



INSTITUTO  
UNIVERSITÁRIO  
DE LISBOA

---

## **A Deep Learning Toolkit for Water Stress Detection in Viticulture**

Gustavo Laginha Santos Ferreira

### **Master's degree in Computer Engineering**

Supervisor:

PhD Octavian Adrian Postolache, Full Professor,  
ISCTE – University Institute of Lisbon

Co-Supervisor:

PhD Pedro Joaquim Amaro Sebastião, Associate Professor with  
Aggregation,  
ISCTE – University Institute of Lisbon

October, 2024





TECNOLOGIAS  
E ARQUITETURA

---

Department of Information Science and Technology

## **A Deep Learning Toolkit for Water Stress Detection in Viticulture**

Gustavo Laginha Santos Ferreira

### **Master's degree in Computer Engineering**

Supervisor:

PhD Octavian Adrian Postolache, Full Professor,  
ISCTE – University Institute of Lisbon

Co-Supervisor:

PhD Pedro Joaquim Amaro Sebastião, Associate Professor with  
Aggregation,  
ISCTE – University Institute of Lisbon

October, 2024



## Acknowledgments

I would like to thank my family and friends, new and old, for their constant support and motivation. To my wonderful girlfriend, no words describe how grateful I am for your continuous support, since this achievement of mine is only possible because of you.

I would also like to thank my supervisors for the role they played, Professor Octavian Postolache for the support and knowledge he shared during this thesis and Professor Pedro Sebastião for all his personal and professional support since my bachelor's, always motivating me to be better and go further, being a central person in my life today. I would also like to thank Professor Tomás Brandão for all his commitment and dedication to teaching his computer vision classes.

I would also like to express my gratitude to Instituto de Telecomunicações and ISCTE-Instituto Universitário de Lisboa for having supported the laboratories and equipment necessary to develop this work. Additionally, I would like to thank projects SmartFarm 4.0 (national project from collaborative lab, SamrtFarmColab, Projeto mobilizador no 46078 POCI01-0247-FEDER-046078. <https://www.sfcollab.org/smartfarm4>) and SmartVitiNet (European project, co-Funded by the European Union under Grant Agreement number 101083737, <https://www.smartvitinet.eu>).



## Resumo

Esta tese apresenta uma abordagem de *Deep Learning* para a detecção e localização de *stress* hídrico em vinhas, através do desenvolvimento de duas redes neuronais convolucionais. Com base na arquitetura *U-Net*, um modelo inicial foi treinado para fazer segmentação de imagens da base de dados *Agriculture-Vision*, que contém várias imagens capturadas por drones de campos agrícolas. Através de técnicas de transferência de conhecimento, um segundo modelo foi treinado para detetar e localizar *stress* hídrico, usando apenas 40 imagens térmicas únicas capturadas por um drone numa vinha em Lisboa, Portugal. Esta decisão foi motivada pela falta de um modelo pré-treinado para desempenhar este tipo de tarefas e pela quantidade reduzida das imagens térmicas capturadas. O modelo foi avaliado através de várias métricas, incluindo *Binary Cross Entropy loss*, *Precision*, *Recall*, *Intersection over Union* e *Accuracy*, sendo este validado ao criar matrizes de confusão, que demonstram as capacidades dos modelos. Os conjuntos de treino e validação atingiram valores respetivos de 0,17 e 0,26 para a *Binary Cross Entropy loss*, 0,88 e 0,82 para a *Intersection over Union*, 0,91 e 0,89 para a *Precision*, 0,96 e 0,91 para a *Recall* e 0,93 e 0,90 para a *Accuracy*.

Além disso, foi desenvolvida uma plataforma para o modelo final, usando a *Django framework*, escrita em *Python*, para apoiar viticultores a gerir vários projetos durante as diferentes estações do ano. A arquitetura permite carregar, processar e visualizar informação de forma eficiente, assegurando a sua viabilidade e aplicabilidade no mundo real. Os resultados experimentais salientam o potencial dos modelos propostos, demonstrando a sua capacidade como ferramentas viáveis para avaliar e mitigar os efeitos de *stress* hídrico em vinhas.

No futuro, será efetuada uma recolha de dados extensiva em diversas vinhas e ao longo do ano, em colaboração com viticultores para melhorar o algoritmo.

**Palavras-chave:** *Deep Learning*, Viticultura de Precisão, *Stress* Hídrico, Redes Neuronais Convolucionais, Drone, Imagens Térmicas



## Abstract

This thesis presents a Deep Learning approach to water stress detection and localisation, in vineyards, through the development of two convolutional neural network models. Leveraging a U-Net based architecture, an initial model was trained to perform image segmentation on an Agriculture-Vision, a large-scale dataset containing UAV-captured images of farmland. Using transfer learning techniques, a second model was trained to detect and localise water stress on only 40 unique UAV-captured thermal images of a vineyard in Lisbon, Portugal. This approach was motivated by the lack of an existing pre-trained model to perform similar tasks and the reduced number of unique UAV-captured thermal images. The models' performance was evaluated using several metrics, including Binary Cross Entropy loss, Precision, Recall, Intersection over Union and Accuracy, further validated by computing confusion matrices that demonstrated their predictive capabilities. The training and validation sets achieved respective values of 0.17 and 0.26 for Binary Cross Entropy loss, 0.88 and 0.82 for Intersection over Union, 0.91 and 0.89 for Precision, 0.96 and 0.91 for Recall, and 0.93 and 0.90 for Accuracy.

In addition, a platform supporting the final model was built using the Python-based Django framework, structured to assist viticulturists in managing multiple projects across various seasons. The architecture enables efficient data upload, processing, and visualisation, ensuring practical usability in real-world applications. The experimental results highlight the potential of the proposed models, demonstrating their potential as a viable tool to assess and mitigate the effects of water stress in vineyards.

As future work, extensive data collection will be performed in various vineyards and seasons, in collaboration with expert viticulturists to improve the models' predictive capabilities.

**Keywords:** Deep Learning, Precision Viticulture, Water Stress, Convolutional Neural Networks, UAV, Thermal Images



# Index

Acknowledgments	iii
Resumo	v
Abstract	vii
Index	ix
List of Figures	xiii
List of Tables	xv
Acronym List	xvii
Chapter 1. Introduction	1
1.1. Background and Motivation	1
1.2. Precision Viticulture	2
1.3. Main Contributions	2
1.4. Literary Review	3
1.4.1. Research Questions	3
1.4.2. Defining Keywords	4
1.4.3. Defining Search String	4
1.4.4. Defining Search Engines	4
1.4.5. Defining Filters	4
Chapter 2. State of the Art	7
2.1. Precision Agriculture and Viticulture	7
2.1.1. Definition and Importance	7
2.1.2. Main challenges	8
2.1.3. Crop Water Stress	9
2.1.4. Unmanned Aerial Vehicles	10
2.1.4.1. Regulations	12

2.1.4.2.	Sensors and Usages	14
2.1.5.	Remote Sensing	16
2.2.	Artificial Intelligence	16
2.2.1.	Machine Learning	17
2.2.1.1.	Transfer Learning	17
2.2.2.	Deep Learning	18
2.2.2.1.	Convolutional Neural Network	19
2.2.2.2.	Computer Vision	19
2.2.2.3.	Image Segmentation Architectures	20
2.2.2.3.1.	Image Processing Techniques	22
2.2.2.4.	Network Configuration Hyperparameters	22
2.2.2.4.1.	Loss Function	22
2.2.2.4.2.	Dataset Split – Training and Validation	23
2.2.2.4.3.	Evaluating Loss Function on Unseen Data	24
2.2.2.4.4.	Additional Performance Metrics	24
2.2.2.4.5.	Optimisers	26
2.2.2.4.6.	Batch Size	28
2.2.2.4.7.	Number of Epochs	28
2.2.2.4.8.	Learning Rate	28
2.2.2.4.9.	Activation Functions	29
2.2.2.5.	Data Augmentation	30
Chapter 3.	Trained Models	31
3.1.	Methodology Overview	31
3.1.1.	Software Used	31
3.2.	U-Net model	32
3.2.1.	Architecture	32
3.2.2.	Network Hyperparameters Fine-tuning	33

3.3.	Dataset Training - Validation Split	34
3.4.	WeedUNet Model	34
3.4.1.	Dataset Creation	35
3.5.	StressUNet Model	35
3.5.1.	Dataset Creation	35
3.6.	Water Stress Monitoring Platform	36
3.6.1.	Django Overview	36
3.6.2.	Model - View - Template Architecture	36
3.6.3.	Platform Architecture and Key Functionalities	36
Chapter 4. Experimental Validation and Discussion		39
4.1.	Materials and Study Methods	39
4.1.1.	UAV and Onboard Sensors	39
4.1.2.	Network Training Equipment	41
4.2.	Study Site	41
4.3.	Best Performing Models	42
4.3.1.	WeedUNet	43
4.3.1.1.	Hyperparameters	43
4.3.1.2.	Training and Validation Metrics	44
4.3.2.	StressUNet	45
4.3.2.1.	Hyperparameters	45
4.3.2.2.	Training and Validation Metrics	45
Chapter 5. Conclusions and Future Work		49
5.1.	Conclusions	49
5.2.	Future Work	50
References		51
Appendix A. Dissertation Article		57
Appendix B. Images of the data collection		63



## List of Figures

2.1. PA Optimising Cycle .....	7
2.2. EASA UAV Open Category Operational Restrictions ( <i>Source: [26]</i> ) .....	13
2.3. Infrared Region of the Electromagnetic Spectrum Highlighted ( <i>Source: [29]</i> ) .....	15
2.4. AI Venn Diagram .....	16
2.5. Representation of an Artificial Neural Network and a DNN ( <i>Source: [34]</i> ).....	18
2.6. U-Net Architecture ( <i>Source: [46]</i> ).....	21
3.1. Toolkit Fluxogram.....	38
4.1. DJI Mavic 3T Mid-flight.....	39
4.2. Aerial Picture of the Vineyard .....	41
4.3. UAV Captured Thermal Image of the Vineyard.....	42
4.4. Plot of the Performance Metrics of the Best WeedUNet Model.....	44
4.5. Average Confusion Matrix for the Best WeedUNet Model.....	45
4.6. Plot of the Performance Metrics of the Best StressUNet Model.....	46
4.7. Average Confusion Matrix for the Best StressUNet Model .....	47
A.1. Article - Page 1 .....	57
A.2. Article - Page 2.....	58
A.3. Article - Page 3.....	59
A.4. Article - Page 4.....	60
A.5. Article - Page 5.....	61
B.1. Vineyard Photo .....	63
B.2. UAV Landing and Take-off Area.....	63
B.3. UAV Shortly After Take-off .....	63
B.4. UAV Mid-flight.....	63
B.5. Measurements of Leaf Temperature to Establish CWSI Parameters .....	64
B.6. UAV Captured Thermal Image .....	64



## List of Tables

2.1. UAV Designs, Characteristics, Advantages and Disadvantages.....	12
3.1. Hyperparameter Configurations .....	33
4.1. DJI Mavic 3T RGB Camera Specifications ( <i>Source:[86]</i> ).....	40
4.2. DJI Mavic 3T Thermal Camera Specifications ( <i>Source: [86]</i> ).....	40
4.3. Optimal WeedUNet Hyperparameters .....	43
4.4. Optimal StressUNet Hyperparameters .....	45



## Acronym List

ADAM	Adaptive Moment Estimation
AI	Artificial Intelligence
AP	Average Precision
API	Application Programming Interface
BCE	Binary Cross Entropy
CV	Computer Vision
CNN	Convolutional Neural Network
CWS	Crop Water Stress
CWSI	Crop Water Stress Index
DL	Deep Learning
DNN	Deep Neural Network
DSM	Digital Surface Model
DTM	Digital Terrain Model
EASA	European Union Aviation Safety Agency
EM	Electron Microscopy
FN	False Negative
FP	False Positive
FRTT	Field Reference Temperature Technique
GPS	Global Positioning System
IoT	Internet of Things
IoU	Intersection over Union
ISBI	International Symposium on Biomedical Imaging
LiDAR	Light Detecting and Ranging
mAP	mean Average Precision
ML	Machine Learning
MNIST	Mixed National Institute of Standards and Technology
NAG	Nesterov Accelerated Gradient
PA	Precision Agriculture
ReLU	Rectified Linear Unit
RS	Remote Sensing

$\Psi_{stem}$	Stem Water Potential
SGD	Stochastic Gradient Descent
$g_s$	Stomatal Conductance
SLR	Systematic Literary Review
SLR	Systematic Literature Review
TN	True Negative
TP	True Positive
UAS	Unmanned Aerial Systems
UAV	Unmanned Aerial Vehicle
VTOL	Vertical Take-Off and Landing
WS	Water Stress

## CHAPTER 1

# Introduction

This chapter first provides an overview of key technologies and farming practises, such as Precision Viticulture (PV) and Precision Agriculture (PA), with an emphasis on the use of an Unmanned Aerial Vehicle (UAV) platform, equipped with a thermal camera for Water Stress (WS) detection in viticulture. These are further explained in Chapter 2 State of the Art. Secondly, it discusses a literature review, that served as a basis for the investigation regarding the assessment of WS in viticulture, via UAV-captured thermal images.

### 1.1. Background and Motivation

Technologies are more than ever readily available, cheaper and with a greater number of usages. As costs decrease, it becomes sensible to use certain technologies, as it can improve productivity, minimize operational costs and provide insights that would be otherwise impossible to gather.

An UAV is a technology that can gather substantial amounts of aerial data, in a timely manner, with extreme versatility. It serves as a platform for several types of sensors, depending on the application, including, but not limited to, Red-Green-Blue (RGB), Multispectral, Hyperspectral, Light Detecting and Ranging (LiDAR) and Thermal. In addition, the UAV's design can also vary, depending on its intended usage, it can either be a fixed-wing design or a rotor-based design. Fixed-wing designs are more power efficient and require a runway for take-off and landing. In contrast, rotor-based designs are less efficient but can take-off and land vertically, which makes them highly versatile. These platforms and sensors have a wide range of applications, such as aerial photography, and agriculture, among others.

In agriculture, WS occurs when the available water resources either exceed or fail to meet a plant's needs. Particularly, in viticulture, WS directly impacts grape quality and yield quantity. To avoid potential losses, it is essential to accurately and promptly detect WS and take appropriate action. For this, traditional methods are often time consuming, labour intensive and lack detail, as it is only possible to assess a small area. In addition, even if just part of the vineyard is experiencing WS, it will be treated as homogeneously, which will ultimately result in wasted resources and sub-optimal yields. In contrast, a UAV equipped with suitable sensors and associated algorithms can provide fast and accurate results, over a large area, pinpointing areas experiencing WS so that only the affected areas could be treated, ensuring optimal yields and resulting in minimal losses. This makes UAVs a critical tool for WS detection in viticulture.

## **1.2. Precision Viticulture**

PV is a vineyard-farming management practise, that using different technologies, such as varying sensors and platforms, ensures that the yield quality and quantity are maximised, while the resources spent are minimised. In Europe, grape production plays a crucial economic role, therefore, PV is necessary to ensure reliable and sustainable grape production. Despite significant advancements in the fields, PV still faces challenges regarding the understanding of the data collected by the sensors, where these need to be processed so that they could be used to support decision-making. Moreover, the high costs associated are not economically viable in small-scale applications. Lastly, there is a lack of trust in PV because the benefits and limitations are not well understood. To build confidence it requires demonstrable success, so that viticulturists can transition to PV.

## **1.3. Main Contributions**

This thesis introduces a toolkit, supported by a UAV, equipped with thermal and RGB cameras, designed to provide additional information about a crop's water status, enabling viticulturists to make more informed decisions, regarding site-specific irrigation practices. The toolkit is designed to be accessible via a dedicated online website and it uses the aerial thermal images to identify and localise areas of the vineyard experiencing water stress via Computer Vision (CV), powered by an Artificial Intelligence (AI) model, specifically trained for this purpose. By offering site-specific project management, it enables viticulturists to address WS effectively and promptly. Using this tool will save water resources, as they will only be used in necessary

locations, while also providing key insights on vineyard water status, thus ensuring optimal vine health and productivity.

To achieve this, the process to train the model involved using different hyperparameters so that an optimal model was obtained, that is, the model which achieve the best overall performance. First, WeedUNet was trained on an existing aerial agricultural dataset to classify the pixel of each image as being background or a weed cluster. This model was then repurposed to classify the pixels of UAV captured thermal images of a vineyard, as grapevines experiencing WS or not. This second model, StressUNet is then used in an online platform so that viticulturists can upload the UAV captured thermal images and it will pinpoint areas experiencing WS, this way, it is possible to only treat the affected areas using targeted irrigation.

In addition, an article was accepted and published in the 2024 International Symposium on Sensing and Instrumentation in 5G and IoT Era (ISSI) (Figure7A.1. Article - Page 1).

## **1.4. Literary Review**

Based on the objectives defined in the previous chapter, a set of research questions are devised to guide the initial research and development. These questions lay the foundation for conducting a Systematic Literature Review (SLR). This SLR adheres to the guidelines outlined in [1] and its purpose is to gather and evaluate existing literature that is relevant to the research objectives. This SLR was initiated on the 23<sup>rd</sup> of October 2023, and it was concluded on the 11<sup>th</sup> of January 2024.

### **1.4.1. Research Questions**

The validation of this contribution involves considering the defined objectives, leading to the following research questions:

- How can a UAV equipped with a thermal camera be used for WS detection in PV?
- What knowledge extraction and Machine Learning (ML) methods are used for thermal camera data in PA, and what strategies can optimise their performance metrics?
- Are there aerial thermal images suitable for ML based WS detection, or is it necessary to collect additional data? If so, what are the best recommendations to do so?

#### **1.4.2. Defining Keywords**

The research questions informed the extraction of keywords specific to each question, forming a comprehensive set of terms used to guide the search process. This keyword set includes terms such as: “thermal camera”, “precision agriculture”, “precision viticulture”, “unmanned aerial vehicles”, “UAV”, “data extraction”, “computer vision”, “knowledge extraction”, “machine learning”, “optimise performance metrics” and “metrics”.

