Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Estado de México

School of Engineering and Sciences

The use of multispectral images and deep learning models for agriculture: the application on Agave

A thesis presented by

José Alberto Montán López

Submitted to the
School of Engineering and Sciences
in partial fulfillment of the requirements for the degree of

Master of Science

in

Computer Science

Mexico City, December, 2022

Dedication

To my family, for all the support they give me in all decisions of my life.

Acknowledgements

I want to thankful my parents and my brother for supporting me with this project and new step on my career, it is wonderful for me to write these words because in my life I never imagine went so far, the way was hard, so many nights with out sleeping or been with my family, and I feel proud to be able to share this moment with my lovely family.

I also want to thank the Tecnológico de Monterrey for the opportunity to enter the master's program and that also provides all the tools making possible this thesis work and also to CONACyT for the financial support that allowed me to concentrate in the project.

The use of multispectral images and deep learning models for agriculture: the application on Agave

by

José Alberto Montán López

Abstract

Agave is an important plant for Mexico, country considered as center of biological diversity of agave, in addition, one variety is used for production of tequila, an important product that brings money to the country. Demand of product has led farmers to pay more attention to plantation and to reduce quality. We can find several solutions regarding agricultural filed such as identification of weed and classification of species implementing aerial imagery along with machine and deep learning reaching good results. However, there are few solutions applied directly on agaves to monitor they health. Moreover, there is not a public dataset about agaves for the purpose of this work, for this reason we have worked to collect data using a drone equipped with a multispectral camera capable to capture five different channels of a different wavelength of the light spectrum. This dataset contains 7ha of agave information into five channels provided by the multispectral camera as well as three Vegetation Indices that were computed from the multispectral bands. In this work we explore the use of recent deep learning (DL) algorithms as well as traditional machine learning (ML) algorithms to segment agaves based on health using aerial multispectral images. On the experiments we found out that ML algorithms were able to segment just one of the two classes defined for agaves. On the experiments of DL models we could define the size of the images we wanted to train where a size of 500x500 was the best for this problem. Experiments for both types of algorithms were done using many combinations of channels such as use just vegetation indices or using all available bands on the dataset. On the other hand, Vision Transformer (ViT) Segmenter model reached an accuracy of 92.96% using vegetation indices data while the best ML algorithm was Random Forest using the five bands captured by the drone reaching 88.06% accuracy. We also test the models using traditional RGB images to compare against multispectral images and see if there is an actual advantage on the use of this type of technology. Results show us that when we introduce the variable of health into agaves, i.e. we have two classes of agaves, models that have additional bands can get better results. Thus, the use of multispectal images actually increase the performance of all models, including ML and DL, for identification of more than one class of agave.

List of Figures

1.1 Quantity of Tequila exported from 2016 to 2020 by category, Tequila and Tequila 100% of agave. Data extracted from ”Consejo Regulador del Tequila” 3
1.2 The production of Tequila and Tequila 100% of Agave from 2016 to 2020. Data extracted from ”Consejo Regulador del Tequila” 3
1.3 An example of an RGB image on the left and a NDVI image on the right . . . 4
1.4 Scheme of the proposal solution 5

3.1 Common devices used in precision farming 13
3.2 Classification of light spectrum by wavelength 14
3.3 Bands of a multispectral image which combination produce a new image with more information 14
3.4 Left image is the RGB and the right image represent the NDVI where high values are in green and low values in orange to red 15
3.5 Example of an orthomosaic map 17
3.6 Image Pre-procesing methods 19
3.7 Topographic representation (on the right) of an grayscale image (on the right). 20
3.8 Initial markers where water will start to rise. 21
3.9 KNN classifier example. A new object (green circle) has to be classified. With k=3, algorithm takes two triangles and one square, then, the object is classified as a triangle. With k=5, two triangles and three squares are considered in the algorithm classifying the new object as a square. 22
3.10 Example of a decision tree to decide whether realize an activity or not depending on the weather 23
3.11 Random forest classifier diagram. Several trees are created and each of them give a vote to have the final class 24
3.12 FCN architecture 25
3.13 Structure of the U-Net model 26
3.14 ViT Segmenter structure. The left side (before the arrow) Encoder is the original ViT, while right side Decoder present Mask Transformer to be used for semantic segmentation 27
3.15 Intersection over Union metric representation 29
3.16 Intersection Over Union metric on segmentation task 29

4.1 Pipeline for this project 30
4.2 Drone P4 multispectral equiped with six cameras 31

4.3 Channels blue and green in left and right respectively. The two images looks similar but they are not aligned. We can see a difference in the bottom-right corner in the wall 33
4.4 Vegetation indices NDVI on the left and GNDVI on the right 34
4.5 NDRE vegetation index on the left and MSAVI on the right 35
4.6 Vegetation indices ARVI and NDWI on the left and the right respectively . . 35
4.7 Tiles obtained in an orthomosaic map 37
4.8 Ground Truth map of an orthomosaic map for three classes. Green for Agaves, yellow for trees and purple for the rest of objects 38
4.9 Input image on the left with a tree and a building, and the correspondent ground truth label on the right. 39
4.10 New data representation for machine learning models of an RGB image and the correspondent label 39
4.11 Flow diagram of the final process to separate agaves 40

5.1 Sample image of an agave crop 44
5.2 Aerial view of the entire plantation. 44
5.3 Zone 1 orthomosaic map 45
5.4 Zone 2 orthomosaic map 46
5.5 Zone 3 orthomosaic map 46
5.6 Zone 4 orthomosaic map 47
5.7 Zone 5 orthomosaic map 47
5.8 Ground truth map for Zone 1. Map contain three colors, blue, green, and black for agave warning, agave normal, and background classes 48
5.9 A comparison of different sizes of tiles on the same map 49
5.10 ROC Curve of KNN algorithm 53
5.11 ROC Curve of random forest algorithm 54
5.12 ROC Curve of decision tree algorithm 54
5.13 Accuracy and loss curves of ViT over iterations 59
5.14 Accuracy and loss curves of Unet model over iterations 60
5.15 Accuracy and loss curves of FCN model over iterations 60
5.16 Predictions of the six models. The target label is showed in the top of the image 63
5.17 Prediction after applying Watershed algorithm. Left side is the final prediction. Right side is a representation where dots are the detected agaves 64

List of Tables

4.1 Minimum and Maximum values of each band 36
4.2 Minimum and Maximum values for Vegetation Indices 36
4.3 Feature representation on an image 38

5.1 Information of the five ares captured on the crop. There are information about the size on pixels of the orthomosaic map, the area covered, the name, and the GSD 45
5.2 Size of the three datasets used to train models 50
5.3 Overall scores of machine learning models 51
5.4 Per class scores for Machine Learning models 52
5.5 Deep Learning models configuration 55
5.6 Overall scores using 500x500 tile size 56
5.7 Per class scores with 500x500 tile size 56
5.8 Per class scores with 300x300 tile size 57
5.9 Per class scores with 300x300 tile size 57
5.10 Per class scores with 120x120 tile size 58
5.11 Per class scores with 120x120 tile size 59
5.12 Overall scores using RGB data 61
5.13 Per class scores using RGB data 61

Contents

Abstract v

List of Figures vii

List of Tables viii

1 Introduction 1
- 1.1 Background 1
- 1.2 Problem Statement and Context 5
- 1.3 Research question 6
- 1.4 Objectives 7
- 1.5 Chapters Structure 8

2 Literature Review 9
- 2.1 Segmentation with visible spectral images 9
- 2.2 Segmentation using multispectral images 10

3 Theoretical Framework 12
- 3.1 Precision Farming 12
- 3.2 Multispectral Images 13
- 3.3 Vegetation Indices 14
  - 3.3.1 Normalized Difference Vegetation Index - NDVI 16
  - 3.3.2 Green Normalized Difference Vegetation Index - GNDVI 16
  - 3.3.3 Normalized Difference Red Edge - NDRE 16
  - 3.3.4 Modified Soil-adjusted Vegetation Index - MSAVI 16
  - 3.3.5 Atmospherically Resistant Vegetation Index - ARVI 17
  - 3.3.6 Normalized Difference Water Index - NDWI 17
- 3.4 Orthomosaic Maps 17
- 3.5 Image Pre-procesing 18
  - 3.5.1 Morphological operations 18
  - 3.5.2 Watershed Algorithm 20
- 3.6 Machine and Deep Learning algorithms 21
  - 3.6.1 K-Nearest Neighbors (KNN) 21
  - 3.6.2 Decision Tree 22
  - 3.6.3 Random Forest 24
  - 3.6.4 Fully Convolutional Network (FCN) 25

3.6.5 U-Net 25
3.6.6 Vision Transformer Segmenter 26
3.7 Metrics to compare models 28
3.7.1 Pixel Accuracy 28
3.7.2 Intersection Over Union 28

4 Methodology **30**
4.1 Introduction 30
4.2 Data Acquisition 31
4.3 Data processing 32
4.3.1 Radiometric correction and orthomosaic map 32
4.3.2 Creation of vegetation indices 34
4.3.3 Adjustment of images and Creation of tiles 36
4.3.4 Ground truth map 37
4.3.5 Machine learning format 38
4.3.6 Watershed algorithm 40
4.4 Machine and Deep learning algorithms 41

5 Analysis and Discussion of Results **43**
5.1 The data 43
5.2 The comparison of algorithms 50
5.2.1 Machine Learning Models 50
5.2.2 Deep Learning Models 55
5.2.3 Models with RGB data 60
5.2.4 Image Segmentation 62

6 Conclusion **65**
6.1 Future Work 66

Bibliography **71**

Chapter 1

Introduction

1.1 Background

Agriculture is an activity that is part of the primary sector, which is related to the treatment of the soil and the cultivation of the land to produce food, a modification of the environment is made to have one more suitable for its use and thus having a higher productivity in the soil with which more resources (food) are obtained for its consumption, either direct or that takes extra processing for its commercialization. It is an important activity since the products generated are used to cover one of the basic needs of humans.

The production of the harvest can be affected by different factors either by factors generated by the environment such as problems related to water, floods or droughts, problems related to the climate, on the one hand, the extremely low temperatures during seasons of frosts and other, high temperatures that can directly damage plants or produce droughts; problems related to cultivation practices such as an excess or lack of nutrients, damage caused by the incorrect use of chemicals, such as the excessive use of pesticides that can lead to the death of the plant, physical damage when handling to the plants. In addition, in the 2030 Agenda on Sustainable Development, where 17 Sustainable Development Goals are defined as a response to improving the lives of all and include issues for the elimination of poverty, climate change, education, and equality for women, among others. Goal number twelve talks about responsible production and consumption where in section 12.2 of that goal they mention ”From now to 2030, achieve the sustainable management and efficient use of natural resources” in section 12.a we can read ”Help developing countries to strengthen their scientific and technological capacity to move towards more sustainable consumption and production patterns” which can be directly related to agriculture, since the development of more sustainable production and the efficient use of resources are mentioned, facts that apply to being a primary activity[39].

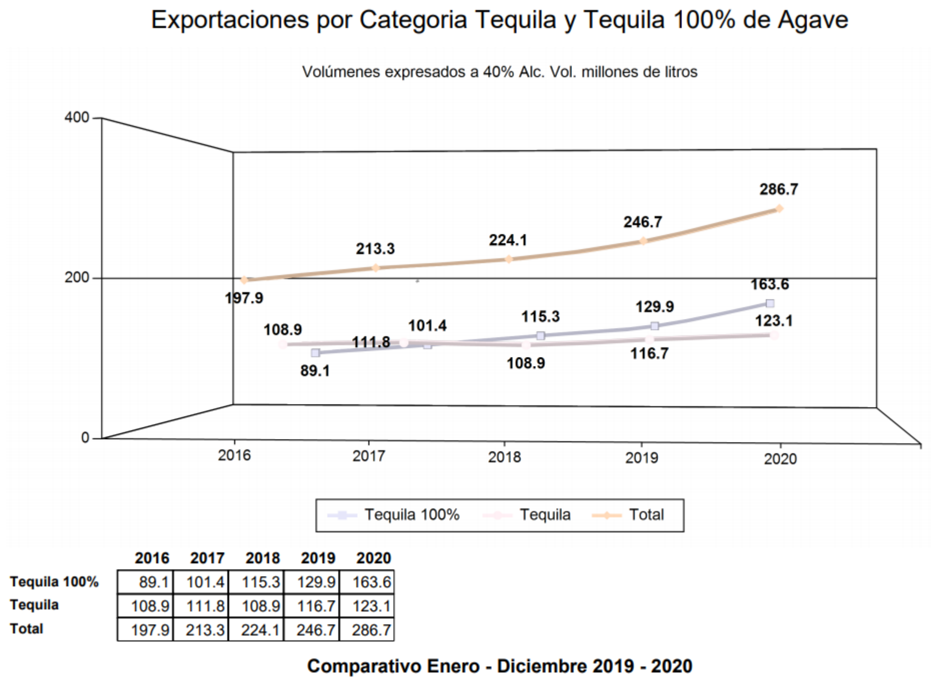
For these reasons, new technology and new techniques have been generated to directly attack these and other problems. Information and communication technology applications are used directly in agriculture. Some of these applications are the internet of things (IoT), Geolocation systems, Big Data, unmanned aerial vehicles (UAV or drones), automatic systems, and robotics. Those technological applications are used to improve the administration of the necessary resources to carry out the harvest, reduce the time for decision-making, and have more precise information. The application of these technologies in agriculture is known as precision agriculture or Smart farming. Within the economic factors, precision agriculture provides higher productivity, higher income, and better product quality. The amount of undergrowth, pests, and diseases are reduced thanks to the application of these technologies [1].

One of the plants that are cultivated in the Mexican territory on a large scale is the agave. The Agave has an origin that dates back approximately 10 million years to the American continent, this being the place where most of the species are found. Worldwide, we can find a total of approximately 211 species, this number may vary a little depending on the author who identifies them, of which 75% are found in Mexican territory being the state of Oaxaca where the greatest variety of species is found, which makes Mexico the center of the biological and cultural diversity of agave[52]. Among the first registered uses, it is found in things such as food and used as a fiber for the manufacture of ropes and nets. Today, its use has grown and reached various areas such as in the agricultural sector as an agent to prevent erosion and as an organic fertilizer, in the area of medicine as an anti-inflammatory, and in household products such as detergents and some kitchen utensils. The importance of this plant is such that research works have been carried out around it, the chemical sector being the one that has the most contribution with 62% being the second the medicine sector with 15%. Within the chemical sector, there are topics related to drinks, biofuels, and sweeteners, among others. In addition, Mexico contributes 54% of scientific publications, its main topics being drinks and sweeteners[10].

However, despite the vast number of applications that this plant has, its main use is the production of drinks, such as mezcal, pulque, and tequila. [16] The latter comes from the Blue Weber Agave, the only variety of agave with which tequila can be produced. This species of agave can only be found in the states of Jalisco, Michoacán, Tamaulipas, Nayarit, and Guanajuato, which is why it has earned the so-called designation of origin. Tequila production is an industry that generates sales of more than 2 billion dollars annually and employs around 70 thousand people. The popularity of Tequila has been on the rise, so much so that the price increased six times during the period from 2016 to 2018. [15] We can see this in the amount of tequila exported and the agave production. Figure 1.1 shows the increase in the exportation of Tequila from 2016 to 2020, we can see a growing demand for Tequila 100% while the amount of Tequila remains the same. On the other hand in fig. 1.2, production of Tequila has increase considerably since 2017 where Tequila 100% leads the production. This data tell us that more and more Agave must be cultivated in order to satisfy the demand and because of the time it takes to grow, farmers needs to put more effort in care the crop.

The main country to which it is exported is the USA, followed by Germany and Spain. Due to this large amount of necessary production and the quality demanded by companies, the care, maintenance and protection of plants is becoming increasingly difficult. Caring for the agave against problems is mainly done by going through the crops and checking that the plant is in good condition. Going through a complete plantation checking the health status of each agave takes a long time, and with the current trend where greater production is required, the times to maintain the care increases.

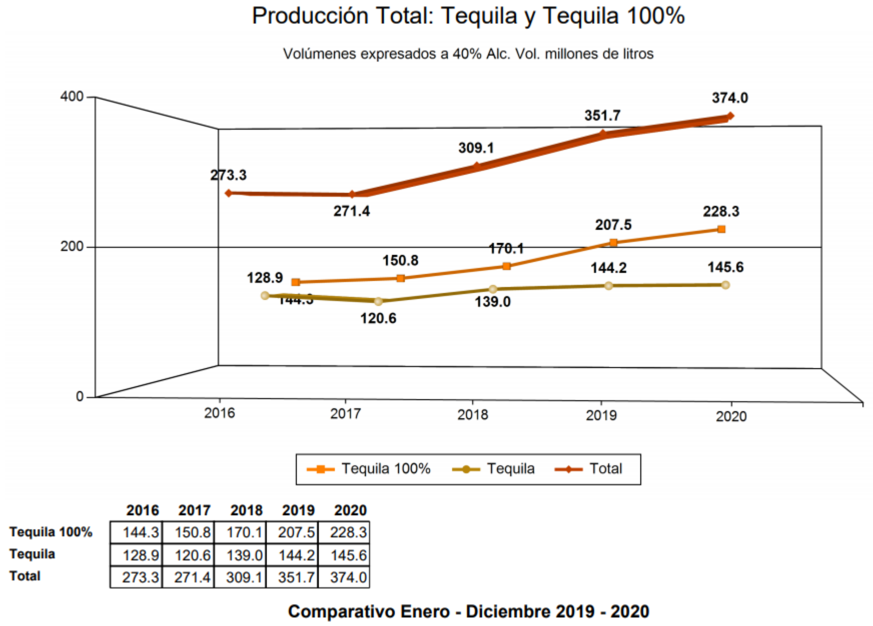
One of the common solution is the use of Vegetation Indices (VI), being the most popular the Normalized Difference Vegetation Index (NDVI). NDVI is an image calculated by two channels captured by a multispectral camera, these are Near Infrared (NIR) and Red channels. With NDVI is possible to know in a faster way the health of a plant allowing a quickly



	2016	2017	2018	2019	2020
Tequila 100%	89.1	101.4	115.3	129.9	163.6
Tequila	108.9	111.8	108.9	116.7	123.1
Total	197.9	213.3	224.1	246.7	286.7

Comparativo Enero - Diciembre 2019 - 2020

Figure 1.1: Quantity of Tequila exported from 2016 to 2020 by category, Tequila and Tequila 100% of agave. Data extracted from ”Consejo Regulador del Tequila”



	2016	2017	2018	2019	2020
Tequila 100%	144.3	150.8	170.1	207.5	228.3
Tequila	128.9	120.6	139.0	144.2	145.6
Total	273.3	271.4	309.1	351.7	374.0

Comparativo Enero - Diciembre 2019 - 2020

Figure 1.2: The production of Tequila and Tequila 100% of Agave from 2016 to 2020. Data extracted from ”Consejo Regulador del Tequila”

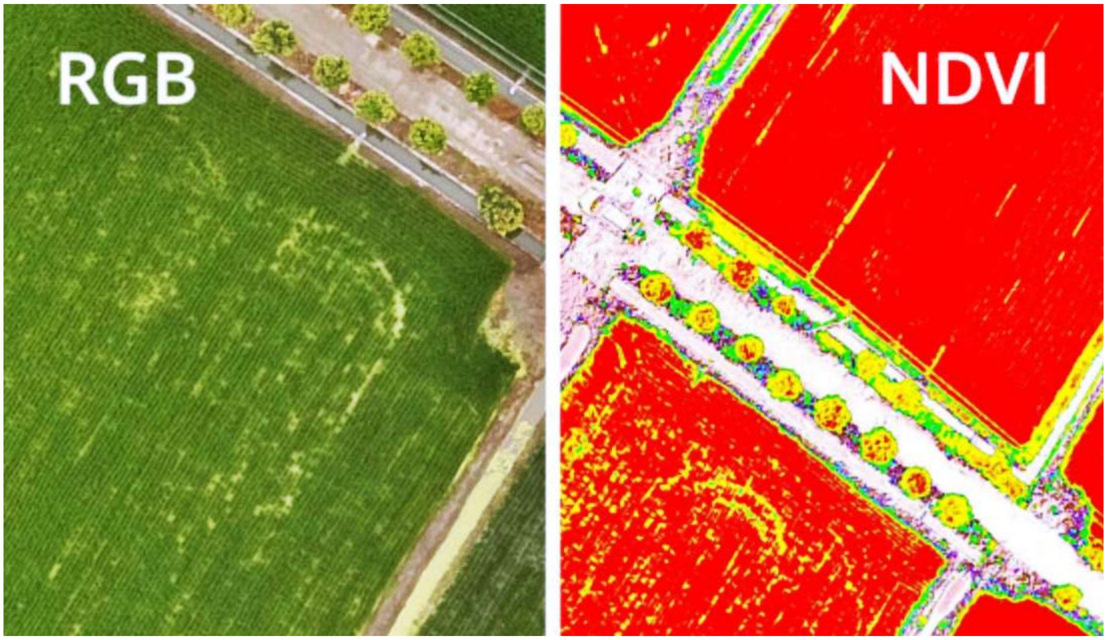


Figure 1.3: An example of an RGB image on the left and a NDVI image on the right

evaluation of the state of the crop. The reason it works well and is very popular is because is based in how the plant interact with light. Pigment on plants absorb the visible red light while it reflects the NIR. Moreover, some drones like those from ”DJI” already count with a computed NDVI output. We can observe an example of the application of NDVI in fig. 1.3 where in the left side a common RGB image is seen while in the right side is the resultant image after compute the NDVI, note that is easy to see whether is plant or not and for the variation of color represent the health. But for those that are not included, a dedicated software is required. There are more VI but if you wish to apply it or explore more option with multispectral images, is necessary first have the software and then learn how to use it.

Thus, the solution we propose to solve these problems can be seen in fig. 1.4. We propose the use of a UAV to collect images of the crop, these images are captured with a multispectral camera that can capture light for a range of wavelengths (called bands). Images are pre-processed to choose a combination of bands that provide more information. For feature extraction to segmentation, we will explore several deep learning algorithms and select the efficient one but, as a main and preferable algorithm to explore more applications, the use of U-Net will be explored deeper. Areas will be segmented by characteristics of Agave and the soil e.g. nutrient level, pests, the health of the plant, and humidity. Finally, given areas that are segmented help make a decision allowing the farmer to make the right decision in a specific area. The main contribution is in the segmentation since there are several classes to classify, from where many problems can be covered at the same time.

By the given solution, we expect to have an algorithm capable of detecting areas and classifying them according to features such as humidity and nutrient levels with high accuracy. In addition, some of the benefits we can reach are the following: an increase in the productivity of agave crops such as other precision farming methods can achieve but in an efficient way that is easy to understand since specific areas are labeled for the corresponding class. Thanks

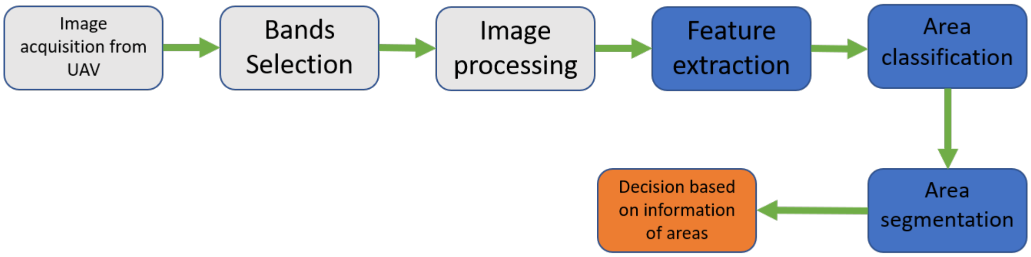


Figure 1.4: Scheme of the proposal solution

to this method we can also reduce the workforce necessary to supervise the entire crop to ensure its quality of it. The proposed solution also performs a helpful way to efficient use of resources such as water, fertilizer, and pesticide and focus on a delimited area if there is a problem and not waste resources in areas where is not necessary. Finally, we can use it as a health supervisor of the plant and the soil and know exactly or with high accuracy where and what the problem is.

