UNIVERSITÁ DEGLI STUDI DI MILANO-BICOCCA

Department of Physics "Giuseppe Occhialini"

Master's Degree in Physics

Phasor-Based Analysis of Hyperspectral Images for Plant Disease Detection

1st Supervisor:

Prof. Laura Sironi

Laura Sironi

Co-Supervisors:

Prof. Sergio Cogliati
Dr. Luca Tuzzi

Candidate:

William Colombini
ID: 892235

ACADEMIC YEAR 2022/2023

Abstract

Fusarium head blight (FHB) is a disease that can significantly reduce the yield of the common wheat. Currently, plant pathologists rely on laboratory methods for its diagnosis, which involves destructive procedures that are harmful to crops. To address this problem, optical methods for acquiring hyperspectral images have emerged in plant pathology, providing a non-destructive approach to detecting and characterizing plant diseases. This thesis project aims to develop and apply computational techniques and artificial intelligence models to automatically detect the presence of Fusarium head blight by exploiting hyperspectral data.

For this reason, a field experiment was conducted to determine the response of wheat spikes under different conditions. Specifically, heads of soft wheat (Triticum Aestivum) of two different varieties were cultivated: Cultivar Bingo, susceptible to the pathogen known as Fusarium Graminearum, and Cultivar Rebelde, resistant to the infectious fungus. For each variety, the experimental site was divided into twenty-four plots, half of which were infected by inoculating the pathogen, while the other half were control spikes. Furthermore, the effectiveness of two potential curative treatments were evaluated: a traditional "chemical" treatment, which acts by eliminating the pathogen, and an innovative "biological" treatment, consisting of a non-harmful fungus capable of competing with the pathogen and hindering its growth. Overall, the spikes can be classified into six categories for each variety: healthy untreated, healthy treated with traditional treatment, healthy subjected to the innovative treatment, infected untreated, infected treated with traditional treatment, infected subjected to the innovative treatment.

The hyperspectral images of both the healthy spikes and the infected spikes, subjected to different treatments, have been acquired by means of the "HyIce" spectrometer, designed to operate in the visible and near-infrared spectral range (400 - 1000 nm). I used the instrument to acquire images of sixteen spikes for each of the abovementioned classes.

The images acquired through the spectrometer were processed using algorithms developed with MATLAB. Initially, background noise, estimated from the signal obtained in the absence of the sample, was removed. Then, reflectance was evaluated by computing the ratio between the radiance reflected by the spikes and the signal reflected by a Lambertian surface (i.e. a white surface with 99% reflectance). Finally, individual grains of the spike were

identified using binary masks for further analysis.

Initially, I selected specific wavelengths corresponding to the colors red (669 nm), green (554 nm), and blue (472 nm) to reconstruct the RGB image associated with the hyperspectral one. This allowed us to observe chromatic differences between healthy and infected grains. This aspect was further confirmed by the observation and analysis of spectral reflectance signals. Indeed, healthy spikes, in contrast to infected ones, exhibit a local minimum around the wavelength of 662 nm. This value is associated with one of the absorption peaks of chlorophyll a. Therefore, infected spikes show, through the infection, a reduction in the pigments' concentration related to the presence of chlorophyll, which is usually high when vegetation is healthy.

Then, the hyperspectral images were analyzed by means of the phasor method to quickly and easily identify spectral differences associated to the spike categories. This method relies on applying the Discrete Fourier Transform (DFT) to project the signals as points in a two-dimensional space called the phasor plane, where the coordinates correspond to the real and imaginary parts of one of the harmonics of the transformed signal. The position of the points is related with the shape and properties of the signal and therefore similar reflectance spectra lie closely in the phasor plane. Specifically, if the DFT algorithm is applied to spectral windows with different widths (7, 15, 20 nm), it is possible to identify regions where spikes belonging to different classes lie in separate positions on the phasor plane. Finally, spectral windows with maximum separation between the centroids of the projections of spectra related to different treatments were identified using MATLAB code. In this way I identified ten spectral regions, ranging between 500 nm and 700 nm and in agreement with the literature, to implement an automatic classification method.

Since the development of Fusariosis entails morphological changes, such as the appearance of spots on the outer coating of the grains, I also analyzed the spatial and texture properties. To extract texture properties, grayscale images associated with the windows identified in the spectral analysis were selected. The Gray-Level Co-occurrence Matrix (GLCM) algorithm was applied to each of these images, allowing for the computation of texture properties in correspondence with the most significant spectral regions. Subsequently, through the application of a non-parametric statistical test, it was verified that some of these properties (contrast, energy, entropy, and homogeneity) are significantly different (p-value < 0.01) for samples of different classes.

Finally, I trained and tested several machine learning algorithms, including Support Vector Machine (SVM), Decision Trees (TREE), Neural Networks (NN), and K-Nearest Neighbors (KNN), by using as features the coordinates of the spectra on the phasor plane and the texture properties in the ten selected spectral windows. The entire dataset, consisting of 5000 pixels for each class, was then separated into two groups: one comprising 80% of the data (training set) and the other containing the remaining 20% (test set). The accuracy in discriminating between healthy and fusariosis-affected samples was 98.3%.

In conclusion, this study introduces an innovative method that leverages both the spectral and spatial characteristics of images to precisely identify diseased areas within wheat spikes. This outcome highlights not only the effectiveness of the selected features in representing pathological conditions but also the power of machine learning algorithms in interpreting these complex data. The potential advantages of such a system are manifold, spanning from early detection of infections in remote sensing analyses conducted, for instance, by drones, to targeted interventions aimed at reducing crop losses in cases where disease detection was not timely. Furthermore, integrating this method with other automated monitoring systems and decision support tools typical of Agriculture 4.0 could substantially enhance the efficiency and sustainability of agricultural practices.

Contents

Introduction

In contemporary agricultural practices, wheat (Triticum aestivum) emerges as a paramount crop. Regrettably, certain pathogens, including Fusarium head blight, pose significant challenges as they not only impair the infected spikes but also pollute the soil, rendering cultivation in subsequent years unfeasible. Presently, the sole method of diagnosis involves laboratory-based sample analysis; a positive result signifies a total loss of the crop. This scenario is particularly problematic in the context of current population growth and climate change, underscoring the necessity of enhancing agronomic efficiency to meet the increased demand for food within limited spatial resources. Forecasting future food demand, although complex, projects a potential demand surge between 59% and 98%. Consequently, minimizing food waste and pathogen incidence becomes imperative. A feasible solution may lie in real-time diagnostics, potentially utilizing remote sensing data. In this respect, the application of hyperspectral imaging tools coupled with advanced computational image analysis techniques offers a quantitative diagnostic approach. Within the scope of this thesis, I have employed a proximity-based approach, utilizing models trained on images of healthy and infected spikes to classify individual grains as either healthy or diseased.

Chapter 1 will introduce the foundations of radiometry and the ways in which it can help check the status of crops in a non-destructive way.

Thereafter, Chapter 2 provides a concise yet formal overview of the principal machine learning concepts and algorithms deployed in this study. This, alongside Chapter 3, constitutes the most technical segment. Chapter 3 elucidates the theoretical underpinnings of the experimental-analytical methodologies engaged in this research. Although not imperative for comprehensive understanding, these chapters offer insights into the methodologies employed in later sections.

The essence of this thesis is encapsulated in Chapter 4, which presents an analysis of hyperspectral images and the feature extraction process, alongside reflections on the applied methods.

Chapter 5 unveils the objectives and findings of this research, detailing the training and testing procedures of machine learning algorithms and the outcomes on the available datasets.

Chapter 1. Optical Analysis for Phytopathology Applications

1.1 Radiometry: A Technical Overview

Radiometry involves the study and application of measuring electromagnetic radiation, encompassing visible light among its components. This field is fundamental to various applications, notably optical analysis for phytopathology, where it plays a pivotal role in diagnosing and monitoring plant diseases through the analysis of hyperspectral images. This section provides an overview of the principles of radiometry, its application in plant health assessment, and the specific role it plays in detecting plant diseases.

Radiometry is vital in the field of optical analysis for plant health. By measuring the spectral characteristics of light reflected from or transmitted through plant tissues, researchers can detect subtle changes indicative of stress or disease. These changes may be attributable to variations in leaf pigmentation, cellular structure, or moisture content, each affecting the plant optical properties and, consequently, its spectral signature.

1.1.1 Fundamental Principles and Equations

Radiometry involves quantifying electromagnetic radiation in terms of its power at various wavelengths, providing a set of concepts and mathematical tools to describe light propagation and reflection. The fundamental units of radiometric measurements include the watt $[W]$ for power, the steradian $[sr]$ for solid angular measurement, and various derived units such as radiant flux, irradiance, radiance, and radiant intensity. Radiant Flux Φ refers to the total power of electromagnetic radiation emitted, transmitted, or received. If a spot is illuminated by light whose power is described by the finite spectral power distribution $P(\lambda)$ with $\lambda \in (\lambda_i, \lambda_f)$, then the Radiant Flux will be defined as:

$$\Phi = \int_{\lambda_i}^{\lambda_f} P(\lambda) d\lambda \tag{1.1}$$

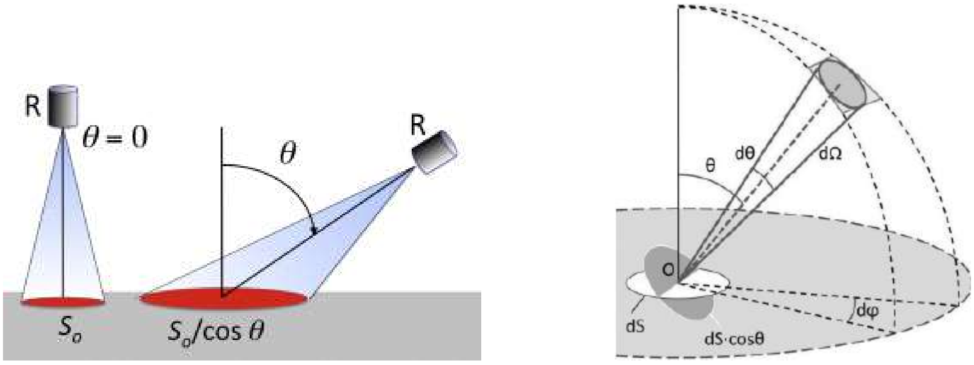
Since a real spot is not an ideal point, it is useful to define the Irradiance E, which is a measure of the power per unit area incident on a surface (i.e., the Radiant Flux Density):

$$E = \frac{d\Phi}{dS} \tag{1.2}$$

This value is influenced by the geometry between the light source (described by the Radiant Flux) and the spot, as illustrated in Figure 1.1.1a.

This necessitates the definition of a directional quantity similar to Irradiance, which is Radiance LL. Radiance describes the power per unit area per unit solid angle emitted or reflected in a specific direction. As depicted in Figure 1.1.1b, it is the radiant flux measured per unit solid angle and per unit projected area in a given direction, on a plane orthogonal to that direction, thus defined as:

$$L = \frac{d^2\Phi}{dScos\theta d\omega} \tag{1.3}$$



(a) Graphical visualization of Lambert's Law. Irradiance arriving at a surface varies according to the cosine of the angle.

(b) Visualization of Radiance through the solid angle dω and the projected area $dA^{\perp}$.

Figure 1.1.1

1.1.2 Application in Phytopathology

In phytopathology, radiometric measurements are indispensable for diagnosing plant health and identifying diseases. These measurements are sensitive to changes in plant optical properties, which can experience significant variations with the onset of stress or disease. For example, diseased plant tissues may undergo alterations in their reflectance, transmittance, and absorbance properties due to physiological and biochemical changes.

The mathematical formalization of radiance is invaluable for understanding and interpreting hyperspectral images, which provide spatially resolved spectral information about plant surfaces. The spectral reflectance R of a leaf, defined as the ratio of reflected radiance

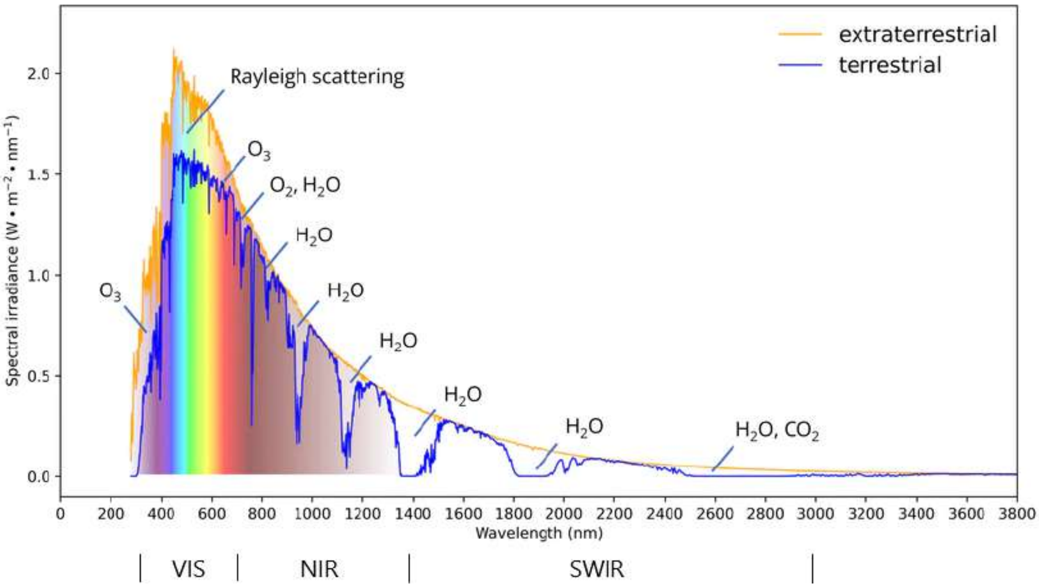


Figure 1.2.1: Absorption and scattering change the spectral distribution of sunlight as it passes through the atmosphere.

to incident irradiance, can be modeled as:

$$R(\lambda) = \frac{L_R(\lambda)}{E_I(\lambda)} \tag{1.1}$$

Here, $L_R(\lambda)$ represents the spectral radiance reflected by the leaf at wavelength λ and $E_I(\lambda)$ is the spectral irradiance incident on the leaf.

In the following section, we will describe the fundamental processes involved in measuring the spectral reflectance to provide a deeper understanding of this crucial quantity.

1.2 Matter Interaction with Sun Radiation

To understand how light interacts with plants, it is beneficial to describe light-matter interaction. Our focus lies on the interaction of matter with sunlight, specifically within the range of visible light and near-infrared light ($380nm\ to\ 1400nm$). As illustrated in Figure 1.2.1, the majority of solar radiation falls within the range of 200 to 3000 nm, peaking at approximately 500 nm. Atmospheric gases absorb light at various wavelengths. O_2 and O_3 absorb most of the ultraviolet radiation and water vapor is the dominant absorber in the NIR and SWIR regions. The visible spectrum ($380 - 750nm$) is less absorbed by the atmosphere. Indeed, approximately half of the incident radiation reaching the Earth surface is within the visible range, while the other half lies in the near and mid-infrared regions. This significantly influences the interaction between solar radiation and plants, explaining why evolution has favored pigments that absorb shorter, more energetic wavelengths for photosynthesis.

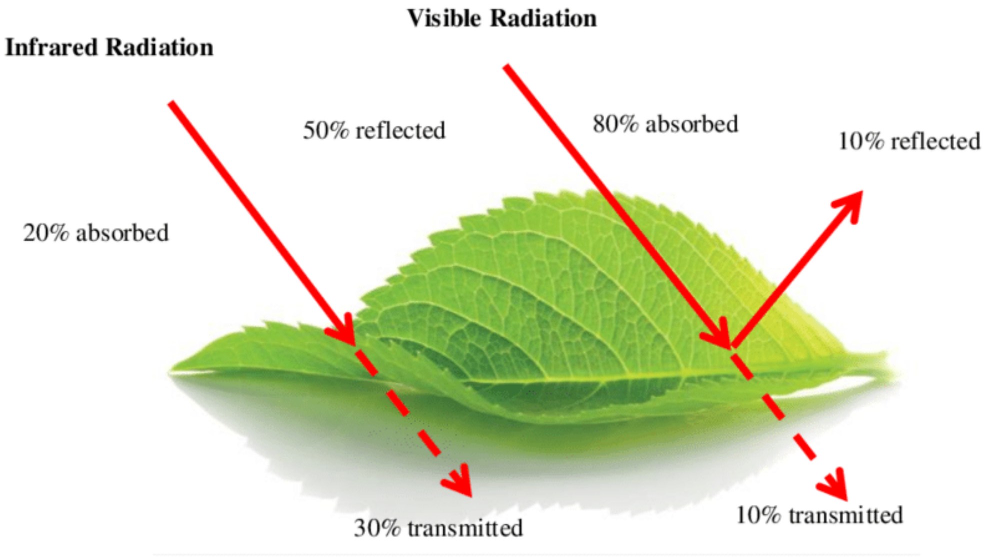


Figure 1.2.2: The effects of visible-IR light interaction with a leaf.

In more detail, light-matter interaction can result in three possible effects, depending on the interaction geometry [1], the wavelength λ involved, and the characteristics of the interacting material[2]: reflection, transmission, and absorption, as depicted in Figure 1.2.2.

Absorption usually occurs when the energy of photons is absorbed by the matter, increased electronic or vibrational energy. In plants, absorption happens primarily through pigments like chlorophyll, which absorb light in specific wavelength bands for photosynthesis.

Reflection and scattering refer to the redirection of light waves away from a surface or after encountering particles within the material. The surface structure and internal composition of leaves, including the waxy cuticle and cellular arrangement, significantly influence the reflection or scattering of light.

Transmission describes the passage of light through a medium. In plants, some light passes through leaf tissues, with the amount and quality of transmitted light being dependent on leaf thickness, cell structure, and pigmentation.

1.3 Pigments Role in Light Interaction

This section explores the function of pigments in light absorption, how these interactions are pivotal in plant health, and their significance in the context of hyperspectral imaging for disease detection.

For growth, plants essentially require CO_2, H_2O and light. These components enable

[1] i.e., the direction of incident radiation and of the observer relative to the interacting object

[2] The response of plants to different spectral radiations and intensities varies among species and also depends on growing conditions

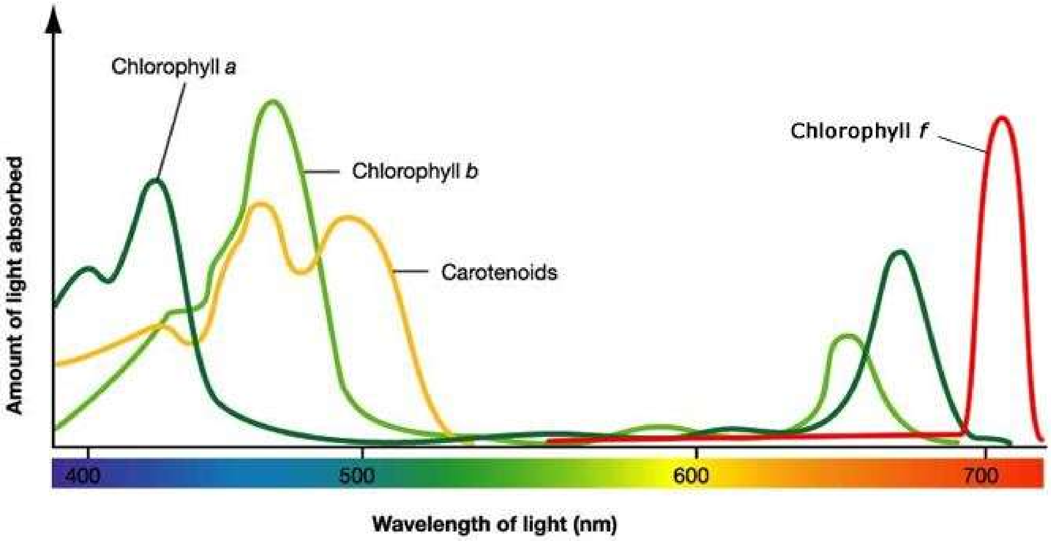


Figure 1.3.1: Absorption peak of the main pigments in plant.

them to synthesize glucose molecules through a chemical process known as photosynthesis. This process unfolds in three stages, involving:

- Sunlight energy is captured.
- Energy-carrying molecules, such as adenosine triphosphate (ATP), and reduced electron carrier molecules, like nicotinamide adenine dinucleotide phosphate (NADPH), are produced by organelles known as chloroplasts.
- ATP and NADPH are utilized to synthesize organic, glucose-based molecules from atmospheric CO_2.

The process can be summarized by the chemical reaction:

$$6CO_2 + 12H_2O \xrightarrow[\text{pigments}]{\text{light}} C_6H_{12}O_6 + 6O_2 \tag{1.1}$$

The primary pigment found in plants is chlorophyll, which exists in five main forms: chlorophylls a, b, c, d, and f.

In plants, chlorophyll a and chlorophyll b serve as the principal photosynthetic pigments, effectively absorbing light energy at wavelengths in two ranges around $450nm$ and $650nm$. Neither of these pigments absorb photons with wavelengths between approximately $500 - 600nm$; thus, light of this range is reflected by plants. When these photons are absorbed by the pigments in our eyes, we perceive this reflection as green.

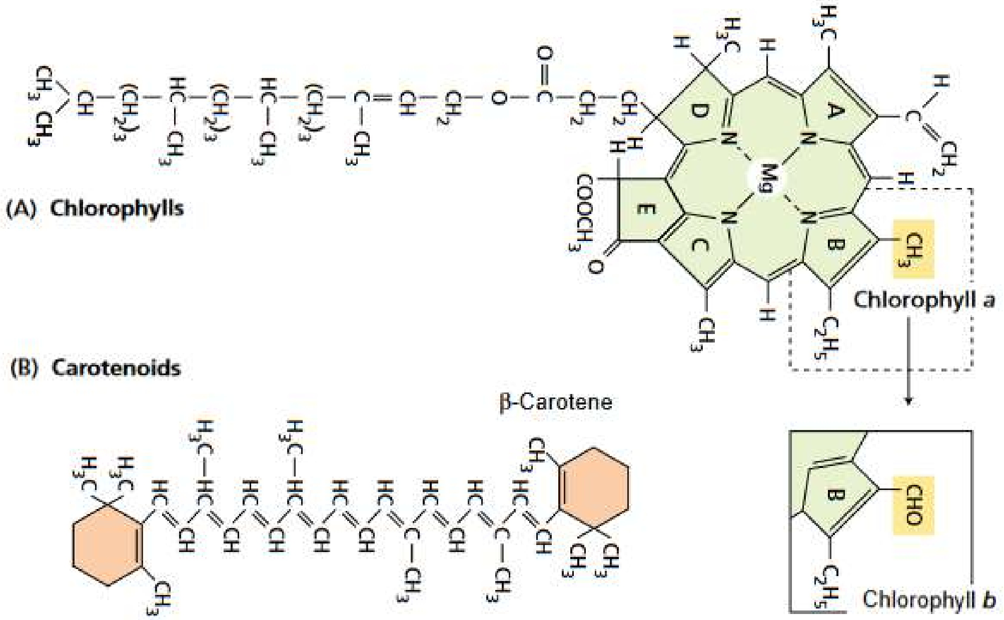


Figure 1.3.2: The chemical structure of the main photosynthetic pigments highlighting the difference between Chlorophyll a and b.

Chlorophyll a

Chlorophyll a is the most prevalent pigment in plants and plays a crucial role in photosynthesis. It acts as the primary light absorber in the photosystems, directly facilitating the conversion of light energy into chemical energy. The molecule of chlorophyll a is characterized by a porphyrin ring—a flat, square structure composed of four nitrogen-containing pyrrole rings linked by methine bridges. This ring system encloses a magnesium ion at its center, vital for its light-absorbing capabilities. Additionally, the porphyrin ring of chlorophyll a features a long phytol tail, a hydrophobic chain that secures the molecule to the thylakoid membrane within chloroplasts.

This distinct structure enables chlorophyll a to predominantly absorb light in the blue-violet and red portions of the electromagnetic spectrum.

During photosynthesis, chlorophyll a captures photons, exciting electrons to a higher energy state. These high-energy electrons are subsequently transferred to the electron transport chain, triggering the sequence of reactions that transform light energy into ATP and NADPH. These energy carriers are then utilized in the Calvin cycle to synthesize sugars.

Chlorophyll b

Chlorophyll b serves as an accessory or secondary light-absorbing pigment, complementing and enhancing the light absorption capabilities of chlorophyll a. Its absorption spectrum is shifted towards the green wavelengths because of its distinct chemical structure, which includes a formyl group instead of a methyl group in the porphyrin ring. As a result, chlorophyll b can absorb photons that chlorophyll a cannot, significantly increasing the range of photons from sunlight that plants are able to utilize. This augmentation enhances the efficiency of the photosynthetic process by enabling the absorption of a broader spectrum of light.

While chlorophyll b does not directly participate in the conversion of light energy to chemical energy, it plays a crucial role in capturing additional light energy and transferring it to chlorophyll a. This process, known as energy transfer, is vital in environments where light intensity and quality vary.

Carotenoids

As we can see in Figure 1.3.1, chlorophylls are not the only pigments involved in light absorption for plants. An important group of accessory pigments is the carotenoids. Carotenoids are a diverse group of pigments that include carotenes and xanthophylls. These pigments absorb light in the blue-green and violet parts of the spectrum, offering a protective role against photooxidation by dissipating excess light energy as heat.

Carotenoids are characterized by their long, conjugated double-bond systems, which allow them to absorb light at different wavelengths with respect to chlorophylls. Their structure makes them highly effective in neutralizing free radicals, protecting the photosynthetic apparatus from oxidative damage.

Besides their protective role, carotenoids are involved in light harvesting for photosynthesis. They absorb and transfer energy to chlorophyll a, although with less efficiency than chlorophyll b. In conditions of high light intensity, carotenoids can prevent photodamage by safely dissipating excess energy.

Anthocyanins

Lastly, another important pigment class to take into account are the anthocyanins. Anthocyanins are a class of water-soluble pigments that belong to the flavonoid group. They are responsible for the red, purple, and blue colors observed in many fruits, vegetables, grains, and flowers. These pigments play a significant role in the plant kingdom, not only contributing to the aesthetic appeal and coloration but also offering protection against various environmental stresses. In fact, they are directly involved in plant defense mechanisms against herbivores and pathogens. Their presence can deter herbivores due to their bitter taste and can also have antimicrobial properties, protecting plants from various diseases.

In addition, they are fundamental for the reproduction, providing attraction to pollinators, such as bees and butterflies, through their coloration. Anthocyanins absorb light in the visible range, primarily between $400 - 550nm$, which contributes to their coloration and protects plants from harmful ultraviolet radiation.

1.4 Photosynthesis Light Absorption

To understand the signal observed by a measurement of the spectral radiance reflected by a leaf, it is useful to focus on the first step of the photosynthetic process: the light absorption.

If we have a molecule of a pigment Pgm in its ground energy state, it will be promoted to a higher energy level following the absorption of a photon with an energy $h\nu$:

$$Pgm + h\nu \rightarrow Pgm^* \tag{1.1}$$

We can describe this process by using a semiclassical perspective. We can assume that pigment molecules are characterized by discrete energy levels, or quantum states, determined by the quantum mechanics governing their electrons and atomic nuclei. Each energy level corresponds to a specific arrangement of the electrons in the molecule.

When a molecule absorbs a photon (a quantum of light), the energy of the photon is transferred to an electron in the molecule. For absorption to occur, the energy of the photon must match the energy difference between the electron initial state and a higher available energy state, so different wavelengths are involved with different pigments, and in each molecule, we can have multiple excited levels. Absorption of a photon causes an electron to 'jump' from the ground state to a higher energy level, creating what is known as an excited state. This state is less stable than the ground state, and the electron in this higher energy level seeks to return to a lower energy state, releasing the absorbed energy in the process.