#### **1.4.3. Defining Search String**

The search string was initially constructed using the defined keywords, separated by the operator "AND." To broaden the scope of the search, a Thesaurus was used to identify synonyms for the original keywords, which were separated by the "OR" operator. Based on the results, the search string was iteratively refined, resulting in the final search string: (“thermal” OR “thermal sensor” OR “thermal imagery” OR “thermal camera” OR “aerial thermal images”) AND (“precision agriculture” OR “precision farming” OR “precision viticulture” OR “precision crop management”) AND (“unmanned aerial vehicles” OR “UAV” OR “drone” OR “unmanned aircraft” OR “remote-controlled aircraft”) AND (“computer vision” OR “machine learning” OR “artificial intelligence” OR “AI” OR “neural networks” OR “deep learning” OR “metrics” OR “optimise performance metrics”).

#### **1.4.4. Defining Search Engines**

To ensure comprehensive coverage of relevant existing literature, several databases were used. The database b-on was chosen due to its capability of aggregating information from several databases, such as IEEE, Springer Link and Web of Science [2]. Furthermore, two other databases were also chosen, namely, ACM, which was selected for its focus on the field of computer science, while Scopus gathers information from numerous fields, not just in the field of computer science [3], [4]. This combination of databases ensures that a wide range of studies are selected and analysed.

#### **1.4.5. Defining Filters**

The search string was applied using several filters to include studies that contained all required conditions in the search string. To achieve this, the following filters were devised:

- Apply search string to title, abstract and keywords;

- Only studies in Portuguese and in English are accepted, and they must be freely available;
- Remove duplicates;
- Analyse the contents of the abstract;
- Analyse the full article or study.

Applying these filters successively resulted in 22 studies which were fully reviewed and integrated into the initial research for this thesis.



## CHAPTER 2

# State of the Art

This chapter focuses on the different aspects related to the development of the toolkit, starting with the definition of Precision Agriculture (PA) and Precision Viticulture (PV). This is followed by an evaluation of Remote Sensing platforms, with a focus on UAV equipped with thermal cameras and how they can be used to evaluate WS in vineyards. Lastly, key aspects of Artificial Intelligence (AI) are outlined, which are necessary to understand how to develop an algorithm that can detect and localise WS in vineyards, based on UAV-captured thermal images.

## 2.1. Precision Agriculture and Viticulture

### 2.1.1. Definition and Importance

The International Society of Precision Agriculture is a non-profit professional scientific organization that defines PA as “a management strategy that gathers, processes and analyses temporal, spatial and individual plant and animal data and combines it with other information to support management decisions according to estimated variability for improved resource use efficiency, productivity, quality, profitability and sustainability of agricultural production.” [5]. With regards to this statement, PA is a farming management technique that utilizes technologies, such as the Internet of Things (IoT), AI and UAVs, to optimize inputs and improve productivity. This technique applies farming inputs, in a time and location-sensitive manner, aiming to maximise yield quality, quantity and profitability [6], [7], [8]. In contrast to traditional farming practises, a field is divided into multiple sections, which can then have a specific approach and thus lead to goals stated previously [7].

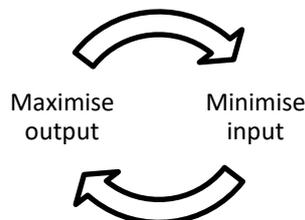


FIGURE 2.1. PA Optimising Cycle

FIGURE 2.1 represents this site-specific strategy, and the relationship between minimising inputs and maximising outputs. Examples of inputs in PA include water, fertilizers, pesticides, and labour. By optimising these inputs through precise application and monitoring, it is possible to achieve outputs such as higher crop yields, improved crop quality, reduced environmental impact, and increased profitability.

In the context of global risks, the World Economic Forum outlines, in the 2024 global risk report, climate-related threats as the dominant risk throughout the next decade. These pose a serious risk, in terms of food, water and health security. PA emerges as a possible solution to reduce the global water consumption [9].

PV suffered significant advancements in recent years, driven by the integration of advanced technologies. The use of sensors to capture data indirectly, remote platforms and ML algorithms allow for precise monitoring and management of vineyards. These technologies can provide real-time information on vine health and growth, pathogen identification and WS detection. Additionally, the development of decision support systems to process the sensor captured data, enable viticulturists to make data-driven decisions [10].

In 2020, Europe produced 63% of the world's total wine production, and in viticulture WS directly impacts grape yield and quality, thus, it is necessary to detect it accurately and reliably, to avoid potential losses [11]. Traditional methods of monitoring WS involve *in situ* measurements, which are time-consuming, labour intensive, costly and, often, destructive [7], [10].

### **2.1.2. Main challenges**

Despite the technological advancements, the practical implementation of PV still faces several challenges. One of the main challenges involves the interpretation and management of the significant amount of data captured by the various sensors and platforms. This data then needs to be processed and displayed in meaningful ways so that it can be interpreted and turned into actionable insights. Furthermore, the technologies used in PV require a considerable investment and, particularly in small-scale applications it's still not economically viable [12].

Another challenge that hinders the implementation of PV involves the scepticism and lack of trust by viticulturists, regarding these new technologies. It is paramount that the benefits and limitations of PV are understood, and that viticulturists are provided with adequate training and education to adopt these technologies effectively. To build confidence it is necessary to provide measurable success, reliable and easy to use systems to ensure viticulturists transition from traditional practises to PV [10], [12].

### 2.1.3. Crop Water Stress

A certain crop naturally transpires at a certain rate, influenced by weather conditions and available resources. As water resources become insufficient to meet a plant's water needs, stomatal closure gradually increases, which causes a reduction in evapotranspiration. The consequence of less evapotranspiration is the increase in temperature which causes WS. In addition, excess water resources also cause WS, leading to diseases that directly impact a plant's health [13], [14].

Traditional methods to determine the WS use parameters, including stomatal conductance, ( $g_s$ ) and stem water potential ( $\Psi_{stem}$ ). Nevertheless, to obtain such parameters is time-consuming, laborious and costly [15].

A plant's temperature thereby serves as an indicator of water availability and is obtained via thermography, as leaf temperatures increase as result of a decrease of  $g_s$ , due to insufficient water [14], [15]. Therefore, thermal data strongly correlates with crop water stress.

To quantify plant level water status by means of thermography, the Crop Water Stress Index (CWSI) is the most widely used metric, as stated by Awais et al.[16]. Initially proposed by Idso et al. [17], obtained by equation (3.1):

$$CWSI = \frac{T_c - T_{wet}}{T_{dry} - T_{wet}} \quad (3.1)$$

In equation (3.1),  $T_c$  is the canopy temperature (in °C),  $T_{wet}$  corresponds to the temperature of a leaf transpiring at its maximum potential rate (in °C), and  $T_{dry}$  is the temperature of a non-transpiring leaf (in °C) and it refers to the temperature of a non-transpiring leaf.

The CWSI is a normalised value, where 0 indicates full transpiration, and values close to 1 signify increasing water stress derived from reduced transpiration. In vineyards, CWSI values above 0.5 serve as a good indicator that the plants are under water stress [15].

Sepúlveda-Reyes et al. [15], evaluates varying methods to estimate the variables necessary to calculate CWSI, namely,  $T_{dry}$  and  $T_{wet}$ , which are measured across different canopy zones and irrigation conditions. These methods are compared based on their relationship to plant-based variables, namely stomatal conductance,  $g_s$  and stem water potential,  $\Psi_{stem}$ . The Field Reference Temperature Technique (FRTT), proposed by Jones et. al.[18] is a thresholding technique used to determine  $T_{wet}$  and  $T_{dry}$ . This technique involves selecting two mature and intact leaves from different zones of the plant canopy (sunlit and shaded<sup>1</sup>), where one leaf is coated with a water-detergent solution, simulating maximum transpiration rate, producing the  $T_{wet}$ . The second leaf is covered in petroleum jelly to prevent transpiration, establishing the  $T_{dry}$  value [15]. The researchers conclude that sunlit and shadowed canopy temperature measurements provide similar results and CWSI estimations via FRTT remains an adequate choice due to its simplicity, minimal external variables and practical application.

Similarly, in a study by Awais et al. [16], researchers also use CWSI, where the temperature of five sunny and five shaded leaves are measured using a handheld thermometer. This methodology allows for the effective monitoring of water stress in crops, as it demonstrates a strong correlation between CWSI values and  $g_s$ , without the need of a non-water stress baseline, as well as strong correlation between UAV  $T_c$  and ground truth  $T_c$ , particularly when measurements are taken around midday. In another study the researchers conclude the optimal parameters to capture aerial thermal images, which is further explained in a later section [8].

#### 2.1.4. Unmanned Aerial Vehicles

Unmanned Aerial Systems (UAS), as defined by the European Union Aviation Safety Agency (EASA), consist of an unmanned aircraft and its associated remote control [19]. The terms ‘UAV’ or ‘drone’ particularly refers to the unmanned aircraft component of a UAS. EASA categorises these UAVs into several distinct groups, each with unique characteristics, advantages and disadvantages. These categories can be grouped into two distinct groups, fixed wing and rotor-based designs, in which the primary distinction lies in their lift generation.

---

<sup>1</sup> Sepúlveda-Reyes et al. [15] note that while some researchers established high correlations between sunlit canopy temperature,  $g_s$  and  $\Psi_{stem}$ , others have concluded that, owing to less temperature variations, shaded canopy temperatures show better correlations with  $g_s$  and  $\Psi_{stem}$ .

Fixed-wing designs use wings to produce lift, similar to an airplane, use stationary wings during flight and rely on a forward thrust to move air both under and over the wings, and control surfaces, to generate lift and manoeuvre. In contrast, rotor-based designs employ powered rotors. Rotor-based designs can be single rotor, multi-rotor and lift and cruise or vectored thrust. Firstly, single rotor designs are comparable to a helicopter, that features a single rotor, which provides both lift and propulsion and requires a tail rotor to maintain stability and control yaw movement. Secondly, multi-rotor designs have at least two rotors and by adjusting their speed, they can generate lift and manoeuvre the UAV (roll, pitch and yaw). Lastly, lift and cruise or vectored thrust designs combine rotors and have at least, a fixed wing, which retains the benefits of both designs, namely Vertical Take-Off and Landing (VTOL) capabilities and flight endurance. This is achieved by having a set of rotors, to provide vertical thrust (lift and cruise) or by having pivoting rotors (vectored thrust) [20], [21], [22]. Table 2.1 summarises the different UAV designs, along with their differences, benefits and drawbacks.

TABLE 2.1. UAV Designs, Characteristics, Advantages and Disadvantages

	Fixed wing <sup>2</sup>	Rotary wing		
		Single rotor <sup>3</sup>	Multi rotor <sup>4</sup>	Lift and cruise or vectored thrust <sup>5</sup>
<b>Characteristics</b>	<ul style="list-style-type: none"> <li>Stationary wings provide lift.</li> <li>Control surfaces on the wings, horizontal and vertical stabilizers control the movement.</li> </ul>	<ul style="list-style-type: none"> <li>Single rotor used for lift and propulsion.</li> <li>Tail rotor maintains stability and controls yaw movement.</li> </ul>	<ul style="list-style-type: none"> <li>Uses at least two rotors control lift, propulsion and movement (quadcopter, hexacopter, etc).</li> </ul>	<ul style="list-style-type: none"> <li>Combines stationary wings and rotors.</li> </ul>
<b>Advantages</b>	<ul style="list-style-type: none"> <li>Longer flight times.</li> <li>Heavier payload capacity.</li> <li>Higher high-wind stability</li> <li>Better power efficiency.</li> </ul>	<ul style="list-style-type: none"> <li>VTOL capabilities.</li> <li>Longer flight times.</li> <li>Heavier payload capacity</li> </ul>	<ul style="list-style-type: none"> <li>VTOL capabilities</li> <li>High versatility and manoeuvrability.</li> <li>Ease of use and automation.</li> </ul>	<ul style="list-style-type: none"> <li>VTOL capabilities</li> <li>Longer flights.</li> <li>Better power efficiency</li> </ul>
<b>Disadvantages</b>	<ul style="list-style-type: none"> <li>Requires a runway to take-off and land.</li> <li>Limited to forward flight.</li> <li>Less manoeuvrable.</li> <li>Design associated risks (stalling, flat spin, etc.)</li> <li>Inability to hover.</li> </ul>	<ul style="list-style-type: none"> <li>More dangerous.</li> <li>Harder to fly, more training needed.</li> </ul>	<ul style="list-style-type: none"> <li>Lower flight time</li> </ul>	<ul style="list-style-type: none"> <li>High development cost.</li> <li>Complex.</li> <li>Expensive</li> </ul>

#### 2.1.4.1. Regulations

As the use of UAVs expands across numerous sectors, various organisations worldwide, including the Federal Aviation Administration [23] in the United States of America, EASA [19] in the European Union, and the Civil Aviation Safety Authority [24] in Australia, have established a set of requirements, restrictions and guidelines, to ensure, above all, safe and ethical UAV operations [25].

<sup>2</sup> Compared to rotary wing UAV designs.

<sup>3</sup> Compared to multi rotor UAV designs

<sup>4</sup> Compared to other rotor-based designs

<sup>5</sup> Compared to other rotary wing UAV designs, when equipped with retractable rotors [22].

To ensure the above mentioned, EASA divides drone operations into three categories: open, specific and certified. On the one hand, the differences between open and specific are when operations have a higher risk and the UAV fails to meet a criterion of open category, or when operating beyond visual line of sight. Furthermore, the open category defines the restrictions and requirements for leisure and low-risk commercial applications. This category is divided into sub-categories<sup>6</sup>, that specify the operational restrictions, such as maintaining a maximum take-off under 25 kg, ensuring visual line-of-sight, flying at altitudes below 120m, and keeping a safe distance from uninvolved people [19], [26].

Figure 2.2 provides an overview of the operational restrictions for a UAV, operating within the open category, as defined by EASA.

C-Class	Max Take off mass	Subcategory
Privately build	<b>&lt;250g</b> 	<b>A1</b> Not over assemblies of people <small>(can also fly in subcategory A3)</small>
legacy < 250g		
C0		
C1	<b>&lt;900g</b> 	<b>A2</b> Fly close to people <small>(can also fly in subcategory A3)</small>
C2	<b>&lt;4kg</b> 	
C3	<b>&lt;25kg</b> 	<b>A3</b> Fly far from people
C4		
Privately build		
Legacy drones (art 20)		

FIGURE 2.2. EASA UAV Open Category Operational Restrictions (Source: [26])

On the other hand, the differences between specific and certified category are that UAV operations involve greater risks that require certification, or that the UAV is larger than 3m and flies over people, transports people or carries dangerous goods [19].

---

<sup>6</sup> The subcategories in the open category are A1 (UAV may fly over people but not assemblies), A2 (UAV may fly close to people, at least 5 meters) and A3 (UAV must fly, at least, 150 meters away from people) [26].

Ultimately, these measures guarantee that a pilot operates UAVs within legal boundaries and prioritises safety, above all, taking into consideration constraints imposed by the UAV's C-Class<sup>7</sup>, its weight and sub-category.

#### **2.1.4.2. Sensors and Usages**

In UAV RS applications for PA, five types of cameras and sensors are considered, including RGB, multispectral, hyperspectral, LiDAR and thermal.

RGB cameras are low cost, easy to use, have limited spectral resolution and high spatial resolution. These have successfully been used for crop phenotyping (such as, canopy height and lodging), to calculate Vegetation Indexes (VIs), to extrapolate Digital Terrain Models (DTMs) and Digital Surface Models (DSMs). Owing to the limited spectral resolution, these cameras are inadequate for the analysis of crop phenotypic knowledge for physiological traits and crop disease diagnosis [6], [27].

Besides RGB cameras, a multispectral camera provides a broader range of spectral bands, from the invisible and visible light spectrum, as it is comprised of several sensors, capable of capturing specific spectral band. However, multispectral cameras have a higher associated cost, compared to RGB cameras. In terms of usability, these cameras are suited for a wide range of applications, such as drought stress detection, estimation of nutrients status, pathogen detection, determination of growth vigour and yield estimation [6], [27].

Additionally, hyperspectral cameras collect data that covers a substantial spectral region between 400 nm and 1000 nm, in the form of continuous narrow bands of less than 10 nm. The main drawback associated with this type of camera is the high acquisition cost. Nevertheless, these have been successfully used to estimate soil nutrient status, pathogen identification and weed detection [6], [27].

Moreover, LiDAR sensors work by emitting a pulsating laser, that travels at a constant speed (the speed of light) and as it hits an object's surface, it bounces back to the sensor. This, in turn, permits LiDAR sensors to calculate accurate distances between two points, the point of collision and the emitting sensor. Similarly to hyperspectral cameras, LiDAR sensors also involve a significant acquisition cost and the need of specialised onboard equipment, such as

---

<sup>7</sup> For UAVs sold in the EU, the manufactures are required to label them with a C-Class identification. This ensures that the UAV complies with applicable regulatory requirements [19].

Global Positioning System (GPS). Nevertheless, these sensors have been used to monitor crop height and health, and to estimate yields, soil properties and pesticides [28].

Lastly, thermal cameras can either be cooled or uncooled. On the one hand, cooled image detectors, while highly sensitive, are often large, expensive, and consume a lot of energy, making them unsuitable for UAVs. On the other hand, uncooled image detectors, such as microbolometers, are lighter, consume less energy and are less complex, as they do not require cooling. This makes uncooled thermal cameras ideal for UAV applications. Despite being less accurate, they are sensitive in the thermal infrared range of the electromagnetic spectrum, that is, the range between 3  $\mu\text{m}$  to 100  $\mu\text{m}$ , as shown in Figure 2.3. The lenses of these sensors are made from a material, that blocks visible light while letting infrared radiation pass through [29].