1.2 Problem Statement and Context

One of the problems that we can face is the wilting of the plant. This problem is derived from variants of the climate, which, although we can not control but we can take preventive measures, and soil variables. These variables are maximum and minimum temperature, relative humidity, soil humidity, precipitation, and solar radiation. By having control of some of these variants such as soil moisture with irrigation and having measures to better manage the other variants, the plant has less chance of suffering a wilting. Plant nutrition is an important factor, having soil with the necessary nutrients will help the plant to develop, and a balance of these nutrients is better for the plant. The amount of water in the soil, which is directly related to humidity, can become a problem in uneven terrain, the amount in a region can be greater due to these irregularities, causing the water to stay in one place or have it run downward by gravity on steep terrain. Finally, we have the problem of pests, an insect known as the Agave Snout Weevil with the scientific name Scyphophorus acupunctures Gyllenhal is considered the most important pest for the Blue Weber agave, causing damage to 24.5% of the population. The damage is so serious that it can lead to the death of the plant [12].

Nowadays, there are solutions for some general problems that can be applied to a number of types of crops. One of the most popular problems is Weed Detection or Weed Mapping, the task consists in segment or identifying if the plant on the image is part of the crop or not. The importance comes since weed is undesired because it consumes resources, such as water and nutrients, affecting directly the entire crop healthy and indirectly the economy of the farming.

Therefore, datasets are published publicly in order to develop an efficient algorithm or solution that can lead to this specific problem. By the year 2020, 15 different datasets of weed detection are available in open source [36]. Datasets use crops of carrot and sugar for weed detection but just a few of them use a UAV to collect images. The main difference between datasets is the kind of image collected, on one side we have those that capture RGB images, and on the other hand, those that use Multispectral Images including NDVI. We want to highlight this difference since it is part of the proposed solution, we are using multispectral images that contain more information. Furthermore, as proof that better results are achievable by using more channels of images, recent research done for Weed Mapping using Deep Learning algorithms and multispectral images demonstrated how by choosing the right combination of channels a high accuracy is achieved and compared against SegNet architecture receives RGB images as input [47]. A significant improvement where Area under the ROC Curve (AUC) score is increased by about 0.2 points is achieved with this method. In addition, the usefulness of NDVI is reinforced in that paper.

On the other hand, we can find datasets for precision farming for diverse applications. For example, in fruit detection there are datasets dedicated to this task commonly with RGB images, fruits to be detected include apples, mango, and almond. Other datasets aim at the detection of diseases of Maize, where images are captured at short distances and with channels of RGB. Finally, there is a dataset of videos captured with a Microsoft Kinect v2, that allows making a segmentation in a 3-dimensional space [36]. Despite having datasets for multiple tasks, there is still no agave dataset that we can use.

In the literature, there are just a few solutions that have been developed directly for Agave. One of these is a counting methodology [6]. This involves a problem in which a number of agave in a field is necessary to estimate the production of Tequila, to avoid counting through all the crops by hand with their respective error margins a method is defined to do it automatically. In this solution, image acquisition is made by a UAV that is equipped with an RGB camera. Their solution is to apply a method based on Mathematical Morphology that involves image processing to apply filters and kernels to create a mask to segment Agave. The solution reaches an accuracy of about 98%.

Another solution is the one proposed by Ramirez et al. [9]. This work aimed to identify specific areas by three restrictions, theses are economic, health, and humidity. They use a supervised classification with multispectral images and again the NDVI. An accuracy of 73% was reached, a percentage that can be improved by applying other methods. An interesting result is shown in this work, which is the values of NDVI, they found that this value changes depending on the age of the agave, a higher value represents an older plant. This work involves the problem of segmentation by characteristics of soil, in this case, was measured by humidity, and the health of the plant. However, we expect to have better accuracy by using deep learning methods.

As we can note for solutions applied directly on Agave, there are still a lot of opportunities to develop tools and to take advantage of UAV and multispectral images. In this work we want to bring a direct solution to the Agave, we saw a good Weed Mapping solution and the usefulness of NDVI so with a combination of those methods and an exploration of U-Net architecture or more deep learning algorithms, we expect to achieve an efficient solution.

1.3 Research question

To attack problems that the Blue Weber Agave has, we can use an unmanned aerial vehicle, in this case, a drone, to fly above the plantation to obtain information on some characteristics like nutrients and humidity levels of the plant and the soil. As we have commented before, by using multispectral images we can get better results. Space between plants, the presence of undergrowth, and the detection of pests are possible with RGB images. We can note that there is a limitation if we want to solve as many as possible of problems that the agave has. So, to improve the number of features, a multispectral image is a good option. A multispectral image is an image that has information on several spectral bands, each of these bands captures the reflation of different wavelengths such as infrared and ultraviolet. Thanks to the information given by the multispectral images we can attack more problems, those for RGB images in a better way plus humidity, nutrients of the plants, and temperature.

Thus, the hypothesis is:

> With the use of UAV multispectral imagery we can identify agaves which their condition or health is not normal corresponding to a problem on the plant that can be pests, low levels of nutrients, or any other condition that can affect the plant. In addition, since multispectral images provide more information, they can get better results compared to traditional RGB images.

Questions that will support the given hypothesis are the following:

- What are the common characteristics of a healthy plant?
- What Vegetation Index can we use in order to obtain a large value or information?
- Does the use of deep learning models is superior to machine learning models?
- Is there an advantage on the use of multispectral images against RGB images?

1.4 Objectives

The general objective of this work is to develop a model based on deep learning algorithms and multispectral images in order to monitor the health of the Agave to ensure quality and identify critical areas that are put at risk the agave production by the end of the Master program. To achieve this objective, we need a list of goals to measure the efficiency of the solution. The proposed goals for this project are the following:

- As the first objective due the lack of information we are capturing data to create a dataset for this project.
- Generate vegetation indices of the captured data
- Creation of ground truth labels to train models
- Segment agaves base on the health level

Note that each objective depends on the past objective, thus, the final solution is a combination of all of them. This combination will be use to identify areas by importance level doing decision making easier.

1.5 Chapters Structure

This work is divided into six chapters, starting with this Introduction chapter and follows this structure: Chapter 2 presents the previous research done for semantic segmentation using aerial imagery; Chapter 3 explains the topics used in this work; Chapter 4 describes the methodology used to achieve the goals of this thesis, using a pipeline to explains the steps followed; Chapter 5 discuss the results obtained using two types of algorithms; Finally, Chapter 6 ends with a conclusion of the entire work and future work that can be done in order to improve the results.

Chapter 2

Literature Review

Literature Review chapter is focused in discuss previous work regarding aerial imagery for agricultural use. The use of unmanned aerial vehicle has been explored by researchers to deal with problems on crops such as weed detection. In the literature we can find solutions using UAVs to capture images in RGB and Multispectral formats, applying machine learning algorithms with interesting results. For both types of images, the high spatial resolution stands out as the main advantage of UAV technology.

As mentioned in the review article [40] by Oghaz et al., there are several studies done using aerial imagery along with deep learning methods in crops. They identify five groups where research have done, these are: 1)vegetation identification, classification and segmentation, 2)crop counting and yield prediction, 3)crop mapping, 4)weed detection, and 5)disease and nutrient deficiency detection. There is a number of solutions for each kind of problem but there is still opportunities to enhance those solutions.

This chapter is divided into two main sections, the first sections aims on the work done using visible spectral images i.e. RGB images where we can find good results for some specific problems. On the other hand, the second sections is focused in work done using multispectral images, here we can find applications using Vegetation Indices to help algorithm have a better performance.

2.1 Segmentation with visible spectral images

Even with more actual technology for agricultural use, RGB imagery is still the cheaper option so it is accessible. Research have showed that it is not necessary the top technology to reach high accuracy by using the correct methodology. Following, it is going to be discussed the use of RGB aerial imagery for vegetation studies. An article of remote sensing by Bhatnagar et al.[4] explore several machine and deep learning algorithms to perform segmentation of vegetation communities in Clara Bog, a nature reserve in Ireland. They acquired data using a DJI drone with optical camera to capture data from bog using a specialized software to generate the path and capture the desired area with a spatial resolution of 30cm. They collect around 75 images of 3000x4000 dimension with five classes to segment. In order to be consistent with all algorithms, and compare them, Bhatnagar et al. decided to resize images into a smaller scale scarifying high spatial resolution.

As mentioned before, Bhatnagar et al.[4] explored a number of machine and deep learning model including random forest, graph cut segmentation, SegNet, VGG16, UNet, ResNet50, and PSPNet. In order to compare the performance of algorithms, they use overall accuracy score. For Machine Learning algorithms they could reach an accuracy of about 85% thanks to random forest classifier. On the other side deep learning algorithms perform better achieving a higher accuracy of 91% with ResNet50 along with UNet.

Comparing algorithm they define Pros and cons for each one for example, machine learning algorithms are capable to train faster than deep learning algorithms besides they need less data to train and no specialized hardware (dedicated graphic card) is required. Oppositely, it requires manual adjustment with extra process to add features. On the other side of the coin, deep learning algorithms can learn more features without extra functions to reach a higher accuracy, however, it requires more time and data as well as GPU and good RAM. They end by considering ML algorithm better suitable for this application due to low time to train, good accuracy, and limited data for train models.

2.2 Segmentation using multispectral images

The use of multispectral images has been recently explored by researchers due to drones already equipped with multispectral cameras and cameras that count with the necessary software to do the corresponding calibration of channels. Multispectral images count with more information due to number of extra bands when capturing images. These channels contribute with information that are not visible with simple RGB images generating new opportunities to explore. However, this kind of images present challenges we do not have with common images but on the other hand, we can generate new channels to gain valuable information for plants. In the article [7], Candigo, S. et al. discuss the use of Multispectral Images and Vegetation Indices in the precision farming field in order to evaluate the performance of such technology in cultivations of vineyards and tomatoes. First, they start with a comparison of different techniques used to collect images from an aerial view where the spatial resolution, cost for data acquisition, and usability are compared. Satellite platforms, such as LansatLook from the USGS, allow users to explore satellite data with a spatial resolution of about 1-25 m but the cost for image acquisition is very high. While the use of UAVs gives a spatial resolution of 0.5-10 cm, depending on the altitude at which images were taken, that is more suitable for a more precise study of the crop. In addition, the cost of data acquisition by using this method is cheaper compared to a helicopter and airborne image collection.

Candigo, S et al.[7] used a UAV hexacopter equipped with a Multispectral camera for data acquisition that captures the green, red, and NIR bands. By using these three bands, they were able to compute three different Vegetation Indices, NDVI, GNDVI, and SAVI. They save two datasets, one for a vineyard crop captured from a flying height of about 100 m and the second one for a tomato crop with a flying height of 80 m. Both datasets go through a pipeline to perform a radiometric calibration, then an image orientation adjustment and finally construct the orthophoto. The final output is an orthoimage with three channels aligned with a 5 cm resolution. After constructing the orthoimage, Vegetation Indices are computed in order to study their values in the entire crop.

For the first crop, they selected four zones for the study with diverse vegetation conditions. Then, they compare vegetation index values in those zones finding that Vis used to perform good discrimination between areas with high or low health of the crop. In addition, they find it very useful to have high-resolution data since it can detect more details and features compared with satellite images. Similar results were presented in the tomato crop. This article compared three vegetation indices in two crop types showing that VI is able to distinguish between plant health for a variety of conditions. Finally, Candigo, S et al.[7] recognize the potential of UAVs for image acquisition since are a flexible, fast, reliable, and economic tool for information collection and for decisions making in the agricultural field.

Sa, I. et al. present in [47] a semantic weed mapping using Multispectral images and deep neural network. In this paper, they identify principal limitations in the use of UAV images, Multispectral bands, and deep neural network algorithms. Such limitations are those involved in the data acquisition, flight altitude and ground sample distances, change of the resolution that generates a loss of information, and channel alignment.

The task of this work was to detect weeds in a sugar crop, datasets were collected in crops in Eschikon and Rheinbach, Germany. They use two different drones for image acquisition, one equipped with a camera of five bands and the other with four bands. In addition, they compute the NDVI vegetation index in order to have more information about the crop.

As mentioned before, Sa, I et al.[47] found limitations in the use of multispectral images, hence, they define a method to solve these problems. The process is the following: first, they start by capturing the data at a specific elevation such that GSD value is similar for each orthoimage, then, they use software that performs the radiometric calibration for the camera and sun sensor as well as the alignment of multispectral bands. By having the orthomosaic map they can compute vegetation indices, RGB images, and color-infrared images. Just after processing the data, a sliding window is used to trim the image into smaller images with a specific size in order to conserve the details of the data. Then, trimmed images are the input for a SegNet model. By using this data in the Deep Learning model, they could reach an AUC of 0.863 for crop segmentation and 0.782 for weed detection.

Chapter 3

Theoretical Framework

In this chapter, we are explaining the topics required for a better understanding of this work. The chapter is divided as follows: first, we explain the definition of precision farming as well as its importance in agriculture. Then, we move to multispectral images which are the principal data for the development of this project. The next section is a description of Vegetation Indices, a tool we can use to extract useful information from multispectral images. In addition, a list of indices for the use of this application is proposed. We continue with orthogonal maps where some adjustments for images are performed. Finally, machine learning algorithms for segmentation are described.

3.1 Precision Farming

Precision Farming (PF), also known as Smart Farming is the task to manage correctly the use of resources by a collection of information from a number of devices that add precision i.e. having efficient farming with the help of technology [31]. In recent days, PF involves the use of IT technologies such as Global Positioning Systems (GPS), Geographic Information Systems (GIS), drones, sensors, and robotics. we can see some examples in fig. 3.1 where 3.1a is a humidity sensor installed close to the plant, 3.1b are drones used to collect information from aerial images, and 3.1c is a robot equipped with sensors and a camera that can collect information through the crop. Some of the uses are to measure soil moisture, nutrients, and pH levels in order to map critical areas. In other words, PF is the use of IT technologies in order to collect data and proceed to make a decision, have better resource management, and increase productivity. However, the result depends on how good the analysis was, thus an expert in the area is crucial for this task.

According to Krishna [31], precision farming contains at least these components: GPS, Remote Sensing Imagery with information on soil fertility and crop productivity, GIS software, and Variable Rate Applicators (VRT) by using robotics. The advantages of PF vary regarding geographic location as well as the kind of farming. In addition, Krishna compares the advantages of rice farming and European Plains finding that they are different even when the techniques used are very similar. Some of the topics related to the variation of the advantages are: geographic area, soil, the pattern and agronomic requirements of the crop, and their economic value. However, the efficient use of agro-consumables as well as the reduction

(a) Humidity sensor (b) Drones for image acquisition

(c) A robot with sensors and a camera that travels the crop to collect information

Figure 3.1: Common devices used in precision farming

of cost are common groups where precision farming helps to enhance the productivity of the crop.

On the other hand, the implementation of PF is not easy at all, for instance, in Latin America the adoption of PF requires a high initial investment for equipment as well as management time and in some cases, the profit is minimum due to the low price of the product in the crop. In another case, in Europa, the implementation is time-consuming and requires specific skills to be used such as the interpretation of digital imagery, it also has a high cost and can be considered a long-term investment. This problem also happens in Asian countries where the cost and the culture are some constraints in the adoption of precision farming.

3.2 Multispectral Images

A multispectral image captures data of a number of bands, commonly five but can be a little more, of the light spectrum 3.2. The visible light comes from 380nm to 750nm, crossing those edges we can find X-Rays, Infrared, Gamma rays, Ultra-violet, and radio waves. Thus, we call a band the capture of a value of wavelength in the light spectrum e.g. if we want to capture the Infrared, we can capture a wavelength of 1000nm and this would be a band. In addition, a combination of those bands is done to obtain more information, we already mentioned the NDVI as a combination of bands Red and NIR.

Fig. 3.3 shows how a multispectral image looks like. We can observe five bands for the same captured image and then by combining them a new image is computed.

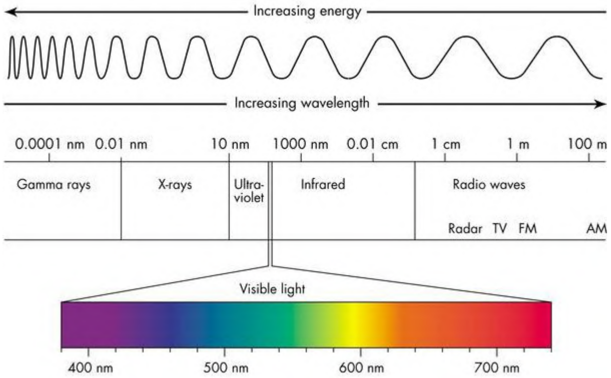


Figure 3.2: Classification of light spectrum by wavelength

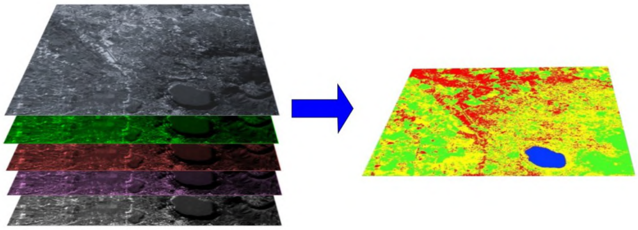


Figure 3.3: Bands of a multispectral image which combination produce a new image with more information

3.3 Vegetation Indices

In the remote sensing field, scientists have developed Vegetation Indices (VIs) which are a combination of reflectance values of two or more wavelengths to measure and detect vegetation properties [17]. A Vegetation Index determines reflectance surfaces such as soil and plants in the function of parameters like color, brightness, and humidity. Depending on these parameters a different wavelength of light is reflected [18]. In multispectral images, we would see the reflected light in a band of the image. Nevertheless, this is not limited to multispectral images, VI is also found for RGB images. Vegetation indices are used as indicators for the assessment of healthy by measuring levels of nitrogen, carbon, and leaf pigments. Most of these indicators are based on the red edge (RE) and near-infra-red (NIR) spectral regions due to the relationship to forest properties [34]. In addition, vegetation indices are designed to maximize vegetation characteristics extraction while factors that can affect the index calculation e.g. soil background reflectance or atmospherical effects are minimized [14]. One of the most popular VIs is the Normalized Difference Vegetation Index (NDVI), whose formula will be explained later, where NIR and Red bands are used to compute the index. In Figure 3.4 we can observe an example of the NDVI against an RGB image. High values of NDVI are shown in green, representing dense vegetation, and low values in orange and red where green areas and soil are easy to distinguish.

According to Jackson et al. [26], the perfect vegetation index must be highly sensitive to vegetation and insensitive to confounding factors. We refer to confounding factors as those characteristics that reduce the sensitivity of the index by affecting the reflectance of

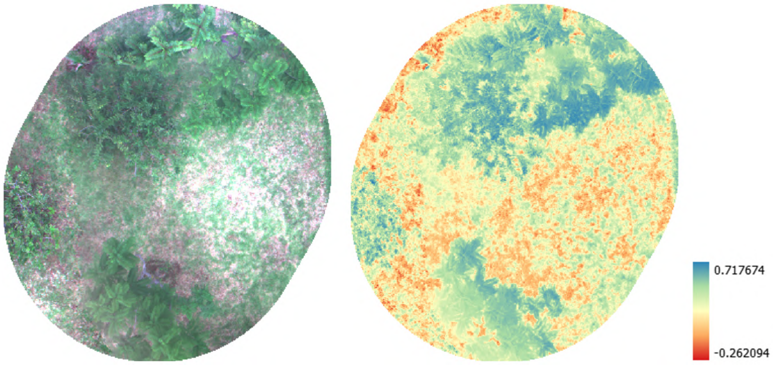


Figure 3.4: Left image is the RGB and the right image represent the NDVI where high values are in green and low values in orange to red

vegetation. These factors include soil brightness, soil color, atmospheric effects, environmental effects, and solar illumination. To solve those problems, we can find several Vegetation indices that solve for some specific factors like the Soil Adjusted Vegetation Index (SAVI) index that add an adjustment factor to the NDVI index in order to minimize the soil background reflected radiation [25].

Due to the number of vegetation indices that we can find in the literature, researchers have classified Vegetation Indices into two categories e.g. Baret et al. [3] separate VIs based on the slope (NDVI, SAVI) and by the perpendicular distance concerning the bare soil line. Bannari et al. [2] define a sophisticated classification for vegetation indices that consider confounding factors and improvements of classical VIs; the two groups are first-generation indices and second-generation indices. First-generation indices do not bear in mind atmospheric effects and soil characteristics due to they were designed using empirical methods. In addition, they have limitations for scalability since they use a specific sensor for specific applications. The second-generation indices are those based on mathematics and experimentation by thinking about sensor calibration and atmospheric effects. Since second-generation indices take into account effects that could impair the correct value, these VI are the best choice for implementation in this work. However, a vegetation index must be chosen based on the environment and the kind of desired application.

Vegetation indices have been widely implemented in many applications in precision agriculture such as virus detection on tomato plants [48]. They have been compared in order to find Vi that can provide the best information for a specific task like soil moisture, weed detection, virus detection, and more [8, 22]. It is clear that VIs needs to be chosen depending on the problem, the equipment data is captured, as well as the environment since they can change the behavior of vegetation indices [55].

In order to generate information for our application, we have selected several vegetation indices of the second-generation indices that consider all confounding factors. In the next sub-sections, we define VI for each confounding factor in order to compare them to select those that are more suitable for the project and to provide more information to the algorithms.

3.3.1 Normalized Difference Vegetation Index - NDVI

The Normalized Difference Vegetation Index is the most popular Vegetation Index and more used in farming topic. This index was developed by the NASA in order to monitor vegetation using satellite data[46]. NDVI uses Red and NIR bands to calculate the health of a plant, equation is as follows:

$$\mathrm{NDVI} = \frac{\mathrm{NIR} - \mathrm{R}}{\mathrm{NIR} + \mathrm{R}}, \tag{3.1}$$

where NIR is the Near Infra-Red band and R is the Red band. Values of NDVI go from -1 to 1. Negative values, close to -1, are for water and clouds; values close to 0 are for bare ground; positive values represent green areas[38][19]. Those values are seen in Figure 3.4 where soil has values close to zero and green areas have positive values, higher values are more healthy plants.

3.3.2 Green Normalized Difference Vegetation Index - GNDVI

Purpose of green normalized difference vegetation index (GNDVI) is to measure leaf chlorophyll content[21], dditionally, GNDVI has a high correlation with nitrogen concentration. Equation is simlar to NDVI but replacing red channel by green channel:

$$\mathrm{GNDVI} = \frac{\mathrm{NIR} - \mathrm{G}}{\mathrm{NIR} + \mathrm{G}} \tag{3.2}$$

3.3.3 Normalized Difference Red Edge - NDRE

Normalized difference red-edge vegetation index (NDRE) is another option to measure chlorophyll content with an equation similar to NDVI using Red-edge instead of Red channel. This change allow measure chlorophyll when they are in the extremes levels [20].

$$\mathrm{NDRE} = \frac{\mathrm{NIR} - \mathrm{RE}}{\mathrm{NIR} + \mathrm{RE}}, \tag{3.3}$$

where RE is the Red-Edge channel and NIR is the near infra-red channel. Higher values represent a higher chlorophyll content.

