UNIVERSIDADE DE COIMBRA

Nuno Gonçalo da Costa Cunha

Multispectral Image segmentation in Agriculture using Deep Learning

Dissertação no âmbito do Mestrado em Engenharia Eletrotécnica e de Computadores, orientada pelo Professor Cristiano Premebida e apresentada ao Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Setembro de 2023

· U C ·

FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

Multispectral Image segmentation in Agriculture using Deep Learning

Nuno Gonçalo da Costa Cunha

September of 2023

Multispectral Image segmentation in Agriculture using Deep Learning

Supervisor:

Cristiano Premebida

Co-Supervisor:

Tiago Barros

Jury:

Paulo Jorge Carvalho Menezes

Vítor Manuel Mendes da Silva

Cristiano Premebida

Dissertation submitted in partial fulfillment for the degree of Master of Science in Electrical and Computer Engineering.

September of 2023

Acknowledgements

Com a conclusão desta dissertação, findo mais uma etapa da minha vida, a jornada académica. Em parte, o sucesso desta etapa deve-se a quem ao longo destes céleres cinco anos, mas também desde sempre, me acompanhou. Desta forma, quero expressar os meus agradecimentos.

Primeiramente, a quem me acompanhou neste trabalho, gostaria de agradecer aos meu orientadores Professor Cristiano Premebida e Tiago Barros pelas suas contribuições, que tornaram este trabalho mais substancial, e por me terem apoiado e motivado na realização do mesmo.

De seguida, aos meus amigos, àqueles em que as amizades se formaram no inicio deste curso, assim como àqueles de longa data, agradeço pelos momentos únicos partilhados e pela influência positiva que exercem na minha vida.

Por fim, mas com um carinho especial, aos meus pais, Carlos e Céu, e ao meu irmão João, agradeço pelo vosso apoio e amor incondicional.

Abstract

Multispectral imagery is frequently incorporated into agricultural tasks, providing valuable support for applications such as image segmentation, crop monitoring, field robotics, and yield estimation. From an image segmentation perspective, multispectral cameras can provide rich spectral information, helping with noise reduction and feature extraction. As such, this work concentrates on the use of data-combination (fusion) approaches to enhance the segmentation process in agricultural applications. More specifically, in this work, different fusion approaches are compared by combining RGB and NDVI as inputs for crop row detection, which can be useful for autonomous robots operating in the field. The inputs are used individually as well as combined at different times of the process (early and late fusion) to perform classical and deep learning (DL)-based semantic segmentation. In this work, two agriculture-related datasets are subjected to analysis using both DL-based and classical segmentation methodologies. The experiments reveal that classical segmentation methods, utilizing techniques such as edge detection and thresholding, can effectively compete with DL-based algorithms, particularly in tasks requiring precise foreground-background separation. This suggests that traditional methods retain their efficacy in certain specialized applications in the agricultural domain. Moreover, among the fusion strategies examined, late fusion emerged as the most robust approach, demonstrating superiority in adaptability and effectiveness across varying segmentation scenarios.

Keywords: Multispectral fusion, Semantic Segmentation, Deep Learning.

Resumo

As imagens multiespectrais são frequentemente incorporadas em tarefas agrícolas, fornecendo um apoio valioso para aplicações como segmentação de imagens, monitorização de culturas, robótica agrícola e estimativa de rendimento. Do ponto de vista da segmentação de imagens, as câmaras multiespectrais podem fornecer informações espectrais valiosas, ajudando na redução de ruído e extração de características. Como tal, este trabalho concentra-se na utilização de abordagens de combinação de dados (fusão) para melhorar o processo de segmentação em aplicações agrícolas.

Mais especificamente, neste trabalho são comparadas diferentes abordagens de fusão fazendo combinação de RGB e NDVI como entradas para a deteção de filas de culturas, o que pode ser útil para robôs autónomos que operam no campo. As entradas são utilizadas individualmente, assim como combinadas em diferentes momentos do processo (fusão precoce e fusão tardia) para realizar a segmentação semântica clássica e baseada em deep learning.

Neste trabalho, dois datasets relacionados com a agricultura são analisados com recurso a metodologias de segmentação baseadas em deep learning e clássicas. Os experimentos revelam que os métodos clássicos de segmentação, utilizando técnicas como deteção de bordas e thresholding, podem competir eficazmente com algoritmos baseados em deep learning, especialmente em tarefas que requerem uma separação precisa entre o primeiro plano e o plano de fundo. Isso sugere que métodos tradicionais mantêm a sua eficácia em aplicações especializadas no domínio agrícola. Além disso, entre as estratégias de fusão examinadas, a fusão tardia emergiu como a abordagem mais robusta, demonstrando superioridade em adaptabilidade e eficácia em cenários de segmentação variados.

Keywords: Fusão Multispectral, Segmentação Semântica, Deep Learning.

"Our greatest weakness lies in giving up. The most certain way to succeed is always to try just one more time.”

Thomas A. Edison

Contents

List of Acronyms

ASPP	Atrous Spatial Pyramid Pooling
AVs	Autonomous Vehicles
CNN	Convolutional Neural Network
DL	Deep Learning
FCN	Fully Convolutional Network
GNSS	Global Navigation Satellite Systems
INS	Inertial Navigation Systems
LiDAR	Light Detection and Ranging
MAV	Micro Aerial Vehicle
ML	Machine Learning
MS	Multispectral
NDVI	Normalized Difference Vegetation Index
NIR	Near-Infrared
RE	Red Edge
RGB	Red, Green, Blue
SVM	Support Vector Machine
VG	Vargem Grande

List of Figures

List of Tables

1 Introduction

1.1 Context and Motivation

Agriculture has a major global impact on human prosperity and sustainability, as approximately 40% of the world's ice-free land is already under agricultural production [4].

In the context of agriculture, the use of mobile robots has the main objective of developing technology-based solutions for supporting/executing labor-intensive, resource-demanding and time-consuming operations [5]. For example, robots can automate a variety of laborious tasks such as cultivation, inspection, spraying, pruning and harvesting [6]. A shortage of agricultural workers coupled with an aging farmer population and rising agriculture wages have prompted farmers and researchers to take an interest in the development of automation systems [6]. Moreover, interest in this application domains has been also encouraged by the decrease in equipment costs, the increase in computational power, and the growing interest in non-destructive food assessment methods [7].

Humans have the innate ability to understand and interpret visual scenes, for that reason is an effortless task for us. Providing such capabilities to robots has been an active research field for decades [8], which is known as robotic perception or simply perception. Robots equipped with such technology are able to function in increasingly complex and harsh environments. This is critical in agricultural environments, since they are challenging to operate in due to the constantly changing conditions and harsh terrain, which makes perception-related tasks such as navigation, object detection, semantic mapping, and plant recognition both difficult and essential.

Autonomous robots, in general, rely on the information captured by a collection of exteroceptive (*e.g.* cameras and LiDAR) and proprioceptive (*e.g.* encoders, IMUs) sensors. In particular, the navigation functions of agricultural robots rely mainly on sensors such as Global Navigation Satellite Systems (GNSS), Inertial Navigation Systems (INS), Light Detection and Ranging (LiDAR), and Red, Green, Blue (RGB) cameras [9]. These sensors (*e.g.* RTK-GPS or High-definition RGB cameras) have allowed much progress in agricultural machinery in terms of autonomous navigation, which were initially inspired by traditional robots/Autonomous Vehicles (AVs) operating in real-world scenarios. In agricultural environments, the robots have to operate in unstructured, harsh and rapidly changing environments dominated by vegetation, where, for instance, RGB cameras are less fitted due to the lack of the capability to capture important vegetation information such as the chlorophyll. This information can be sensed in other spectral bands such as Near-Infrared (NIR) and Red Edge (RE), which serve as vital indicators for assessing the plants' health status [10]. Additionally, within the context of this work, they play a important role in enhancing the differentiation between vegetation (crop) and soil (Fig. 1.1).

The different spectral bands can be captured by Multispectral (MS) sensors, where the MS images are captured at specific frequencies across the electromagnetic spectrum [11], which can provide a more detailed and comprehensive view of an object or scene when compared to an image captured at a singular wavelength.

A particular perception task used in agriculture for detection of weeds in crops [12], crop mapping [13] and disease detection [14] is image segmentation. This task involves classifying an image into distinct regions or segments. In particular, semantic segmentation, also known as pixel-level classification, is the process of clustering parts of an image together that belong to the same object class [15].

