POLITECNICO DI TORINO

Department of Electronics and Telecommunications (DET)
School of Computer Engineering, Cinema, and Mechatronics

Master's degree in Mechatronic Engineering

Master's Degree Thesis

RGB and Thermal Camera Integration for Advanced Perception of an Agricultural Robot

Supervisors:

Prof. Marcello Chiaberge

Prof. Alba Perez

Co-supervisor:

Prof. David Caballero

Candidate:

Giovanna Guaragnella

Student: 295308

Academic year 2023/2024

A chi ha creduto in me fin dall'inizio.

Ai miei genitori, per il loro amore e i loro sacrifici.

Ai miei nonni, sempre con me.

Contents

List of Figures

List of Tables

Abstract

The intersection of agriculture and energy production offers unique opportunities for optimizing land use through innovative solutions. This thesis explores the development of a dual-camera system combining a thermal camera and an RGBD camera for monitoring plant health, focusing on water stress assessment which is a crucial factor for optimizing agricultural productivity. By integrating these advanced technologies, the project aimed to provide real-time measurements of water stress in crops, particularly in lettuce.

The research involved the design and implementation of a system that integrates the Optris Xi400 thermal camera with the Intel RealSense D457 RGBD camera. The successful integration of these technologies enabled simultaneous capture of thermal and visual data, which was used for real-time Crop Water Stress Index (CWSI) calculations. A key achievement of the project was the effective application of the YOLOv8 model for plant segmentation, enabling accurate and real-time analysis of crop conditions.

Results demonstrated that the system successfully visualized CWSI values, providing actionable insights into plant health and water stress levels. This approach not only supports efficient water management but also contributes to the broader goals of the SYMBIOSYST project, which aims to harmonize agricultural practices with photovoltaic energy production for sustainable land use.

Preface

The intersection of agriculture and energy production presents unique opportunities to optimize land use for both crop cultivation and photovoltaic (PV) energy generation. The SYMBIOSYST project aims to capitalize on these opportunities by developing innovative agri-PV solutions that harmonize these traditionally separate sectors. Launched in January 2023, SYMBIOSYST is a Horizon Europe Innovation Action that involves 18 partners from six European countries, including the United Kingdom, and will run until December 2026. The project demonstrates various PV solutions in open field and greenhouse agriculture across four scenarios in three countries, emphasizing sustainability, social acceptance, and technological advancement.

A critical component of SYMBIOSYST is ensuring that PV installations do not negatively impact crop yields while promoting sustainable farming practices. This involves adopting advanced monitoring and control systems, digitalizing farming tools, enhancing water management, mitigating climate change impacts, and engaging local organizations to ensure biodiversity and community benefits.

In the context of this ambitious project, my research focuses on developing a dual-camera system to monitor plant health, specifically targeting water stress levels. Water stress is a significant factor affecting crop productivity, and early detection is crucial for timely intervention. By integrating computer vision technologies with thermal imaging, this system aims to provide real-time, precise monitoring of plant water stress, thus contributing to the overall goals of SYMBIOSYST.

Introduction

Researchers and engineers have been dedicating approximately six decades to the pursuit of enabling machines to perceive and comprehend visual information.

In particular, the advancements in Computer Vision (CV) have found significant applications in the field of robotics and environmental monitoring, enhancing their capabilities and enabling groundbreaking solutions. In the domain of robotics, computer vision has ushered in a new era of automation and intelligence. Robots equipped with advanced vision systems can navigate complex environments autonomously, recognize objects, and interact with the surroundings.

Additionally, in the realm of environmental monitoring, computer vision plays a crucial role in collecting and analyzing data for ecological studies and climate research. Not only unmanned aerial vehicles (UAVs) equipped with high-resolution cameras and computer vision algorithms can monitor vast ecological landscapes, detect changes in vegetation, and assess environmental health, but also agricultural autonomous robots (AMRs) have the potential to tackle numerous challenges faced by farmers, ranging from labor shortages to precision farming requirements. These advancements have significantly improved our ability to understand and address environmental challenges, including deforestation, climate change, biodiversity conservation and, water shortage.

Nevertheless, there remains significant work ahead in the advancement and seamless integration of diverse sensors essential for endowing autonomous robots with robust perception capabilities. These capabilities are crucial for observing crops and conducting precise measurements, thereby offering critical decision support to agronomists in the agricultural sector. This endeavor entails not only selecting the optimal technologies for

environmental monitoring but also crafting effective methodologies for data analysis to attain predefined objectives.

The core objective of this thesis is to explore the potential of integrating two advanced camera systems—the Optrix Xi400 thermal camera and the Intel RealSense D457 RGB camera—mounted on a mobile robot to advance the state-of-the-art in object recognition, localization, and inspection for crop monitoring.
In particular, by combining computer vision with thermal imaging, the system developed in this research provides real-time monitoring of plant water stress, thereby contributing to the overall goals of SYMBIOSYST.

This project involves a series of interrelated tasks aimed at achieving several key goals:

- Integration and alignment: Focuses on the successful integration and alignment of the thermal and RGB cameras to ensure that they work together effectively. This alignment is critical for accurate and synchronized data capture, which is the foundation for subsequent analysis.
- Calibration process: Involves a meticulous calibration process to ensure that the data from both cameras are accurately synchronized, enabling precise image fusion and analysis.
- Image segmentation: Following calibration, the project will implement advanced image segmentation techniques on RGB images to facilitate the fusion of these images with thermal data. This step is crucial for creating a comprehensive dataset that combines visual and thermal information for improved object recognition and inspection.
- Data acquisition: To train the machine learning models effectively within the specific domain of agricultural applications, a dedicated dataset was constructed. This

involved data acquisition to build a robust dataset tailored for the training of the neural network.

- Assessment of water stress levels: The ultimate aim of these efforts is to develop a robust system capable of assessing water stress levels in plants, which is vital for optimizing agricultural practices.
- Machine Learning application: By leveraging the combined data from both camera systems and applying machine learning techniques for image segmentation and analysis, the project seeks to contribute valuable insights into plant health and water management strategies.

The thesis is organized as follows:

- *Chapter 2*: Theoretical background
  This chapter provides an analysis of the theoretical background, reviewing existing technologies and methodologies relevant to our study.
- *Chapter 3*: State of the art - Visual robotic systems in agricultural context
  This chapter discusses the state of the art in visual robotic systems within the agricultural context.
- *Chapter 4*: System definition
  Here, the system definition is presented, detailing the primary components of the system, including data acquisition sensors and the overall hardware setup design.
- *Chapter 5*: Methodology
  This chapter describes the developed methodology, outlining the steps taken to implement the system.
- *Chapter 6*: Results
  In this final chapter, the obtained results are detailed and explained, providing an evaluation of the system's performance and effectiveness based on the conducted experiments and analyses.

A graphical representation of the steps followed in this project is shown in the following Figure.

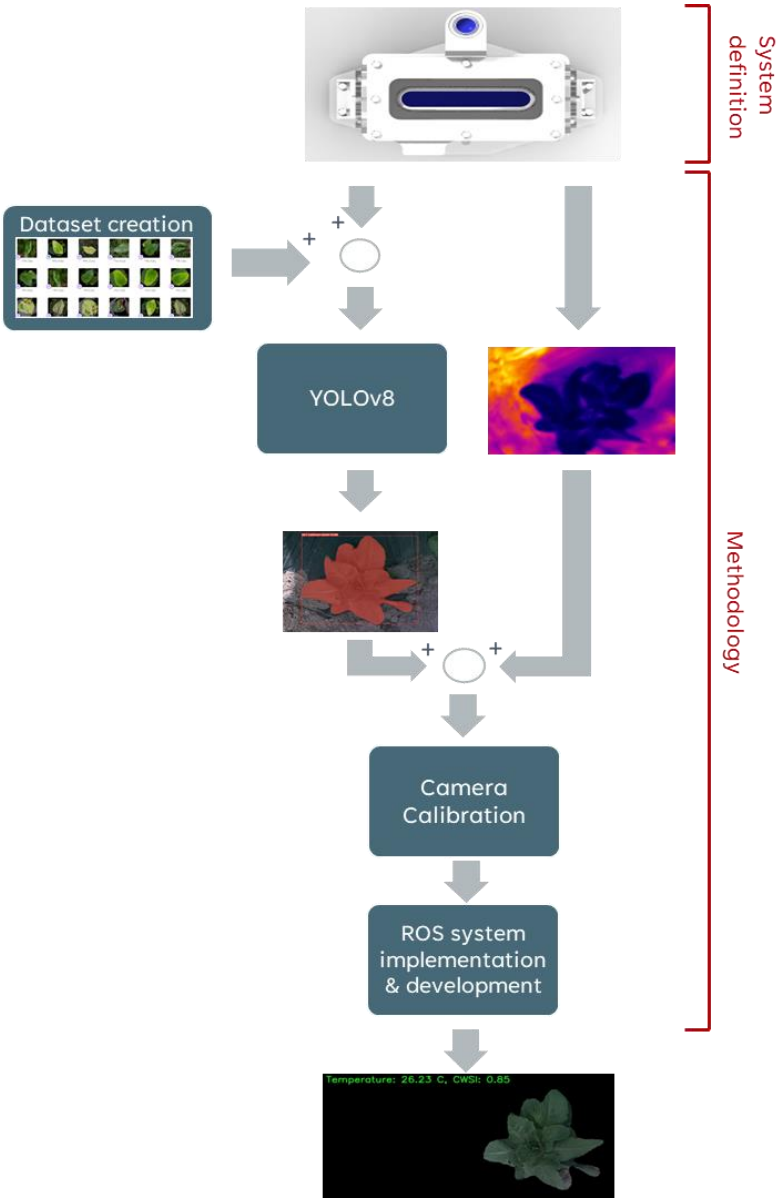


Figure 1 - Project explanation

Chapter 1.

Theroretical background

In this chapter, an overview of the theoretical aspects essential for the development of the research project is provided. The chapter begins by introducing the field of computer vision, a multidisciplinary domain focused on enabling computers to interpret and understand visual information from the world.

Following this, the chapter explores the fundamentals of machine learning, a branch of artificial intelligence that allows systems to learn from data and improve their performance over time without explicit programming.

The discussion then moves on to Deep learning, a subfield of Machine Learning that employs deep neural networks to extract complex features from data. Deep Learning has driven significant advancements in computer vision, enabling more sophisticated applications for image recognition, classification, and analysis.

Finally, the chapter introduces YOLOv8 (You Only Look Once version 8), one of the most advanced architectures for object detection and segmentation. YOLOv8 is designed to deliver high-speed and high-accuracy object detection capabilities, making it a powerful tool for real-time applications. In the research project, YOLOv8 was employed for image segmentation tasks due to its efficiency and precision in detecting and segmenting objects within images.

Through this theoretical overview, the chapter aims to lay the groundwork for understanding the technical and methodological choices made in the research project.

1.1 Computer Vision

Computer vision pertains to the realm of endowing machines with the capability to perceive visual data. It employs a combination of cameras and computational systems to detect, monitor, and analyze objects, aiming to replicate or surpass human visual perception using automated systems. It involves retrieving, interpreting, and comprehending valuable information from images, typically through algorithmic processing.

From a perspective rooted in biological science, computer vision endeavors to construct computational models mirroring the intricacies of the human visual system. Conversely, from an engineering standpoint, the objective of computer vision is to fabricate autonomous systems capable of executing tasks similar to those performed by the human visual system, often surpassing its capabilities in various scenarios. Many tasks within the realm of vision involve extracting three-dimensional and temporal information from dynamically changing two-dimensional data, typically captured by one or more cameras, with a broader aim of comprehending such dynamic scenes [1].

Figure 2 - Example application of computer vision system [2]

The pursuit of these dual objectives is closely intertwined. Insights gathered from the properties and behaviors of the human visual system frequently inspire engineers in the design of computer vision systems. Hence, algorithms developed within the domain of computer vision can offer valuable perspectives on the workings of the human visual system.

1.1.1 History and evolution

The history of computer vision traces back to the 1950s when neurophysiologists conducted experiments involving the presentation of various images to a cat, aiming to observe corresponding neural responses. Remarkably, the findings revealed that the cat's brain exhibited initial reactions to distinct features, particularly hard edges and lines. Scientifically, this discovery meant that the initial stages of image processing involved the recognition of elementary shapes, such as straight edges, marking a significant milestone in the understanding of visual cognition [3]. A widely acknowledged figure in the field of Computer Vision is Larry Roberts, credited as a pioneering force. During his doctoral studies around 1960 at MIT, Roberts explored the potential of deriving three-dimensional geometric data from two-dimensional perspective views of objects, particularly blocks or polyhedra [4]. By the 1970s, the first commercial use of computer vision involved interpreting handwritten or typewritten text through optical character recognition, enabling text interpretation for the visually impaired. Consequently, extensive research efforts were directed towards what are termed as "low-level" vision tasks, such as edge detection and segmentation. A significant breakthrough occurred with the framework introduced by David Marr around 1978 at MIT. Marr's approach, characterized by a bottom-up methodology, revolutionized the comprehension of scenes within the field [5].

The advent of the Internet in the 1990s facilitated the availability of vast amounts of online images for analysis, leading to the development of facial recognition programs.

Consequently, the increase in data volume fueled advancements in machines capable of identifying specific individuals across photo and video media.

Today, a convergence of factors revitalizes enthusiasm for computer vision. The omnipresence of mobile devices sporting built-in cameras floods society with a big amount of visual content. Simultanooeously, the accessibility and affordability of computing power facilitate widespread experimentation and implementation. Moreover, specialized hardware designed explicitly for computer vision tasks is becoming more prevalent, further fueling advancements. By comprehending the fundamental principles of visual cognition, researchers developed sophisticated algorithms that enable computers and systems to extract meaningful information from digital images, videos and other visual inputs [6].

For instance, the emergence of cutting-edge algorithms like convolutional neural networks capitalizes on both hardware and software capabilities, amplifying the potential for sophisticated visual processing.

One of the driving factors behind the growth of computer vision is the amount of data we generate today that is then used to train and make computer vision better. These advancements have yielded remarkable effects in the field of computer vision. In less than a decade, object identification and classification accuracy levels surged from 50 to 99%.

Moreover, contemporary systems exhibit greater precision than the human eye in swiftly detecting and reacting to visual inputs.

1.1.2 How it works and applications

The landscape of computer vision resembles that of solving a puzzle. Just as pieces together scattered puzzle tiles to form an image, neural networks for computer vision operate on a similar principle. They discern the myriad elements constituting an image, identify edges, and then model subcomponents. Subsequently, through filtering and layer-by-layer actions within the network, they assemble all elements akin to completing a puzzle.

Computers lack the final image, typically depicted on the puzzle box, but are instead fed hundreds, or thousands, of related images for training to recognize specific objects.

To recognize a cat, instead of programming computers to search for whiskers, tails, and pointed ears, programmers input millions of cat photos. This method enables the model to autonomously learn to distinguish the various features comprising a cat.

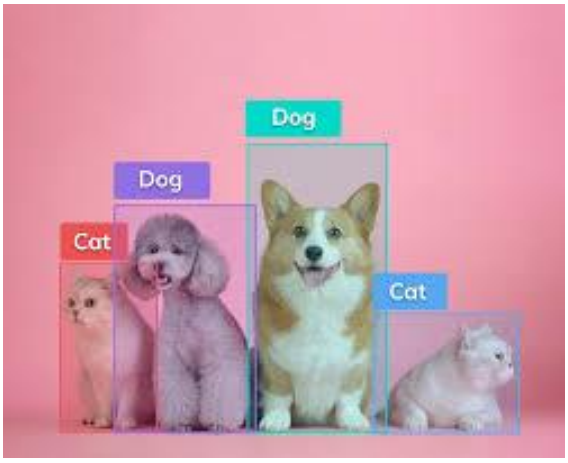


Figure 3 - Animal recognition [7]

The realm of computer vision transcends human visual capabilities across various domains, from facial recognition to analyzing gameplay actions during soccer matches.
The combination of deep learning and computer vision has revolutionized the field, enabling machines to "see" and interpret visual data with unprecedented accuracy and efficiency.
Hence, Deep learning is very effective for computer vision being faster and easier to develop. Therefore, it allows to develop or choose a preconstructed algorithm and train it with examples of objects it must detect.

Applications of computer vision span across various domains and are seamlessly integrated into everyday products.
In the realm of self-driving cars, computer vision facilitates real-time analysis of surroundings, enabling vehicles to navigate roads, recognize traffic signs, and detect

obstacles and pedestrians. For example, Tesla's self-driving cars employ multiple cameras to assess their environment, allowing for advanced functionalities like autopilot. These 360-degree cameras use computer vision to identify and categorize objects around the vehicle [8]. Facial recognition technology utilizes computer vision to authenticate users on consumer devices, tag individuals on social media, and aid law enforcement agencies in identifying criminals. In manufacturing, computer vision is used for AI-powered inspection systems, predictive maintenance, quality control, and automation. It helps in detecting machinery breakdowns and product defects, and in automating assembly processes, especially for delicate items like electronics. Computer vision aids in medical diagnostics by enhancing image analysis in fields like pathology, radiology, and ophthalmology. For example, it is useful to detect cancerous moles in skin images and identifying symptoms in medical scans [9].

Machine learning uses algorithm-based models to help computers learn context through visual data analysis. Once provided with sufficient data, the model can see the "big picture" and differentiate between various visual inputs. Instead of being programmed to recognize and distinguish images, the machine uses AI algorithms to learn autonomously.

Convolutional neural networks assist ML models in "seeing" by breaking images down into pixels. Each pixel is labeled, and these labels are used to perform convolutions, a mathematical process that combines two functions to produce a third. Through this process, CNNs can process visual inputs. To interpret images like a human, neural networks execute convolutions and check the accuracy of the output through numerous iterations. Similar to how humans discern distant objects, a CNN starts by identifying basic shapes and edges, then fills in the data gaps and iterates its output until it accurately predicts the result.

In the next section, a more detailed view of ML and its evolution will be presented.

1.2 Machine learning

Machine learning (ML) is a branch of artificial intelligence (AI) that focuses on developing algorithms capable of improving automatically over time and making predictions based on data [10].

When thinking of AI and ML, we tend to imagine something very contemporary, something that has appeared recently. The truth is that the history of machine learning is a rich and evolving narrative, tracing back to the mid-20th century when foundational ideas about artificial intelligence began to take shape. The history of machine learning is marked by significant milestones and the contributions of pioneering researchers.