3.3.4 Modified Soil-adjusted Vegetation Index - MSAVI

This vegetation index is a modification of the Soil adjusted vegetation index that aims to correct the factor caused by soil radiation where a factor L has to be selected depending on vegetation density. Modified Soil-adjusted vegetation index (MSAVI) does not require a correction factor since a function that minimizes soil effects is inside the equation[44]:

$$\mathrm{MSAVI} = \frac{2\mathrm{NIR} + 1 - \sqrt{(2\mathrm{NIR} + 1)^2 - 8(\mathrm{NIR} - \mathrm{R})}}{2} \tag{3.4}$$

3.3.5 Atmospherically Resistant Vegetation Index - ARVI

The Atmospherically resistant vegetation index, as the name implies, is a vegetation index designed to correct atmospheric effects caused by atmospheric particles, i.e., fog, dust, smoke, or pollution[27]. ARVI has a similar dynamic range to NDVI but is less sensitive to atmospheric effects.

$$\mathrm{ARVI} = \frac{\mathrm{NIR} - (2\mathrm{R} - \mathrm{B})}{\mathrm{NIR} + (2\mathrm{R} - \mathrm{B})}, \tag{3.5}$$

where NIR is the Near Infra-red band, R is red band, and B the blue band.

3.3.6 Normalized Difference Water Index - NDWI

The use of Normalized Difference Water Index (NDWI) is to describe water features[37]. NDWI uses green and NIR channels to maximize water reflectance and the soil moisture can be obtained. Equation is as follows:

$$\mathrm{NDWI} = \frac{\mathrm{G} - \mathrm{NIR}}{\mathrm{G} + \mathrm{NIR}}. \tag{3.6}$$

3.4 Orthomosaic Maps

An Orthomosaic Map can be defined as an aerial representation of the captured area created by joining several photos that have been geometrically corrected. The scale is uniform, so they have the same lack of distortion. Since orthomosaic is an accurate representation of the surface, it can be used to measure distances with high precision. Figure 3.5 shows a representation of an orthomosaic map formed by smaller images. The use of multispectral images requires extra steps to compute the ortomosaic map such as corrections related to sensor settings (aperture time and shutter speed) and scene conditions (camera location and sunlight).

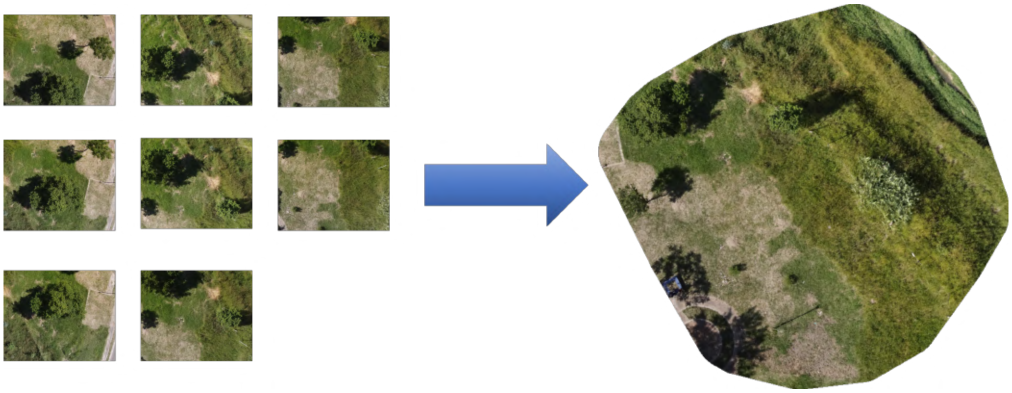


Figure 3.5: Example of an orthomosaic map

As mentioned before, orthomosaic maps can be used to measure distances thanks to the Ground Sample Distance (GSD). GSD refers to the distance representation in a pixel on the ground. There is a relation between the spatial resolution of the image and GSD, a low value of GSD represents a higher spatial resolution and can capture more details. For instance, a GSD of 20mm represents 20mm of the ground and has a high spatial resolution while a GSD of 10cm represents 10cm of the ground and has a low spatial resolution. Having a small GSD requires flying low and taking more images.

We can find specialized software that performs those corrections and generate orthomosaic maps even with multispectral images such as Pix4D, ArcGIS, as well as open-source options like Open Drone Map.

3.5 Image Pre-procesing

While working with images, we can face images that look of bad quality, whether high or low illumination, in other cases, we can note a kind of noise that makes it looks fuzzy or in general bad. Thus, before entering the deep learning method, it is combined to apply computer vision methods in order to enhance the quality of images [30].

In general, some of the methods used are the following and can be seen in figure 3.6:

- **Gaussian Filter 3.6a.** This filter aims to reduce image noise applying a kernel where pixel in the center have more weight and pixels around are decreasing. Resultant pixel is the value of the average of apply the kernel.
- **Equalization 3.6b.** This method uses histogram of an monochrome image, the aim is to enhance image contrast by transform it with an histogram with uniform distribution.
- **Binarization 3.6c.** Is the reduction of information of an image. Creates an image with values 1 and 0 for each pixel given the original value and a threshold that decides the final value.
- **Color adjustment 3.6d.** Adjust colors of an image, reducing or increasing values of a channel to have colors balance and avoid saturation.
- **Morphological Operations 3.6e.** Mathematics operations applied to binary images based on forms, pixel value is computing by comparing with neighbours pixel values. Common operations are dilation and erosion, while dilation add pixels, erosion remove it.

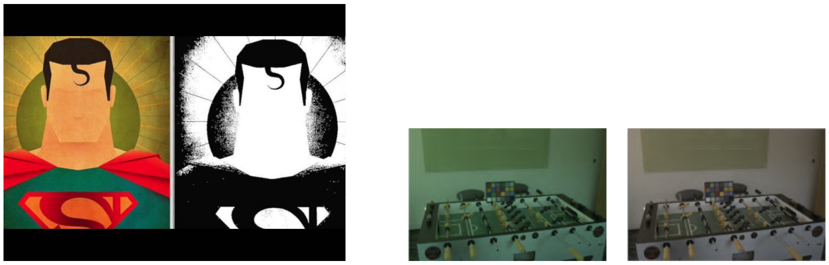
3.5.1 Morphological operations

Morphological operations are a broad set of image processing similar to filtering that is based on shapes. The structuring element is applied to an input image, then it is moved across every pixel to create the processed image of the same size. The value of each pixel in the output depends on the comparison of the pixel with its neighbors. The most basic operations are dilation and erosion.

The erosion operation removes pixels on an object by selecting the minimum value of all pixels in the neighborhood. In a binary image, the pixel value is set to 0 if any pixel in the

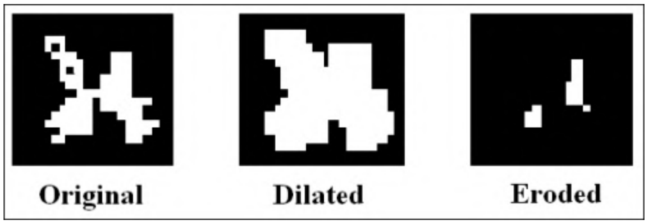
(a) Gaussian Filter. On the left side is the original image and the right side is the resultant to apply Gaussian filter, note that resultant image is fuzzy

(b) Histogram Equalization. Left side the histogram is grouped in the middle and right is distributed for all values.



(c) Binarization. Left side RGB image and right side image in white and black.

(d) Color adjustment. Left image with channel green saturated and right image with correction.



(e) Morphological Operations. Most common operations, dilation and erosion.

Figure 3.6: Image Pre-procesing methods

neighborhood has the value 0. It is used to remove floating pixels, and thin lines, strip away extrusions, and split apart joint objects. The equation defining erosion operation is described as follows:

$$A \ominus B = \{z \in E | B_z \subseteq A\}, \tag{3.7}$$

where A is the binary image, B is the structuring element, E is the Euclidean space or an integer grid, and B_z is the translation of B by the vector z defined as: $B_z = \{b + z | b \in B\}$.

Erosion operation can also be defined as follows:

$$A \ominus B = \cap_{binB} A_{-b} \tag{3.8}$$

The dilation operation expands the image pixels, or it adds pixels, by taking the maximum value of all pixels in the neighborhood. In a binary image, the pixel value is set to 1 if any pixel in the neighborhood has the value 1. It is used to make objects more visible, fill small holes, repair brakes, and repair intrusions. Dilation is defined as follows:

$$A \oplus B = \{z \in E | (B^s)_z \cap A \neq \emptyset\}, \tag{3.9}$$

where B^s denotes symmetric of B such that: $B^s = \{x \in E| - x \in B\}$. Dilation operation can also be defined as:

$$A \oplus B = \cap_{b \in B} A_b \tag{3.10}$$

Whit these two basic operations we can create more complex operations to remove small objects, fill small holes or find perimeters on objects. Opening and closing are the most common combinations of morphological operations.

Opening is obtained by first applying erosion of A by B followed by dilation of the output of erosion. It is defined as follows:

$$A \circ B = (A \ominus B) \oplus B \tag{3.11}$$

Closing operation is obtained by applying dilation of A by B followed by erosion on the resultant image. Closing operation is defined as:

$$A \bullet B = (A \oplus B) \ominus B \tag{3.12}$$

3.5.2 Watershed Algorithm

The watershed algorithm is a transformation defined on a grayscale used commonly for segmentation, i.e. for separate objects in an image. It is useful for counting objects or for the analysis of separated objects. The name comes from the geological watershed which separates adjacent drainage basins. The watershed transformation considers any gray-scale image as a topographic surface where intensity denotes peaks and hills when is high, and valleys when is low. We can see an example in figure 3.7

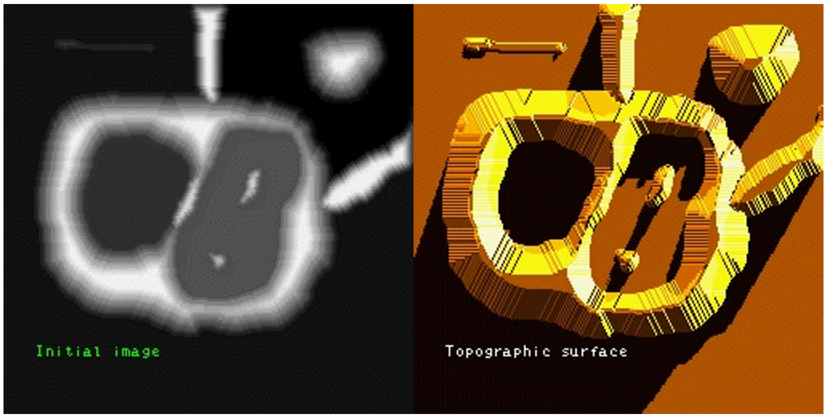


Figure 3.7: Topographic representation (on the right) of an grayscale image (on the right).

The algorithm starts by filling isolated valleys, local minima in the image, with waters coming from different sources representing labels. This transformation is applied to the image gradient such that water as the water rises, water from different sources will start to merge on the peaks. Every time different water starts to merge, a barrier in the location is created to avoid merging. This process is done until all peaks are underwater, thus, all the barriers that were created are the segmentation result. However, using the transformation in this base form produces over-segmentation, i.e., it separates a single object into several objects due to local irregularities in the gradient image. In order to solve the over-segmentation problem, a method name marker-controlled watershed can be used. This is done by specifying the initial water source will start, i.e. we can specify which valley points are to be merged and which are not. We can observe an example of initial markers in figure 3.8 where reed areas are the water sources.

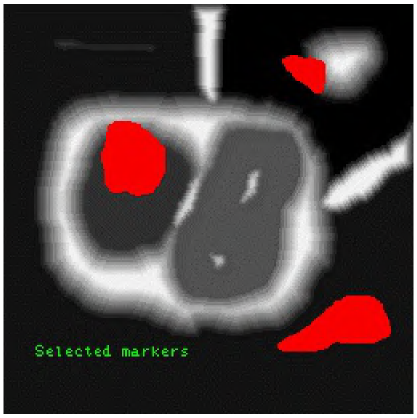


Figure 3.8: Initial markers where water will start to rise.

3.6 Machine and Deep Learning algorithms

3.6.1 K-Nearest Neighbors (KNN)

The K-Nearest Neighbors (KNN) is a simple non-parametric supervised machine learning algorithm that can be used for both classification and regression [42]. For the classification task, the output of the algorithm is a class membership. A new object is classified based on the vote of its nearest neighbors, where a class is selected by the most common class among its ”k” nearest neighbors. The algorithm relies on distance for classification that assumes similar objects are close to each other while different objects are far. On the other hand, the classification error is bounded above twice the Bayes error rate, which is the minimum error rate given the data distribution.

Each object is considered as a vector on a multidimensional vector space, for instance, a 2-dimension object consists of a pair of features (x, y). The algorithm does not require a defined training step since it only stores vectors and classes of the training set. The classification phase takes the k nearest neighbor, previously defined by the user, in order to classify a new unlabeled vector.

One of the most common distances used in the KNN algorithm is the euclidean distance that is computed between the test sample and the training set. Let x_i be an input sample of the test set with p features $(x_{i1}, x_{i2}, x_{i3}, ..., x_{ip})$, and x_j a sample of the training set with the same number of features. The Euclidean distance between the two points is defined as follows:

$$d(x_i, x_j) = \sqrt{(x_{i1}, x_{j1})^2 + (x_{i2}, x_{j2})^2 + (x_{i3}, x_{j3})^2 + ... + (x_{ip}, x_{jp})^2} \tag{3.13}$$

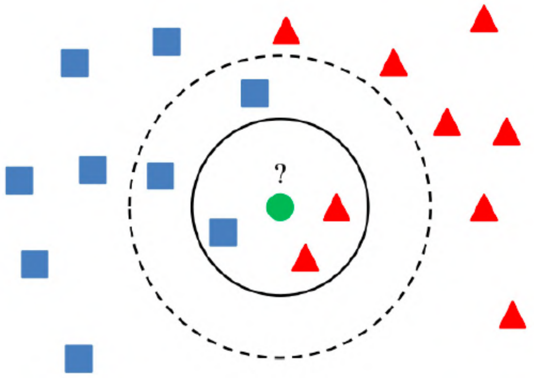


Figure 3.9: KNN classifier example. A new object (green circle) has to be classified. With k=3, algorithm takes two triangles and one square, then, the object is classified as a triangle. With k=5, two triangles and three squares are considered in the algorithm classifying the new object as a square.

With $k = 1$, 1-nearest neighbor, class is defined as:

$$C(x_i) = min_j(d(x_i, x_j)), \tag{3.14}$$

where j goes from 1 to n, and n is the number of objects in the training set. The algorithm assign the class of its closest neighbor in the feature space. When $k > 1$, the algorithm consider the next $k - 1$ minimums in the feature space and then the class is selected with the majority class.

Figure 3.9 shows a green circle, the new point, along with the training set represented as red triangles and blue squares. The classification can vary depending on the selected number of k neighbors. For instance, let $k = 3$, for the new object green circle the algorithms takes two triangles and one square as the nearest neighbors, the majority class is the red triangle, thus, the green circle is classified as a red triangle. A different result is caused when $k = 5$, the algorithm takes three blue squares and two red triangles, thus, the circle is classified as a blue square.

3.6.2 Decision Tree

Decision trees are another non-parametric supervised learning algorithm used for classification and regression [5]. It is a flowchart-like structure in which nodes represent ”tests” on a given attribute, each branch represents the output of the test, and leaves represent class labels. Paths from the root to the leaf are called classification rules.

The tree is built as follows: Given a training vector $x_i \in R^n, i = 1, ..., l$ and a vector containing the classes $y \in R^l$, the decision tree recursively splits the feature space such that objects with the same class are grouped together. A data node m can be represented by Q_m containing n_m samples. Each split $\theta = (j, t_m)$ consist on a feature j and a threshold t_m, data is divided into two subsets $Q_m^{left}(\theta)$ and $Q_m^{right}(\theta)$ such that:

$$Q_m^{left}(\theta) = \{(x, y) | x_j \leq t_m\} \tag{3.15}$$

$$Q_m^{right}(\theta) = Q_m \backslash Q_m^{left}(\theta) \tag{3.16}$$

In order to choose the best partition, it is necessary to compute the quality of a feature candidate of node m by using an impurity function or loss function $H()$.

$$G(Q_m, \theta) = \frac{n_m^{left}}{n_m} H(Q_m^{left}(\theta)) + \frac{n_m^{right}}{n_m} H(Q_m^{right}(\theta)) \tag{3.17}$$

Then, the parameter is selected such that it minimizes the impurity by:

$$\theta^* = \text{argmin}_\theta G(Q_m, \theta) \tag{3.18}$$

Decision tree is created by recursively creating subsets $Q_m^{left}(\theta^*)$ and $Q_m^{right}(\theta^*)$ until the maximum depth is reached, $n_m < \text{min}_{samples}$ or $n_m = 1$

We can choose between two options to compute the impurity of a split, gini and log loss, also called entropy. The gini function is defined as follows:

$$H(Q_m) = \sum_k p_{mk}(1 - p_{mk}) \tag{3.19}$$

And the Entropy function is defined as:

$$H(Q_m) = -\sum_k p_{mk} \text{log}(p_{mk}) \tag{3.20}$$

where $p_m k$ is the proportion if the class k in a node m defined by:

$$p_{mk} = \frac{1}{n_m} \sum_{y \in Q_m} I(y = k) \tag{3.21}$$

Figure 3.10 shows an example of a decision tree to decide whether a person can do an activity or not depending on the weather of the day. Outlook was selected as the first feature to split the data. This feature contain three variables so the data was divided into three parts. On the second level, when outlook is overcast the decision is ”Yes”, and for the other values, a second feature is selected. For instance, we can define a rule as (Outlook=Sunny and Hum=Normal) = Yes or (Outlook=Rain and Wind=Strong)=No.

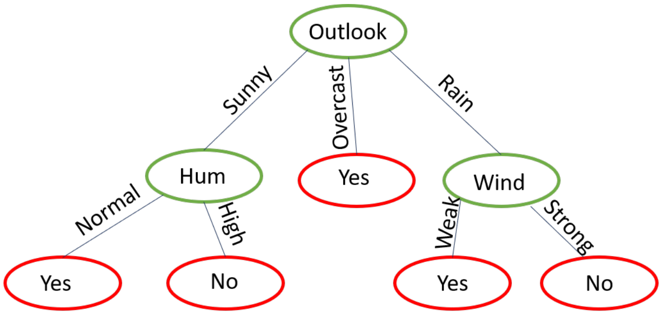


Figure 3.10: Example of a decision tree to decide whether realize an activity or not depending on the weather

3.6.3 Random Forest

The random forest classifier, created by Kam, T. [23], is an ensemble machine learning algorithm used for classification and regression tasks that implement several decision trees where the output is selected by the majority class given by the trees. The fundamental of random forest is the low correlation of the trees that operate as a committee voting for a class. The base of uncorrelated trees is given by two methods, bagging and feature randomness.

The first technique is feature randomness. In a typical decision tree, a tree is constructed using all available features in the training set. In a random forest, trees can only choose a random subset of features in the training set. This condition creates more variation for all trees causing a low correlation among trees and creating more diverse trees.

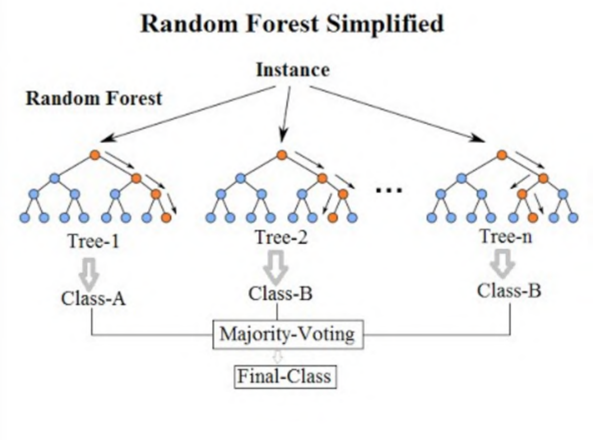


Figure 3.11: Random forest classifier diagram. Several trees are created and each of them give a vote to have the final class

For the second technique, bagging, decision trees can have different structures with small changes on the training set since they are very sensitive to the data. This condition can be reached by selecting a random sample with a replacement of the training set. The procedure is the following:

Let $X = x_1, x_2, ..., x_n$ the training set and $Y = y_1, y_2, ..., y_n$ the corresponding class. The bagging algorithm is repeated B times to create decision trees. For each $b = 1, 2, ..., B$: 1. A new sample with replacement of n elements from X and Y is selected; these are X_b and Y_b. 2. A decision tree f_b is trained with sets X_b and Y_b.

After training all decision trees, the prediction for a new sample $x^{'}$, when used as regression tree, is defined as follows:

$$\hat{f} = \frac{1}{B} \sum_{b=1}^{B} f_b(x^{'}), \tag{3.22}$$

where $f_b(x^{'})$ is the prediction of the tree b. On the other hand, for a classification tree the majority vote is used to select the final class, similar to KNN algorithm but in this case we use predictions of trees to get votes.

The figure 3.11 shows the general structure of the random forest algorithm. This consist of several decision trees, n number of trees, where each of them is different. As we can note, for a new instance the tree follows a path (nodes in orange) predicting a class. In the final step, the algorithm takes the majority class among all trees to give the final class.

3.6.4 Fully Convolutional Network (FCN)

Before the invention of Fully Convolutional Networks (FCN), models required fixed-size images causing problems when resizing images, creating distortions, or even over-fitting when using pre-trained models. In order to solve such problems, Shelhamer et al. [49] created the FCN model that is capable to receive arbitrary sizes to produce a segmentation of the output of the corresponding size. The FCN is a neural network architecture that only contains convolutions, sub-sampling, and up-sampling operations. The model takes as base a Convolutional Neural Network (CNN), used for image classification, that consists of several layers to extract features where the size is reduced until creating a large vector, and goes through the output layer that contains fully-connected layers to give the final classification [50, 32]. Then, Fully connected layers of CNN are converted into convolution layers in order to create a heat map on the output for image segmentation, in addition, more layers as well as a spatial loss are added to guarantee dense learning. Finally, a 1x1 convolution of $n + 1$ channels is added, where n is the number of classes in the dataset, to predict scores of each class including background. Figure 3.12 shows the general architecture of the model, we can note a down-sampling path where features are extracted, and then the up-sampling for pixel-wise prediction to create the final segmentation.

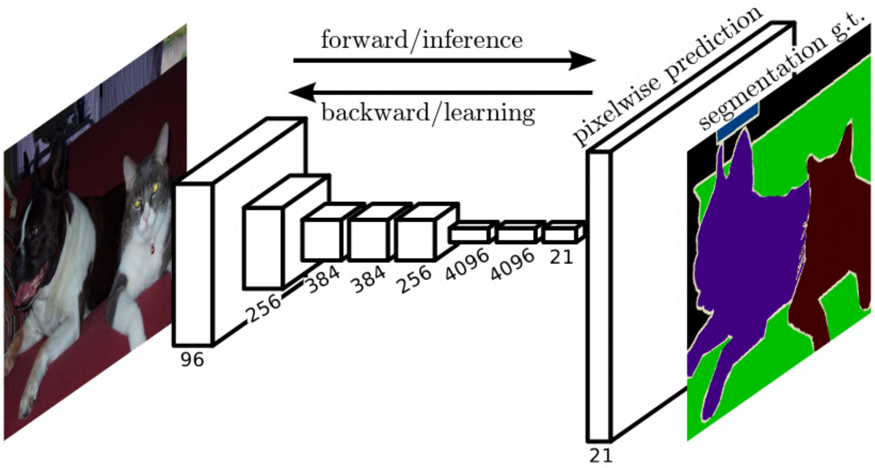


Figure 3.12: FCN architecture

3.6.5 U-Net

For the second deep learning model, we will use a deep neural network known as U-Net for area segmentation. U-Net is an architecture that uses Convolutional Networks and was designed for biomedical use so the interest area is identified by pixels, the name comes from the letter ”U” that is formed by the structure [45]. Although its main use is in the medical sector, it has been explored for the segmentation of objects and also for Weed Mapping [47] and seems to have good results. That is one of the reasons why we choose this model for the problem.