Recently, the rise of Deep Learning (DL) techniques has greatly promoted semantic segmentation research [16]. For example, the introduction of Fully Convolutional Network (FCN) [17] and encoder-decoder architectures has significantly increased the segmentation accuracy and paved the way for DL-based semantic segmentation. Compared to the traditional methods, the DL-based approaches have demonstrated remarkable effectiveness improvements [16].

Part of the work developed in this Dissertation has been utilized to contribute to a scientific paper, which has subsequently been accepted for presentation at the ROBOT2023: Sixth Iberian Robotics Conference [18].

1.2 Objectives

This dissertation aims to assessing the applicability of data-combination (a.k.a. data fusion) approaches using MS data for segmentation-related agricultural tasks. To support the experimental part and respective results, two distinct datasets have been employed, (i) a

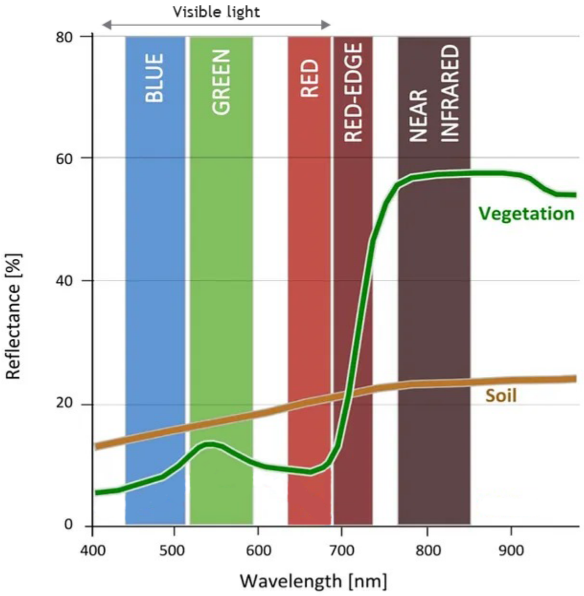


Figure 1.1: Reflectance of vegetation and soil across different wavelengths. Based on [1].

dataset for assessing where the drivable areas are within maize crops, so robots and/or agricultural machinery may keep the trajectory without damaging the crop, and (ii) a vineyard dataset captured via drone with the goal of vine identification [19]. Specifically, the intention is to leverage the rich and complementary information available in the various spectral bands to guaranty robust operation in agricultural environment conditions.
The specific objectives of this thesis are the following:

- Collecting and annotating a MS dataset obtained from maize crops using a robotic platform, annotated with crop row information;
- Carrying out a comparative study encompassing both deep learning (DL)-based and classical segmentation techniques;
- A thorough comparison between fusion and non-fusion methodologies, examining the impact of fusion in agricultural tasks;
- In-depth evaluation of two distinct fusion techniques: early fusion and late fusion.

1.3 Dissertation Outline

The structure and key content of each chapter in this dissertation are summarized as follows:

- **Chapter 1** (the current chapter) presents the context and motivation and the objectives for this dissertation.

- **Chapter 2** presents background information on image segmentation, multi-sensor multi-modality combination and DL fundamental concepts, and related work on DL semantic segmentation with different fusion approaches.

- **Chapter 3** provides an overview of the materials (datasets) employed and outlines the methodology utilized in this work.

- **Chapter 4** reviews the implementation details, results and discussion.

- **Chapter 5** presents conclusions from the results of the work developed and future work.

2 Background and Related Work

2.1 Image Segmentation

Image segmentation is a computer vision process that involves dividing an image into distinct segments, each representing a particular object or region. The problem can be addressed by classifying pixels with semantic labels (semantic segmentation), partitioning objects by instances (instance segmentation), or both (panoptic segmentation) [20] (See Fig. 2.1).

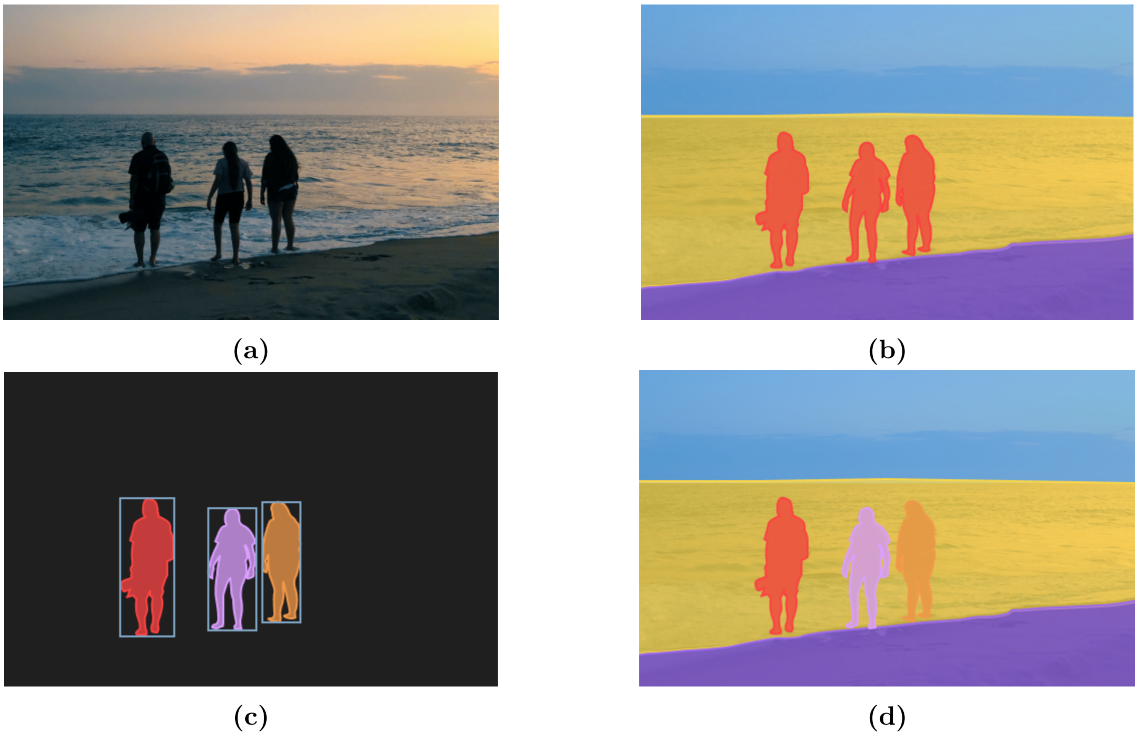
Given an image (Fig. 2.1a), semantic segmentation describes the task of classification of individual pixels with a set of classes, which is exemplified in Fig. 2.1b. Instance segmentation advances this concept by distinguishing individual object instances within the image, as demonstrated in Fig. 2.1c. Finally, panoptic segmentation takes on a more challenging task by merging semantic labeling and instance differentiation, as illustrated in Fig. 2.1d.

Due to the specific objective of solely isolating crops from the background, the semantic segmentation method serve as the key element of this work.

2.1.1 Classical Image Segmentation Methods

In the field of computer vision, traditional image segmentation techniques have been used for decades to extract meaningful information from images [21]. In these techniques, regions of an image with common characteristics, such as color, texture, or brightness, are identified using mathematical models and algorithms. Such techniques are usually computationally efficient and relatively simple to implement.

Although there are a variety of traditional image segmentation techniques, this section will focus on addressing three of the most common techniques, which were used in this work: Thresholding (Otsu's method), Edge-based Segmentation and Region-based Segmentation.



(a)

(b)

(c)

(d)

Figure 2.1: Example of the three types of image segmentation, where (a) represents the input image, while (b), (c), and (d) correspond to the semantic, instance, and panoptic segmentation masks, respectively. Source: [2].

Thresholding

Thresholding, the simplest image segmentation technique, utilizes a threshold value to divide pixels based on their intensity. This technique is particularly effective for segmenting objects with high intensity in contrast to other objects or backgrounds. The threshold can serve as a constant in low-noise images, but can also be dynamic in some cases. By dividing a grayscale image into two segments based on their intensity relationship, thresholding results in a binary image [22].

A problem with simple thresholding is that is necessary to manually specify the threshold value. One widely-used thresholding method that solves this problem is the Otsu's method, which automatically determines an optimal threshold based on the image's intensity distribution.