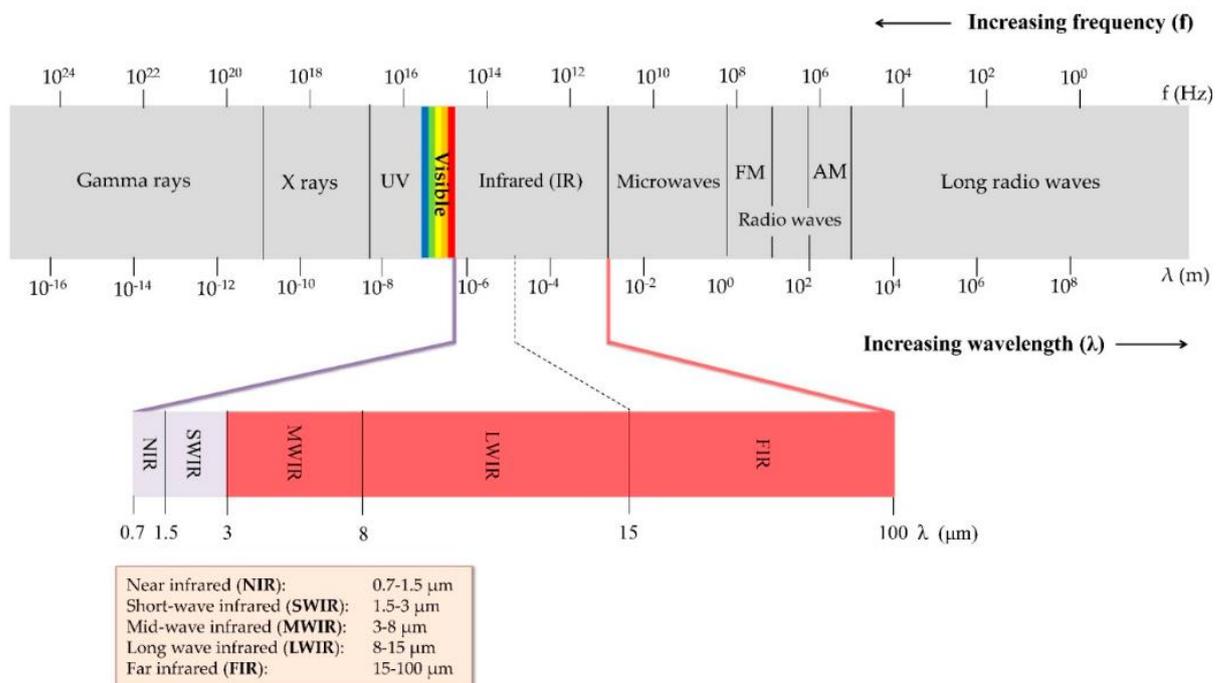


FIGURE 2.3. Infrared Region of the Electromagnetic Spectrum Highlighted (Source: [29])

These sensors have been used to assess water stress, extract canopy temperatures, detect pathogens and determine soil salinity. When using these sensors, the main drawbacks associated include errors in measurements, induced by atmospheric attenuation, sensor calibration and higher cost, compared to RGB cameras [6].

To mitigate these drawbacks associated with thermal cameras and reduce measurement errors, Awais et. al. [8] concludes that the optimal flying height of a UAV equipped with a thermal camera to extract canopy temperatures is at 60 meters, at 11 am. In addition, they also propose that before each flight the UAV should be allowed to warmup for at least 30 minutes

and that the flight mission should be conducted with a forward and side overlap varying from 80 to 85 percent.

### 2.1.5. Remote Sensing

Remote Sensing (RS) involves the use of technologies to indirectly collect information that is both accurate and timely. In the context of PA, RS provides a way to collect data that would be impractical, costly or difficult to gather by other means [7], [30]. The collection methods associated with RS can either be aerial or ground based. With this, ground-based methods involve direct contact with the study subject and are time consuming, labour-intensive, costly and destructive [7]. In contrast, aerial-based RS platforms such as satellites, airplanes, and UAVs, have been used in PA, with varying applications, characteristics, advantages and disadvantages [6], [13], [30]. High altitude platforms such as satellites are characterised by having low spatial resolution, long revisit times, high associated costs and affected by weather phenomena, such as clouds [31]. Airplanes are a lower altitude alternative, which offer finer spatial resolution and greater control over temporal resolution [32]. Nevertheless, for small scale applications, these are not a viable option [13].

## 2.2. Artificial Intelligence

Artificial Intelligence (AI) enables a machine the ability to simulate human intelligence, by learning without the need of being programmed for that specific task. Additionally, ML is a subset of AI that contains Deep Learning (DL). The main difference between ML and DL algorithms resides in the fact that ML algorithms learn computational behaviours from data, whereas DL algorithms attempt to extract features from data [33], [34]. Figure 2.4 explains the relationship between AI, ML and DL.

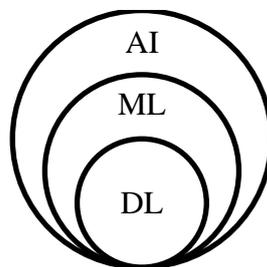


FIGURE 2.4. AI Venn Diagram

### **2.2.1. Machine Learning**

ML is a branch of AI that enables computers to learn new skills, through data, without directly programming the skills. With regards on how a machine learning algorithm is trained, the learning process can be supervised, unsupervised or by reinforcement.

Supervised ML involves training a model on a dataset that includes both input and expected output data. With the provided data, the model gains knowledge from the relationship between the input and output data, that is, it learns from examples with a known outcome, which enables it to classify and predict new inputs.

Conversely, unsupervised ML algorithms train on data that has no expected output. Without the expected outcome, the model learns to establish connections between the provided data, based on their characteristics. Thus, these algorithms learn solely based on the characteristics of the provided data.

Lastly, reinforcement learning is a process where the algorithm learns as it interacts with an environment, where based on its decisions it receives either an encouragement or penalty [33], [34], [35].

#### **2.2.1.1. Transfer Learning**

Transfer learning is a ML learning technique that leverages the knowledge gained from a previously trained network to address similar tasks or problems. This approach improves the model's generalisation capabilities by applying learned features from the initial task to a new, related task.

Alternatively, traditional ML methods require building a new model for each task from scratch, considering that the data for training, validation and testing originates from the same feature space. In contrast, transfer learning avoids the need for extensive retraining by repurposing pre-trained models, thus reducing computational costs and the need for large-scale datasets [36], [37].

The steps to implement transfer learning involve:

- Freezing the layers from the pre-trained model, to prevent overwriting the existing weights;
- Add additional trainable layers that will use the pre-trained features to make predictions for the new task.

### 2.2.2. Deep Learning

DL takes inspiration from nature, as it processes data analogously to a human brain. These networks learn to map the input and generate an output, in either regression, that is, to predict a continuous value, or in classification tasks, i.e., to produce class predictions.

DL algorithms use a Deep Neural Network (DNN) which organise the base units called neurons, or perceptrons, across multiple layers, like traditional neural networks. Associated to each neuron, there are weights and bias. The weights determine the strength of the connection between neurons in adjacent layers. In addition to weights, the bias shifts the activation function by adding a constant, allowing neurons to activate when the input is zero.

A DNN consists of several different layers, namely, an input layer, several hidden layers and an output layer. This contrasts traditional artificial neural networks, which have a single hidden layer, instead of several hidden layers, as shown in Figure 2.5. Having more than one hidden layer enables DL algorithms to solve more convoluted and abstract problems [34].

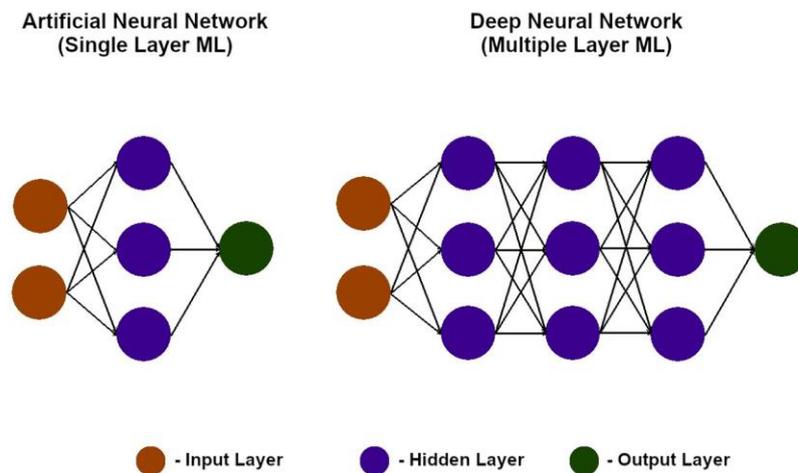


FIGURE 2.5. Representation of an Artificial Neural Network and a DNN (*Source: [34]*)

The input layer receives data and feeds it to the next layers. This also occurs in subsequent layers, where the next layer's input is the output of the previous layer. Hidden layers consist of several neurons that perform hierarchical representations of that data, gradually extracting higher level features<sup>8</sup>, i.e., each hidden layer performs feature selection as it transforms the data in a more abstract representation [34], [35].

---

<sup>8</sup> For example, a colour is a low-level feature while a facial structure is a high-level feature.

During training, there are two distinct phases, the forward propagation and the backpropagation. The process of data moving through the network, from the input layer to the output layer, is called the forward propagation phase. Once forward propagation is completed, the model evaluates its performance by calculating the errors in its predictions. These errors are, then, used in backward propagation, where the model adjusts the weights and biases by moving backward through the layers. This cycle is repeated during training until the model converges [38], [39], [40].

### **2.2.2.1. Convolutional Neural Network**

A Convolutional Neural Network (CNN) is a supervised learning, feed-forward, neural network that learns to map the input to the expected output. A CNN consists of convolutional layers, pooling layers and fully connected layers, where in each layer the neurons are arranged in three dimensions<sup>9</sup>. The convolutional layers use filters or kernels with coefficients<sup>10</sup> that have been obtained during the training phase, to extract features from the data. Each filter convolutes over the input, to create a feature map. The feature map contains the highest activation values, obtained via the filter's convolutions. The pooling layers reduce the dimension of feature maps and contribute to a reduction in computation requirements. These layers form the basis to extract features from the data. In these networks, the fully connected layers integrate localized attributes to combine overarching characteristics [35], [36].

### **2.2.2.2. Computer Vision**

CV enables computers to interpret and derive meaningful information found in digital images. It encompasses several tasks such as image recognition<sup>11</sup>, object detection<sup>12</sup> and image segmentation [41], [42].

---

<sup>9</sup> The neurons' dimensions are the width, height and depth. The depth refers to the third dimension of an activation volume. For example, in the case of an RGB image, the depth would be the numbers of channels in that image, i.e., 3.

<sup>10</sup> The filter's coefficients are the weights of the neuron connections.

<sup>11</sup> Involves the identification of objects or features, found within an image [42].

<sup>12</sup> Localizes and classifies multiple objects within an image [42].

Image segmentation specifically involves the process of labelling each pixel in an image with its corresponding class, which requires recognising objects and accurately delineating their boundaries. Modern approaches to image segmentation utilise advanced neural network architectures, such as fully convolutional networks and feature pyramid networks, to enhance accuracy and resolution [42].

### **2.2.2.3. Image Segmentation Architectures**

Firstly, YOLO11 is the latest model in the YOLO series from Ultralytics, it introduces higher accuracy, speed and efficiency, compared to previous architectures. Improvements to the architecture and training pipelines have resulted in better feature extraction and faster training speeds, while reducing the number of parameters. Depending on the model, YOLO11 can perform several CV tasks, including object detection and image segmentation. The largest model for image segmentation contains 62.1 million parameters and it was trained on the COCO dataset. However, by using YOLO11, the software developed is required to be open source, unless a paid license is acquired [43].

Secondly, VGG19 is a CNN comprised of 16 convolutional layers and 3 fully connected layers, totalling 19 layers. The model was designed to perform classification and localisation tasks on the 2012 ImageNet dataset. Despite the robust performance, the elevated number of convolutional layers results in the network having 144 million parameters, which required 1.3 million training images which results in high computational demands, long training times and training [44].

Thirdly, MobileNet is designed for mobile applications, where computational resources are limited. It focuses on reducing the architecture size and latency, and uses depthwise separable convolutions, which applies a single filter to each input channel, followed by pointwise convolution, that is, a 1x1 convolution, resulting in 28 layers. The MobileNet contains 4.2 million parameters, minimising computational cost while maintaining high accuracy and a reduced model size [45].

Lastly, the U-Net architecture (Figure 2.6), proposed by Ronneberger et al. [46] is a CNN designed for biomedical image segmentation, where the output is designed to include the class label assigned to each pixel. This network combines a contracting path for context capture and an expanding path for precise localisation, which results in an effective model for image segmentation tasks having won the Electron Microscopy (EM) segmentation challenge at International Symposium on Biomedical Imaging (ISBI) 2012, with only 30 training images.

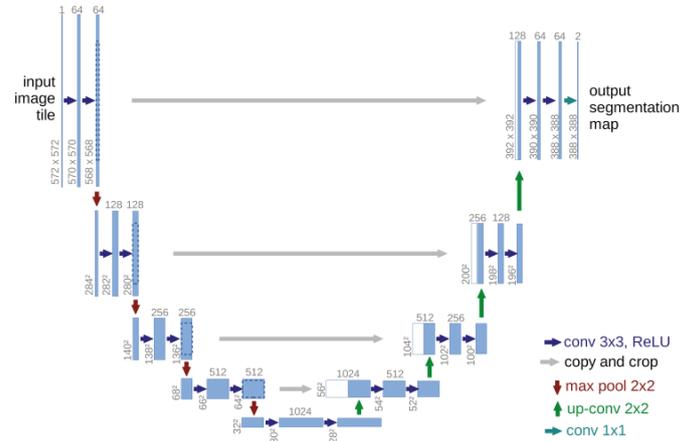


FIGURE 2.6. U-Net Architecture (*Source: [46]*)

The use of skip connections between the contracting and expanding paths allow high-resolution features from the contracting path to directly transfer to the corresponding layers in the expanding path. This approach preserves information that would otherwise be lost during down-sampling, ensuring that the network retains fine details and context. Additionally, U-Net’s architecture is absent of fully connected layers, which enables it to process images of varying sizes without needing to adjust the network’s structure. Instead, the network relies on unpadded convolutions, resulting in a segmented output that is smaller than the input but achieves pixel-level accuracy by minimising border-related inaccuracies through mirroring [46].

The contracting path uses repeated 3x3 convolutions with ReLU activation, followed by 2x2 max-pooling operations, which down-sample the feature maps and increase the number of filters at each stage. This path captures increasingly complex features at progressively lower resolutions. In the expanding path, up-convolutions (2x2 transposed convolutions) are used to up-sample the feature maps, followed by 3x3 convolutions. Each up-sampling step halves the number of feature channels, whilst skip connections, with the corresponding contracting path layers, enhance spatial precision by combining contextual and fine-grained details [46].

Moreover, U-Net employs extensive data augmentation techniques, due to the limited number of available images. The resulting network’s architecture and training strategy enable it to generalise effectively from few samples, making U-Net a robust solution for high-precision segmentation tasks with limited data availability [46].

### **2.2.2.3.1. Image Processing Techniques**

In the context of ML, image processing includes the usage of techniques or algorithms to modify existing data samples, to produce new data samples. Techniques such as resizing, and data augmentation, and algorithms such as Otsu binarisation and Gaussian blur are applied with the goal of improving the generalisation capabilities of the model.

Thereby, the Otsu binarisation is a technique that separates foreground from background objects, by applying a threshold to minimise intra-class variance. This algorithm analyses the histogram of an image, segmenting it into background and foreground, iteratively searching for the grey level that maximises class separation [47], [48]. Gaussian blur is another filtering technique that reduces image noise and detail, smoothing variations by averaging pixel values based on a Gaussian function. This operation is useful for preprocessing tasks, such as edge detection and feature extraction [7], [49]. When used in conjunction, Otsu binarisation and Gaussian blur enhance segmentation results by reducing noise, resulting in better object separation.

Using techniques including Otsu binarisation and Gaussian blur, Han et. al [7] developed an aerial thermal image calibration and processing method to extract the canopy temperature for analysing WS in fruit trees, under different irrigation conditions. With these techniques, canopy temperatures from aerial thermal images were effectively isolated from background data, such as soil.

### **2.2.2.4. Network Configuration Hyperparameters**

#### **2.2.2.4.1. Loss Function**

A loss function is responsible for evaluating the model's performance, so that it can be used, during training, to adjust the model's weights. This function applies a penalty to the model, when the output predictions deviate from the expected output, that is, the ground truths. Ultimately, the goal is to minimise this function, so that the model can generate better, and more accurate predictions [50].

There are several loss functions, and their application depends on the type of data being used, which can be either continuous<sup>13</sup> or discrete<sup>14</sup>. When the output predicts a continuous, real number, this is a regression problem. In contrast, when the target output is a discrete value, the task is a classification problem [50], [51].

In a binary classification problem, Binary Cross Entropy (BCE), also known as Logarithmic Loss (LL), adjusts the models' weights by penalising the model when it makes incorrect class predictions to improve the models' predictive capabilities [50], [51].

$$BCE = -\frac{1}{N} \sum_{i=1}^N y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i)) \quad (3.2)$$

Equation (3.2) represents the equation to obtain the BCE, where  $N$  is the number of samples considered. Each sample has a true class label, denoted as  $y_i$ , where  $y_i = 1$  for positive class and  $y_i = 0$  for the negative class. The function  $p(y_i)$  is the predicted probability that the model assigns to the positive class ( $y_i = 1$ ).

The equation is divided into two distinct terms:

- $y_i \cdot \log(p(y_i))$  applies when  $y_i = 1$ , focusing on the log of the predicted probability for the positive class;
- $(1 - y_i) \cdot \log(1 - p(y_i))$  applies when  $y_i = 0$ , measuring the error for the negative class based on  $1 - p(y_i)$ , which represents the probability that the sample belongs to the negative class.

The negative sign that precedes the formula ensures that larger errors produce higher positive values, hence, it penalises the model for making incorrect predictions. The summation and multiplication of the inverse of  $N$  averages the loss across all samples, providing a single value that reflects the overall model performance. This loss guides the model's weight adjustments during training, optimising its accuracy by reducing prediction errors [50], [51].

#### 2.2.2.4.2. Dataset Split – Training and Validation

Despite the importance of the loss function penalising incorrect predictions during training, it is fundamental that the model produces accurate predictions on both the training set, and, more importantly, on the new data, unseen during training. Generalisation refers to the model's ability

---

<sup>13</sup> A continuous variable can only take a single value from an uncountable set of values.

<sup>14</sup> A discrete variable can only take a single value from a limited number of possible values.

to perform well on data used during training and new data. To assess the model's ability to generalise, it is necessary to evaluate its performance on unseen data [50], [51]. To achieve this, ML techniques such as holdout validation and k-fold cross-validation are used during training to split the original dataset [52], [53], [54].

Thereby, holdout validation involves dividing the dataset into two distinct sets, a training set and a validation set. The training set is used to train the model, while the remaining data is reserved for validating the model's performance on unseen data. In addition, this method, compared to others, is easy to implement and computationally inexpensive. However, a major drawback is the potential for bias, in case the validation set contains patterns not captured in the training set [52].