In a simple view for an easy explanation, we can divide the structure showed in fig. 3.13 into three general steps. Each steps will be described next.

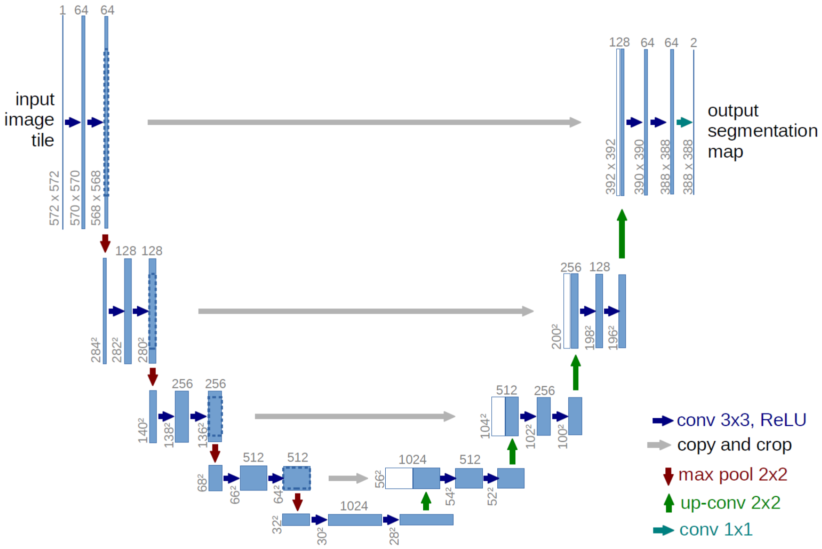


Figure 3.13: Structure of the U-Net model

- **Feature extraction**

  The first step when receive the input image, we are in the left side of fig. 3.13. In this step they apply a conventional convolutional network applying 3x3 convolutions with an activation function Rectified Linear Unit (RELU). Going down, in each row a set of feature maps is achieved and saved for a future use.

- **Classification**

  In this step is where, by the computed feature maps of the past step, each pixel is mapping to the corresponding class. This step is in the bottom of the structure.

- **Reconstruction with segmented areas**

  The final step is to re-construct the image with pixels identify. This is a reverse process of the first step but, since the image is being extended, there are space that needs to be filled. This is when the saved information from step one is used, with this information space are filled. In the structure of U-Net, we can see gray arrows identify as ”copy and crop”.

3.6.6 Vision Transformer Segmenter

The last and most recent model is the Vision Transformer for image segmentation (ViT Segmenter). Transformers, first introduced by Vaswani et al. [53] for natural language processing (NLP) is based on the attention concept. The model extract features using a self-attention mechanism to understand the important all words in the sentence. Architecture is formed by two parts, an encoder, and a decoder containing blocks of attention and a feed-forward network. In general, self-attention is a sequence-to-sequence operation where a sequence of vectors X goes in, and a sequence of vectors Y goes out. In order to produce a vector y_i self-attention operation performs a dot product using a weighted average over all input vectors. After the creation of transformers, they have been extended to other fields such as natural language processing, computer vision, and audio processing [35]. The variant of transformers is known as X-formers where the aim is to improve the original transformer model from different perspectives including Model Efficiency, Model Generalization, and Model Adaptation. In the Computer Vision task, we can find an application for Video Understanding, Image and Scene Generation, Segmentation, and Object detection [29].

The vision transformer (ViT) Segmenter is based on the previous work by Dosovitskiy et al. [13] for image classification. Strudel et al. [51] extended the original ViT to be used for the semantic segmentation task by adding a mask transformer to the model. ViT segmenter consists of two parts as can be seen in figure 3.14 where the left side, before the arrow, is the ViT or Encoder part and the right side, after the arrow, is the segmentation part or Decoder. It starts by splitting the image into fixed-size patches, a portion of the input image, then, they are linearly embedded, a position embed. The resultant vectors enter the Transformer Encoder that contains the self-attention mechanism used on the first transformer. Finally, an MLP with one hidden layer is used for the classification in the case of image classification. On the other hand, the process continues for segmentation. For the segmenter part, right side of figure 3.14, the sequence of patches, and output of the encoder pass through the Mask Transformer that generates a set of class masks by computing the scalar product between class embeddings and path embeddings. A softmax is applied followed by a layer norm to obtain the final segmentation map.

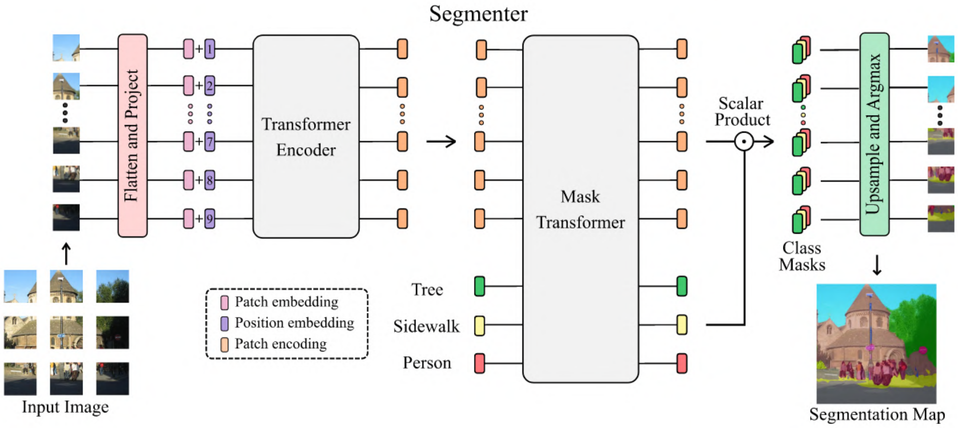


Figure 3.14: ViT Segmenter structure. The left side (before the arrow) Encoder is the original ViT, while right side Decoder present Mask Transformer to be used for semantic segmentation

3.7 Metrics to compare models

Since we are comparing models, machine learning, and deep learning, we need to define the scores that measure the performance of models in order to select the best one.

3.7.1 Pixel Accuracy

The pixel accuracy metric, a metric that can be use for classification and segmentation, reports the percent of pixels that were correctly classified. This metric can be reported for each class as well as globally across all classes. The formal definition of accuracy is:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \tag{3.23}$$

When we consider the per-class accuracy, considering just one class at time, we are doing a binary classification. In this case, accuracy can be defined in terms of positives and negatives as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.24}$$

Where TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

On the other hand, the overall accuracy can be affected when working with imbalanced classes, i.e, there are more pixels of one class than the others. For instance, suppose a class covers 90% of the image, and we have a model that can always predict correctly that class but other classes predict incorrectly, the overall accuracy of the model would be 90%. We can think this is a good model, however, if we look at the other classes we would see that the model is actually bad since it can only predict one class correctly.

3.7.2 Intersection Over Union

The Intersection Over Union (IoU), also known as Jaccard Index, represents the ratio of overlap area between the target mask and the prediction output. In other words, IoU metric measure the number of pixels that can be found in both the target mask and the prediction mask divided by the total number of pixels present in both masks. Intersection over Union metric is defined as follows:

$$IoU = \frac{target \cap prediction}{target \cup prediction} \tag{3.25}$$

The equation shows the two parts of the metric. The intersection (or overlap), represented as $A \cap B$, are the pixels that can be found in the prediction and target mask, while the union part, defined as $A \cup B$, are all the pixels present on the prediction or the target mask. This equation can be visualized in figure 3.15 where we can distinguish between the intersection part and the union part.

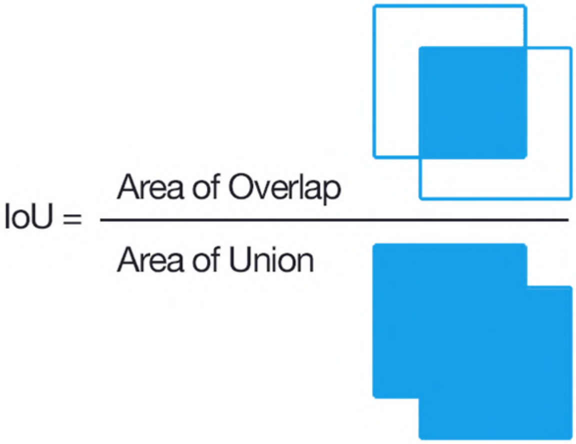
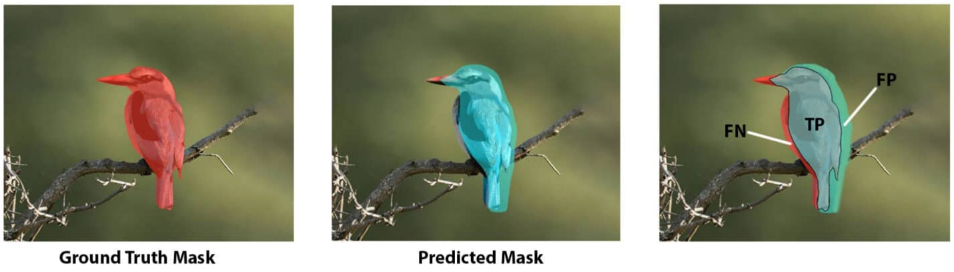


Figure 3.15: Intersection over Union metric representation

This metric can also be defined in terms of positives and negatives such as accuracy metric. Figure 3.16 shows an example of how the metric is used to measure the performance of the prediction. IoU equation can be written as follows:

$$IoU = \frac{TP}{TP + FP + FN} \tag{3.26}$$


Ground Truth Mask

Predicted Mask

Figure 3.16: Intersection Over Union metric on segmentation task

Chapter 4

Methodology

4.1 Introduction

In this chapter, the method used in the project is described defining a pipeline with all steps for the final classification. Starting from data acquisition to test classifiers for semantic segmentation. It is important to have a defined sequence of steps to follow for the replicability of experiments. In the next sections, the process is going to be explained detailed step by step in order to have a clear understanding of the pipeline as well as reasons to follow this methodology.

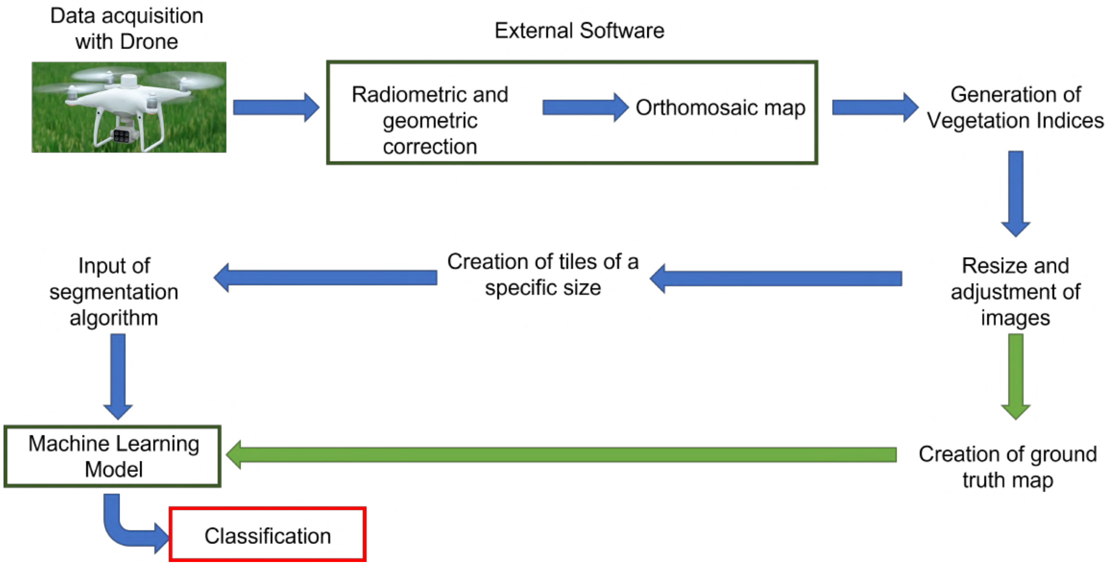

Data acquisition with Drone
External Software
Radiometric and geometric correction
Orthomosaic map
Generation of Vegetation Indices
Resize and adjustment of images
Creation of tiles of a specific size
Input of segmentation algorithm
Machine Learning Model
Classification
Creation of ground truth map

Figure 4.1: Pipeline for this project

Figure 4.1 shows the pipeline used in the project to segment areas in an orthomosaic map starting with data acquisition with the drone, then radiometric correction of images and orthomosaic map using external software, creation of vegetation indices using the orthomosaic map, input for algorithms and classification. This chapter is divided into several sections separated by the type of task involved in the process, sections are the following:

1. **Data Acquisition**, where we give a description of the drone used to capture images as well as configuration used. In addition, we provide details about locations and dates when data was capture.
2. **Data Processing**, in this chapter we are explaining the process to generate the orthomosaic map using an external software. Using the orthomosaic we can generate desired vegetation indices that are used as the input of algorithms.
3. **Deel Learning Algorithms** is dedicated to mention algorithms used as well as the configurations of them.

As an additional objective, we have included the task of counting the number of agaves in the crop. Even when is a different task, it can be included as a branch in the pipeline just after the adjustment of the image. In the last the section, the methodology followed for this task is going to be explained.

4.2 Data Acquisition

As mentioned in previous chapters, there is no agave dataset to work with, and for that reason, the creation of an agave dataset was necessary for this project. The drone we are using throughout the project is the "P4 Multispectral" by DJI, this device was provided by ITESM Campus Guadalajara to make easier the development and the success of this project. Additionally, the institution provided an iPad necessary to use the software, which is available just for iOS devices, to control the drone and collect the information.

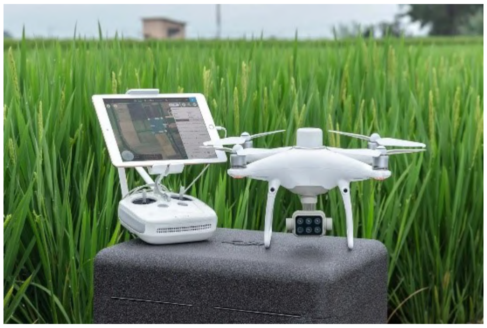


Figure 4.2: Drone P4 multispectral equiped with six cameras

The P4 Multispectral (see figure 4.2) is a drone designed by DJI for precision agriculture and environmental monitoring use providing features and tools for a quick interpretation of what is recording the drone even in real-time. The principal characteristic that makes this drone perfect for these applications is the implementation of a Multispectral image system (here is the "why" of the name of this drone) that in general terms is a set of cameras that can capture several ranges of the wavelength of light. The set of cameras are the following: the typical camera RGB; Red Edge (RE) captures the band 730 nm ± 16 nm; Near Infrared (NIR) captures 840 nm ± 26 nm; Green (G) with 560 nm ± 16 nm; Red (R) 650 nm ± 16 nm and Blue with 450 nm ± 16 nm. The drone also has a light sensor to capture the intensity of the sun and adjust all images with the correct illumination. By capturing a Multispectral system, we can use the Vegetation Index (VI) which is a combination of the different channels, the drone can process the Normalized Difference Vegetation Index (NDVI) in real-time. This device also implements a system to include positions in real-time and measures of what is being captured.

Data was captured in an agave crop located in Zapopan, Jalisco, this crop contains several conditions such as different soil levels and separations of agaves that makes it useful for this project. The crop was divided into several zones delimited by natural roads and divisions with other kinds of a plantation. We captured data flying at the same altitude to have the same spatial conditions. All zones have different conditions, different sizes of agaves, and the presence of weeds and water. In the next chapter, Analysis, and Discussion of Results, we are giving more details about the dataset, including the size of the captured map, the number of zones we captured, and the GSD for orthomosaic maps.

4.3 Data processing

Data processing involves all the data transformation done before entering the algorithm for the image segmentation and represents almost the entire process of the whole task. These steps are mentioned in the pipeline (fig. 4.1) and go after data acquisition to before the input of the segmentation algorithm. In general terms, the data processing is divided into the following steps:

- Radiometric correction
- Creation of the orthomosaic map
- Extraction of Vegetation Indices
- Adjustment of the image
- Creation of tiles
- Creation of the ground truth map

Radiometric calibration along with the creation of the orthomosaic map is considered the same step since they can be computed at the same time. On the other hand, vegetation indices have their section since the equation is required to generate the index. At this moment, we have generated the necessary data to train the models, however, images need to be adjusted in order to have the proper size and characteristics algorithms need to work properly. The next stage is to transform images in the correct format for each algorithm. Finally, we explain the way to create the ground truth label.

4.3.1 Radiometric correction and orthomosaic map

Once data was captured by the drone, images are processed in order to generate the orthomosaic map. Using the drone we can create a flight plan where we select the area to map, each flight plan creates a set of images that contains five channels (R, G, B, NIR, and Red-Edge) of the same size that contains Geographic Information given by the GPS of the drone. Before the creation of orthomosaic maps, it is necessary to align individual channels to have the same scene captured, correcting defects caused by variations of conditions, additionally, values of pixels of images are saved into a format for the correct visualization in a computer but it needs to be transformed into reflectance values. DJI provides a guide for image processing taking Near Infra-Red (NIR) band as the central band to be aligned. The guide gives a sequence of steps in order to first, generate individual bands reflectances and then, correct channels due to the effects of the camera, corrections include vignetting correction and distortion calibration; the next step is the alignment of the phase difference, which is caused by the camera position, based on the NIR band and then an additional alignment due to different exposure times. The difference of positions can be seen in figure 4.3 where the same scene was captured but there is an offset between images.

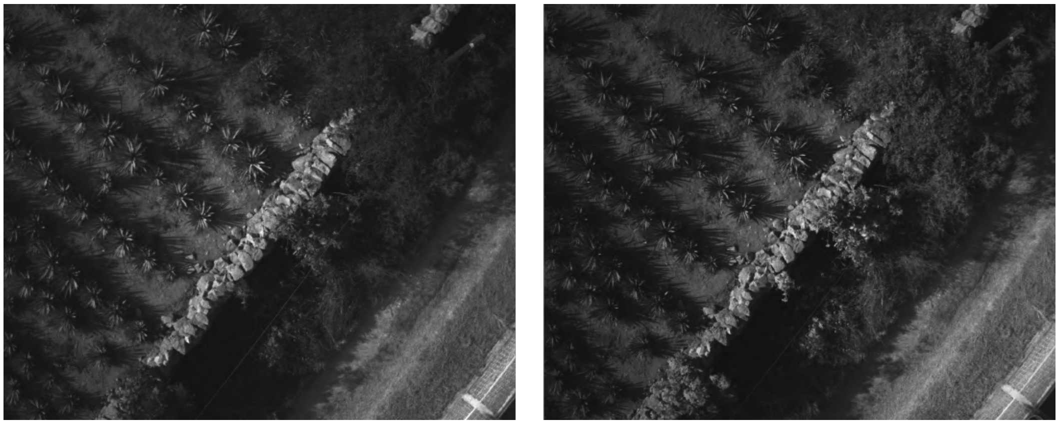


Figure 4.3: Channels blue and green in left and right respectively. The two images looks similar but they are not aligned. We can see a difference in the bottom-right corner in the wall

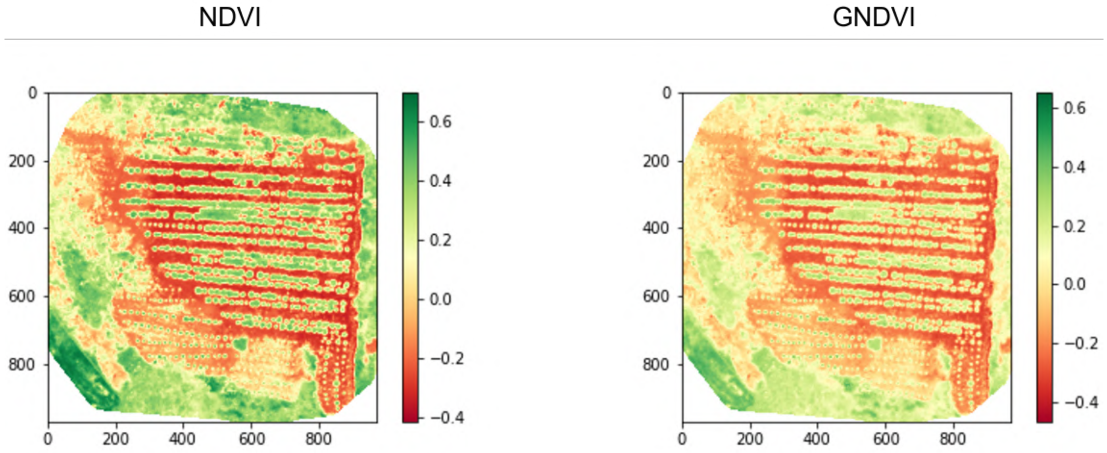
The manufacturer recommends the use of an Enhanced Correlation Coefficient (ECC) for the alignment of all bands but applies an edge detection filter to have better results with ECC. Once all channels are aligned with the hole flight plan, we can generate the orthomosaic map by performing image stitching to join all images into a single one. The output image is an orthomosaic map with five channels. This process can be used to generate the map from scratch however, there are already many options for this task that just needs the set of images to stitch and the software will generate the map.

P4 multispectral drone has software for a number of tasks such as 3D reconstruction, mapping, measurements, and more for applications in construction, infrastructure, public security, filming, and agriculture. Nonetheless, agricultural use, which is the area of this project, is limited since it can create vegetation indices NDVI, GNDVI, NDRE, LCI, and OSAVI, but there is no tool for vegetation indices calculation. Another popular option is Pix4D, this software for photogrammetry use has more applications than the past one and has a specialized version for agriculture with a vegetation index calculator. This second software is more powerful and makes the task easier. On the other hand, there is an open-source option called Open Drone Map (ODM) that can generate the orthomosaic map but without vegetation indices. Like other open-source options, it has limitations and requires learning how to use it.

4.3.2 Creation of vegetation indices

After creating orthomosaic maps for a flight plan, we can use individual channels in order to generate vegetation indices by applying equations from chapter 3. Vegetation indices use reflectance values to compute the result as well as channels aligned so that a pixel represents the same information in all channels. For instance, to compute the NDVI we would need Red and Near Infra-Red channels and use the proper equation(see equation 3.1) to finally obtain the index. For each VI we use the corresponding equation.

In total, we generate six vegetation indices including NDVI, GNDVI, NDRE, MSAVI, ARVI, and NDWI, each one with unique information. Figure 4.4 shows NDVI and GNDVI where two indices look similar but one has light colors (GNDVI), however, the range of values is different and it can be seen next to the graph, while NDVI has values higher than 0.6 GNDVI limit are near to that value. The reason is that the channels they use are not the same by changing Red for blue and the operation is inverted, in addition, vegetation indices have a different meaning. Since both maps have unique values and ranges (the VI generated), they are considered extra data.