Edge-based segmentation

Edge-based segmentation is a technique for detecting the boundaries or edges of objects in images. As edges represent rapid changes in color or intensity, edge-based segmentation attempts to separate objects from their surrounding environment by identifying these transitions. An algorithm often used as a preprocessing step to locate and highlight edges is the

Canny edge detection.

The Canny edge detection algorithm was developed by John F. Canny in 1986 and is multi-stage process that can be described in 4 steps [23]:

1. Noise Reduction: This step involves removing noise from the image using a Gaussian filter. This will enhance the quality of edge detection by reducing the influence of small variations that may be mistaken for edges.

2. Intensity Gradient Calculation: The smoothed image is then filtered by applying Sobel filters in both the horizontal and vertical directions. The resulting gradient components, horizontal (G_x) and vertical (G_y), obtained from the Sobel filtering process, are needed to calculate the gradient magnitudes and angles. Their equations are described as follows:
$$\text{Edge Gradient } (G) = \sqrt{G_x^2 + G_y^2} \tag{2.1}$$
$$\text{Angle } (\theta) = \tan^{-1}\left(\frac{G_y}{G_x}\right) \tag{2.2}$$

3. Non-maximum Suppression: In this step, the goal is to thin out the edges by only keeping the pixels that are the most likely to be on an edge. This is achieved by considering the gradient magnitudes and angles calculated in the previous step. For each pixel, non-maximum suppression involves checking its neighbors along the gradient direction and preserving the pixel's value only if it's the local maximum in that direction.

4. Hysteresis Thresholding: The final step involves determining which edges to keep and which to discard based on thresholding. This is done using two threshold values: a high and a low threshold. Pixels with gradient magnitudes above the high threshold are definitely considered edges, while pixels with gradient magnitudes below the low threshold are discarded. Pixels with gradient magnitudes between the low and high thresholds are included as edges only if they are connected to pixels above the high threshold.

The Canny edge detection algorithm is going to be used later in this work for the edge-based segmentation as part of the so-called classical methods.

Region-based segmentation

The technique of region-based segmentation consists in dividing an image into regions based on similarity criteria, such as color, texture, or intensity. Unlike edge-based segmentation that focuses on detecting object boundaries, region-based segmentation aims to group pixels or regions with similar characteristics. This technique is particularly useful when objects of interest have consistent characteristics throughout their regions. Regions can be formed either by grouping pixels into regions (region growing algorithm) or by subdividing a single region successively (split and merge algorithm) [24].

One powerful method within region-based segmentation is the watershed segmentation technique. The watershed transform is a traditional method within gray-scale mathematical morphology used for image segmentation [25].

It establishes an analogy with geographical watersheds [25], which separate areas drained by different river systems. In image analysis, the watershed transform treats an image as a topographic surface, where high-intensity regions represent peaks and low-intensity areas correspond to valleys. The process starts by "filling" isolated valleys (local intensity minima) with distinct-colored "water" (labels). As water levels rise, valleys with nearby peaks gradually merge, similar to filling basins. To prevent excessive merging, barriers form where water meets. These barriers correspond to the boundaries that separate the image's drainage basins.

This method naturally identifies object and region boundaries. However, due to noise or irregularities, it may lead to over-segmentation (generating too many segments). To mitigate this, a marker-based watershed algorithm can be employed, making the process interactive.

In marker-based watershed, specific regions are marked as foreground, background, or uncertain using labels. In this approach, the segmentation results are improved by incorporating the user's knowledge. This type of marker-based watershed is the technique used in this work for region-based segmentation.

2.2 Multi-sensor multi-modality combination

Image fusion refers to the process of combining two or more images by integrating the information present in each of the individual images. The outcome is an image-representation that has more valuable, or complementary, information than the single source images individually. The goal of this process is to assess the data at each pixel location in the input images and preserve the data from that image that best represents the true scene content or enhances the usefulness of the fused image for a particular application [26].

As pointed out in the previous chapter, the use of only RGB cameras is not sufficient to capture essential vegetation information, such as chlorophyll absorption, in agricultural contexts. MS imagery achieves this thanks to their spectral bands outside the visible spectrum, providing a superior scene understanding. Some notable applications of MS images in agriculture include plant disease detection, fruit maturity, and crop production analysis [11]. To make a better use of these images, fusion methods can be applied, which encompass, usually, three steps: (i) understanding which modalities should be fused, (ii) determining the appropriate method for fusing the information, and (iii) specifying where the information should be fused along the network [27][28].

Focusing on 'where' the information is fused, there are three common stages, (i) the early fusion, (ii) the middle fusion, and (iii) the late fusion. Early fusion consists of combining (merging) the data at the input layer. Early fusion is more straightforward in the case of MS fusion, as the inputs are of the same type (images). When dealing with different modalities, the data must be integrated at the middle or later stages of the process, due to the differences between the data types [29]. Middle or intermediate fusion, a concept employed within network-style learning frameworks, involves the combination of feature maps from different modalities at an intermediate layer of the model architecture. The most common method, yet simplistic, for early and middle fusion is the concatenation. Late fusion, also referred to as decision-level fusion, consists of training features separately for each modality and merging them at later layers. A simplified visualization of these three fusion approaches is presented in Fig. 2.2, where a multimodal semantic segmentation framework is used to illustrate the three fusion approaches, using RGB and Normalized Difference Vegetation Index (NDVI) representations as inputs.

Despite being a image-fusion of one sensor, MS image combination uses the same techniques as multimodal fusion. While multimodal fusion in autonomous navigation usually combines images and other types of sensors, such as LiDAR or radar, MS fusion specifically refers to the fusion of images captured at different frequencies across the electromagnetic spectrum.

Fusion techniques have demonstrated their value in outdoor settings. In their work, [30] employed fusion techniques combining multispectral and multimodal data, providing evidence that the fused outcomes, specially late fusion, surpass segmentation based solely on RGB data under challenging outdoor conditions. In the same study, they also noted that segmentation using individual spectra yielded the best results with RGB data compared to the other modalities they experimented with, such as NIR.

In this thesis, the approaches chosen to carry out multi-spectral combination (multi-channels w.r.t. CNN framework) are:

- Early fusion: Achieved by by merging preprocessed inputs through concatenation before feeding them into the network.

- Late fusion: Achieved by computing the pixel-wise weighted sum of the class likelihoods of each model before the final class decision.

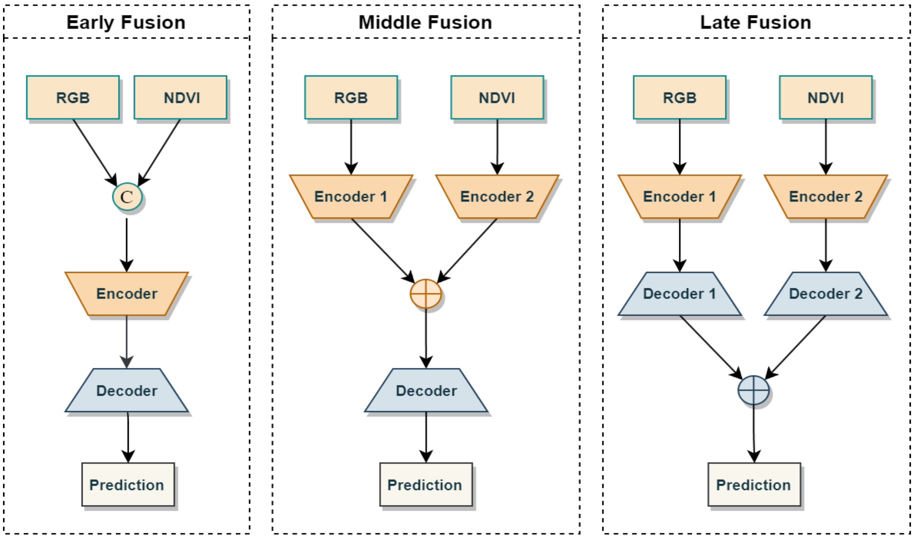


Figure 2.2: Simplified approach of early, middle and late fusion, using RGB and NDVI as inputs.

2.3 Deep Learning - A Short Review

Deep learning is a subcategory of Machine Learning (ML) that aims to model high-level abstractions of data using multiple layers of neurons consisting of complex structures or non-linear transformations [31].