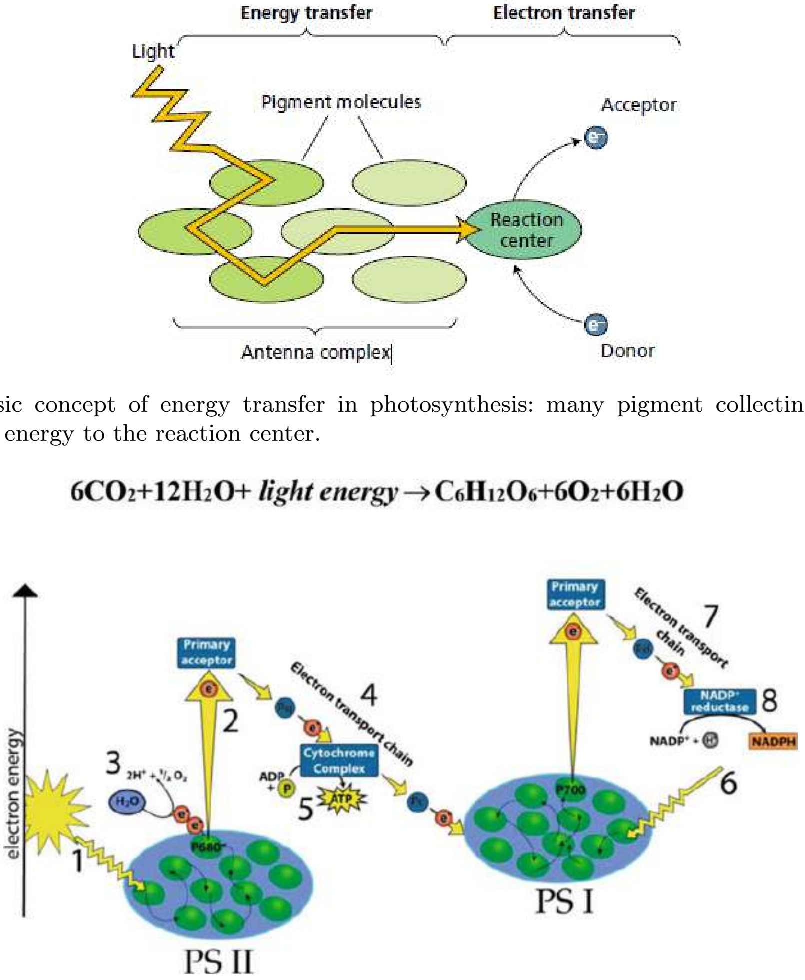
In general pigments can follow different emission paths:

- *Photon re-emission:* the pigment may emit a photon to revert to its ground state, a phenomenon called fluorescence. During this process, the emitted light has a slightly longer wavelength (and thus lower energy) than the absorbed light, due to some energy being lost as heat prior to the emission of the fluorescent photon.

- *Heat conversion:* the excited pigment can convert its excitation energy directly into heat, returning to its ground state without emitting any photons;

- *Energy transfer:* the pigment in its excited state can transfer its energy to a different molecule, facilitating processes where this energy is utilized elsewhere;

- *Photochemical reactions:* the excited state energy can initiate chemical reactions, known as photochemistry. These reactions are crucial for processes like photosynthesis



(a) The basic concept of energy transfer in photosynthesis: many pigment collecting light and transferring energy to the reaction center.

(b) A more complete scheme of the whole photosynthetic process involving multiple energy absorption and transfer

Figure 1.4.1: A schematic of the photosynthetic process is presented. A single step is depicted in (a). This process is repeated multiple times, culminating in the comprehensive process illustrated in (b)

and are among the fastest chemical reactions, to ensure efficiency over the other energy disposal methods.

To optimize the energy utilized in photochemical reactions—thus facilitating the production of essential molecules for plant survival—the majority of pigments function as an *antenna complex*. This complex directs energy towards the area where chemical oxidation and reduction reactions occur, resulting in long-term energy storage. This critical area is known as the *reaction center complex*.

1.5 Solar-Induced Chlorophyll Fluorescence

Being one of the ways molecular excited levels return to their initial state, it is important to consider the chlorophyll fluorescence emission ranges of the photosystems involved in the emission, which are PSI and PSII (photosystems one and two). During this electron transport, some electrons can be lost and react with oxygen, creating harmful substances called reactive oxygen species. To prevent this, plants have a protective mechanism called non-photochemical quenching (NPQ). NPQ dissipates excess energy as heat, which can be observed as fluorescence emission from PSII. PSI, on the other hand, transfers electrons from the molecule ferredoxin to NADP+, generating more NADPH. This NADPH is crucial for the Calvin cycle, which converts carbon dioxide into organic compounds. PSI also contributes to ATP production through a process called cyclic electron flow.

Optical sensing allows for the monitoring of the pivotal process of solar-induced chlorophyll fluorescence (F), which can be used to monitor vegetation health and functioning.

1.6 Optical Properties of Leaves and Canopy

Once described from a microscopical point of view, the processes that contribute to the optical response of a plant, it is possible to offer a more macroscopical description. Chlorophyll and carotenoids not only play unique roles and display specific absorption patterns but also contribute significantly to the leaf optical characteristics observable through reflectance techniques. The levels and ratios of these pigments within a leaf profoundly affect its reflectance spectrum. Chlorophyll, the key pigment in photosynthesis, absorbs light primarily in the red and blue wavelengths, leading to decreased reflectance in these areas and creating dips in the spectrum known as the "chlorophyll absorption bands." Conversely, carotenoids absorb light across a wider range, particularly in the blue-green wavelengths. Analyzing leaf reflectance spectra enables the quantitative evaluation of chlorophyll and carotenoid content, offering insights into the leaf physiological and biochemical status.

Reflectance measurement is a method to explore the optical effects of chlorophyll and carotenoids within leaves, providing a deeper understanding of their functions in photosynthesis and plant health.

The optical domain of solar radiation (380 nm - 2500 nm) is crucial for vegetation research, encompassing most of the absorption features relevant to plant studies. Leaves interact with solar radiation directly or indirectly through radiation scattered by other leaves, leading to energy being either absorbed or scattered via reflection and/or transmission. This interaction, influenced by the leaf cellular structure, morphology, physiology, and surface characteristics, results in varying energy distribution. Leaves partially reflect solar radiation

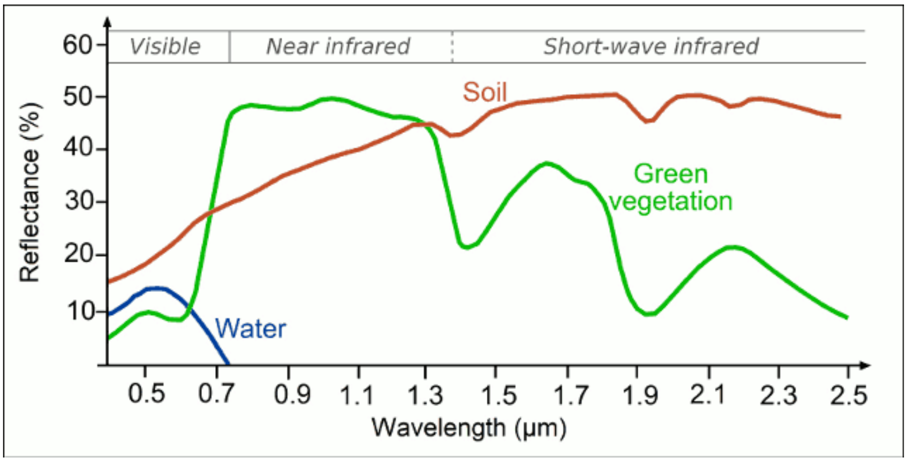


Figure 1.6.1: Examples of spectral signatures for water, soil, vegetation.

in both diffuse and directional manners due to their internal structures, showcasing distinct absorption, reflection, and transmission patterns across the visible (VIS), near-infrared (NIR), and mid-infrared (MIR) spectra.

These spectral signatures are shaped by the absorption of photosynthetic pigments like chlorophyll in the visible spectrum, structural characteristics of the leaf in the NIR spectrum, and water and protein content in the SWIR spectrum. Chlorophyll absorption in the blue and red parts of the spectrum and its reflection in the green part, accounting for the leaf green appearance, alongside the high reflectance attributed to leaf structure in the NIR range and the impact of water content in the MIR range, underline the complex interaction between light and vegetation.

Literature provides a qualitative description of the characteristic reflectance spectrum of vegetation compared to soil and water, as reported in Figure 1.6.1.

It is useful to note that in Remote Sensing measurements, such as drone in-loco optical measurements, is heavily influenced by the properties of leaves, including their number, arrangement, and orientation. These interactions are crucial for analyzing the canopy structural qualities, like the Leaf Area Index (LAI), and for assessing leaf biochemical characteristics, such as chlorophyll levels, through hyperspectral imaging.

The term "canopy architecture" describes the spatial arrangement of plants at the ground level, including how densely and in what orientation the leaves are positioned within a vegetated area. In a uniform canopy, the assumption is that there is an even distribution of leaf elements, quantified by the LAI, which measures the leaf area per unit ground surface area, representing the collective presence of plants in a particular region.

1.7 Fusarium Head Blight (FHB) in Wheat

Plants face challenges from living (biotic) and non-living (abiotic) factors that impact their survival. Biotic factors include diseases caused by pathogens like Fusarium, while abiotic factors include extreme temperatures, water stress, and nutrient deficiencies.

Understanding these stress factors is crucial for plant survival. Stress triggers biochemical responses, affecting processes like photosynthesis. Optical analysis can detect plant stress by analyzing vegetation spectral signature. Fusarium, a pathogenic fungus, can stimulate or decrease photosynthetic capacity and cause visible leaf lesions. Assessing multiple plant traits is important for evaluating recovery from stress.

Fusarium Head Blight (FHB) is a significant fungal disease that affects wheat crops worldwide. FHB is caused by various species of Fusarium fungi, with F. graminearum being the most aggressive. The disease can lead to yield losses and reduce grain quality. FHB progresses through phases of infection, colonization, toxin production, harvest, and storage, impacting grain development and leading to small, discolored grains with reduced protein and gluten content. FHB also produces mycotoxins that pose health risks. Integrated crop protection strategies and breeding resistant wheat varieties are important for managing FHB. Identifying and characterizing disease symptoms in the early stages is crucial for effective management.

1.7.1 Symptoms of FHB in Wheat

The symptoms of FHB are distinctive, starting with the appearance of small, water-soaked lesions on the wheat heads. These lesions quickly become whitish or light pink, and the infection can spread, causing the entire head to blight. The fungus can also infect the stem or peduncle, resulting in brown or purple discoloration. Prolonged wet weather leads to the production of pink to orange salmon spore masses on infected caryopses as shown in Figure 1.7.1. Symptom expression is influenced by environmental conditions, with high humidity and moderate temperatures ($15 - 30°C$) during flowering being particularly conducive to disease development.

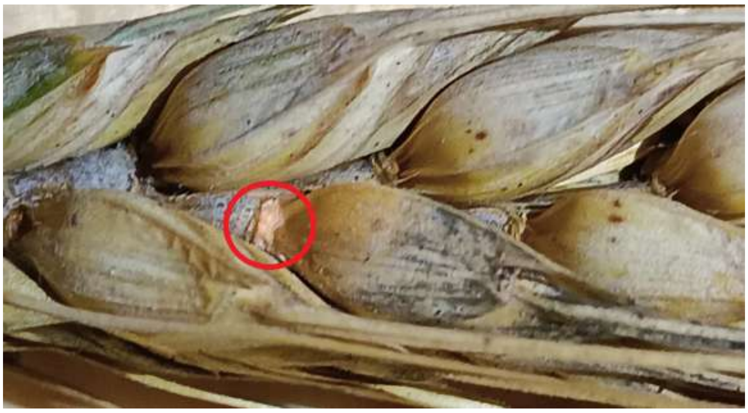


Figure 1.7.1: A portion of a wheat spike affected by FHB with pink lesion (red circle).

Chapter 2. Theoretical Aspects of ML Techniques

2.1 Overview of Machine Learning

Machine Learning (ML) is a subset of artificial intelligence that focuses on building systems that learn from data. Instead of being explicitly programmed to perform a task, these systems are trained using large dataset that give them the ability to learn how to perform the task.

A dataset is a structured collection of data that is used to train and evaluate machine learning models. In other words, it is a set of examples or instances that represent the input and output pairs for the learning algorithm. A dataset typically consists of two main components: the features, which are the measurable properties or characteristics of the data instances, and the labels, which represent the desired output or the value to be predicted. Features are also referred to as input variables, independent variables, or predictors. Features can be characterized by numerical or categorical values, or more complex structures such as images or text. A set of features, associated with the corresponding label, is defined as a pattern.

2.1.1 Supervised Learning and Unsupervised Learning

ML is fundamentally divided into two main approaches: supervised learning and unsupervised learning. Each of these approaches has its own unique methodologies, applications, advantages and disadvantages.

Supervised learning involves training a model on a labeled dataset, which means that each training example is paired with an output label. The model is then used to make predictions or decisions without being explicitly programmed to perform the task. More formally, this approach produces a model trained from a dataset which can be described with a function $f : X \to Y$ that maps an input X to an output Y. For example, if we have N known points in our dataset and we want to use them to train our model, the data will be represented as a given set of input-output pairs $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$, with y_i being the labels of the observation x_i. The observation can be described by a set of characteristics, called features.

For example, if we assume to have a set of N observations about the health state of a person based on three parameters: gender, height and weight. For each person x_i, characterized by three features, we will have a qualitative (or logical) response y_i: healthy or unhealthy.

The primary advantage of supervised learning is its ability to predict outcomes for new, unseen data, making it highly effective for classification and regression problems. Its main disadvantage, however, lies in the necessity for a large and well-labeled dataset, which can be time-consuming and costly to obtain. Supervised learning is most applicable in scenarios where the relationship between the input and output variables is well-understood and where historical data is available to train the model, such as in spam detection or real estate pricing.

Unsupervised learning, on the other hand, involves learning patterns from a set of inputs X without reference to any labels. Its major advantage is the ability to discover hidden structures in data without the need for labeling, making it useful for exploratory data analysis, clustering, and dimensionality reduction.

In the previous example, if we have just the features about each person x_i, we could use unsupervised methods to understand if observations with different features have a particular pattern.

The main disadvantage is that the outcomes of unsupervised learning are often more challenging to interpret, and there is no straightforward way to validate the model performance since the correct outputs are unknown. Unsupervised learning is particularly applicable in situations where the goal is to explore data to find patterns or groupings, such as customer segmentation in marketing or anomaly detection in network security.

2.1.2 Classification and Regression

In the context of supervised learning, regression and classification problems are distinguished by the output variable Y.

Regression involves predicting a continuous quantity. For a given dataset $\mathcal{D}$ described by $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$, where x_i is an observation and y_i is the real number related to it, the goal is to learn a mapping function $R : X \to \mathbb{R}$ such that the predicted values $\hat{y}$ are as close as possible to the actual values y in terms of a predefined loss function $L(y, \hat{y})$ that measures how "good" the prediction is. An example of a loss function can be the sum of the Euclidean distance between the expected value y and the predicted one $\hat{y}$.

Classification involves predicting a discrete label. A general label of an observation will be $y_i \in 1, 2, \ldots, K$ for a K-class classification problem. The goal is to learn a mapping function $C : X \to 1, 2, \ldots, K$ that accurately predicts the class label y of new inputs. In this case, the loss function $L(y, \hat{y})$ is harder to define. The simplest way is to evaluate the number of well-classified elements over all the test set, but this is not always representative or fully useful for the goal of the algorithm.

The Process of Creating a Trained Algorithm

The development of a trained classification algorithm involves a systematic process outlined in the following steps:

1. **Data Collection and Preprocessing:** This initial phase involves the accumulation of a comprehensive dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, where x_i represents the features and y_i corresponding labels for each instance. Preprocessing steps such as normalization, handling missing values, and data augmentation are applied to prepare $\mathcal{D}$ for effective analysis.

2. **Feature Selection and Extraction:** Critical to enhancing the algorithm efficiency, this step focuses on identifying a subset of relevant features $\mathcal{F} \subseteq \{f_1, f_2, \ldots, f_n\}$ that significantly influence the classification outcomes.

3. **Model Selection:** Based on the characteristics of $\mathcal{D}$ and the problem at hand, an appropriate model is chosen from a set of candidates. Some common techniques include Decision Trees, SVM, Neural Networks, and K-NN, etc. The selection criteria include complexity of the model, interpretability, and suitability for the data nature.

4. **Training the Model:** During training, the model $\mathcal{M}$ learns to map inputs to desired outputs by minimizing a loss function $\mathcal{L}(\mathcal{M}(\mathbf{x}), y)$ over the training data, adjusting its characterizing parameters to fit data patterns and relationships.

5. **Validating the model:** This step involves the iterative adjustment of the model hyperparameters to enhance performance, guided by the evaluation metrics. The performance of the model is quantified using different metrics such as accuracy, precision, and recall. These metrics estimate the performance of the algorithm, which is the ability of the algorithm to achieve its predictions on the training dataset.

6. **Testing the model:** The final step in the classification process is testing the algorithm on new data. This phase is critical to ascertain the model generalizability and effectiveness in real-world scenarios. The test set, ideally, should be a collection of data that the model has never seen before, ensuring that the performance metrics are indicative of the algorithm true predictive power. Testing provides valuable insights into how the model will perform in actual applications, highlighting areas of strength and identifying any limitations.

2.1.3 Performance of the Algorithm

The evaluation of the true performance of an algorithm whenever its configuration changes necessitates a vast amount of data, which is often not available. Consequently, the performance estimated during training and validation steps diverges from the "actual" performance assessed during the testing phase. The testing error offers insights into the model ability to generalize its predictions to new data. Achieving a low testing error is paramount as it signifies that the model excels not only with the training dataset but also in processing unseen datasets. However, the evaluation of the testing error invariably requires fresh, unseen data.

Conversely, the training error reflects how well the model has assimilated the training dataset. A minimal training error indicates a good fit to the training data, but it does not ensure superior performance on new, unseen data due to the potential risk of overfitting. Overfitting happens when a machine learning model overly familiarizes itself with the training data, capturing noise or random fluctuations instead of the underlying pattern. Hence, while the model might exhibit exceptional performance on the training data, its performance deteriorates on new, unseen data because of poor generalization. This issue arises when the model becomes overly complex, paying undue attention to trivial details irrelevant to the larger dataset.

Nevertheless, assessing training error is crucial as it offers some indication of the algorithm actual performance. Resampling methods, such as K-Fold Cross-Validation, are among the several strategies employed to glean useful information during the training and validation phase. These methods help in estimating the model performance more accurately by mitigating the limitations posed by the unavailability of extensive data.

2.1.4 K-Fold Cross-Validation and Performances

The idea behind this method is to divide the dataset $\mathcal{D}$ into K different subsets of unique elements, called *folds*, of approximately equal sizes.

At this point, $K-1$ folds are used to train the algorithm, and the last one is utilized for validation and evaluation of performance. This process is repeated K times, with a different fold used for validation during each iteration.

A concise overview of key metrics is crucial to assess the predictive capability of an algorithm. This discussion focuses on a binary classification problem, where one class is labeled as Positive and the other as Negative. During the validation phase, the number of observations correctly or incorrectly predicted by the algorithm can be identified. These observations are typically divided into four distinct categories:

- **True Positives (TP):** Instances where the model correctly predicts the positive

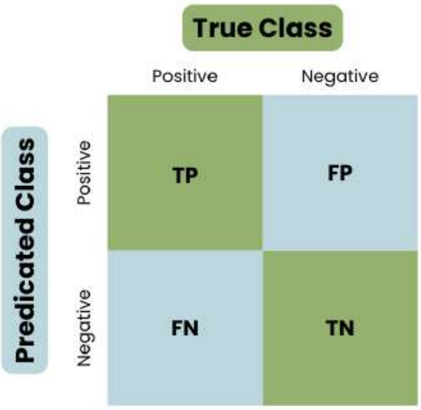


Figure 2.1.1: The structure of a simple binary class Confusion Matrix.

class. These are cases that are actually positive and are also predicted as positive by the model.

- **True Negatives (TN):** Instances where the model correctly predicts the negative class. These are cases that are actually negative and are also predicted as negative by the model.
- **False Positives (FP):** Instances where the model incorrectly predicts the positive class. These are cases that are actually negative but are mistakenly predicted as positive by the model, also known as "Type I error."
- **False Negatives (FN):** Instances where the model incorrectly predicts the negative class. These are cases that are actually positive but are wrongly predicted as negative by the model, also known as "Type II error."

Based on these values, it is possible to estimate various metrics. The most common ones include:

- **Accuracy**, which is the proportion of true results (both true positives and true negatives) among the total number of cases examined. The formula for accuracy is as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.1}$$

- **Precision**, which measures the proportion of positive identifications that were actually correct. Precision is crucial in scenarios where the cost of false positives is high. It is calculated as:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2.2}$$

- **Recall** (or Sensitivity), which indicates the proportion of actual positives that were correctly identified. This metric is vital in situations where failing to detect a positive case could have serious implications. The formula for recall is:

$$\text{Recall} = \frac{TP}{TP + FN} \tag{2.3}$$

- **Specificity**, which measures the proportion of actual negatives that were correctly identified. Specificity is important for accurately identifying negative cases in certain contexts. It is given by:

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{2.4}$$

A common graphical representation for these metrics is known as the *Confusion Matrix*. It is a table with two dimensions, "Actual" and "Predicted", each containing the categories "Positive" and "Negative", resulting in four possible outcomes. Figure 2.1.1 illustrates an example of this representation.

2.2 K-Nearest Neighbors (KNN)

The KNN model aims to predict the class (label) of data by identifying the label of the nearest observation with a known label. Assume the model is trained on a dataset consisting of pairs $(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)$ with L different labels $l_1, \ldots, l_L$. For a positive integer $K < N$ and an unknown observation x_0, the KNN classifier selects the first K points from the training dataset that are closest to x_0, termed the Nearest Neighbors. The probability that x_0 has label l_j is estimated using these K points, and based on this, the new label is predicted.

The method evaluates this probability simply by counting how many of the K points have a specific label and applying a majority rule. The probability for each label l_j is calculated as:

$$P\left(y(x_0) = l_j\right) = \frac{1}{K} \sum_{x_i \in NN} V(x_i) \tag{2.1}$$

where $V(x_i)$ is a vote function. An example of $V(x_i)$ is:

$$V(x_i) = \begin{cases} 1, & \text{if } y_i = l_j \\ 0, & \text{otherwise} \end{cases} \tag{2.2}$$

The label with the highest probability is selected for the new observation.

Adjustments can be made to enhance this approach. A more advanced method might consider the distance from each point in the set of the Nearest Neighbors to x_0 to weight the votes. This requires a metric to evaluate the distance for each feature. Assuming the

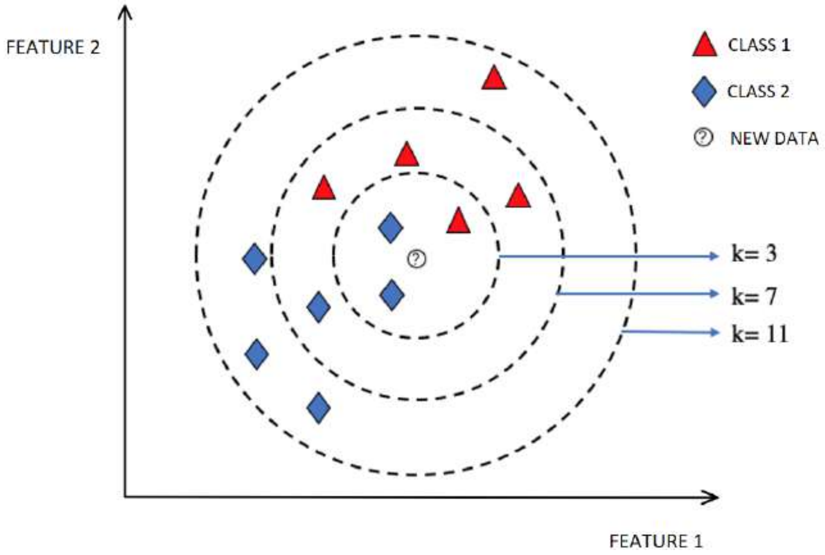


Figure 2.2.1: Representation of different selection of K points based on two feature plot.

feature set F consists of real numbers, one option for distance measurement is the Euclidean distance. The label is then chosen by finding:

$$\max_{l_1,\ldots,l_L}\left[\sum_{f\in F}\sum_{x_i\in NN} d_f(x_i,x_0)V(x_i)\right] \tag{2.3}$$

where $d_f(x_i, x_0)$ measures the distance between the f-th feature of x_i and x_0.

Given that these algorithms are distance-based, they can be sensitive to the scale of the data. Features with larger scales can disproportionately influence the distance calculation, potentially leading to biased results. This issue needs data normalization or standardization before applying KNN. Another significant concern is choosing the number of neighbors k, which is crucial for KNN effectiveness. A small k can make the model overly sensitive to noise (overfitting), while a large k can overly smooth the decision boundary (underfitting). Optimal k selection often involves cross-validations.

2.3 Logistic Regression

Despite its name, logistic regression is a statistical method used for binary classification tasks, and it can also be extended to multiclass classification within certain frameworks. It calculates probabilities using a logistic function, and the output label is determined based on the most probable value.

2.3.1 Mathematical Foundations

The logistic function is crucial in the model training process. Consider a training set of N observations $x_1, \ldots, x_N$, each characterized by a set of F features. Therefore, a specific observation x_i is defined as $x_{i,1}, x_{i,2}, x_{i,F}$. During training, the logistic model aims to minimize a loss function and selects a vector of *weights* $w_1, \ldots, w_F$ associated with the features, as well as a *bias* term. All terms are real numbers. The weight w_j indicates the significance of the input feature f_j in the classification decision. It can be positive, providing evidence that the instance should be classified in the positive class, or negative, indicating that the instance belongs in the negative class.

The logistic regression model is based om the logistic (or sigmoid) function, which transforms any real-valued number into a value between 0 and 1, suitable for interpretation as a probability. This function is expressed as:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{2.1}$$

With $\sigma(z) \in [0, 1]$ ranging from 0 to 1, the class can be determined using a straightforward criterion:

$$label(z) = \begin{cases} 0, & \text{if } \sigma(z) < 0.5 \\ 1, & \text{if } \sigma(z) > 0.5 \end{cases} \tag{2.2}$$

The value z is central to the algorithm. For a given observation x_i, z is calculated as a linear combination of x_i features and a vector of F *weights* values $w_1, \ldots, w_F$ plus a *bias* term. Thus, z is determined by:

$$z(x_i) = \left(\sum_{f=1}^{F} w_j x_{i,j} \right) + b \tag{2.3}$$

During model training, weights and bias are adjusted based on the training dataset X, a learning rate α and a cost (or loss) function L.

The training begins with initial (often random) settings of weights and bias, establishing temporary values to evaluate an observation label. It then proceeds as follows:

1. Select a point $x_i \in X$, compute z using the current weights and bias;
2. Predict x_i label using $\sigma(z(x_i))$, then select the label as described;
3. If the predicted label differs from the actual label y_i, adjust weights and bias according to α

This process iterates over the entire dataset X, aiming to minimize the loss function.

Upon completing training, the weights and bias are finalized. For a new observation x_0, its label is predicted by evaluating $\sigma(z(x_0))$ and subsequently determining the label.

A significant challenge with this method is its sensitivity to unbalanced training datasets, which can impair performance if there is a notable class imbalance. Additionally, because the decision boundary is linear, stemming from the linear combination used to compute z, the model may struggle with complex patterns.