Alan Turing's seminal work in the 1950s laid the theoretical groundwork, proposing that machines could simulate any process of formal reasoning, a concept that would later underpin the development of machine learning algorithms [11]. In 1952, Arthur Samuel coined the term "machine learning" with his development of a checkers-playing program that improved through experience [12]. The 1960s saw Frank Rosenblatt's introduction of the Perceptron, an early neural network capable of binary classifications, which laid the groundwork for future neural network research [13]. The 1980s marked a significant leap with the rediscovery of the backpropagation algorithm by Geoffrey Hinton and colleagues, enabling the efficient training of deep neural networks [14]. The 1990s brought forth support vector machines [15] and ensemble methods like boosting, which improved model robustness and accuracy. The 21st century witnessed a data explosion and advances in computational power, facilitating the rise of deep learning. Landmark achievements such as AlexNet's triumph in the 2012 ImageNet competition [16] and AlphaGo's victory over a human Go champion in 2016 [17] demonstrated the immense potential of deep learning and reinforcement learning. More recently, transformer-based models like BERT and GPT have revolutionized natural language processing, enabling machines to understand and generate human-like text with unprecedented accuracy [18].

As machine learning continues to advance, ethical considerations and the pursuit of fairness and transparency have become paramount, ensuring that these powerful technologies are developed and deployed responsibly. This historical journey reflects the continuous innovation and interdisciplinary collaboration that drive the field of machine learning forward.

Throughout history, machine learning techniques have been applied to various computer vision tasks, revolutionizing the field and enabling groundbreaking advancements. From early attempts at pattern recognition to the development of sophisticated deep learning models, machine learning has played a crucial role in enhancing computer vision capabilities. Tasks such as image classification, object detection, segmentation, and facial recognition have benefited from the application of machine learning algorithms. These techniques have enabled computers to interpret and understand visual data with increasing accuracy and efficiency, paving the way for applications in areas such as medical imaging, autonomous vehicles, surveillance systems, and augmented reality. By leveraging the power of machine learning, researchers and practitioners continue to push the boundaries of what is possible in computer vision, driving innovation and shaping the future of technology.

1.2.1 Machine learning paradigms

Machine learning is broadly categorized into three main types: supervised learning, unsupervised learning, and reinforcement learning.

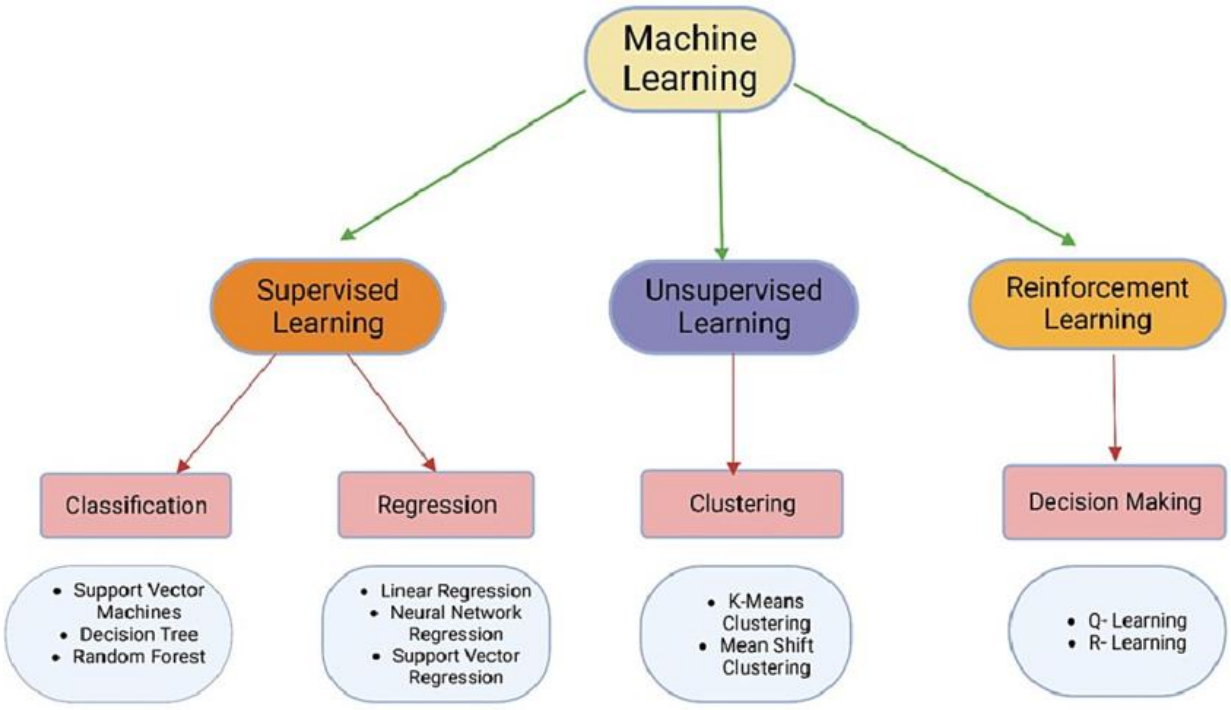


Figure 4 - Types of Machine Learning [19]

Supervised learning involves training models with labeled data, enabling them to perform classification and regression tasks. In contrast, unsupervised learning seeks to uncover patterns and relationships within unlabeled data, often through clustering methods. Reinforcement learning, on the other hand, focuses on enhancing model performance through continuous interaction with an environment and learning from the outcomes of its actions.

The following sections will explore each type in detail, discussing their methodologies and applications.

Unsupervised learning

Unsupervised learning is a method used to identify patterns and relationships in unlabeled data, often employed to form groups or clusters.

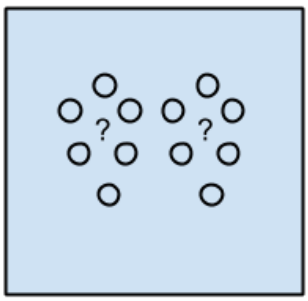


Figure 5 - Unsupervised learning algorithm [20]

The system is characterised by the lack of labels given to the learning algorithm. Moreover, input data does not have a known result. It is used when we want to find natural groups in data or whenever datasets are large, and it is expensive to assemble all data [20].
Example problems are clustering, dimensionality reduction and association rule learning.

Consider an email marketing campaign. Your dataset might include details about recipients, such as their past purchasing behavior, the last time they visited a website, and their average purchase amount. Without predefined customer groups, you can use unsupervised learning to analyze this behavioral data and automatically cluster customers into distinct groups. A key advantage of this approach is that it doesn't require prior knowledge of the group's structure - the clusters are generated based on the data itself. Once these groups are formed, you can label them with business-relevant terms and decide which customer segments to target in your email campaign.

Clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar to each other than to those in other groups [21]. These classes should have high intra-class similarity and low inter-class similarity. Clustering methods are typically organized by the modelling approaches such as centroid based and hierarchal. All methods are concerned with using the inherent structures in the data to best

organize the data into groups with the bigger similarities. The most important division in clustering is between hierarchical algorithms and partitional algorithms. Hierarchical clustering is characterised by the creation of a hierarchical decomposition of the set of objects using some criterion. An advantage of this kind of clustering is the fact that there is no need to specify the number of classes, but it does not scale well and the interpretation of results is subjective based on different criteria of division. Partitional clustering, differently from hierarchical clustering, sets the number of clusters in advance and puts each data in one of the clusters given.

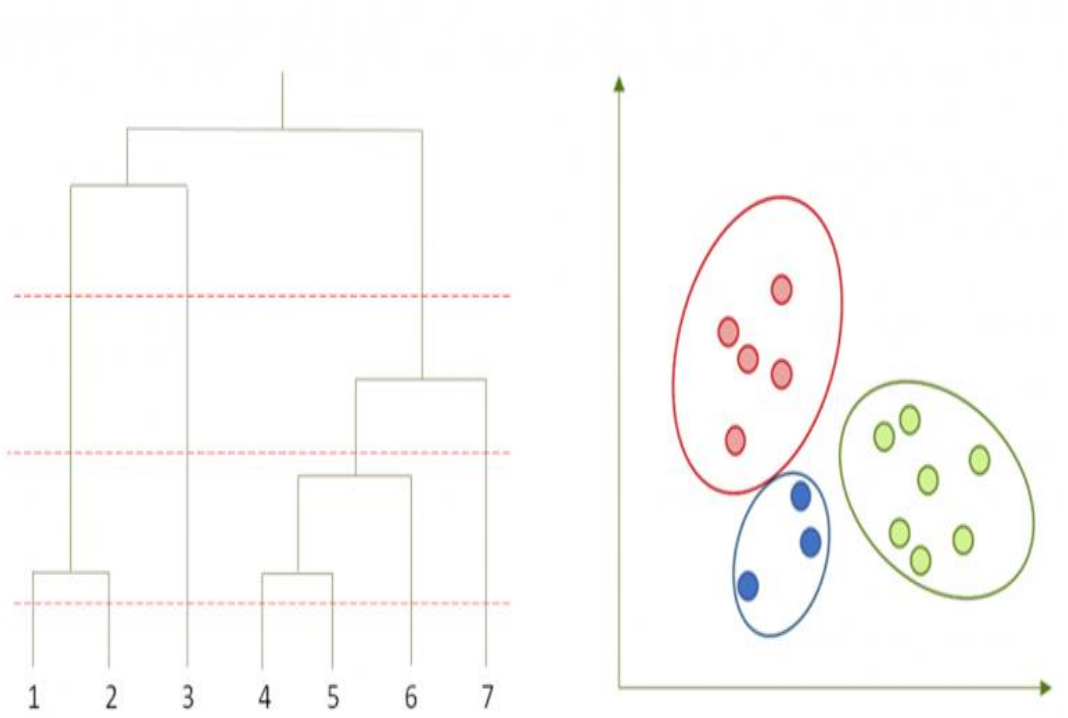


Figure 6 - Hierarchical and non-hierarchical Clustering

Supervised learning

Supervised learning is a fundamental approach in data science used to develop models that can predict outcomes based on labeled datasets. Essentially, labeled data consists of various features (variables) paired with a corresponding output that the model aims to predict.

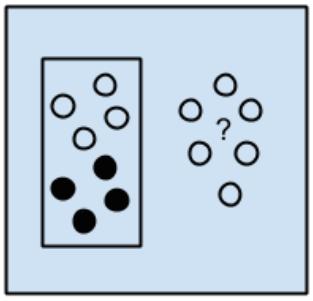


Figure 7 - Supervised learning algorithm [20]

The system receives input data, called training data, with known labels or results.

Example problems are classification and regression.

For instance, consider an ML model designed to identify whether fruits are apples or bananas. In this scenario, the label would be either "apple" or "banana," while the feature set might include attributes such as weight, length, width, and other relevant measurements of the fruits.

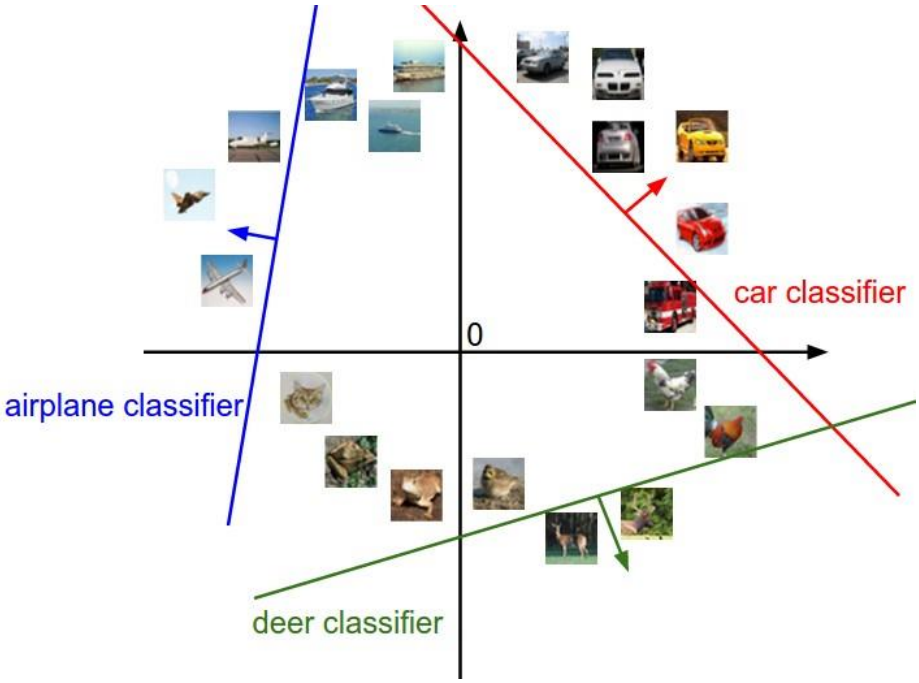


Figure 8 - Example of Linear Classifier [22]

Due to the presence of labels, it is possible to correct an algorithm in order to perform better on data. This kind of learning is characterized by two phases: the learning phase and the testing phase. The learning phase consists of a training process that prepares the model by making predictions and corrects the outliers with the cost function whenever the predictions are wrong. The training process continues until the model achieves a desired level of accuracy on the training data. The testing phase uses unseen data in order to measure the performances of the algorithm. An example of classifier is the linear classifier. It makes a classification considering a linear combination of the characteristics. The decision boundaries, which are the region of a problem space in which the output label of a classifier is ambiguous, in the feature space are linear (red, green and blue lines in *Figure 8*). This type of classifier works better when the problem is linearly separable.

Reinforcement learning

Reinforcement learning is a method that uses a reward-based system to provide training feedback. The learning process occurs as a machine, or Agent, that interacts with an environment and tries various strategies to achieve a specific goal. The Agent receives rewards or penalties based on whether it reaches a desirable or undesirable state [23].

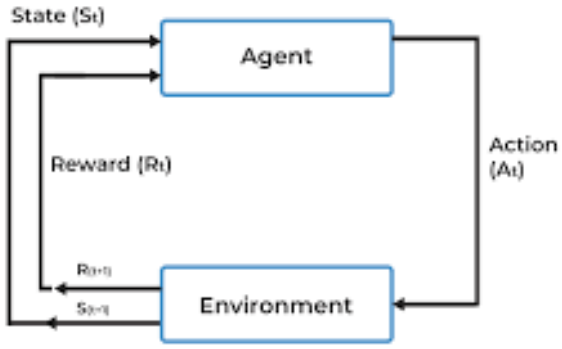


Figure 9 - Reinforcement learning algorithm [24]

Through this feedback, the Agent learns which actions lead to positive outcomes and which should be avoided. Success is measured using a score (often referred to as Q, hence the term Q-learning), allowing the Agent to iteratively improve its performance to achieve higher scores. Substantially, it acquires knowledge through trial and error carrying out actions with the goal of maximizing rewards, essentially learning through practice to attain the best possible results.

A practical example of reinforcement learning is controlling a car on a winding road. The Agent monitors its current state by measuring speed, direction relative to the road, and distances to the road's edges. It can then take actions like steering, accelerating, or braking to alter its state.

Rewards are given for desired behaviors, such as staying in the middle of the road and completing the course, while penalties are imposed for crashing or moving too slowly.

Effective reinforcement learning strikes a balance between short-term and long-term rewards, helping the car to both avoid collisions and reach its destination.
It is a good technique to use for automated systems that have to make a lot of small decisions without human guidance. Examples of applications of reinforcement learning include robotics, autonomous driving, gaming.

Comparison between Supervised, Unsupervised and Reinforcement learning

As shown before, Supervised learning and Unsupervised learning have many substantial differences which make them useful or not, based on the data you have to consider. In supervised learning, you train the machine using well-labelled data. This means that some data is already tagged with the correct answer. As a matter of fact, by making the algorithm learn from the training data you can predict outcomes from unforeseen data with highly accurate and trustworthy methods. Unsupervised algorithms, on the contrary, are used against not labelled data, giving less accurate results. Another important difference lies in the variables and number of the given classes. While in the supervised learning model input and output will be specified, just as the number of classes, in the unsupervised learning only input data will be provided and the number of classes will not be known. Due to all these observations, it is possible to notice that unsupervised learning, therefore Clustering, has big limits. The main drawback is that you cannot get precise information regarding data sorting. In other words, since clustering is based on the similarity between data, it tries to draw inference from the data such as finding patterns or clusters causing ill posed problems.
Reinforcement learning differs from supervised learning in a way that in supervised learning the training data has the answer key with it. This means that it does not require labeled data, nor does it use an unlabeled dataset like unsupervised learning. Instead, it continuously optimizes outcomes based on past experiences and creates new data with each attempt. In the absence of a training dataset, it is bound to learn from its experience.

1.3 Deep learning architectures for object detection

Deep learning is a subset of machine learning that involves the use of neural networks with multiple layers to learn intricate representations of data. It has gained widespread popularity in recent years due to its ability to automatically discover and extract features from raw data, leading to remarkable advancements in various fields, particularly computer vision.

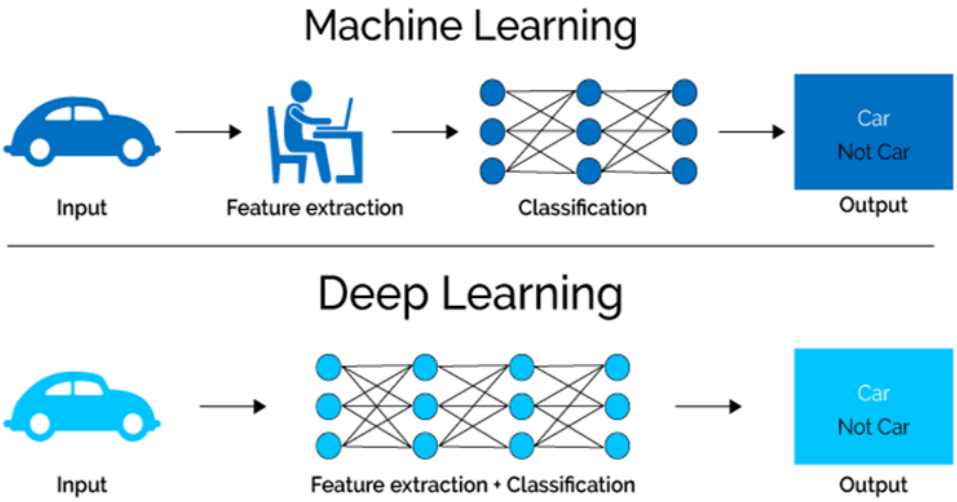


Figure 10 - Machine learning & Deep learning

For example, if humans are asked to say if a particular image is showing a car or not, they first need to identify the unique features or features of a car (shape, size, windows, wheels, etc.) extract the features and give them to the algorithm as input data.
A machine learning model imitates this behaviour so that, after an initial phase in which the network selects automatically some features, then performs the classification of the images basing on such features. That is, while in machine learning, a programmer must intervene directly in the action, in the case of a Deep Learning (DL) model, the feature extraction step is completely unnecessary. The model would recognize these unique characteristics of a car and make correct predictions. [25]

If the first huge advantage of deep learning lays in the lack of the help of a human, the second advantage that made it so popular is that it is powered by massive amounts of data. The “Big Data Era” of technology will provide huge amounts of opportunities for innovations in deep learning. Deep learning models tend to increase their accuracy with the increasing amount of training data, instead of traditional machine learning models which stop improving after a saturation point.