In deep learning, but also in traditional machine learning, two learning approaches are fundamental in how models are trained: supervised and unsupervised learning. In this dissertation, supervised learning techniques were utilized for the task of semantic segmentation with DL methods. Additionally, it's worth noting that, in this work, the classical approaches are categorized as unsupervised segmentation methods in a broader sense, as they do not incorporate a sophisticated learning process.

Supervised Learning

In the supervised learning approach, models are trained using ground truth data, which, in the context of semantic segmentation, corresponds to masks with pixel-wise labels. It is provided to the model the input data paired with corresponding outputs [32]. This training data is commonly labeled by a data scientist during the preparatory phase, prior to its utilization for training and evaluating the model. At end of the training phase, it is expected that the model will have acquired a substantial understanding of the relationship between inputs and outputs, then it may be able to classify unknown datasets and predict their outcomes [33].

The main advantage of this technique is the ability to collect data or generate data output from prior knowledge. However, the disadvantage of this approach arises when the training set lacks samples that should belong to a particular class, potentially causing the decision boundary to become overstrained. In general, this technique is simpler than other techniques in the way of learning while still achieving high performance [32].

2.3.1 Convolutional Neural Networks (CNNs)

CNN stands out as one of the most popular and utilized DL networks [34]. CNN is a type of artificial neural network designed specifically for processing structured grid data, such as an RGB image made up of three 2D arrays representing pixel intensities across the three color channels [35]. These networks are capable of performing tasks such as segmentation, classification, and detection.

On a review of DL concepts, [32] pointed out three benefits of these networks:

1. The primary advantage of CNNs is their weight sharing property, which reduces the number of trainable parameters, thereby contributing to improved generalization and preventing overfitting.

2. Simultaneously training the feature extraction layers and the classification layer leads to a model output that is not only well-structured but also heavily reliant on the extracted features.

3. Implementing large-scale networks is notably simpler using CNNs compared to other neural network architectures.

The essential elements of a CNN are convolutional, pooling and fully connected layers (see Fig. 2.3).

For image classification, the goal is to transform the spatial tensor obtained from convolutional layers into a fixed-length vector. To achieve this, fully connected layers are utilized. As a result of this process, spatial information is lost. In semantic segmentation tasks, spatial information preservation is essential. Various architectural approaches have been developed

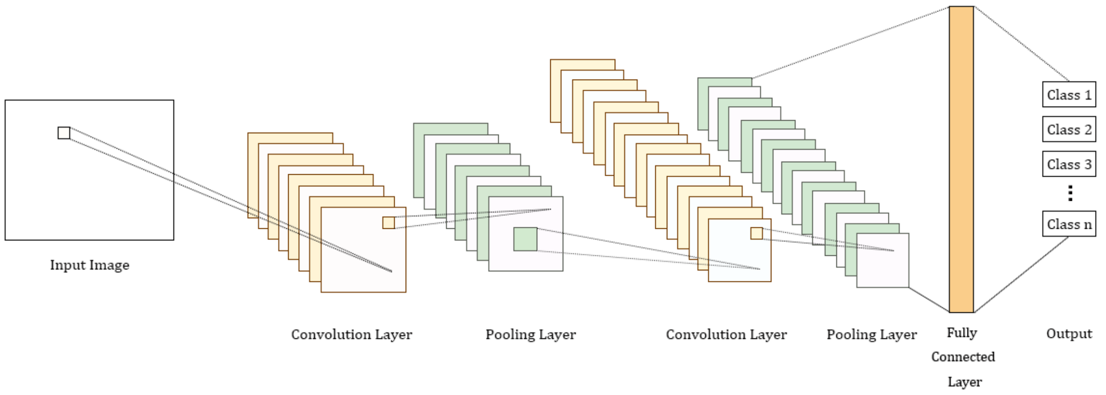


Figure 2.3: Schematic diagram of a basic CNN.

to address this requirement. FCN, exemplified by models like DeepLab, and encoder-decoder architectures, such as SegNet, are two types of networks usually employed for these tasks.

Convolutional Layer

Convolutional layers are the backbone of CNNs. The convolutional layer employs convolutional filters, also known as kernels. These kernels operate through a process called convolution, in which they slide across the input image's pixels. At each position, the kernel's elements are multiplied with the corresponding input values, and the results are summed to generate a single value in the output feature map. This operation is performed repeatedly as the kernel goes through the input image, resulting in a feature map that highlights specific patterns or features. Initially, the kernel starts with random values, and throughout the training process the kernel weights suffer fine-tuning, facilitating the extraction of the desired features [32]. Figure 2.4 provides a visual representation of this operation.

These layers are typically used in sequence with each other. In the initial convolutional layers, low-level features are extracted, whereas higher-level features are captured in the subsequent convolutional layers.

Several parameters can influence the convolution process, such as kernel size, stride, and padding. The kernel size refers to the dimensions of the filter, typically represented as a matrix. The choice of kernel size impacts the scale of features the filter can detect. Larger kernels are able to capture broader features, while smaller kernels focus on finer details.

The stride parameter determines how the kernel moves as it convolves across the input image. A larger stride skips more pixels, resulting in down-sampled feature maps and reduced computational load.

Padding involves adding additional pixels around the input image, often with zero values, before convolutions are performed. Padding helps to prevent details from being lost at the edges of the image.

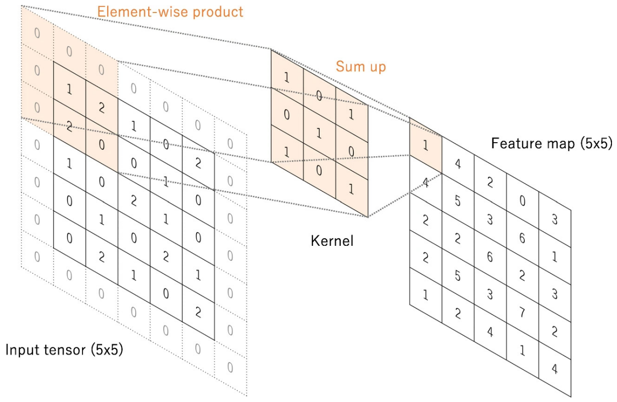


Figure 2.4: Convolution operation with zero padding, with a kernel size and a stride of 3×3 and 1, respectively. Source: [3].

Pooling Layer

Pooling layers aim to progressively reduce the dimensionality of the representation, reducing both parameter count and the model's computational complexity [36]. Among the various pooling approaches, one of the most widely employed is max-pooling. Max-pooling involves selecting the maximum value from a set of values within a specific region of a feature map, and uses it to create a down-sampled (pooled) feature map (see Fig. 2.5). This operation highlights the most dominant feature present in that region.

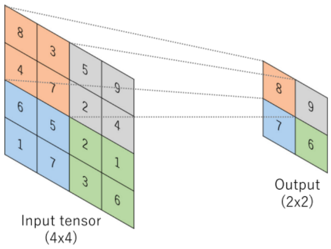


Figure 2.5: Max-pooling with 2×2 filter. Source: [3].

Non-linearity Layer

The Non-linearity Layers, also known as the activation function layers, are employed after all layers with weights (also called learnable layers), such as convolutional layers. It is designed to introduce non-linearities into the network, which allows it to capture and model complex relationships in data that cannot be captured through linear transformations alone.

Two of the most common used activation functions are ReLU (Rectified Linear Unit) and Sigmoid. The Sigmoid function takes any real value as input and outputs values in the range of 0 to 1. It is commonly used for models in classification and segmentation tasks, where it is necessary to compute the likelihood of the input belonging to a given class. Since probability of anything exists only between the range of 0 and 1, sigmoid is the right choice because of its range. The sigmoid function is expressed as follows:

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.3}$$

ReLU is a popular activation function that replaces all negative input values with zero and leaves positive values unchanged. The advantages of this characteristic include speeding up network training, which results in faster convergence of gradient descent [37]. Mathematically, the ReLU function can be expressed as in equation 2.4.

$$f(x) = \begin{cases} x & \text{if } x > 0 \\ 0 & otherwise \end{cases} \tag{2.4}$$

In Figure 2.6, the plots of the ReLU and Sigmoid functions are represented.

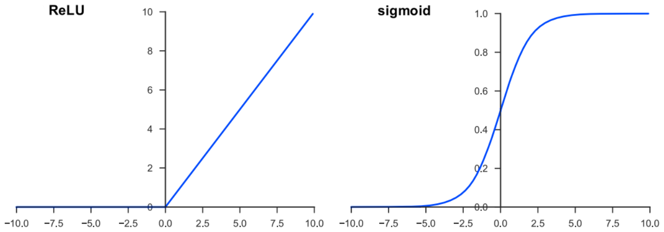


Figure 2.6: Activation functions: Rectified linear unit (ReLU) and Sigmoid [3].