2.4 Neural Networks (NN)

Neural networks are inspired by the structure and function of the brain neurons, serving as the fundamental units of computation. Artificial neural networks (ANNs) are composed of interconnected units or nodes, referred to as artificial neurons. These neurons are structured in layers: an input layer receives the data, hidden layers process the data, and an output layer delivers the final decision or prediction.

The learning process in neural networks involves adjusting the connections (weights) between neurons. This adjustment is primarily performed through backpropagation, in conjunction with optimization techniques such as gradient descent. The network makes initial predictions, measures them against the actual outputs, and modifies its weights based on the discrepancies, incrementally enhancing its predictive accuracy.

2.4.1 Mathematical Elements of a Simple Neural Network

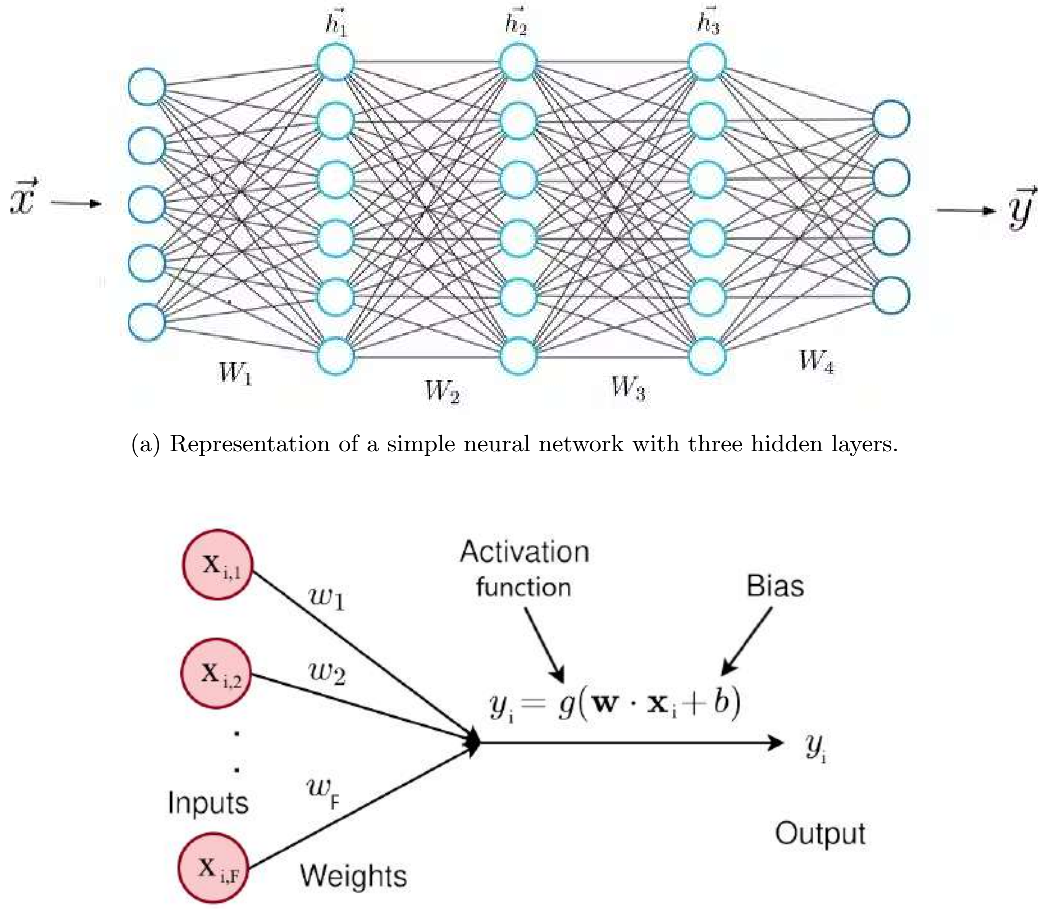
The fundamental element of a neural network is the connections between neurons. Consider a dataset $\mathcal{D}$ with elements featuring F neurons, assuming it is the simplest neural network form, as illustrated in the Figure 2.4.1b.

Upon receiving an observation $x_i = x_{i,1}, \ldots, x_{i,F}$, the output label y_i is determined by considering the weight of each connection $W = w_1, \ldots, w_F$, the bias b, and an activation function g, in a manner akin to logistic regression. Indeed, the computation of the value z is similar to that in logistic regression.

A common and simple activation function is the sigmoid function (Eq. 2.3), alongside others like the hyperbolic tangent function and the rectified linear unit (ReLU), defined as::

$$\begin{aligned} tanh(z) &= \frac{e^z - e^{-z}}{e^z + e^{-z}} \\ ReLU(z) &= max(0, z) \end{aligned} \tag{2.1}$$

In more complex neural networks, the output of one layer becomes the input for the next, creating significantly more complex connections. An example is illustrated in the Figure 2.4.1a.



(a) Representation of a simple neural network with three hidden layers.

(b) Representation of the input to output process: a layer to a single output.

Figure 2.4.1: Visual representations of neural network concepts.

2.4.2 Training of a Simple Neural Network

The training of a simple neural network with a single layer begins with the initial assignment of weights and biases, which can be randomly selected. The training process for all items in dataset $\mathcal{D}$ includes:

1. Passing an observation x_i through the network to produce a response y_i;

2. Computing a Cost Function or Loss Function to evaluate the deviation of the prediction y_i from the actual label $y_{i,REAL}$. An example of such a function is the squared error cost:

$$L(y_i, y_{i,REAL}) = (y_i - y_{i,REAL})^2 \tag{2.1}$$

   $(y_i - y_{i,REAL})^2$. It is crucial to note that the derivative of the Cost Function indicates whether the label value is overestimated or underestimated;

3. Adjusting the weights based on the derivative of the cost function and a *Learning Rate*

(LR), which dictates the degree of weight adjustment after an error. This step, known as backpropagation, can be simplified as:

$$W_{NEW} = W_{OLD} \left(1 - LR * \frac{\partial L}{\partial y_{i,REAL}}\right) \tag{2.2}$$

This iterative training process, excluding the initial setup, is repeated multiple times, each iteration termed an *Epoch*. In multilayer Neural Networks, the *Loss Function* is tailored to correct the weights at each layer. Here, backpropagation begins from the last layer, closest to the output.

2.5 Support Vector Machines (SVM)

The core principle of Support Vector Machines (SVM) is to find the optimal hyperplane (or more generally, a hypersurface) that most effectively separates data points of different classes in the feature space. Given a dataset X, with F features (whether continuous or discrete), these data points can be plotted in an F-dimensional space (known as the hyperspace), where each axis represents a feature.

The main challenge in developing an SVM model is to discover an $(F-1)$-dimensional hypersurface that splits the hyperspace into L distinct areas, each area encompassing all the points $x \in X$ that have the same label y_l. It is recognized that, in real-world applications, it might not be possible for a hypersurface to entirely separate all points of different classes.

These hypersurfaces, characterized by different functional forms, are termed *kernels*.

The most basic SVM classifier uses a linear kernel, which is simply a flat hyperplane, to divide the space into two sections for binary classification purposes.

The ideal separation is achieved by the *Maximal Margin hypersurface* or *Optimal Separating hypersurface*, which is positioned as far away as possible from the nearest data points of different classes.

2.5.1 Details About SVM Formalism

In a basic scenario where an SVM with a linear kernel is used for binary classification, it is presumed that the dataset can be perfectly divided in the hyperspace, leading to an pure separation into subspaces, with each containing only one class. This model is referred to as the *Maximal Margin Classifier*, serving as the foundation for the SVM technique, which extends this original concept.

The use of a linear kernel suggests that the *Maximal Margin hypersurface* will manifest as an $(F-1)$-dimensional hyperplane, determined by $F+1$ parameters β_f. For any given point x within the feature space, expressed as an F-dimensional vector $\vec{x}$, this hyperplane

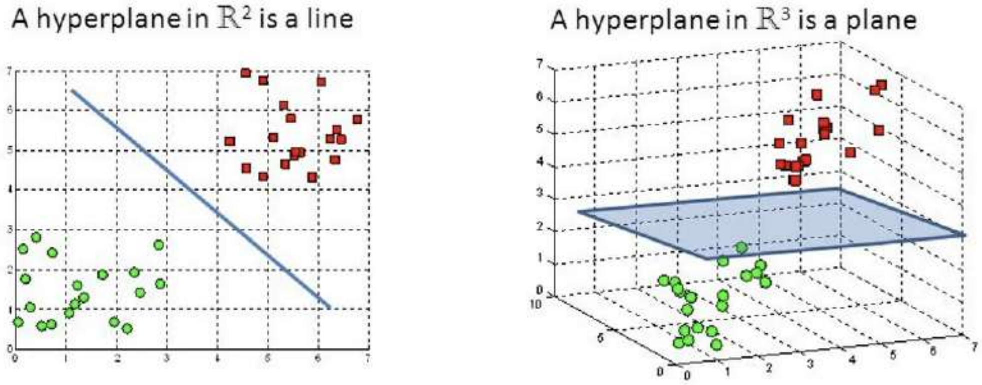


Figure 2.5.1: Representation of the *Maximal Margin hypersurface* in a $\mathbb{R}^2$ and $\mathbb{R}^3$.

is mathematically characterized as follows:

$$\sum_{i=1}^{F} \beta_i x_i + \beta_0 = \vec{\beta} \cdot \vec{x} + \beta_0 = 0 \tag{2.1}$$

where $\vec{\beta}$ is a F-dimensional vector of *weights*.

For each point $\vec{x}$, three scenarios are possible:

- The point lies on the hyperplane, satisfying the Equation 2.1;
- The point is "above" the hyperplane, indicated by: $\vec{\beta} \cdot \vec{x} + \beta_0 > 0$;
- The point is "below" the hyperplane, indicated by: $\vec{\beta} \cdot \vec{x} + \beta_0 < 0$;

Ideally, points above and below the hyperplane belong to different classes.

Labeling $\vec{x}$ with $y = \pm 1$, the ideal *Maximal Margin hyperplane* must fulfill:

$$\begin{cases} \vec{\beta} \cdot \vec{x} + \beta_0 > 0, \text{ if } y = +1 \\ \vec{\beta} \cdot \vec{x} + \beta_0 < 0, \text{ if } y = -1 \end{cases} \tag{2.2}$$

Simplified as:

$$y \left(\vec{\beta} \cdot \vec{x} + \beta_0 \right) > 0 \tag{2.3}$$

It is noteworthy that if the dataset is perfectly separable by a hyperplane, an infinite number of such planes exist, achieved by minor translations or rotations. Thus, the above condition, while necessary, is insufficient for defining a unique *Optimal Separating hyperplane.* . To identify this plane, an assessment of the distance between the plane and the observations in the training dataset is required. The smallest of these distances, known as the *margin*, is maximized by the *Maximal Margin hyperplane.*

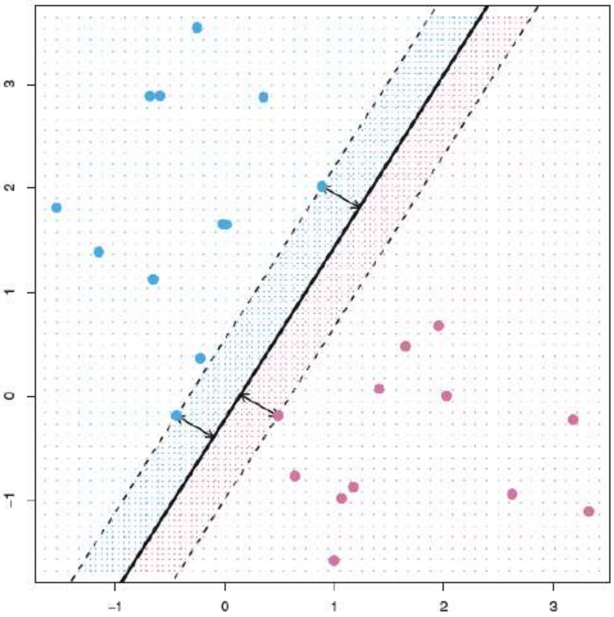


Figure 2.5.2: The maximal margin hyperplane is shown as a solid line. The margin is the distance from the solid line to either of the dashed lines. The two blue points and the purple point that lie on the dashed lines are the support vectors.

The *margin* is determined by specific points, termed support vectors, which are crucial for defining the hyperplane parameters.

A two dimensional example can be seen in the Figure 2.5.2.

2.5.2 Evaluate the Maximal Margin Hyperplane

Delving deeper into the construction of the *Optimal Separating Hyperplane*, consider a set of N training observations $x_1, \ldots, x_N \in \mathbb{R}^F$ with corresponding labels $y_1, \ldots, y_N \in \{-1, +1\}$. The *Maximal Margin Hyperplane* is determined by solving an optimization problem that aims to maximize the margin M, which is characterized by the $(F+1)$ β values that satisfy:

$$\max_{\beta_0, \ldots, \beta_F} M \tag{2.1}$$

subject to the conditions:

$$\begin{cases} \sum_{j=1}^{F} \beta_j x_{i,j} + \beta_0 > M, \ \forall i = 1, \ldots, N \\ \sum_{j=0}^{F} \beta_j^2 = 1 \end{cases} \tag{2.2}$$

2.5.3 Non-Separable Dataset

The previous formulation presupposes that the dataset is perfectly separable into two distinct regions by a hyperplane. However, achieving perfect separation is often unrealistic.

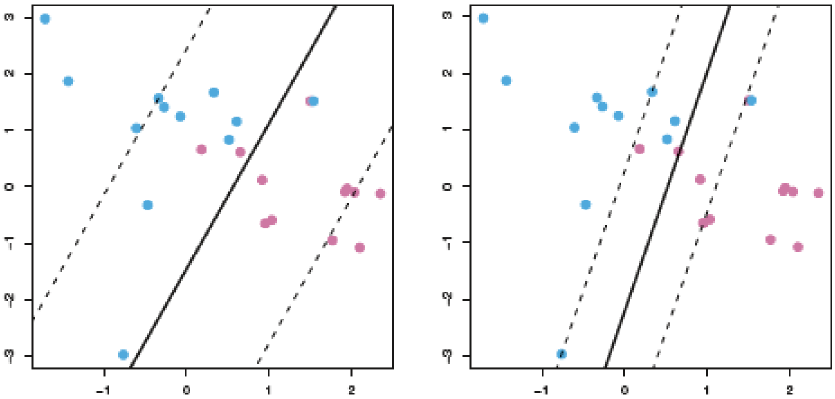


Figure 2.5.3: Optimal Separating Hyperplane: higher C value with larger margin on the left, lower C value with smaller margin on the right.

In addition, a hyperplane that separates perfectly the training dataset is not always the goal. This could in fact bring to an overfitted model, too sensitive to the single dataset observation.

Hence, the SVM aim shifts towards a hyperplane that more effectively classifies the majority of the training dataset, demonstrating greater resilience to individual data points. This adjustment allows for some observations to be on the incorrect side of the hyperplane or closer than the margin. Consequently, SVMs with this flexibility are known as *soft margin classifiers*. The problem described by the Equations 2.1 and 2.1 becomes:

$$\max_{\substack{\beta_0,\ldots,\beta_F\\ \epsilon_1,\ldots,\epsilon_N}} M \tag{2.1}$$

with the new set of conditions:

$$\begin{cases} \sum_{j=1}^{F} \beta_j x_{i,j} + \beta_0 > M(1-\epsilon_i), \ \forall i = 1,\ldots,N \\ \sum_{j=0}^{F} \beta_j^2 = 1 \\ \epsilon_i \geq 0 \ \forall i = 1,\ldots,N \\ \sum_{i=1}^{N} \epsilon_i \leq C \end{cases} \tag{2.2}$$

This formulation introduces a layer of complexity with C as a non-negative parameter, M indicating the margin width, and $\epsilon_1, \ldots, \epsilon_N$ as *slack variables*. For each observation $\epsilon_i = 0$ signifies correct margin placement, $0 > \epsilon_i > 1$ indicates incorrect margin side but correct hyperplane side, and $\epsilon_i > 1$ denotes incorrect hyperplane side. Therefore, C quantifies the tolerable violations within the training dataset. A larger C value permits more margin violations, affecting the margin width as illustrated in the referenced figure. Examples of varying results with different values of C are illustrated in the Figure 2.5.3.

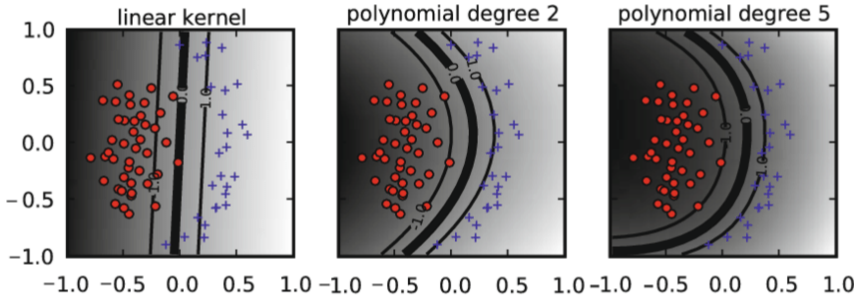


Figure 2.5.4: An Optimal Separating Hyperplane with different kernel functional form.

2.5.4 Different Kernels

So far, we have discussed only the simplest form of the hypersurface, the linear hyperplane. Depending on the dataset, various functional forms can be adopted for the hypersurface. The underlying concept is to use an operation other than the classical linear product to define the distance between vectors. For instance, for two F-dimensional vectors $\vec{a}$ and $\vec{b}$, the *polynomial kernel* of degree d, where $d \in \mathbb{N}$, can be expressed as:

$$K(a,b) = \left(1 + \sum_{j=1}^{F} a_j b_j\right)^d \tag{2.1}$$

Another widely used kernel is the *radial kernel*, defined by:

$$K(a,b) = exp\left(-\gamma \sum_{j=1}^{F} a_j b_j\right)^2 \tag{2.2}$$

Once the kernel (i.e., the functional form of the hypersurface) is determined, the solution involves evaluating the entire training dataset $\mathcal{S}$ using the equation:

$$f(x) = \beta_0 + \sum_{x_i \in \mathcal{S}} \alpha_i K(x, x_i) \tag{2.3}$$

The choice of kernel and the parameters influenced by the data points, β_0 and the set of N α, introduce non-linearity into the model. Examples of SVM with different kernels are depicted in the Figure 2.5.4.

2.6 Tree-Based Classification

Tree-based classification involves dividing the feature space into simpler regions and constructing a decision tree through binary splits for each feature. This process aims to create regions ideally containing only a single class of points. Consider a dataset $\mathcal{D}$ where observations are points $x_i \in \mathcal{R}^F$, each described by a class y_i. To construct a tree, a threshold is selected for each feature to partition the feature space into distinct, non-overlapping regions (T regions). Each region is characterized by the *most commonly occurring class* y_i. In an ideal scenario, each region would be *pure*, meaning all points within a region share the same label, though achieving such purity is not always possible.

Despite the complexity implied by this description, the process itself is quite straightforward. An example of an ideal case, illustrated in the Figure 2.6.1, helps clarify how a tree is constructed and how it operates.

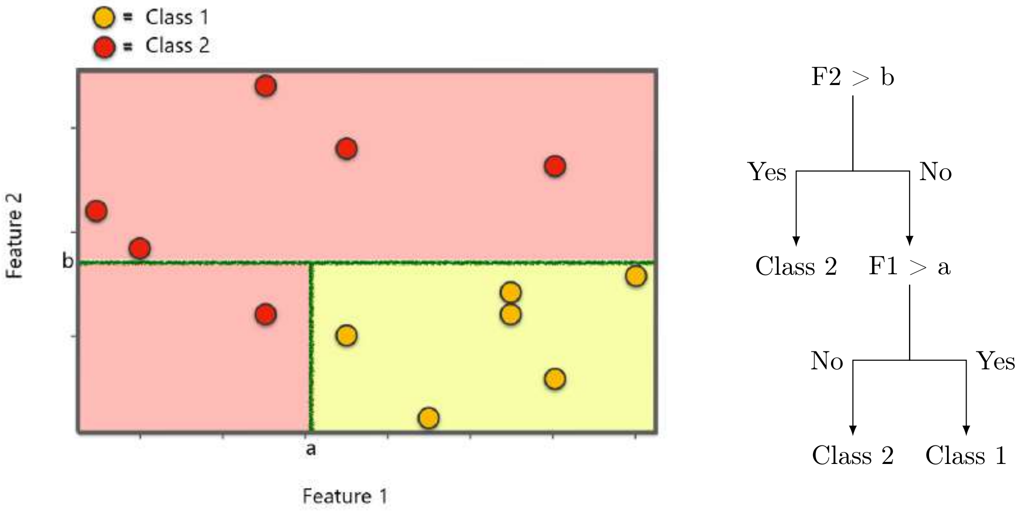


Figure 2.6.1: On the left the representation of the region on the feature space created by the algorithm, on the right a graphical representation of the decisional tree.

In certain scenarios, it is impractical to precisely partition the feature space into distinct classes using straightforward logic. Consequently, a method to evaluate various possibilities becomes essential. The process of tree construction necessitates making key decisions, such as determining the sequence in which features are considered and selecting appropriate thresholds for each feature. These choices are critical for effectively segmenting the feature space and enhancing the classification accuracy of the tree.

2.6.1 General Structure of a Tree

Tree classification algorithms employ a hierarchical node structure, as illustrated in Figure 2.6.2. This structure spans from the root to the leaf nodes, efficiently grouping the data into increasingly homogenous categories. The decision-making process starts at the root node, which selects the attribute offering the highest information gain for the first division. As the process descends, decision nodes refine this division based on specific criteria, leading to leaf nodes. These nodes represent decision outcomes and carry associated class labels, enabling

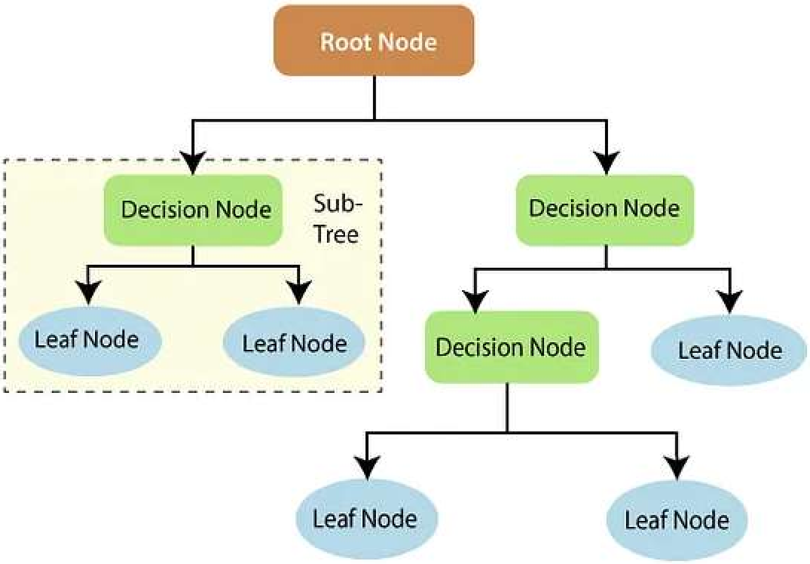


Figure 2.6.2: A general scheme of a tree with the name of the main components.

effective data classification via binary decisions from root to relevant leaf.

These models are appreciated for their explainability. Smaller trees provide straightforward graphical representations, underscoring key features without the need for deep mathematical knowledge. Nonetheless, tree-based methods can be sensitive; slight data variations may significantly alter the structure of the tree, making the method predictability dependent on dataset specifics.

2.6.2 Evaluation of the Tree

Different situations yield various trees. The "goodness" of each region created by a tree can be assessed through the classification error rate of the training set, defined as the fraction of observations not correctly classified within a region. However, other parameters provide a more detailed tree description.

A commonly used metric is the *Gini index* (or Gini impurity). Considering a classification problem with C classes, the impurity of each leaf node l can be calculated as:

$$I_G(l) = \sum_{c=1}^{C} p_{l,c}\left(1 - p_{l,c}\right) = 1 - \sum_{c=1}^{C} p_{l,c}^2 \tag{2.1}$$

where $p_{l,c}$ is the proportion of training observations in the l-th region belonging to the c-th class. This index approaches zero when class frequencies near zero or one, indicating a well-divided feature space.

2.6.3 Variation of Simple Trees

Simple trees can serve as building blocks for more complex models, combining multiple trees to enhance performance while reducing intuitiveness and explainability. *Boosting* is a popular method for aggregating trees, training them sequentially to amend the errors of their predecessors. This method involves training decision tree models in a sequential manner, where each tree attempts to correct the errors of the previous ones in the series. The process starts with the training of a decision tree on the initial dataset. The initial tree trains on the dataset, and each subsequent tree adjusts to the errors of the last by reweighting the data: instances incorrectly predicted by previous trees gain weight, while correctly predicted instances lose weight. This approach ensures that each new tree focuses more on the difficult cases. The final prediction usually results from a majority vote among all trees. Critical parameters, like the learning rate, tree depth, and the number of trees, must be carefully managed to prevent overfitting, a significant risk given the algorithm structure.

2.7 K-Means Clustering

2.7.1 General Overview

The clustering analysis emerges as a valuable approach for the analysis and evaluation of the phasor plane, constituting a pivotal method in the exploration of diverse datasets. Clustering, within the context of data analysis, encompasses a suite of techniques aimed at identifying inherent subgroups or clusters within a given dataset. This analytical process entails an examination of the degree of similarity or dissimilarity among the various elements comprising the dataset.

One preeminent clustering technique is K-Means Clustering, a well-established method in unsupervised learning. In this approach, the input comprises a set of features represented as $(X_1, X_2, \ldots, X_p)$, along with a predefined number of clusters (defined a-priori). The algorithm then allocates each element to its respective subgroup based on the specified cluster parameters. It is noteworthy that the primary objective of such clustering methods is to discern the presence of subgroups within the dataset, rather than establishing associations between features and a known response variable (Y value). This is particularly challenging due to the absence of a ground truth for comparison.

Unsupervised methods, such as K-Means Clustering, present inherent challenges, largely stemming from the absence of labeled data for training. The complexity is compounded by the absence of a target variable that allows for direct validation of results against real-world outcomes. Notably, in this work the difficulties associated with evaluating the quality of results are mitigated by our knowledge of the actual class membership of the dataset. This unique circumstance provides a rare advantage, as the ground truth information allows for a more meaningful assessment of the clustering outcomes.

2.7.2 Formal Approach

Before continuing, it is necessary to define some terms. The term n refers to the number of observations, with each observation characterized by p values, which are formally defined as *features*. For instance, we could represent n points in a $\mathbb{R}^p$ space. The term K represents the number of clusters resulting from the K-Means evaluation, resulting in a set of clusters $C_1, \ldots, C_K$ such that:

- Every observation is included in one cluster, with no exclusions, represented as $C_1 \cup C_2 \cup \ldots \cup C_K = 1, \ldots, n$.
- Clusters do not overlap, meaning an observation belongs to only one cluster, which is indicated by $\forall i, j \in K,\ C_i \cap C_j = \emptyset$.

The cluster set is formed by minimizing a function known as *within-cluster variation* $W(C_k)$. Therefore, the problem is described by:

$$\min_{C_1,\ldots,C_K} \left[\sum_{i=1}^{K} W(C_i) \right] \tag{2.1}$$

There are several ways to define the *within-cluster variation* function.A commonly used method, and the default in MATLAB codes[1] is the *squared Euclidean distance*, expressed as:

$$W(C_k) = \frac{1}{|C_k|} \sum_{i,l \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{lj})^2 \tag{2.2}$$

Here, $|C_k|$ represents the number of elements in the k-th cluster, and x_{ij} denotes the j-th feature of the i-th element in cluster C_k. This function calculates the sum of the squared distances between the features of each pair of elements in the cluster, normalized by the cluster size.