It is important to highlight how Deep learning has revolutionized computer vision, a field dedicated to enabling machines to interpret and understand visual information from the world. Traditional computer vision methods relied heavily on manual feature extraction, which was time-consuming and limited by human intuition. Deep learning, particularly through Convolutional Neural Networks (CNNs), has automated this process, allowing for the discovery of intricate patterns and features directly from raw images. This automated feature extraction is especially beneficial in computer vision, where the diversity and complexity of visual information make manual feature engineering impractical and often insufficient [26].

To give a brief insight about Convolutional neural networks (CNNs), CNN stand as a crucial milestone in the history of machine learning, particularly within the realm of computer vision. Discovered and developed by Yann LeCun and his colleagues in the 1980s and 1990s, CNNs made significant contributions to understanding neural networks and their application to visual processing [27]. Specialized in grid-like data, such as images and videos, CNNs are designed to recognize spatial patterns using convolutional filters and pooling layers. Their ability to capture local and hierarchical features makes CNNs highly effective in object recognition, feature detection, and image classification [26]. The rise of CNNs has revolutionized the field of computer vision, enabling human-level performance across a wide range of visual tasks. Their importance in computer vision is evident in the success of

practical applications like facial recognition, video surveillance, autonomous driving, and many others. In summary, CNNs represent a fundamental tool in the machine learning arsenal, empowering machines to comprehend and interpret the visual world with unprecedented precision and capability.

1.3.1 Deep Learning architectures applications in Computer Vision

Due to the paramount importance of deep learning in the field of computer vision, several advanced architectures and methods have been developed for tasks such as image classification, object detection, and image segmentation. This section will explore these architectures, highlighting their innovations and impact on the field.

Image classification is one of the most fundamental tasks in computer vision, where the goal is to assign a label to an input image. Deep learning models, especially CNNs, have become the standard for image classification due to their ability to learn hierarchical representations of visual data.
For instance, AlexNet's victory in the 2012 ILSVRC highlighted the power of deep learning in this domain, reducing the top-5 error rate from 26% to 15.3% [26].
Following architectures like VGGNet and ResNet have improved upon this, achieving even lower error rates and demonstrating the robustness of deep learning for image classification tasks [28].

Object detection involves identifying and localizing objects within an image. Deep learning has significantly advanced this field with models like YOLO (You Only Look Once), SSD (Single Shot MultiBox Detector), and Faster R-CNN. YOLO, introduced by Redmon et al., offers real-time object detection by framing detection as a single regression problem, simplifying the process and increasing speed [29]. Faster R-CNN, developed by Ren et al.,

introduces the Region Proposal Network (RPN) to generate high-quality region proposals, enhancing detection accuracy and efficiency [30]. These models are crucial for applications like autonomous vehicles, robotics, and security systems, where quick and accurate object detection is essential.

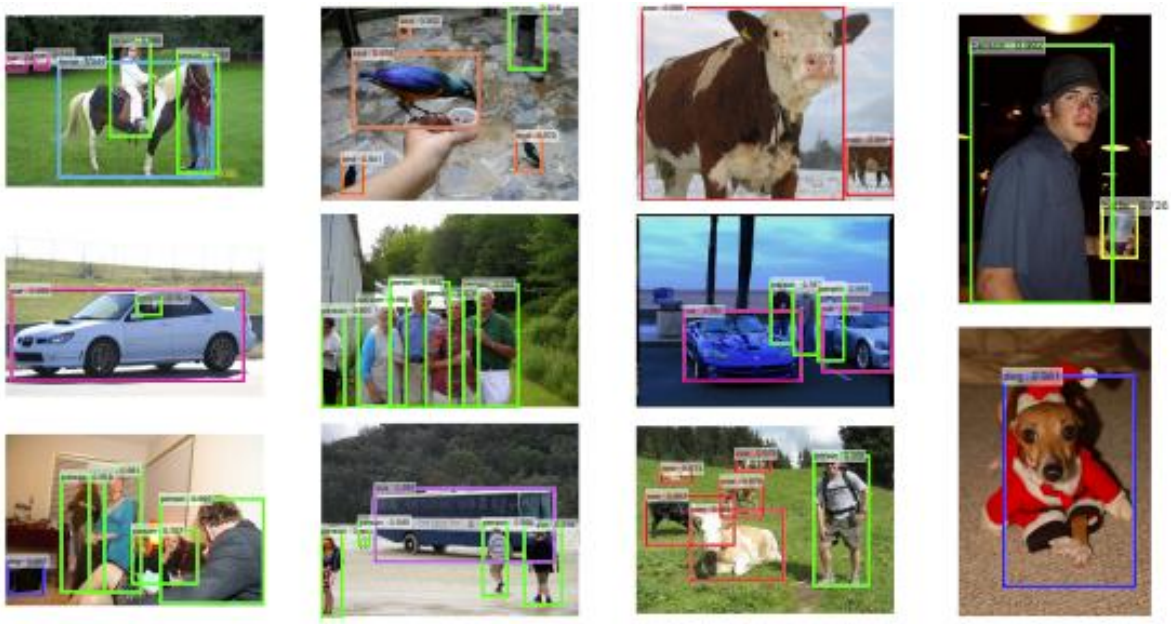


Figure 11 - Examples of object detection results using the Faster R-CNN [30]

Image segmentation involves partitioning an image into meaningful segments, often at the pixel level. This task is crucial for applications requiring precise object delineation, such as medical imaging, scene understanding, and augmented reality. U-Net, introduced by Ronneberger et al., has become a leading architecture for biomedical image segmentation due to its ability to perform well with limited training data and its innovative use of skip connections to capture both contextual and spatial information [31]. Mask R-CNN, developed by He et al., extends Faster R-CNN by adding a branch for predicting

segmentation masks alongside bounding boxes, enabling instance segmentation and enhancing the ability to separate overlapping objects in an image [32].

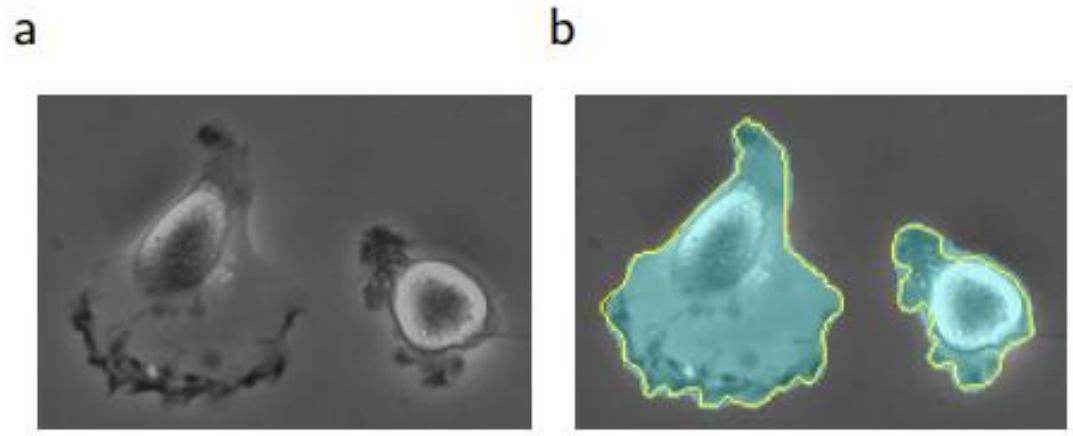


Figure 12 - Cell tracking challenge segmentation using U-Net [31]

The diversity of deep learning architectures available for computer vision tasks is a testament to the field's rapid evolution and the range of challenges it addresses. From image classification techniques like AlexNet, VGGNet, and ResNet, which have set benchmarks in recognizing objects, to advanced object detection methods such as YOLO, SSD, and Faster R-CNN, each architecture offers unique strengths tailored to different aspects of visual recognition tasks. Similarly, segmentation methods like U-Net and Mask R-CNN have revolutionized image analysis by enabling detailed object delineation and instance segmentation for applications ranging from medical imaging to scene understanding.

The choice of a particular deep learning model depends on various factors including the specific task requirements, the need for accuracy versus speed, and the nature of the data. For real-time applications where both speed and efficiency are crucial, models like YOLO stand out due to their ability to perform object detection in a single pass, achieving high performance with rapid inference times.

In the following chapter, we will delve deeper into the YOLO architecture, exploring its evolution from the original version to the latest advancements. We will discuss how YOLO's unique design features make it an ideal choice for real-time object detection in practical scenarios, and how it was selected for the specific needs of the project.

1.4 YOLO

Now, let's delve into YOLO (You Only Look Once) in detail, as it was chosen as the neural network for this project.

YOLO is renowned for its real-time object detection capabilities and speed of implementation, making it an ideal choice for applications that require fast and accurate detection. Its unique approach to framing object detection as a single regression problem allows YOLO to predict bounding boxes and class probabilities directly from full images in one evaluation, significantly enhancing processing speed and efficiency.

Being an object detection model, YOLO has an object detection model architecture which comprises three main components: the backbone, neck, and head.

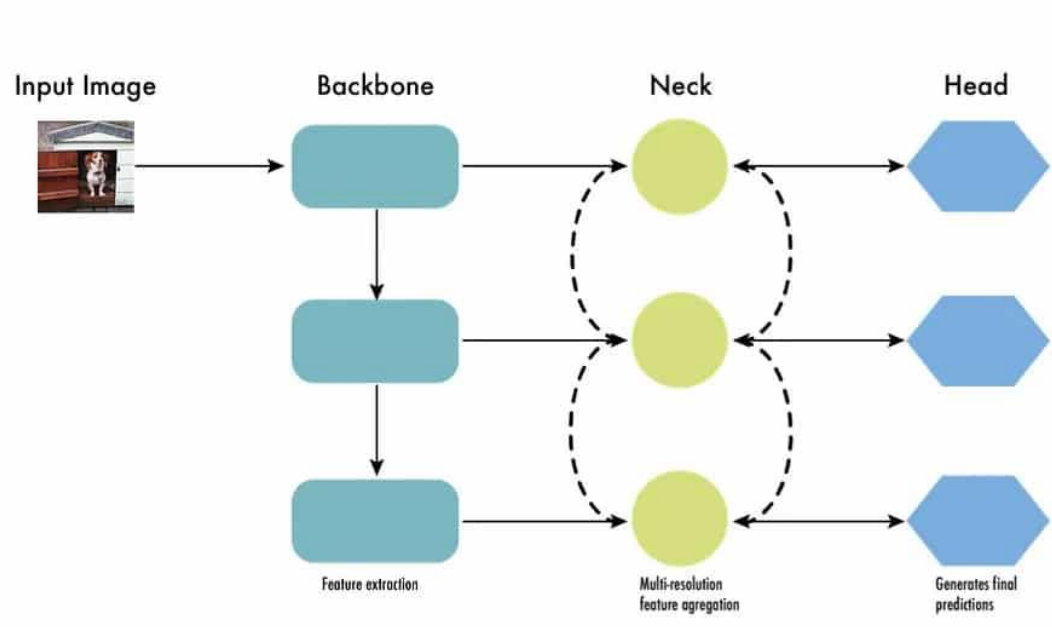


Figure 13 - Modern object detector architecture [33]

- Backbone: it is typically a pre-trained Convolutional Neural Network (CNN) that processes an input image to extract feature maps at various levels (low, medium, and high). These feature maps capture essential information about the image, such as edges, textures, and shapes.

- Neck: the neck component integrates these feature maps using techniques like the Feature Pyramid Network (FPN). Path aggregation blocks in the neck combine the information from different feature maps to ensure that both high-resolution and low-resolution details are utilized effectively.

- Head: this component is responsible for classifying objects within the image and predicting their bounding boxes. It can consist of one-stage or dense prediction models, such as YOLO or Single-shot Detector (SSD). Alternatively, it can feature two-stage or sparse prediction algorithms like the R-CNN series [34].

The YOLO (You Only Look Once) framework, introduced by Joseph Redmon et al. in their CVPR 2016 paper [35], revolutionized real-time object detection by proposing an innovative end-to-end approach. Unlike earlier methods, which relied on sliding windows with classifiers running numerous times per image, or more sophisticated two-step processes involving region proposals followed by classification, YOLO simplified the task by accomplishing detection in a single network pass by simultaneously identifying all bounding boxes in an image. This single-pass strategy enabled YOLO to predict detection outcomes directly through regression.

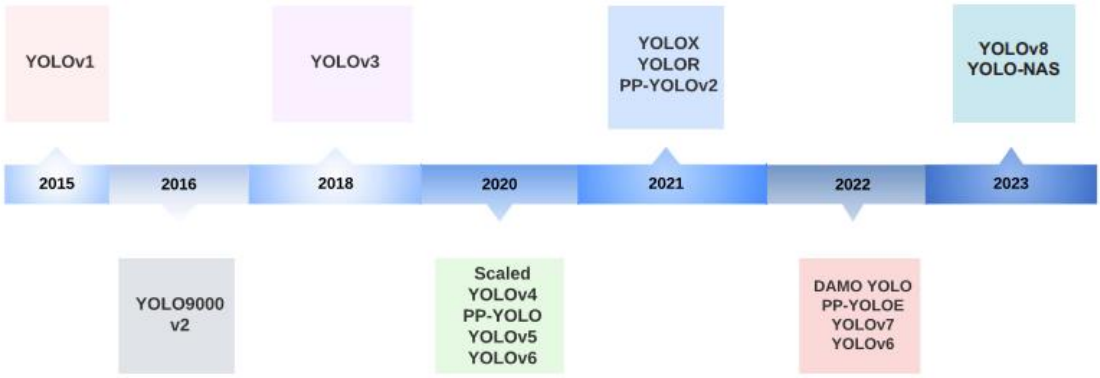


Figure 14 - YOLO versions timeline [36]

Over the years, several versions of YOLO have been developed, each introducing refinements and enhancements to improve performance and accuracy as shown in *Figure 14*. Each version of YOLO builds upon its predecessors, introducing new techniques and optimizations that address limitations and enhance capabilities.

YOLOv1 revolutionized object detection by simultaneously identifying all bounding boxes in an image. It does this by dividing the input image into an S × S grid and predicting B bounding boxes for each grid cell, alongside confidence scores for C different classes.
Each bounding box prediction includes five values **Pc**, **bx**, **by**, **bh**, and **bw**:

- **Pc (confidence score):** it indicates the model's confidence that the bounding box contains an object and the accuracy of the box itself.
- **bx and by:** they represent the coordinates of the box's center relative to the grid cell.
- **bh and bw:** they represent the height and width of the box relative to the entire image.

The output of YOLO is a tensor of dimensions S × S × (B × 5 + C) [36].

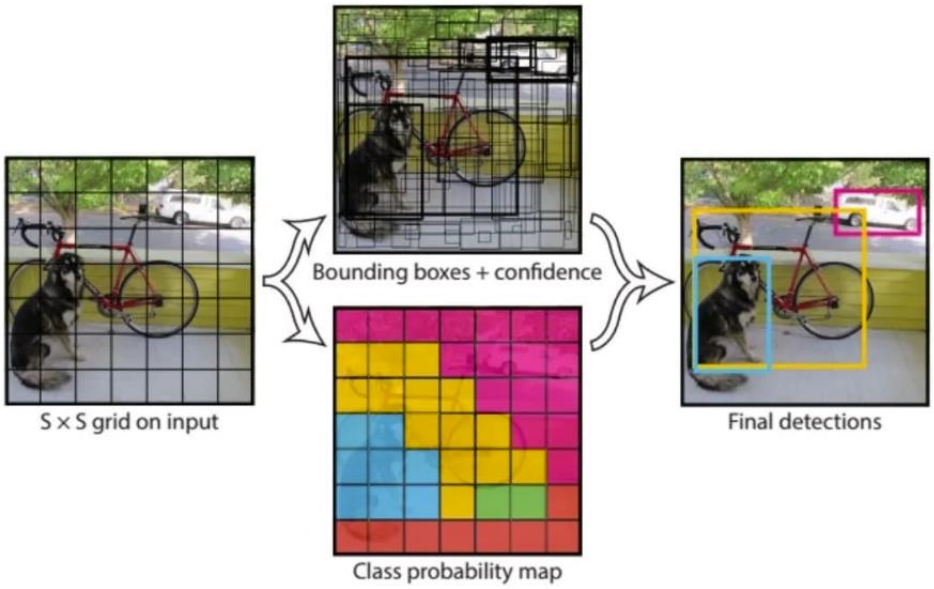


Figure 15 - YOLOv1 model [33]

This tensor can be optionally processed with non-maximum suppression (NMS) post-processing technique which allows to reduce the number of overlapping bounding boxes and improve the overall detection quality by filterin out redundant and irrelevant bounding boxes, keeping only the most accurate ones.

Figure 16- Non-Maximum Suppression (NMS) application on an image [36]

YOLOv1 achieved an average precision (AP) of 63.4 on the PASCAL VOC2007 dataset, which is a dataset that contains 20 classes (C = 20); a grid of 7 × 7 (S = 7) and at most 2 classes per grid element (B = 2), giving a 7 × 7 × 30 output prediction.

By framing object detection as a single regression problem, YOLOv1 was able to achieve significant improvements in speed and efficiency, laying the groundwork for the subsequent advancements in the YOLO series.

While YOLO is known for its fast object detection capabilities, it does have certain limitations. One significant drawback is its higher localization error compared to state-of-the-art methods like Fast R-CNN [33]. This can be attributed to several factors: YOLO's restriction to detecting a maximum of two objects of the same class within each grid cell, which limits its ability to accurately predict objects that are close to each other; its difficulty in predicting objects with aspect ratios that were not present in the training data; and the fact that YOLO learns from coarse object features due to the down-sampling process, which can reduce detection accuracy.

YOLO models have consistently evolved to balance speed and accuracy, adapting to new benchmarks and expanding their applications.

Each version introduced innovative techniques to enhance performance, from anchor boxes in YOLOv2 to anchor-free models in later versions like YOLOX and YOLOv8.

The progression from DarkNet to frameworks like PyTorch and PaddlePaddle reflects the adaptability and continuous improvement of YOLO models, making them suitable for a wide range of real-time object detection tasks.

YOLOv8, with its segmentation mask capabilities, represents the latest in this series, emphasizing efficient, high-speed detection with strong performance metrics.

I have provided a synthesis of the evolution of all the versions with their improvements and features in the table below.

