieh, C. “Review Paper Global optimization m e t h o d s for engineering applications : a review”. In: *Structural Optimization 9,* 9 (1995), pp. 137–159.

[42] Tereshatov, V. V. and Senichev, V. Y. *Natural computing algorithms*. Vol. 132. 7. 2015, n/a–n/a. ISBN: 3662436302. DOI: wiley.com/10.1002/app.41481.

[43] Magalhães, J. P. and Löh, A. “Generic programming”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8324 LNCS.November 1994 (2014), pp. 216–231. ISSN: 03029743. DOI: 10.1007/978-3-319-04132-2_15.

[44] Ping, G., Chunbo, X., Yi, C., Jing, L., and Yanqing, L. “Adaptive ant colony optimization algorithm”. In: *Proceedings - 2014 International Conference on Mechatronics and Control, ICMC 2014* 4.3 (2015), pp. 95–98. DOI: 10.1109/ICMC.2014.7231524.

[45] Corwin, D. L. and Scudiero, E. *Review of soil salinity assessment for agriculture across multiple scales using proximal and/or remote sensors*. 1st ed. Vol. 158. Elsevier Inc., 2019, pp. 1–130. ISBN: 9780128174128. DOI: 10.1016/bs.agron.2019.07.001. URL: http://dx.doi.org/10.1016/bs.agron.2019.07.001.

[46] Lesch, S. M., Corwin, D. L., and Robinson, D. A. “Apparent soil electrical conductivity mapping as an agricultural management tool in arid zone soils”. In: *Computers and Electronics in Agriculture* 46.1-3 SPEC. ISS. (2005), pp. 351–378. ISSN: 01681699. DOI: 10.1016/j.compag.2004.11.007.

[47] Huang, S., Tang, L., Hupy, J. P., Wang, Y., and Shao, G. “A commentary review on the use of normalized difference vegetation index (NDVI) in the era of popular remote sensing”. In: *Journal of Forestry Research* 32.1 (2021), pp. 1–6. ISSN: 19930607. DOI: 10.1007/s11676-020-01155-1. URL: https://doi.org/10.1007/s11676-020-01155-1.

[48] Israeli, A., Emmerich, M., Litaor, M. I., and Shir, O. M. “Statistical learning in soil sampling design aided by pareto optimization”. In: *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-2019, New York, NY, USA, ACM Press (2019) 1198–1205* (2019), pp. 1198–1205. DOI: 10.1145/3321707.3321809.

[49] Friedman, S. P. “Soil properties influencing apparent electrical conductivity: A review”. In: *Computers and Electronics in Agriculture* 46.1-3 SPEC. ISS. (2005), pp. 45–70. ISSN: 01681699. DOI: 10.1016/j.compag.2004.11.001.

[50] Chakraborty, S., Weindorf, D. C., Deb, S., Li, B., Paul, S., Choudhury, A., and Ray, D. P. “Rapid assessment of regional soil arsenic pollution risk via diffuse reflectance spectroscopy”. In: *Geoderma* 289 (2017), pp. 72–81. ISSN: 00167061. DOI: 10.1016/j.geoderma.2016.11.024. URL: http://dx.doi.org/10.1016/j.geoderma.2016.11.024.

[51] Vafaie, H. and De Jong, K. “Genetic algorithms as a tool for feature selection in machine learning”. In: *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI* 1992-Novem (1992), pp. 200–203. ISSN: 10823409. DOI: 10.1109/TAI.1992.246402.

[52] Ritchie, M. D., White, B. C., Parker, J. S., Hahn, L. W., and Moore, J. H. “Optimization of neural network architecture using genetic programming improves detection and modelling of gene-gene interactions in studies of human diseases”. In: *BMC Bioinformatics* 4 (2003), pp. 1–14. ISSN: 14712105. DOI: 10.1186/1471-2105-4-28.

[53] Salehinejad, H., Valaee, S., Dowdell, T., and Barfett, J. “Image augmentation using radial transform for training deep neural networks Department of Electrical & Computer Engineering , University of Toronto , Toronto , Canada Department of Medical Imaging , St . Michael ’ s Hospital , University of Toronto , Toro”. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2018), pp. 3016–3020.