In contrast, k-fold cross-validation addresses some limitations of the previous method, by dividing the dataset into  $k$  subsets or folds. The model trains  $k - 1$  times, while one-fold is reserved for validating, in a process that repeats until each fold has served as a validation set once. This method helps reduce overfitting while providing a more accurate model performance. It is, however, computationally expensive and time-consuming, compared to the holdout validation [53].

#### **2.2.2.4.3. Evaluating Loss Function on Unseen Data**

Analysing the loss function during training on unseen data ensures that the model generalises beyond the training data. By examining the model's performance on separate data sets, it is possible to observe how the model adapts to new data points. This confirms whether the model is learning meaningful patterns, rather than memorising the training data [55].

Overfitting happens when a model becomes too complex, fitting the training data very closely but failing to generalise well on new, unseen data. During training, this phenomenon becomes apparent as the training loss value reduces, while the value of the validation loss starts to increase. To avoid this undesirable behaviour, training is interrupted when the validation loss starts to increase, thus, preventing the model from fitting the training data perfectly [54], [55].

#### **2.2.2.4.4. Additional Performance Metrics**

Depending on the type of ML problem, either a regression problem or a classification problem, there are numerous available metrics to evaluate the model's performance.

In a binary classification problem, the metrics that are presented in this section are calculated using a single threshold, which can be tuned to achieve a desired model performance. The threshold determines the classification of an output, that is, if the prediction exceeds this output, it indicates that the class belongs to the positive or target class. Conversely, if the prediction falls below the threshold, it signifies that the class belongs to the negative class [56].

In the context of a binary classifier, each predicted output results in one of four possible outcomes. The classifier either correctly identifies the positive class, known as a True Positive ( $TP$ ), or it correctly identifies the negative class, known as a True Negative ( $TN$ ). Alternatively, when the classifier incorrectly identifies the positive class, this is a False Negative ( $FN$ ), or when it incorrectly identifies a negative class, is called a False Positive ( $FP$ ) [56], [57], [58]. After these values are calculated, they can be organised to form a confusion matrix. This matrix maps the model's predictions against the actual classes, providing a visual representation of the classifier's performance [59].

### **Accuracy**

The Accuracy metric measures the portion of all classifications that are correctly made and is defined by the equation (3.3):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.3)$$

In a well-balanced dataset, accuracy can measure the performance of the model. However, when there is a reduced number of target samples, meaning that the dataset is imbalanced, it results in a high accuracy, or, if it generates incorrect predictions, namely  $FP$  and  $FN$ , these will be indistinguishable, creating a need to use other metrics [56], [57].

### **Recall**

The recall or true positive rate measures the proportion of actual positive cases that are properly predicted by a model. Recall is defined by the equation (3.4):

$$Recall = \frac{TP}{TP + FN} \quad (3.4)$$

A high recall ensures that a model detects most of the actual positive cases, that is, the  $TP$ , as there will be a reduced number of  $FN$ . Conversely, if there is a reduced number of  $TP$ , caused by an unbalanced dataset, the recall will be low. Recall is helpful when minimising  $FN$  is more critical than reducing  $FP$  [56], [57].

## Precision

Precision refers to the fraction of the model's positive predictions that are truly positive. It is mathematically expressed by the equation (3.5):

$$Precision = \frac{TP}{TP + FP} \quad (3.5)$$

When a model displays a high Precision, it confirms that the model finds the actual positive cases,  $TP$  out of all the positive predictions, as it implies that there is a reduced number of incorrect positive predictions,  $FP$ . In an unbalanced dataset that contains few  $TP$  cases, Precision performs poorly, similarly to Recall. Nevertheless, Precision is useful when reducing  $FP$  cases have a higher priority over  $FN$  cases [56], [57].

Changing the classification threshold reveals that Precision and Recall have an inverse relationship. This behaviour is caused as higher threshold values increase the amount of  $FN$ s and decrease the occurrences of  $FP$ s. The opposite happens with lower threshold values [56].

## Intersection over Union

Intersection over Union (IoU), also known as Jaccard index, is a metric used in classification tasks, including object detection and image segmentation, that evaluates the resemblance between two shapes. It calculates a normalised value, based on the dimensions and positions of the shapes, with values closer to 1 indicating a higher degree of overlapping [58], [60]. IoU is defined by the equation (3.6):

$$IoU = \frac{TP}{TP + FN + FP} \quad (3.6)$$

### 2.2.2.4.5. Optimisers

The choice of optimiser directly impacts the convergence speed and the predictive performance of a fully trained algorithm. An optimiser defines an update rule that directs how the model parameters are adjusted based on the loss function and gradient information. The behaviour of the optimiser is influenced by a set of hyperparameters, such as the learning rate. Selecting an optimal optimiser involves conducting both empirical studies and benchmarking. The latter involves fine-tuning the learning rate, a hyperparameter on a validation dataset, to maximise validation dataset performance [61].

The Stochastic Gradient Descent (SGD), initially proposed by Robbins et al. [62], updates the model parameters by moving them in the opposite direction of the gradient of the loss function with respect to the parameters. The step size of this movement is controlled by a learning rate.

Gradient descent with momentum is built upon SGD and it is designed to accelerate convergence. It incorporates part of the previous parameter update into the current update, smoothing out the optimisation trajectory by taking larger steps. The momentum coefficient determines how much of the previous update to carry over. Using momentum leads to faster convergence and avoids local minimum, where the gradient value is 0. However, despite oscillating around the global minimum until reaching it, it is still faster to conventional gradient descent [63], [64], [65].

Nesterov Accelerated Gradient (NAG) improves on momentum as the update happens in two steps. Initially, the gradient at the look-ahead point is calculated and it is then used to calculate the final update. If the gradient at the look-ahead point is positive, the NGA works similarly to a momentum-based optimiser, however, if it is negative, the final update will be smaller. This additional measure results in NGA converging faster than momentum, as the chances of overshooting the global minimum are reduced, and it allows greater values of its momentum [61], [66], [67].

The Adaptive Moment Estimation (ADAM) optimiser, proposed by Kingma et al. [68], adjusts the learning rate for each parameter based on estimates of the first and second moments of the gradients. ADAM combines the advantages of Adaptive Gradient Algorithm and Root Mean Square Propagation. Both methods adapt learning rates for each parameter individually, but the second further incorporates a running average of squared gradients, which helps smooth updates. However, in ADAM, updates are estimated using a running average of the first and second moment of the gradient. In addition, a bias-correction term is introduced to avoid large step sizes and divergence, in the presence of sparse gradients. These combinations and modifications make ADAM particularly effective for tasks involving sparse gradients, when only a few features appear frequently in a dataset, as well as for on-line learning or when dealing with non-stationary data, such as noisy environments. Moreover, ADAM requires fewer computational resources while providing stable convergence compared to other methods [61], [68].

Choi et al. [61] compares several optimisers, using a set of hyperparameters and demonstrates that more general optimisers, such as ADAM, generally perform adequately compared to SGD, Nesterov or Momentum. Moreover, the researchers outline that, if possible, all hyperparameters should be tuned of the adaptive gradient methods, as they were unable to approximate the compared optimisers.

#### **2.2.2.4.6. Batch Size**

Batch size is a hyperparameter that characterises the number of samples used in one forward and backward pass through a CNN. It directly impacts model performance, and the computational resources required for training [69].

Radiuk et. al. [53] uses two datasets, Mixed National Institute of Standards and Technology (MNIST) and the CIFAR-10, along with two custom CNNs with batch sizes ranging from 16 to 1024 to evaluate the impact of using different batch sizes during training. They prove that smaller batch sizes can accelerate training but lead to lower accuracy and increased sensitivity to data fluctuations. Conversely, larger batch sizes improve accuracy and stability, offering greater resistance to data variation, to the detriment of longer training times, higher computational resources and slower convergence speed [69]. Therefore, depending on the computational resources, a higher batch size will result in a more accurate model.

#### **2.2.2.4.7. Number of Epochs**

In ML, an epoch refers to one complete pass through the entire training dataset by the learning algorithm. The number of epochs determines how many times the model will process all the samples in the dataset during training. A low number of epochs may cause the training to stop prematurely, resulting in underfitting due to insufficient knowledge about the data. However, too many epochs can lead to overfitting, where the model is unable to generalise new and unseen data [70].

#### **2.2.2.4.8. Learning Rate**

The learning rate is a hyperparameter in ML that, during the model training, determines the magnitude of the step size used to update the weight and bias of each layer to minimise the loss function. The learning rate ranges from 0.0 to 1.0, where a high learning rate can cause the model to overshoot the global minimum and fail to converge. In contrast, a low learning rate increases the number of epochs necessary to train the model and it could cause the model to reach a local minimum [38], [71].

### 2.2.2.4.9. Activation Functions

An activation function introduces the non-linearity that enables a neural network to model complex data. Otherwise, without activation functions, these networks would function as linear models, unable to learn from non-linear data [72].

During backpropagation, a gradient descent algorithm is responsible for updating the weights of each layer, with the goal of minimising the loss function, thus, finding the optimal weights. The equation (3.7) updates the weights using gradient descent until convergence, where  $W_{old}$  is the old weight,  $W_{new}$  is the new weight,  $\eta$  is the learning rate and  $\frac{\partial L}{\partial W_{old}}$  is the gradient of the loss function  $L$  with respect to the old weight [73].

$$W_{new} = W_{old} - \eta \frac{\partial L}{\partial W_{old}} \quad (3.7)$$

The Sigmoid activation function is a non-linear function that maps input values, ranging from  $(-\infty; \infty)$  to output values between 0 and 1 [72]. The Sigmoid function, also known as Logistic Activation Function, is defined by the equation (3.8):

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (3.8)$$

Since the Sigmoid function outputs a value between 0 and 1, and its derivative is comprehended between 0 and 0.25, in a deep network during backpropagation, the reduced derivative range can lead to small gradients, causing the vanishing gradient problem [72].

The Tanh function is similar to the Sigmoid activation function, however, it outputs a zero-centred value, ranging from -1 to 1 and its derivative ranges from 0 to 1. Despite having a steeper gradient, it still affected by the vanishing gradient [72]. This function is given by the equation (3.9):

$$\tanh(x) = \frac{2}{1 + e^{-2x}} - 1 \quad (3.9)$$

Lastly, the Rectified Linear Unit (ReLU) activation function overcomes the drawbacks associated with the previous functions, that is, the vanishing gradient problem, and it also results in superior training performance. This function returns the input directly if positive and zero otherwise [72]. The ReLU function is provided by the equation (3.10):

$$ReLU(x) = \max(0, x) \quad (3.10)$$

Despite its advantages, ReLU is prone to the “dying ReLU” problem, where neurons stop learning entirely if their output consistently falls below zero [72].

In the context of binary classification problems, when using an activation function for the output layer, it is conventional to use a Sigmoid activation function, as labels are either 0 or 1.

#### **2.2.2.5. Data Augmentation**

Data augmentation is a technique that increases the number of existing data samples present in a dataset, by copying and modifying existing data samples, using image processing techniques. These modifications include colour-space and geometric changes, such as, applying horizontal or vertical flip to the samples. This step improves the model's ability to better generalise by exposing it to a broader range of in-image variations [74].

# Trained Models

This chapter outlines the methodologies and tools used in developing and fine-tuning the U-Net-based models to, in a first phase, perform image segmentation on an existing UAV-captured agricultural dataset, and in a second phase, to use transfer learning to retrain part of the previous model to detect and localise WS in a vineyard, using UAV-captured thermal images.

It begins with a methodology overview that summarises the approach taken to develop the platform for WS detection and localisation. This is followed by a description of the software and the libraries used, including Python, Tensorflow, Keras, Django, Thermal Parser, among others. Next, the architecture and its implementation are described, discussing the reasons behind some of the made decisions. It also covers the hyperparameters and metrics used during fine-tuning. The dataset split into training and validation sets is described, with an emphasis on the choice behind holdout validation. Furthermore, the chapter details each specific model, namely WeedUNet and StressUNet, along with their dataset creation process. Finally, the developed platform is described, highlighting its architecture and key functionalities.

### 3.1. Methodology Overview

To develop the toolkit to visually pinpoint areas experiencing water stress in vineyards, that could be scaled to a production-ready system, a U-Net based architecture was chosen. This decision was motivated by the lack of a freely available pre-trained network for a similar PA application, which also eliminates the need for any licensing requirements and restrictions. In addition, this gives a greater flexibility by allowing the network to receive an input image with additional channels. Combined with the availability of the Agriculture-Vision large scale dataset of UAV-captured agricultural images (further defined in later sections), this resulted into two distinct models being created, WeedUNet and StressUNet. As both models perform binary segmentation, the models share the same architecture.

#### 3.1.1. Software Used

The software developed for implementing and training these models relies on a combination of several different libraries and frameworks, however, only key libraries are outlined. The programming language is Python as it provides the necessary support for ML applications.

These frameworks include TensorFlow and Keras, which provide the necessary aspects to build and deploy ML models. Tensorflow provides a comprehensive platform to create end-to-end ML models by using Keras, allowing an easier model prototyping and experimentation [75], [76].

To optimise computations, CUDA and cuDNN libraries were used, and are both developed by NVIDIA. With this, CUDA allows for parallel processing on the GPU hardware, thus, speeding up training times, while cuDNN is a GPU-accelerated library of primitives for DNNs, providing tuned implementations of standard routines, like convolution and pooling [77]. These tools were crucial for developing both WeedUNet and StressUNet models, enabling efficient training and fine-tuning of the network architectures.

Herewith, the platform where the StressUNet model makes predictions is developed using the Django web framework, as it uses Python, and it already contains the necessary components to create a marketable solution [78], while the Thermal Parser Python library extracts the temperature value of the UAV captured thermal images, which enables the creation of the CWSI-based ground truth masks [79]. This library was used as the UAV manufacturer does not provide a solution with these functionalities and ease of use.

## **3.2. U-Net model**

### **3.2.1. Architecture**

The U-Net model architecture follows the specifications outlined by Ronneberger et al. [46] and it was based on an existing U-Net Python implementation [80]. The architecture consists of a contracting path and an expansive path. The contracting path consists of using multiple downsampling blocks that double the number of feature channels. One downsampling block consists of two 3x3 convolutions, each with a ReLU activation function followed by a 2x2 max pooling operation and a 0.3 dropout layer. Similarly, the expansive path consists of several upsampling blocks, where the feature map is upsampled, and a 2x2 convolution reduces the number of feature channels to half. This is followed by a concatenation with the corresponding cropped feature map from the contracting path and two 3x3 convolutions, each with a ReLU activation function.

The implementation utilises the Keras functional Application Programming Interface (API), as it allows the creation of models with non-linear topology and shared layers, which offers greater flexibility compared with Keras sequential API [81].

The architecture processes input images with the shape of 128x128x3 (height, width and depth, respectively). This was motivated by available images from the Agriculture-Vision dataset dimensions being 512x512x3, which, during training, caused out-of-memory exceptions to occur. In addition, given the limited number of original thermal images with the dimensions of 640x512, data augmentation posed serious challenges. Resizing the images could result in information loss, while transformations involving colour or shape are unrealistic in a vineyard context. Therefore, using images with the shape of 128x128x3 allows the architecture to work with the images from both datasets, without compromising the information contained within each image.

The architecture outputs a single channel image with a height and width of 128 pixels. Each pixel contains a value in the range of 0 to 1, as result of the sigmoid activation function. In doing this, values closer to 1 indicate that the model predicts that pixel as a foreground object (positive class), whereas values closer to 0 indicate a background object (negative class).

### 3.2.2. Network Hyperparameters Fine-tuning

To optimise performance of each model, several models were trained using different sets of hyperparameters, including batch sizes, optimisers and learning rates across 500 epochs. Table 3.1 summarises the hyperparameters and the possible values tested during the model optimisation. These hyperparameters were based on the hyperparameters found in the literature [61], [68], [69], [71].

TABLE 3.1. Hyperparameter Configurations

<b>Hyperparameters</b>	<b>Values</b>
<b>Optimiser</b>	ADAM, SGD, SGD with momentum of 0.9 and Nesterov acceleration enabled
<b>Batch size</b>	8, 16
<b>Learning rate</b>	0.01, 0.001, 0.0001
<b>Epochs</b>	500

Different optimisers were explored to find the most effective method to minimise the loss function. The optimisers tested included ADAM, SGD and SGD with a momentum of 0.9 and Nesterov acceleration enabled. Similarly, to find the optimal batch size, models were trained with batches of 8 and 16 samples, to verify if a higher batch size would improve accuracy and

stability. Models were also trained with learning rates of 0.01, 0.001 and 0.0001, where lower values could result in a more precise conversion at the cost of training times.

In binary image segmentation tasks, the BCE loss function penalises incorrect class predictions, ensuring accurate pixel-level labelling, as either foreground (positive class) or background (negative class). Furthermore, training incorporates metrics like Precision, Recall, IoU, and Accuracy. Each provides unique insights into different performance aspects of the model:

- **Accuracy:** Measures the portion of all correct classifications;
- **Recall:** Measures the portion of actual positive cases correctly predicted;
- **Precision:** Measures the amount of actual positive predictions out of all positive predictions;
- **IoU:** Evaluates the overlap between the predicted masks and the ground-truth Masks.

### **3.3. Dataset Training - Validation Split**

In this study, given the balance between efficiency and model accuracy, holdout validation using a 50/50 split is used for every trained model [80]. This choice ensures that each model generalises well without the computational overhead associated with multiple training iterations in k-fold cross-validation. Although holdout validation may lead to less precise estimates, it proves advantageous when time and resources are constrained, enabling additional models and different hyperparameters to be tested and validated. In addition, using validation data during training allows the model to evaluate itself at the end of each epoch. Monitoring validation metrics during training ensures that the model generalises with unseen data. Specifically, by monitoring validation loss, it is possible to verify when overfitting occurs, as validation loss start to increase. Ultimately, this monitoring helps to mitigate overfitting.

### **3.4. WeedUNet Model**

The initial model, named WeedUNet, was trained from scratch to detect weed clusters in RGB aerial images of agricultural farmland. These images were sourced from the Agriculture-Vision dataset [82]. This model was trained and optimised to accurately distinguish weed clusters from non-weed areas, based on pixel-wise classification. This pattern was chosen after determining that it was the most abundant on the dataset. However, due to missing near infrared-**RGB** image pairs, only **RGB** images were considered.