Table 1 - YOLO versions characteristics synthesis

Version	**Release Date**	**Key Features**	**Improvements**	**Performance**	**Framework**
YOLOv1	2015	Single-stage detection	Simplified detection pipeline	Real-time detection	DarkNet
YOLOv2	2016	Batch normalization, high-res classifier, anchor boxes	Better convergence, reduced overfitting, high-res performance	Detects over 9000 categories	DarkNet
YOLOv3	April 2018	Multi-scale feature extraction, logistic regression	Improved accuracy and speed, anchor boxes with three sizes	60.6% mAP at 20 FPS	DarkNet
YOLOv4	April 2020	Bag-of-freebies, bag-of-specials, mosaic augmentation	Enhanced accuracy and speed balance, DropBlock regularization	Optimized for various applications	DarkNet
YOLOv5	June 2020	Developed in PyTorch	User-friendly, frequent updates	50.7% AP on MS COCO at high speed	PyTorch

Version	Release Date	Key Features	Improvements	Performance	Framework
Scaled-YOLOv4	2021	Scaling techniques, YOLOv4-tiny and YOLOv4-large	Optimized for different hardware	High accuracy with scalable performance	PyTorch
YOLOR	May 2021	Multi-task learning for classification, detection, pose	Unified network, efficient past experience usage	55.4% mAP on MS COCO at 30 FPS	PyTorch
YOLOX	July 2021	Anchor-free, MixUP and Mosaic augmentations	Simplified training, separate classification and regression	50.1% mAP on MS COCO	PyTorch
YOLOv6	September 2022	Anchor-free, industrial application models	Balance speed and accuracy for industrial use	52.5% AP on MS COCO	PyTorch
YOLOv7	July 2022	E-ELAN, model scaling, re-parametrization	State-of-the-art performance, reduced parameters and computation	55.9% AP on MS COCO at 50 FPS	PyTorch
YOLOv8	January 2023	Anchor-free, segmentation masks, mosaic augmentation	Fast Non-maximum Suppression, efficient detection	53.9% AP on MS COCO with 640-pixel images	PyTorch

1.4.1 YOLOv8

Ultralytics YOLOv8 represents the latest advancement in the YOLO series, building on the strengths of its predecessors while introducing new features to enhance performance and flexibility. As a state-of-the-art (SOTA) model, YOLOv8 is designed for exceptional speed and accuracy, making it ideal for various tasks, including object detection, tracking, instance segmentation, image classification, and pose estimation [37].

Its advanced capabilities make it suitable for a wide range of applications, from industrial use cases to academic research.

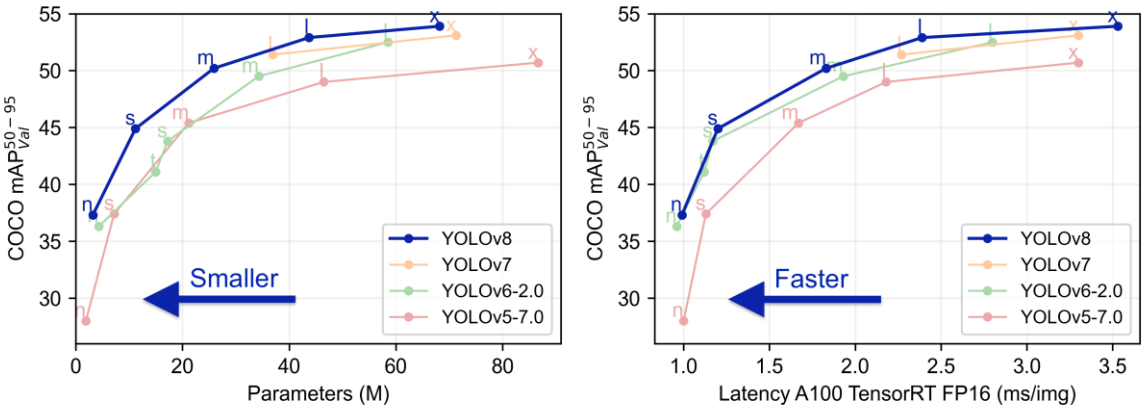


Figure 17 - Performance comparison of YOLO object detection models [37]

As it is possible to notice in *Figure 17*, the left plot illustrates the relationship between model complexity and detection accuracy. Model complexity is measured by the number of parameters, while detection accuracy is represented by COCO mAP50-95 (mean Average Precision at IoU thresholds ranging from 50% to 95%). Each model version is represented by a distinct color. The right plot highlights the tradeoff between inference speed and accuracy for the same models. Inference speed is measured as latency on an A100 TensorRT FP16, and accuracy is again represented by COCO mAP50-95. Each model version and its size variants are distinguished by different colors and markers, respectively.

The decision to use YOLOv8 for this project is driven by its superior instance segmentation capabilities, which are essential for applications requiring precise object localization and detailed analysis. Instance segmentation involves identifying and segmenting individual objects within an image, providing detailed masks or contours along with class labels and confidence scores for each object. This feature is particularly useful when precise knowledge of object shapes is necessary. By combining high accuracy with comprehensive object shape information, YOLOv8 is particularly well-suited for the project needs.

1.4.2 Challenges and limitations

It is of paramount importance to highlight the significant challenges and limitations segmentation algorithms in computer vision face, impacting their performance and reliability in real-world applications.

One of the primary challenges is *dataset bias*. Segmentation algorithms are highly dependent on the quality and diversity of the training data. If the dataset used for training is biased, containing images with limited diversity in terms of lighting, background, object types, and environmental conditions, the algorithm will likely perform poorly on real-world data that deviates from this narrow scope. This limitation makes it difficult to generalize models across different scenarios, leading to reduced accuracy and reliability.

Occlusions occur when objects in an image are partially or fully blocked by other objects, making it challenging for segmentation algorithms to accurately identify and delineate the boundaries of the occluded objects [38]. This is particularly problematic in dynamic environments where objects frequently overlap, such as crowded urban areas or dense foliage in agricultural settings. Occlusions can significantly degrade the performance of segmentation algorithms, leading to incomplete or incorrect segmentation.

Changes in *environmental conditions*, such as lighting, weather, and seasonal variations, pose another significant challenge. Segmentation algorithms trained under specific conditions may struggle to adapt to new or changing environments. For instance, shadows, reflections, and varying light intensities can alter the appearance of objects, leading to misclassifications. Similarly, weather conditions like rain, fog, and snow can obscure object features, making accurate segmentation difficult [39].

These are just some examples of the challenges that computer vision problems may face. Hence, these collectively impact the performance of segmentation algorithms, limiting their effectiveness and generalizability in diverse real-world applications. Addressing these issues requires robust dataset collection, advanced preprocessing techniques, and the development of algorithms capable of adapting to varying conditions.

By systematically tackling these challenges, researchers and practitioners can enhance the performance of computer vision systems and achieve more accurate and reliable results in practical scenarios.

Chapter 2.

State of the art - Visual Robotic systems in agricultural context

In the realm of agriculture, cutting-edge research is actively focused on advancing robotic systems integrated with sophisticated sensors designed to gather comprehensive crop data. These systems leverage state-of-the-art computer vision techniques and machine learning algorithms to address specific challenges. Among these, there are tasks such as object recognition, assessing plant health, classifying different plant species, diagnosing plant diseases, and optimizing plant life cycles. This chapter explores the current landscape of visual robotic systems in agriculture, highlighting their transformative potential in enhancing agricultural efficiency, sustainability, and precision farming practices.

2.1 Robotics

Robotics is the interdisciplinary field encompassing the design, construction, operation, and utilization of robots [40]. This field merges principles from engineering, computer science, and technology to create machines capable of autonomously or semi-autonomously performing tasks. Robots are designed to execute diverse functions, ranging from repetitive actions to intricate tasks requiring high precision and adaptability.

The origins of robotics can be traced back to ancient civilizations [40], but its modern development gained momentum during the Industrial Revolution with the advent of electrical engineering techniques enabling the operation of machines via small motors.

Figure 18 – Agricultural robot

Since the 2000s, there has been notable progress in the development of digitally programmed industrial robots equipped with artificial intelligence. In particular, the agricultural sector has witnessed a significant transformation with the advent of advanced robotics. From land preparation to harvesting, robotic systems have streamlined various farming processes, enhancing efficiency and productivity. These technological advancements have enabled farmers to manage large-scale operations with precision and reduced labor costs.

2.2 Sensor-equipped robots

In contemporary agriculture, two primary types of robots are commonly utilized: Autonomous Mobile Robots (AMRs) and Unmanned Aerial Vehicles (UAVs). These robots play crucial roles in tasks such as fruit picking, precision spraying, and soil management.

AMRs, as their name suggests, operate autonomously within agricultural environments, employing a diverse range of sensors to navigate and perform tasks effectively [41].

Unmanned Aerial Vehicles (UAVs) have significantly revolutionized precision agriculture. For instance, the DJI AGRAS MG-1P, developed by DJI, represents a notable advancement in aerial capabilities for agriculture. This octocopter is specifically designed for the precise application of liquid fertilizers, pesticides, and herbicides over large agricultural areas, utilizing omnidirectional radar systems for safe and efficient operation [42].
Moreover, UAVs equipped with uncooled thermal cameras can be used to gather detailed thermal images of crops. However, these uncooled thermal cameras can lead to lower precision due to the microbolometer not being stabilized to a constant temperature [43]. This can affect the accuracy of thermal images, but new calibration algorithms based on neural networks have been developed to improve measurement accuracy significantly.

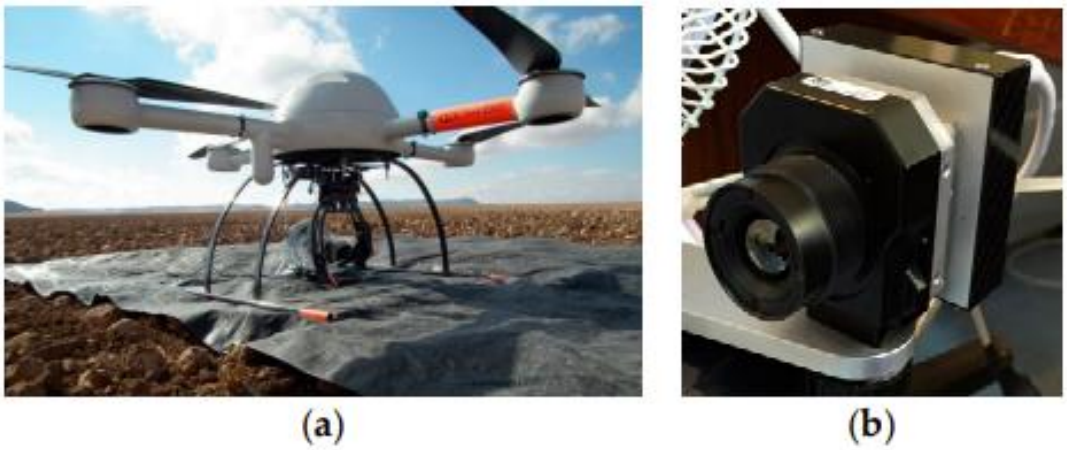


Figure 19 – (a) Unmanned aerial vehicle; (b) and uncooled thermal camera [43]

On the other hand, Autonomous Mobile Robots (AMRs) are ground-based robots designed to autonomously perform various agricultural tasks. These robots integrate a range of sensors tailored for agricultural applications, including visual cameras for navigation and object recognition, GPS for precise positioning, LIDAR for mapping environments, and ultrasonic sensors for collision avoidance.

For example, the Cäsar robot developed by Raussendorf GmbH in Germany exemplifies AMR capabilities. It utilizes Real-Time Kinematic (RTK) technology for precise navigation and features ultrasonic sensors for collision avoidance during tasks such as soil fertilization, pest control, harvesting, and transportation [42].

Figure 20 - Cäsar robot [42]

Another significant example is the Greenbot, equipped with a Four-Wheel Steering (4WS) system and collision detection sensors, enabling continuous fertilizing, plowing, and seeding operations safely in agricultural fields.

2.2.1 Visual plant inspection

Computer vision systems are integral to modern agricultural robots, enabling tasks such as plant detection, plant health assessment, and crop monitoring. For example, the eAGROBOT utilizes RGB cameras and artificial intelligence algorithms (K-means and neural networks) to identify pests in cotton and groundnut crops. It achieves high precision in disease identification, with accuracy ranging from 83% to 96% [42].

Additionally, the AgBot robot is still in the research stage but shows promise in agricultural applications [42]. Designed for use on corn farms, the AgBot applies fertilizers and

herbicides using a Two-Wheel Drive (2WD) system and features four distinct reservoirs for different types of herbicides and fertilizers. It employs a low-cost RGB camera and the Haar feature-based cascade classifiers machine learning algorithm to detect specific weed species such as Giant ragweed, Redroot pigweed, and Cocklebur. However, the low-cost camera used proved unsuitable for external use, indicating the need for further research and improvements.

Figure 21 – (a) VINBOT; (b) VineRobot [42]

Further advancements in agricultural robotics include projects funded by the European Union's Seventh Framework Program, such as VINBOT and VineRobot. VINBOT uses Convolutional Neural Networks (CNN) to detect grapes, compute the area of grape occupation in images, and estimate their respective weight in kilograms [42]. VineRobot monitors parameters such as grape yield, vegetative growth, vineyard water status, and grape composition, using various advanced techniques [42].

Thermal imaging has been used for years in agriculture and has also found significant applications in mobile robotics. For instance, in forest management, thermal imaging combined with deep learning has been employed to detect tree trunks for tasks such as inventory and autonomous navigation. A recent study used deep learning models on a dataset of visible and thermal images to improve the accuracy of trunk detection, highlighting the potential of thermal imaging to enhance robotic perception systems in forestry [44].

Figure 22 - Robotics platforms used to acquire forest images [44]

Both RIPPA and Ladybird robots use hyperspectral and thermal cameras along with RTK/GPS/INS systems to manage weeds and enhance crop health through targeted herbicide spraying. The spectral data helps in assessing plant health and administering appropriate treatments [42].

Figure 23 - (a) RIPPA, (b) Ladybird [42]

2.2.2 Water stress assessment

Among the various tasks that robots can perform in agriculture, water stress assessment is crucial for evaluating plant health and conserving water resources. The key value used to assess the plant's condition is the Crop Water Stress Index (CWSI).

This vaue is based on the concept that the temperature of plant leaves increases when they are under water stress. CWSI values range from 0 (no stress) to 1 (maximum stress), providing a clear indication of the water status of crops. This index is especially useful for irrigation management, allowing farmers to optimize water usage and improve crop yield and quality.

Related to this aim, the study conducted in commercial vineyards in Douro Superior, Portugal, during the 2019 and 2020 seasons focuses on using the VineScout, a ground robot developed under the H2020 EU project, to assess and map vineyard water status using thermal infrared radiometry. The robot recorded canopy temperature (Tc) values with an infrared radiometer, alongside environmental data (air temperature, relative humidity, and atmospheric pressure) and NDVI measurements with a multispectral sensor. These data were used to develop spatial-temporal variation maps, helping to reduce water consumption and implement efficient irrigation strategies. The promising results indicate the need for further studies to enhance the accuracy and robustness of predictive models for sustainable viticulture [45].

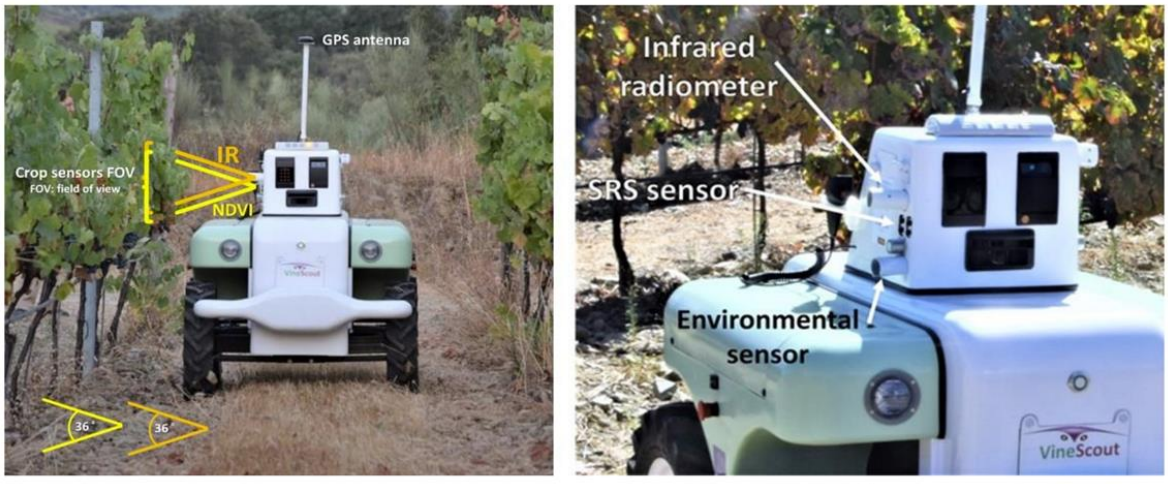


Figure 24 – (a) Vinescout VS-3 autonomous ground vehicle used to monitor grapevine water status; (b) Detail of the crop sensing unit used for on-the-go measurement of water status [45]

Another way to compute the water stress level of plants is through the use of of multispectral imagery. Hence, the research conducted at CphFarmHouse in Denmark aimed to develop a widely applicable methodology for the early detection of water and nitrogen stress through the use of low-altitude multispectral imagery [46]. Multispectral reflectance systems were utilized to measure crop reflectance, which increased significantly due to water and nitrogen deficiencies. The study focused on the Normalized Difference Vegetation Index (NDVI) and the Photochemical Reflectance Index (PRI), which showed significant differences between control and stress treatments. These findings suggest that multispectral images can be another effective tool for rapidly estimating the physiological status of plants, highlighting spatial variation in vertical farms.

Table 2 - Robotic applications

Task	**Robot**	**Final Application**	**Sensors**	**Sensors Used to Perform the Task**	**Computer Vision Algorithm**
Land Preparation	Cäsar	Soil fertilization	RTK, ultrasonic	RTK for navigation, ultrasonic for collision avoidance	None
Sowing	Lumai-5	Wheat sowing	Speed, angle, pressure	Speed, angle, and pressure sensors for sowing accuracy	None
Disease Identification	eAGROBOT	Pest identification in crops	RGB camera	RGB camera for image acquisition	K-means, Neural Networks
Weed Control	RIPPA, Ladybird	Weed management	Hyperspectral, thermal, RTK/GPS/INS	Hyperspectral and thermal cameras for weed detection	None

Task	Robot	Final Application	Sensors	Sensors Used to Perform the Task	Computer Vision Algorithm
Harvesting	Vegebot	Lettuce harvesting	RGB cameras	RGB cameras for lettuce identification	R-CNN
Harvesting	Noronn AS	Strawberry harvesting	RGB-D camera	RGB-D camera for strawberry detection	R-CNN
Yield Estimation	Shrimp	Apple yield estimation	RGB cameras, INS, GPS	RGB cameras for image capture, INS and GPS for location	MLP, CNN
Vineyard Monitoring	VINBOT, VineRobot	Grape yield and health monitoring	RGB, NIR, thermography	RGB and NIR cameras for monitoring, thermography for health assessment	CNN, Chlorophyll-based fluorescence, machine vision
Water Stress Assessment	VineScout	Vineyard water status mapping	Thermal infrared radiometer, environmental sensors, multispectral	Thermal infrared radiometer for Tc, environmental sensors (Tair, RH, AP), NDVI for mapping	Partial Least Squares (PLS) regression

The analysis of various agricultural robots in the table highlights the extensive use of RGB and RGB-D cameras for object detection, segmentation, and classification tasks. These cameras have been consistently employed across different robotic systems, such as Vegebot for lettuce harvesting and the Noronn AS robot for strawberry harvesting. The similarity in their application underscores the reliability and effectiveness of RGB and RGB-D cameras in visual recognition tasks within agricultural environments. The computer vision algorithms typically used for these purposes include well-established techniques like Region-based

Convolutional Neural Networks (R-CNN), K-means clustering, and artificial neural networks, which have proven successful in accurately identifying and classifying objects.