NDVI

GNDVI

Figure 4.4: Vegetation indices NDVI on the left and GNDVI on the right

On the other hand, in figure 4.5 there is a clear distinction between VIs, starting with the color of the map whit colors red and yellow with a few of green and the range of values that goes almost to 0.4, unlike GNDVI that reach 0.6. NDRE works better with the more mature crop when NDVI gets saturated, this is seen in the orthomosaic map where green values appear a few. In this example, agaves are in the initial years of their growth. On the flip side, MSAVI looks very similar to GNDVI because of the color, but MSAVI goes from 0.2 to 0.3. We have included two more channels to the dataset providing more information to the algorithms.

Finally, figure 4.6 shows ARVI and NDWI. The first Vegetation Index was intended to be used as a correction caused by atmospheric particles in the environment but seeing the map generated it looks like everything is in red but agaves highlight with a light color (in yellow). However, all the agaves do not have a clear distinction between them. Even with this characteristic, it can provide information to distinguish the soil and green areas.

NDWI is the vegetation index to measure moisture conditions in an area. Unlike others

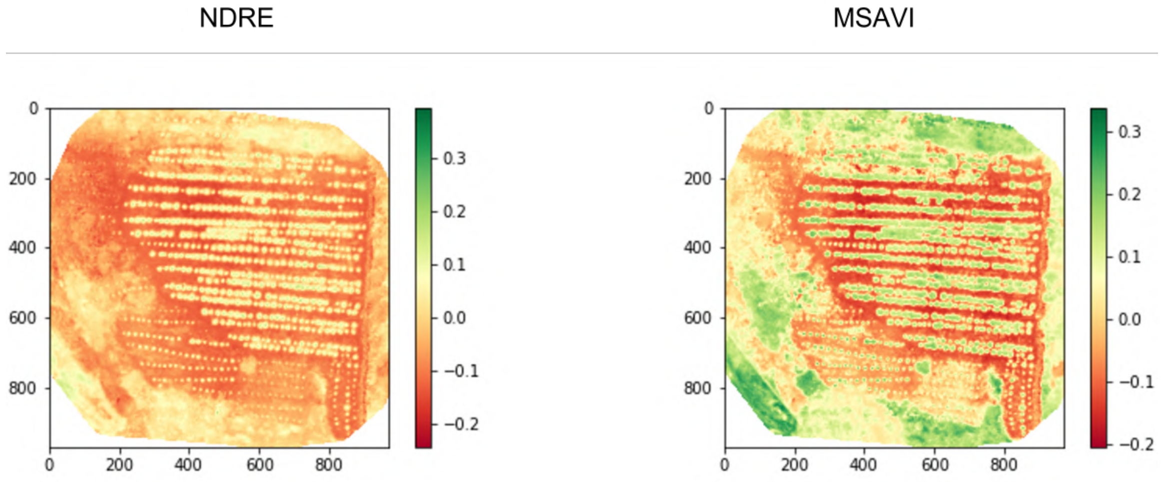


Figure 4.5: NDRE vegetation index on the left and MSAVI on the right

vegetation indices, NDWI highlights areas with more quantity of water in green while vegetation has lower values. This is contrary to the others Vegetation Indices that highlight green areas and soil have low values. We can note that the negative value is higher than the positive one, the opposite of NDVI. As we can note, all vegetation indices have different values to characterize the crop. Furthermore, the range of values of the same map is different standing out specific characteristics on each one.

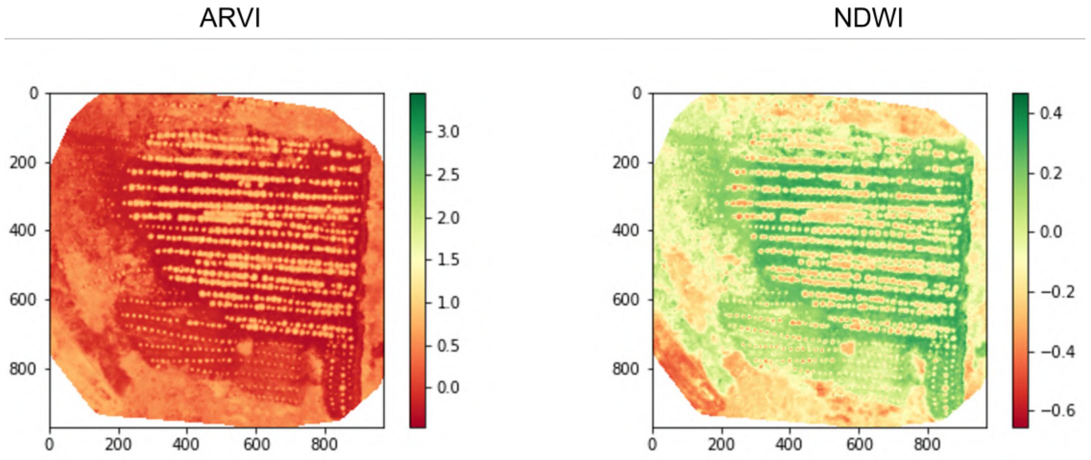


Figure 4.6: Vegetation indices ARVI and NDWI on the left and the right respectively

Originally the map consist of five channels (red, blue, green, NIR, and red-edge), but after computing all VIs we create an image with eleven channels, all of them aligned with their respective meaning in the health of the crop. At this point generated Vegetation Indices are considered as another channel in the image but in a different range of values, while original channels (Red, Blue, Green. NIR, Red-Edge) have values from 0 to over 0.3, vegetation indices go from -1 to 1.

Thanks to the implementation of vegetation indices in the agricultural topic we can extract useful information not seen with the naked eye by combining bands that capture different ranges of light wavelength producing new visualization explaining the status of the entire crop such as nitrogen levels, moisture condition or the health of individual plants. The vegetation indices stand out from RGB images since they highlight areas of interest and by combining several VIs we can obtain more precise information.

4.3.3 Adjustment of images and Creation of tiles

Just after the creation of vegetation indices, explained in the previous section, we need to perform an adjustment othomosaic maps in oirder to use it in the algorithms. The first adjustment is in the values of the images and the second one in the size.

As we observe in the creation of vegetation, all maps have different values in the same orthomosaic map and some of them manage a wide range of vales. Moreover, the five bands also have different range of values. Those conditions create a variety of values creating disperse data. We can see how values of each channels vary in the table 4.1.

Table 4.1: Minimum and Maximum values of each band

Band	Min Value	Max Value
Red	0.0084	0.3067
Green	0.0083	0.3073
Blue	0.0153	0.3087
NIR	0.0216	0.2872
Red Edge	0.0117	0.2986

We can observe that values of all channels stays in the same range of values and are very close of each others. On the other hand in table 4.2 we can observe minimum and maximum values for six vegetation indices. As described in the previous chapter, most of VIs have values in the range of -1 to 1 and there is a clear distinction in one of them, ARVI exceed the maximum value of the others vegetation indices. Bands values and vegetation indices values have different ranges so they need to be normalized. In this step we explore some methods to normalize the data of all channels and vegetation indices to avoid problems with the weights of them.

Table 4.2: Minimum and Maximum values for Vegetation Indices

VI	Min Value	Max Value
NDVI	-0.4157	0.7021
ARVI	-0.4678	3.4657
GNDVI	-0.4684	0.6547
NDRE	-0.2451	0.3950
MSAVI	-0.2047	0.3392
NDWI	-0.6547	0.4684

The second adjustment is related to the size of the images. Some algorithms just accept

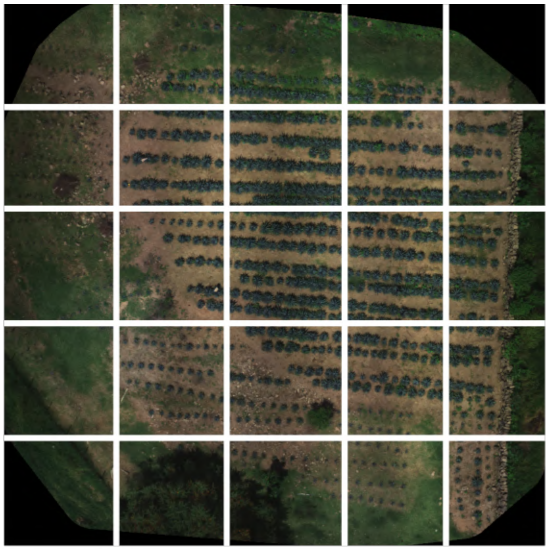


Figure 4.7: Tiles obtained in an orthomosaic map

an specific size of the input image to work. In order to always have the correct input, commonly images are resized in the correct size before use it in the algorithm. Due the importance of the detail of the image to classify each plant, we have decided to follow a similar approach given by Sa, I. et al. [47] where the image is divided into tiles of the desire size, for example, in the case of the u-net the algorithm needs an input of 160x160 so the original image is divided into smaller images of that size. Figure 4.7 shows the obtained images derived from the orthomosaic map. By following this method we can preserve the detail of the images while generating the dataset so that for a single orthomosaic map we can extract several images, depending of the size of the original image an the size of the tiles.

4.3.4 Ground truth map

An important step in the implementation of machine learning for classification or segmentation is the creation of labels of the images because it defines the classes for the classification and separate the objects for the rest of the image. We can observe an example of a ground truth map in figure 4.8 where we find three classes in the orthomosaic map, the first class in green represent the Agave class, the second in yellow are the trees and the rest of objects in purple are labeled as another class.

Just like the orthomosaic map, the ground truth map is divided into tiles of the same size such as the same area of a tile is captured into the orthomosaic map and the ground truth. The process of the creation of the tiles is done for both maps at the same time. First both images are loaded and compared in order to have the same size, then the orthomosaic is taken as the template to select an area. After the selection, the area is cropped in both images at the same position so that we have generated a new image with they respective label. The process is repeated for all images used for the training phase.

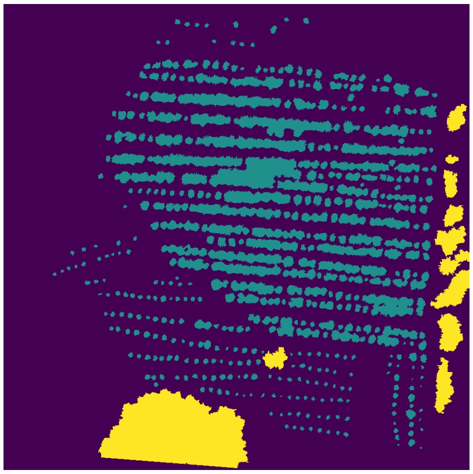


Figure 4.8: Ground Truth map of an orthomosaic map for three classes. Green for Agaves, yellow for trees and purple for the rest of objects

4.3.5 Machine learning format

Unlike deep learning models that are trained with the images without any changes, except for the pre-processing step, data in machine learning requires an specific format such that an instance is described by a set of features that are used by the models. For instance, on image classification, an image is transformed by creating a vector of pixels, i.e. each pixel represent a feature describing the image. Then, an algorithm uses all the features to predict the target label. If we have a 200x200 image, the total features that describes it are 40,000. The next table is a data format representation of an image, we can observe that each pixel is a feature and there is a label for each image. It is clear that there is a huge number of features on a 200x200 image size, for this reason image format is also accompanied by a Principal Component Analysis in order to reduce the number of features.

Table 4.3: Feature representation on an image

Index	Pixel 1	Pixel 2	Pixel 3	Pixel 4	...	Label
Image 1	23	23	123	75	...	1
Image 2	93	67	80	13	...	0
Image 3	54	12	69	38	...	1

On the image segmentation task, all individual pixels needs to be classified so we follow a different approach. Lets define an RGB image of 5x5 pixel size and their respective ground truth table of the same size. The image consist of three channels (red, green, and blue) and there are two objects inside, a tree and a building. The ground truth has the label for the tree on yellow and blue for the building. This can be seen in the figure 4.9. The tree is covered by two pixels while the building is covered by six.

As mentioned before, the procedure for image segmentation is different to classification task where pixels are features. In this case since classification must to be done for individual pixels, instead of the entire image, we need to take a pixel as an instance and a band as the feature. In our example, a pixel contains three values for the three channels, so, each one has three features an one correspondent target label.

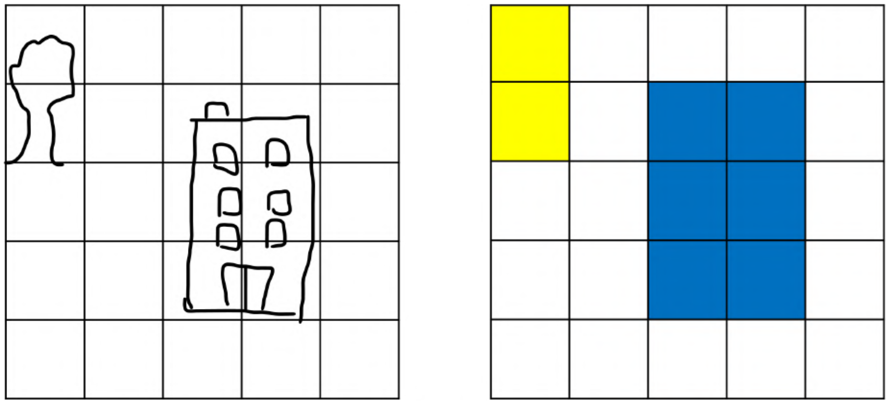


Figure 4.9: Input image on the left with a tree and a building, and the correspondent ground truth label on the right.

Figure 4.10 shows the format mentioned before where we can see that the tree is on the 3 channels representing the feature and there is also a label for each pixel. This means that, in this case, we have 25 instances and three features that models will use. By using this format, we can provide a lot of instances to machine learning models. In the application in the project, the channels are not just RGB but all channels available in the camera as well as Vegetation Indices. Thus, the number of features will vary depending on the selected channels.

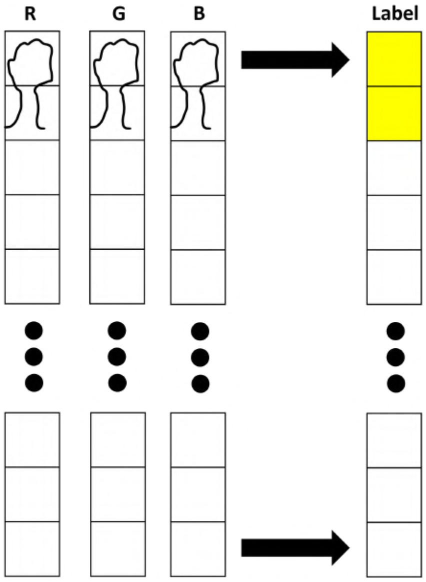


Figure 4.10: New data representation for machine learning models of an RGB image and the correspondent label

4.3.6 Watershed algorithm

Image semantic segmentation is a pixel-level classification forming clusters that represent objects in an image. Using this technique we are not able to identify individual plants in the crop to count them. In addition, we face a condition where a single agave is classified into two classes due to the difference in the center and the edge. In order to fix this condition and to separate agaves on the segmentation, we have created an additional step that is applied to the final prediction of models. We included this process in this section since it uses techniques described in the pre-processing section even when it is applied at the final of the segmentation.

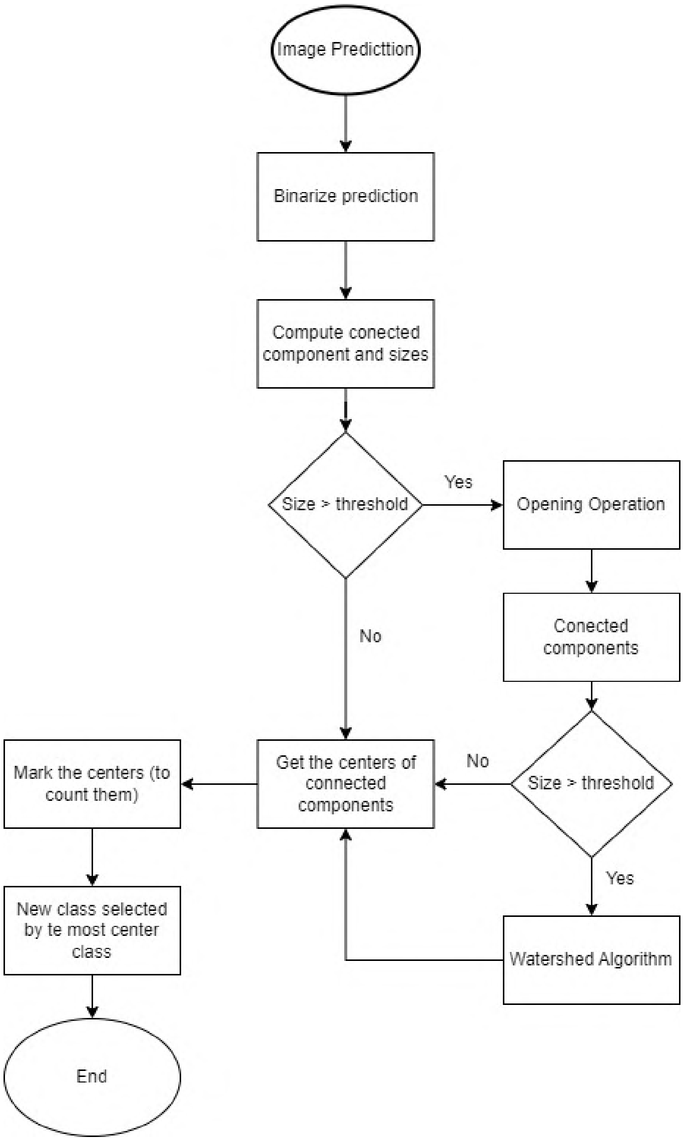


Image Predicttion
Binarize prediction
Compute conected component and sizes
Size > threshold
Yes
Opening Operation
Conected components
No
Size > threshold
No
Yes
Get the centers of connected components
Mark the centers (to count them)
New class selected by te most center class
Watershed Algorithm
End

Figure 4.11: Flow diagram of the final process to separate agaves

Figure 4.11 shows the flow diagram with steps to create the final prediction of an image. We start by providing the image prediction of any algorithm, in the case of machine learning model we need to do the inverse transform of the format described in the past subsection. The prediction is an image containing the segmentation of pixels with values corresponding to the class. The next step is to create a copy and binarize it in order to perform morphological operations. We continue by computing connected components to identify the segments of clusters created by the model as well as their sizes of them. Then we separate clusters based on their size using a defined threshold. This step is to divide clusters from those that contain a single agave to those containing a group of them. We expect that larger clusters contain more than one agave since the covered area is bigger. For the small-size group, we continue with a step we are explaining later. The next step on the big size group is to apply opening morphological operation in order to reduce some noise, small points segmented that are not actually agaves. Additionally, this operation also helps to separate some agaves since sometimes they are joined just with a thin line. Then, we compute connected components and sizes, and we separate again clusters based on the size using a threshold. This time we are ensuring large clusters contains more than one agave so we can apply a process to identify single agaves inside the cluster. As we did we the previous small group, we are explaining the next step later. For the big-size group, we continue by applying the watershed algorithm. The algorithm is able to separate the agaves in order to identify each one as well as the centers. At this point, we have several clusters in which each there is a single agave. The following step applies to the small-size clusters as well as those generated by the watershed algorithm. We start by joining the splits into a single image, then we get the center of the clusters. For each of the centers, we mark in order to count the number of agaves in the image. Once we have counted the agaves, we use the center to generate the final prediction or segmentation. As we commented before, there are cases where agave is segmented into two classes, so, in order to solve this problem, we use the class that is most centered in the agave since the center is the most important part of the plant. The image obtained after marking the centers is a binary mask with zeros and ones representing areas where there is agave, thus, we take the mask to multiply it with the initial input image, the original prediction of an algorithm, such that all agaves are separated. From the output of this operation, we use the centers and define a small area using a circle, such that we are covering the same distance from the center to all directions. Then, we take the most frequent class, or majority class, in the covered area and we expand it to the whole cluster. Using this technique we can have a single class in the agaves that are already separated. We end up with a segmentation image where all agaves can be identified individually and just contains a single class that describes the agave, in addition, we can know the number of agaves in the image.

4.4 Machine and Deep learning algorithms

As commented in the previous section, in this project we are going to implement several models in order to select the best method for this specific task of segmentation of the agave based on health. The algorithms selected are the following:

Machine Learning

- KNN
- Random Forest
- Decision Tree

Deep Learning

- U-Net
- FCN-8
- ViT Segmenter

We implemented deep learning models using the mmsegmentation toolbox [11] since it is easy to implement and we can find several algorithms including those mentioned before. Mmsegmentation toolbox allows us to create models by creating a configuration file where we specify the model, data pre-processing, and data augmentation as well as the pipeline to be used on the training and test steps. For each configuration file, we specified the model we want to use including U-Net, FCN-8, and ViT Segmenter.

For the models we included data augmentation techniques to enhance the results of all algorithms, techniques were resized on several scale ranges, crop, random flip, and padding. In addition, image pre-processing like normalization is included in the pipeline of models. We also use an iteration approach for training models instead of epochs like common training since this is faster when using the toolbox. In addition, the tool also provides scores that are most used for semantic segmentation by default including accuracy and intersection over union (IoU) on average and per class. Graphs are also computed to visualize accuracy and loss over iterations. Depending on the size of the dataset we select the number of iterations, as well as the frequency model, which is validated. In the next chapter, we are discussing more configurations of each model, the optimizer used, and the learning rate.

On the other hand, for machine learning models we use the scikit learn library to train models [41]. This library already contains the models and it is easy to use since they only need to be initialized, with a number of parameters defined by the user, and then train with the data in the correct format. Since we want to compare these models and deep learning models, we are using the same metrics, accuracy, and intersection over union.

Chapter 5

Analysis and Discussion of Results

In this chapter, we are presenting and discuss the results obtained from the project. First, we start with the data captured from the agave crop giving details about the dataset created for the project. Then, we continue with the comparison of algorithms where scores like Overall Accuracy are compared to choose the best method to be used in the final solution. Afterward, we give the conclusions of the entire work. Finally, we discuss future work that can be done to improve the results of the whole process.

5.1 The data

As mentioned in the past chapter 4 Methodology, source data was created from an agave plantation located in Guadalajara, Jalisco. The area contains agaves of different sizes related to their ages, other plants like trees, and some foreign objects, a sample map is shown in figure 5.1. The whole plantation was divided into five zones delimited by roads and other kinds of plants. For each zone, the drone passed for about 200 waypoints capturing six images on each point (one image for each band) and flying at an altitude of about 30 meters to have a ground sampling distance (GSD), a measure of resolution that represents the distance between two-pixel centers based on flight height and camera specifications [33, 43], of 1cm/pix such that each pixel represents 1cm^2 of real area.

In order to create orthomosaic maps we used software that performs the necessary radiometric and geometric corrections removing distortions created by camera sensors and lenses, those properties related to camera characteristics, and those related to the environment like light intensity caused by the sun. This pre-processing of images allows us to create an orthomosaic map with no distortion having all bands aligned [28, 54, 24]. In this work, we used the PIX4Dfields software to generate those maps since it already supports DJI multispectral cameras. The entire map can be seen in figure 5.2, the background is green in color and the agave plantation stands out from the map in a brown tone due to the soil.