### 3.4.1. Dataset Creation

The Agriculture-Vision dataset contains over twenty thousand UAV-captured RGB and near infrared farmland images. These images were captured throughout 2019, from varying farmland regions, each with a width and height of 512 pixels, an annotation label mask, a corresponding boundary mask and boundary map. The annotation label mask indicates the pixels of the image where a particular agricultural pattern is present. The boundary mask highlights the valid pixels in the image while the boundary map shows the valid farmland region [82].

Therefore, constructing the dataset to train WeedUNet involves obtaining a valid mask that indicates the presence of the selected agricultural pattern within the boundary mask and map.

To minimise class-wise bias<sup>15</sup>, only images containing between 30% and 70% of pixels labelled as “weed cluster” are selected. This repeats until the threshold of 6000 images is reached. If the available images prove insufficient, the previously added images are duplicated and undergo a data augmentation step that applies a random rotation.

## 3.5. StressUNet Model

Building on the knowledge contained within WeedUNet, a second model, StressUNet, was developed using transfer learning to detect WS in vineyards. In this approach, StressUNet was initialised with the weights from the weed detection model but adapted to perform binary image segmentation on aerial thermal images, identifying areas in a vineyard experiencing WS. Using transfer learning, the best performing WeedUNet and unfreezing the last 5 layers enabled the StressUNet to achieve accurate results, with only 40 original images.

### 3.5.1. Dataset Creation

The thermal dataset acquisition involved using a thermally equipped UAV, flying at a height of 60 meters at around 11 am [8]. This methodology and the equipment are further described in chapter 4.

The captured thermal images are first processed using the Thermal Parser library, to extract the temperature within each pixel [79]. To create the ground truth masks, image processing

---

<sup>15</sup> In a binary classification problem, this class wise bias arises when the number of samples of one class are much greater than the other (for example, 90% of the pixels being labelled as background and 10% foreground).

involving Otsu binarisation and Gaussian blur were used to separate the vines from the background [7]. In addition, valid vineyard regions were manually labelled using Label Studio software [83], creating masks that indicated the areas of interest, ensuring that background and noise in the data were discarded. A sliding window approach was employed to augment the dataset. The images were split into smaller segments of 128x128 pixels with a 32-pixel stride, ensuring overlap and thorough coverage of the thermal data. This method enables the creation of several unique images from a single image, while retaining the details in the original image, as resizing operations are avoided. The resulting images are further filtered by only selecting the ones containing between 20% and 80% of foreground objects. This threshold helps balance the dataset by excluding images with excessive or minimal foreground coverage.

To further address class-wise bias, sample weights were calculated using the Scikit-learn library [84]. This approach ensures that the model receives the necessary weights to use during training, so that it pays additional attention to the underrepresented class. This helps to mitigate the impact of the greater number of background pixels, compared to the number of foreground pixels. Ultimately, this process resulted in a dataset comprising 1500 unique images, which were used to train the StressUNet model.

## **3.6. Water Stress Monitoring Platform**

### **3.6.1. Django Overview**

Django is A Python web development framework that includes all the necessary libraries, functionalities and logic to develop and deploy modular scalable applications [78].

### **3.6.2. Model - View - Template Architecture**

Django follows a 3 layers architecture, the Model-View-Template, where each part serves a distinct role. The model serves as the data access layer that handles the database. It defines the data structure and provides methods to interact with it. The view processes user requests and returns responses. This is done by interacting with the model to retrieve data and rendering templates. Finally, the template is the presentation layer that handles the user interface, defining how data is displayed to the user [85].

### **3.6.3. Platform Architecture and Key Functionalities**

The platform is designed to support viticulturists manage one or more vineyards, across various seasons. To achieve, the platform is organised into a modular structure using Django

applications. Each application encapsulates specific functionalities, facilitating easy maintenance and scalability. The core hierarchy is defined as follows:

- **Project:** Represents the management entity, that is, a vineyard over an indefinite period;
- **Campaign:** A Campaign manages specific periods or objectives, such as the monitoring of a particular growth cycle. A Project may contain several Campaigns;
- **Mission:** These correspond to individual data collection efforts, involving UAV flights. These can happen several times during the growing season of a vineyard, encapsulating information of key growth stages;
- **Picture:** The UAV-captured thermal images from each mission are uploaded to the platform. Each image is then transformed into a Picture, which displays both the original thermal image and its corresponding prediction side by side, allowing users to visually assess and compare the data.
- **User:** Represents a user of the platform and encapsulates user related tasks, such as authentication.

Figure 3.1 illustrates the workflow of the platform, detailing its internal workings. This involves the creation and access of the above defined Django applications, and the interactions between the platform, the predictive model (WeedUNet) and the UAV captured thermal images.

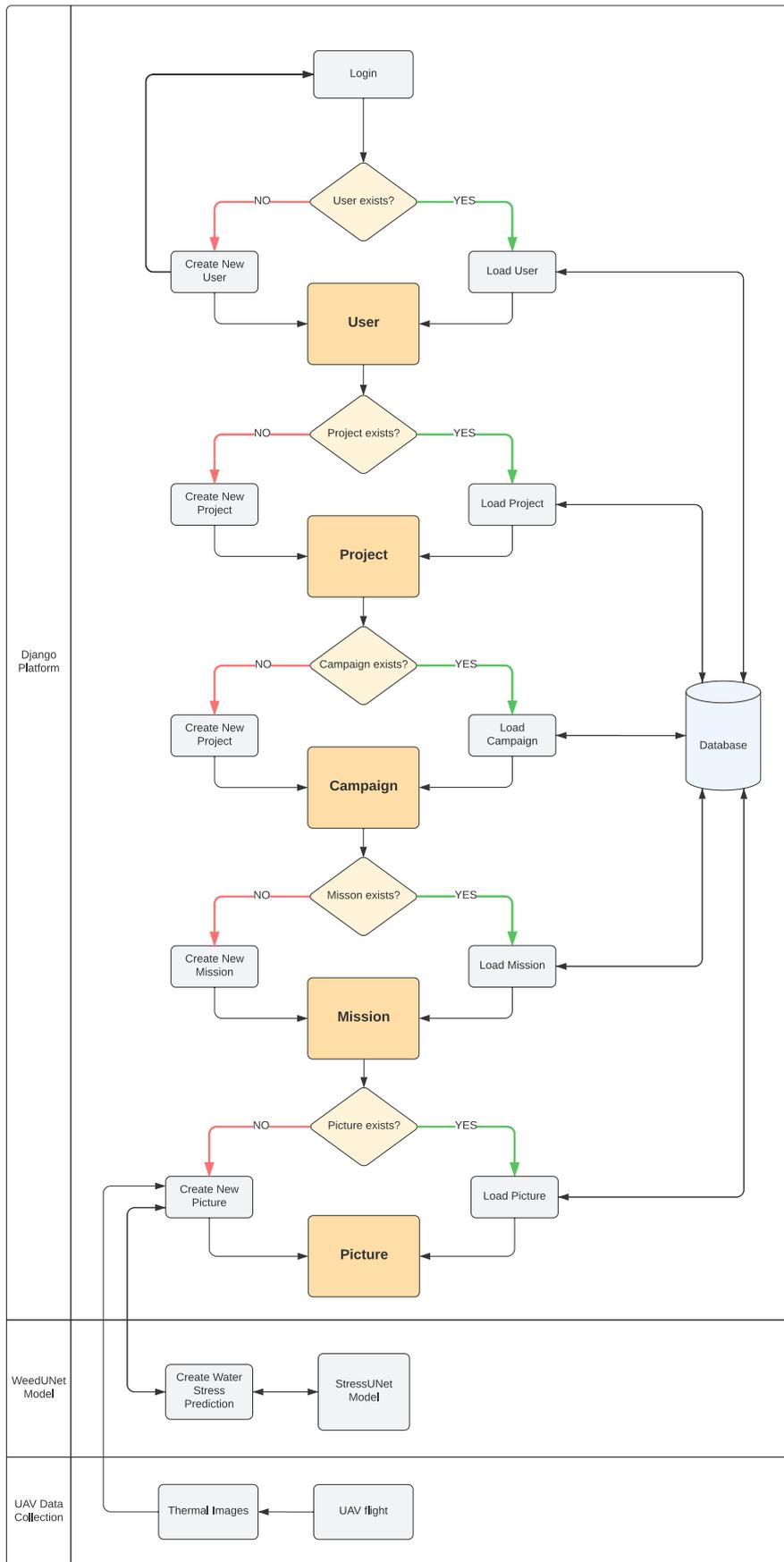


FIGURE 3.1. Toolkit Fluxogram

## Experimental Validation and Discussion

This chapter starts with a detailed description of the materials and study methods involved in the creation of StressUNet. Afterwards, the performance of WeedUNet and StressUNet is discussed.

### 4.1. Materials and Study Methods

#### 4.1.1. UAV and Onboard Sensors

The UAV chosen for this study is a DJI Mavic 3 thermal (Figure 4.1), from Da-Jiang Innovations. This UAV is a multi-rotor design, specifically a quadcopter, having VTOL capabilities and precise manoeuvrability, which is necessary for accurate data collection. Moreover, the UAV can be easily transported as it features foldable rotors and can fly for up to 45 minutes, having the ability to swap batteries and resume the last data collection point [86].



FIGURE 4.1. DJI Mavic 3T Mid-flight

This specific UAV has a maximum take-off weight of 1050 grams, and it features two sensors, a 48 Megapixels RGB wide main camera and an uncooled thermal camera, sensitive to the infrared wavelength range of 8  $\mu\text{m}$  to 14  $\mu\text{m}$ , with a temperature measurement accuracy of  $\pm 2^\circ\text{C}$  or  $\pm 2\%$  (using the greater value), in the range of  $-20^\circ\text{C}$  to  $150^\circ\text{C}$ . The sensors are attached to a 3-axis (roll, tilt and pan) gimbal, which greatly reduces any distortions that may occur. Table 4.1 and Table 4.2 fully summarise the information related to each main sensor equipped on the UAV [86].

TABLE 4.1. DJI Mavic 3T RGB Camera Specifications (Source:[86])

<b>Sensor</b>	<b>RGB</b>
<b>Sensor type</b>	1/2-inch CMOS
<b>Effective pixels</b>	48MP
<b>Field of view</b>	84°
<b>Format equivalent</b>	24mm
<b>Aperture</b>	f/2.8
<b>ISO Range</b>	100-25600
<b>Shutter type and speed</b>	Electronic Shutter: 8-1/8000 s
<b>Max Image Size</b>	8000×6000

TABLE 4.2. DJI Mavic 3T Thermal Camera Specifications (Source: [86])

<b>Sensor</b>	<b>Thermal</b>
<b>Thermal Imager</b>	Uncooled VOx Microbolometer
<b>Pixel Pitch</b>	12 $\mu\text{m}$
<b>Frame Rate</b>	30 Hz
<b>Lens</b>	Display Field of View: 61° Format Equivalent: 40 mm Aperture: f/1.0 Focus: 5 m to $\infty$
<b>Noise Equivalent Temperature Difference (NETD)</b>	$\leq 50 \text{ mK}@F1.0$
<b>Temperature Measurement Method</b>	-20° to 150° C (-4° to 302° F, High Gain Mode)
<b>Photo Format</b>	JPEG (8-bit) R-JPEG (16-bit)
<b>Max Image Size</b>	640×512
<b>Infrared Wavelength</b>	8-14 $\mu\text{m}$
<b>Infrared Temperature Measurement Accuracy</b>	$\pm 2^\circ \text{ C}$ or $\pm 2\%$ (using the larger value)

Additionally, the DJI Mavic 3 thermal is labelled as C-2 class label by EASA, which permits the UAV to be operated near involved and uninvolved personal. Acquiring a license to operate a UAV subcategory A2 ensures that operations follow the law, and the requirements attributed to the UAV's open category. This license can be obtained through online courses, such as EUROCONTROL Learning Zone, which provides all the necessary information to conduct safe and lawful UAV operations [87].

#### 4.1.2. Network Training Equipment

The network training process utilises an Asus TUF A15 FA507NU, which features an AMD Ryzen 7 7735HS CPU, an NVIDIA GeForce RTX 4050 GPU and 32 GB of RAM.

### 4.2. Study Site

The study site is a vineyard located in Cabanas de Torres, Alenquer, Lisbon, Portugal, at coordinates 39°08'34.5"N 9°04'11.1"W (Figure 4.2). This vineyard lacks artificial irrigation and covers an area of approximately 3600 square meters, offering a representative landscape for viticultural research. The region's climate, typical of the Lisbon district, is influenced by both Atlantic and Mediterranean conditions.

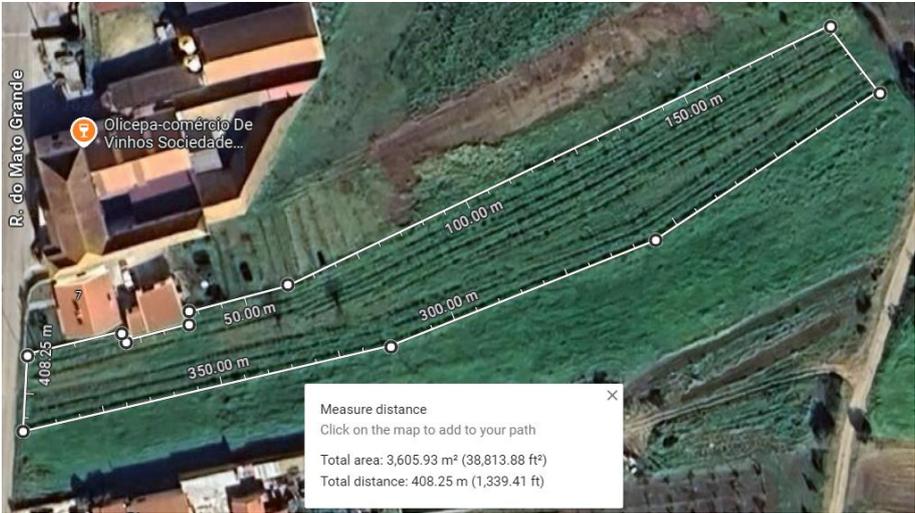


FIGURE 4.2. Aerial Picture of the Vineyard

Data collection was performed at 11 am on the 15<sup>th</sup> of September 2024, under clear skies. The UAV thermal data collection process follows the methodologies outlined by Awais et al. [8]. The UAV acclimated in the shade, 30 minutes before take-off. During this period, mission planning was created using the UAV's integrated mission planning software. The forward and side overlap were set to 80%, the height of 60 meters and the speed to 1m/s. This resulted in a ground sampling distance of 7cm<sup>2</sup>/pixel.

Just before take-off, the CWSI parameters, specifically  $T_{wet}$  and  $T_{dry}$ , were determined by taking temperature measurements of intact leaves at ten different locations, five shaded and five sunlit, using a handheld infrared thermometer (Uni-T UT300A+). After the flight was concluded, these measurements were repeated. Every measurement related to the CWSI parameters followed the same process, as outlined by Sepúlveda-Reyes et al. [15]:

- $T_{wet}$  was measured by applying a water-soap solution to both sides of the leaf, waiting a few minutes, and then recording the temperature;
- $T_{dry}$  was measured by covering both sides of the leaf with petroleum jelly, and then taking the temperature measurement after a few minutes.

Figure 4.3 shows one of the UAV-captured thermal images and Annex B contains additional images captured in the study site.

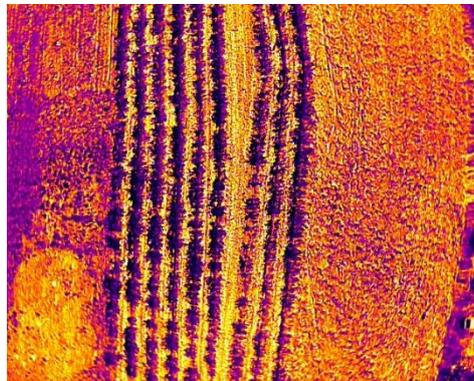


FIGURE 4.3. UAV Captured Thermal Image of the Vineyard

### 4.3. Best Performing Models

This section outlines the results and parameters of the best performing WeedUNet and Stress UNet models, which underwent an optimisation process, involving training several models with different sets of parameters. These parameters include the optimiser, learning rate, batch size, evaluation metrics and relevant functions that resulted in an optimised model. In addition, the evolution of training and validation metrics, as well as confusion matrices results are displayed and analysed.

To visually validate the performance of each model, the metrics used during training are plotted and saved to a file, along with a confusion matrix. For each metric, the values calculated for both training and validation data are plotted over the epochs, with blue representing the training values, and orange representing the validation values. Additionally, a legend outlines the lowest and highest values achieved for both training and validation data, along with the metric's values on each dataset when the model was restored.

To visually validate the performance of each model, the metrics used during its training are output to a file, along with an aggregate confusion matrix. For each metric, the training and validation values are plotted for each epoch, where blue represents the training values and orange the validation values. In addition, a legend outlines the lowest and highest achieved values for both training and validation data, as well as an indication of the metric's values on each data set, when the model was restored. On the one hand, the restored model is related with a mechanism from TensorFlow to prevent overfitting and reduce training times by prematurely stopping the model training. This occurs if the validation loss does not reduce for 10% of the total number of epochs. If this mechanism is engaged, the model restores itself to the epoch that had the lowest validation loss, before saving it to a file. On the other hand, the aggregate confusion matrix involves a batch-wise confusion matrix calculation, where predictions and true labels are organised into batches. For each batch, a confusion matrix is computed, capturing the counts of  $TP$ ,  $TN$ ,  $FP$  and  $FN$ . These individual matrices are summed to create an aggregate confusion matrix that represents a normalised view of model's performance over all the validation batches.

### 4.3.1. WeedUNet

#### 4.3.1.1. Hyperparameters

The best performing WeedUNet model was trained using the outlined in Table 4.3.

TABLE 4.3. Optimal WeedUNet Hyperparameters

<b>Hyperparameters</b>	<b>Values</b>
<b>Optimiser</b>	ADAM
<b>Batch size</b>	8
<b>Learning rate</b>	0.0001
<b>Epochs</b>	500

### 4.3.1.2. Training and Validation Metrics

Figure 4.4 shows consistent improvement in all metrics over the epochs. The training and validation metrics are closely aligned, indicating good generalisations. Additionally, it is possible to see that the model achieved the lowest validation loss of 0.14 at epoch 174 before stagnating and starting to overfit, which made the model training to be stopped early and restored to that epoch.