Furthermore, the integration of thermal imaging in agricultural robots has opened new avenues for health assessment and weed detection. For instance, the RIPPA and Ladybird robots leverage hyperspectral and thermal cameras to assess plant health and manage weeds through targeted herbicide spraying. This use of thermal imaging extends beyond simple detection, providing valuable data for making informed decisions about crop health and management. By analyzing thermal signatures, these robots can detect stress in plants, identify specific weed species, and ensure precise application of treatments, thereby enhancing overall agricultural productivity and sustainability.

All things considered, while RGB and RGB-D cameras remain fundamental in agricultural robotics for object detection and classification, the incorporation of thermal imaging introduces advanced capabilities for health assessment and targeted interventions. These combined technologies represent significant strides in the development of intelligent agricultural systems, paving the way for more efficient and sustainable farming practices.

Chapter 3.

System Definition

In this chapter, we will explore the primary components of the system designed for our study, focusing on both the data acquisition sensors and the design of the overall hardware setup. The objective of this thesis is to develop an easily installable sensor system for agricultural robots capable of measuring plant water stress in real time. To address this objective, we conducted an analysis of the available technologies and decided to use these specific sensors. This chapter is divided into two main sections: the first section addresses the technical specifications and functionalities of the two types of cameras used in the system, while the second section delves into the design and configuration of the entire hardware setup.

3.1 Cameras

3.1.1 Intel Realsense camera D455/D457

RGB-D cameras, also known as depth cameras, are highly beneficial in agriculture when mounted on agricultural robots or tractors. These cameras provide detailed crop information, assisting farmers in making better-informed decisions and optimizing their farming practices. Unlike standard RGB cameras that only capture color information, RGB-D cameras also capture depth information through infrared sensors. These sensors measure the distance between the camera and each pixel, generating a depth map. This depth map can then be used to create a 3D model from the captured two-dimensional image, offering valuable insights for various agricultural applications [47].

These cameras are also very useful in robotics and autonomous devices like drones; so these features make them suitable for use onboard an agricultural vehicle.
In 2015, Intel introduced advanced RGB-D sensors with improved subpixel disparity precision, enhanced lighting capabilities, and outdoor functionality. Building on this technology, Intel developed the RealSense D400 series, which uses stereo vision for depth measurement [48]. This system includes a left imager, a right imager, and an optional infrared projector. The projector emits an invisible infrared pattern, and the depth is calculated by correlating features between the two images to determine pixel-wise depth values.

Among this family of sensors, an example of a stereo camera can be the Intel Realsense D455 or D457.

Figure 25 - Intel Realsense D457

In particular, the Intel RealSense is a USB-powered camera that includes depth sensors and an RGB sensor. This camera has attracted increasing interest since it is cost-effective and can work under different ambient light conditions.

3.1.2 Optris Xi 400 thermal camera

Thermal remote sensing is a sophisticated imaging technology that converts the infrared radiation emitted by objects into visible images, known as thermograms or thermal images.

This method, which can be implemented through portable hand-held devices or advanced systems mounted on planes and satellites, is non-invasive, non-contact, and non-destructive. It is used to analyze the thermal properties of objects and environments, making it a valuable tool for various fields where heat changes are significant [49].

As explained in the state of art, the technology is employed for numerous agricultural tasks including monitoring plant health in nurseries and greenhouses, scheduling irrigation, detecting diseases, estimating fruit yields, evaluating the maturity of fruits, and identifying bruises on produce.

Among the thermal cameras, the Optris Xi 400 is a specialized thermal imaging camera designed for condition monitoring and early fire detection.

In the project, the Optris Xi 400 camera was selected for its ability to measure the temperature of crops, which is crucial for assessing their water stress levels. This measurement data is used to apply the Crop Water Stress Index (CWSI) formula using Python, enabling real-time determination of plant hydration needs.

3.1.3 Cameras characteristics and comparison

RGB-D cameras, like the Intel RealSense D455 and D457, capture both color (RGB) and depth information. These cameras provide a comprehensive view of the scene by combining

high-resolution visual images with depth data, which can be used for tasks such as 3D mapping and object detection [50].

For this project, the Intel RealSense D455 and D457 cameras were used to collect RGB and depth data. The project initially started with the D455, but due to a hardware issue, it was replaced with the D457. While the depth data from these cameras was collected, it was not directly utilized for the final analysis of crop water stress. Instead, the primary focus was on the RGB images to visually assess plant conditions.

The table below shows the characteristics of each camera allowing an easy comparison and a synthesis of their characteristics [50], [51].

Table 3 - Comparative Table: Optris Xi 400, Intel RealSense D455/D457

Feature	Optris Xi 400	Intel RealSense D455	Intel RealSense D457
Camera Type	Thermal Infrared Camera	Depth Camera with RGB and IR Sensors	Depth Camera with RGB and IR Sensors
Resolution (Thermal)	382 x 288 pixels	1280 x 720 pixels (Depth)	1280 x 720 pixels (Depth)
Resolution (RGB)	1280 x 720 pixels (VIS)	1920 x 1080 pixels (RGB)	1920 x 1080 pixels (RGB)
Frame Rate (Thermal)	80 Hz	30 fps (Depth & RGB)	30 fps (Depth & RGB)
Frame Rate (RGB)	30 Hz	30 fps (RGB)	30 fps (RGB)
Temperature Range	-40°C to 900°C	Not Applicable	Not Applicable
Depth Range	Not Applicable	0.4 m to 10 m	0.4 m to 10 m
Depth Accuracy	Not Applicable	±2% of measured distance	±1% of measured distance

Feature	Optris Xi 400	Intel RealSense D455	Intel RealSense D457
IR Camera Resolution	640 x 480 pixels (for VIS)	640 x 480 pixels	640 x 480 pixels
RGB Camera Resolution	1280 x 720 pixels	1920 x 1080 pixels	1920 x 1080 pixels
Field of View (Depth)	80° x 54° (Wide), 53° x 38° (Narrow)	87° x 58°	87° x 58°
Field of View (RGB)	65°	69.4° x 42.5°	69.4° x 42.5°
Pixel Size (80° x 54° Lens)	3.4 mm at 0.8 m distance	N/A	N/A
Pixel Size (53° x 38° Lens)	2.1 mm at 0.8 m distance	N/A	N/A
Measurement Width (80° x 54° Lens)	~1.3 m at 0.8 m distance	N/A	N/A
Measurement Width (53° x 38° Lens)	~0.8 m at 0.8 m distance	N/A	N/A
Focus Mechanism	Manual Motorized Focus (Software Control)	Fixed	Fixed
IP Rating	IP66	N/A	IP65
Environmental Operating Range	-40°C to 50°C	Not Specified	Not Specified
Connectivity	USB 3.0, PoE	USB 3.1 Type-C	USB 3.1 Type-C
Dimensions (L x W x H)	135 mm x 80 mm x 80 mm	130 mm x 50 mm x 30 mm	130 mm x 50 mm x 30 mm
Weight	~1.2 kg	145 g	145 g

For the project's purpose, it is important to highlight the difference between thermal and RGB-D camera in terms of resolution. Hence, the D455 and D457 cameras provide high-resolution RGB images at 1920 x 1080 pixels, which are crucial for detailed visual inspections of crops. In contrast, the Optris Xi 400 thermal camera offers a thermal resolution of 382 x 288 pixels, vital for detecting small temperature differences in the crop canopy and essential for accurate water stress analysis. This difference in resolution is significant for the implementation and data fusion process, presenting a challenge due to the varying resolutions.

The D455 and D457 cameras, with a range from 0.4 to 10 meters, allow for precise spatial measurements and 3D modeling of the crop canopy. The Xi 400, on the other hand, excels at close-range measurements, functioning optimally at distances from 0.3 to 0.8 meters.

Lens options play an important role in the functionality of these cameras. The Xi 400 offers different lenses, such as an 80° x 54° lens for broader views and a 53° x 38° lens for higher resolution. For the project, the narrower 53° x 38° lens was chosen to maintain high resolution and minimize distortion while measuring temperatures at close distances. Meanwhile, the RGB cameras perform well at longer distances, up to 10 meters.

In terms of environmental protection, the D457 camera has an IP65 rating, making it resistant to dust and capable of withstanding low-pressure water jets, suitable for both outdoor and greenhouse conditions. Although the D455's IP rating isn't explicitly stated, it is designed for similar outdoor applications. The Xi 400 surpasses both with an IP66 rating, ensuring it is dust-tight and can endure powerful water jets, providing superior protection in harsh environments.

To sum up, the Intel RealSense D455/D457 cameras were utilized for their high-resolution RGB imaging and depth data capabilities, though the depth data was not used for the final analysis. Their main role was to provide visual assessments of crop conditions. In contrast, the Optris Xi 400 was crucial for its high-resolution thermal imaging and broad temperature range, which were directly used for calculating CWSI and assessing crop water stress.

By integrating the visual data from the RealSense cameras with the thermal data from the Optris Xi 400, the project achieved a comprehensive analysis of crop conditions, which was fundamental for effective irrigation management.

3.2 Hardware setup

In the realm of computer vision and multimodal sensor fusion, the design and configuration of camera setups are crucial for achieving optimal data acquisition and system performance. Various strategies have been explored in the literature to align different sensors for specific applications.

For instance, Spremolla et al. (2020) describe a hardware setup where an RGB-D sensor and a thermal camera are mounted side-by-side using a rigid support structure. This configuration allows for simultaneous capture of RGB, depth, and thermal images, which is essential for applications like person tracking where data from multiple modalities must be synchronized and fused effectively [52].

Similarly, Vidas et al. (2017) present a prototype that features a hand-held RGB-D camera mounted in conjunction with a thermal infrared camera. Their setup emphasizes the importance of geometric and temporal calibration between the sensors to ensure accurate 3D thermal mapping of building interiors. Their work illustrates that the alignment of the sensors

and the management of timing irregularities are key considerations in multimodal sensor configurations [53].

These examples reflect common practices in the field, where sensors are often mounted side-by-side or in a fixed arrangement to facilitate the integration of various types of data, such as RGB, depth, and thermal information.

3.2.1 Design of the setup

In our project, the team designed a unique camera configuration to meet the specific needs of our agricultural robotics application. After an in-depth review of existing methods and a thorough assessment of our project's requirements, we opted for a vertical stacking arrangement for the cameras.

Our specific setup includes:

- Camera 1: Optris xi 400 Thermal Camera placed at the top.
- Camera 2: Intel RealSense D455/D457 positioned directly below Camera 1.

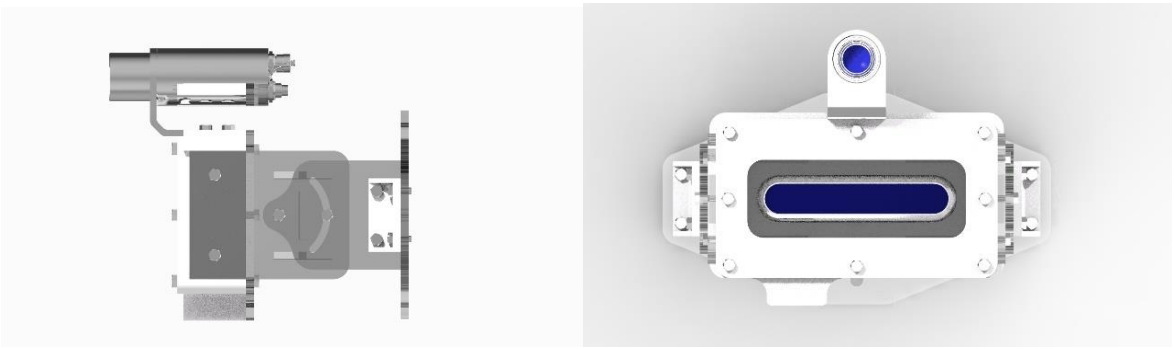


Figure 26 - Setup project design

By aligning the cameras in this manner, we aim to achieve several key advantages:

- *Improved spatial coherence*: The vertical arrangement allows the cameras to share a common optical axis, which helps maintain spatial coherence between the thermal

and RGB-D data. This is particularly important because the Optris xi 400 thermal camera has a smaller resolution compared to the Intel RealSense D455/D457. If the cameras were positioned horizontally, the disparity in resolution could lead to challenges in real-time data synchronization and integration. By stacking the cameras vertically, we avoid potential misalignments and ensure that the thermal and RGB-D images correspond more accurately in the same scene.

- *Effective data integration*: The vertical setup facilitates better alignment of the fields of view for simultaneous image capture. This alignment is crucial for the effective fusion of RGB-D and thermal data, allowing us to combine high-resolution RGB-D images with the thermal data captured in a complementary manner.
- *Compact and functional design*: This configuration supports a more compact and streamlined design for the robot's turret. By stacking the cameras, we reduce the overall footprint of the camera system, which is beneficial for mounting on the mobile agricultural robot. The design also incorporates an adjustable mechanism that allows the setup to be tilted, providing flexibility for different viewing angles and operational scenarios (*Figure 27*).

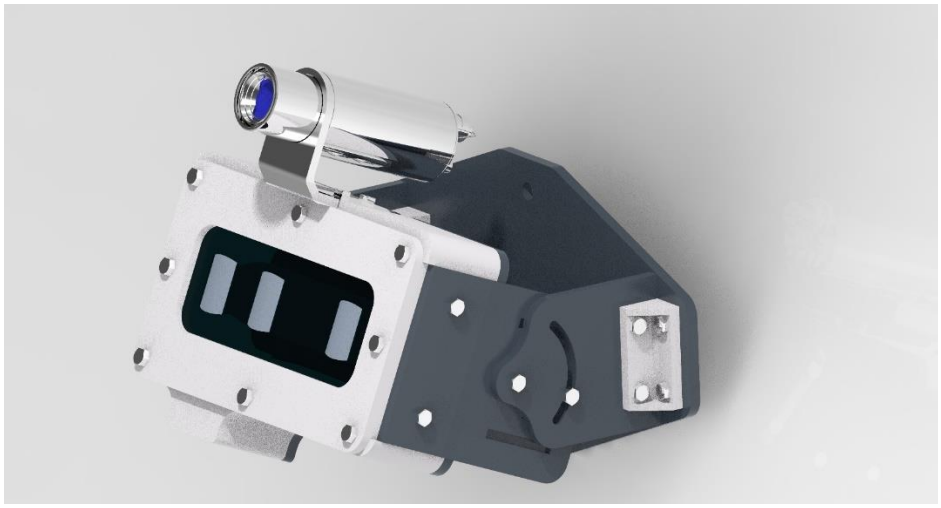


Figure 27 - Cameras rotation mechanism

This vertical stacking configuration thus effectively supports our application's goal of capturing high-resolution images for the monitoring and analysis of plants in both outdoor fields and indoor greenhouses.

3.2.2 Realization of the setup

To bring our camera configuration design to life, we utilized 3D printing technology to create an initial prototype. This prototype served as a crucial first step in our development process, allowing us to conduct experiments and evaluate the effectiveness of our design before final production.

The 3D printed prototype was crafted to test the feasibility of the design and to make necessary adjustments. This iterative process involved creating a physical model of the camera mount, which was then used for a series of tests to assess its performance in real-world conditions.

Figure 28 - 3D printed prototype of the cameras configuration

The prototype allowed us to:

- *Verify design concepts*: The initial prototype enabled us to validate the design concepts and ensure that the theoretical models translated effectively into a physical structure.
- *Test functionality*: We used the prototype to test the mechanical features of the setup, including the tilt adjustment mechanism and camera alignment, to confirm that the design met our operational requirements.
- *Identified future improvements*: Through testing, we gained insights into potential areas for future improvements. Although we did not yet make these changes, the feedback we collected will guide future refinements to optimize the camera mount for our application.

Chapter 4.

Methodology

This section outlines the systematic approach used to gather, annotate, and integrate datasets crucial for training an object recognition model focused on agricultural applications. It begins with dataset acquisition and annotation using Roboflow.

Training utilized transfer learning with YOLOv8 on Google Colab's GPU, optimizing model performance across various plant types and environments.

A critical aspect of this study involves sensor and data fusion. Calibration of RGB and thermal cameras within the ROS framework ensured precise data fusion for real-time agricultural monitoring and analysis.

By detailing these methodologies, this section underscores their significance in ensuring the accuracy and validity of the study's findings.

4.1 Dataset acquisition, annotation, creation

The first step in the project involved acquiring and preparing datasets for training the object recognition model. Various types of plant data and imagery were worked with, including close-up shots of individual leaves, images of entire plants, and specific plant varieties such as vineyards, broad beans, and lettuce.

The creation of the datasets, annotation of individual images, and the split into training, validation, and test sets was done using Roboflow. Roboflow is a tool that provides extensive support for image annotation. It streamlines the process by offering an intuitive interface for labeling and organizing images. However, it is necessary to manually refine the mask contours of each image to ensure accuracy and precision. This manual refinement is crucial

for achieving high-quality annotations, which in turn enhances the performance of the object recognition model.

Initially, publicly available online datasets of plant images were searched for. Several datasets of leaves were found and combined to create a comprehensive dataset of individual leaves. A similar process was applied to vineyard images, creating a specific dataset for vine leaves. After annotation, the datasets were uploaded and used for training the neural network.

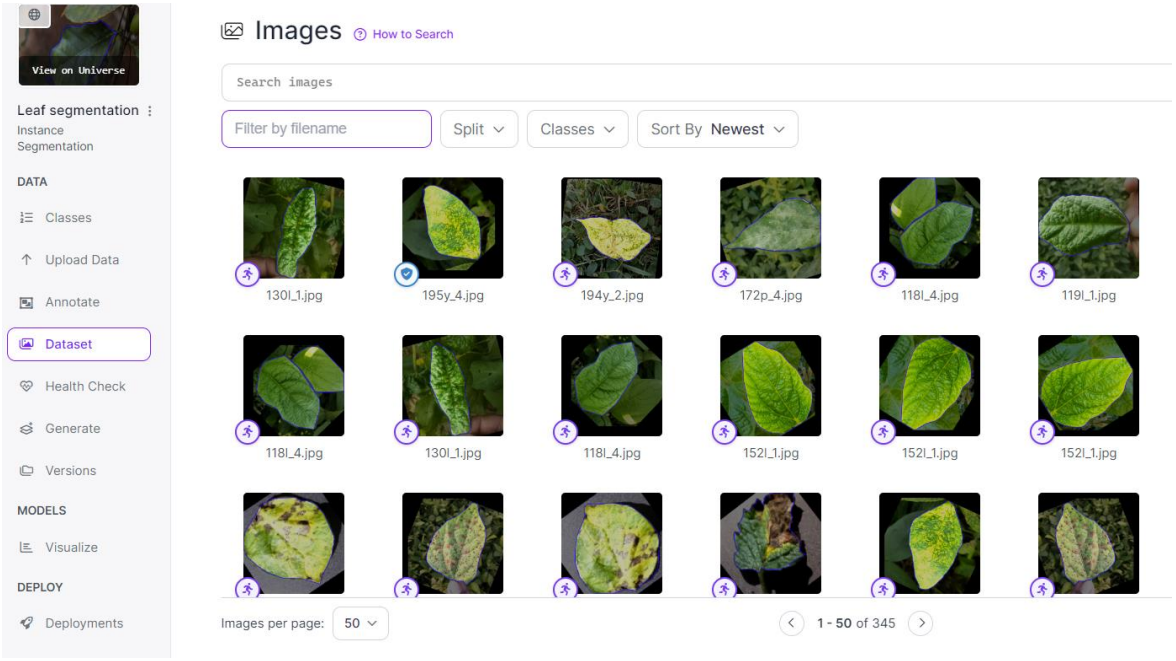


Figure 29 - Leaf Dataset on Roboflow

Throughout the months, several field experiments were conducted to verify and test the progress of the project, ensuring it was heading in the right direction.