Finally, we divided the map into five maps of different sizes that go from 9529x7366 to 15797x12897 pixels providing a high-quality resolution that captures more details. Maps were cropped such that information of the zone captured is inside the image avoiding duplicated information of other zones. The software provides two resources of information, one for an RBG map for easy visualization and a tiff file containing raw information of five bands. This

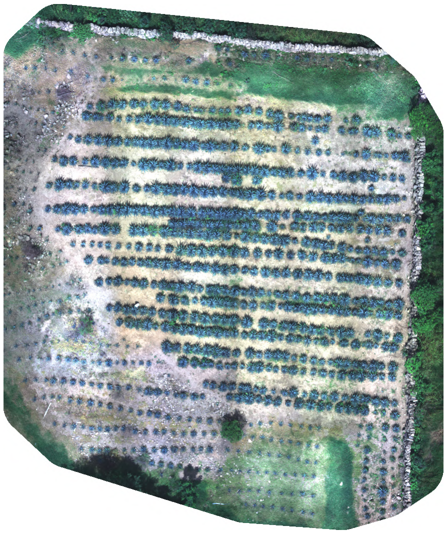


Figure 5.1: Sample image of an agave crop

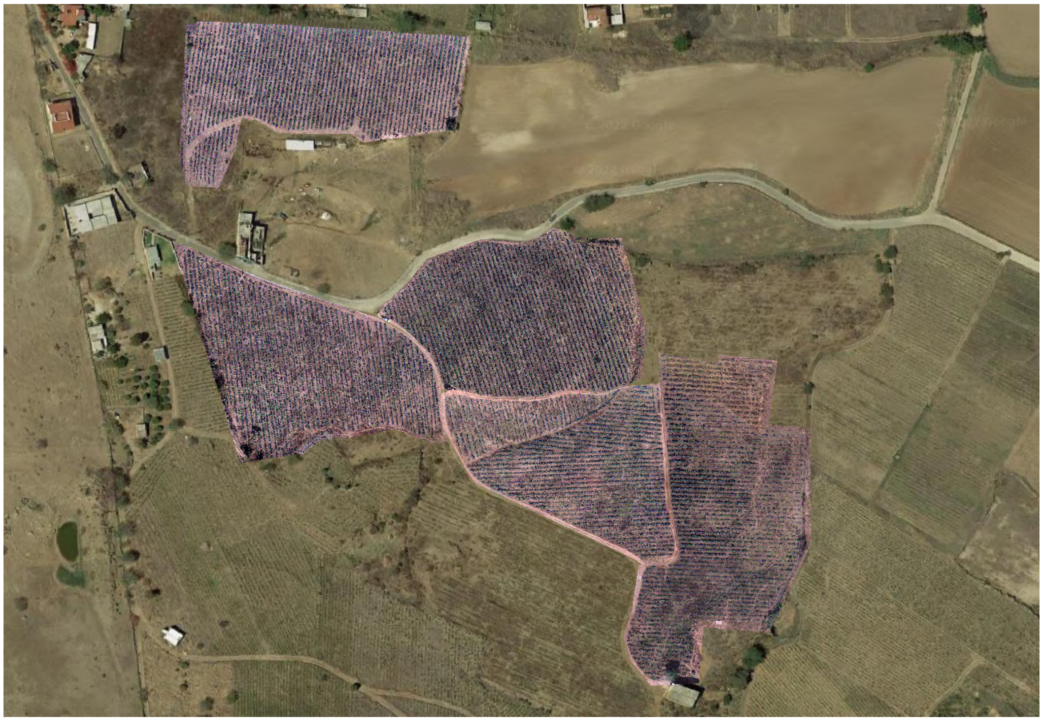


Figure 5.2: Aerial view of the entire plantation.

Table 5.1: Information of the five ares captured on the crop. There are information about the size on pixels of the orthomosaic map, the area covered, the name, and the GSD

Name	Size (px)	Area(ha)	GSD (cm/px)
Zone 1 (fig. 5.3)	12,717 x 8,078	1.24	1.28
Zone 2 (fig. 5.4)	15,797 x 12,897	1.23	1.05
Zone 3 (fig. 5.5)	9,529 x 7,366	1.02	1.49
Zone 4 (fig. 5.6)	7,221 x 12,926	1.48	1.57
Zone 5 (fig. 5.7)	11,685 x 6,626	1.16	1.51

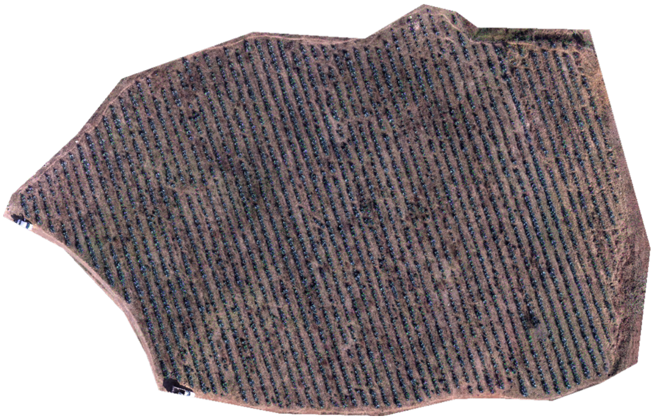


Figure 5.3: Zone 1 orthomosaic map

last information is what we will be using in this work to train the model while RGB images are going to be used to visualize the result of segmented agaves.

In table 5.1 we can observe characteristics of the five zones captured by the drone like the area covered, the size of the orthomosaic map, and the GSD. Combining all maps we are covering an area of almost 7 ha. As we can note, all zones have a GSD close to 1cm/px, variation was caused by soil altitude change on the field and obstacles like trees that force drones to fly higher. However, we could maintain spatial resolution in all five zones.

From the entire map (see figure 5.2) it is clear the division we make. The roads on the crop help us to define the zones. We created fly plans with the DJI application for each zone, a flight plan is a configuration and path the drone follows to take photos to cover the entire area, this allows us to re-capture the zone in the future and compare data, on how plants change over time. We can view each zone individually in figures 5.3, 5.4, 5.5, 5.6, and 5.7, for zones 1, 2, 3, 4, and 5 respectively. The next process, which is explained immediately, is done for each zone.

Once orthomosaic maps are created, it is time to add more information with vegetation indices and construct ground truth maps based on that. Vegetation indices are treated as another band on the tiff file so, adding three VIs on the map generates an image of eight bands. As mentioned previously, we are following the procedure of Sa, I et.al [47] to create the dataset, using smaller images from the entire map to maintain image quality and detail. Size of tiles is a variable that can change the final results and is indeed used on experiments to record the impact of information provided to algorithms. For ground truth creation we defined

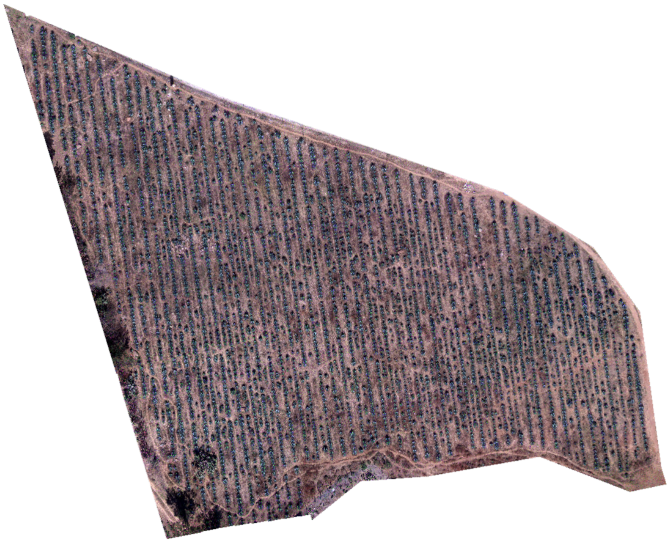


Figure 5.4: Zone 2 orthomosaic map

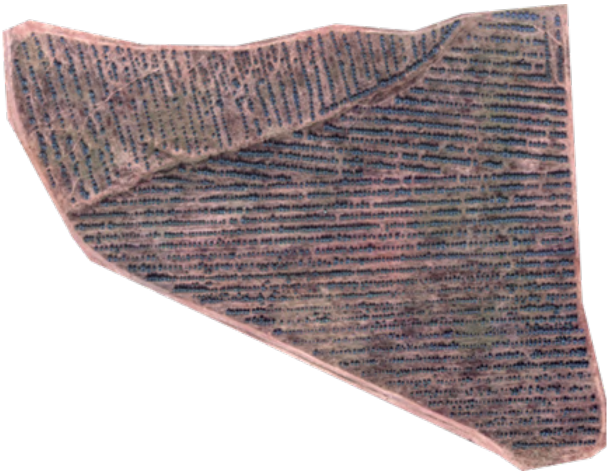


Figure 5.5: Zone 3 orthomosaic map

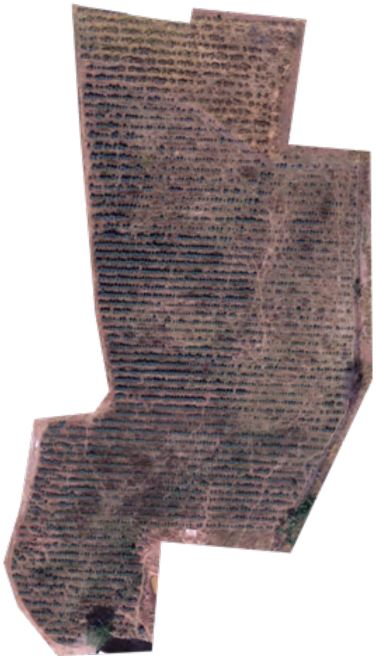


Figure 5.6: Zone 4 orthomosaic map

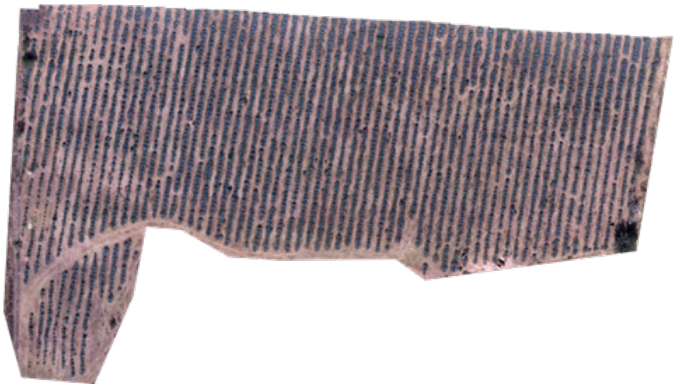


Figure 5.7: Zone 5 orthomosaic map

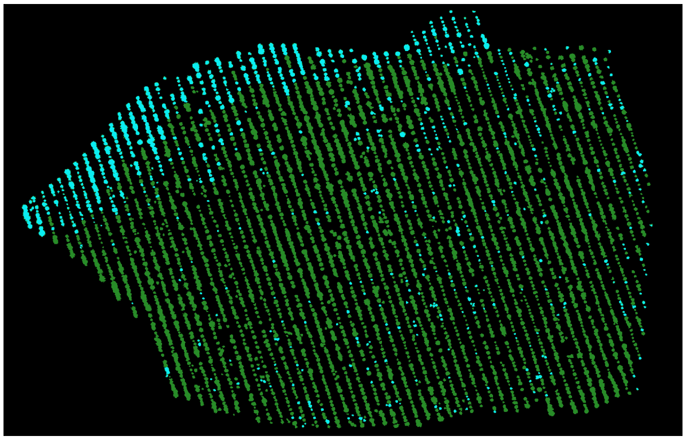


Figure 5.8: Ground truth map for Zone 1. Map contain three colors, blue, green, and black for agave warning, agave normal, and background classes

three classes, agave with a normal health, agave with a kind of problem, and background or zones with no agave where soil, trees and other kind of plants are included. Agave class was selected by the NDVI value since high positive values represent a healthy plant and positive small numbers represent soil and plants with problems [18, 38]. In order to create the ground truth map, we first labeled individual agaves separating them from other objects like trees, other plants and the soil. Once all agaves were labeled with an arbitrary class, we changed to the actual class by averaging the NDVI value on the agave, then a threshold is used to select the final class. We can define the process as follows:

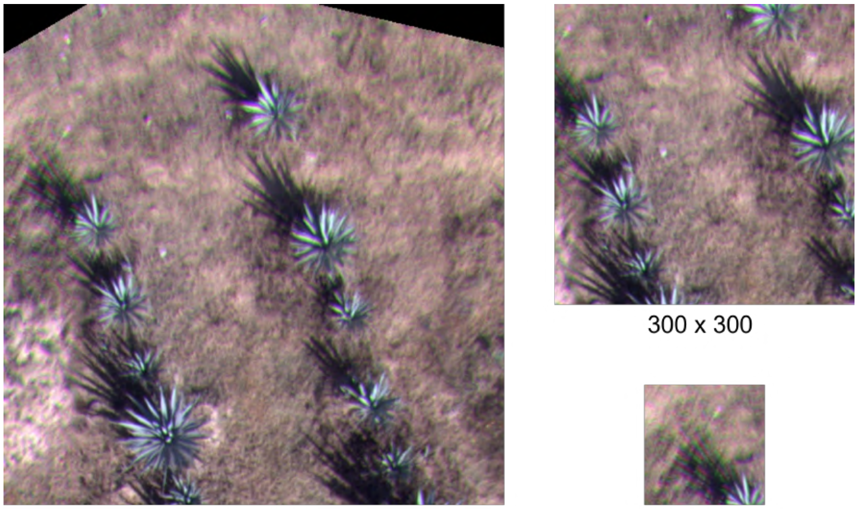
$$\text{For each agave: } class = \begin{cases} val < 0.5, & \text{agave warning} \\ val >= 0.5, & \text{agave normal} \end{cases}, \text{ where } val = average(\text{agave}) \tag{5.1}$$

In figure 5.8 we can observe an example of the ground truth map for Zone 1. There are three colors in the map for each of the classes, blue for agave warning, green for agave normal, and black for the background.

As discussed before, we have size tile as a parameter to change on experiments. We have chosen three different sizes for square images, 500x500, 300x300 and 120x120. As we can note, we have created three datasets containing the same information but in a different format. A large size correspond to a small dataset while smaller size correspond to a large dataset. By varying dataset size we expect to get better results on large dataset since we have more instances to train models, however, this also implies give less information, where models can not see the entire context of what they are learning, i.e. images contain a little amount of area as we can see in Figure 5.9 that shows the covered area for each size, we can note how area is reduced on small size.

In order to create the final datasets, depending on the tile size, and to avoid using images without data we follow the next procedure: For each individual band:

- Get the maximum value on the tile
- If the value is negative the image is not added in the dataset



300 x 300

500 x 500

120 x 120

Figure 5.9: A comparison of different sizes of tiles on the same map

- It the maximum value is positive the image is added on the dataset
- Repear for the next tile

The orthomosaic maps are rectangle images that contain data inside, however, as we could see in the images of the zones and the ground truth, there is space with no data since zones have variant shapes. In the TIFF file, to represent there is no data in a pixel, a value of -1000 is assigned on each band, thus we can use this to filter tiles and use just those that have data. Since the values captured for each band are positive, and RGB has values from 0 to 255 on each band, we can say that if an image has at least one pixel with a positive value, there is valid data on the image. After the filter of data, we can now create the dataset. The directory is defined as follows:

```
Dataset Size
 |__Data
 |   |__RGB
 |   |__Red
 |   |__Green
 |   |__Blue
 |   |__NIR
 |   |__Red Edge
 |   |__NDVI
 |   |__GNDVI
 |   |__NDRE
 |__Label
 |__Splits
```

Each dataset follows the same format as described before. In the ”Data” folder, there is a directory for each band including RGB and Vegetation Indices where each of which contains all the tiles created by the past process. There is also a ”Label” directory that contains the ground truth in tile format that matches with information of bands, in addition, there is a

”Splits” directory that contains the name of the tiles that correspond to the training and test sets. Names of tiles follow the next structure: ”xxxx.tif” for bands, ”xxxx.png” for the label, and ”xxxx.jpg” for RGB where x are numbers. Then, the name of images is defined with four digits numbers, and extension changes depending on the type of data. As we mentioned before, we have defined three tile sizes for datasets, in table 5.2 we can find the number of tiles or images created by the tile size, thus the size of the dataset depends on the size of the tiles.

Table 5.2: Size of the three datasets used to train models

Dataset	**Tile size (px)**	**Number of Tiles**
Agaves 500	500 x 500	1541
Agaves 300	300 x 300	4116
Agaves 120	120 x 120	24710

5.2 The comparison of algorithms

In this section, we are analyzing the results of models and compare the scores among all machine and deep learning models. We start by analyzing scores of machine learning models, also providing the ROC curve of the best models and the data they use to get those results. Then, we continue with deep learning models. In this part, we compare the models using different sizes of the images as well as the combinations of bands. Finally, we are comparing machine learning versus deep learning models to see what the differences are in the final predictions.

5.2.1 Machine Learning Models

We started by training the machine learning models since it is easier and we do not need to run more than one experiment on each one. As we discussed in the previous chapter we first need to change the format of the images into instances, features and labels. However, because limitations of models and the toolkit used to train, we are using just half of the data by sampling every two pixels. We can think of resizing the images by half. In addition, since we are using pixels as the instances we do not need to use the different sizes of the dataset. This is due to pixel quantity being the same for all datasets, the difference is given just by the area captured (see figure 5.9).

Even with this transformation, the total instances on the dataset are 27,720,000. As mentioned before, we have available eight different bands to train models. We decided to perform many experiments using a different combination of the bands. Combinations include: 1.- All the eight bands. 2.- Three Vegetation Indices. 3.- Five multispectral bands. Therefore, the number of bands is the number of features the algorithms will use in the training step. We use the training set on the ”Splits” directory to train models and the test set to report the results

Table 5.3 shows the overall scores of machine learning models for all the combinations of bands including all bands, vegetation indices, and original bands on the camera. We can

Table 5.3: Overall scores of machine learning models

Overall scores			
Algorithm	aAcc	mAcc	mIoU
KNN	85.35	**61.62**	**51.02**
KNN - VI	79.15	48.46	39.12
KNN - 5Bands	84.31	59.99	49.13
RF	86.91	53.86	44.83
RF- VI	87.72	58.84	48.10
RF- 5Bands	**88.06**	59.70	49.45
DT	84.10	58.47	46.84
DT- VI	82.90	56.01	44.80
DT- 5Bands	83.37	56.98	45.60

observe different behavior of models depending on the number of bands or the information available to train. We start by using all channels, using this data we expect to have the best scores since it contains more information. Random forest obtained the highest score on accuracy reaching 87.72% followed by KNN with 85.35%, about 2% less than RF, and Decision Tree with 84.10%. This accuracy does not consider individual classes at all, it is the accuracy of the whole prediction. We can use the mean Accuracy (mAcc) to have an overview of the class accuracy to get a more realistic score. In this case, the higher mAcc was reached by KNN with 61.62% followed by a decision tree with 58.47%, and then random forest with 53.86%. On the other hand, we also are reporting the Intersection over Union score, a value that tells how well the segmented area was covered. The score is reported as the mean value for all the classes, KNN obtained again the highest score with 51.02 %, followed by decision tree with 46.84%, and random forest with 44.83%. As we can note, even when random forest obtained the best aAcc, when we look at the mean accuracy of classes, KNN algorithm is better.

For the second training round, we use just three vegetation indices including NDVI, NDRE, and GNDVI. This time models just have three attributes to be trained. We compare again the same metrics over all the models. Again, this time random forest has the best aAcc with 87.72%, followed by decision tree with 82.90%, and KNN with 79.15%. We have similar behavior on mAcc where the random forest has the best score with 58.84%. For the intersection over union score, the random forest also reached the best score at 48.10%, and KNN obtained the lowest score at 39.12%. In this case, random forest obtained the best scores on both aAcc and mAcc.

The last round of training was using the five original bands captured by the drone. If we can obtain better scores using this set of data, we would be able to use raw data without the necessity to compute vegetation indices reducing the time to get the final prediction. Random forest algorithm again obtained the best aAcc score with 88.06%. For mAcc, KNN obtained the best score with 59.99% for just 0.29 points. Comparing mIoU random forest was better than KNN with a score of 49.45% just above 0.32. As we can note, the random forest was better again on almost all the scores but there is not much difference between RF and KNN.

Models change the score based on the number of bands they use to train to go from 75% accuracy to over 85%. However, when looking for mAcc some of the scores go down so the

Table 5.4: Per class scores for Machine Learning models

Per class Accuracy and IoU						
Algorithm	Background		Agave normal		Agave warning	
	Acc	IoU	Acc	IoU	Acc	IoU
KNN	97.76	90.81	58.56	39.67	28.54	**22.59**
KNN - VI	95.34	82.14	38.10	25.36	11.93	9.85
KNN - 5Bands	96.98	89.52	59.36	39.29	23.63	18.57
RF	98.83	89.78	60.60	42.61	2.16	2.11
RF- VI	97.76	91.51	**72.65**	**47.15**	6.11	5.65
RF- 5Bands	**98.04**	**92.02**	70.71	**47.15**	10.35	9.19
DT	94.05	89.06	48.79	32.77	**32.57**	18.68
DT- VI	93.31	87.40	46.38	30.87	28.35	16.13
DT- 5Bands	93.60	88.08	47.34	31.60	30.01	17.11

accuracy for some classes is worst than others. For this reason, we also compute the score for each of the classes in the dataset.

Table 5.4 shows the per class accuracy and IoU of models using the combinations of bands we discussed before. Comparing the scores for all models we can observe scores reached by the VI data were the lowest on most of the models and classes in the dataset. As we can see, there are three classes we are computing the scores, background, agave normal, and agave warning. In addition, we can note that the score given by IoU is less than those given by the accuracy. For the case of the Background class, both scores Acc and IoU drop to the minimum when just using Vegetation Indices data, with 93.31% and 82.14% accuracy and IoU respectively. However, the background class has higher scores compared to the other two classes. These high values rely on the number of background labels present on the ground truth maps, most of the area is covered by this class.

On the other hand, agave normal and agave warning had lower scores dropping to about 2% accuracy and IoU. In addition, we can note how the scores are low for the Agave warning class. Looking at the agave normal class, the highest accuracy is given by random forest using VI data reaching 72.65% accuracy and 47.15% IoU. However, when we look at the score for agave warning of the same model, the score drops to about 6%. Overall, we can say that all models can segment just one kind of agave decently by the random forest model but it is not able to segment agaves with a kind of problem. Some models detect better agave normal class while reducing the detection of the agave warning class.

As we could see, it is not sufficient to just look at the overall scores to select the best model but also look at individual scores for each of the classes. For instance, overall scores tell us KNN with all bands and random forest using five bands are the better models for this application, however, the random forest was not able to distinguish the agave warning class. Since we want to have a balance of the classes, we would choose a model that can predict all of the classes. In this case, KNN algorithm has better scores for the three classes, but, on the other hand, the decision tree also has good results on the individual scores even when was the model with the lowest overall scores. In one of the next sections, we are comparing the final predictions of the machine and deep learning models as well as examples of the output images.