Solving this minimization problem is complex, requiring an algorithm to find a local optimal solution. The most common approach is outlined as follows:

1. Choose K arbitrary initial centroids $\{c_1, \ldots, c_K\}$ that represent the clusters;
2. $\forall i \in \{1, \ldots, n\}$ each observation in x_i it will be in the cluster which has the nearest centroid c_k with distance evaluated according to the *within-cluster variation* function $W(C_k)$ established.

[1] as indicated by `https://it.mathworks.com/help/stats/kmeans.html`

3. $\forall k \in \{1, \ldots, K\}$ it is evaluated the new centroid value c_k of each cluster C_K being the center of mass of al the points in the cluster:

$$c_k = \frac{1}{|C_k|} \sum_{x \in C_k} x \tag{2.3}$$

4. The steps 2 and 3 are repeated each time with different starting centroids given by the previous step. This happens until all the clusters C_k no longer changes.

Typically, initial centroids are chosen uniformly among the observations, but this is not the sole method. For instance, in MATLAB *kmeans* function, the default method for selecting starting centroids is known as *k-means++*. This method involves defining $D(x)$ as the distance between an observation x and the nearest centroid. Consequently, the initial selection of centroids in the k-means algorithm involves a specific process:

- **1a)** Randomly select a single center c_1 from all observations with uniform probability;
- **1b)** Choose the next centroid $c_k = x'$, where x' is an observation not yet selected as a centroid. The selection probability for each point is

$$p(x') = \frac{D(x')^2}{\sum_{x \in X} D(x)^2} \tag{2.4}$$

- **1c)** Repeat step *(1b)* until all K centroids have been selected.

It is essential for the algorithm to ensure that each iteration of centroid updating reduces the sum described by Equation 2.1, aiming for a lower result than the previous step to reach a local minimum.

Consider the identity:

$$\frac{1}{|C_k|} \sum_{i,l \in C_k} \sum_{j=1}^{p} (x_{ij} - x_{lj})^2 = \frac{1}{|C_k|} \sum_{i \in C_k} \sum_{j=1}^{p} (x_{ij} - c_{kj})^2 \tag{2.5}$$

where c_{kj} is the j-th feature value of the k-th centroid, as calculated in Equation 2.3. The algorithm aims to find the mean value in the cluster (that is the centroid) which minimize the right-hand term of the equation. Consequently, the left part of the equation is also reduced. The algorithm concludes when there are no more changes, indicating a local minimum has been found.

Since the algorithm might reach a local minimum, it may be necessary to run the K-Means Clustering method multiple times and select the outcome where Equation 2.1 is minimized. Assuming m trials yield minimization problem results $\{R_1, \ldots, R_m\}$, each associated with a set of K clusters and centroids, the final set $\{C_{1,fin}, \ldots, C_{K,fin}\}$ is determined by selecting

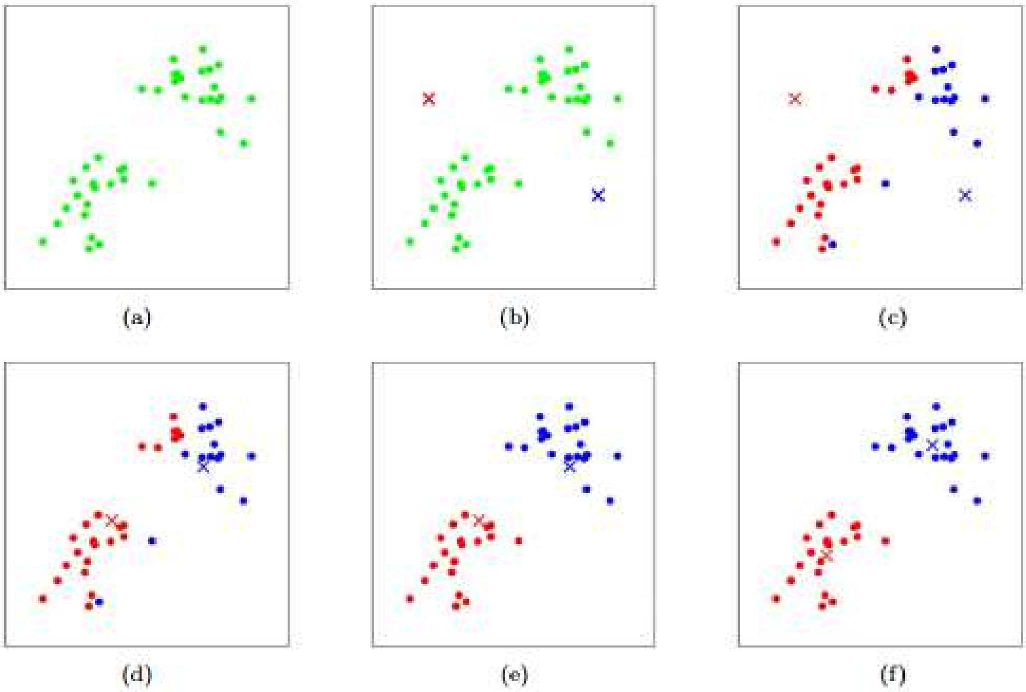


Figure 2.7.1: A figure representing how centroids evolve in a K-Means Clustering algorithm during the various steps.

the set with result R_{fin} which satisfies:

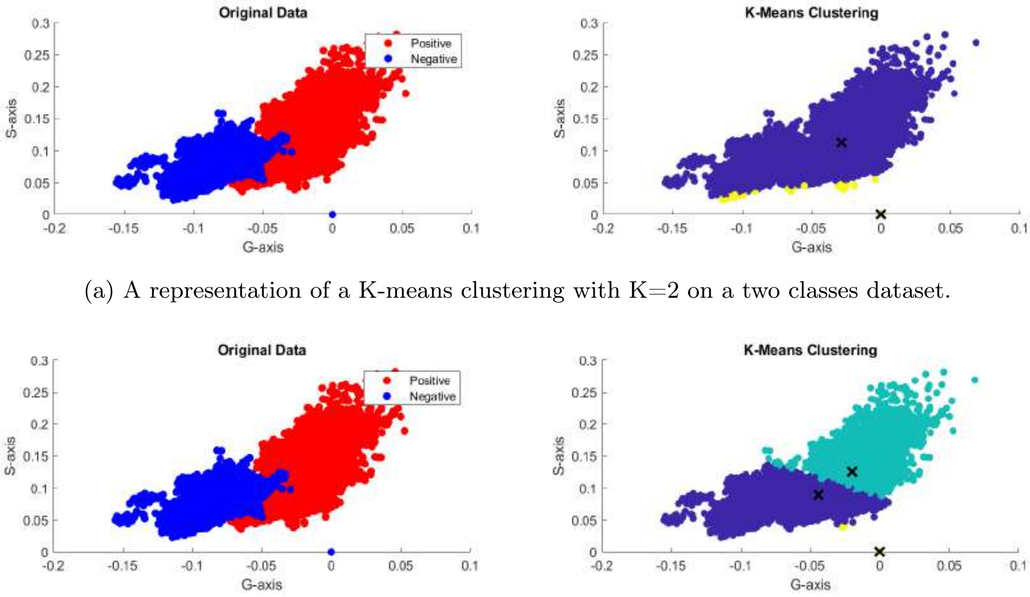
$$R_{fin} = \min_{\forall i \in (1,\dots,m)} [R_i] \tag{2.6}$$

An illustration of improvements during various steps of the algorithm is shown in Figure 2.7.1, demonstrating how centroids evolve in a K-Means Clustering algorithm.

2.7.3 Limits and Issues of the Method

One of the limitations of clustering methods is given by its foundations: the fact that each observation must belong to a cluster, and no observation can be excluded, sometimes creates some "useless subgroups". For example, if a few points are quite different from the others, some more relevant clusters could be ignored because of such outliers.

In addition to this, it is important to mention that these methods are particularly sensitive to small variations in the observations. For example, by deleting a few observations, the cluster created could be completely different from the previous ones. Examples of these issues can be found in Figure 2.7.2.



(a) A representation of a K-means clustering with K=2 on a two classes dataset.

(b) The same representation but K=3.

Figure 2.7.2: As we can see the k-mean clustering is sensitive to isolated elements. In this case despite the original dataset is composed by two classes a K=3 clustering give better results. This occurs because, for example, the isolated points are background residual of the signal represented in the plot.

Chapter 3. Theoretical Aspects of Analytical Methods and Algorithms

3.1 Hyperspectral Imaging (HSI) and Analysis

Hyperspectral Imaging (HSI) is a sophisticated imaging technique that captures a three-dimensional data array, known as a hypercube. This array allows each pixel in an image to contain a full spectrum of information across multiple wavelengths. This capability facilitates non-destructive imaging that enhances spatial data with spectral information traditionally obtained through spectroscopy. HSI offers profound insights into material composition and characteristics, proving invaluable in remote sensing, medical imaging, and environmental monitoring.

HSI stands apart from conventional RGB imaging and Multispectral Imaging (MSI) in the density of its spectral information, measured by the number of wavelengths per nanometer. While RGB imaging captures only three wavelengths, reflecting the human visual spectrum, MSI and HSI encompass a broader wavelength range. However, HSI provides more continuous spectral information, with intervals between wavelengths typically less than 1 nm, compared to MSI broader 10 nm or more intervals.

In both HSI and MSI, as well as in RGB imaging, each image pixel corresponds to a spectrum of intensities, constituting a hypercube. Consider an image described by an $M \times N$ matrix. The hypercube dataset can thus be depicted as a three-dimensional matrix $M \times N \times L$. In RGB imaging, $L = 3$, while in MSI and HSI, L exceeds this, reflecting a greater spectral range.

Each pixel is experimentally characterized by an intensity $I(x, y, \lambda)$, with $x \in (x_1, \ldots, x_M)$, $y \in (y_1, \ldots, y_N)$ and $\lambda \in (x\lambda_1, \ldots, \lambda_L)$, as illustrated in Figure 3.1.1.

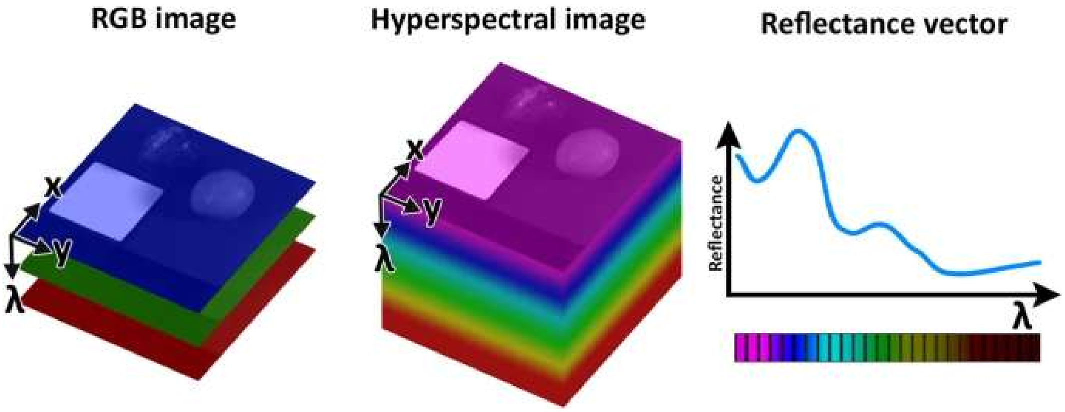


Figure 3.1.1: A figure representing HSI cube and pixel spectrum.

3.2 Phasor Spectral Analysis

3.2.1 General Overview

Imaging technologies have advanced rapidly, now offering detailed nanometer resolution and quantitative insights into biomolecular dynamics in vivo. We previously discussed one of the latest innovative techniques in Section 3.1. However, the adoption of such technologies is often limited by complex hardware requirements and the need for extensive post-analysis.

Phasor spectral analysis presents itself as an intuitive, model-free method for information extraction, simplifying the data representation process. It provides a straightforward 2D visualization of datasets, which is beneficial for preliminary assessments or can be integrated with cluster analysis methods, such as the K-Means Clustering.

Applying this approach to Hyperspectral Imaging (HSI) enables the mapping of complex spectra to a two-dimensional plot, known as a *phasor plot*, through a pair of Fourier sine and cosine transforms. In this plot, each pixel p_i, defined by a set of n_λ wavelength intensities, is represented on an imaginary plane. The angular position (phase) correlates with the intensity distribution across various wavelengths, while the radial distance from the center (modulation) reflects the spectrum width.

A graphical representation of this concept is depicted in Figure 3.2.1.

3.2.2 Formal Approach

Different methods and approaches exist for transforming spectra. One innovative multi-parameter method, the $i - \phi - MaLe$ *approach*, has shown significant promise in analyzing biological reflectance spectra to derive information about various biological parameters.

In this method, the spectrum is modeled as a function $S(\lambda)$. From a theoretical standpoint, if we define k as the harmonic number of the transformation, the coordinates on the

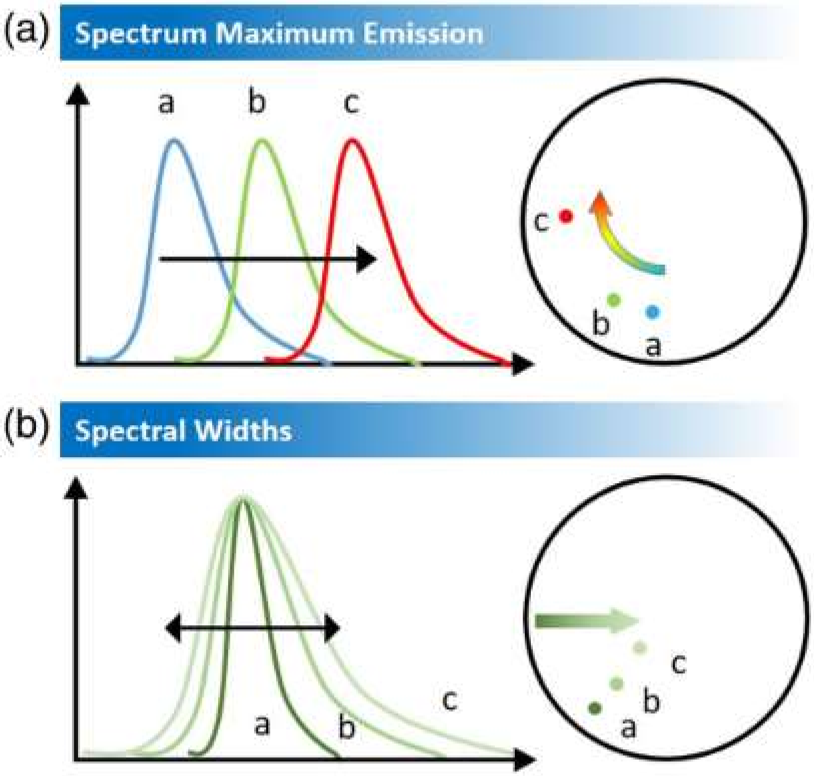


Figure 3.2.1: A representation of the phasor analysis transformation of a spectra to the *phasor plane*.

2D phasor plane for the pixel p_i described by the spectrum $f_i(\lambda)$ are determined as follows:

$$g_i(k) = \frac{\int_{-\infty}^{\infty} f_i(\lambda)\cos(2\pi k\lambda)d\lambda}{\int_{-\infty}^{\infty} f_i(\lambda)d\lambda} \qquad s_i(k) = \frac{\int_{-\infty}^{\infty} f_i(\lambda)\sin(2\pi k\lambda)d\lambda}{\int_{-\infty}^{\infty} f_i(\lambda)d\lambda}$$

Experimentally, the spectrum $f_i(\lambda)$ is depicted through a non-continuous array of acquired values. Consequently, the Equation 3.2.2 is expressed as a summation. Assuming the presence of a hyperspectral image characterized by $M \times N$ spatial pixels, each associated with spectral values $\lambda_1, \ldots, \lambda_L$, every pixel p_i within the image is mapped onto a 2D phasor plane using the discrete Fourier Transform (DFT), as demonstrated by:

$$g_i(k) = \frac{\sum_{\lambda=\lambda_1}^{\lambda=\lambda_L} f_i(\lambda)\cos(2\pi k\lambda)d\lambda}{\sum_{\lambda=\lambda_1}^{\lambda=\lambda_L} f_i(\lambda)d\lambda} \qquad s_i(k) = \frac{\sum_{\lambda=\lambda_1}^{\lambda=\lambda_L} f_i(\lambda)\sin(2\pi k\lambda)d\lambda}{\sum_{\lambda=\lambda_1}^{\lambda=\lambda_L} f_i(\lambda)d\lambda}$$

3.2.3 Parameters of Interest

Upon obtaining a 2D set of points representing the pixels $\{p_1, \ldots, p_{M\times N}\}$, we can estimate the average position of these pixels on the phasor plane. This process yields what is known as the *centroid* of the sample. Suppose we have a collection of Ω images $\{I_1, \ldots, I_\Omega\}$, with each image I_j comprising P_j spatial pixels, thereby contributing P_j points to the phasor plane. For each image I_j, the corresponding centroid $C_j(k)$ is identified by 2D coordinates $(G_j(k), S_j(k))$, calculated as:

$$G_j(k) = \frac{\sum_{p_i \in P_j} g_{ij}(k)}{P_j} \qquad S_j(k) = \frac{\sum_{p_i \in P_j} s_{ij}(k)}{P_j} \tag{3.1}$$

The centroid of the j-th mage denotes the average position of its pixels on the phasor plane. By considering this average position as the "true position" of the pixels, the error associated with the centroid can be assessed through the standard deviation of the mean. The standard error of the mean (SEM) measures the expected difference between the sample mean (in this case, the centroids) and the true population mean. This is different from the standard deviation (SD), which assesses the spread or variability of individual data points around the mean. Consequently, each centroid $C_j(k)$ is described by a mean value and an associated error, calculated as:

$$SEM(G_j(k)) = \frac{\sigma G_j(k)}{P_j} \qquad SEM(S_j(k)) = \frac{\sigma S_j(k)}{P_j} \tag{3.2}$$

Where $\sigma G_j(k)$ and $\sigma S_j(k)$ are defined by:

$$\sigma G_j(k) = \sqrt{\frac{\sum_{p_i \in P_j} (g_{ij} - G_j(k))^2}{P_j - 1}} \qquad \sigma S_j(k) = \sqrt{\frac{\sum_{p_i \in P_j} (s_{ij} - S_j(k))^2}{P_j - 1}} \tag{3.3}$$

3.3 RGB Analysis

References to the complexity of human visual perception and the process of converting hyperspectral images (HSI) to RGB are discussed in this section. RGB representations create a 3D hypermatrix, allowing users to view samples similarly to direct observation. However, this perception is not always accurate, necessitating an exploration of how our eyes physically perceive objects and the translation of HSI data into RGB images.

3.3.1 Colour Human Perception

The perception of color involves a series of steps, as outlined in Figure 3.3.1. Without delving into each step's specifics, it is crucial to understand that color perception in the optical cortex is influenced by the optical properties of the light source, the object, and its background. Before light reaches the optical cortex for processing, it passes through the photoreceptors of the eye: rod cells and cone cells.

In essence, rods and cones serve different functions based on ambient light levels. Rods are effective in low light conditions (less than $1cdm^{-2}$), while cones operate in brighter environments, with a transitional phase allowing for vision across a broad luminance range. At high luminance levels (greater than $100cdm^{-2}$), only cones are active, as rods become saturated. Scotopic vision, dominated by rods, occurs in low light; photopic vision, reliant on cones, in high light; and mesopic vision, involving both, in intermediate light levels.

Furthermore, rods and cones differ in spectral sensitivity. Rods, of which there is only one type, peak at around 510 nm. Cones are categorized into three types—L (long-wavelength), M (middle-wavelength), and S (short-wavelength) cones—offering a more precise classification than the simplistic RGB model. These cones' spectral responsivities significantly overlap, as illustrated in Figure 3.3.2.

This figure displays the normalized physical absorption spectra for each cone type. When analyzing an object with a given spectral distribution $O(\lambda)$ and describing cone responses with $l(\lambda), m(\lambda)\ and\ s(\lambda)$, the responsivity is quantified by three values:

$$\begin{aligned} L &= \int_0^\infty O(\lambda)l(\lambda)d\lambda \\ M &= \int_0^\infty O(\lambda)m(\lambda)d\lambda \\ S &= \int_0^\infty O(\lambda)s(\lambda)d\lambda \end{aligned} \tag{3.1}$$

This contrasts with the color separation responsivities typically integrated into physical imaging systems. Incorporating these sensitivities into imaging devices presents challenges in achieving precise color reproduction.

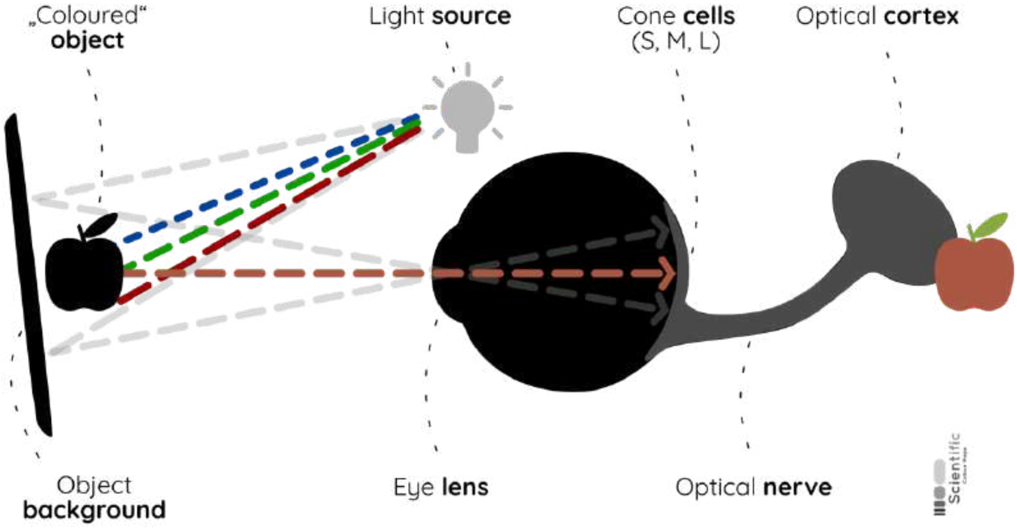


Figure 3.3.1: Simplified schematic of human colour perception. Source: This graphic by Fabio Crameri from Crameri et al. (2020) is available via the open-access s-Ink repository.

3.3.2 Generating RGB Images from Hyperspectral Images

To generate an RGB image, which is a $M \times N \times 3$ matrix from a hyperspectral image, we select three specific wavelengths of interest from the hyperspectral image Λ wavelengths. This approach simplifies the image representation to three peak functions, forming the foundation of the RGB color space:

$$R = \begin{pmatrix} r \\ 0 \\ 0 \end{pmatrix} \qquad G = \begin{pmatrix} 0 \\ g \\ 0 \end{pmatrix} \qquad B = \begin{pmatrix} 0 \\ 0 \\ b \end{pmatrix} \tag{3.1}$$

This method, while not perfectly mirroring real-world perception, proves particularly useful for depicting plants. The chlorophyll pigment predominantly influences the green spectral peak, allowing visual differentiation with relative ease. However, it is important to note that there are various ways to determine RGB values, affecting the final image representation.

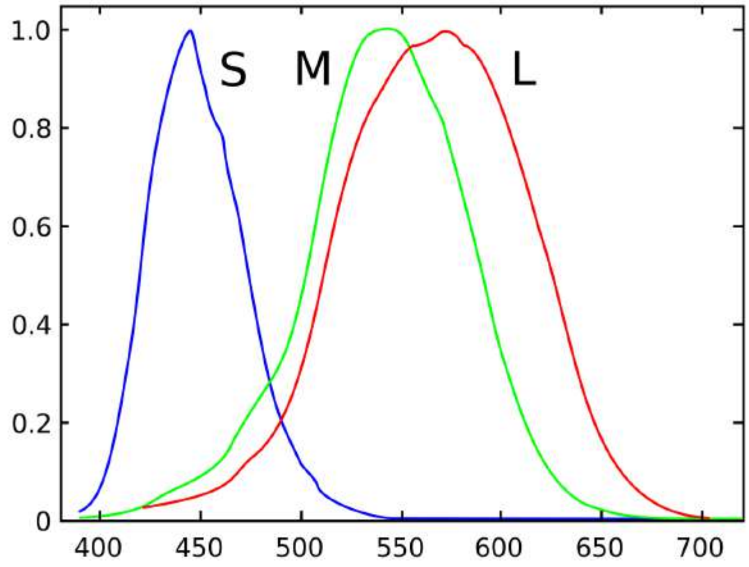


Figure 3.3.2: Normalized responsivity spectra of human cone cells, S, M, and L types.

3.4 Texture Analysis

In everyday life the texture is strictly related to the touch and described with words like rough and smooth. In general by texture we refer to some characteristics of the surface of objects and samples. By a mathematical point of view, texture analysis strongly depends on the spatial relationship among gray levels of pixels intensities. References to texture analysis are crucial in various fields such as remote sensing, and medical imaging, where the surface characteristics of objects are of interest.

Such analyses are fundamental in identifying similarities and differences between neighboring pixels, enabling the delineation of regions of interest through comparison. Texture metrics play a significant role in image segmentation, feature extraction, and classification, employing various approaches for the description and extraction of relevant texture features. Predominantly, these approaches are categorized into statistical and structural methods.

Statistical methods analyze the distribution and relationships of pixel intensities, employing mathematical and statistical techniques to characterize textures. These might encompass the evaluation of autocorrelation functions, co-occurrence matrices, or conducting spectral analysis to detect patterns in pixel values that define an image texture.

Conversely, structural methods view texture as comprising repetitive units or patterns, termed primitives, focusing on their placement within the image. This approach identifies specific shapes or motifs arranged systematically to create the texture, analyzing these elements and the rules governing their spatial organization.

Each approach has its merits and limitations, with statistical methods often yielding more meaningful results in analyzing micro-textures and irregular patterns. Thus, the discussion will center on these analyses.

Describing texture features involves various spatial parameters, such as the size of the neighborhood for a texture region or the quantization of gray levels. Specifically, in hyperspectral imaging (HSI) applications, selecting the appropriate wavelength is crucial, given that texture analysis is fundamentally spatial.

3.4.1 Gray-Level Co-Occurrence Matrix Method

The Gray-Level Co-Occurrence Matrix (GLCM) stands as one of the pioneering statistical techniques in image texture analysis. A Co-Occurrence Matrix measures the frequency of paired elements (like words, symbols, or values) occurring together within a specified context or dataset. The GLCM applies this concept to quantify the frequency of different gray-level combinations (pixel brightness values) in an image based on a defined spatial relationship, facilitating the extraction of statistical measures to describe the texture characteristics of the image.

To elaborate, consider an image represented as a grayscale $M \times N$ matrix, where the value of each point reflects the pixel intensity p_{mn}.

Define G as the number of intensity levels, with each pixel intensity ranging from 1 to G. A spatial separation δ specifies the pixel pair distance (e.g. $\delta = 1$ for immediately adjacent pixels), and an angle α (e.g. $\alpha = 45\,\text{deg}$ for diagonal comparison) determines their directional relationship. The GLCM is then a $G \times G$ matrix, where each element g_{ij} counts the occurrences of neighboring pixel pairs with intensity levels i and j, calculated as:

$$g_{ij}(\delta, \alpha) = \sum_{m=1}^{M} \sum_{n=1}^{N} \begin{cases} 1, & \text{if } p_{mn} = i \text{ and } p_{m+\Delta m, n+\Delta n} = j \\ 0, & \text{otherwise} \end{cases} \tag{3.1}$$

Here, Δm and Δn represent the offsets determined by the chosen (δ, α) pair. Normalizing the GLCM can be beneficial for feature extraction, resulting in each value being expressed as:

$$g_{NOR,ij} = \frac{g_{ij}}{\sum_{i=1}^{G} \sum_{j=1}^{G} g_{ij}} \tag{3.2}$$

3.4.2 Texture Features Derived by the GLCM

The Gray-Level Co-Occurrence Matrix (GLCM) allows for the extraction of several texture features that provide insights into the texture of the original image. Notably, these include Contrast, Correlation, Energy, Entropy, and Homogeneity.