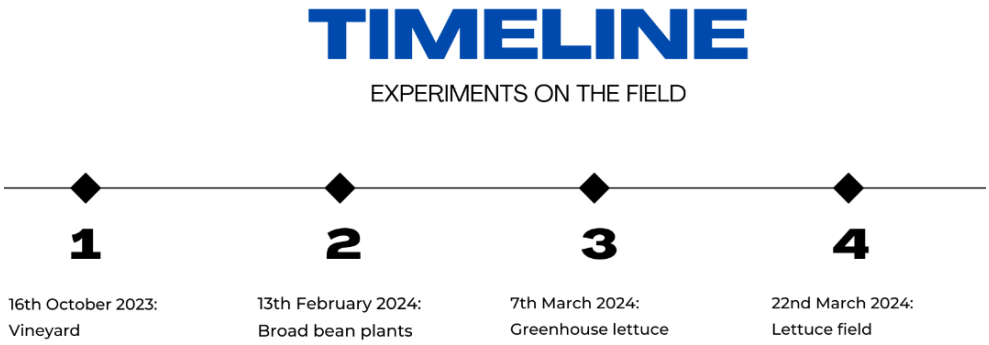


Figure 30 – Experiments timeline

On field acquisition 1, fieldwork was conducted to collect real-world images using an RGB camera. Despite it being the off-season for many plants, images of vine leaves in a vineyard were successfully captured. During this session, it was discovered that the D455 camera had a hardware issue causing a purple hue due to sunlight interference. Nevertheless, the images were used to test the pre-trained network on the vine dataset, which produced promising results.

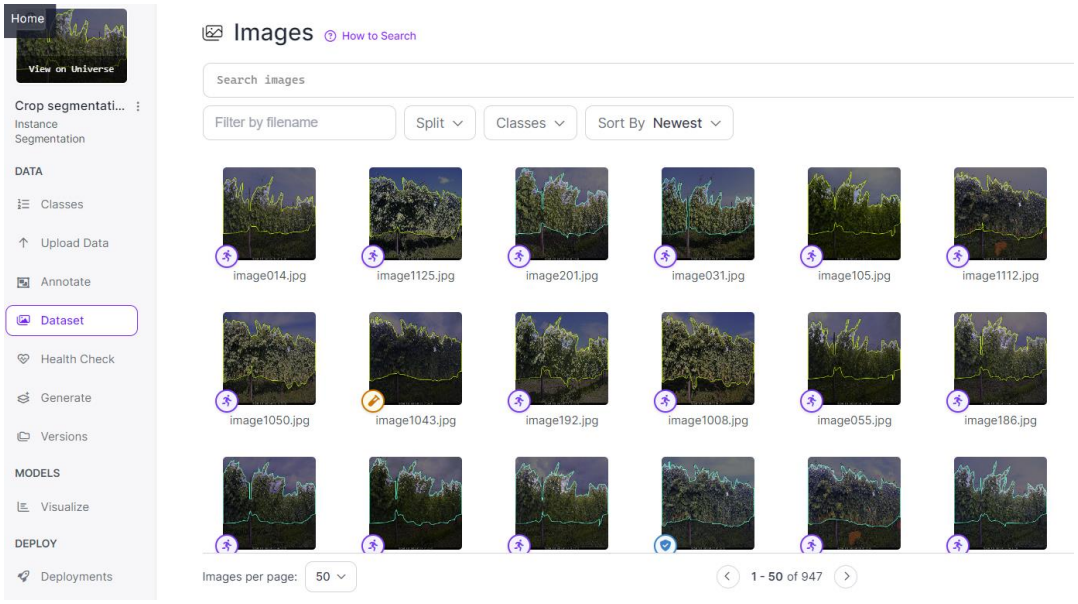


Figure 31 - Grapevine dataset on Roboflow

Figure 32 - Vine segmentation test through YOLOv8 (HW error of camera D455)

The process was repeated for various types of plants, including individual leaves. Experiments for single leaves were conducted both in laboratory settings and in the fields to ensure a diverse and robust dataset.

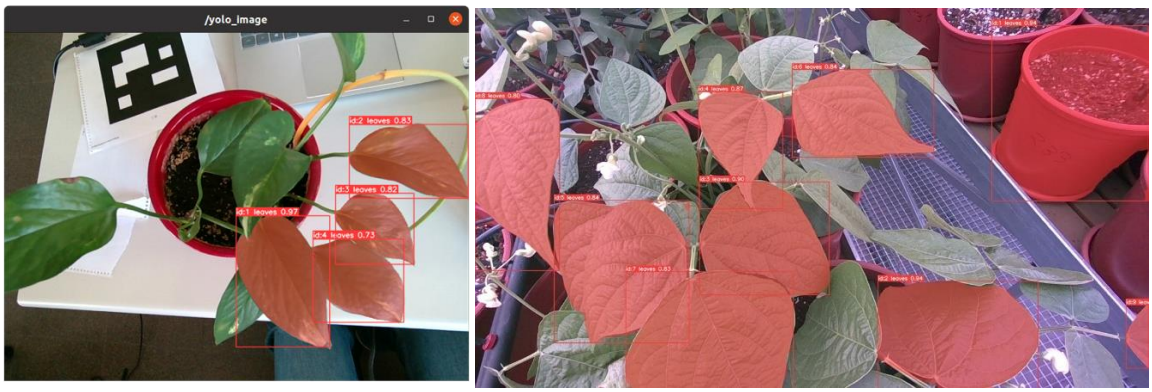


Figure 33 - Leaves segmentation test with YOLOv8: (a) laboratory, (b) field

During field acquisition 2, another field was visited to collect images of broad bean plants, which grow in bush-like formations. Using a cart-mounted camera, data was captured and photos and videos were saved. A new dataset was then created from these images, which were carefully annotated and used to train the neural network.

Figure 34 - Setup images acquisitions on field

In response to the unresolved hardware issues with the D455 camera, a decision was made to procure a new D457 camera to prevent future development problems in the project.

During field acquisition 3, images of lettuce plants in a greenhouse were collected at another university near Barcelona. Additional online images of lettuce had been found by this time, enabling the testing of a dual-camera system with a network trained on this expanded dataset. This test was successful and validated our approach.

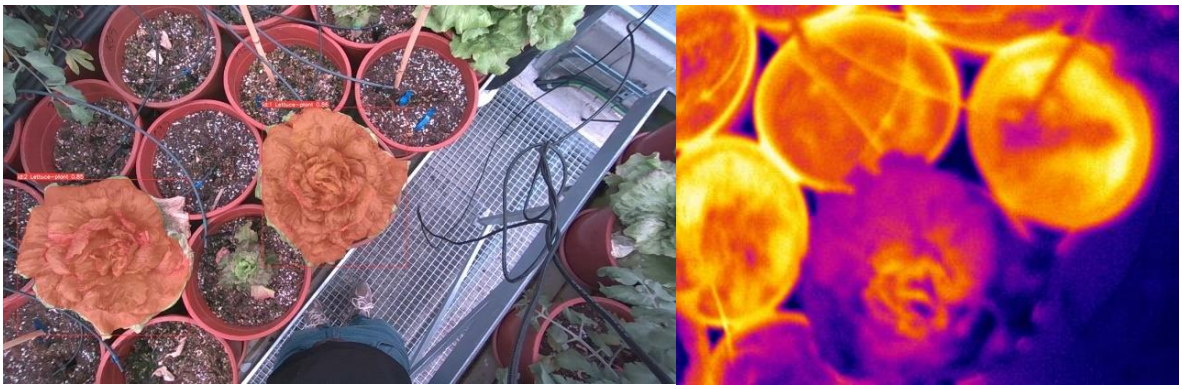


Figure 35 – Greenhouse test:(a) Lettuce realtime segmentation with YOLOv8; (b) corresponding thermal image realtime

Finally, during field acquisition 4, a final field test of the dual-camera system on lettuce plants was conducted. Various types of lettuce were discovered and new images were captured to further enrich and improve the dataset, with the aim of enhancing the system's performance.

Figure 36 - Lettuce segmentation with YOLOv8 (field experiment 4)

Throughout this process, the datasets were meticulously divided into training, validation, and testing sets to ensure a balanced and effective training regimen for the neural network. This iterative process of acquiring, annotating, and training on diverse plant datasets was crucial in building a robust and comprehensive dataset, thereby improving the accuracy and reliability of our object recognition model.

4.2 Training YOLOv8

Once the datasets were prepared, YOLOv8 neural network was trained using Google Colab's GPU resources, which are available for free. Leveraging the computational power of Google Colab allowed for efficient and effective training of the neural network. The training process primarily relied on transfer learning, which involves using pre-trained weights and fine-tuning them on our specific datasets.

The training process involved several steps:

1. Uploading the dataset: First, the annotated datasets were uploaded from Roboflow to Google Colab.

2. Transfer Learning: the pre-trained weights available online were used to initialize the YOLOv8 model. This method leverages the pre-existing knowledge encoded in these weights, providing a strong starting point for our training process.

3. Custom training: the training script used was:

```
%cd {HOME}

!yolo task=segment mode=train model=yolov8m-seg.pt data={dataset.location}/data.yaml epochs=150 imgsz=640
```

Figure 37 - Code script explanation

This script performs custom training on the YOLOv8 model for the segmentation task where:

- *task=segment*: specifies that the task is segmentation.
- *mode=train*: indicates that the model is in training mode.

- *model=yolov8m-seg.pt*: it uses pre-trained weights for the YOLOv8 medium-sized segmentation model.
- *data={dataset.location}/data.yaml*: it points to the dataset configuration file, which includes paths to the training and validation data.
- *epochs=150*: specifies the number of training epochs.
- *imgsz=640*: sets the input image size for the model.

4. Fine-tuning the model: During training, the number of epochs were primarily adjusted to optimize performance. Given the transfer learning approach, this iterative process involved monitoring the loss functions and evaluation metrics to ensure the model was learning effectively without extensive parameter adjustments.

5. Evaluating the performance: The evaluation was done using:
    - *mode=val*: to validate the model using the validation dataset, ensuring it generalizes well to unseen data.
    - *mode=predict*: to test the model's performance on new images, verifying its practical application in real-world scenarios.

Using transfer learning significantly accelerated the training process, as the model started with a strong foundation provided by the pre-trained weights. The use of Google Colab's GPU resources was instrumental in speeding up the training process, allowing for more iterations and better model optimization. Each iteration of training helped improve the model's accuracy, making it well-suited for our object recognition and segmentation tasks in agricultural environments.

4.3 Camera calibration

Calibration is a fundamental step in any vision-based system, especially when working with multiple sensors like RGB and thermal cameras. Calibration ensures that the data captured by different sensors can be accurately aligned and interpreted in a unified manner. This process is critical for applications requiring precise image fusion, such as our agricultural monitoring system where we combine RGB and thermal data to calculate metrics like the Crop Water Stress Index (CWSI).

Calibrating the cameras was a crucial and challenging part of the project. The goal was to calibrate both the RGB and thermal cameras to ensure accurate and synchronized data capture. This process was particularly difficult because finding a checkerboard pattern that is visible to both RGB and thermal cameras is not straightforward. Environmental factors like lighting and heat sources further complicated the visibility and accuracy of the checkerboard pattern. Despite these challenges, successful calibration is essential for achieving high-quality, reliable data from multi-sensor systems.

To streamline data acquisition and processing, the cameras were operated through the Robot Operating System (ROS). This integration helped manage the complexity of the calibration process and ensured that the captured data could be effectively used for precise image fusion and analysis.

4.3.1 Calibration process

To address these challenges, a series of steps were taken involving custom calibration scripts and specific environmental conditions:

- *Custom calibration scripts in ROS:*

  ROS (Robot Operating System) scripts were developed to handle the calibration process, which streamlined data acquisition and processing from both cameras.

  The ROS framework facilitated the synchronization and real-time data handling necessary for effective calibration.

- *Checkerboard design and setup:*

  A thick cardboard with an A4-sized black and white checkerboard pattern was chosen to be used after some tests with other kind of materials.

  The natural sunlight served as a powerful and immediate heat source, making the checkerboard visible to both RGB and thermal cameras.

Figure 38 - Calibration process: images acquisition in the sun

- *Calibration procedure:*

  the checkerboard was positioned in the field of view of both cameras under natural sunlight. The custom ROS scripts captured images from both cameras simultaneously. The images were processed to detect the checkerboard corners, and the data was used to compute the homography matrix (H).

Figure 39 - Checkerboard pattern detection

- *Homography matrix (H):*

  The homography matrix is essential for transforming and aligning one image to match the other. This matrix ensures that the corresponding points in the RGB and thermal images overlap correctly, enabling accurate data fusion and analysis.

Through this meticulous calibration process, accurate alignment between the RGB and thermal cameras was achieved, laying the foundation for reliable and precise image fusion. This calibration was pivotal for the subsequent steps in the project, ensuring that the data captured by both cameras could be effectively used for real-time agricultural monitoring and analysis.

4.4 ROS - System Implementation

The implementation of the camera systems through ROS involved several crucial steps and coding tasks to ensure real-time data processing and integration.

After the calibration process, various tests were conducted to verify the accuracy of the homography matrix (H) obtained from the calibration.

Initially, scripts were written to perform an overlay of the topics generated by the two cameras, specifically overlaying the thermal image onto the RGB image. This step was essential to confirm that the homography matrix was functioning correctly and that the images were properly aligned.

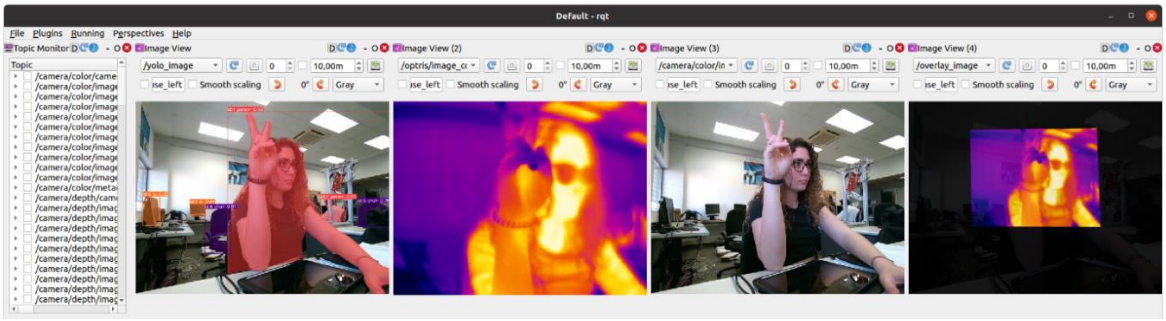


Figure 40 - Testing ROS topics Overlay with H matrix

Once ensured that the overlay made sense and the matrix H was accurate, a ROS node for further processing proceeded to be developed. This node was designed to interpret the pixel data based on the segmentation provided by the YOLOv8 network. Specifically, the YOLOv8 network detects and segments the plant in the RGB image. The ROS node then maps the corresponding thermal image pixels based on this segmentation mask.

Subsequently, additional tests were performed to ensure the ROS node could correctly identify the segmented pixels in the RGB image and match them with the corresponding thermal pixels. This real-time processing enabled the accurate calculation of the average temperature value and the Crop Water Stress Index (CWSI) for the plants.

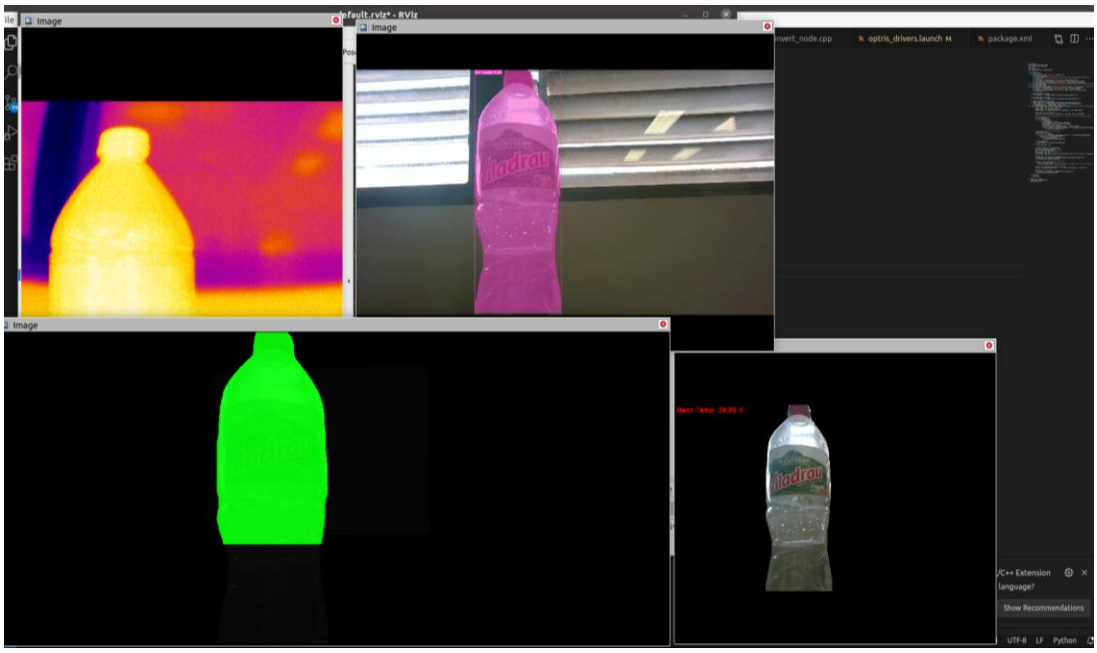


Figure 41 - Testing Pixel Selection for Mean Temperature Computation

The implementation ensured that the system could perform these tasks efficiently on a robotic platform, making it a powerful tool for agricultural monitoring and analysis. To conclude, through this implementation, YOLOv8 effectively detects and segments plants in RGB images. Subsequently, custom ROS code maps this segmented data onto thermal images, enabling real-time computation of average temperature and Crop Water Stress Index (CWSI).