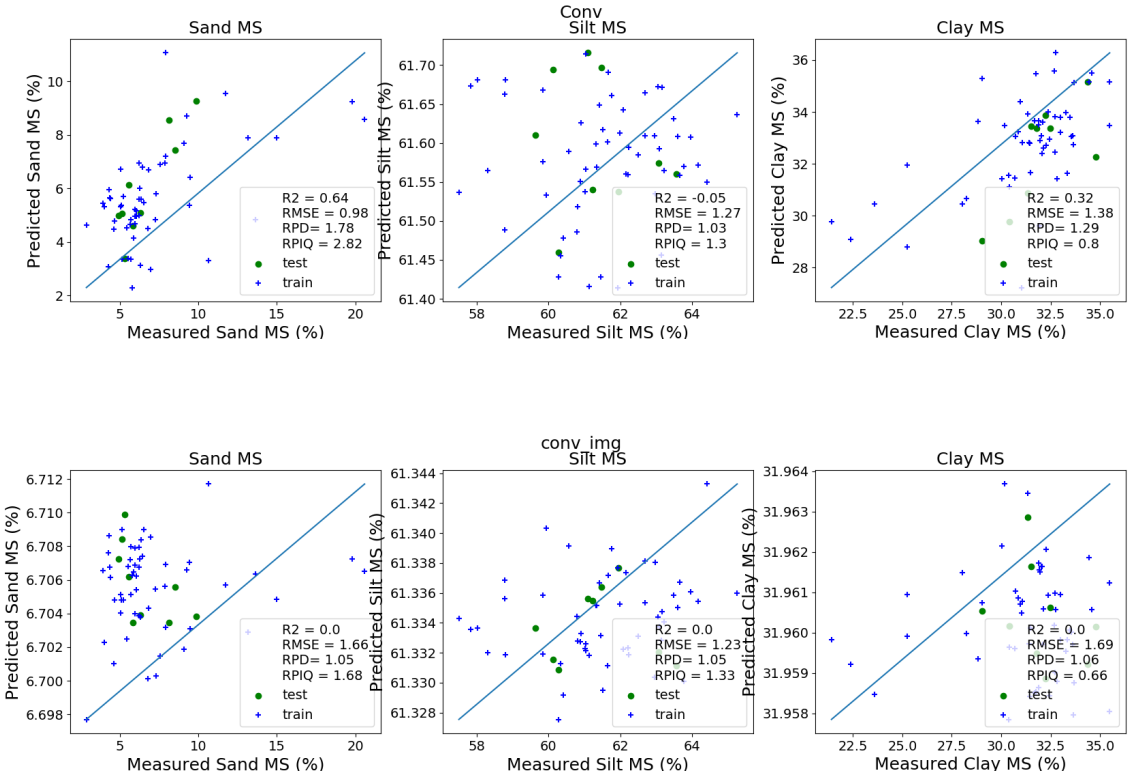
[54] Wu, S., Zhang, M., Chen, G., and Chen, K. “A new approach to compute CNNs for extremely large images”. In: *International Conference on Information and Knowledge Management, Proceedings* Part F1318 (2017), pp. 39–48. DOI: 10.1145/3132847.3132872.

[55] Cai, J., Luo, J., Wang, S., and Yang, S. “Feature selection in machine learning: A new perspective”. In: *Neurocomputing* 300 (2018), pp. 70–79. ISSN: 18728286. DOI: 10.1016/j.neucom.2017.11.077. URL: https://doi.org/10.1016/j.neucom.2017.11.077.

[56] Chandrashekar, G. and Sahin, F. “A survey on feature selection methods”. In: *Computers and Electrical Engineering* 40.1 (2014), pp. 16–28. ISSN: 00457906. DOI: 10.1016/j.compeleceng.2013.11.024. URL: http://dx.doi.org/10.1016/j.compeleceng.2013.11.024.