The *Contrast*[1] quantifies the intensity variation between neighboring pixels across the

[1] also known in literature as variance or inertia

image, emphasizing areas of strong local variation. It is calculated as:

$$Contrast = \sum_{i,j=1}^{G} (i-j)^2 g_{NOR,ij} \tag{3.1}$$

A constant image, lacking variation, yields a Contrast of 0.

The *Correlation* measures the linear predictability between neighboring pixels, indicating how pixel intensities are linearly related across the image. Defined by:

$$Correlation = \frac{\sum_{i,j=1}^{G}(i-\mu_i)(j-\mu_j)g_{NOR,ij}}{\sigma_i\sigma_j} \tag{3.2}$$

Here, μ_i,μ_j are the row and column means of the GLCM, and σ,σ_j are the respective standard deviations.

The *Energy*[2] reflects the uniformity or concentration of GLCM element distributions. High Energy values indicate a concentrated distribution. It is defined as:

$$Energy = \sum_{i,j=1}^{G} g_{NOR,ij}^2 \tag{3.3}$$

Energy equals 1 for a constant image.

The *Entropy* assesses the randomness or complexity within the GLCM, calculated via:

$$Energy = -\sum_{i,j=1}^{G} g_{NOR,ij} \log(g_{NOR,ij}) \tag{3.4}$$

High Entropy values denote a high level of disorder or complexity.

The *Homogeneity* evaluates how closely the GLCM elements are distributed to the diagonal of the matrix, focusing on the similarity of pixel pair intensities. Defined as:

$$Energy = \sum_{i,j=1}^{G} \frac{g_{NOR,ij}}{1+|i-j|} \tag{3.5}$$

Homogeneity reaches 1 in a diagonal GLCM, indicating maximum similarity.

[2] also known in literature as uniformity or angular second moment

Chapter 4. Experimental Data Acquisition and Analysis

4.1 Sample Extraction

The wheat spikes exploited in the experiment were grown near San Piero a Grado by the University of Pisa. The experimental field was organized into various parcels, each assigned a unique identification code to detail the wheat variety, treatment applied, and the presence or absence of pathology, encompassing a total of 12 distinct scenarios.

The research focused on examining Fusarium Head Blight (FHB) in two specific wheat varieties: *Triticum aestivum L. cultivar Bingo*, which is more susceptible to the disease, and *Triticum aestivum L. cultivar Rebelde*, known for its resistance. Each wheat variety was segregated into six groups: three healthy and three infected with spores of *Fusarium graminearum* and *F. langsethiae*. Within these groups, one subset remained untreated, while the others were subject to one of two treatments. The conventional treatment involved Binal Pro, a systemic fungicide aimed at controlling the disease.

An experimental treatment involved the application of *Trichoderma* before the inoculation of the pathogen. This method focuses on fostering the growth of fungi antagonistic to FHB on the wheat spikes, utilizing their natural production of hydrolytic enzymes and antimicrobial compounds to compete for nutrients and space, thereby inhibiting the disease progression. Such fungi not only help in controlling phytopathogens but also support plant growth, forming a key component of many biopesticides and biofertilizers currently available.

4.2 Hyperspectral System of Measurement and Setup

In partnership with the Department of Environmental and Earth Sciences at the University of Milano-Bicocca, 16 wheat spikes from each group were collected and transported to Milan for analysis. To preserve the plant material condition, the spikes were kept below $Temp < 0°C$ during transport and stored in a cryogenic room at the EuroCold Laboratory (University of Milano-Bicocca). Measurements were conducted using the HyIce hyperspectral system,

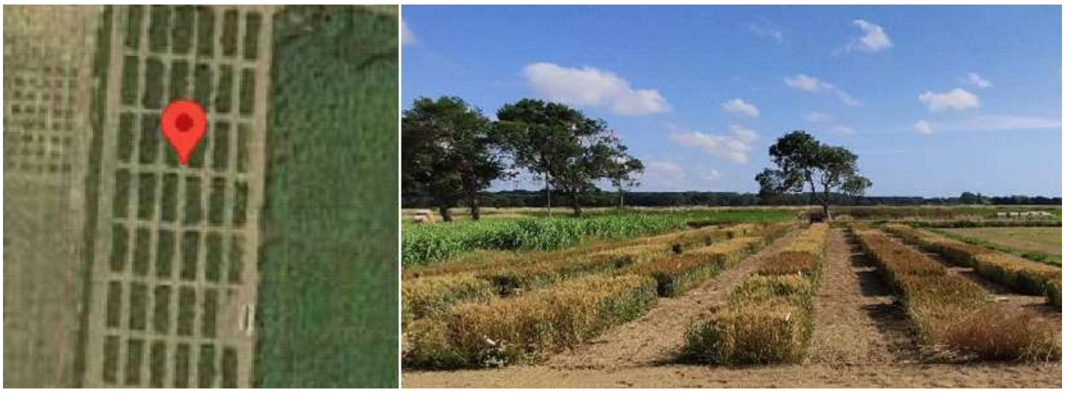
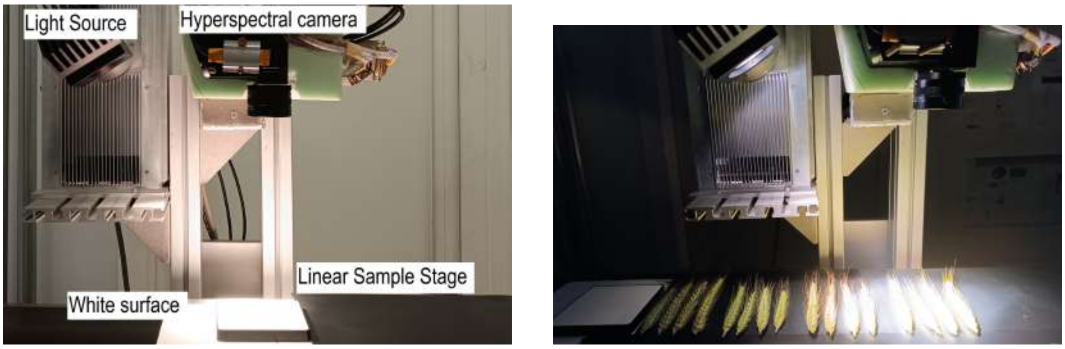


Figure 4.1.1: The wheat field, showcasing two views: aerial (left) and frontal (right).

Index	ID	Treatment description
1	RUN	Cultivar Rebelde, without pathogen inoculation, control parcel
2	RUT	Cultivar Rebelde, without pathogen inoculation, Trichoderma treatment
3	RUS	Cultivar Rebelde, without pathogen inoculation, Systemic treatment
4	RIN	Cultivar Rebelde, with pathogen inoculation, control plot
5	RIT	Cultivar Rebelde, with pathogen inoculation, Trichoderma treatment
6	RIS	Cultivar Rebelde, with pathogen inoculation, Systemic treatment
7	BUN	Cultivar Bingo, without pathogen inoculation, control plot
8	BUT	Cultivar Bingo, without pathogen inoculation, Trichoderma treatment
9	BUS	Cultivar Bingo, without pathogen inoculation, Systemic treatment
10	BIN	Cultivar Bingo, with pathogen inoculation, control parcel
11	BIT	Cultivar Bingo, with pathogen inoculation, Trichoderma treatment
12	BIS	Cultivar Bingo, with pathogen inoculation, Systemic treatment

Table 4.1: The summary table with the 12 groups of wheat growth by the University of Pisa

as depicted in Figure 4.2.1.



(a) Main components of the HyIce system showcasing the imaging spectrometer and illumination setup.

(b) The system in operation, measuring a group of wheat spikes.

Figure 4.2.1

This system comprises a stable 600W halogen lamp for sample illumination and a hyperspectral imaging spectrometer (Hyperspec VNIR, HeadWall Photonics) for collecting reflected light. The spectrometer and light source are mounted on a platform that allows for movement across a black surface measuring 80 cm in width and 200 cm in length, where samples are placed for imaging. The system design enables vertical movement up to 120 cm above the sample surface, facilitating high-resolution imaging with a spatial resolution capability of up to $20\mu m$.

The HyIce system core, the HeadWall Hyperspec VNIR imaging spectrometer, captures spectral radiance across 840 bands in the visible and near-infrared (VIS and NIR) wavelengths, from 380 to 1000 nm, with a spectral resolution of 2–3 nm. The spectrometer placement ensures even lighting across the sample surface, with adjustable illumination and view angles to optimize measurement conditions.

Measurements for each group involved all 16 spikes, arranged vertically on the black surface with sufficient spacing to aid subsequent image processing, as illustrated in Figure 4.2.2. The system evaluates the Dark Current before each measurement, storing the results externally.

To enhance data quality and accuracy, a thorough process of spatial and spectral oversampling was employed, yielding a hyperspectral matrix of dimensions $4450 \times 1000 \times 840$, with each wavelength interval approximately 0.70 nm apart. The system also automatically generates an RGB image matching the spatial dimensions of the hyperspectral image.

For further analysis, a subset of four relevant groups, as detailed in Table 4.1, were selected: BUN, BIN, BIS, and BIT, resulting in four distinct hyperspectral images.

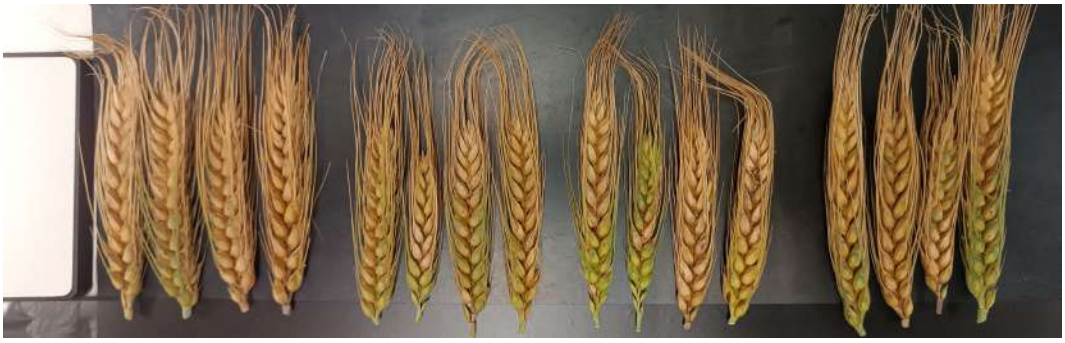


Figure 4.2.2: Experimental setup example with wheat spikes spaced evenly on the black surface.

4.3 Data Preprocessing

After image acquisition, the preprocessing phase started with two primary objectives: segmenting the images of the 16 spikes from each group into individual samples to minimize RAM load during analysis, and transforming the incoming irradiance data into apparent reflectance (R_{app}) for each pixel.

The initial step involved segmenting the composite image into 17 discrete images, including the 16 spikes and a white reference surface. To facilitate semi-automatic segmentation, a basic thresholding and stepping algorithm was implemented due to the characteristic high intensity levels on the white reference surface and lower levels in the empty spaces between spikes against the black background.

Assuming the incoming image as a hyperspectral matrix of size$M \times N \times \Lambda$, with M and N being the spatial dimensions and Λ the number of wavelengths[1], the segmentation process aimed at isolating each spike and the reference surface through a predetermined intensity threshold T and a column jump value J: This approach effectively identified sixteen regions corresponding to the spikes and an additional region for the white reference surface.

Algorithm 1 Image Splitting Algorithm

```
Establish a threshold $T$
Set a column jump value $J$
Evaluate $OldIntensity$ = sum of the first column intensities
Select the first column as the start of the white surface
for $i\_column = 2$ to $N$ do
    $NewIntensity$ = sum of column intensities
    if $OldIntensity < T$ and $NewIntensity > T$ or vice-versa then
        Select the column
        $i\_column = i\_column + J$
    end if
end for
```

[1] By referring to the acquired images as 4.2.2 M>N

A parallel procedure ensured the inclusion of the white reference surface in the vertical calibration, illustrated in Figure 4.3.2. Manual adjustments were made where necessary.

The vertical boundaries of the white surface were also defined, identifying the spatial positions of pixels within the white surface as $WS = [(x_1, y_1), \ldots, (x_R, y_R)]$.

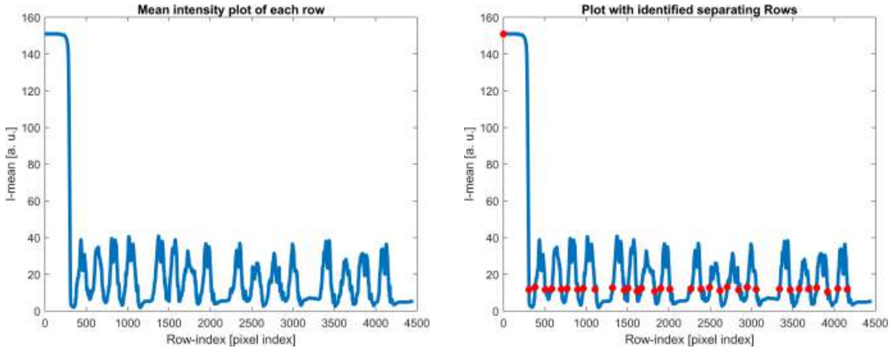


Figure 4.3.2: Semi-Automatic Segmentation of Hyperspectral Images: (Left) Original image showing the spatial arrangement of wheat spikes. (Right) Result of applying the Image Splitting Algorithm to identify horizontal limits of individual spikes.

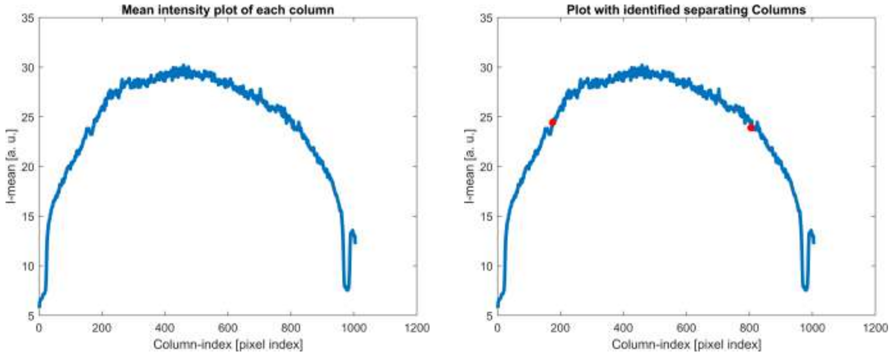


Figure 4.3.3: Identifying Vertical Limits of the White Reference Surface: (Left) The hyperspectral image before processing. (Right) Outcome of the Image Splitting Algorithm, highlighting the vertical delineation of the white reference area.

Following segmentation, a further preprocessing step adjusted the intensity levels to accurately compute the apparent reflectance (R_{app}). This entailed subtracting the Dark Current (DC), recorded by the measurement instrument before each data acquisition session. For each pixel $(x, y) \in 1, \ldots, M \times 1, \ldots, N$ of each image:

$$I_{CORR}(x, y, \lambda) = I(x, y, \lambda) - DC(\lambda) \tag{4.1}$$

The mean value of the white surface was then calculated using pixels within WS:

$$I_{WHITE}(\lambda) = \sum_{(x,y)\in WS} \frac{I_{CORR}(x, y, \lambda)}{N} \tag{4.2}$$

Finally, R_{app} for each wavelength was determined by:

$$R_{app}(\lambda) = I_{CORR}(x, y, \lambda)/I_{WHITE}(\lambda) \tag{4.3}$$

A threshold was applied to $R_{app} = 1$ values at 1 to address instances where elements of the white surface might exceed this value.

Thus, each hyperspectral matrix contains R_{app} data across 840 wavelengths, corrected for dark current and normalized against the white surface, as depicted in Figure 4.3.4.

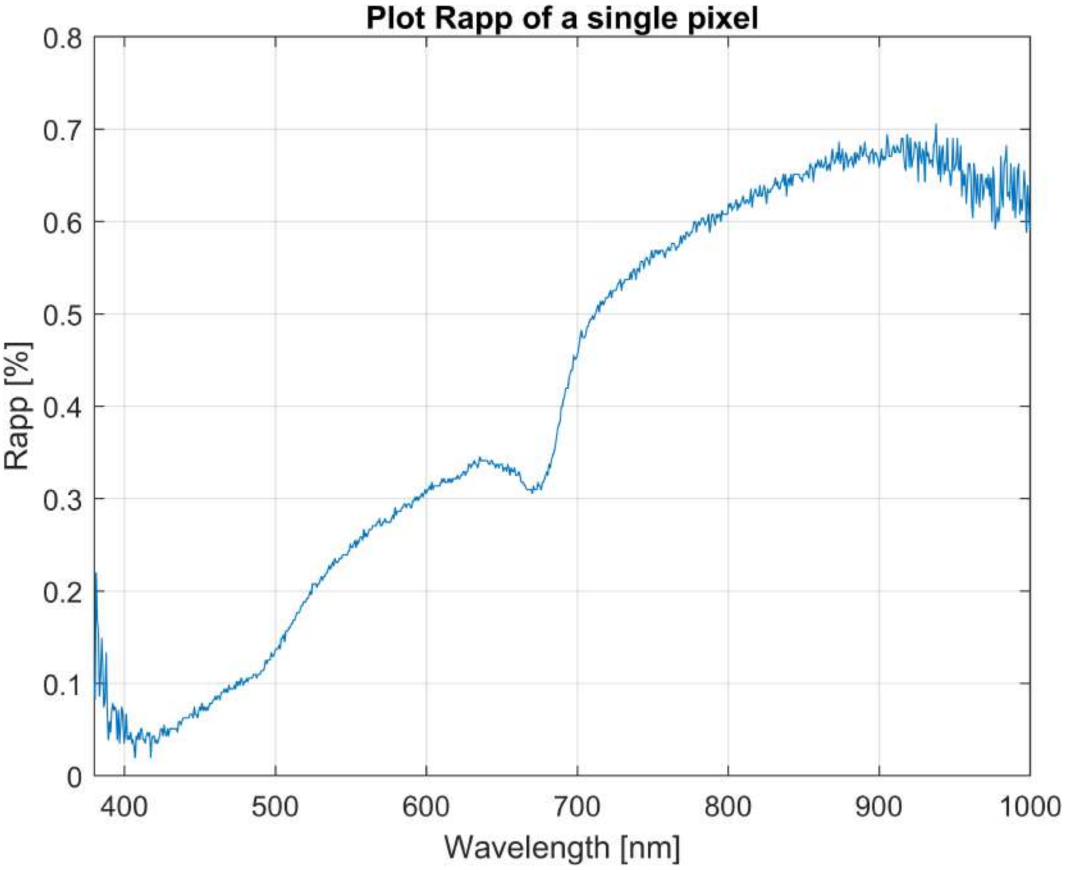


Figure 4.3.4: Pixel Spectrum Analysis: the graph displays the apparent reflectance (R_app) spectrum for a single pixel from an inoculated 'Cultivar Bingo' wheat spike.

4.4 RGB Creation

To provide an initial, qualitative insight into the analysis, RGB images were generated from the hyperspectral preprocessed data. While various methods exist for converting hyperspectral data to RGB images, the goal here was not to create photorealistic images but rather functional ones to support analysis tasks. Consequently, a straightforward algorithm was applied.

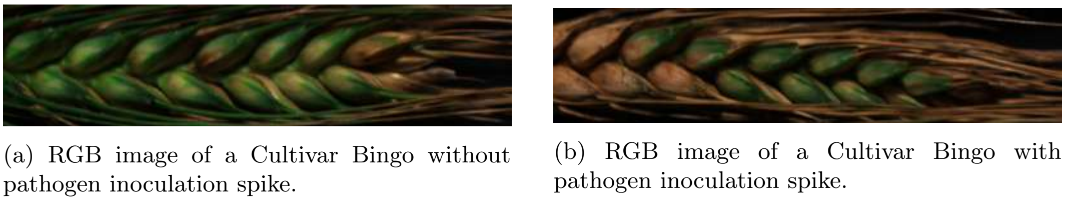
The preprocessed hyperspectral data are represented by $M \times N \times \Lambda$ hypercube matrices of R_{app}, where M and N are the spatial pixel dimensions and Λ ranges from 1 to 840, containing the spectral information for each pixel.

To construct RGB images, three wavelengths representing red, green, and blue were selected, leading to the creation of new matrices $M \times N \times 3$. The specific wavelengths chosen for this purpose were:

$$\begin{aligned} RED &= 669nm \\ GREEN &= 554nm \\ BLUE &= 472nm \end{aligned} \tag{4.1}$$

Following this, the resulting images were saved, facilitating a preliminary, qualitative comparison among different spikes, as illustrated in Figure 4.4.1.

Additionally, this process aids in identifying potential issues encountered during data collection with the HyIce instrument, enabling the exclusion of problematic samples from further analysis, as depicted in Figure 4.4.2.



(a) RGB image of a Cultivar Bingo without pathogen inoculation spike.

(b) RGB image of a Cultivar Bingo with pathogen inoculation spike.

Figure 4.4.1: These figures enable a preliminary differentiation between various spike groups based on color.

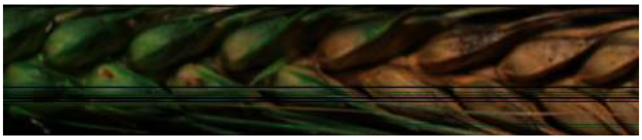
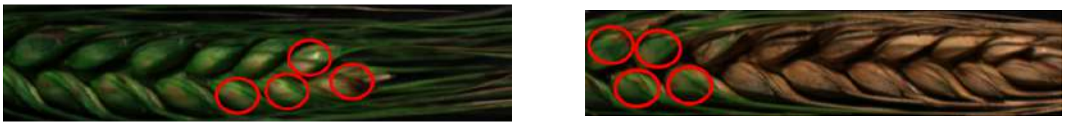


Figure 4.4.2: RGB image of a Cultivar Bingo with pathogen inoculation spike, highlighting some issues. Some lines were not correctly saved, suggesting parts of the image may need to be ignored or excluded from further analysis.

4.4.1 Observations Based on RGB Analysis

The RGB analysis of the four selected groups highlighted distinct features in the spikes, revealing both green and yellow portions. A qualitative difference was noted in the yellow tones of the uninfected spikes compared to those infected, as depicted in Figure 4.4.3. Moreover, dark-brown or black spots, consistent with literature findings, were observed in the yellow areas of infected spikes, indicating the presence of the pathogen.

Based on these observations, subsequent analyses involving specific parts of the spikes



(a) RGB image of a Cultivar Bingo spike without pathogen inoculation. The yellow caryopses of the spike are circled in red.

(b) RGB image of a Cultivar Bingo spike with pathogen inoculation. The green caryopses of the spike are circled in red.

Figure 4.4.3: These images facilitate an initial distinction within each spike, identifying different kinds of caryopses in the same spike group.

(such as caryopses) were categorized into 8 different classes as outlined in Table 4.2.

Index	**ID**	**Treatment description**
1	sBUNgreen	Cultivar Bingo, without pathogen inoculation, green portion of the image.
2	sBUNyellow	Cultivar Bingo, without pathogen inoculation, yellow portion of the image.
3	sBINgreen	Cultivar Bingo, with pathogen inoculation, green portion of the image.
4	sBINyellow	Cultivar Bingo, with pathogen inoculation, yellow portion of the image.
5	sBITgreen	Cultivar Bingo, with pathogen inoculation, Trichoderma treatment, green portion of the image.
6	sBITyellow	Cultivar Bingo, with pathogen inoculation, Trichoderma treatment, yellow portion of the image.
7	sBISgreen	Cultivar Bingo, with pathogen inoculation, Systemic treatment, green portion of the image.
8	sBISyellow	Cultivar Bingo, with pathogen inoculation, Systemic treatment, yellow portion of the image.

Table 4.2: The summary table with the 8 classes defined based on RGB images considered for the analysis.

4.5 Mask Creation

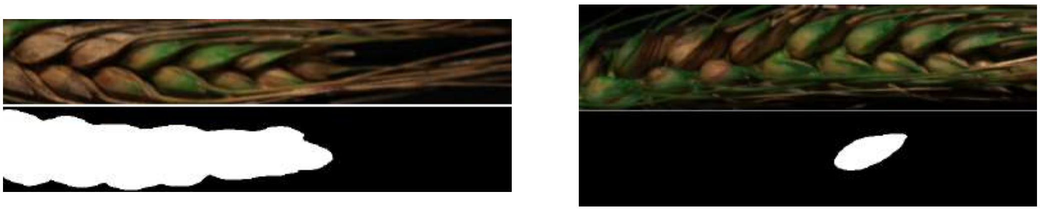
For further analysis steps, it became necessary to develop a method for selecting specific portions of images. This requirement served two main purposes: to exclude unwanted pixels, such as background remnants from semi-automatic preprocessing, and to isolate specific parts of a spike, such as a caryopsis, for detailed examination.

To accomplish this, several masks were created using commercial or open-source software like Fiji (ImageJ), as illustrated in Figure 4.5.1. The masks were designed in two categories:

- Masks that exclude the awn and background for broader analyses involving the entire spike.

- Masks that isolate a single caryopsis for more detailed analyses focusing on specific spike portions or to highlight differences within the same image.

Approximately 500 masks were manually created, while preliminary tests for automatic background removal were also carried out to lay the groundwork for future projects.



(a) A spike alongside its "macroscopic" mask, with the background and awns removed.

(b) A spike with a "microscopic" mask focusing on a single caryopsis.

Figure 4.5.1: Examples of manually created masks.

4.6 Rapp Spectral Analysis

Following the creation of masks, we focused on the spectral analysis of the preprocessed images.

This analysis involved a qualitative examination of the *Rapp* spectrum across different groups, facilitated by the following steps:

1. *Image and pixel selection:* exploiting the designated mask, the selected image is filtered, and a set $S \in \mathcal{R}^2$ of pixels is chosen.

2. *Evaluation of the mean Intensity:* The spatial mean of the *Rapp* intensity is calculated for each wavelength λ_i as:

$$\overline{R_{app}}(\lambda_i) = \sum_{(x,y)\in S} \frac{R_{app}(x,y,\lambda_i)}{Npixels} \tag{4.1}$$

where $Npixels$ is the count of points in S;

3. *Normalization of the total spectrum:* The $\overline{R_{app}}(\lambda_i)$ value is normalized across the total area:

$$R_{norm}(\lambda_i) = \frac{\overline{R_{app}}(\lambda_i)}{\sum_{i=1}^{840} \overline{R_{app}}(\lambda_i)} \tag{4.2}$$

This process is replicated for each image-mask pair, with results recorded for subsequent analysis phases.

4.6.1 Results with "Macroscopic" Masks

Upon determining $R_{norm}(\lambda_i)$ for each region of interest, comparisons were made. The initial analysis assessed the mean spectral response across different spike groups using "macroscopic" masks. This comparison did not differentiate between caryopsis colors, thus evaluating four distinct groups. The outcomes are presented in Figure 4.6.1.

Uninfected spikes (sBUN) exhibit a significant dip in R_app around 680 nm and a less pronounced one near 500 nm. Infected spikes lack these dips, whereas treated spikes display an intermediate pattern.

4.6.2 Results with Single Caryopsis Masks

A more detailed analysis was conducted using "microscopic" masks to observe variations among different caryopses within the same spike. This examination leveraged the 8 classes derived from RGB analysis, as outlined in Table 4.2.

The comparison of the mean $Rnorm$ spectra across different caryopsis categories revealed distinct patterns. As illustrated in Figure 4.6.2, a noticeable separation was observed between yellow and green caryopses. The spectral behavior of yellow caryopses closely matched that of infected spikes, whereas green caryopses exhibited similarities to uninfected spikes. Figure 4.6.3 serves as a reference for distinguishing between healthy and infected classes in this analysis.