Chapter 5.

Results

This section presents the quantitative and qualitative results from experiments that cover a range of datasets and real-time field tests. The focus is on demonstrating the performance of YOLOv8 model across various datasets and assessing the effectiveness of the real-time Crop Water Stress Index (CWSI) calculation system in practical agricultural scenarios.

5.1 Quantitative analysis

The advanced YOLOv8 framework was employed for image segmentation tasks, utilizing transfer learning techniques to adapt pre-trained models to specific datasets. This section presents the training and validation results of the YOLOv8 segmentation models, showcasing their performance across different datasets.

Each model's effectiveness was assessed through several key metrics, which are explained below:

- *Box loss:* This metric evaluates the alignment between predicted bounding boxes and the ground truth boxes. Lower box loss values signify better performance in terms of accurately locating and sizing objects within images.
- *Segmentation loss (Seg loss):* This loss quantifies the accuracy of the predicted segmentation masks. A lower segmentation loss indicates that the model is more effective at defining the boundaries of objects.

- *Recall (B):* Recall for bounding boxes measures the proportion of actual objects that were detected by the model. A higher recall value reflects the model's ability to detect a larger proportion of the objects present in the dataset.
- *Precision (B):* Precision for bounding boxes assesses the proportion of correctly identified bounding boxes out of all the bounding boxes predicted by the model. Higher precision indicates that the model's predictions are more accurate, with fewer false positives.
- *Mean Average Precision at 50% overlap (mAP50(M)):* This metric evaluates the mean average precision for object classification at an Intersection over Union (IoU) threshold of 50%. It provides an overall measure of the model's accuracy for classifying objects at this level of overlap.
- *Mean Average Precision across IoU thresholds from 0.5 to 0.95 (mAP50-95(M)):* This metric assesses the mean average precision across multiple IoU thresholds, ranging from 0.5 to 0.95. It offers a comprehensive evaluation of the model's performance across different levels of overlap between predicted and ground truth bounding boxes.

The training and validation processes were conducted using several custom datasets, followed by evaluation on unseen test images.

5.1.1 Grapevine Dataset

The YOLOv8m-seg model, trained and validated over 150 epochs, achieved the following results:

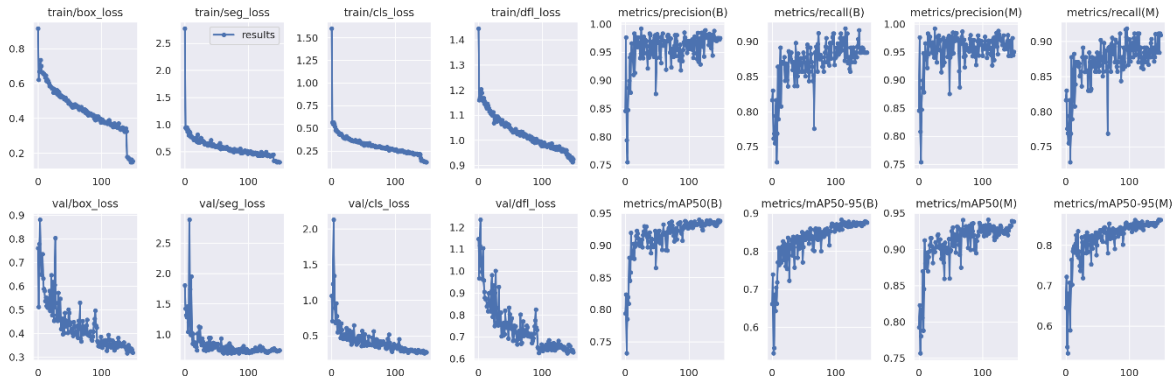
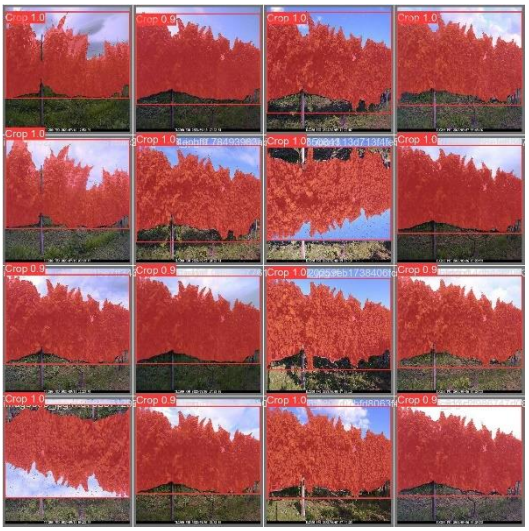


Figure 42 - Training and Validation results for Grapevine



- Model: YOLOv8m-seg
- Training and validation: 150 epochs
- Metrics:
  - mAP: 91.2%
  - Precision: 101.9%
  - Recall: 95.0%

Figure 43 - Vine segmentation: validation pictures results

The model achieved high precision and recall, indicating strong performance in detecting and segmenting vine leaves. The following image displays the training and validation results and showcases the model's predictions on unseen test images, demonstrating its effectiveness.

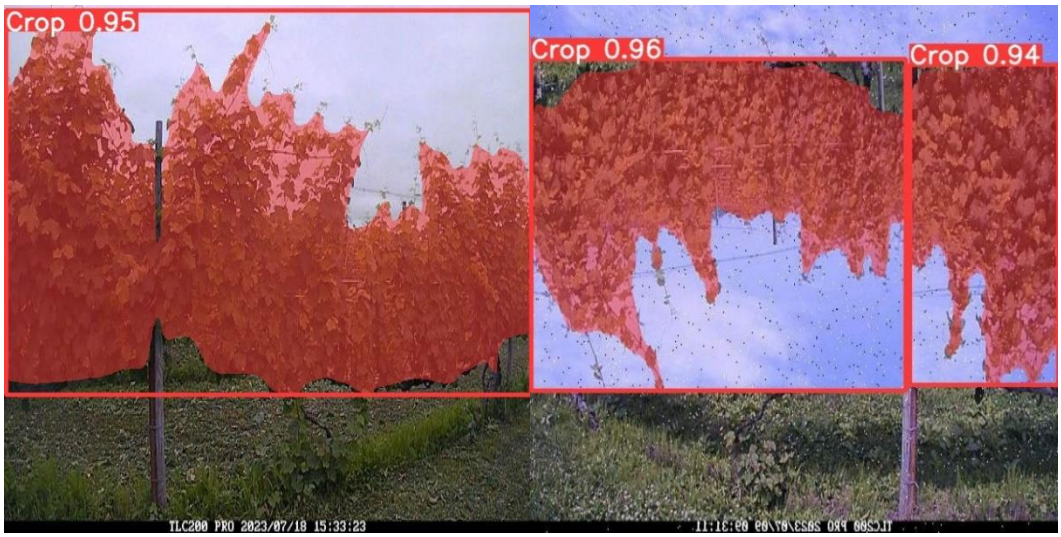


Figure 44 - Examples of vine segmentation prediction on unseen data

5.1.2 Lettuce Dataset

Similarly, the YOLOv8m-seg model, also trained and validated for 150 epochs, demonstrated strong performance:

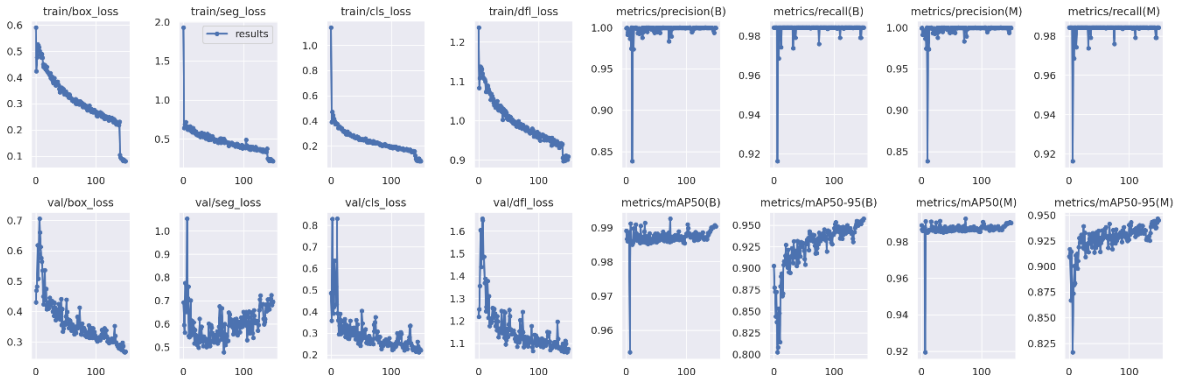
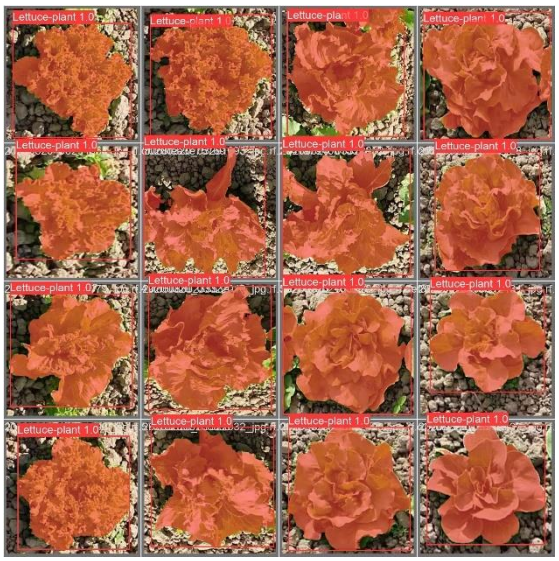


Figure 45 - Training and Validation results for Lettuce



- Model: YOLOv8m-seg
- Training and validation: 150 epochs
- Metrics:
  - mAP: 98.4%
  - Precision: 98.3%
  - Recall: 100.0%

Figure 46 - Lettuce segmentation: validation pictures results

The model performed exceptionally well on the lettuce dataset, achieving near-perfect recall. This demonstrates its ability to accurately detect and segment lettuce plants. The results for the training, validation, and test phases are presented below.

Figure 47 - Examples of lettuce segmentation prediction on unseen data

5.1.3 Leaves Dataset

Utilizing the YOLOv8n-seg (Nano) model, trained for 150 epochs, yielded outstanding results:

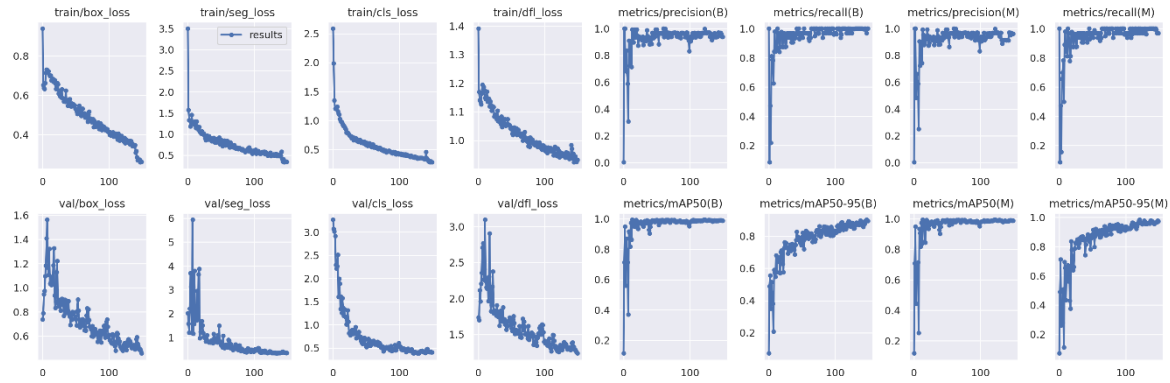


Figure 48 - Training and Validation results for Leaves

- Model: YOLOv8n-seg
- Training and validation: 150 epochs
- Metrics:
  - mAP: 100.0%
  - Precision: 117.0%
  - Recall: 99.2%

Figure 49 - Leaf segmentation: validation pictures results

Using the YOLOv8n-seg (Nano) model for the leaves dataset, the model achieved perfect mAP and very high precision and recall. This model, despite having fewer parameters, was highly effective for this specific task.

Figure 50 - Examples of leaf segmentation prediction on unseen data

Table 4 - YOLOv8 results

Dataset	Model	Training & Validation Epochs	mAP	Precision	Recall
Grapevine	YOLOv8m-seg	150	91.2%	101.9%	95.0%
Lettuce	YOLOv8m-seg	150	98.4%	98.3%	100.0%
Leaves	YOLOv8n-seg	150	100.0%	117.0%	99.2%

The results from these experiments highlight the effectiveness of the YOLOv8 segmentation models across various scenarios. By selecting appropriate model sizes and utilizing transfer learning techniques, strong performance was achieved on different datasets, demonstrating the versatility and capability of YOLOv8 for object detection and segmentation tasks.

5.2 Qualitative analysis

The Crop Water Stress Index (CWSI) serves as a pivotal measure for assessing plant water stress, a crucial factor for optimizing irrigation practices and enhancing crop yields.
The CWSI is determined based on the temperature differences between the plant canopy and reference surfaces under varying environmental conditions.

The primary components involved in the CWSI calculation are:

- *T_canopy*: The average temperature of the plant canopy.
- *T_wet*: The temperature of a reference surface that is fully wet, which simulates the maximum leaf transpiration under the given environmental conditions.
- *T_dry*: The temperature of a reference surface that is completely dry, representing minimal leaf transpiration under the same conditions.

$$CWSI = \frac{T_{dry} - T_{wet}}{T_{canopy} - T_{wet}}$$

This formula quantifies the water stress level of plants by comparing the current temperature conditions to those of fully wet and fully dry reference surfaces. The resulting CWSI value helps in understanding plant water stress, guiding irrigation decisions to improve crop health and yield.

In practical terms, a good CWSI value typically ranges between 0.1 to 0.3, indicating moderate stress levels where irrigation may need adjustment but plants are generally managing. A good CWSI value is usually below 0.1, suggesting minimal water stress and optimal conditions for crop growth and development.

Understanding the CWSI value allows farmers and agricultural specialists to make informed decisions regarding irrigation scheduling, ensuring efficient water use and maximizing crop productivity while minimizing water waste.

5.2.1 Real-time CWSI calculation and visualization

The system designed for real-time CWSI calculation employs a dual-camera setup: one camera captures the RGB image of the plant, while the other captures the thermal image. The YOLOv8 network detects and segments the plant in the RGB image, and custom ROS (Robot Operating System) code processes this segmented data to map the corresponding pixels in the thermal image. This real-time processing allows for the calculation of the average temperature value of the plant canopy as well as the CWSI for the plants.

In practice, the system achieves the following:

- *Detection and segmentation*: YOLOv8 detects and segments the plant in the RGB image, identifying the area of interest for CWSI calculations.
- *Temperature mapping*: The segmented plant data from the RGB image is used to map corresponding pixels in the thermal image to obtain the average temperature of the plant canopy.
- *CWSI calculation*: The CWSI is computed using the average temperature of the plant canopy along with the temperatures of the wet and dry reference surfaces.
- *Real-time display*: The ROS-based application generates a real-time output displaying the CWSI value and the average canopy temperature (T_canopy) on the segmented plant image. Additionally, it overlays the thermal image with the RGB image to provide a visual representation of the temperature distribution and water stress.

The effectiveness of this real-time system is demonstrated through the results obtained in a practical field scenario, in particular related to the lettuce.

A qualitative assessment was conducted on lettuce, initially measuring Twet and Tdry using a separate ROS node for wet and dry lettuce mean computation (*Figure 51*).

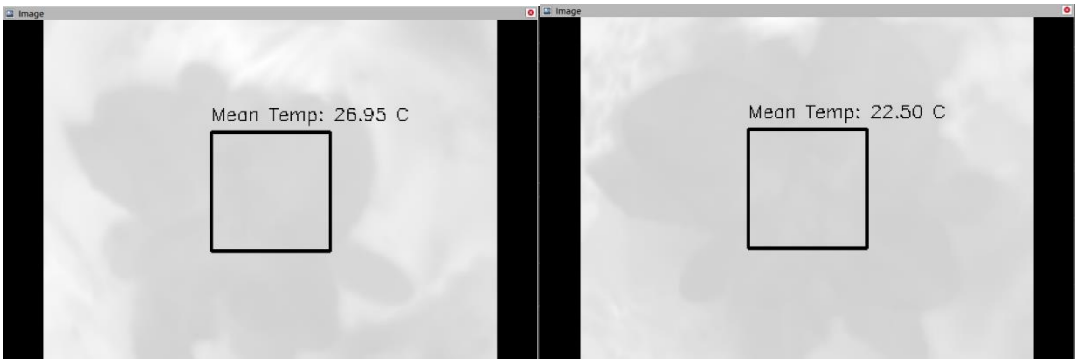


Figure 51 – Lettuce mean temperature: (a) T_dry, (b) T_wet

Figure 52 - Segmented Lettuce with Real-Time CWSI value

Figure 52 shows the segmented lettuce along with the real-time CWSI value and the average temperature of the plant canopy. The visual output includes the CWSI calculation overlayed

on the segmented plant image, providing an immediate assessment of the plant's water stress condition. The result shows how CWSI value indicates minimal water stress, falling within an optimal range, demonstrating effective functionality.

The real-time CWSI calculation and visualization system showcases its capability to detect plant water stress and provide actionable insights. The combination of YOLOv8's segmentation capabilities with the thermal imaging data allows for precise temperature measurements and effective CWSI calculations. The system's real-time output, which includes both the CWSI value and the average canopy temperature, offers valuable information for making irrigation decisions.

The visualization provided by the overlay of thermal and RGB images facilitates an intuitive understanding of plant water stress. By displaying the CWSI value directly on the segmented plant image, the system offers a clear and immediate assessment of plant health. Additionally, the thermal image overlay enhances the ability to visually interpret temperature variations and identify areas of stress.

The qualitative results affirm that the system not only performs accurate CWSI calculations but also delivers effective real-time visualization for practical agricultural applications.

5.3 Further developments

While the initial results from the system are promising, several avenues for improvement exist to enhance its performance, robustness, and applicability. The following sections outline potential improvements or tests for future development:

- **Dataset improvement with application on different crops**

Expanding the dataset to include a diverse range of crop types is essential for enhancing the model's generalizability and robustness. By incorporating various crops into the dataset, the system can be adapted to different agricultural contexts and plant species, leading to more versatile and effective applications in agricultural monitoring. This broader dataset will help the model learn a wider array of features and conditions, improving its performance across different types of crops and environmental scenarios.

- **Automatic calibration each time the cameras are moved**

One critical area for improvement involves the calibration process of the thermal and RGB cameras. Currently, the extrinsic calibration between the cameras is delicate and can be easily disrupted by even minor movements or rotations. For instance, during transportation, as shown in *Figure 53*, the calibration can become imperfect, leading to misalignment between the thermal and RGB images. Implementing an automatic calibration process that adjusts each time the cameras are repositioned would ensure accurate data capture, minimize the need for manual adjustments, and enhance the efficiency of the system. This feature would not only simplify the setup process but also maintain high-quality image overlays for consistent CWSI calculations.