Moreover, it is relevant to highlight that the output layer of a network, often an activation function, must be carefully selected based on the task at hand. This functions used for the final layer often differ from those applied to preceding layers. The function need to be chosen according to the task in hands. For tasks involving multiclass classification, a common choice is the softmax, given that this function normalizes the raw output scores from the last layer into a set of target class probabilities, ensuring that values lie between 0 and 1 while collectively summing up to 1. Additionally, for tasks involving binary classification, another common choice is the sigmoid activation function, previously introduced in this section [3].

Backpropagation

Backpropagation is an algorithm used during the train of artificial neural networks. During the training process, the network modifies its internal parameters, known as weights and biases, to learn the relationships within the processed data. This adjustment is made by backpropagation.

In practice, input data enters the network during the forward pass, generating an output prediction [38]. This predicted output is then compared against the actual target output, resulting in a numerical value named "loss". The loss quantifies how far off the network's predictions are from the true values, and the objective of backpropagation is to minimize this loss by recalibrating the network's parameters. This process involves two main steps:

- Backward Pass (Backpropagation): This step involves calculating the gradients of the loss based on each parameter (weight and bias) in the network. These gradients, which indicate each parameter's impact on the error, are propagated backward through the network's layers, enabling the network to determine how to adjust its parameters to reduce the loss.

- Optimization: Once the gradients are known, an optimization algorithm is employed to update the network's parameters in a way that reduces the loss. This optimization process is repeatedly applied over multiple iterations (epochs), making the network gradually converge towards a set of parameter values that result in minimal loss. While gradient descent optimization is commonly used, there are other optimization algorithms available, such as Adam [39], RMSProp [40], Adagrad [41], etc.

2.3.2 Encoder-decoder

An extensively used architecture for semantic segmentation is the encoder-decoder (represented in Fig. 2.7), which has to main modules: an encoder and an decoder. The encoder uses convolutional and pooling layers to map the input image to a fixed-size representation, often called a latent space representation. The decoder maps the latent space representation to an output space, which, in segmentation tasks, corresponds to a mask with pixel-wise labels.

The down-sampling layers form the encoder, while the up-sampling layers form the decoder. In summary, the encoder outputs a tensor encompassing object information, shape, and size, while the decoder utilizes such tensor-representation to generate segmentation maps.

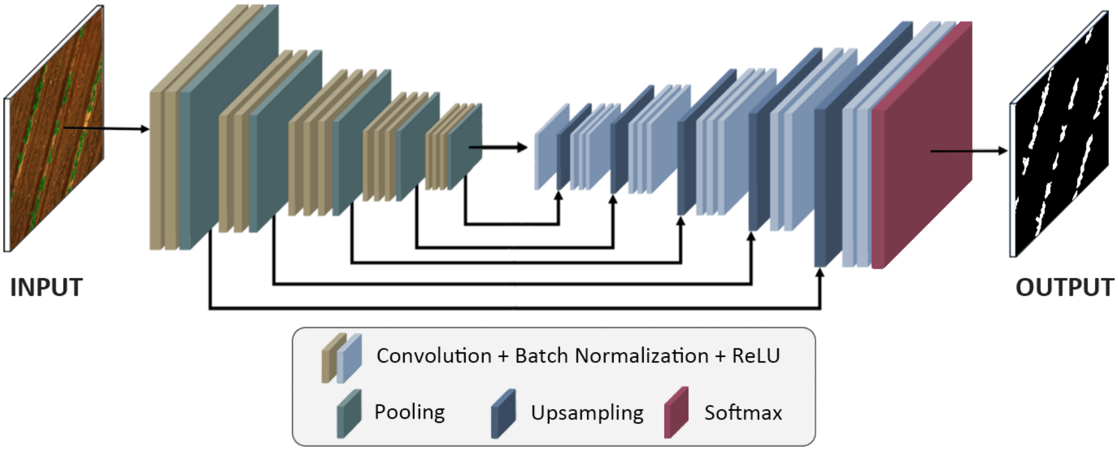


Figure 2.7: Encoder-Decoder architecture.

SegNet

SegNet is an Encoder-Decoder-based architecture specifically designed for semantic image segmentation tasks. It was introduced by Badrinarayanan et al. [42] in their work. The key innovation of SegNet lies in its architecture that allows for pixel-wise semantic segmentation while preserving spatial information through an encoder-decoder structure.

In SegNet, the decoder part, in addition to using up-sampling layers, incorporates a specialized operation called "max-pooling indices". These operations are similar to skip connections in the way they contribute to preserve important information from the encoder component. Specifically, these pooling indices represent the positions of the maximum values that were recorded during the corresponding max-pooling operations in the encoder. The up-sampling layers then use these indices to place the pooled values back in their original locations, effectively reconstructing the high-resolution feature map.

In the Fig. 2.7 is possible to see a SegNet architecture, where the arrows originating from the encoder and pointing towards the decoder represent the pooling indices.

2.3.3 Fully Convolutional Networks (FCNs)

The concept of FCNs was introduced by Long et al. in their work [17]. An FCN includes only convolutional layers, allowing it to generate a segmentation map of the same size as the input image [20].

FCNs are primarily designed for semantic segmentation tasks and, due to their ability to make accurate pixel-level predictions, FCNs are used in a variety of fields, such as medical image analysis [43], autonomous driving [44].

DeepLab

DeepLab is a FCN-based segmentation architecture tailored for semantic image segmentation, proposed by Chen et al. [45]. One of the main contributions of the network is its use of atrous (dilated) convolutions.

The dilated convolution expands the receptive field size, without adding more parameters, by introducing gaps in the kernel. Figure 2.8 illustrates an Atrous convolution with a 3×3 dilated kernel, where the dilation rate (r) is set to 2. This configuration illustrates how atrous convolutions effectively increase the receptive field while preserving a compact kernel size.

One of the notable variants of DeepLab is DeepLabV3 [46]. Apart from using Atrous Convolution, this variant uses an improved Atrous Spatial Pyramid Pooling (ASPP) module, which was presented in its predecessor, DeepLabV2. The enhanced ASPP module in DeepLabV3 is effective at capturing features across multiple spatial scales, enabling the model to manage objects of diverse sizes, improving the performance of the network.

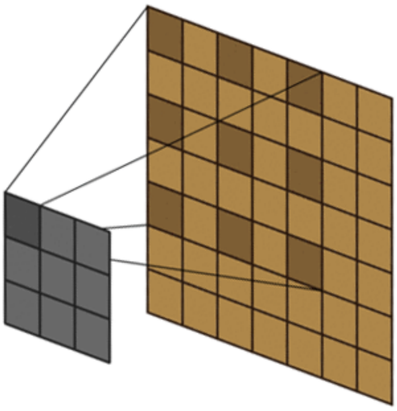


Figure 2.8: Atrous convolution.

2.4 Semantic Segmentation on MS Combination

Before the DL era, semantic segmentation relied primarily on handcrafted features and classifiers such as SVM [47] and Random Forest, and clustering techniques such as K-means [48], which were used to used to group similar pixels together. These methods had the advantage of simplicity and low computational cost. However, nowadays CNNs have revolutionized the field in recent years and are now the most effective technique in pattern recognition-based applications [49]. One of the strongest advantages of using DL in image processing is the reduced need of feature engineering, thanks to the ability to automatically extract features from raw data, with features at higher levels of the hierarchy being formed by the extraction of features from lower levels [35].

Agricultural-related work on segmentation relies heavily on DL networks. Common deep learning algorithms are FCN [17], SegNet [42], U-Net [50], DeepLab [51], PSPNet [52].