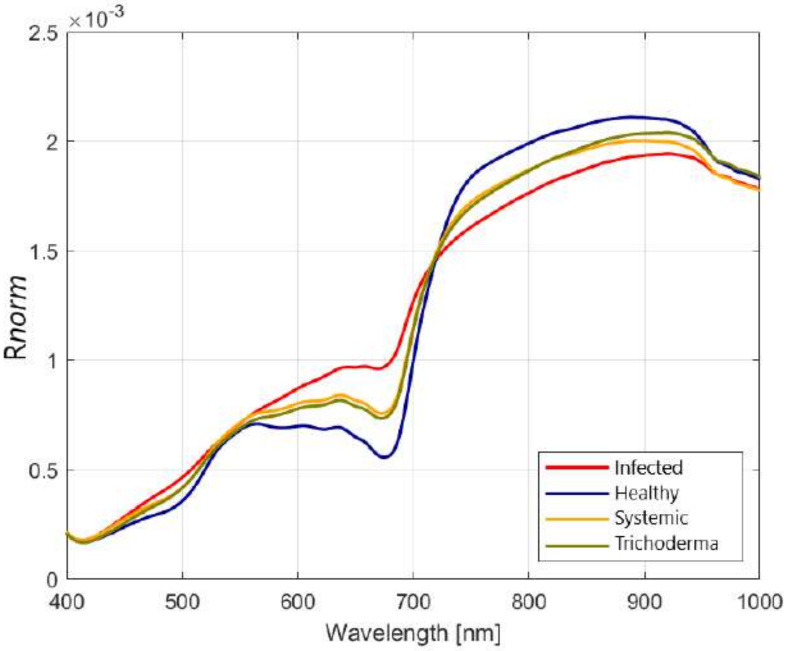


Figure 4.6.1: Comparison of R_{norm} spectra among different spike groups: Healthy (without pathogen), Infected (with pathogen), Systemic (systemic treatment), and Trichoderma (Trichoderma treatment).

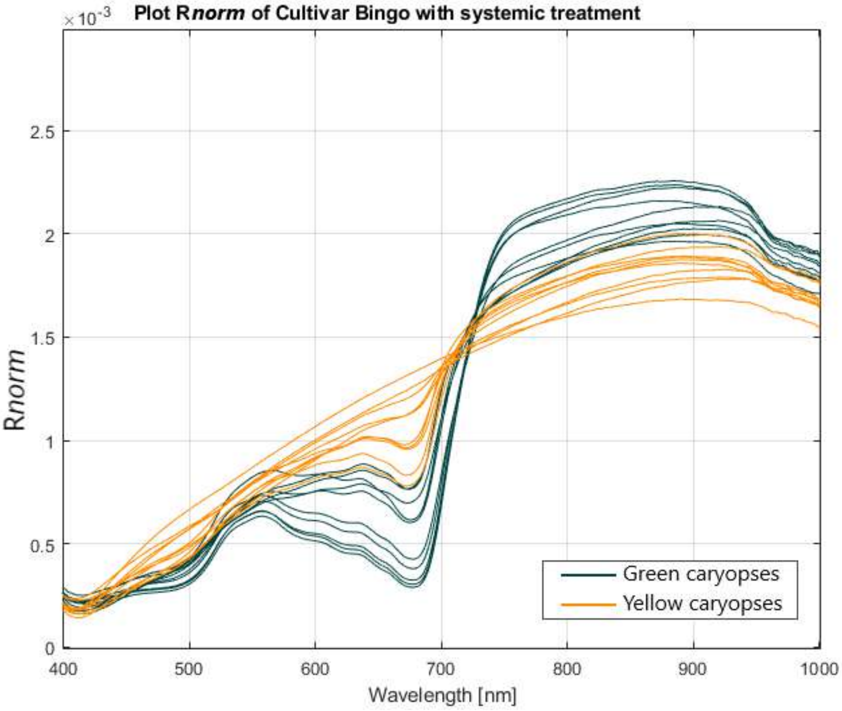


Figure 4.6.2: Comparison of R_{norm} spectra between green and yellow caryopses in a Cultivar Bingo with systemic treatment, depicted in different colors.

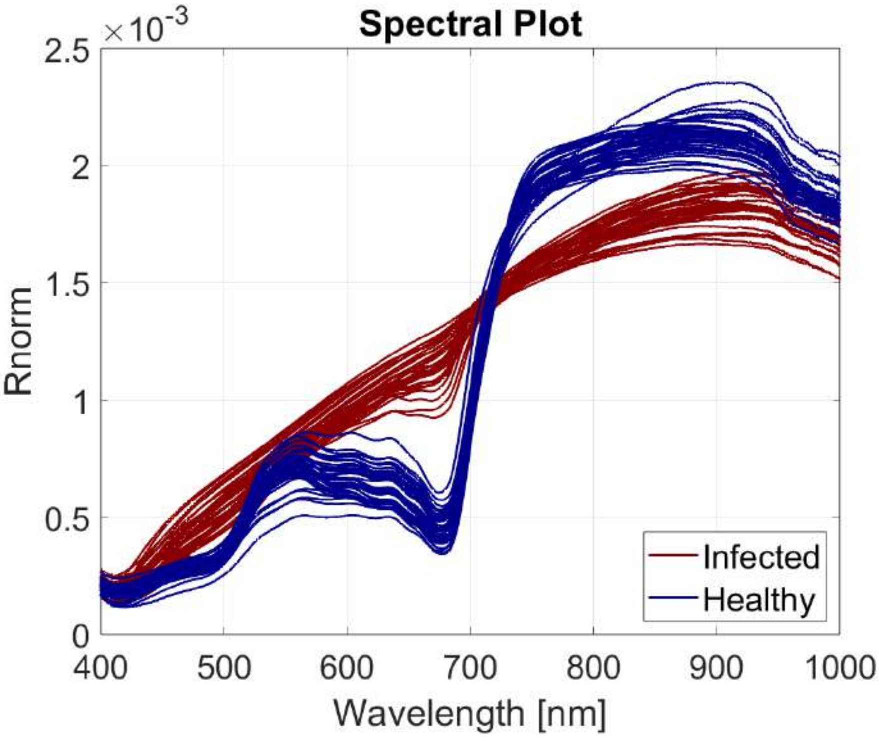


Figure 4.6.3: Comparison of R_{norm} spectra between green caryopses in healthy spikes and yellow caryopses in infected spikes.

4.7 Phasor Plane Analysis

The subsequent phase of the analysis, central to this thesis, focuses on phasor plane analysis and centroid examination. The objective is to pinpoint the spectral regions displaying the most significant differences in gs-phasor coordinates among the classes. This necessitates calculating the gs coordinates, a task that involved a subset of 4 spikes selected from each group.

Each image in the dataset underwent Discrete Fourier Transform (DFT) processing to derive the phasor coordinates for each pixel.

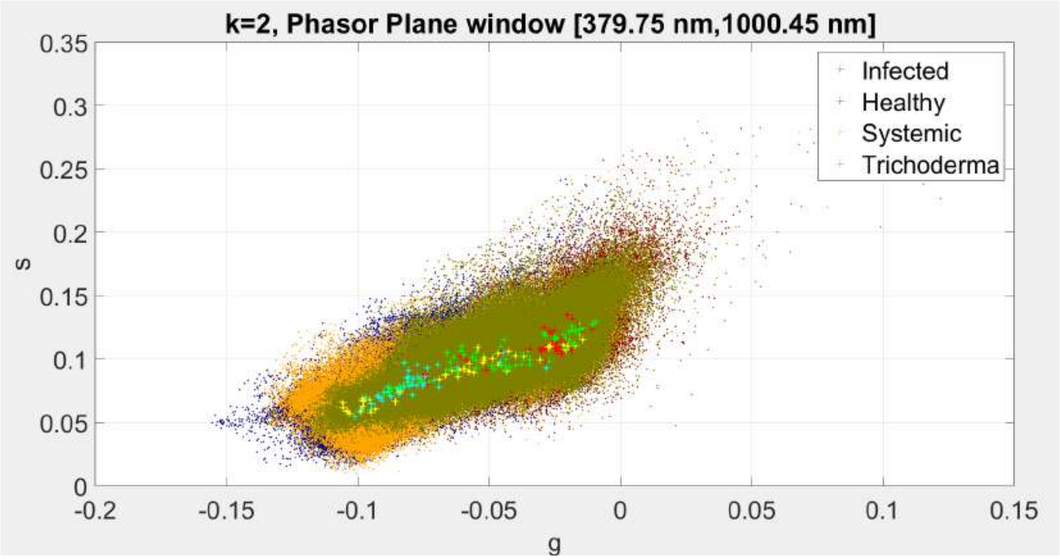
The algorithm hinges on two main parameters: the harmonic kk and the spectral range, with outcomes varying based on the initial λ_{in} and final λ_{fin} wavelengths chosen for transforming the R_{norm} spectra. The analysis employs two strategies:

- *Whole spectrum analysis:* Applies the transformation across the entire spectrum.
- *Multiple bands (or windows) analysis:* Divides the original spectral range $[400nm, 1000nm]$ into bandwidths of 20 nm, 15 nm, 10 nm, 7 nm, 3.5 nm.

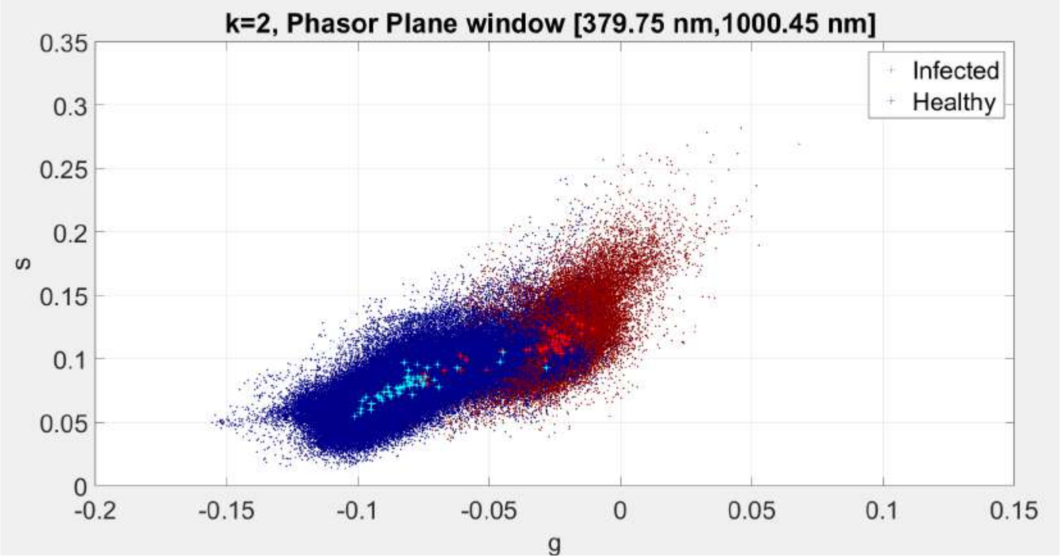
While the whole spectrum method provides a straightforward way to compare mean spectral differences between groups via a 2D plot, the multiple bands approach helps identifying spectral regions where pixels from different classes are mostly separated.

4.7.1 Phasor Plane Analysis Results

The *Whole spectrum analysis* with the first harmonic did not reveal significant separation among classes. However, the second harmonic analysis, shown in Figure 4.7.1, indicates that while the clouds of the four groups overlap to some extent, a distinction emerges when analyzing the colors of caryopses in line with RGB analysis, as evident in Figure 4.7.3. Here, a clearer separation between infected and healthy gs-clouds is observed. Centroids calculated for each caryopsis suggest that healthy green caryopses and infected yellow ones cluster in distinct phasor plane regions, with treated spike centroids positioned in the middle, indicating a mix of "healthy" and "infected" caryopses.

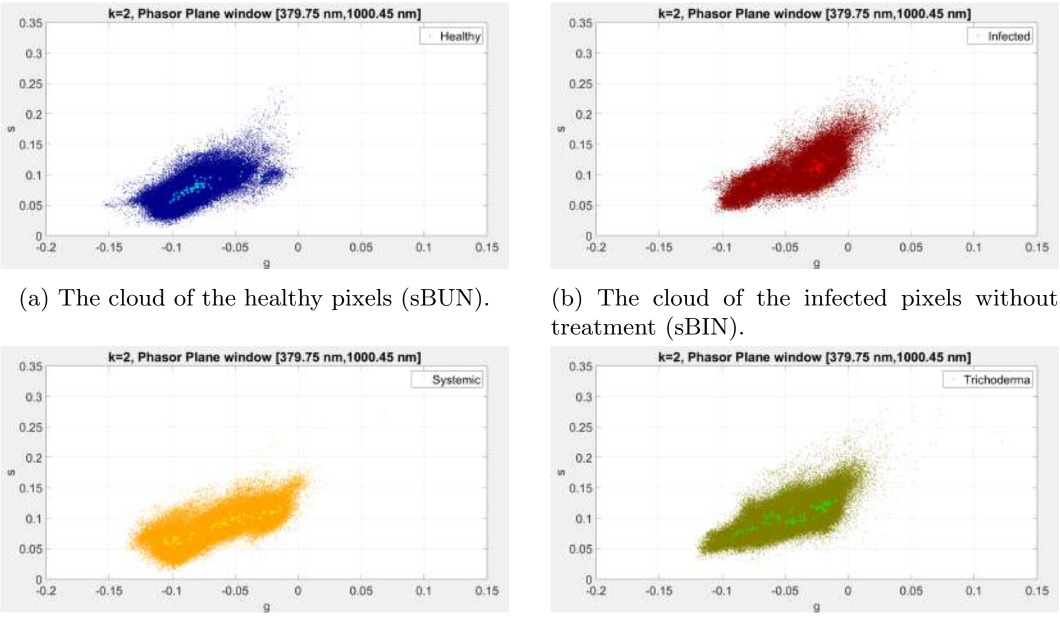


(a) Clouds of the four spike types analyzed together.



(b) The clouds of the healthy and infected spikes without treatment.

Figure 4.7.1: gs clouds of spikes analyzed without color distinction among caryopses, employing the DFT approach on the entire spectrum.



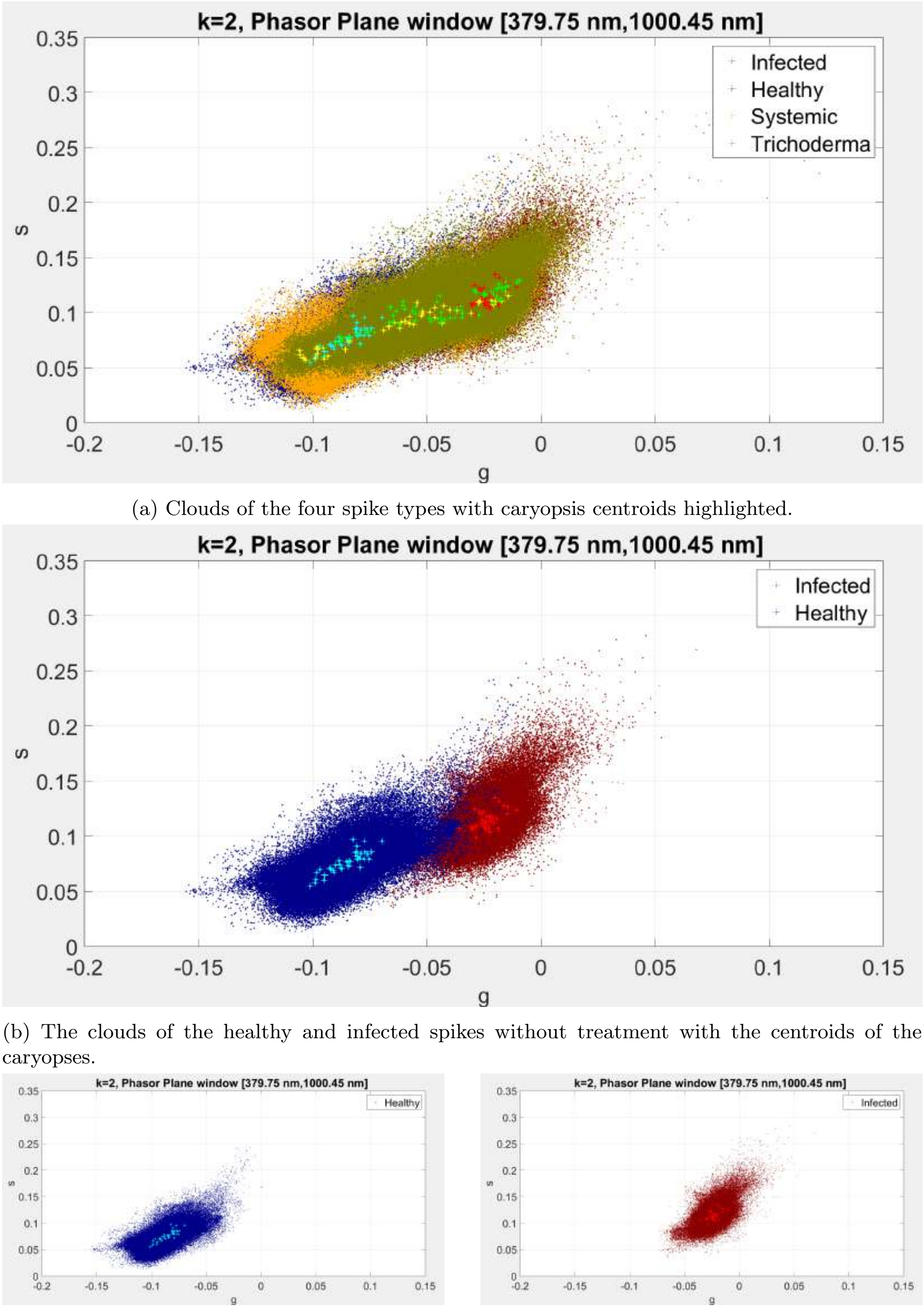
(a) The cloud of the healthy pixels (sBUN).

(b) The cloud of the infected pixels without treatment (sBIN).

(c) The cloud of the infected pixels with Systemic treatment (sBIS).

(d) The cloud of the infected pixels with Trichoderma treatment (sBIT).

Figure 4.7.2: gs clouds of spikes analyzed without color distinction among caryopses, employing the DFT approach on the entire spectrum.



(a) Clouds of the four spike types with caryopsis centroids highlighted.

(b) The clouds of the healthy and infected spikes without treatment with the centroids of the caryopses.

(c) The cloud of the healthy pixels (sBUN) with the centroids of the caryopses.

(d) The cloud of the infected pixels without treatment (sBIN) with the centroids of the caryopses.

Figure 4.7.3: The gs clouds of the spikes with the centroids of the caryopses. For the infected spikes just the yellow caryopses are plotted, while for the healthy ones the green caryopses are plotted.

4.7.2 Centroid Comparison Results

The spectral analysis aimed to identify spectral regions capable of distinguishing different spike classes, integrating "Multiple bands (or windows) analysis" with a quantitative evaluation of centroids distances. This approach sought to pinpoint the optimal spectral band that maximizes the distance between class groups on the phasor plane.

Initially, it was necessary to compute each caryopsis centroid within each spectral window, averaging the pixel positions as outlined in Section 3.2. Based on preliminary findings, four groups were chosen for detailed examination. For uninfected spikes, only the green caryopses were considered, whereas for infected but untreated spikes, focus was placed on the yellow caryopses. For treated spikes, no color distinction was made to avoid overly complicating group comparisons.

For each group, a singular mean centroid was determined by averaging the centroids of all caryopses within the group. If N green caryopses from uninfected spikes were selected, each with a centroid c_i, the mean centroid for the "sBUNgreen" class was calculated as:

$$\overline{C(g,s)}_{sBUNgreen} = \sum_{i=1}^{N} \frac{c_i(g,s)}{N} \tag{4.1}$$

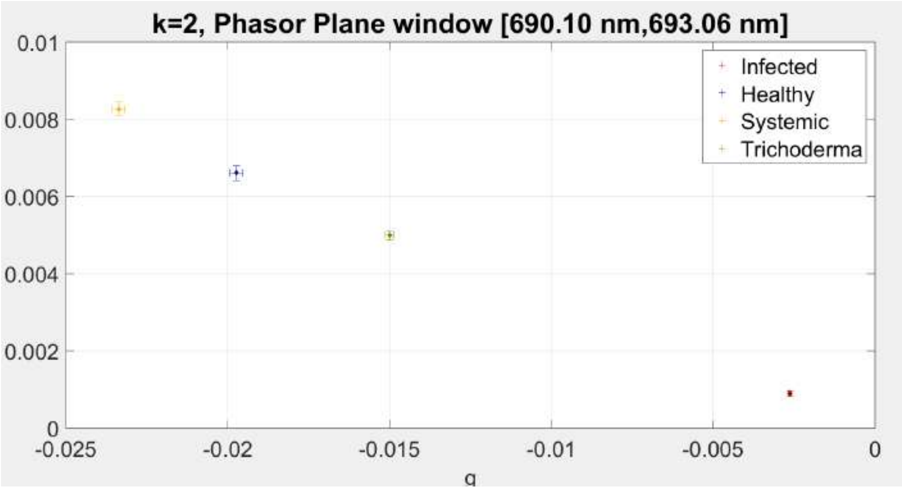
These values facilitated the creation of two types of plots:

- Centroids with standard deviation errors plotted on the gs-phasor plane for each window, as depicted in Figure 4.7.4. This visually distinguished potentially informative from non-informative bands for class differentiation. It also highlighted the first portion of the spectrum large centroid position errors.

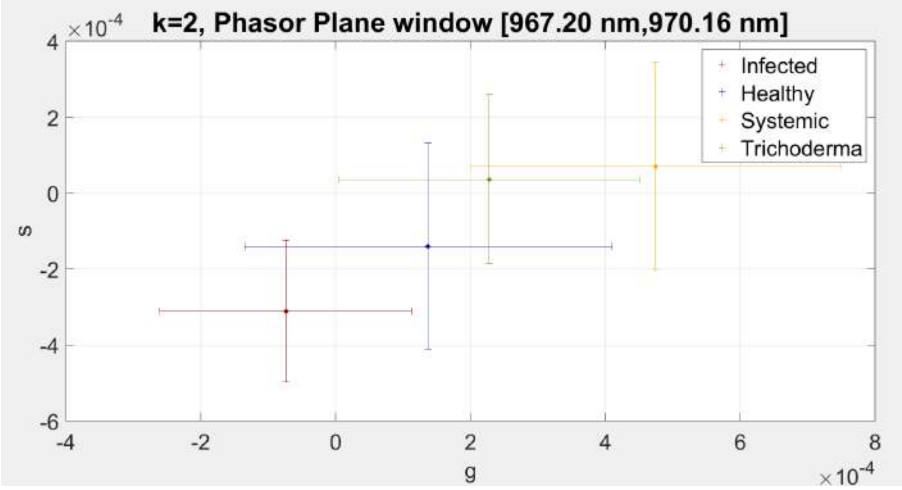
- A quantitative analysis assessing the Euclidean distance between centroids across the full spectral range $[380nm, 1000nm]$ is obtained for each class pair, illustrated in Figure 4.7.5. Matlab peak-finding algorithm pinpointed wavelengths where distance peaked, identifying ten key wavelengths for further analysis and machine learning training.

For a quantitative analysis the euclidean distance on the 2D plane is evaluated for the centroids and a comparison over the whole spectral range $[380nm, 1000nm]$ for each pair is shown in the Figure 4.7.5. At this point a built-in MATLAB peak-finder algorithm was applied in order to identify the specific wavelengths where the distance has a relative maximum.

This analysis highlights, for the second harmonic, ten wavelengths: 512 nm, 567 nm, 571 nm, 579 nm, 612 nm, 645 nm, 660 nm, 664 nm, 689 nm, 693 nm. These specific wavelengths were used for further analysis and for the training of the machine learning algorithms, as it will be explained later.

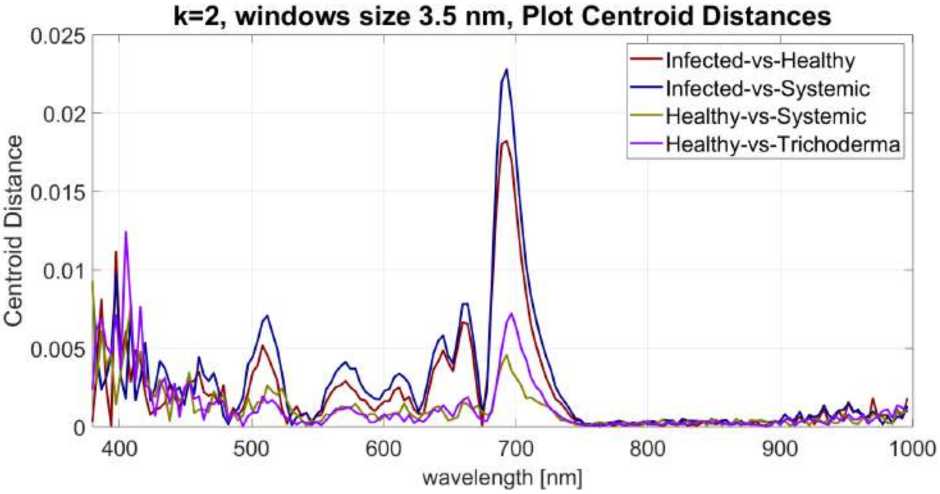


(a) Centroids of the four groups in an optimal band.

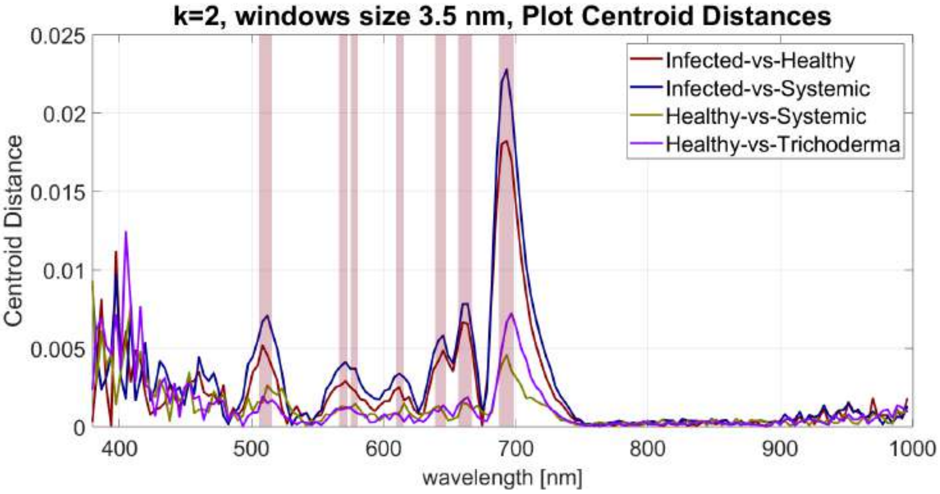


(b) Centroids of the four groups in a discarded window.

Figure 4.7.4: Examples of gs-plots of mean centroids for the four selected classes. Analysis used the second harmonic with a bandwidth of 3.5 nm.



(a) Distance between mean centroids of each class.



(b) Distances between mean centroids of each class with highlighted selected bands.

Figure 4.7.5: Plot of mean centroid distances for four selected classes over the wavelength range. Analysis exploited the second harmonic and a bandwidth of 3.5 nm.