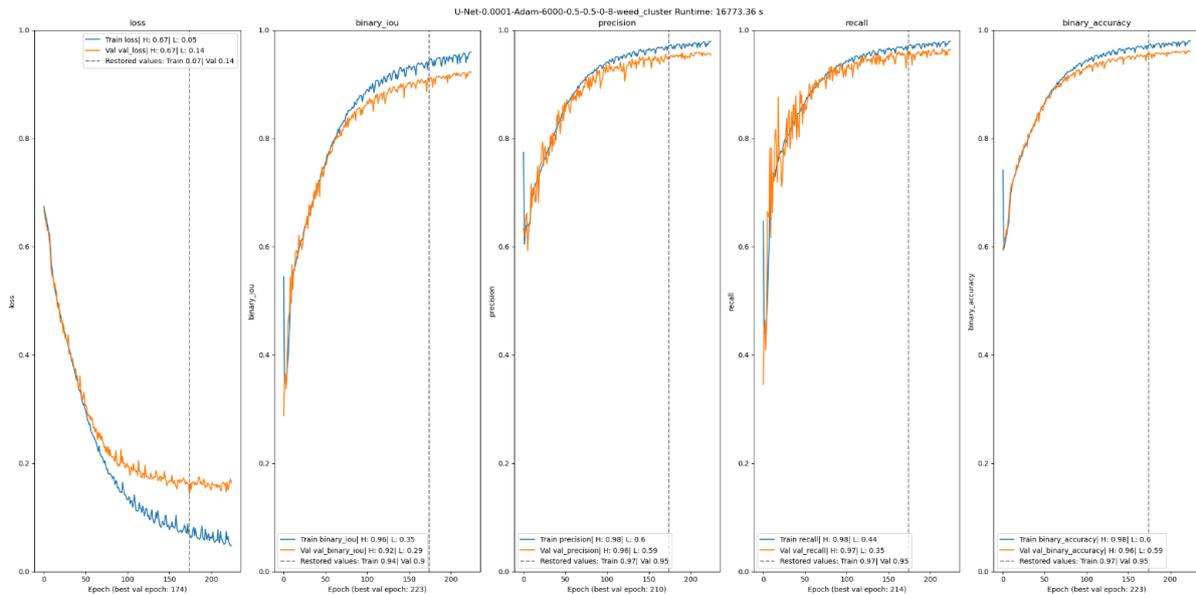


FIGURE 4.4. Plot of the Performance Metrics of the Best WeedUNet Model

In addition to these plots, Figure 4.5 represents the average confusion matrix computed on the validation dataset after the model is trained, as described previously, which helps to validate the predictive capabilities of the model. The number of samples,  $n$ , is calculated by multiplying the number of complete batches available in the validation dataset, 375, by the width, 128, and height, 128, dimensions of each image. Each quadrant contains a normalised value based on the relationship between the value of each pixel from every image in the validation dataset and the corresponding pixel value of the ground truth mask.

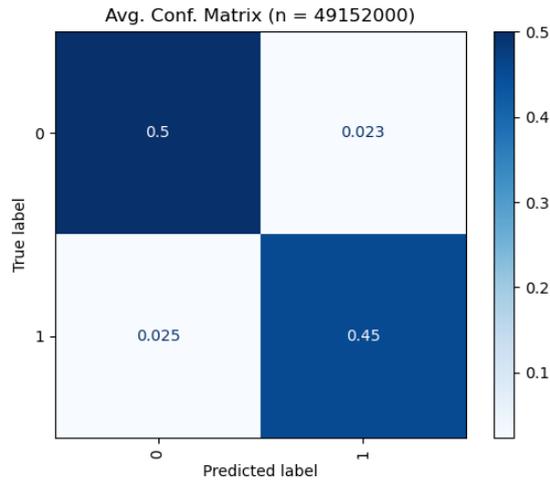


FIGURE 4.5. Average Confusion Matrix for the Best WeedUNet Model

The first and fourth quadrants represent the percentage of pixels that were correctly labelled as background and foreground, respectively. This amounts to 95% of the total number of samples. In contrast, the remaining 5% belong to the second and third quadrant, which denote the proportion of pixels that were incorrectly labelled as foreground and background, respectively. These values highlight the predictive capabilities of this model as 95% of the predicted pixels are correctly made.

## 4.3.2. StressUNet

### 4.3.2.1. Hyperparameters

The best performing StressUNet model was trained using the outlined in Table 4.4.

TABLE 4.4. Optimal StressUNet Hyperparameters

Hyperparameters	Values
Optimiser	ADAM
Batch size	8
Learning rate	0.01
Epochs	500

### 4.3.2.2. Training and Validation Metrics

Figure 4.6 plots the performance metrics of the best performing StressUNet model, which shows consistent improvement in all metrics over the epochs, until the model prematurely stopped training at epoch 105, to avoid overfitting. The training took exactly 996.16 seconds, which corresponds to just over 15 minutes.

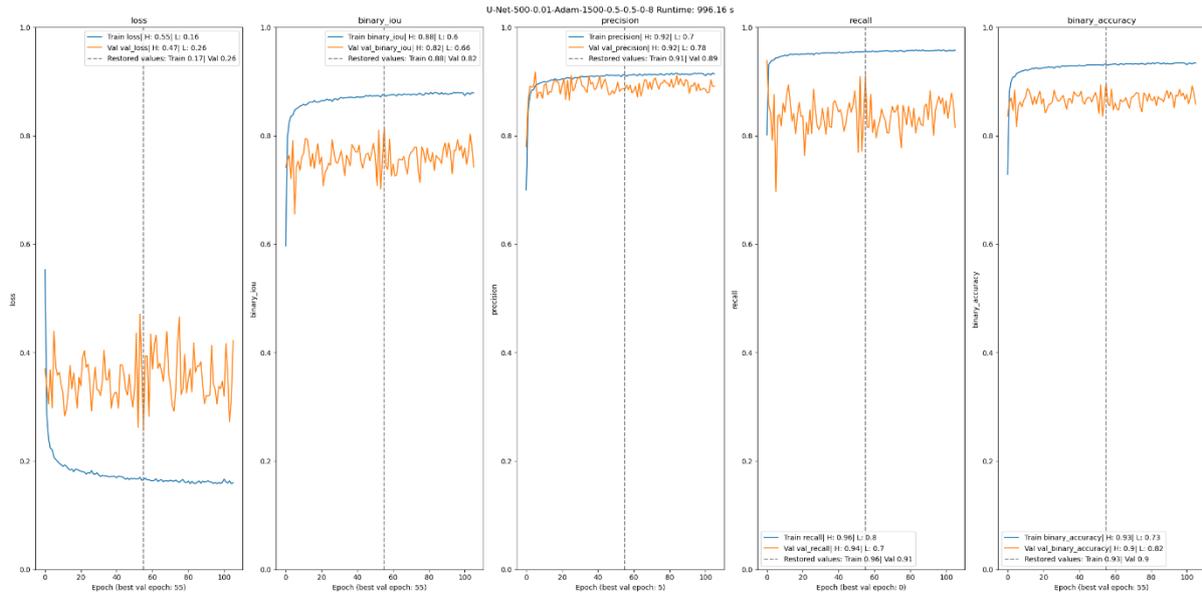


FIGURE 4.6. Plot of the Performance Metrics of the Best StressUNet Model

The restored model from epoch 55 achieved the lowest validation loss of 0.26 and a training loss of 0.17. The analysis of each plotted metric reveals that:

- **Loss:** The training loss shows an exponential decay, decreasing rapidly in the initial epochs before starting to plateau, indicating a quick convergence;
- **IoU:** The binary IoU values improved consistently, with the training IoU stabilising around 0.88 and the validation IoU achieving a high of 0.82;
- **Precision:** The precision values remained stable throughout, with the training precision reaching a high of 0.91 and the validation Precision achieving a high of 0.89. This validates that out of all WS predictions most contain WS;
- **Recall:** The recall values demonstrated consistent improvement, with the training recall stabilising around 0.96 and the validation recall reaching a high of 0.91. This ensures that most pixels labelled as containing WS are correctly predicted as WS;
- **Accuracy:** The binary accuracy values showed consistent improvement, with the training accuracy stabilising around 0.93 and the validation accuracy achieving a high of 0.90. This metric ensures that the model correctly predicts most pixels.

As described previously, Figure 4.7 displays the average confusion matrix for the best performing StressUNet model. Like the previous model, the number of samples,  $n$ , is calculated by multiplying the number of batched samples available in the validation dataset, 93.75, by the width, 128, and height, 128, dimensions of each image. However, by design, the incomplete batches are removed, hence only 93 validation batches were considered. Each quadrant contains a normalised value based on the relationship between the value of each pixel of every image in the validation dataset and the corresponding ground truth value of the segmentation mask.

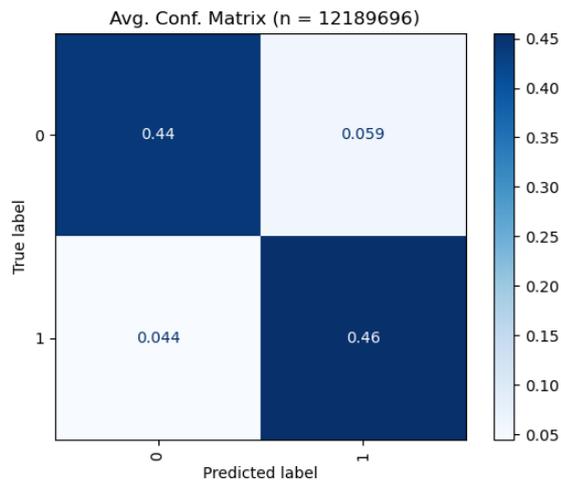


FIGURE 4.7. Average Confusion Matrix for the Best StressUNet Model

An analysis of the matrix reveals that 90% of pixels are correctly predicted, where the first quadrant reveals that 44% of the pixels are correctly predicted as not containing WS, while 46% are correctly predicted as WS, as indicated by the fourth quadrant. However, the second quadrant reveals that around 6% of the pixels were incorrectly predicted as containing WS, while the third quadrant shows that around 4.4% of the pixels were incorrectly labelled absent of WS.

To further improve the results, it is necessary to acquire additional images, in different vineyards during the growth cycle, as it would improve the model generalisation capabilities, since it would be exposed to a greater amount of information with different characteristics. Moreover, training different models by gradually unfreezing more layers from the WeedUNet model would result in a model that would balance the information retained from the previous model and the information necessary to the task of WS detection and localisation.



## Conclusions and Future Work

### 5.1. Conclusions

In this thesis it was possible to create a toolkit that can accurately identify vineyard areas experiencing WS. With these areas identified, it is possible to exclusively irrigate them, saving water resources and reducing potential yield losses. In addition, this PV WS toolkit relies on the usage of an DJI Mavic 3T, an UAV equipped with a thermal camera. By analysing the thermal images, it is possible to calculate the CWSI which serves as an indicator of early WS in vineyards.

To achieve this, two models were trained with the metrics BCE, IoU, Precision, Recall and Accuracy, to provide a complete representation of the predictive capabilities of the models. In addition, the training of the models involved holdout validation, where 50% of the samples were allocated for training and the remaining were used for validating the model during training. That way, by evaluating the validation loss metric, it is possible to prevent the model from overfitting.

The initial model was trained on Agriculture-Vision, a dataset containing aerial agricultural images, as existing pre-trained models are trained on datasets without aerial agricultural images and restrict the input dimensions. This model achieved a training loss of 0.07 and a validation loss of 0.14, where the other metrics used, namely IoU, Precision, Recall and Accuracy, for both training and validation sets, achieved over 90%. Afterwards, using transfer learning, the model was retrained using UAV captured thermal images to perform WS detection and localisation in vineyards. The UAV thermal image collection was conducted optimally in accordance with the literature, at 11 am, at a height of 60 meters, with 80% forward and side overlap, after letting the UAV warmup for 30 minutes. During flight, the parameters necessary to calculate the CWSI were established using a handheld thermometer. It is to note that the training process of each model undertook an optimisation process, using different hyperparameters, as well as relevant performance metrics to validate the obtained results.

The final model creates accurate WS predictions, by analysing aerial thermal images captured by a DJI Mavic 3T, as it achieves a training loss of 0.17 and a validation loss of 0.26. It also demonstrates strong performance across multiple metrics. Firstly, a training IoU of 0.88 and a validation IoU of 0.82, indicate that the models' predictions closely match the ground truths. Secondly, the training precision of 0.91 and a validation precision of 0.89 indicate that the model correctly identifies the actual positive prediction, out of all positive predictions. Thirdly, a training recall of 0.96 and a validation recall of 0.91 convey that out of all actual positive cases, most are correctly predicted as positive. Finally, a training accuracy of 0.93 and a validation accuracy of 0.9 confirm that most predictions are correctly made.

Despite the obtained results, data collection was only performed once during a single vineyard's growth cycle. This potentially limits the ability of the model to generalise, as different vineyards and growth cycles could have specific unseen details during training.

## **5.2. Future Work**

In the future, additional data collection during the entire growth cycle of different vineyards will be conducted. This helps to ensure that the model accurately detects and localises WS. Moreover, to further validate the results, WS assessments should be conducted in vineyards with drip irrigation, so that WS predictions may be compared to *in situ* water conditions. In addition, the process to create ground-truth masks highlighting WS, should be supervised by an agronomist or viticulturist to ensure that the created masks convey accurate site conditions. With additional images, labelled under the supervision of an agronomist or viticulturist, the model would be further trained so that it can perform accurate predictions, regardless of the specific characteristics of the vineyard and growth cycle.

Furthermore, it would be interesting to use a combination of additional hyperparameters and models, to validate whether the existing hyperparameters are optimal and to compare the purpose-built model for WS detection compared to other pre-trained models.

Finally, the application will be further developed to include other functionalities, such as the ability to reconstruct the images into a single georeferenced image. This opens the possibility to integrate the platform with other systems, such as agricultural UAVs, to create a pipeline from WS detection to automatic dispatch of agricultural UAVs to irrigate designated areas.

## References

- [1] R. Silva and F. Neiva, "Systematic Literature Review in Computer Science - A Practical Guide," Nov. 2016. doi: 10.13140/RG.2.2.35453.87524.
- [2] "What is b-on?," <https://www.b-on.pt/en/what-is-b-on/>.
- [3] "Scopus: Comprehensive, multidisciplinary, trusted abstract and citation database," [https://www.elsevier.com/products/scopus?dgcid=RN\\_AGCM\\_Sourced\\_300005030](https://www.elsevier.com/products/scopus?dgcid=RN_AGCM_Sourced_300005030).
- [4] "About the ACM Digital Library," <https://dl.acm.org/about>.
- [5] "Precision Ag Definition | International Society of Precision Agriculture." Accessed: Aug. 19, 2024. [Online]. Available: <https://www.ispag.org/about/definition>
- [6] N. Amarasingam, A. S. Ashan Salgadoe, K. Powell, L. F. Gonzalez, and S. Natarajan, "A review of UAV platforms, sensors, and applications for monitoring of sugarcane crops," *Remote Sens Appl*, vol. 26, p. 100712, Apr. 2022, doi: 10.1016/j.rsase.2022.100712.
- [7] Y. Han *et al.*, "Calibration and Image Processing of Aerial Thermal Image for UAV Application in Crop Water Stress Estimation," *J Sens*, vol. 2021, no. 1, Jan. 2021, doi: 10.1155/2021/5537795.
- [8] M. Awais *et al.*, "Assessment of optimal flying height and timing using high-resolution unmanned aerial vehicle images in precision agriculture," *International Journal of Environmental Science and Technology*, vol. 19, no. 4, pp. 2703–2720, Apr. 2022, doi: 10.1007/s13762-021-03195-4.
- [9] World Economic Forum, "The Global Risks Report 2024," 19th Edition. [Online]. Available: <https://www.weforum.org/publications/global-risks-report-2024/>
- [10] M. V. Ferro and P. Catania, "Technologies and Innovative Methods for Precision Viticulture: A Comprehensive Review," *Horticulturae*, vol. 9, no. 3, p. 399, Mar. 2023, doi: 10.3390/horticulturae9030399.
- [11] C. Yang *et al.*, "Assessing the grapevine crop water stress indicator over the flowering-veraison phase and the potential yield lose rate in important European wine regions," *Agric Water Manag*, vol. 261, p. 107349, Aug. 2022, doi: 10.1016/j.agwat.2021.107349.
- [12] M. Ammoniacci, S.-P. Kartsiotis, R. Perria, and P. Storchi, "State of the Art of Monitoring Technologies and Data Processing for Precision Viticulture," *Agriculture*, vol. 11, no. 3, p. 201, Feb. 2021, doi: 10.3390/agriculture11030201.
- [13] E. Tunca, E. S. Köksal, and S. Ç. Taner, "Calibrating UAV thermal sensors using machine learning methods for improved accuracy in agricultural applications," *Infrared Phys Technol*, vol. 133, p. 104804, Aug. 2023, doi: 10.1016/j.infrared.2023.104804.