In [53], RGB images were used as input to test the U-Net under various scenarios, including shadows, weeds, gaps in the crop row, intense sunlight conditions, and different growth stages. On the other hand, authors in [54] compared the DeeplabV3+ model with UNet using both RGB and MS images. They employed transfer learning techniques to extract wheat lodging area. Authors in [19] compared different encoder-decoder architectures and demonstrated that SegNet, U-Net and ModSegNet achieved similar results for vineyard segmentation. Authors in [55] conducted semantic segmentation of tree-like plants. They employed an asynchronous training approach, training two separate networks on RGB and depth data, which was encoded as a 3-channel HHA (depth, height from the ground, and angle of the surface normal with gravity) image. These networks were later combined using a late fusion architecture. Authors in [56] employed SegNet for the dense semantic classification of weeds using multispectral images acquired via a Micro Aerial Vehicle (MAV). To accommodate multiple inputs, they customized the network by incorporating concatenation, thus implementing a form of early fusion. In a different study from [57], using a combination of RGB and NIR data they made a sunflower planting areas segmentation across a range of deep learning architectures, including Support Vector Machine (SVM), FCN, SegNet, and a novel SegNet variant.

It is clear from the existing work that crop row detection has progressed over time and there are a number of advantages that deep learning based systems have over classical approaches in semantic segmentation. In Table 2.1 a summary of all the articles mentioned above is presented.

Article	Bands	Architecture	Fusion	Application
[53]	RGB	Encoder-Decoder (UNet)	Not Performed	Crop Row Detection
[58]	RGB + RGN	DeepLabV3+/Encoder-Decoder (UNet)	Not Performed	Extraction of wheat lodging area
[19]	RGB + RE + NIR	Encoder-Decoder (SegNet/ U-Net/ModSegNet)	Early	Vineyard segmentation
[55]	RGB + Depth	HHA-Net/Encoder-Decoder(SegNet)	Early/Late	Segmentation of Tree-like Vegetation
[56]	NIR + Red + NDVI	Encoder-Decoder (SegNet)	Early	Weed detection
[57]	RGB + NIR	SVM/FCN/Encoder-Decoder (SegNet)	Early	Sunflower planting areas segmentation

Table 2.1: Related work for semantic segmentation in agriculture.

3 Materials and Methods

This section outlines the methods, tools, and processes employed to conduct the experiments in this work. Firstly, the segmentation problem is formulated in generic terms, then a comprehensive characterization of the study sites is provided, along with the technical details of the recorded maize data. In addition, the necessary geometric calibration is also described. Finally, the segmentation problem is formulated in a multispectral fusion context by focusing, specifically, on early and late fusion techniques from two distinct information sources.

3.1 Problem Formulation

Image segmentation involves the task of dividing an image into regions, or objects, based on their shared characteristics. Image segmentation can be defined as a function that maps an input image to a class likelihood mask. Thus, let I represent the input image, defined as a three-dimensional array $I = [p_{ijk}]_{h \times w \times b}$, where $p_{ijk} \in [0, ..., 255]$ denotes the pixel intensity at coordinates (i, j, k). The image dimensions are given by h (height), w (width), and b (number of spectral bands), with $i \in [1, h]$, $j \in [1, w]$, and $k \in [1, b]$. To perform image segmentation, we aim to obtain a class likelihood mask Q, represented by $Q = [q_{ijk}]_{h \times w \times c}$. Here, $q_{ijk} \in [0, 1]$ indicates the likelihood of the pixel at coordinates (i, j, k) belonging to each of the $C = \{1, ..., c\}$ segmentation classes, constrained by $\sum_{k=1}^{c} q_{ijk} = 1$.

In the specific context of this study, we focus on binary segmentation. This means that only one target-class is considered, resulting in a single-channel likelihood matrix $Q = [q_{ij}]_{h \times w \times 1}$. Hence, the final segmentation mask with a class per pixel $M = [m_{ij}]_{h \times w} \in \{0, 1\}$, is obtained through a threshold-based approach:

$$m_{ij} = \begin{cases} 1 & \text{if } q_{ij} \geq T \\ 0 & \text{if } q_{ij} < T \end{cases} \tag{3.1}$$

where T is a threshold value chosen to distinguish between the positive and negative classes in the segmentation.

The binary segmentation framework is used to compare classical methods with deep learning (DL)-based approaches using two input modalities: RGB (I^{RGB}) and NDVI (I^N). The RGB image I^{RGB} is defined as a tree-dimensional array $I^{RGB} = [p_{ijk}^{RGB}]_{h\times w\times 3}$, capturing the visible spectrum (400-700 nm) with the Red, Green, and Blue bands. On the other hand, the NDVI image I^N is a two-dimensional array $I^N = [p_{ij}^N]_{h\times w}$, representing the Normalized Difference Vegetation Index. The NDVI is calculated as:

$$I^N = \frac{NIR - Red}{NIR + Red}\,, \tag{3.2}$$

where Red and NIR correspond to specific spectral bands. The Red band lies within the visible spectrum, while the NIR band extends beyond the visible range (700 to 1100 nm). These bands are particularly valuable for agricultural monitoring, capturing the absorption of chlorophyll in visible light and its reflection in the NIR spectrum.

3.2 Study Site and Materials

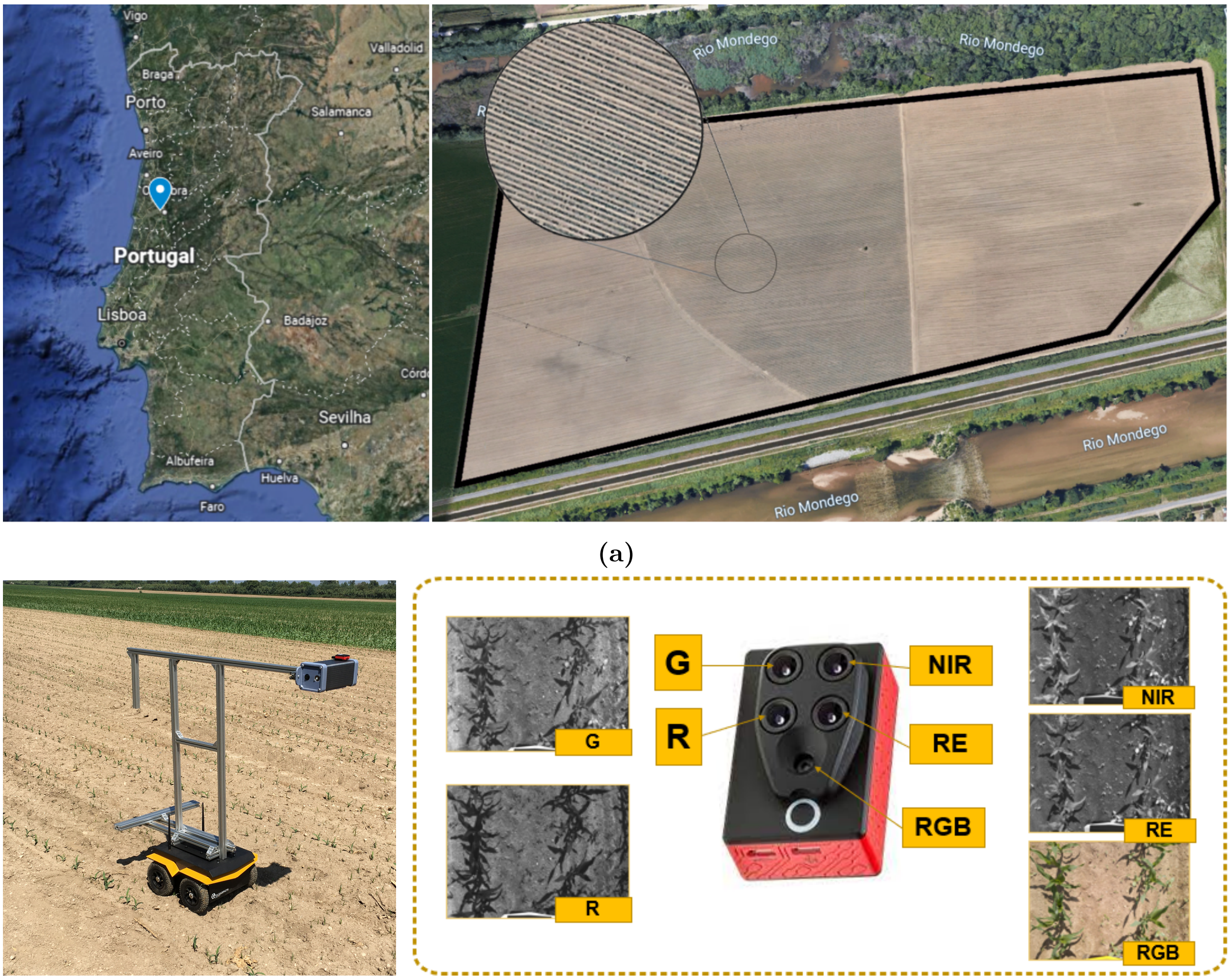
The study was conducted using data collected from a maize crop known as Vargem Grande (VG) located in the Coimbra region, in the center of mainland Portugal (see Fig. 3.1a). The data was collected in July 2022, specifically during the early growth stage of the plants. To ensure optimal lighting conditions and minimize shadow interference, the data was collected around midday under sunny weather conditions.