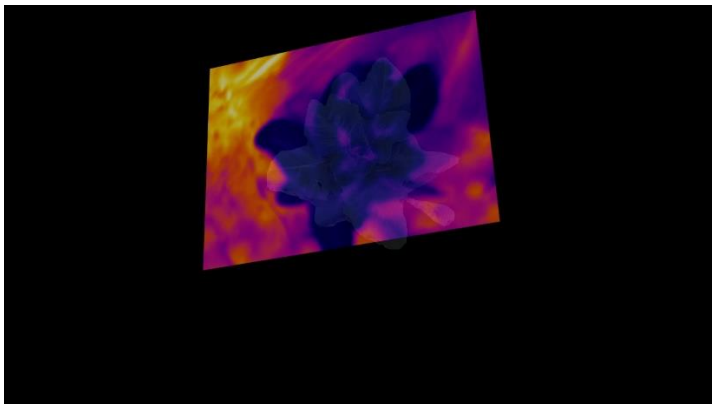


Figure 53 - Mask & thermal image overlay

Figure 53 demonstrates the overlay of the thermal image with the mask of the RGB image of the plant. This visual comparison of the temperature distribution across the plant canopy highlights the importance of accurate calibration for effective CWSI data interpretation.

- **Parallelization of the average temperature calculation**

To improve the real-time performance of the system, it is beneficial to parallelize the calculation of the average temperature when multiple masks need to be processed simultaneously. By distributing the computational workload across multiple processing units, the system's processing time can be significantly reduced. This enhancement would lead to faster CWSI calculations and enable more efficient real-time analysis, thus improving the system's overall responsiveness.

- **Code optimization**

Further optimization of the code can enhance the system's overall efficiency and performance. This includes refining the algorithms used for object detection and temperature measurement, reducing computational overhead, and ensuring smooth integration with the robotic platform. Code optimization efforts should focus on streamlining processes, minimizing resource consumption, and improving the system's responsiveness to changes in environmental conditions.

- **Creation of a turret or robotic arm for camera mounting**

To enhance the versatility and deployment of the system, the development of a turret or robotic arm for mounting the thermal and RGB cameras is proposed. The existing mobile robot platform provides a foundation for this enhancement, allowing for the integration of a new turret or robotic arm mechanism to support various camera configurations and automate adjustments of camera angles and positions.

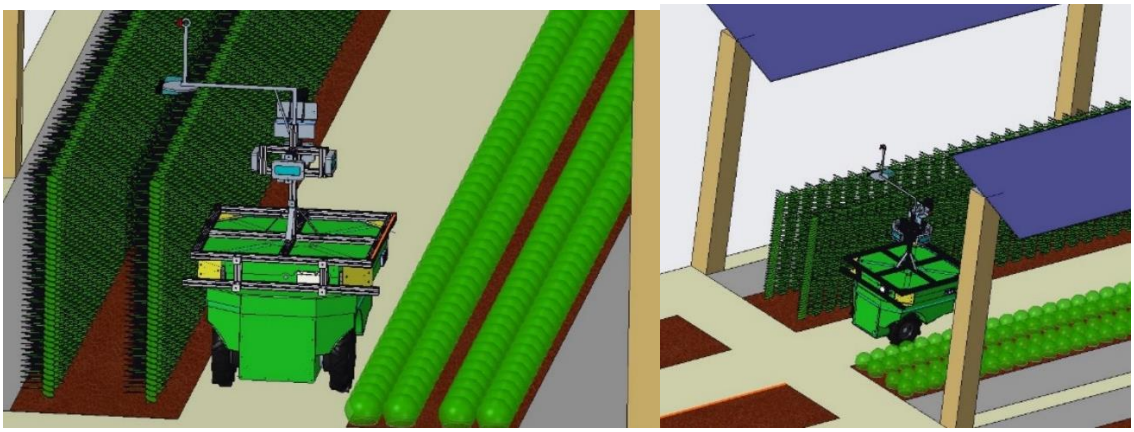


Figure 54 - Agricultural robot design with turret

Figure 54 shows a conceptual design for the turret or robotic arm mounted on the existing mobile robot platform. This design illustrates how the turret or arm would facilitate automated image capture and enhance the system's capabilities for autonomous field operations.

- **Experimenting with thermal camera for YOLO-based segmentation**

A potential experiment involves using the thermal camera directly, ensuring it is properly focused, and training the YOLO (You Only Look Once) model with thermal images for object segmentation. By doing so, the system could effectively identify and segment specific crops, such as lettuce, based on their thermal signatures. This approach could leverage the

unique thermal patterns of plants, potentially improving segmentation accuracy and robustness in various environmental conditions.

By addressing these areas for improvement, the system can be enhanced to better meet the demands of agricultural monitoring and analysis, leading to more effective water stress assessments and improved crop management practices.

Conclusions

In this thesis, an approach to the problem of advanced agricultural monitoring through integrated dual-camera systems was developed. The reason why this approach deserves special attention lies in the chance it gives to contribute to the overall goals of SYMBIOSYST by developing a real-time system able to monitore plant water stress.

Therefore, the successful integration of a dual-camera system combining a thermal camera with an RGBD camera was achieved. This integration enabled simultaneous data acquisition, significantly boosting the system's ability to analyze the environment comprehensively. By merging visual and thermal data, the dual-camera setup provided richer information for subsequent processing tasks.

Furthermore, the thesis optimized plant segmentation using the YOLOv8 model. Through extensive development, YOLOv8 facilitated real-time object detection and segmentation, essential for precise agricultural monitoring and analysis.

One standout achievement was the successful real-time calculation of the Crop Water Stress Index (CWSI) for segmented lettuce. This test demonstrated the correct real-time functionality of the system, providing valuable insights into plant health and water stress levels using thermal data from the integrated camera system.

To conclude, this project demonstrated the feasibility and effectiveness of using integrated dual-camera systems for advanced agricultural monitoring. The achievements in system ideation, camera integration, object detection, CWSI computation, and real-time processing collectively contribute to a robust solution for monitoring plant water stress levels. Future work will focus on refining the system, expanding its applicability to different crops, and further optimizing the processing algorithms to ensure even greater efficiency and accuracy.

Acknowledgements

As I reach the end of this challenging and formative thesis journey, I would like to express my gratitude to those who have supported me throughout this experience.

First and foremost, I am deeply grateful to my supervisor, Prof. Marcello Chiaberge, for granting me the opportunity to embark on this thesis project in collaboration with UPC Barcelona. Despite the geographical distance between us, Prof. Chiaberge was consistently available and engaged, providing invaluable guidance and demonstrating a genuine interest in the progress of my work.

I extend a special thank you to Prof. Alba Perez for her generous sharing of knowledge, as well as for her exceptional professionalism and readiness to offer advice whenever needed. Her insights were instrumental in shaping the direction of my research and her support was greatly appreciated.

I would also like to express my sincere thanks to Prof. David Caballero. During my time at CDEI Barcelona, he welcomed me warmly into the team and played a crucial role in my integration into the office environment despite initial language barriers. His enthusiasm for the research, his willingness to answer all my questions, and his generous help in transporting me to remote locations for image acquisition truly inspired me and enriched my experience.

A special note of gratitude goes to my parents, whose never-failing support has been essential to my academic progress. Their encouragement allowed me to pursue my master's studies in Turin and their continuous backing made my decision to complete my thesis in Barcelona a reality.

I am also thankful to my entire family for their unshakable belief in my abilities and for always standing by me. To my friends, your presence during the most stressful times, your ability to make me laugh, and our study sessions over video calls from various corners of the world have been a source of immense comfort and motivation.

Thank you all for being a part of this significant chapter in my life.










Barcelona

2023-2024

Bibliography

[1] T. S. Huang, "Computer Vision: Evolution and Promise," University of Illinois at Urbana-Champaign.

[2] J. Portley, "KnowHow," August 2023. [Online]. Available: https://knowhow.distrelec.com/it/trasporti/in-che-modo-i-robot-stanno-cambiando-il-panorama-dei-trasporti/.

[3] R. Demush, "A Brief History of Computer Vision (and Convolutional Neural Networks)," *HackerNoon.com,* 28 February 2019.

[4] Y. Aloimonos, Special Issue on Purposive and Qualitive Active Vision, CVGIP B: Image Understanding, 1992.

[5] D. Marr, `Vision: A Computational Investigation into the Human Representation and Processing of Visual Information, San Francisco, 1982.

[6] B. Marr, "7 Amazing Examples Of Computer And Machine Vision In Practice," *Forbes,* 8 April 2019.

[7] N. Pandey, "Lesson 4: Object Detection and Recognition: Advancements in Computer Vision," June 2023. [Online]. Available: https://medium.com/@naveenpandey2706/lesson-4-object-detection-and-recognition-advancements-in-computer-vision-349e61162726.

[8] H. Ashatari, "What is Computer Vision? Meaning, Examples, and applications in 2022," 13 May 2022. [Online].

[9] I. Mihajlovic, "Everything You Ever Wanted To Know About Computer Vision.," 25 Apr 2019. [Online]. Available: https://towardsdatascience.com/everything-you-ever-

wanted-to-know-about-computer-vision-heres-a-look-why-it-s-so-awesome-e8a58dfb641e.

[10] Wikipedia, "Machine learning," [Online]. Available: https://en.wikipedia.org/wiki/Machine_learning.

[11] A. M. Turing, "Computing Machinery and Intelligence," *Mind,* pp. 433-460, 1950.

[12] A. L. Samuel, "Some Studies in Machine Learning Using the Game of Checkers.," *IBM Journal of Research and Development,* pp. 210-229, 1959.

[13] F. Rosenblatt, "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain.," *Psychological Review,* pp. 386-408, 1958.

[14] D. E. H. G. E. &. W. R. J. Rumelhart, "Learning representations by back-propagating errors.," *Nature,* pp. 533-536, 1986.

[15] V. Vapnik, The Nature of Statistical Learning Theory, Springer, 1995.

[16] A. S. I. &. H. G. E. Krizhevsky, "ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems," in *NeurIPS*, 2012.

[17] D. H. A. M. C. J. G. A. S. L. V. D. D. G. .. &. H. D. Silver, "Mastering the game of Go with deep neural networks and tree search.," *Nature,* pp. 529(7587), 484-489, 2016.

[18] J. C. M. W. L. K. &. T. K. Devlin, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019.

[19] R. K. S. P. S. K. S. Shruti Singh, "Artificial Intelligence and Machine Learning in Pharmacological Research: Bridging the Gap Between Data and Drug Discovery," *Cureus ,* August 2023.

[20] J. Brownlee, "Machine learning Mastery," October 2023. [Online]. Available: https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/.

[21] Wikipedia, "Cluster Analysis," [Online]. Available: https://en.wikipedia.org/wiki/Cluster_analysis.

[22] "leonardoaraujosantos.gitbook," [Online]. Available: https://leonardoaraujosantos.gitbook.io/artificial-inteligence/machine_learning/supervised_learning/linear_classification. [Accessed 2024].

[23] A. Corbo, "What Is Reinforcement Learning?," 2023. [Online]. Available: https://builtin.com/artificial-intelligence/reinforcement-learning.

[24] V. Kanade, "What Is Reinforcement Learning? Working, Algorithms, and Uses," [Online]. Available: https://www.spiceworks.com/tech/artificial-intelligence/articles/what-is-reinforcement-learning/.

[25] A. Oppermann, "What Is Deep Learning and How Does It Work?," 2023. [Online]. Available: https://builtin.com/machine-learning/deep-learning.

[26] I. S. G. E. H. Alex Krizhevsky, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems 25 (NIPS 2012)*, 2012.

[27] Y. B. L. B. Y. &. H. P. LeCun, "Gradient-based learning applied to document recognition.," in *Proceedings of the IEEE*, 1998.

[28] A. Z. Karen Simonyan, "Very Deep Convolutional Networks for Large-Scale Image Recognition.," in *International Conference on Learning Representations (ICLR).*, 2015.

[29] S. D. R. G. a. A. F. J. Redmon, "You Only Look Once: Unified, Real-Time Object Detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, 2016.

[30] K. H. R. G. J. S. Shaoqing Ren, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, 2015.

[31] P. F. T. B. Olaf Ronneberger, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2015.

[32] X. Z. S. R. a. J. S. K. He, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, 2016.

[33] D. Team, "The History of YOLO Object Detection Models from YOLOv1 to YOLOv8," June 2023. [Online]. Available: https://deci.ai/blog/history-yolo-object-detection-models-from-yolov1-yolov8/.

[34] G. Boesch, "A Guide to YOLOv8 in 2024," December 2023. [Online]. Available: https://viso.ai/deep-learning/yolov8-guide/.

[35] S. D. R. G. a. A. F. J. Redmon, " "You Only Look Once: Unified, Real-Time Object Detection,"," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA, 2016.

[36] D. C.-E. Juan Terven, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Machine Learning and Knowledge Extraction,* p. 36 pages, 2023.

[37] Ultralytics, "YOLOv8 Documentation," 2023. [Online]. Available: https://docs.ultralytics.com/models/ yolov8/.

[38] Exposit, "Computer Vision Object Detection: challenges faced," April 2021. [Online]. Available: https://becominghuman.ai/computer-vision-object-detection-challenges-faced-9a927f9c5623.

[39] R. Mondal, "Challenges in Computer Vision," Jan 2024. [Online]. Available: https://medium.com/@datasciencejourney100_83560/computer-vision-definition-and-challenges-16051681e6ed.

[40] Wikipedia, "Robotics," [Online]. Available: https://en.wikipedia.org/wiki/Robotics.

[41] Agritecture, "The Role of Robots in Agriculture," April 2024. [Online]. Available: https://www.agritecture.com/blog/exploring-the-future-of-agriculture-a-deep-dive-into-robots.

[42] L. F. P. M. A. P. &. S. M. F. Oliveira, "Advances in Agriculture Robotics: A State-of-the-Art Review," *Robotics,* 2021.

[43] D. H.-L. J. F. O. ,. R. B. T. P. M. A. M. Krishna Ribeiro-Gomes, "Uncooled Thermal Camera Calibration and," *Sensors,* September 2017.

[44] F. N. d. S. A. J. S. V. F. Daniel Queirós da Silva, " Visible and Thermal Image-Based Trunk Detection with Deep," *Journal of Imaging,* September 2021.

[45] V. S.-R. I. B. F. R.-M. ,. e. a. Juan Fernández-Novales, "Monitoring and Mapping Vineyard Water Status Using," *Remote sensing,* 2021.

[46] I. A. C. C. M. J. S. G. X. Dafni Despoina Avgoustaki, " Autonomous Mobile Robot with Attached Multispectral Camera to Monitor the Development of Crops and Detect Nutrient and Water Deficiencies in Vertical Farms," *Agronomy,* 2022.

[47] F. G. J. W. R. L. M. K. Q. Z. Longsheng Fu, "Application of consumer RGB-D cameras for fruit detection and localization in field: A critical review," *Computers and Electronics in Agriculture,* August 2020.

[48] A. BEYAZ, "ACCURACY DETECTION OF INTEL REALSENSE D455 DEPTH CAMERA FOR," in *Agrosym 2022*, 2022.

[49] K. A. F. A. Roselyne Ishimwe1*, "Applications of Thermal Imaging in Agriculture - A review," *Advances in Remote Sensing,* 2014.

[50] I. Realsense, "Intel Realsense," [Online]. Available: https://www.intelrealsense.com/compare-depth-cameras/. [Accessed 2024].

[51] Optris, "Xi 400," [Online]. Available: https://www.optris.com/es/producto/camaras-infrarrojas/serie-xi/xi-400/. [Accessed 2024].

[52] M. A. D. A. a. B. O. Ignacio Rocco Spremolla, "RGB-D and Thermal Sensor Fusion: Application in Person Tracking," in *International Conference on Computer Vision Theory and Applications*, 2016.

[53] P. M. M. B. Stephen Vidas, "3D Thermal Mapping of Building Interiors using an RGB-D and Thermal Camera," in *Proceedings - IEEE International Conference on Robotics and Automation*, 2013.

[54] Augmented Startups, "The Role of Computer Vision in Robotics: Advancements, Applications, and Future Implications," 27 June 2023. [Online].

[55] V. K. B. P. B. S. a. S. H. V. Arakeri M. P., "Computer vision based robotic weed control system for precision agriculture," *International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, 2017,* 2017.

[56] Chiu M. T et al., "Agriculture-Vision: A large aerial image database for agricultural pattern analysis," *IEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* pp. 2825-2835, 2020.

[57] S. P. Kuricheti G., "Computer Vision Based Turmeric Leaf Disease Detection and Classification: A Step to Smart Agriculture," in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*, Tirunelveli, India, 2019.

[58] Wikipedia, "Machine learning," [Online]. Available: https://en.wikipedia.org/wiki/Machine_learning.

[59] J. Brownlee, "A Tour of Machine Learning Algorithms," 2019. [Online]. Available: https://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/.

[60] S. Mishra, "Towards Data Science-Unsupervised Learning and Data Clustering," 2017. [Online]. Available: https://towardsdatascience.com/unsupervised-learning-and-data-clustering-eeecb78b422a.

[61] L. A. Santos, "Leonardo araujo santos book," [Online]. Available: https://leonardoaraujosantos.gitbook.io/artificial-inteligence/machine_learning/supervised_learning/linear_classification.

[62] Y. Xiaozhou, "Towards Data Science," 2020. [Online]. Available: https://towardsdatascience.com/linear-discriminant-analysis-explained-f88be6c1e00b.

[63] "ML," 2017. [Online]. Available: https://ml-cheatsheet.readthedocs.io/en/latest/logistic_regression.html.

[64] N. Kumar, "Geek for Geeks," 2019. [Online]. Available: https://www.geeksforgeeks.org/understanding-logistic-regression/#:~:text=Logistic%20regression%20is%20basically%20a,regression%20IS%20a%20regression%20model.

[65] S. Ruder, " An overview of gradient descent optimization algorithms.," 2016. [Online]. Available: https://arxiv.org/pdf/1609.04747.pdf.

[66] "Guru99," [Online]. Available: https://www.guru99.com/supervised-vs-unsupervised-learning.html#9.

[67]

[68] N. M. I. M. E. R. C. H. S. E. P. Spyros Fountas, "Agricultural Robotics for Field Operations," *Sensors,* 2020.

[69] B. Donaldson, " The challenges posed by global broadacre crops in delivering smart agri- robotic solutions : A fundamental rethink is required.," *Glob. Food Sec,* 2019.

[70] C. F. Ehlig, "Measurement of Energy Status of Water in Plants," *Plant Physiology,* May 1962.

[71] J. Hatfield, "Measuring Plant Stress with an Infrared Thermometer".*National Soil Tilth Laboratory.*

[72] D. S. P. F. M. S. F. Ahmed Kayad, "Latest Advances in Sensor Applications in Agriculture," *Agriculture,* August 2020.