Chapter 7

Appendix



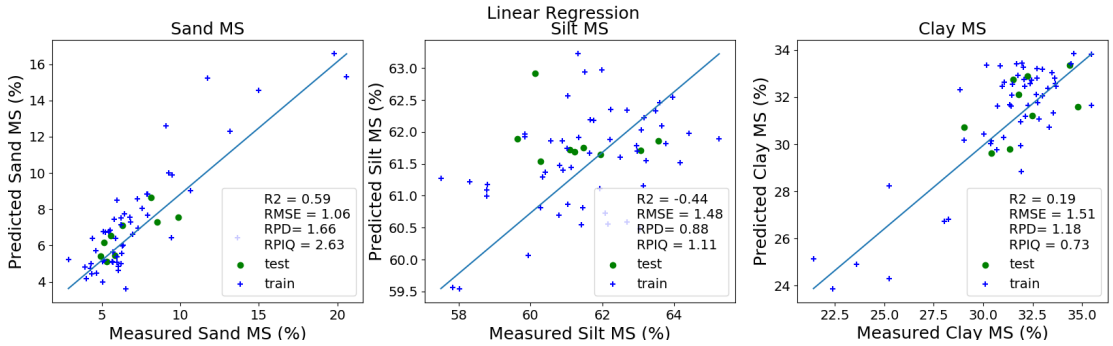
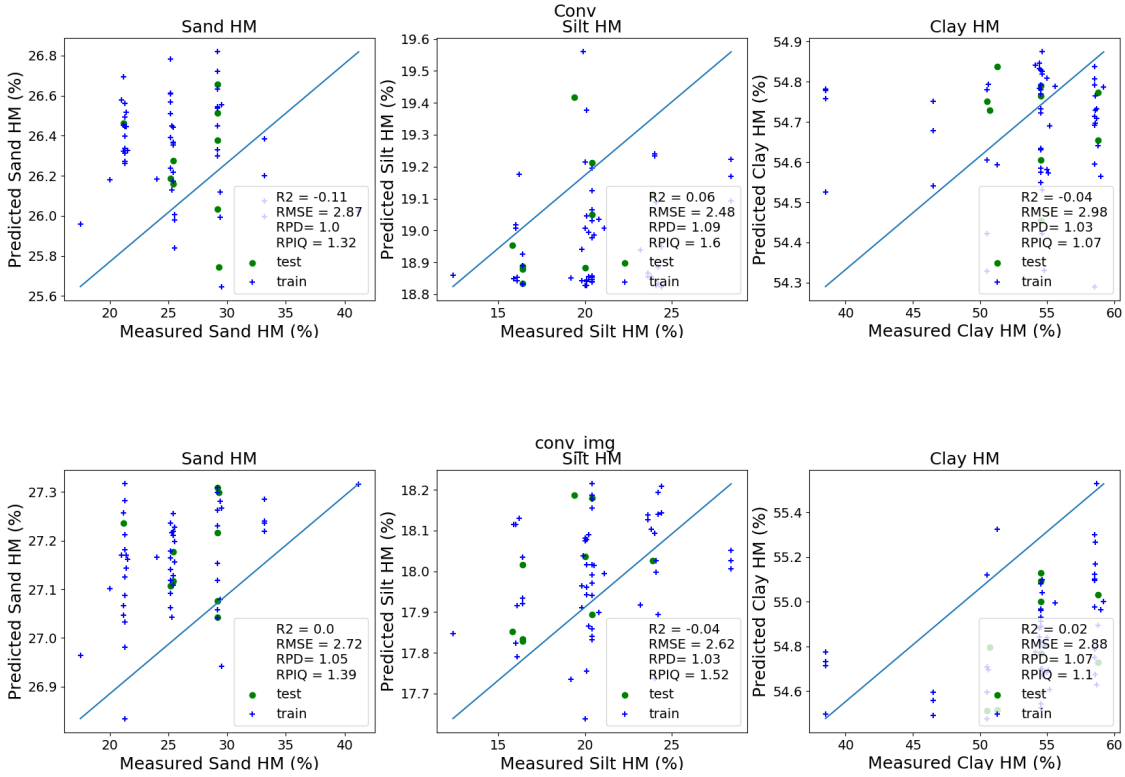


Figure 7.1: Conv, ConvImg and Linear Regression predicted vs laboratory MS measured values. The green dots are for test set and the blue for train set. the blue line represents the 1:1 line.



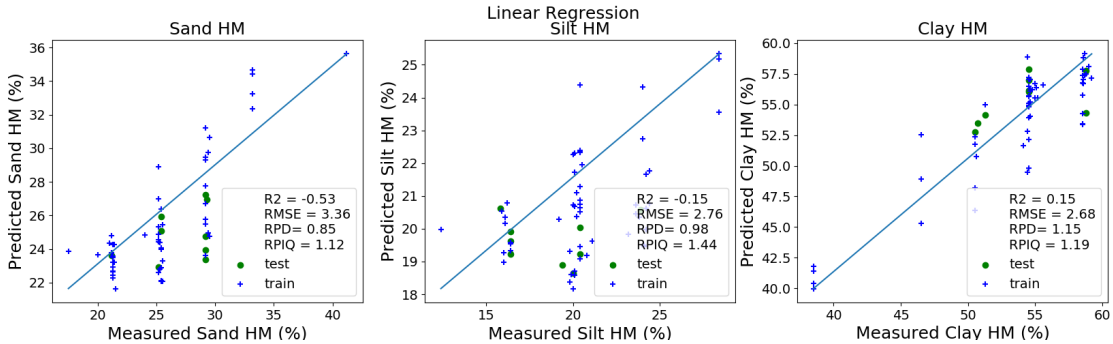


Figure 7.2: Conv, ConvImg and Linear Regression predicted vs laboratory HM measured values. The green dots are for test set and the blue for train set. the blue line represents the 1:1 line.