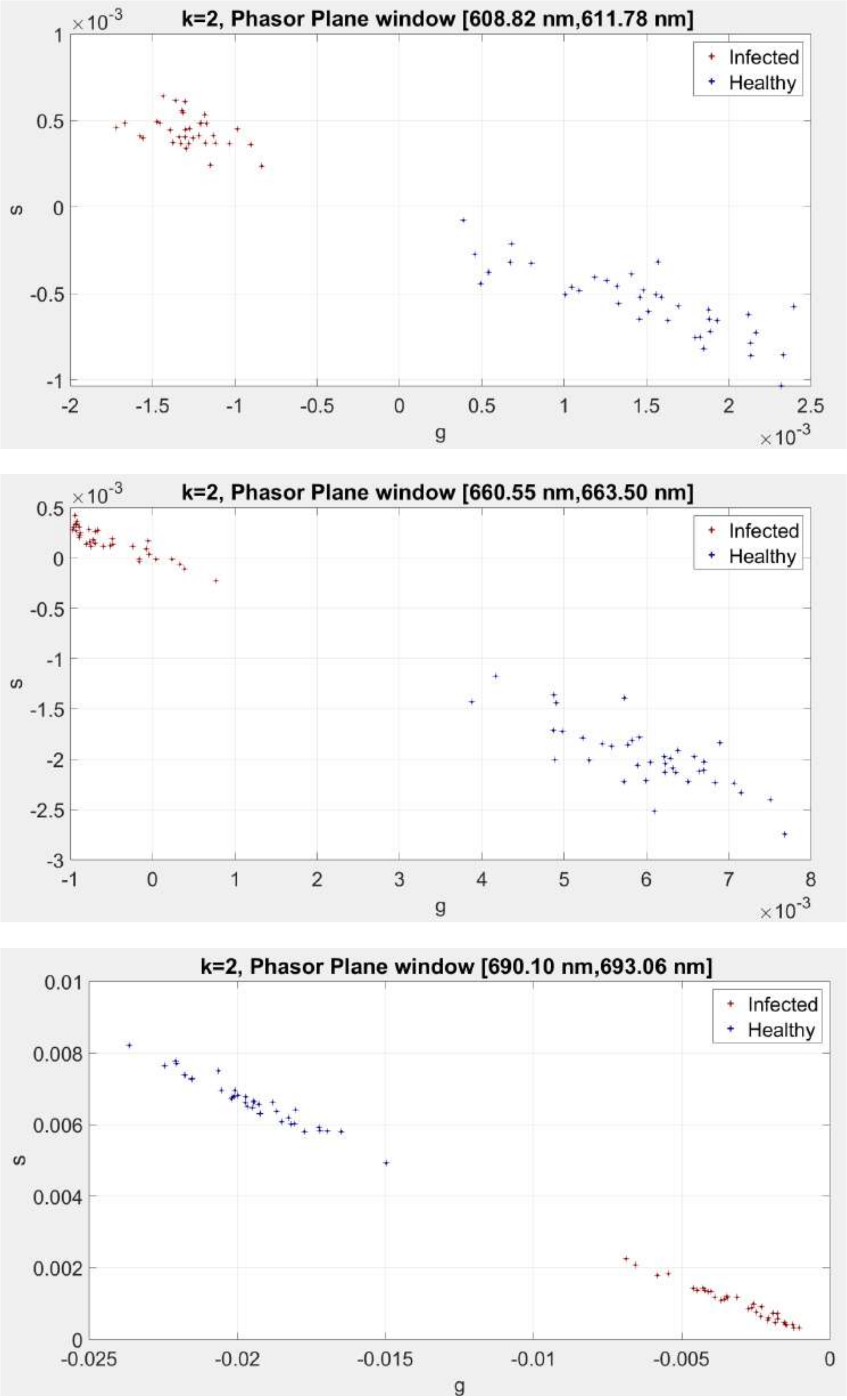


Figure 4.7.6: Centroids of the healthy and infected caryopses on the spectral regions selected by the centroid analysis.

4.8 Texture Analysis

This analysis aimed to determine if specific texture properties, introduced in Section 3.4, could highlight differences among classes of spikes, thereby assessing their suitability as features for machine learning algorithms. A challenge in GLCM-based analysis is managing the algorithm input parameters: wavelength λ, the number of gray levels G, and the size of the region of interest (ROI), which dictates the "pixel" size exploited in the machine learning analysis. The strategy involved the separation of the image into squares of $[ROI_{Size} \times ROI_{Size}]$ pixels and applying GLCM analysis within each ROI to assess the texture properties. This method allowed the identification of specific texture-related values for each pixel group within the region.

The ROI size significantly influenced the results, as demonstrated in Figure 4.8.1. Various trials sought to identify optimal input values that reveal distinctions between spikes of different classes. The graphical representation of the outcomes in Figure 4.8.2 highlights visible differences in some GLCM values across spike types.

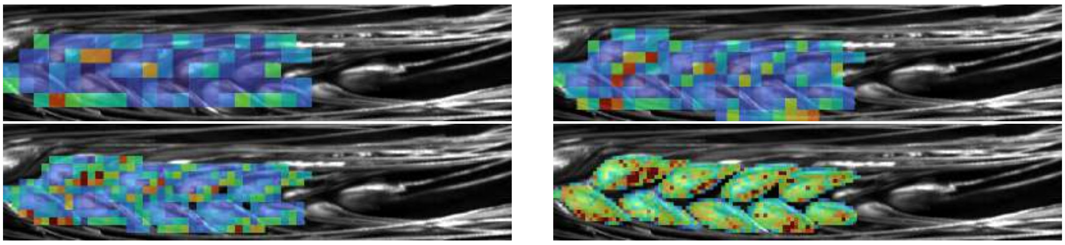
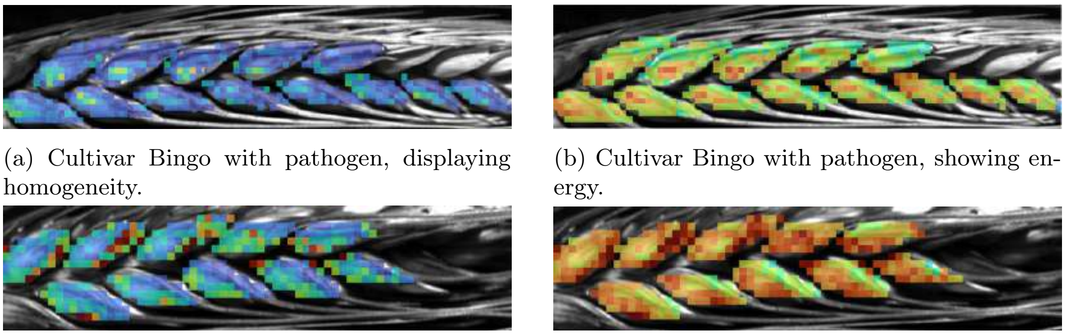


Figure 4.8.1: Graphical representation of the energy at different sizes for an uninfected spike, with varying ROI sizes: 20, 15, 10, 5 pixels



(a) Cultivar Bingo with pathogen, displaying homogeneity.

(b) Cultivar Bingo with pathogen, showing energy.

(c) Cultivar Bingo without pathogen, displaying homogeneity

(d) Cultivar Bingo with pathogen, showing energy.

Figure 4.8.2: Graphical representation of homogeneity and energy values with a color scale for numerical values. Caryopsis masks are applied. Input parameters: ROI size = 8 pixels, G-Levels = 25; $\lambda = 693nm$.

4.8.1 Results of the GLCM-Based Analysis

A more quantitative approach was considered to analyze the GLCM data obtained for each group. This facilitated a comparison between the distributions of GLCM results across different groups, aiding in the identification of input parameters that highlight class distinctions. Furthermore, it allowed the evaluation of texture parameters effectiveness in classifying the caryopses into four categories: healthy (green caryopses from uninfected spikes), infected (yellow caryopses from infected spikes), systemic treated, and Trichoderma treated.

Histograms were created under various scenarios: analyzing single groups, pairing infected and healthy caryopses, and comparing treated groups with both infected and healthy caryopses. The figures below showcase the distribution of these parameters with chosen input values: $\lambda = 693nm$, $G - levels = 25$, $ROI_{Size} = 8$ pixels.

Correlation

The Correlation results, depicted in Figure 4.8.3, do not exhibit a distinct pattern across classes. The distributions of healthy and infected caryopses are similar, leading to the exclusion of Correlation as a potential feature for subsequent analyses.

Contrast, Energy, Entropy, Homogeneity

The analyses for Contrast, Energy, Entropy, and Homogeneity, shown in Figures 4.8.4 - 4.8.7, indicate varying distributions between healthy and infected caryopses. These differences suggest that these parameters could serve as viable features for machine learning models. Kolmogorov-Smirnov tests confirmed the significance of these distinctions, yielding a $p - value < 0.01$ for the distributions across different classes for each selected wavelength (λ). The distributions for treated caryopses display intermediate patterns, aligning with observations from previous spectral analyses.

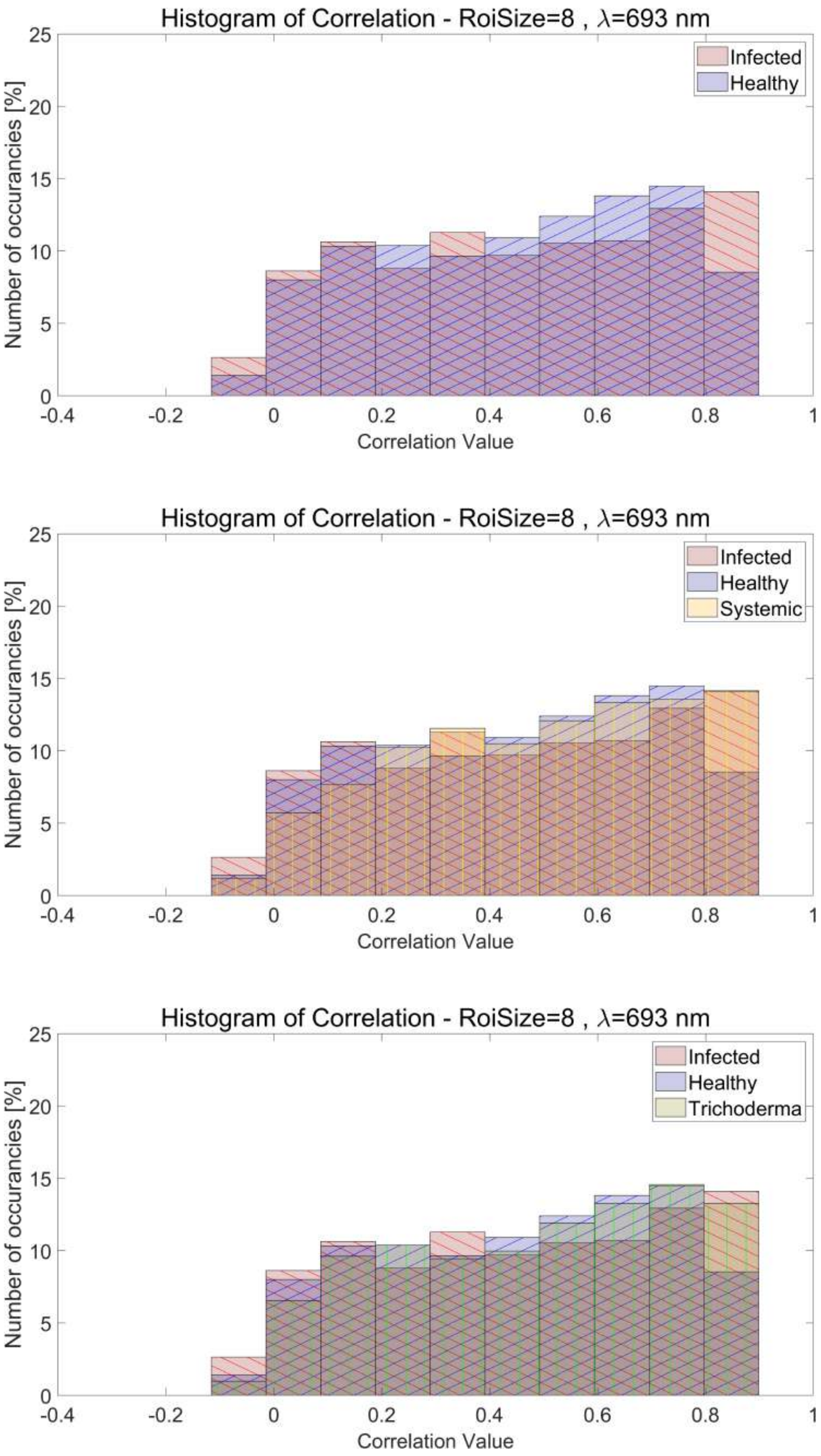


Figure 4.8.3: The histogram of the Correlation comparison for several classes.

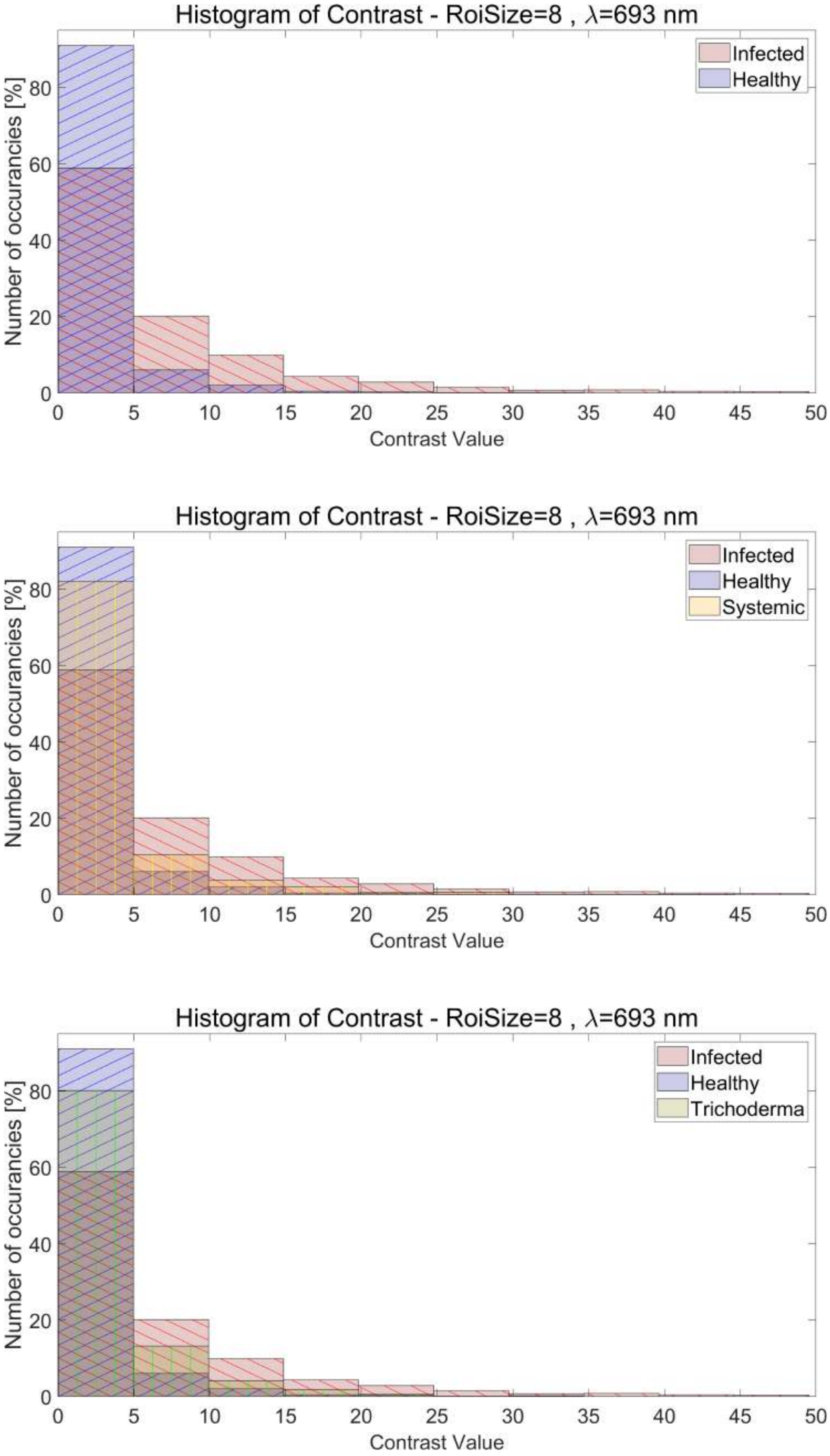


Figure 4.8.4: The histogram of the Contrast comparison for several classes.

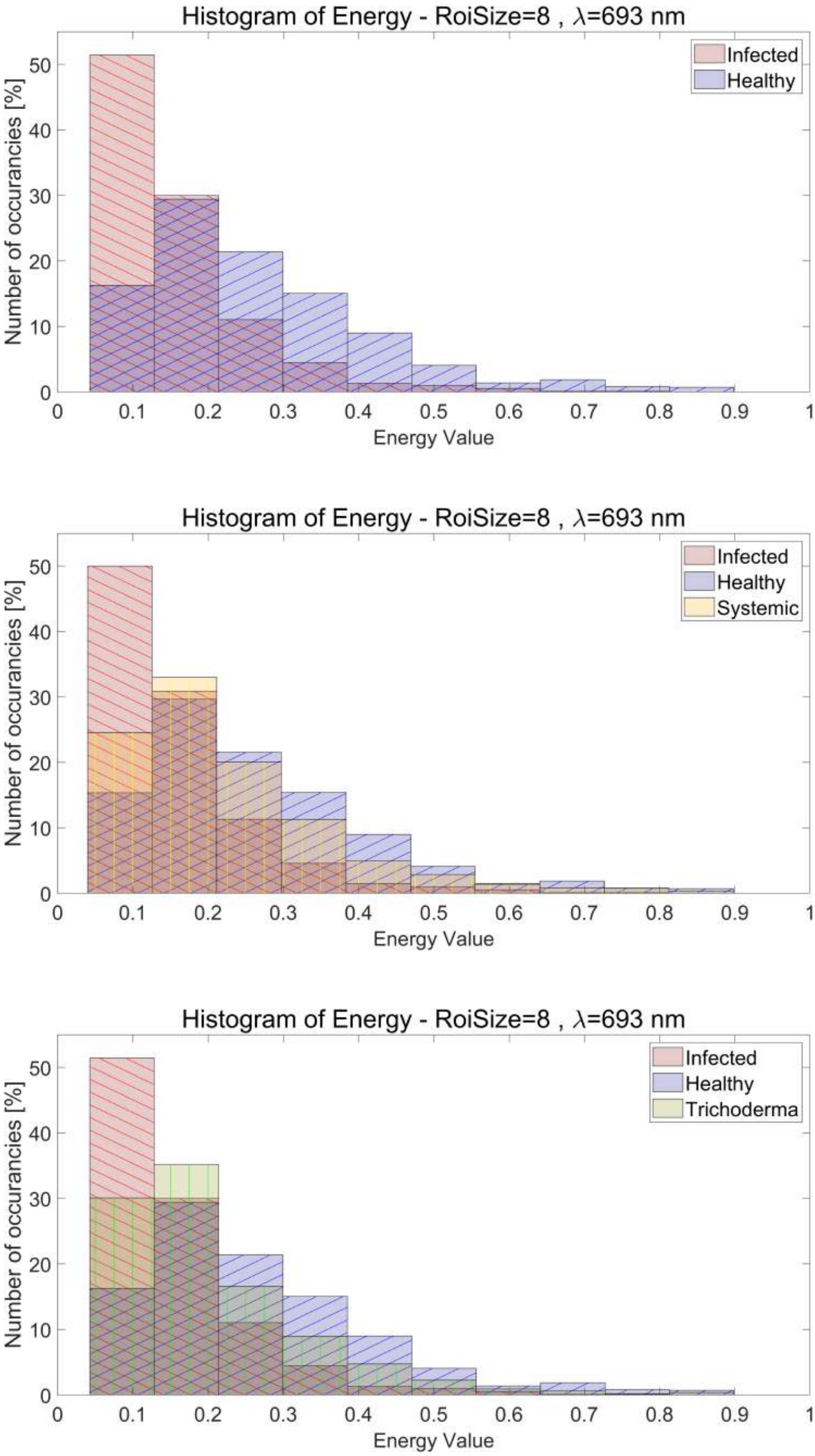


Figure 4.8.5: The histogram of the Energy comparison for several classes.

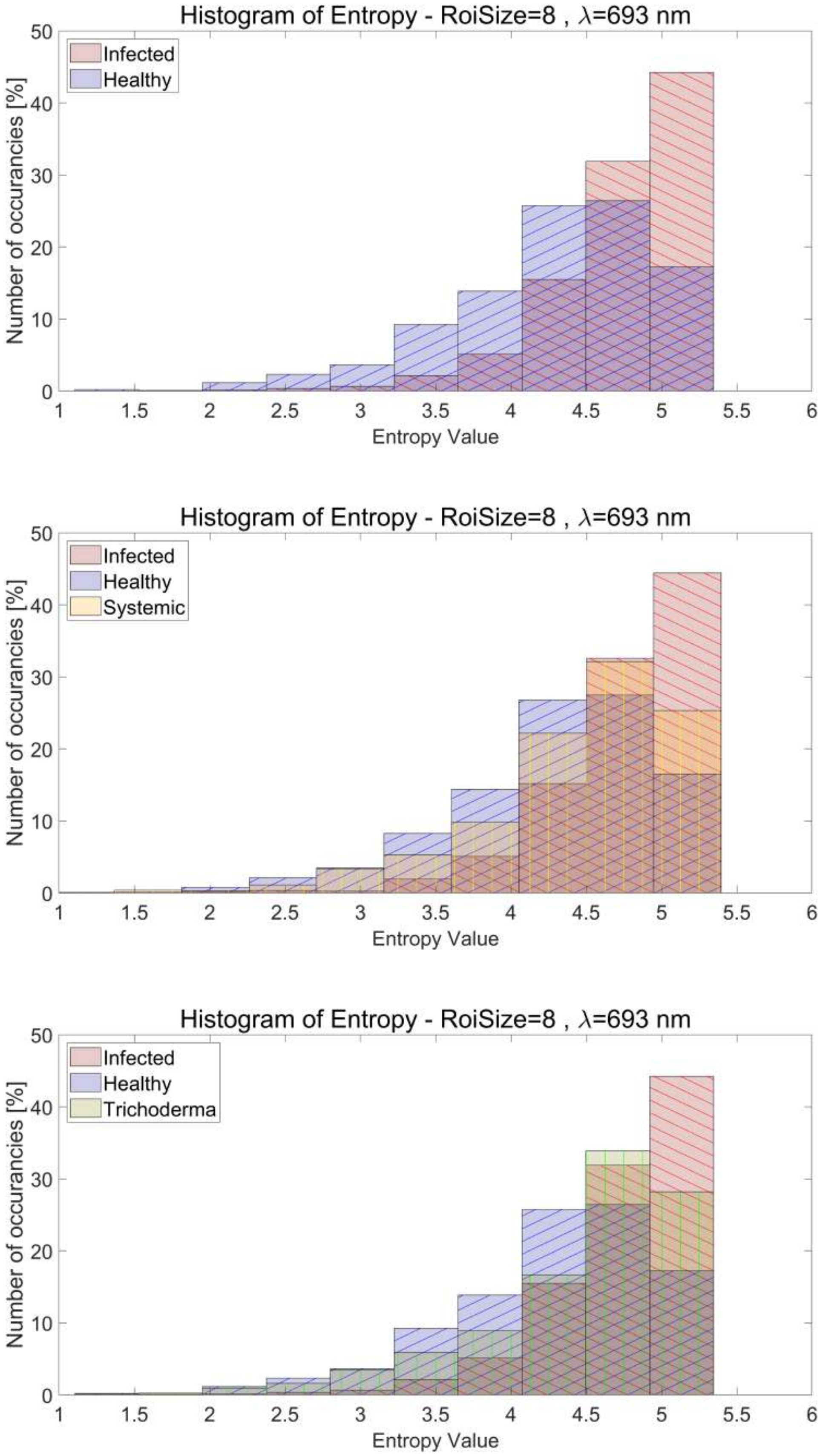


Figure 4.8.6: The histogram of the Entropy comparison for several classes.

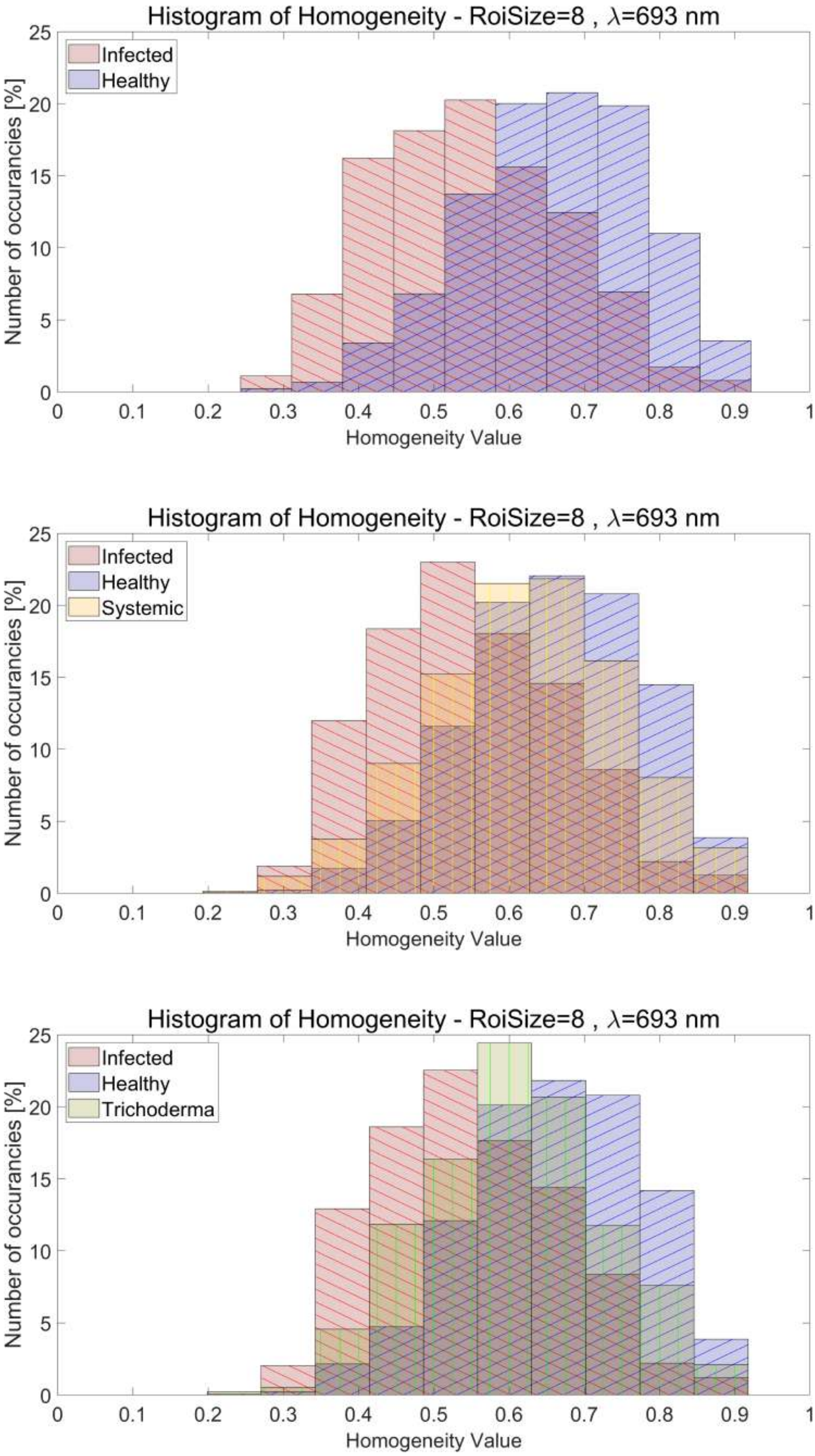


Figure 4.8.7: The histogram of the Homogeneity comparison for several classes.