- [14] R. D. Jackson, S. B. Idso, R. J. Reginato, and P. J. Pinter, "Canopy temperature as a crop water stress indicator," *Water Resour Res*, vol. 17, no. 4, pp. 1133–1138, Aug. 1981, doi: 10.1029/WR017i004p01133.
- [15] D. Sepúlveda-Reyes, B. Ingram, M. Bardeen, M. Zúñiga, S. Ortega-Farías, and C. Poblete-Echeverría, "Selecting Canopy Zones and Thresholding Approaches to Assess Grapevine Water Status by Using Aerial and Ground-Based Thermal Imaging," *Remote Sens (Basel)*, vol. 8, no. 10, p. 822, Oct. 2016, doi: 10.3390/rs8100822.
- [16] M. Awais *et al.*, "Remotely sensed identification of canopy characteristics using UAV-based imagery under unstable environmental conditions," *Environ Technol Innov*, vol. 22, 2021, doi: 10.1016/j.eti.2021.101465.
- [17] S. B. Idso, R. J. Reginato, R. D. Jackson, and P. J. Pinter, "Foliage and air temperatures: Evidence for a dynamic 'equivalence point,'" *Agricultural Meteorology*, vol. 24, pp. 223–226, Jan. 1981, doi: 10.1016/0002-1571(81)90046-7.
- [18] H. G. Jones, "Use of infrared thermography for monitoring stomatal closure in the field: application to grapevine," *J Exp Bot*, vol. 53, no. 378, pp. 2249–2260, Nov. 2002, doi: 10.1093/jxb/erf083.
- [19] EASA, "Easy Access Rules for Unmanned Aircraft Systems - Revision from July 2024 ."
- [20] EASA, "Drones & EVTOL Designs."
- [21] M. Murison, "Fixed-Wing vs Multirotor: Which Drone Should You Choose for Aerial Surveying?"
- [22] A. Bacchini, E. Cestino, B. Van Magill, and D. Verstraete, "Impact of lift propeller drag on the performance of eVTOL lift+cruise aircraft," *Aerosp Sci Technol*, vol. 109, p. 106429, Feb. 2021, doi: 10.1016/j.ast.2020.106429.
- [23] FAA, "Certificated Remote Pilots including Commercial Operators," [https://www.faa.gov/uas/commercial\\_operators](https://www.faa.gov/uas/commercial_operators).
- [24] CASA, "Remote pilot licence," <https://www.casa.gov.au/drones/get-your-operator-credentials/remote-pilot-licence>.
- [25] T.-H. Tran and D.-D. Nguyen, "Management and Regulation of Drone Operation in Urban Environment: A Case Study," *Soc Sci*, vol. 11, no. 10, p. 474, Oct. 2022, doi: 10.3390/socsci11100474.
- [26] EASA, "Open Category - Low Risk - Civil Drones," <https://www.easa.europa.eu/en/domains/civil-drones-rpas/open-category-civil-drones>.
- [27] W. H. Maes and K. Steppe, "Perspectives for Remote Sensing with Unmanned Aerial Vehicles in Precision Agriculture," *Trends Plant Sci*, vol. 24, no. 2, pp. 152–164, Feb. 2019, doi: 10.1016/j.tplants.2018.11.007.

- [28] G. Rivera, R. Porras, R. Florencia, and J. P. Sánchez-Solís, "LiDAR applications in precision agriculture for cultivating crops: A review of recent advances," *Comput Electron Agric*, vol. 207, p. 107737, Apr. 2023, doi: 10.1016/j.compag.2023.107737.
- [29] G. Messina and G. Modica, "Applications of UAV Thermal Imagery in Precision Agriculture: State of the Art and Future Research Outlook," *Remote Sens (Basel)*, vol. 12, no. 9, p. 1491, May 2020, doi: 10.3390/rs12091491.
- [30] Y. Shendryk, J. Sofonia, R. Garrard, Y. Rist, D. Skocaj, and P. Thorburn, "Fine-scale prediction of biomass and leaf nitrogen content in sugarcane using UAV LiDAR and multispectral imaging," *International Journal of Applied Earth Observation and Geoinformation*, vol. 92, p. 102177, Aug. 2020, doi: 10.1016/j.jag.2020.102177.
- [31] H. Sheng, H. Chao, C. Coopmans, J. Han, M. McKee, and Y. Chen, "Low-cost UAV-based thermal infrared Remote sensing: platform, calibration and applications," *IEEE*, Aug. 2010, doi: 10.1109/mesa.2010.5552031.
- [32] M. P. Finn *et al.*, "Remote sensing of soil moisture using airborne hyperspectral data," *Gisci Remote Sens*, vol. 48, no. 4, pp. 522–540, Aug. 2011, doi: 10.2747/1548-1603.48.4.522.
- [33] C. Stryker and E. Kavlakoglu, "What is Artificial Intelligence (AI)?," <https://www.ibm.com/topics/artificial-intelligence>.
- [34] J. Kufel *et al.*, "What is machine Learning, artificial neural networks and Deep Learning?— Examples of Practical applications in medicine," *Diagnostics*, vol. 13, no. 15, p. 2582, Aug. 2023, doi: 10.3390/diagnostics13152582.
- [35] M. Khan, B. Jan, and H. Farman, *Deep Learning: Convergence to Big Data Analytics*. Singapore: Springer Singapore, 2019. doi: 10.1007/978-981-13-3459-7.
- [36] L. Aversano, M. L. Bernardi, and M. Cimitile, "Water stress classification using Convolutional Deep Neural Networks," *JUCS - Journal of Universal Computer Science*, vol. 28, no. 3, pp. 311–328, Mar. 2022, doi: 10.3897/jucs.80733.
- [37] J. Murel and E. Kavlakoglu, "What is transfer learning?," <https://www.ibm.com/topics/transfer-learning>.
- [38] "Neural Network: Introduction to Learning Rate," <https://studymachinelearning.com/neural-network-introduction-to-learning-rate/>.
- [39] "The role of bias in Neural Networks," <https://www.pico.net/kb/the-role-of-bias-in-neural-networks/>.
- [40] "Forward Propagation, Backward Propagation, and Computational Graphs," [https://d2l.ai/chapter\\_multilayer-perceptrons/backprop.html](https://d2l.ai/chapter_multilayer-perceptrons/backprop.html).
- [41] V. Wiley and T. Lucas, "Computer Vision and Image Processing: A paper review," *International Journal of Artificial Intelligence Research*, vol. 2, no. 1, p. 22, Aug. 2018, doi: 10.29099/ijair.v2i1.42.

- [42] R. Szeliski, *Computer Vision: Algorithms and Applications*. Cham: Springer International Publishing, 2022. doi: 10.1007/978-3-030-34372-9.
- [43] "Ultralytics YOLO11," <https://docs.ultralytics.com/models/yolo11/>.
- [44] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," Sep. 2014.
- [45] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," Apr. 2017.
- [46] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," 2015, pp. 234–241. doi: 10.1007/978-3-319-24574-4\_28.
- [47] "Image Thresholding," [https://docs.opencv.org/4.x/d7/d4d/tutorial\\_py\\_thresholding.html](https://docs.opencv.org/4.x/d7/d4d/tutorial_py_thresholding.html).
- [48] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Trans Syst Man Cybern*, vol. 9, no. 1, pp. 62–66, 1979, doi: 10.1109/TSMC.1979.4310076.
- [49] "Smoothing Images," [https://docs.opencv.org/4.x/d4/d13/tutorial\\_py\\_filtering.html](https://docs.opencv.org/4.x/d4/d13/tutorial_py_filtering.html).
- [50] W. Deuschle, "Undergraduate Fundamentals of Machine Learning," Harvard, 2019.
- [51] A. Roberts, "Binary Cross Entropy: Where To Use Log Loss In Model Monitoring," <https://arize.com/blog-course/binary-cross-entropy-log-loss/>.
- [52] "Introduction of Holdout Method," <https://www.geeksforgeeks.org/introduction-of-holdout-method/>.
- [53] "Cross Validation in Machine Learning," <https://www.geeksforgeeks.org/cross-validation-machine-learning/>.
- [54] S. B. Kotsiantis, "Supervised Machine Learning: A Review of Classification Techniques," in *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real World AI Systems with Applications in EHealth, HCI, Information Retrieval and Pervasive Technologies*, NLD: IOS Press, 2007, pp. 3–24.
- [55] J. Sen, Ed., *Machine Learning - Algorithms, Models and Applications*, vol. 7. IntechOpen, 2021. doi: 10.5772/intechopen.94615.
- [56] "Classification: Accuracy, recall, precision, and related metrics," <https://developers.google.com/machine-learning/crash-course/classification/accuracy-precision-recall>.
- [57] S. Kumar, "Metrics to Evaluate your Classification Model to take the Right Decisions," <https://www.analyticsvidhya.com/blog/2021/07/metrics-to-evaluate-your-classification-model-to-take-the-right-decisions/>.
- [58] D. Shah, "Intersection over Union (IoU): Definition, Calculation, Code," <https://www.v7labs.com/blog/intersection-over-union-guide>.

- [59] “Thresholds and the confusion matrix,” [https://developers.google.com/machine-learning/crash-course/classification/thresholding#confusion\\_matrix](https://developers.google.com/machine-learning/crash-course/classification/thresholding#confusion_matrix).
- [60] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, and S. Savarese, “Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression,” Feb. 2019.
- [61] D. Choi, C. J. Shallue, Z. Nado, J. Lee, C. J. Maddison, and G. E. Dahl, “On Empirical Comparisons of Optimizers for Deep Learning,” Oct. 2019.
- [62] H. Robbins and S. Monro, “A Stochastic Approximation Method,” *The Annals of Mathematical Statistics*, vol. 22, no. 3, pp. 400–407, Sep. 1951, doi: 10.1214/aoms/1177729586.
- [63] V. Baskaran, “Momentum optimizer,” <https://medium.com/@vinodhb95/momentum-optimizer-6023aa445e18>.
- [64] Y. Dong, “Hyper Parameter—Momentum,” <https://medium.com/ai%C2%B3-theory-practice-business/hyper-parameter-momentum-dc7a7336166e>.
- [65] B. T. Polyak, “Some methods of speeding up the convergence of iteration methods,” *USSR Computational Mathematics and Mathematical Physics*, vol. 4, no. 5, pp. 1–17, Jan. 1964, doi: 10.1016/0041-5553(64)90137-5.
- [66] M. Yawar, “Nesterov Accelerated Gradient,” <https://www.naukri.com/code360/library/nesterov-accelerated-gradient>.
- [67] Yu. E. Nesterov, “One class of methods of unconditional minimization of a convex function, having a high rate of convergence,” *USSR Computational Mathematics and Mathematical Physics*, vol. 24, no. 4, pp. 80–82, Jan. 1984, doi: 10.1016/0041-5553(84)90234-9.
- [68] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” 2017. [Online]. Available: <https://arxiv.org/abs/1412.6980>
- [69] P. M. Radiuk, “Impact of Training Set Batch Size on the Performance of Convolutional Neural Networks for Diverse Datasets,” *Information Technology and Management Science*, vol. 20, no. 1, Jan. 2017, doi: 10.1515/itms-2017-0003.
- [70] “Epoch in Machine Learning,” <https://www.geeksforgeeks.org/epoch-in-machine-learning/>.
- [71] J. Brownlee, “Understand the Impact of Learning Rate on Neural Network Performance,” <https://machinelearningmastery.com/understand-the-dynamics-of-learning-rate-on-deep-learning-neural-networks/>.
- [72] A. D. Rasamoelina, F. Adjailia, and P. Sincak, “A Review of Activation Function for Artificial Neural Network,” in *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMII)*, IEEE, Jan. 2020, pp. 281–286. doi: 10.1109/SAMI48414.2020.9108717.
- [73] A. Amini, A. Amini, and S. Lolla, “Introduction to Deep Learning,” <http://introtodeeplearning.com/>.

- [74] A. Awan, "A Complete Guide to Data Augmentation," <https://www.datacamp.com/tutorial/complete-guide-data-augmentation>.
- [75] "Keras About," <https://keras.io/about/>.
- [76] "TensorFlow About," <https://www.tensorflow.org/about>.
- [77] "NVIDIA cuDNN," <https://developer.nvidia.com/cudnn>.
- [78] Django Software Foundation, "Django." [Online]. Available: <https://djangoproject.com>
- [79] C. Humhong and R. Roldán, "Thermal Parser," [https://github.com/SanNianYiSi/thermal\\_parser](https://github.com/SanNianYiSi/thermal_parser).
- [80] M. Maynard-Reid, "U-Net Image Segmentation in Keras," *PyImageSearch*, 2022.
- [81] F. Chollet, "The Functional API," [https://keras.io/guides/functional\\_api/](https://keras.io/guides/functional_api/).
- [82] M. T. Chiu *et al.*, "Agriculture-Vision: a large aerial image database for agricultural pattern analysis," *IEEE*, Aug. 2020, doi: 10.1109/cvpr42600.2020.00290.
- [83] M. Tkachenko, M. Malyuk, A. Holmanyuk, and N. Liubimov, "Label Studio," 2020.
- [84] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [85] "Django Project MVT Structure," <https://www.geeksforgeeks.org/django-project-mvt-structure/>.
- [86] "Specs - DJI Mavic 3 Enterprise - DJI Enterprise." [Online]. Available: <https://enterprise.dji.com/mavic-3-enterprise/specs>
- [87] "A1/A3 Course & Examination - UAS Remote Pilot Open Category - [UAS-OPEN-A1+A3]," <https://learningzone.eurocontrol.int/ilp/pages/description.jsf#/users/@self/catalogues/8264768/courses/20790716/description>.

# Dissertation Article

This dissertation article was accepted and published in the 2024 International Symposium on Sensing and Instrumentation in 5G and IoT Era (ISSI), available at IEEE Xplore 10.1109/ISSI63632.2024.10720501.

## A Deep Learning Toolkit for Water Stress Detection in Viticulture

Gustavo Ferreira  
Department of Information Science and  
Technology  
ISCTE-IUL & IT-IUL  
Lisbon, Portugal  
glsfa@iscte-iul.pt

Octavian Postolache  
Department of Information Science and  
Technology  
ISCTE-IUL & IT-IUL  
Lisbon, Portugal  
oapen@iscte-iul.pt

Pedro Sebastião  
Department of Information Science and  
Technology  
ISCTE-IUL & IT-IUL  
Lisbon, Portugal  
pedro.sebastiao@iscte-iul.pt

**Abstract**— The paper addresses the critical issue of water stress in viticulture, which is vital for improving grape yield and quality. The use of advanced deep learning methods and UAVs for data collection significantly enhances the accuracy and efficiency of water stress monitoring. The development of a Django-based web platform for interactive prediction and reporting to users is a substantial contribution to the practical applicability of the research. The proposed algorithm, based on the U-Net neural network, segments images to detect water stress using aerial RGB and thermal imagery. The model was successfully trained on the Agriculture-Vision dataset, showing promising results in segmenting agricultural patterns. Due to unfavourable weather conditions, data collection was limited, which may affect the completeness and reliability of the results. The use of transfer learning requires further refinement of the model to optimize parameters and adapt to new data.  
**Index Terms**— UAV, RGB, Thermal, Deep learning, Convolutional neural networks, Water stress detection

advancements in the remote sensing and machine learning offer promising alternatives reducing operational costs and optimising watering tasks. Thus, using aerial imagery and complex algorithms, it is possible to assess water stress over large vineyard areas with greater accuracy and efficiency.

In this work is proposed the development of a web-based platform that supports Artificial Intelligence (AI). An Unmanned Aerial Vehicle (UAV), equipped with Red-Green-Blue (RGB) and thermal sensors are used to gather the necessary data that it is used to determine the presence of water stress. The proposed platform will be able to evaluate the images to:

- Create a visual representation of the areas with water stress.
- Provide a report with relevant information, related to aggregated data from the collected images

### I. INTRODUCTION

In the context of global risks, the World Economic Forum outlines in the 2024 global risk report, climate-related threats as the dominant risk throughout the next decade. These pose a serious risk, in terms of food, water and health security [1]. It is of great importance to reduce factors that could negatively affect food production, ensuring food security. One of these factors is water stress.

Water stress occurs when the available water resources are insufficient to meet a plant's water needs, or, when the water resources exceed the plant's water needs. For reduced water availability, this gradually increases stomatal closure, which leads to reduced evaporation and increases the plants' temperature. Conversely, excess water could lead to diseases that impact a plant's health [2].

In 2020, Europe produced 63% of the world's total wine production, and in viticulture, water stress directly impacts grape yield and quality, thus, it is necessary to detect it accurately and reliably, to avoid potential losses [3]. Traditional methods of monitoring water stress involve in-situ measurements, which is time-consuming, labour intensive, costly and often, destructive [4]. Recent

### II. RELATED WORK: PRECISION AGRICULTURE

Precision Agriculture (PA) is a farming management technique that utilises different technologies, such as UAVs and Artificial Intelligence (AI), with the goal to optimise the inputs and maximise quality and productivity. A consequence of PA is the reduction of operational costs and more profitability, as resources are spent in a time and location sensitive manner [5], [6]. Figure 1 summarises PA as a cycle, alternating between minimising farming inputs while maximising outputs.

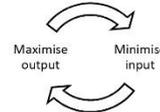


Fig. 1 - PA optimising cycle

Contrasting traditional farming practises, a field is treated in a heterogeneous way, where different parts may have different characteristics and requirements.

### A. Remote Sensing

Remote Sensing (RS) involves the use of technologies to indirectly collect timely and accurate data. In PA, RS enables the collection of data that would otherwise be impractical or challenging to collect. Particularly, aerial RS is less time consuming, labour intensive and costly, when compared to ground-based methods [4], [7].

With regards to aerial-based RS, there are several platforms that have been used, each with different characteristics, advantages and disadvantages. High altitude platforms such as satellites are characterized by having low spatial resolution, long revisit times, high associated costs and affected by weather phenomena, such as clouds [8]. Airplanes are a lower altitude alternative, which offer finer spatial resolution and greater control over temporal resolution [9]. Nevertheless, for small scale applications, these are not a viable option [2].

### B. UAV

The European Union Aviation Safety Agency categorises UAVs into several distinct categories. These include fixed-wing and rotor-based designs, namely single-rotor, multi-rotor or Vertical Take-Off and Landing (VTOL) [10]. Multi rotor UAVs have more two or more rotors that control lift, propulsion and movement (example, quadcopters or hexacopters). These designs have VTOL capabilities, are versatile and manoeuvrable, but have a lower flight time, compared to other designs (namely, fixed-wing designs) [6, 10].

Compared to other aerial-based RS platforms, an UAV is an excellent candidate as it operates at a lower altitude, provides higher spatial and temporal resolution. Additionally, these have lower operational costs, shorter revisit time and higher independence of climatic variables [11], [12].

UAVs have successfully been used as an aerial RS platform, in PA, using a multitude of sensors. These include, RGB, Multispectral, Hyperspectral, Light Detecting and Ranging (LiDAR) and Thermal. RGB cameras are low cost, easy to use and have high spatial resolution for limited spectral bands [6, 13]. In 2018, in France, the researchers of [20] utilised an RGB equipped UAV to collect aerial data of a vineyard, to train an algorithm to detect grape plant disease. On the other hand, uncooled thermal sensors are less complex, smaller and less expensive than cooled thermal sensors. This makes them ideal for UAV applications. These are sensitive to the thermal infrared radiation of the electromagnetic spectrum. The main drawbacks associated with these include measurement error, induced by atmospheric attenuation, sensor calibration and higher cost, compared to RGB cameras [6], [13]. The researchers of [14] utilise UAV captured RGB and thermal images to train an algorithm to detect and classify water stress in tomato plants.

### C. Artificial Intelligence

Artificial Intelligence (AI) enables a machine the ability to simulate human intelligence by learning, without being programmed to do so.