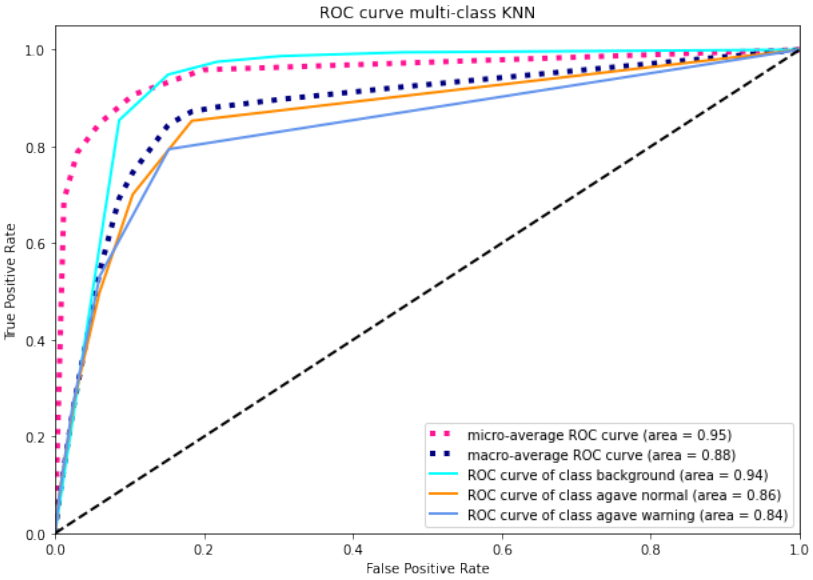


Figure 5.10: ROC Curve of KNN algorithm

Finally, we compute the ROC curve using true positive and false positive rates of the best configuration of each of the models. Data selected for each model are: 1.- KNN: All bands. 2.- Random Forest: Five original bands. 3.- Decision Tree: All bands. Data was selected by the highest overall score as well as the more distributed best individual scores. For instance, KNN model has similar scores with using all bands and using just five original bands, in addition, the score given for the agave normal class are also very close, however, there is a difference for the agave warning class being the model that uses all bands the one that got the highest score.

All plots have the same symbology so we can compare them correctly. Light blue represents the background class, orange represents the agave normal class, and blue represents the agave warning class. On the other hand, the pink squares line is the micro-average score, and the dark blue squares line is the macro-average score. In this case, the micro-average line is the most important score since it is generated by taking into account each of the classes so each of them has a weight depending on the amount of label while the macro-average generates the score independently of the classes in the dataset. In addition, a micro-average score is used commonly when there are more than two classes or classes are imbalanced.

We started with KNN algorithm, we can see the ROC Curve in figure 5.10. As we can note, the area covered by the background class is higher than the others, similar to accuracy and IoU scores. Both classes, agave normal and agave warning cover a similar area of about 0.86 but the performance is not similar since agave warning is not detected very well by the model. We can also observe how the score of the micro-average is higher than the macro-average. The second model was the random forest using the five original bands. As we can see in figure 5.11 the ROC curve looks better and the area is above 0.90 for all the classes, and micro and macro averages. We can note a similar behavior of the scores of the classes, background covers most areas, followed by agave normal class and finally agave warning class. In addition, we can see the area of the micro-average score for random forest algorithm

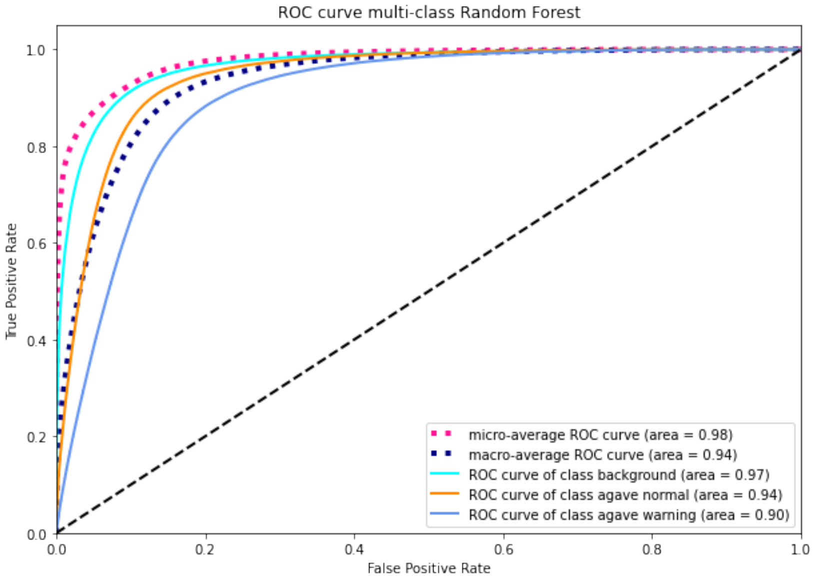


Figure 5.11: ROC Curve of random forest algorithm

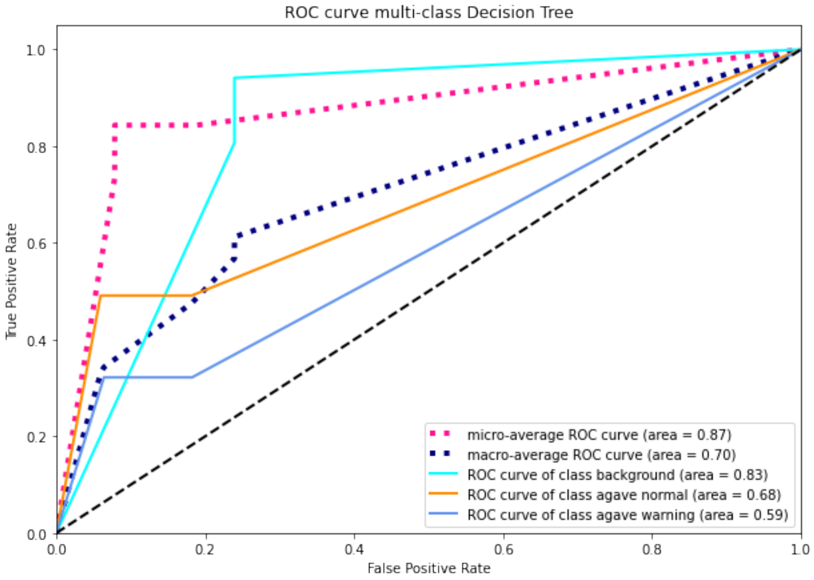


Figure 5.12: ROC Curve of decision tree algorithm

is greater than the area of KNN. The last model is decision tree whose ROC curve can be seen in figure 5.12. It is clear that this is the worst model since are drops over 0.6 and lines are close to the limit black line. Behavior of classes is the same, with background class as the best and agave warning as the worst.

Comparing the three graphs along with the scores provided in the tables we can conclude random forest classifier is the best machine-learning model of the three when using just five bands or features. However, models are just able to identify one of the classes. After analysis of deep learning models, we are showing the predictions of the models.

5.2.2 Deep Learning Models

We perform several experiments for deep learning models, first using different sizes of tiles and then varying the bands used. In order to train models, we decided to use mmsegmentation repository, a toolbox based on PyTorch to easily implement deep learning models for semantic segmentation [11]. Models were trained following an iteration-based runner approach instead of epoch based since it is faster and recommended by developers of mmsegmentation.

Table 5.5: Deep Learning models configuration

Model	**Data Augmentation**	**Optimizer**	**Learning Rate**	**Iterations**
FCN	Resize: ratio(0.5, 2.0) Random Crop 0.75 Random Flip 0.5	SGD	0.01	30,000
U-Net	Resize: ratio(0.5, 2.0) Random Crop 0.75 Random Flip 0.5	SGD	0.01	30,000
ViT Segmenter	Resize Random Crop Random Flip	SGD	0.001	30,000

Even though we train models using the iteration approach, we have selected iterations to train for 10 epochs. As a principal score, we are considering accuracy since it is most commonly used in others papers, however, we are also reporting mean accuracy, individual accuracy for each class as well as IoU scores similar we did with machine learning models. As discussed in the methodology chapter, we have three datasets of different tile sizes and we have also three different combinations of bands generating nine training sets for each of the models. We have divided the results into three groups in order to have the same format as machine learning, thus, each group is separated by the tile size or the dataset.

We begin with a 500x500 tile size dataset to analyze the scores given by the models similar to ML models where we analyze first overall scores and then individual scores. Table 5.6 shows the overall scores of selected deep learning models. At first view, we can note a better performance against classical ML algorithms since the scores are close 90% accuracy and 65% IoU. In addition, there is not much difference between the variant of models, for instance, all ViT models have similar values for the three scores, aAcc is close to 92%, mAcc is about 80% and mIoU is close to 69%. The accuracy of most of the models are higher than

Table 5.6: Overall scores using 500x500 tile size

Overall scores size 500			
Algorithm	aAcc	mAcc	mIoU
ViT	**92.96**	78.93	68.78
ViT - VI	92.68	**80.81**	**69.05**
ViT - 5Bands	92.98	79.28	68.88
Unet	92.54	75.16	64.62
Unet - VI	88.46	51.31	51.31
Unet - 5Bands	92.41	75.31	63.95
FCN	85.74	48.25	40.54
FCN - VI	89.08	77.79	62.29
FCN - 5Bands	88.63	68.21	52.66

92% and IoU is higher than 60%, thus, in this case, it is most important to consider individual scores. Furthermore, it seems that FCN is the worst of the models since none of the models reach 90% accuracy.

Table 5.7 shows individual scores of deep learning models using 500x500 tile size. In this table, we can observe differences between models and variants of them in the scores given for agave normal and agave warning classes. ViT model is the most stable model since scores on all classes are very similar and they remain in a very close range. The unet, model that has 92% accuracy on all the variants, got low scores on the agave warning class but it also got the best scores for the agave normal class reaching almost 90% accuracy. On the other hand, FCN model that had one of the worst scores among the models got the best accuracy on the agave warning class reaching 82%. However, both models got very poor performance in the other class.

Table 5.7: Per class scores with 500x500 tile size

Per class Accuracy and IoU size 500						
Algorithm	Background		Agave normal		Agave warning	
	Acc	IoU	Acc	IoU	Acc	IoU
ViT	98.3	95.45	72.93	61.23	65.55	49.66
ViT - VI	97.27	95.04	73.59	61.01	71.57	**51.11**
ViT - 5Bands	98.02	95.42	77.98	**62.72**	61.83	48.51
Unet	98.51	96.00	84.68	61.16	42.29	36.69
Unet - VI	96.8	94.46	78.71	46.05	16.68	13.44
Unet - 5Bands	98.12	**96.03**	**89.17**	61.26	38.63	34.55
FCN	**99.65**	89.3	41.21	28.45	3.88	3.86
FCN - VI	93.13	90.44	79.4	54.29	60.83	42.14
FCN - 5Bands	97.68	95.24	24.56	23.29	**82.39**	39.44

Using this table we could identify the best combination of bands for each of the models using a tile size of 500x500. Even when FCN did not get more than 90% accuracy, the use of vegetation indices could help the model to have a good distribution of segmentation of the classes. Models selected were the following: 1.- ViT using vegetation indices bands. 2.- Unet using all bands. 3.- FCN using vegetation indices bands. As we can note, two of the models use vegetation indices bands, just three bands were necessary to segment the agaves, but they have information on the five bands captured by the drone. We can say that deep learning models are better than machine learning models since they can identify better the three classes without scarifying the performance of the others.

Table 5.8: Per class scores with 300x300 tile size

Overall scores size 300			
Algorithm	aAcc	mAcc	mIoU
ViT	**90.99**	**73.09**	**62.19**
ViT - VI	89.69	69.61	57.56
ViT - 5Bands	90.30	72.54	58.43
Unet	84.48	61.62	42.35
Unet - VI	82.67	61.42	41.13
Unet - 5Bands	71.75	47.11	30.67
FCN	90.24	65.54	53.50
FCN - VI	90.16	67.97	55.88
FCN - 5Bands	90.40	67.18	53.99

The second training round was using a tile size of 300x300 on the same combination of bands eight, five, and three bands to train all the models. The overall scores for this training dataset can be seen in table 5.8. In this dataset we expected to have an increase in the scores since there are more instances to train the models, however, contrary to the expected result there was a little drop in the majority of models. We can see how accuracy maintains at 90% but IoU is under 60%. Furthermore, FCN has similar values for all the combinations of bands, contrary to Unet where accuracy is between 71% and 84% being the worst model in this dataset. ViT trained with all bands was the best model in this round of training on overall scores.

Table 5.9: Per class scores with 300x300 tile size

Per class Accuracy and IoU size 300						
Algorithm	Background		Agave normal		Agave warning	
	Acc	IoU	Acc	IoU	Acc	IoU
ViT	97.94	94.49	64.83	52.5	56.49	39.56
ViT - VI	98.06	94.03	41.26	37.83	69.51	40.83
ViT - 5Bands	98.05	95.43	35.73	34.50	83.83	**45.35**
Unet	94.88	93.59	0.45	0.45	89.52	33.02
Unet - VI	92.45	91.55	0.74	0.71	**91.05**	31.13
Unet - 5Bands	82.39	74.18	0.0	0.0	58.95	17.83
FCN	**98.89**	95.09	85.78	53.79	11.97	11.61
FCN - VI	98.02	94.87	83.27	52.54	22.63	20.24
FCN - 5Bands	98.39	**95.67**	**90.92**	**54.38**	12.22	11.91

As we did in the previous analysis, we continue with scores given by each of the classes. Table 5.9 shows the per class scores trained with 300 tile size. In this table, we can observe unique behavior on the segmentation of classes like the case of the unet with five bands and VI where the model was not able to identify the agave normal class but the accuracy on the agave warning class increases until reached 91% accuracy. This high score in this class and the low value on IoU tell us that the model segments all agaves as the same class, thus, the model works as a two-class segmentation. We can see the same results but with the other class on the FCN model. In this case agave, the normal class reached 90% accuracy with 54% IoU but 12% accuracy on agave warning class. ViT was the unique model that could segments all classes properly but not too well as models in the past dataset. In addition, the model with all bands was the best in this case followed by VI bands but got a good distribution of class detection. There was a change in the scores of all models in a negative way where models were just able to detect one of the two classes of agaves and just two versions of ViT could identify them decently.

Table 5.10: Per class scores with 120x120 tile size

Overall scores size 120			
Algorithm	aAcc	mAcc	mIoU
ViT	**90.73**	**74.24**	**63.13**
ViT - VI	89.34	67.97	55.85
Unet	83.77	59.74	45.29
Unet - VI	86.77	57.42	48.00
FCN	87.90	65.18	53.60

For the last training session, we use the 120x120 tile size, the bigger dataset since the images are very small compared to the rest. Since the training time increases by the name of instances in the dataset, we decided to train models with two variants of bands and one of FCN due to the large amount of time necessary to train this model. Table 5.10 shows the overall scores of this training round. As we can note, the table is smaller since with did not include the five bands version. The scores are even lower than the other tile sizes, dropping for about 86%similar to machine learning models. ViT model obtained again the best scores among the others. The values of scores, the training time, and the low variation of scores with different combinations of bands were the reason we did not continue with the training of the rest of the variation of models.

Finally, table 5.11 shows the per class scores with the 120x120 dataset. Scores are very similar to those obtained in the last training round having models that can just identify one class of agave but with high accuracy such as the case of the Unet. Moreover, the accuracy of Unet is close to 50% for both classes of agave. The most interesting model was ViT, a model that has obtained good results in all the datasets, obtaining IoU scores close to 50% and accuracy in the range of 59% and 65% in the agave classes.

As we could observe in the analysis of results using a different dataset, changing the size of the image, a more small size creates lower scores for all the models. One of the reasons that can produce this situation is the size of the image since models need the information that is around the objects to create a good segmentation and as we could see in the methodology section, the covered area changes depending on the tile size meaning less covered are for

Table 5.11: Per class scores with 120x120 tile size

Per class Accuracy and IoU size 120						
Algorithm	Background		Agave normal		Agave warning	
	Acc	IoU	Acc	IoU	Acc	IoU
ViT	97.48	93.77	65.95	**53.65**	59.31	**41.96**
ViT - VI	97.32	93.06	**83.26**	53.47	23.32	21.03
Unet	94.44	92.47	18.76	14.11	**66.02**	29.30
Unet - VI	**98.24**	90.64	61.31	42.86	12.71	10.51
FCN	97.25	**94.55**	51.69	39.98	46.61	26.87

small size. We are continuing with the comparison of algorithms using just models that were trained with 500x500 tile size since they obtained the best scores. We are also considering machine learning models selected in the comparison.

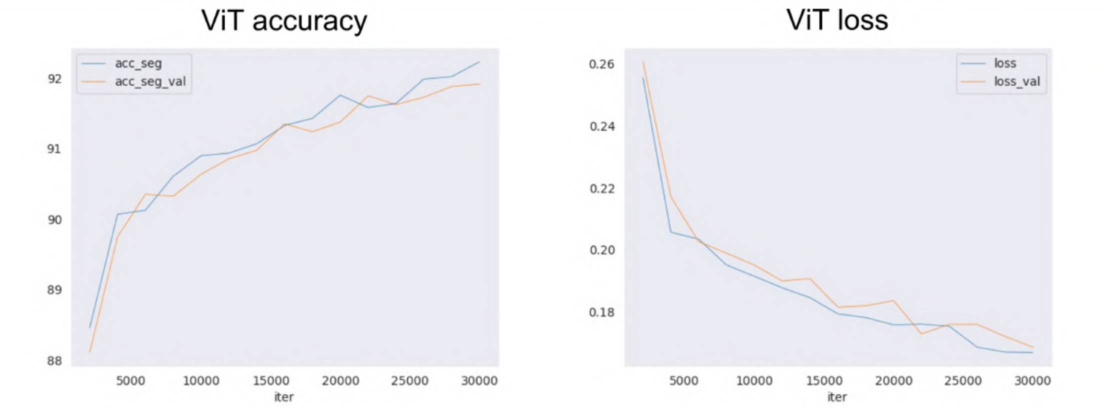


Figure 5.13: Accuracy and loss curves of ViT over iterations

Then, we continue by visualizing the loss and accuracy curves of models to see if there is over-fitting in the models or if there is a change to continue training the model to get better segmentation. It is important to note we are only analyzing curves of the best algorithms we mentioned in the 500-size dataset. We start with ViT model using vegetation indices bands. Figure 5.13 shows the accuracy and loss curves of the model over the 30,000 iterations in the training step where we can see the values on the training data (blue line) and the validation data (orange data). On the accuracy graph we can observe the two lines, training and validation data, are very close to each other and they are still going up in the score. Meanwhile, the loss on training and validation sets go down over all the iterations, and they are very close to them similar to the accuracy curve. We do not see any behavior of overfitting or underfitting in the model and the trend tell us the model can still be trained for getting a higher accuracy and lower loss.

Figure 5.14 shows the accuracy and loss graphs of the Unet model over the 30,000 iteration model that was trained. In this case, we can see a difference in the training and validation data, while training accuracy grows quickly, accuracy on the validation set barely grows. This condition repeats on the loss where the value on the training set reduces while the

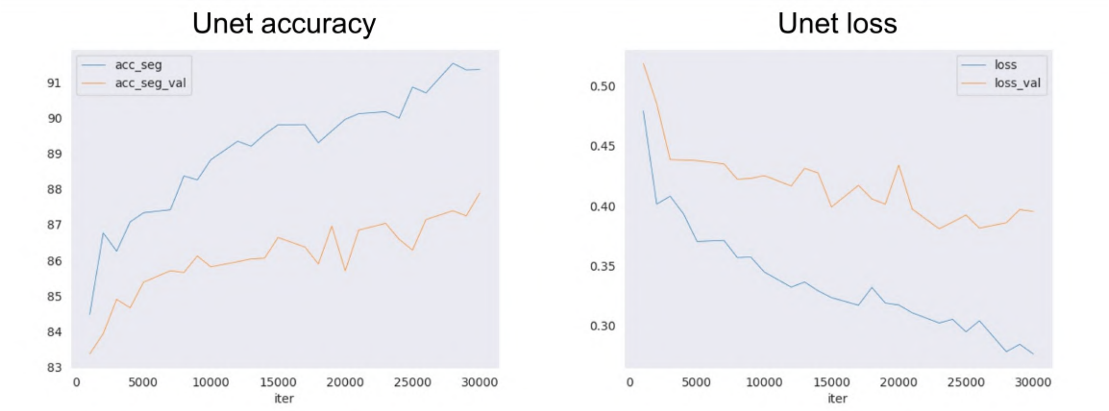


Figure 5.14: Accuracy and loss curves of Unet model over iterations

loss in the validation set has a low slope. This may cause overfitting on the training dataset since the scores do not change on the validation dataset.

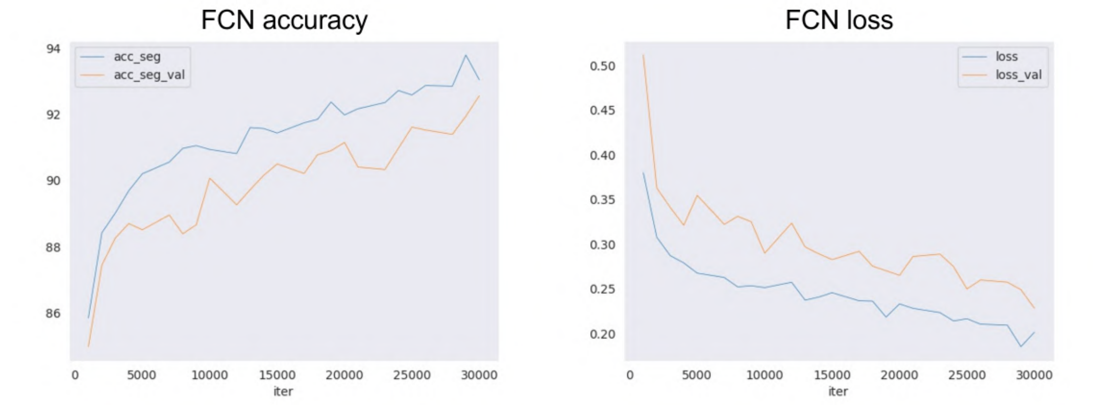


Figure 5.15: Accuracy and loss curves of FCN model over iterations

The last model was FCN which accuracy and loss curves are seen in figure 5.15. Graphs look similar to ViT model, but there is a gap between the training and validation sets. In addition, this also happens on the loss graph, the score on the validation and training sets goes up while maintaining a distance. Even when there is a difference in the scores, there is no overfitting in the model so we can continue training the model to get better results.

5.2.3 Models with RGB data

On the other hand, as we can note, models reached the highest score when using images of 500x500 size, thus, we are training models but now using RGB images to have a comparison and see if there is a difference when using multispectral images. We use the same initial configuration for the models, for instance, the optimizer is SGD, and the learning rate is 0.01 for deep learning models.

Table 5.12 shows the overall scores of machine learning and deep learning models using RGB data, images that can be captured by any common camera. As per previous results, deep

Table 5.12: Overall scores using RGB data

Overall scores size 500 RGB			
Algorithm	aAcc	mAcc	mIoU
ViT	89.22	71.00	58.69
Unet	**90.93**	74.24	61.43
FCN	90.81	**75.98**	**62.92**
KNN	84.56	56.80	44.85
RF	87.02	57.07	45.28
DT	87.15	57.48	47.96

learning models are superior to machine learning models on all scores whereas DL can reach 90% accuracy and 62% mIoU, ML just can produce about 87% accuracy and 47% mIoU. Moreover, the performance of models is worst than those given with Multispectral data such as ViT with vegetation indices. In this case, FCN and Unet were the best models in accuracy and mIoU. On the other hand, FCN got a slightly better result compared to FCN with VI data. As we can note there is a difference in scores when using RGB and Multispectral images on models.