4.9 Final Considerations for the Analysis

As demonstrated in the GS-plots from Sections 4.7 and 4.7.2, there is a clear distinction on the GS-plane between the green caryopses of Cultivar Bingo spikes without pathogens and the yellow caryopses of the Cultivar Bingo spikes with pathogens. These distinctions could be considered potentially effective features for distinguishing healthy and diseased portions of the plant.

The data points for the treated plants are located between the two reference groups (healthy and infected) in both the spectral and texture analyses. This suggests that some portions of the treated spikes remain healthy due to the treatment, while others are affected by the pathogen.

Although the plots are not presented in this thesis, the green caryopses of the treated groups resemble those of the healthy class, whereas the yellow ones display patterns similar to those of the infected class.

Through these analyses, a set of features has been selected, and the structure of the dataset for machine learning has been established as follows. Ten wavelengths λ were identified. For each λ, two spectral features can be derived, which are the GS coordinates. Additionally, by setting the texture input parameters to $G - levels = 25$ and $ROI_{Size} = 6$ pixels, it was possible to identify four features for each λ: Contrast, Energy, Entropy, and Homogeneity.

The total number of features selected is 60.

A final consideration must be made: the GLCM features are related to the region selected with the ROI, while the GS features are related to the pixel. A one-to-one correspondence is necessary to use the data as input for machine learning models.

To achieve this, two different approaches were applied:

- **Pixel ROI:** For each ROI identified with the GLCM analysis, a corresponding one is created for the spectral features by averaging the GS value of each pixel within the ROI and using the result as the coordinates of the ROI. With this approach, the dataset observation corresponds to the ROI identified in the GLCM analysis.

- **Caryopsis:** For both the GLCM and GS features, an average evaluation over each caryopsis is performed. In this case, the observation is the caryopsis itself.

Each approach has its strengths and weaknesses: the caryopsis structure helps to mitigate issues arising from anomalous pixel behavior within a caryopsis. However, given the limited size of the dataset, the number of observations in the caryopsis structure might be insufficient for the purpose of training an algorithm effectively.

Chapter 5. Application of Machine Learning Algorithms

5.1 Dataset Organization

Dataset sizes are detailed in Tables 5.1 and 5.2.

Due to insufficient data in the *Caryopsis Dataset*, the *Pixel ROI Dataset* was chosen for training purposes. The training process focused on two main classes, in line with the previous analysis results: the green caryopses of Cultivar Bingo without pathogens (sBUNgreen), representing the 'healthy' class, and the yellow caryopses of Cultivar Bingo with pathogen inoculation (sBINyellow), representing the 'diseased' class.

The resulting dataset was divided, with 80% used for training and the remaining 20% reserved for the 'Internal Testing'.

In other words an initial training was conducted by selecting only the two main groups from the *Pixel ROI Dataset*: sBUNgreen and sBINyellow.

Table 5.1: Pixel ROI Dataset

class	number of elements
sBINgreen	1900
sBINyellow	5800
sBUNgreen	4600
sBUNyellow	1800
sBISgreen	2000
sBISyellow	1200
sBITgreen	1400
sBITyellow	2700

Table 5.2: Caryopsis Dataset

class	number of elements
sBINgreen	40
sBINyellow	110
sBUNgreen	85
sBUNyellow	40
sBISgreen	30
sBISyellow	20
sBITgreen	20
sBITyellow	40

For this analysis, the Matlab Classifier Learner tool was utilized, which offers a comprehensive set of algorithms. The algorithms underwent training through K-Fold Cross-Validation resampling, with $K = 5$.

Different models were trained across three distinct feature conditions: using only texture features, using only spectral features, and using both spatial and spectral features.

The resulting models were than subjected to different tests.

5.2 Internal Testing

During this initial testing phase, the test dataset is composed of the remaining 20% of the data not used for the training. Accuracy served as the primary metric for evaluating performance. The results for the various scenarios are detailed as follows:

- For GS+GLCM: The best performing models include SVM Quadratic at 98.3%, SVM Cubic at 98.2%, Boosted Trees at 98.1%, Bagged Trees at 98.1%, and KNNweighted at 98.1%. The top tree model is the Medium Tree at 97.8%, and the leading neural network model is the Medium NN at 97.5%.

- For GS alone: The Ensemble Subspace KNN leads with 98.4%, followed closely by SVM Linear at 98.3%, Bagged Trees at 98.3%, and SVM Quadratic at 98.2%. The best tree model is the Medium Tree at 98%, and the top neural network model is the Medium NN at 96.8%.

- For GLCM alone: The highest scores are SVM Quadratic at 80.5%, Trilayered NN at 80.1%, Bilayered NN at 80.1%, SVM Fine Gaussian at 76.8%, Ensemble Subspace KNN at 75.9%, and Bagged Trees at 75.9%. The leading tree model is the Fine Tree at 71.2%.

These models, which are the top performers in the internal testing, will be applied to several other tests.

5.3 External Testing on Treated Spikes

The objective of this second scenario was to assess the performance of the top models across various classes. To guide this evaluation, we formulated reasonable hypotheses about the test groups.

For testing purposes, the green caryopses from treated spikes were considered healthy while yellow caryopses from treated spikes were considered infected.

Based on this premise were conducted two separate tests: one using the *Pixel ROI Dataset*, selected for its extensive data volume, and the other employing the *Caryopsis*

Dataset, known for providing a more accurate representation of the average state of the caryopses, thus ensuring the precision of the labeling for each observation.

5.3.1 External Testing: *Pixel ROI Dataset*

The models with the best performance were:

- GS+GLCM: Boosted Trees at 83.9%.
- GS: Ensemble Subspace KNN at 84.2%.
- GLCM: Quadratic SVM at 68.7%.

Observations indicate that algorithms trained with GS features outperform others, achieving a peak accuracy of 84.2% in classifying treated regions. This suggests that incorporating GLCM features might reduce performance, although the minor differences make this hypothesis speculative and uncertain.

5.3.2 External Testing: *Caryopsis Dataset*

The top-performing models for the *Caryopsis Dataset* were:

- GS+GLCM: Weighted KNN at 91.5%.
- GS: Medium NN at 87.7%.
- GLCM: Fine Gaussian SVM at 79.2%.

Contrary to the previous dataset, the highest accuracy achieved in treated regions for this dataset was an impressive 91.5%. This difference, in comparison to the earlier testing scenario, could suggest that the selected ROI size may not ideally leverage GLCM features, although this hypothesis remains speculative.

5.3.3 Conclusive Considerations on External Testing

Defining the infected class as 'Positive' and the healthy class as 'Negative' allows us to present the Confusion Matrix for the most effective algorithm under each testing condition in Tables 5.3 - 5.9.

The results indicate that while GS features are generally more informative, integrating GLCM features effectively requires careful consideration, especially in selecting ROI sizes.

Further analyses are necessary to better evaluate performance using different metrics. Additionally, expanding the dataset, particularly for the *Caryopsis Dataset*, is essential for achieving statistically significant results.

Exploring various GLCM input values to identify the optimal GS-GLCM feature combination could also enhance ML model performance.

Table 5.3: Confusion Matrices of the best Internal Testing Model (Ensemble Subspace KNN with GS features).

Table 5.4: Absolute Values

	Real Positive	Real Negative
Predicted Positive	1187	15
Predicted Negative	18	867

Table 5.5: Percentage Values

	Real Positive	Real Negative
Predicted Positive	98.5%	1.7%
Predicted Negative	1.5%	98.3%

Table 5.6: Confusion Matrices of the best External Testing Model on the *Pixel ROI Dataset*(Ensemble Subspace KNN with GS features).

Table 5.7: Absolute Values

	Real Positive	Real Negative
Predicted Positive	3296	507
Predicted Negative	665	2975

Table 5.8: Percentage Values

	Real Positive	Real Negative
Predicted Positive	83.2%	14.6%
Predicted Negative	16.8%	85.4%

Table 5.9: Confusion Matrices of the best External Testing Model on the *Caryopsis Dataset*(Weighted KNN with GS+GLCM features).

Table 5.10: Absolute Values

	Real Positive	Real Negative
Predicted Positive	96	4
Predicted Negative	14	98

Table 5.11: Percentage Values

	Real Positive	Real Negative
Predicted Positive	87.3%	4.0%
Predicted Negative	12.7%	96.0%

Conclusions

Some machine learning models were identified as potentially useful for detection the presence of FHB in wheat spikes starting with non-destructive measure collected with a hyperspectral instrument.

A field experiment was conducted at The University of Pisa to determine the effectiveness of different treatments on two different varieties of wheat: Cultivar Bingo, susceptible to the pathogen known as Fusarium Graminearum, and Cultivar Rebelde, resistant to the infectious fungus. Different plots were created and the experimental site was divided into region with different conditions of pathology and treatment. From these plots, sixteen spikes were extracted for each of the twelve total groups. Some samples from each plot were extracted from the field. I used the "HyIce" spectrometer to acquire hyperspectral images of the spikes across 840 wavelengths.

I selected from the Cultivar Bingo variety four main groups to analyze: the healthy spikes without treatment, the spikes infected with the Fusarium Graminearum, the spikes infected but treated with the Systemic treatment and the spikes infected but treated with the Trichoderma treatment. I processed the acquired images related to these groups by using algorithms developed with MATLAB code. First, I removed background noise, estimated from the signal obtained in the absence of the sample. Then, I obtained reflectance by calculating the ratio between the radiance reflected by the spikes and the signal reflected from a Lambertian surface. Finally, the individual caryopses of the spike were identified using binary masks to allow analysis on specific regions of the spike.

I employed RGB analysis to distinguish the dataset in eight different groups based on the group of the spike and the color of the caryopses. The green caryopses of the uninfected spikes are identified control for the healthy class, while the yellow caryopses of the spikes inoculated with the pathogen and untreated are labeled as infected.

Based on this separation I analyzed the reflectance signals. Healthy spikes, unlike the infected ones, show a local minimum around the wavelength of 662 nm. This value is associated with one of the absorption peaks of chlorophyll a. Infected spikes, due to the infection, are characterized by a reduction in the concentration of pigments such as chlorophyll, which is usually high when the vegetation is healthy. Subsequently, I analyzed the hyperspectral images using the phasor method, based on the application of the Discrete Fourier Transform (DFT), to quickly and easily identify spectral differences associated with different categories of spikes. Specifically, by applying the DFT algorithm on spectral windows of 20, 15, 7, and 3.5 nm within the original spectrum, I identified certain signal regions where pixels of spikes belonging to different classes end up in separate positions on the phasor plane. Healthy and infected pixels were located in different positions of the phasor plane, while the treated spikes are located between the healthy and unhealthy groups, suggesting some portions are infected while other are effectively cured by the treatment.

I computed the centroids of points projected onto the phasor plane across various spectral regions using Matlab. This approach enabled me to recognize windows where the distances between centroids of distinct groups are largest. Through this analysis, I identified ten spectral regions of utmost interest, suggesting them as features for creating an automatic classification method. These identified spectral ranges span from 500nm to 700nm, aligning with findings reported in the literature.

Furthermore, I also conducted an analysis of spatial and texture properties. To extract texture properties, I chose to select grayscale images associated with the windows identified in the spectral analysis. For each of these, the GLCM (Gray-Level Co-Occurrence Matrix) algorithm was applied, enabling the calculation of some image texture properties (contrast, correlation, energy, entropy, and homogeneity) corresponding to the most significant spectral regions.

Subsequently, I used non-parametric statistical tests on these measures. The results of these tests showed significant differences (P-value$<$0.01) for samples from different classes, except for correlation, which was so excluded as feature for the algorithms.

Finally, I trained and tested various machine learning algorithms using different feature configurations. I partitioned the dataset using two distinct methods: the *Pixel ROI approach*, where each Region of Interest (ROI) identified through GLCM analysis is treated as an observation and the *Caryopsis approach*, which involves calculating an average evaluation of both GLCM and GS features for each caryopsis, treating the caryopsis itself as the observation.

The final step, which is the objective of this work, is to train and test various machine learning algorithms under different conditions. This is done to determine which algorithm most effectively differentiates between uninfected spike portions and infected ones independently. Due to the size of the dataset, the *Pixel ROI Dataset* has been chosen for training. The train involved two classes compatibly with the analysis: the green caryopses of the Cultivar Bingo without pathogen representing the healthy portion of the spike and the yellow caryopses of the Cultivar Bingo with pathogen inoculation representing the portion of spike with the disease.

The training set was composed by a random selection observations for these classes from the *Pixel ROI approach*. 80% of these observations provided the training set while the rest of the dataset is used for the first test. Several tests followed on the remaining data from both *the Pixel ROI approach* and the *Caryopsis approach* dataset.

The tests were conducted on models trained in three distinct ways using the same dataset: solely with GLCM features, solely with GS features, and by combining both GLCM and GS features. The models utilizing only GLCM features generally produced less accurate classifications, achieving a maximum accuracy of 79.5% for identifying healthy caryopses in 'External Tests', and reaching up to 80.5% accuracy in 'Internal Tests'. In contrast, the method using only GS features significantly outperformed these results, achieving accuracies of 87.7% and 98.41% under the same testing conditions, respectively.

Additionally, there is a noteworthy pattern observed when examining tests on treated spikes: the algorithm trained with GS features demonstrates superior performance on the Pixel ROI Dataset, achieving an optimal accuracy of 84.2% when applied to treated areas. This suggests that GLCM features might be detrimental in models trained with both sets of features. However, the situation changes with the Caryopsis Dataset, where the highest performance peak is achieved by a model trained with all features, reaching 91.5% accuracy in 'External Tests'. This turns out to be the best result overall for the 'External Tests'. This observation may imply that the chosen ROI size might not be the most effective for optimizing GLCM features.

It is crucial to note that the models presented here are preliminary. Various models leveraging different features could be developed and assessed using the pipeline established in this study.

Future studies and experiments might explore adjusting the GLCM features to enhance their contribution to the ML models or could utilize the whole spectrum of GS coordinates to see if the accuracy is enanched or not.

In addition, some improvement on the evaluation of the ML performances could be done, by taking into account more representative metrics than the accuracy and normalize the feature in order to apply some feature-reduction techniques as the Principal Component Analysis (PCA).

Moreover, once optimized and developed, integrating this method with remote sensing analysis, automated monitoring systems and decision-support tools typical of Agriculture 4.0, could significantly improve the efficiency and sustainability of agricultural practices.

In conclusion, this study introduces an innovative methodology that utilizes both spectral information and spatial conformation of images to accurately identify diseased areas in wheat spikes. This approach offers a potentially effective methodology for representing and evaluating pathological conditions of the crops.

Bibliography

[1] O. B. C A. A. Gitelson Y. Zur and M. N. Merzlyak. “Assessing Carotenoid Content in Plant Leaves with Reflectance Spectroscopy”. In: *Photochemistry and Photobiology* 3 (2002).

[2] David Arthur and Sergei Vassilvitskii. “k-means++: The Advantages of Careful Seeding”. In: (2007), pp. 1027–1035. DOI: `https://dl.acm.org/doi/10.5555/1283383.1283494`.

[3] Asa Ben-Hur and Jason Weston. “A User’s Guide to Support Vector Machines”. In: *Methods in molecular biology* (2010). DOI: `https://doi.org/10.1007/978-1-60327-241-4_13`.

[4] D. Camuffo. *Microclimate for Cultural Heritage Conservation, Restoration, and Maintenance of Indoor and Outdoor Monuments.* 2nd ed. The MIT Press, 2014. DOI: `https://doi.org/10.1016/C2013-0-00676-7`.

[5] G. A. Carter and A. K. Knapp. “Leaf optical properties in higher plants: linking spectral characteristics to stress and chlorophyll concentration”. In: *American Journal of Botany* 88 (2001).

[6] D. Durgalakshmi et al. “Principles and Mechanisms of Green Photocatalysis”. In: *Environmental Chemistry for a Sustainable World* 34 (2019). DOI: `https://doi.org/10.1007/978-3-030-15608-4_1`.

[7] J. Williams F. C. McKenzie. “Sustainable food production: constraints, challenges and choices by 2050”. In: (2015). DOI: `http://dx.doi.org/10.1007/s12571-015-0441-1`.

[8] Mark D. Fairchild. *Color Appearance Models.* 2nd ed. John Wiley & Sons, Ltd, 2005.

[9] D. H. Foster and K. Amano. “Hyperspectral imagin in color vision research: tutorial”. In: *Journal of the Optical Society of America A* 36 (2019).

[10] S. Fowler, R. Roush, and J. Wise. *Concepts of Biology.* OpenStax, 2012.

[11] R. Garszonio et al. "A novel hyperspectral system for high resolution imaging of ice cores: Application to light-absorbing impurities and ice structure". In: *Cold Regions Science and Technology* 155 (2018). DOI: https://doi.org/10.1016/j.coldregions.2018.07.005.

[12] R. M. Haralick. "Statistical and Structural Approaches to Texture". In: *IEEE* 67 (1979). DOI: https://doi.org/10.1109/PROC.1979.11328.

[13] R. Hermosa et al. "Plant-beneficial effects of Trichoderma and of its genes". In: *Cold Regions Science and Technology* 155 (2012). DOI: https://doi.org/10.1099/mic.0.052274-0.

[14] C. C. Hung, E. Song, and Y. Lan. *Image Texture Analysis Foundations, Models and Algorithms.* Springer, 2019.

[15] Gareth James et al. *An Introduction to Statistical Learning with Applications in R.* 8th ed. Springer, 2017.

[16] Dan Jurafsky and James H. Martin. *Speech and Language Processing.* 1st ed. 2023.

[17] M. Magnusson et al. "Creating RGB Images from Hyperspectral Images Using a Color Matching Function". In: (2020). DOI: https://doi.org/10.1109/IGARSS39084.2020.9323397.

[18] Leonel Malacrida. "Phasor plots and the future of spectral and lifetime imaging". In: *Nature methods* 20 (2023), pp. 3965–967. DOI: https://doi.org/10.1038/s41592-023-01906-y.

[19] M. Pharr, W. Jakob, and G. Humphreys. *Physically Based Rendering, fourth edition: From Theory to Implementation.* 4th ed. The MIT Press, 2023.

[20] R. Prihatmanti, N. Taib, and F. S. Yeok. "Multi-Layer Balcony Planting: A Biomimetic Concept of Tropical Rainforest". In: (2019). DOI: https://doi.org/10.15405/epms.2019.12.17.

[21] E. F. Schubert. *Light-Emitting Diodes.* 2nd ed. Cambridge University Press, 2006.

[22] R. Scodellaro et al. "A novel hybrid machine learning phasor-based approach to retrieve a full set of solar-induced fluorescence metrics and biophysical parameters". In: *Remote Sensing of Environment* (2022). DOI: https://doi.org/10.1016/j.rse.2022.113196.

[23] I. Shafiq et al. "Crop photosynthetic response to light quality and light intensity". In: *Journal of Integrative Agriculture* 20 (2021). DOI: https://doi.org/10.1016/S2095-3119(20)63227-0.

[24] Michael Steinbach and Pang-Ning Tan. *The Top Ten Algorithms in Data Mining*. 1st ed. CRC, 2009.

[25] L. Taiz and E. Zeiger. *Plant Pèhysiology*. 3rd ed. Sinauer Associates, 2002.

[26] P. Trisolinao et al. *La Campagna di Misure PAMELA 2017 nel Mediterraneo Centrale*. ENEA, 2017.

[27] B. Sevastian V, A. Unnikrishnan, and K. Balakrishnan. "Grey Level Co-occurrence Matrices: Generalisation And Some New Features". In: *International Journal of Computer Science, Engineering and Information Technology (IJCSEIT)* 2 (2012). DOI: `https://doi.org/10.5121/ijcseit.2012.2213`.

[28] Alex Vallmitjana, Belén Torrado, and Enrico Gratton. "Phasor-based image segmentation: machine learning clustering techniques". In: *Biomedical Optics Express* 12.6 (2021), pp. 3410–3422. DOI: `https://doi.org/10.1364/BOE.422766`.

[29] Wei Zhang et al. "A Distributed Storage and Computation k-Nearest Neighbor Algorithm Based Cloud-Edge Computing for Cyber-Physical-Social Systems". In: *Preparation of Papers for IEEE TRANSACTIONS and JOURNALS* (2020). DOI: `https://doi.org/10.1109/ACCESS.2020.2974764`.

Appendix A: K-Means Method on the Phasor Plane for Background Removal

In this Appendix, we analyzed a 'Cultivar Bingo' spike that was subjected to pathogen inoculation and systemic treatment.

Upon examining the phasor plane, we observed that the pixels were distinctly positioned in two clusters, as illustrated in Figure 5.3.2a.

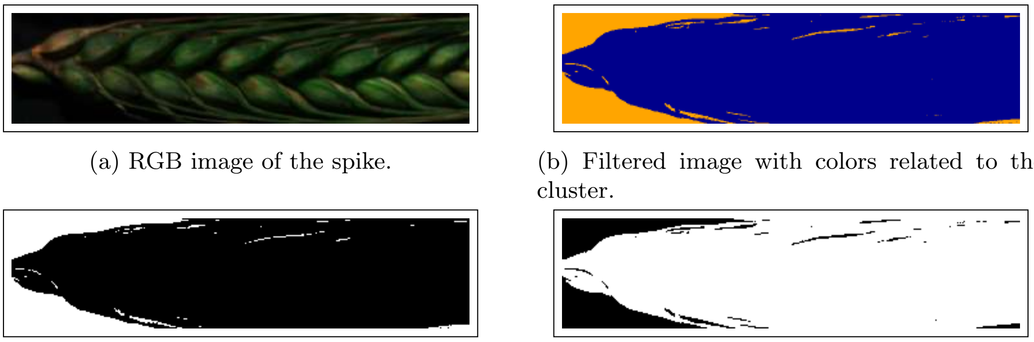
Therefore, we employed the built-in k-means algorithm in MATLAB to separate the points on the plane into two clusters, as shown in Figure 5.3.2b.

Following this, we applied a color filter based on the k-means separation to the original image, reflecting the cluster division, as shown in Figure 5.3.1b.

This effective separation enabled us to create two masks: one representing the background and the other delineating the spike, as shown in Figures 5.3.1c and 5.3.1d.

Subsequently, we conducted a spectral analysis of the mean normalized reflectance (Rnorm) for both segments, as depicted in Figure 5.3.3.

This analysis demonstrated a qualitative agreement with the spectral data discussed in Section 4.6, highlighting the method efficacy in distinguishing between the background and the spike based on their spectral characteristics.



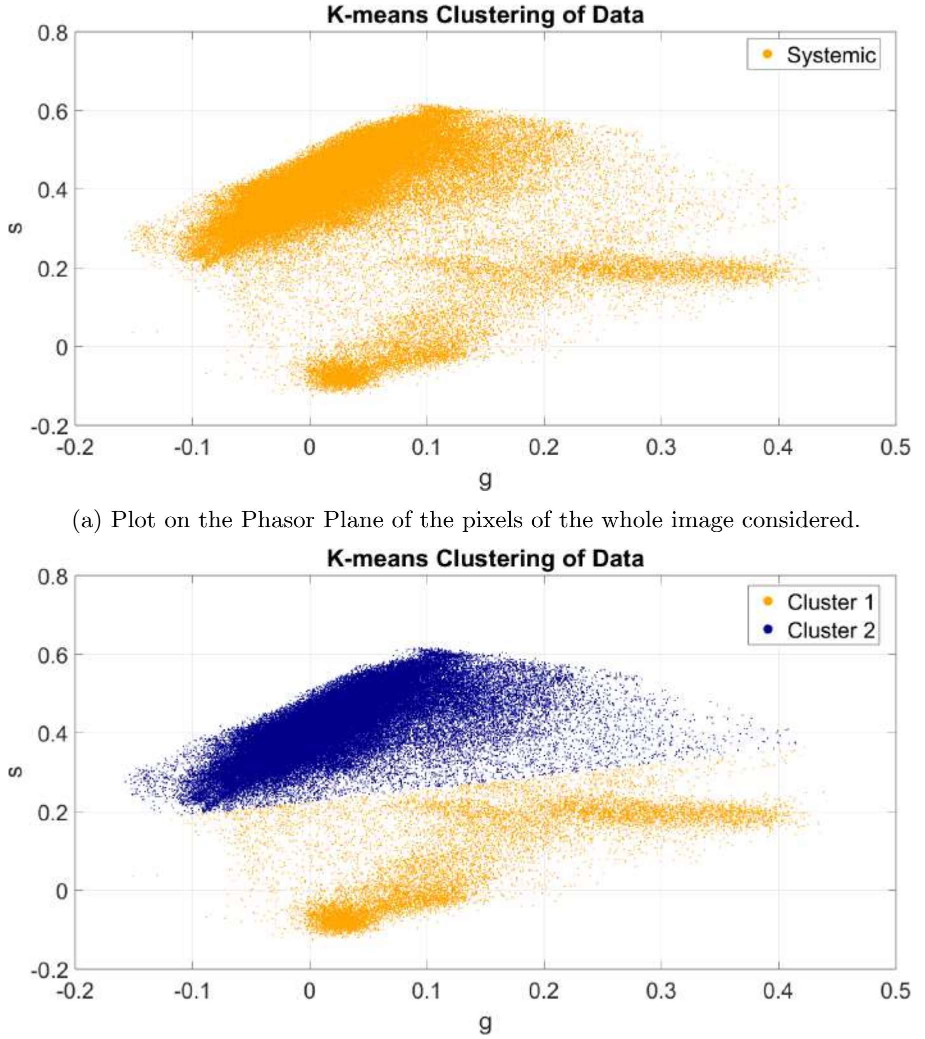
(a) RGB image of the spike.

(b) Filtered image with colors related to the cluster.

(c) Logical mask of the spike highlighing the background.

(d) Logical mask of the spike excluding the background.

Figure 5.3.1: The results of the k-Means Background Removal Algorithm on the selected spike from the Systemic treated group.



(a) Plot on the Phasor Plane of the pixels of the whole image considered.

(b) Plot of the different clusters found by the kMeans algorithm.

Figure 5.3.2: The results of the kMeans separation applied on the Phasor Plane. The method is applied on the whole spectrum and the (DTF) harmonic is k=1.

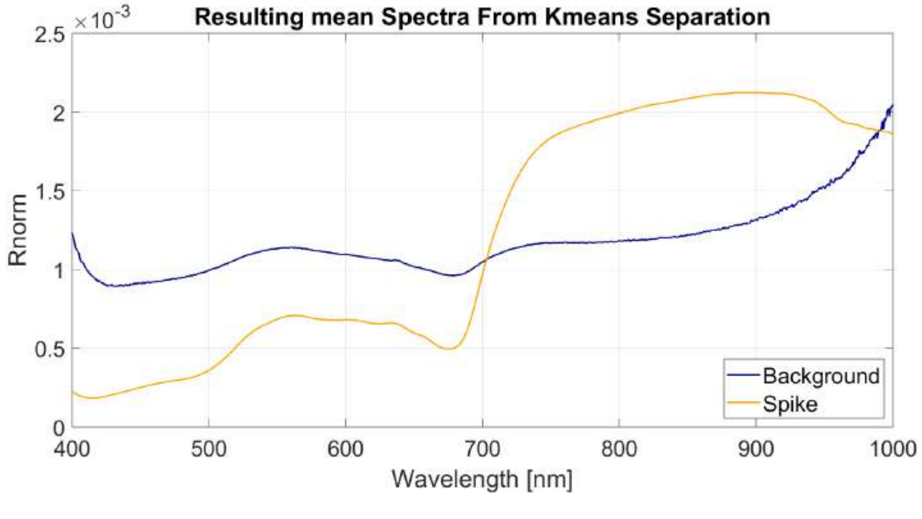


Figure 5.3.3: Plot on the mean spectra of the kmeans mask.