Machine Learning (ML) is a subset of AI that enables computers to learn new skills, through data [15]. The learning process of these algorithms can be supervised, unsupervised or by reinforcement, where the choice depends on the task they are designed to perform. In supervised learning the algorithm is provided an expected output and the model gains knowledge as it establishes a relationship between input and output data. In contrast, unsupervised algorithms only receive the input data, and learn based on the characteristics of the data. Lastly, in reinforcement learning as the model interacts with an environment, it is rewarded or penalised, based on his decisions [15], [16].

Deep Learning (DL) draws inspiration from natural processes, functioning similarly to the human brain. DL algorithms use Deep Neural Networks (DNNs), which organize neurons across multiple layers. Unlike traditional neural networks with a single hidden layer, DNNs have several hidden layers. This architecture enables DNNs to tackle complex and abstract problems. The input layer receives data features and passes them to subsequent layers. Hidden layers gradually extract higher-level features, performing feature selection as they transform data into more abstract representations [15], [17].

A Convolutional Neural Network (CNN) is a supervised, feed-forward neural network designed for tasks like image recognition. It consists of convolutional layers, pooling layers, and fully connected layers. CNNs use filters or kernels to extract features from data during training. The feature maps generated by convolutions contain the highest activation values. Pooling layers reduce feature map dimensions, aiding computational efficiency. Fully connected layers integrate localized attributes to synthesize overarching characteristics [17]. In the field of Computer Vision (CV), DL enables computers to interpret and extract meaningful information from digital images, encompassing tasks such as image recognition, object detection, and image segmentation [18]. In [14], the researchers chose to use two CNN models to classify the level of water stress in the UAV captured images.

Transfer learning is a ML technique that uses the knowledge from a pre-trained network, to solve other, similar problems. The steps to implement transfer learning involve [14]:

- Freezing the layers from the pre-trained model, to prevent overwriting the existing weights;
- Add additional trainable layers that will use the pre-trained features to make predictions for the new task;

### III. MATERIALS AND METHODS

#### A. UAV Imagery

The UAV chosen for this study is a DJI Mavic 3 thermal, from Da-Jiang Innovations. This UAV is a multi-rotor design, specifically a quadcopter, having VTOL capabilities and precise manoeuvrability, which is necessary for accurate data collection. Moreover, the UAV can be easily transported as it features foldable rotors and can fly for up to 45 minutes, having the ability to swap batteries and resume the last data collection point [19]. Figure 2 shows the UAV mid-flight.



Fig. 2 - DJI Mavic 3T

This UAV has a maximum take-off weight of 1050 grams, and it features two sensors, a 48 Megapixels RGB wide main camera and an uncooled thermal sensor, sensitive to the infrared wavelength of 8-14 $\mu$ m, with a temperature measurement accuracy of  $\pm 2^\circ$  C or  $\pm 2$ , in the range of  $-20^\circ$ C to  $150^\circ$ C. The sensors are attached to a 3-axis (roll, tilt and pan) gimbal, which greatly reduces any distortions that may occur. Table 1 and Table 2 fully summarize the information related to each sensor equipped on the UAV [19].

Table 1 - DJI Mavic 3T RGB camera specifications [19]

Sensor	RGB
Sensor type	1/2-inch CMOS
Effective pixels	48MP
Field of view	84°
Format equivalent	24mm
Aperture	f/2.8
ISO Range	100-25600
Shutter type and speed	Electronic Shutter: 8-1/8000 s
Max Image Size	8000 $\times$ 6000

Table 2 - DJI Mavic 3T Thermal camera specifications [19]

Sensor	Thermal
Thermal Imager	Uncooled VOx Microbolometer
Pixel Pitch	12 $\mu$ m
Frame Rate	30 Hz
Lens	Display Field of View: 61° Format Equivalent: 40 mm Aperture: f/1.0 Focus: 5 m to $\infty$
Noise Equivalent Temperature Difference (NETD)	$\leq 50$ mK@F1.0
Temperature Measurement Method	$-20^\circ$ to $150^\circ$ C ( $-4^\circ$ to $302^\circ$ F, High Gain Mode)
Photo Format	JPEG (8-bit) R-JPEG (16-bit)
Max Image Size	640 $\times$ 512
Infrared Wavelength	8-14 $\mu$ m
Infrared Temperature Measurement Accuracy	$\pm 2^\circ$ C or $\pm 2\%$ (using the larger value)

#### B. Data collection

To mitigate any thermal measurement error that may occur, the data collection process follows the methodologies outlined in [20]. Before each flight, the UAV is allowed to warm up for 30 minutes, where it then takes off and flies in optimal predefined flight path with the following characteristics:

- Relative height of 60 m;
- 80% forward and side overlap;
- Flights are performed under clear skies, similar weather conditions, from 11:00h until 13:00h, local time.

#### C. Image processing algorithms

The U-Net architecture, proposed by [21] is a CNN that was designed for biomedical image segmentation. This network combines a contracting path for context capture and an expanding path for precise localisation, which results in an effective model for image segmentation tasks (Figure 3), having won the EM segmentation challenge at ISBI 2012 by a large margin.

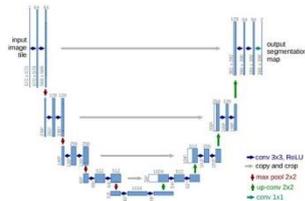


Fig. 3 U-Net model architecture [21]

The model also includes skip connections that retain high-resolution features, that would otherwise be lost during down-sampling with the ability to learn from limited annotated data. For these reasons U-Net was chosen for the image segmentation task at hand.

In this approach, the model is initially trained on the Agriculture-Vision dataset [22] using RGB images that have been resized to have a height and width of 128 by 128 pixels, respectively, to detect weed clusters. This step helps the model learn essential features and patterns related to weed identification. Afterwards, using transfer learning, the model is adapted to receive the UAV captured images, containing the RGB and thermal channels, for the task of detecting the presence of water stress. This adaptation is crucial as it leverages on both the visible and infrared parts of the light spectrum to effectively identify signs of water stress.

#### D. Prediction Platform

The Django Web Framework was utilised to develop an interactive platform for making water stress predictions. This enables users to upload images to the website and receive a report relating to the water stress found in the provided images [23]. Figure 4 shows the initial page of the website.



Fig. 4 - AVision Initial Webpage

#### IV. RESULTS AND DISCUSSION

Despite challenges faced during the acquisition of the UAV and more recently, impossibility to collect data due to poor weather conditions, unusual for the time of year, efforts to capture data for water stress detection are ongoing and expected to be resolved soon. In the meantime, the U-Net has successfully been trained using the Agriculture-Vision dataset [22], with a focus on the agricultural pattern "weed\_cluster", as there are over ten thousand valid samples containing this pattern. The model was provided a dataset

containing 6000 images, with 70% used for training, 20% for validation and 10% for testing.

The TensorFlow, open source library for numerical computation was then used to train the model to detect weed clusters and demonstrated promising results for segmenting this pattern. Figure 5 displays the input image, true mask and model prediction, where despite a roughly created true mask, the output closely resembles the pattern on the input mask.



Fig. 5 - Input Image, True Mks and Predicted Mask for "weed\_cluster"

These promising results indicate the model's capability and effectiveness, suggesting that once the new images are acquired, the model will likely perform well in detecting water stress in aerial RGB and thermal images.

#### V. CONCLUSIONS

The current implementation of the algorithm leverages on the U-Net architecture, in conjunction with the Agriculture-Vision dataset, which has shown promising results in segmenting agricultural images, which highlights its potential for water stress detection.

The next steps will involve the data collection and using this pre-trained model, train it to determine the presence of water stress. The model will be further fine-tuned to find the optimal parameters, as opposed to using fixed parameters. After this step, the model will be integrated with the website, to provide a report to its user, enabling them to treat the affected areas effectively, thus reducing any yield losses related to water stress. Finally, this tool will be further developed, to work with different cultures and integrate other types of sensors, such as multispectral, to perform different types of detections.

#### ACKNOWLEDGMENT

I would like to thank Instituto de Telecomunicações/ISCTE-Instituto Universitário de Lisboa for having supported the laboratories and equipments necessary to develop this work. Also, I would like to thank projects SmartFarm 4.0 (national project from collaborative lab, SmartFarmColab, Projeto mobilizador no 46078 POCI-01-0247-FEDER-046078. <https://www.scolab.org/smartfarm4>) and SmartVitiNet (European project, co-Funded by the European Union under Grant Agreement number 101083737, <https://www.smartvitet.net>).

REFERENCES

- [1] "Global Risks Report 2024 | World Economic Forum," Aug. 2024. [Online]. Available: <https://www.weforum.org/publications/global-risks-report-2024/>
- [2] E. Tunca, E. S. Köksal, and S. Ç. Taner, "Calibrating UAV thermal sensors using machine learning methods for improved accuracy in agricultural applications," *Infrared Phys Technol*, vol. 133, p. 104804, Aug. 2023, doi: 10.1016/j.infrared.2023.104804.
- [3] C. Yang *et al.*, "Assessing the grapevine crop water stress indicator over the flowering-veraison phase and the potential yield loss rate in important European wine regions," *Agric Water Manag*, vol. 261, p. 107349, Aug. 2022, doi: 10.1016/j.agwat.2021.107349.
- [4] Y. Han *et al.*, "Calibration and image processing of aerial thermal image for UAV application in crop water stress estimation," *J Sens*, vol. 2021, pp. 1–14, Aug. 2021, doi: 10.1155/2021/5537795.
- [5] Precision AG Definition | International Society of Precision Agriculture., "International Society of Precision Agriculture.," <https://www.ispag.org/about/definition>.
- [6] N. Amarasingam, A. S. A. Salgadoe, K. Powell, L. F. Gonzalez, and S. Natarajan, "A review of UAV platforms, sensors, and applications for monitoring of sugarcane crops," *Remote Sens Appl*, vol. 26, p. 100712, Aug. 2022, doi: 10.1016/j.rsase.2022.100712.
- [7] Y. Shendryk, J. Sofonia, R. Garrard, Y. Rist, D. Skocaj, and P. Thorburn, "Fine-scale prediction of biomass and leaf nitrogen content in sugarcane using UAV LiDAR and multispectral imaging," *International Journal of Applied Earth Observation and Geoinformation*, vol. 92, p. 102177, Aug. 2020, doi: 10.1016/j.jag.2020.102177.
- [8] H. Sheng, H. Chao, C. Coopmans, J. Han, M. McKee, and Y. Chen, "Low-cost UAV-based thermal infrared Remote sensing: platform, calibration and applications," *IEEE*, Aug. 2010, doi: 10.1109/imesa.2010.5552031.
- [9] M. P. Finn *et al.*, "Remote sensing of soil moisture using airborne hyperspectral data," *Glsci Remote Sens*, vol. 48, no. 4, pp. 522–540, Aug. 2011, doi: 10.2747/1548-1603.48.4.522.
- [10] "Drones & EVTOL Designs | EASA." [Online]. Available: <https://www.easa.europa.eu/en/domains/drones-air-mobility/drones-cvtol-designs>
- [11] Y. Han *et al.*, "Calibration and image processing of aerial thermal image for UAV application in crop water stress estimation," *J Sens*, vol. 2021, pp. 1–14, Aug. 2021, doi: 10.1155/2021/5537795.
- [12] W. Li *et al.*, "A UAV-aided prediction system of soil moisture content relying on thermal infrared remote sensing," *International Journal of Environmental Science and Technology*, vol. 19, no. 10, pp. 9587–9600, Aug. 2022, doi: 10.1007/s13762-022-03958-7.
- [13] W. H. Maes and K. Steppe, "Perspectives for Remote Sensing with Unmanned Aerial Vehicles in Precision Agriculture," *Trends Plant Sci*, vol. 24, no. 2, pp. 152–164, Aug. 2019, doi: 10.1016/j.tplants.2018.11.007.
- [14] L. Aversano, M. L. Bernardi, and M. Cimitile, "Water stress classification using Convolutional Deep Neural Networks," *JUCS - Journal of Universal Computer Science*, vol. 28, no. 3, pp. 311–328, Aug. 2022, doi: 10.3897/jucs.80733.
- [15] J. Kufel *et al.*, "What is machine Learning, artificial neural networks and Deep Learning?—Examples of Practical applications in medicine," *Diagnostics*, vol. 13, no. 15, p. 2582, Aug. 2023, doi: 10.3390/diagnostics13152582.
- [16] C. Staff, "3 types of machine learning you should know," Aug. 2024. [Online]. Available: [https://www.coursera.org/articles/types-of-machine-learning?utm\\_medium=sem&utm\\_source=gg&utm\\_campaign=B2C\\_EMEA\\_coursera\\_FTcoF\\_career-academy\\_pmax-multiple-audiences-country-multi-act2&campaignid=20882109092&adgroupid=&device=c&keyword=&matchtype=&network=x&deviceid=&adposition=&creativeid=&hide\\_mobile\\_promo&gad\\_source=1&gclid=CjwKCAjwvwmzBhA2EiwATHVrb3Z3MJXdYvHQoBmXIIyQ\\_g7jRyDdBowJK7bNI-IjZNe7JHSO8FOhOChRRIQAvD\\_BwE](https://www.coursera.org/articles/types-of-machine-learning?utm_medium=sem&utm_source=gg&utm_campaign=B2C_EMEA_coursera_FTcoF_career-academy_pmax-multiple-audiences-country-multi-act2&campaignid=20882109092&adgroupid=&device=c&keyword=&matchtype=&network=x&deviceid=&adposition=&creativeid=&hide_mobile_promo&gad_source=1&gclid=CjwKCAjwvwmzBhA2EiwATHVrb3Z3MJXdYvHQoBmXIIyQ_g7jRyDdBowJK7bNI-IjZNe7JHSO8FOhOChRRIQAvD_BwE)
- [17] M. Khan, B. Jan, and H. Farman, *Deep Learning: convergence to big data analytics*. 2019. doi: 10.1007/978-981-13-3459-7.
- [18] V. Wiley and T. Lucas, "Computer Vision and Image Processing: A paper review," *International Journal of Artificial Intelligence Research*, vol. 2, no. 1, p. 22, Aug. 2018, doi: 10.29099/ijair.v2i1.42.
- [19] "Specs - DJI Mavic 3 Enterprise - DJI Enterprise." [Online]. Available: <https://enterprise.dji.com/mavic-3-enterprise/specs>
- [20] M. Awais *et al.*, "Assessment of optimal flying height and timing using high-resolution unmanned aerial vehicle images in precision agriculture," *International Journal of Environmental Science and Technology*, vol. 19, no. 4, pp. 2703–2720, Aug. 2021, doi: 10.1007/s13762-021-03195-4.
- [21] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," 2015, pp. 234–241. doi: 10.1007/978-3-319-24574-4\_28.
- [22] M. T. Chiu *et al.*, "Agriculture-Vision: a large aerial image database for agricultural pattern analysis," *IEEE*, Aug. 2020, doi: 10.1109/cvpr42600.2020.00290.
- [23] "About the Django Software Foundation." [Online]. Available: <https://www.djangoproject.com/foundation/>



APPENDIX B

**Images of the data collection**

These images were taken in the study site during UAV data collection.



FIGURE B.1. Vineyard Photo



FIGURE B.3. UAV Shortly After Take-off



FIGURE B.2. UAV Landing and Take-off  
Area

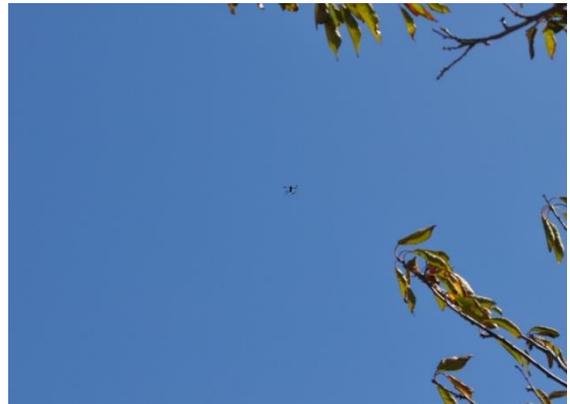


FIGURE B.4. UAV Mid-flight



FIGURE B.5. Measurements of Leaf Temperature to Establish CWSI Parameters

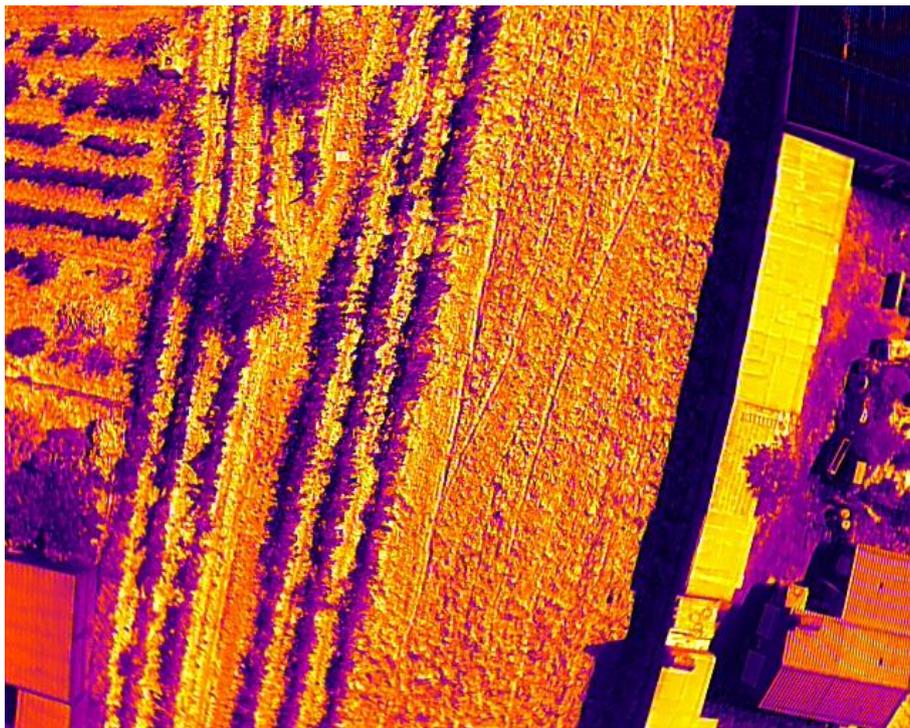


FIGURE B.6. UAV Captured Thermal Image