Table 5.13: Per class scores using RGB data

Per class Accuracy and IoU size 500 RGB						
Algorithm	Background		Agave normal		Agave warning	
	Acc	IoU	Acc	IoU	Acc	IoU
ViT	96.51	92.24	53.35	45.3	63.14	38.53
Unet	97.56	**95.78**	59.18	48.75	**65.97**	39.77
FCN	96.56	94.25	66.59	**49.93**	64.8	**44.59**
KNN	**97.98**	90.71	52.73	26.55	19.68	17.27
RF	97.26	90.71	**73.49**	44.68	0.46	0.46
DT	97.86	91.41	59.97	40.81	14.60	11.68

We continue analyzing individual scores on each class for the six models trained with RGB data. Table 5.13 shows this information to take a look at how models segments each class. Starting with ML models (KNN, RF and DT) we can see how they can only segment one of the agave classes such as when using multispectral data. The best accuracy score for the agave normal class was reached by random forest model with 73.49% accuracy. On the side of DL models (ViT, Unet, and FCN), they got balanced scores on agave classes with accuracy around 60% and 48% IoU. Moreover, as we mentioned in the overall scores, values obtained by the models with this kind of data were lower than those obtained with multispectral images. Even when FCN model looked to be better in this configuration, individual scores showed that it was just a little inferior on agave normal class.

As we could see, there is a difference in the performance of models when using RGB or Multispectral images on the agave segmentation based on the health of the plant being models trained with bands on multispectral images superior for this specific task. Since multispectral bands provide information not available in RGB, models can enhance their performance.

5.2.4 Image Segmentation

In order to visualize the segmentation created by each model, we have selected those whose give the highest scores and best balance of all classes. Models and multispectral data selected are the following:

- **Decision Tree:** All bands
- **KNN:** All bands
- **Random Forest:** Five original bands
- **ViT:** Vegetation Indices
- **Unet:** All bands
- **FCN:** Vegetation Indices

Figure 5.16 shows the segmentation of the models of a random instance in the dataset. The image at the top of the figure is the ground truth label, i.e., the expected output segmentation. In this instance, we can find several agaves, a group of agaves in the centers that are very close, and an agave of the class ”agave warning”. On the left side, we can see segmentations of ML models while on the right DL models. For ML models, we can see classification is done at a pixel level, thus, we can observe points, like salt and pepper noise, over the entire image causing noise in the segmentation. For instance, the noise can be seen clearly in the Decision tree segmentation, where there are pixels classified as agaves. However, most pixels are grouped inside agaves even when two classes are mixed. KNN algorithm also shows this condition but noise is reduced as well as the mix of classes. Finally, Random Forest model, which obtained the best scores among ML models, generates a better segmentation, the noise is still present on the image but it is almost imperceptible compared to the others and the mix of classes is just found on the edge of the plants.

On the side of DL models, we have different kinds of segmentation starting with the noise that is not present in these models. We can see how the mix of classes inside the agave is different in these models since they create groups of classes. FCN, the model which obtained the worst performance compared to the rest of DL models, could segment agaves, however, we can see that the model was able to segment agaves without problems. In addition, we can see the mix of classes where inside the agave there are parts classified as another class. The Unet got a better result than FCN, all agaves are segmented but there is a little mixture of classes. Unet segmentation is very close to the ground truth label for agave segmentation and the single ”agave warning” is detected correctly. The last ViT also has a good performance on the segmentation similar to Unet but there is no mix of classes in this instance. On the other hand, the ”agave warning” class was not detected.

As we could see, segmentation is different depending on the type of model either ML or DL, and the model itself such as Random Forest or ViT. Even when there is noise in ML segmentation, it can easily be filtered to get just the correct area. On the other hand, DL models can create a better segmentation on agaves without the noise produced by ML techniques and segmented areas are very close to the ground truth label. Despite the good performance of DL techniques, there is still a combination of classes in a single agave. For this reason, we are implementing a watershed algorithm with some extra steps to correct the label and to separate groups of agaves in order to identify individual plants.

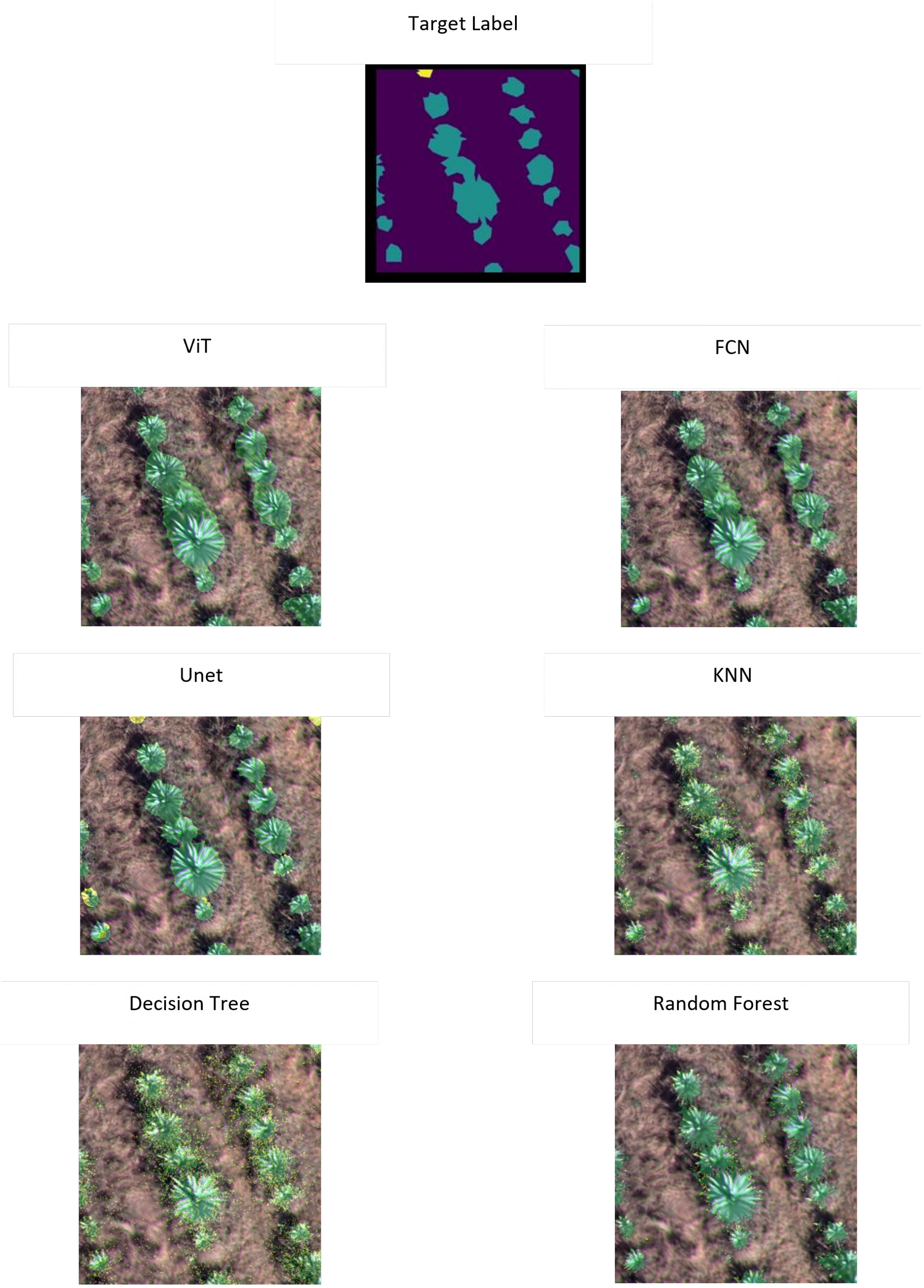


Figure 5.16: Predictions of the six models. The target label is showed in the top of the image

We have selected the prediction of the Unet since it has all the conditions that can be corrected by the algorithm (see Watershed algorithm section of the Methodology chapter for more details of the steps). Figure 5.17 shows the result of applying the watershed algorithm to the prediction of the Unet model. The left side of the figure is the correction done after the watershed algorithm and class selection. We can see how agaves that were grouped in the center of the image are now separated by the algorithm enclosed in a blue line. In addition, each agave has only one class representing the state of the plant. The purpose of the watershed algorithm is to separate objects when they are very close to each other and it is not possible to distinguish how many objects, or plants in this scenario, are in a cluster. Once agaves are separated such that each cluster represent a single plant, the center is used for computing the final class by taking the majority class of values around a defined radius. For instance, look at the prediction of Unet in Figure5.16, there is one agave in the bottom left side of the image that has been classified with two classes, after applying the algorithm we can see how the agave is classified with ”Agave Normal” class since it is present in the center of the plant. In addition, we can see how the segmentation is even closer to the ground truth label on the classified classes. With the use of this algorithm, we have separated agaves as well as eliminated the noise of mixed classes.

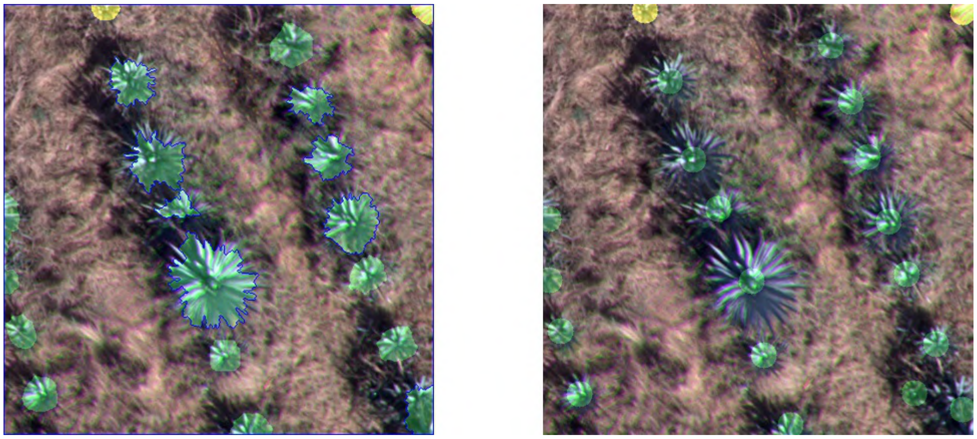


Figure 5.17: Prediction after applying Watershed algorithm. Left side is the final prediction. Right side is a representation where dots are the detected agaves

On the other hand, we have created an extra representation to show the results of segmentation that can be seen in the right side of figure 5.17. In order to create this representation we take the image created by the algorithm, then, the center is used for extracting the agave class. Then, a circle of a defined size is created in the center of each agave using agave class to use the color. As we can see in the image, circles are in the center of the image with the color of the class. This is just an extra representation that can be used for a different view on the segmentation that for instance it is easier for a user to count plants.

Chapter 6

Conclusion

The implementation of multispectral images has been explored for many applications such as in segmentation of trees, weed detection, and classification of areas in crops. They are commonly implemented along with drones, a tool that has been also used in agriculture since it is a useful tool to obtain data in a fast way. However, in the specific application in agaves, we have just a few implementations or solutions related to this plant including agave count. Thus, for this reason, an since is an important plant for Mexico, we decided to use agaves as the case study for multispectral images.

In this work, we have presented two main contributions related to the identification of agaves based on their health. The first one is the acquisition of multispectral data using a drone about agave crops containing information on two kinds of agaves defined as ”Agave Normal” and ”Agave Warning”. Captured images were processed to create orthomosaic maps that were used to create three datasets. Each dataset was created using orthomosaic maps and cropping small images of fixed sizes. Datasets contain a directory for each of the bands captured including Red, Green, Blue, Red Edge, Near Infrared, and RGB, in addition, there is a directory for ground truth labels. The second contribution is the development of deep learning models for the segmentation of the two classes of agaves obtaining good results in the predictions.

In the comparison of algorithms, we found that deep learning models were superior to machine learning in the segmentation of agaves by classes, however, both can be used to segment agaves from the rest of the objects in a map. In special ML models, the output of the prediction requires an extra step to eliminate the noise and create consistent clusters. The best ML algorithm was random forest using five bands of the dataset, the 5 bands captured by the drone, reaching 88.06% accuracy and 49.45% mIoU, but as mentioned before the model can just identify one of the classes, in this case, the ”Agave Normal” class.

For deep learning models, we perform several experiments to train models, varying the size of the image and the bands used by algorithms. On the experiments, we conclude that even when we have more instances to train models with a small size, the performance is negatively affected by having scores similar to Ml models. Then, we got the best results using the 500x500 size dataset reaching an accuracy of 92.96% by ViT model using Vegetation Indices bands, in addition, the model was also the most balanced for both agave classes. Furthermore, we also provide an additional step that fixes some conditions of the predictions like the mixing of classes in a single agave that also separates groups of agaves to have each plant separated from the others.

On the other hand, we also train models using RGB images to compare against multispectral images to see if there is an advantage to the use of more complex data. In the results, we could see that they got good results reaching 90% accuracy for most of the models and they could also segment the two agave classes. With these results, we could conclude that multispectral images can help models get better results on the segmentation of agaves.

6.1 Future Work

This work makes use of a custom dataset captured by a drone equipped with a multispectral camera in a crop located in Zapopan, Jalisco. Thus, since the crop contains certain characteristics, we are limited by the types of conditions of the data that can be extended to have more classes to identify. For instance, we can find agaves infected by pests, affected by a known disease, with different ages of difference, or even identify other plants that are inside the crop.

On the other hand, we selected just six models to be trained with the dataset, three for Machine Learning and three for Deep Learning. We know that several models can be trained for this task like Support Vector Machine, XGBoost, DMNet, and new DL models that are created every year. In that sense, we are saying that the dataset can be used to test many models for agave segmentation. Moreover, in the dataset, we provide three vegetation indices including NDVI, NDRE, and GNDVI, but since multispectral data is also in the dataset more vegetation indices can be computed.

Another work that can be done related to the use of multispectral images in agriculture is to measure the advantages of this kind of data. As we could see, there is an improvement when using more channels along with vegetation indices increasing accuracy on the segmentation of agaves based on health, thus it would be interesting if the application of this solution can reduce the cost of the maintenance of a crop.

Bibliography

[1] Athanasios T Balafoutis, Frits K Van Evert, and Spyros Fountas. Smart farming technology trends: Economic and environmental effects, labor impact, and adoption readiness. *Agronomy*, 10(5):743, 2020.

[2] A Bannari, D Morin, F Bonn, and AjRsr Huete. A review of vegetation indices. *Remote sensing reviews*, 13(1-2):95–120, 1995.

[3] Fred Baret and Gerard Guyot. Potentials and limits of vegetation indices for lai and apar assessment. *Remote sensing of environment*, 35(2-3):161–173, 1991.

[4] Saheba Bhatnagar, Laurence Gill, and Bidisha Ghosh. Drone image segmentation using machine and deep learning for mapping raised bog vegetation communities. *Remote Sensing*, 12(16):2602, 2020.

[5] Leo Breiman, Jerome H Friedman, Richard A Olshen, and Charles J Stone. *Classification and regression trees*. Routledge, 2017.

[6] Gabriela Calvario, Teresa E Alarcón, Oscar Dalmau, Basilio Sierra, and Carmen Hernandez. An agave counting methodology based on mathematical morphology and images acquired through unmanned aerial vehicles. *Sensors*, 20(21):6247, 2020.

[7] Sebastian Candiago, Fabio Remondino, Michaela De Giglio, Marco Dubbini, and Mario Gattelli. Evaluating multispectral images and vegetation indices for precision farming applications from uav images. *Remote sensing*, 7(4):4026–4047, 2015.

[8] Maria Casamitjana, Maria C Torres-Madroñero, Jaime Bernal-Riobo, and Diego Varga. Soil moisture analysis by means of multispectral images according to land use and spatial resolution on andosols in the colombian andes. *Applied Sciences*, 10(16):5540, 2020.

[9] R Ceja Ramírez, DR González Eguiarte, JA Ruiz Corral, LA Rendón Salcido, JG Flores Garnica, et al. Detection of restrictions on production of blue agave (agave tequilana weber var. blue) using remote sensing. *Terra Latinoamericana*, 35(3):259–268, 2017.

[10] MODET CLARKE. Estudio sobre la “aplicación industrial de la planta del agave sp”. `https://www.clarkemodet.com/news-posts/presentacion-del-estudio-sobre-la-aplicacion-industrial-de-la-plan` 2017.

[11] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. `https://github.com/open-mmlab/mmsegmentation`, 2020.

[12] Consejo Regulador del Tequila. 1er foro de discusión fitosanitaria en el cultivo del agave azul tequilero. `https://www.crt.org.mx/images/documentos/MEMORIA1erFORODEDISCUSIONFITOSANITARIA28FINAL29.pdf`, 2011.

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[14] H. Fang and S. Liang. Leaf area index models. In *Reference Module in Earth Systems and Environmental Sciences*. Elsevier, 2014.

[15] El Financiero. El tequila está en problemas. `https://www.elfinanciero.com.mx/empresas/el-tequila-esta-en-problemas/`, 2018.

[16] E Javier García-Herrera, S de J Méndez-Gallegos, and Daniel Talavera-Magaña. El género agave spp. en méxico: principales usos de importancia socioeconómica y agroecológica. *Revista Salud Pública y Nutrición*, 5:109–129, 2010.

[17] L3Harris Geospatial. Vegetation indices. `https://www.l3harrisgeospatial.com/docs/vegetationindices.html`.

[18] M Amparo Gilabert, José González-Piqueras, and Javier García-Haro. Acerca de los índices de vegetación. *Revista de teledetección*, 8(1):1–10, 1997.

[19] Patricia Girimonte and Javier García Fronti. El indice ndvi y la clasificación de áreas sembradas aprendizaje automático no supervisado "k-means".

[20] Anatoly Gitelson and Mark N Merzlyak. Quantitative estimation of chlorophyll-a using reflectance spectra: Experiments with autumn chestnut and maple leaves. *Journal of Photochemistry and Photobiology B: Biology*, 22(3):247–252, 1994.

[21] Anatoly A Gitelson, Yuri Gritz, and Mark N Merzlyak. Relationships between leaf chlorophyll content and spectral reflectance and algorithms for non-destructive chlorophyll assessment in higher plant leaves. *Journal of plant physiology*, 160(3):271–282, 2003.

[22] Yingxin Gu, Eric Hunt, Brian Wardlow, Jeffrey B Basara, Jesslyn F Brown, and James P Verdin. Evaluation of modis ndvi and ndwi for vegetation drought monitoring using oklahoma mesonet soil moisture data. *Geophysical Research Letters*, 35(22), 2008.

[23] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

[24] Ryan Hruska, Jessica Mitchell, Matthew Anderson, and Nancy F Glenn. Radiometric and geometric analysis of hyperspectral imagery acquired from an unmanned aerial vehicle. *Remote Sensing*, 4(9):2736–2752, 2012.

[25] Alfredo R Huete. A soil-adjusted vegetation index (savi). *Remote sensing of environment*, 25(3):295–309, 1988.

[26] RD Jackson, PN Slater, and PJ Pinter Jr. Discrimination of growth and water stress in wheat by various vegetation indices through clear and turbid atmospheres. *Remote sensing of environment*, 13(3):187–208, 1983.

[27] Yoram J Kaufman and Didier Tanre. Atmospherically resistant vegetation index (arvi) for eos-modis. *IEEE transactions on Geoscience and Remote Sensing*, 30(2):261–270, 1992.

[28] Joshua Kelcey and Arko Lucieer. Sensor correction of a 6-band multispectral imaging sensor for uav remote sensing. *Remote sensing*, 4(5):1462–1493, 2012.

[29] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM Computing Surveys (CSUR)*, 2021.

[30] Reinhard Klette. *Concise computer vision*. Springer, 2014.

[31] Kowligi R Krishna. *Precision farming: soil fertility and productivity aspects*. Apple Academic Press, 2019.

[32] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[33] Jon C Leachtenauer and Ronald G Driggers. *Surveillance and reconnaissance imaging systems: modeling and performance prediction*. Artech House, 2001.

[34] Shunlin Liang. *Comprehensive Remote Sensing*. Elsevier, 2017.

[35] Tianyang Lin, Yuxin Wang, Xiangyang Liu, and Xipeng Qiu. A survey of transformers. *arXiv preprint arXiv:2106.04554*, 2021.

[36] Yuzhen Lu and Sierra Young. A survey of public datasets for computer vision tasks in precision agriculture. *Computers and Electronics in Agriculture*, 178:105760, 2020.

[37] Stuart K McFeeters. The use of the normalized difference water index (ndwi) in the delineation of open water features. *International journal of remote sensing*, 17(7):1425–1432, 1996.

[38] Carlos Merg, Daniel Petri, Fernando Bodoira, Martín Nini, MJ Fernández Díez, Federico Schmindt, Rodolfo Montalva, Leonardo Guzmán, Karina Rodríguez, Fernando Blanco, et al. Mapas digitales regionales de lluvias, índice estandarizado de precipitación e índice verde. *Pilquen-Sección Agronomía*, (11):5, 2011.

[39] United Nations. Take action for the sustainable development goals. `https://www.un.org/sustainabledevelopment/sustainable-development-goals/`, 2020.

[40] Mahdi Maktab Dar Oghaz, Manzoor Razaak, Hamideh Kerdegari, Vasileios Argyriou, and Paolo Remagnino. Scene and environment monitoring using aerial imagery and deep learning. In *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 362–369. IEEE, 2019.

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

[42] Leif E Peterson. K-nearest neighbor. *Scholarpedia*, 4(2):1883, 2009.

[43] Pix4D. Ground sampling distance (gsd) in photogrammetry. `https://support.pix4d.com/hc/en-us/articles/202559809-Ground-sampling-distance-GSD-in-photogrammetry`.

[44] Jiaguo Qi, Abdelghani Chehbouni, Alfredo R Huete, Yann H Kerr, and Soroosh Sorooshian. A modified soil adjusted vegetation index. *Remote sensing of environment*, 48(2):119–126, 1994.

[45] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[46] JW Rouse, Rüdiger H Haas, John A Schell, Donald W Deering, et al. Monitoring vegetation systems in the great plains with erts. *NASA special publication*, 351(1974):309, 1974.

[47] Inkyu Sa, Marija Popović, Raghav Khanna, Zetao Chen, Philipp Lottes, Frank Liebisch, Juan Nieto, Cyrill Stachniss, Achim Walter, and Roland Siegwart. Weedmap: A large-scale semantic weed mapping framework using aerial multispectral imaging and deep neural network for precision farming. *Remote Sensing*, 10(9):1423, 2018.

[48] Halil Mertkan Sahin, Bruce Grieve, and Hujun Yin. Automatic multispectral image classification of plant virus from leaf samples. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 374–384. Springer, 2020.

[49] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(4):640–651, 2017.

[50] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[51] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7262–7272, 2021.

[52] Boletín UNAM-DGCS-045. México cuenta con 159 especies de agave; investigadores de la unam encontraron 4 nuevas. `https://www.dgcs.unam.mx/boletin/bdboletin/2018_045.html`, 2018.

[53] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[54] Maider Vidal and José Manuel Amigo. Pre-processing of hyperspectral images. essential steps before image analysis. *Chemometrics and Intelligent Laboratory Systems*, 117:138–148, 2012.

[55] Jinru Xue and Baofeng Su. Significant remote sensing vegetation indices: A review of developments and applications. *Journal of sensors*, 2017, 2017.

Curriculum Vitae

José Alberto Montán López was born in CDMX, México, on August 5, 1996. He earned the Mechatronics Engineering degree from the Instituto Tecnológico y de Estudios Superiores de Monterrey, Guadalajara Campus in December 2019. He was accepted in the graduate programs in Computer Science in January 2021.

This document was typed in using LaTeX 2_{ε}[a] by José Alberto Montán López.

---

[a] The style file `phdThesisFormat.sty` used to set up this thesis was prepared by the Center of Intelligent Systems of the Instituto Tecnológico y de Estudios Superiores de Monterrey, Monterrey Campus