The multispectral dataset was captured using a Parrot Sequoia multispectral camera[1]. This camera consists of four monochrome sensors (Green, Red, Red Edge, and Near Infrared) along with an RGB sensor (see Fig. 3.1c). To facilitate the data collection process, the camera was mounted on a mobile platform known as the Jackal from Clearpath[2]. The camera was positioned 1.2 meters above the ground, with the sensors facing downward (see Fig. 3.1b).

To gather the data, the robot was teleoperated in-between the crop rows. Images from all five sensors were captured every two seconds, ensuring a comprehensive dataset for analysis.

[1] Parrot Sequoia User Guide

[2] Jackal Homepage



(a)

(b)

(c)

Figure 3.1: Study site and material used to record the dataset, where (a) illustrates the studied maize crop denominated Vargem Grande, (b) is the recording setup with which the dataset was recorded, and (c) is the multispectral sensor with its five sensors.

3.2.1 Geometric Calibration

Geometric calibrations were required before data labeling could proceed. In addition to the small physical offset between monochrome and RGB sensors, the RGB camera had a different resolution and focal length from the other bands, as shown in Table 3.1.

First, the RGB camera's resolution was lowered to match that of the other images in the dataset, and went from 4068×3456 to 1280×960. After this initial step, a radial distortion correction was applied to the different bands images to rectify any disparities in the focal length compared to the RGB camera. The RGB image served as the reference, guiding the adjustments made to the other images to ensure their alignment with the same parameters as the RGB image. This correction was performed using the Brown-Conrady model, which accounts for both tangential and radial distortion in an image. This model uses the following equation:

Sensors	Band: Center wavelength (width)	Resolution	Focal Length	HFoV	VFoV
	[nm]	[px]	[mm]	[º]	[º]
Monochrome	G: 550(40); R: 660(40); RE: 735(10); NIR: 790(40)	1280 × 960	3.98	62	49
RGB	R, G, B	4068 × 3456	4.88	64	50

Table 3.1: Specifications of the sensor. Field of View (FoV)

$$\begin{aligned} x_u &= x + \rho_x + \omega_x \\ y_u &= x + \rho_y + \omega_y \end{aligned} \tag{3.3}$$

$$\begin{aligned} \rho_x &= \underbrace{(x - x_c) \cdot \Big(k_1 \cdot r^2 + k_2 \cdot r^4 + \cdots\Big)}_{\text{radial terms}} \\ \rho_y &= \underbrace{(y - y_c) \cdot \Big(k_1 \cdot r^2 + k_2 \cdot r^4 + \cdots\Big)}_{\text{radial terms}} \end{aligned} \tag{3.4}$$

$$\begin{aligned} \omega_x &= \underbrace{\Big[P_1 \cdot \Big(r^2 + 2 \cdot (x - x_c)^2\Big) + 2 \cdot P_2 \cdot (x - x_c) \cdot (y - y_c)\Big] \cdot \Big(1 + P_3 \cdot r^2 + \cdots\Big)}_{\text{tangential terms}} \\ \omega_y &= \underbrace{\Big[2 \cdot p_1 \cdot (x - x_c) \cdot (y - y_c) + p_2 \cdot \Big(r^2 + 2 \cdot (y - y_c)^2\Big)\Big] \cdot \Big(1 + p_3 \cdot r^2 + \cdots\Big)}_{\text{tangential terms}} \quad , \end{aligned} \tag{3.5}$$

where (x, y) are distorted image points, (x_u, y_u) are the corresponding undistorted image points, (x_c, y_c) is the center of distortion, k_i is the i^{th} radial distortion coefficient, P_j is the j^{th} tangential distortion coefficient, and $r = \sqrt{(x - x_c)^2 + (y - y_c)^2}$.

Finally, after all images had been corrected, a labeling process was conducted on all the images that met a quality criterion, making a total of 532 annotations. The VG dataset was made specifically for maize field navigation. In this context, the annotations are focused on classifying the rows, while the remaining areas are considered as background. An example of this annotation is present in Fig. 3.2.

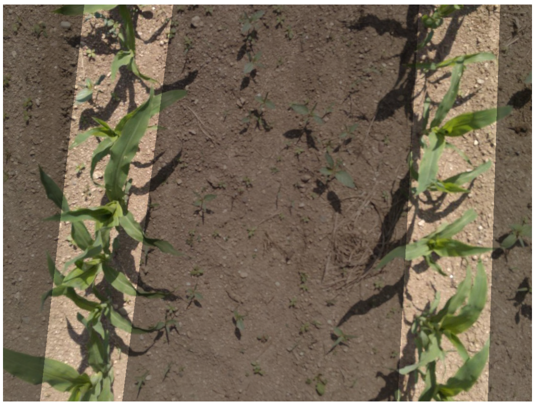


Figure 3.2: Overlay of an image from the VG dataset with its corresponding mask.

3.3 Image Fusion

Fusion, in the context of image segmentation, refers to the integration of information derived from diverse sources into a unified representation. The fusion process can be applied at various stages, depending on the segmentation methods employed [30].

3.3.1 Early Fusion

In the context of image processing, early fusion involves the merging of information at the input level, specifically within the pixel space. In this work, early fusion is employed using two different approaches: classical segmentation methods and DL-based segmentation methods.

In the comparison between classical and DL-based methods, the representation of early fusion varies depending on the approach used. Specifically, when employing classical approaches, the RGB image I^{RGB} is transformed into a grayscale representation denoted as $I^{Gr} = [p_{ij}^{Gr}]_{h\times w}$. This conversion is achieved using the standard formula:

$$p_{ij}^{Gr} = 0.299\, p_{ij}^{R} + 0.587\, p_{ij}^{G} + 0.114\, p_{ij}^{B} \; , \tag{3.6}$$

where p_{ij}^{R}, p_{ij}^{G}, and p_{ij}^{B} represent the pixel intensities of the Red, Green, and Blue bands at the coordinate (i, j), respectively, with $i \in [1, h]$ and $j \in [1, w]$. The resulting grayscale image I^{Gr} has dimensions given by $h \times w$.

For classical approaches, the fused representation I^{Ec} is obtained by computing the pixel-wise mean between the NDVI image I^{N} and the grayscale image I^{Gr}:

$$p_{ij}^{Ec} = \frac{p_{ij}^{N} + p_{ij}^{Gr}}{2} \; , \tag{3.7}$$

here, p_{ij}^{N} and p_{ij}^{Gr} represent the pixel intensities of the NDVI and grayscale images at the (i, j) coordinate, respectively. The resulting fused representation I^{Ec} is an image of dimensions $h \times w$. On the other hand, when employing DL-based segmentation methods, the fused representation I^{Ed} is obtained by channel-wise concatenation of the RGB image I^{RGB} and the NDVI image I^{N}. This is represented as:

$$I^{Ed} = [I^{RGB}, I^{N}] = \left[p_{ijk}^{RGB} \,|\, p_{ijk}^{N}\right]_{h \times w \times 4} \tag{3.8}$$

where p_{ijk}^{RGB} and p_{ijk}^{N} represent the pixel intensities of the RGB and NDVI images at the (i, j, k) coordinate, respectively. The resulting fused representation I^{Ed} is a tensor with dimensions $h \times w \times 4$, where the first three channels correspond to the RGB image and the fourth channel corresponds to the NDVI image. Figure 3.3 provides a visual representation of this approach in a DL architecture.

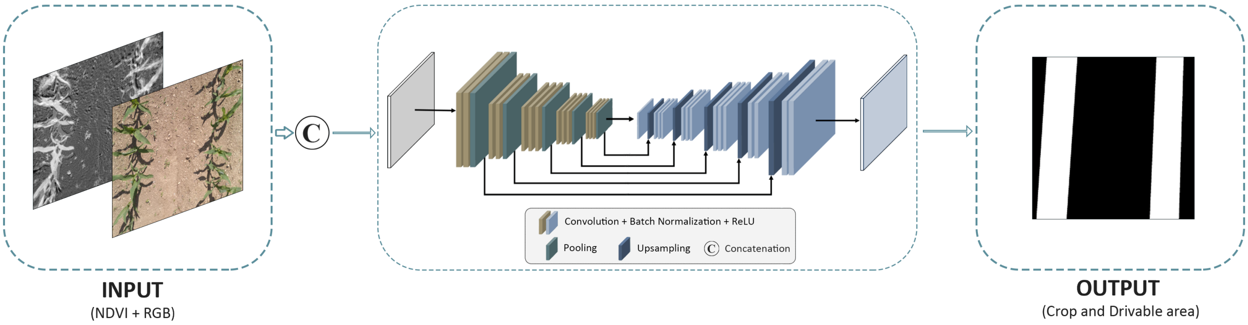


Figure 3.3: Deep learning architecture employing an early fusion approach, where the NDVI and RGB are concatenated as inputs.

3.3.2 Late Fusion

Early fusion involves merging information at the input space, while late fusion performs the merging at the output space. In this study, late fusion is achieved by computing the pixel-wise weighted sum of the class likelihoods of each model before the final class decision.

In the context of a late fusion framework, the segmentation process involves two input images: I^{N} and I^{RGB}. Each image is individually processed through a segmentation model, generating respective output likelihood masks: $Q^{N} = [q_{ij}^{N}]_{h \times w \times 1}$ and $Q^{RGB} = [q_{ij}^{RGB}]_{h \times w \times 1}$, where, q_{ij}^{N} and $q_{ij}^{RGB} \in [0, 1]$ represent the likelihood of the positive class at the pixel coordinates (i, j).

The fused representation is obtained by computing a pixel-wise weighted sum of the likelihoods from both segmentation models. Hence, the fused likelihood q_{ij}^{L} at the pixel coordinates (i,j) is calculated using the following formula:

$$q_{ij}^{L} = \alpha \cdot q_{ij}^{N} + \beta \cdot q_{ij}^{RGB} \ , \tag{3.9}$$

where α and β are weights that can be adjusted to balance the contribution of each likelihood according to the models' performance. By controlling the values of α and β, the fusion process can be fine-tuned to achieve optimal segmentation results based on the strengths of the individual models.

For a more concise summary of the methods employed in this dissertation, Figure 3.4 illustrates a simplified approach to both early and late fusion.

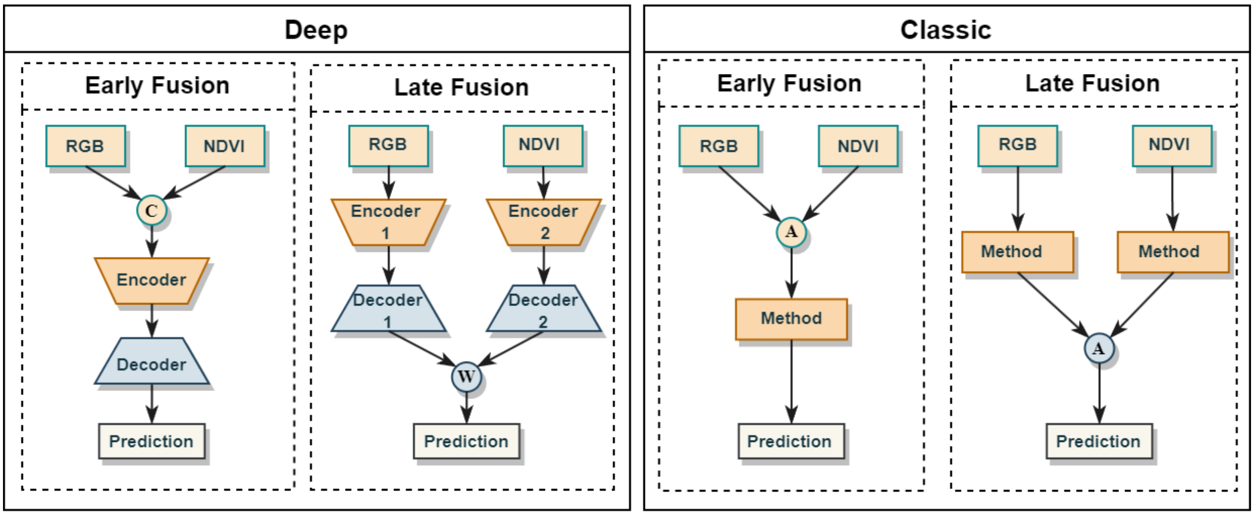


Figure 3.4: Simplified approach of early and late fusion using RGB and NDVI as inputs on deep (encoder-decoder architecture) and classic methods.

4 Experimental Evaluation and Discussion

This section provides a comprehensive assessment of early and late fusion techniques within a multispectral image segmentation framework applied to the AgRA domain. The section outlines the datasets used for evaluation, describes the implementation details and evaluation metrics employed, and presents a thorough discussion of the quantitative and qualitative results obtained.

4.1 Datasets

The proposed approaches undergo evaluation using primarily the maize crop dataset (referred to as VG) described in the section 3.2. Complementary, a dataset collected from vineyards are used to assess cross-domain generalization capability. For the VG dataset, a total of 532 images were recorded for each of the five sensors (R, G, RE, NIR, and RGB). The images were aligned and cropped to a final size of 1100 × 825, and for evaluation purposes, they were resized to 240 × 240. The dataset was then split into an 80/20 ratio for training and testing, respectively. Regarding the vineyard data, the dataset encompasses images of 240 × 240 from three distinct vineyards. The evaluation follows the approach proposed in [19], employing a cross-validation method that involves training on data from two vineyards and testing on the third. Relevant information about the datasets can be found in Table 4.1.

Dataset	Vargem Grande	Qta Baixo	ESAC	Valdoeiro
Sample Size (Train/Test)	532 (425/107)	150	189	120
Bands	R, G, RE, NIR, RGB	B, G, R, RE, NIR, Thermal		
Dimensions Fusion	1100×825	240×240		

Table 4.1: Dataset information, where B,G,R,RE and NIR represent Blue, Green, Red, Red-edge and Near-infrared, respectively.

4.2 Implementation Details

This section outlines the implementation details of both the classical and DL-based segmentation approaches. Python was chosen as the programming language for both classical and DL approaches.

Classical Approach

Three classical segmentation methods were employed: Otsu's thresholding[1], edge-based[2], and region-based[2] techniques.

For Otsu's thresholding, was used a opencv *threshold* function with an automatic threshold value on a grayscale image to perform the segmentation. This threshold value was determined using Otsu's algorithm, which finds the threshold (t) that minimizes the weighted within-class variance, as expressed by the following equation:

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t) \tag{4.1}$$

where, $q_i(t)$ is the probability of class i pixels at threshold t, with $i \in \{0, 1\}$ representing discrete class labels. $\sigma_j^2(t)$ is the variance of class j pixels at threshold t, where j can be either 0 (background) or 1 (foreground).

In the case of edge-based segmentation, to detect the edges of the objects was used the Canny edge detector from *scikit-image* library, with *Sigma* (standard deviation of the Gaussian filter) value as default. After this first step, to fill the contours was applied, the *binary_fill_holes* function from the *SciPy* library. Finally, a process was applied to remove small objects from the segmented image. To achieve this, the *remove_small_objects* function from the *scikit-image* library was employed. Specifically, objects smaller than a specified minimum size were filtered out. In this case, a minimum size of 21 was chosen, meaning that objects with fewer than 21 connected pixels were removed from the segmented image.

Lastly, a region-based segmentation was performed. Initially, an elevation map was created by applying the Sobel gradient to the grayscale images. Following this step, markers were assigned for background and plants based on gray value histograms. This assignment process involved trial and error to achieve the best results. Finally, the watershed transform,

[1] OpenCV Image Thresholding - Otsu's thresholding.

[2] Edge-based and Region-based segmentation - Canny edge-detector and Watershed transform.

as detailed in section 2.1.1, was applied to fill regions of the elevation map with these markers.

Deep Learning Approaches

In this work, two distinct DL-based segmentation models were utilized: SegNet[3] and DeepLabV3[4]. SegNet employs an encoder-decoder architecture, where the input is gradually encoded to a latent space and then gradually decoded to an output mask. In contrast, DeepLabV3 upsamples the latent representation in fewer steps.

Both models were implemented using the PyTorch [59], which is a ML and DL framework based on the Torch library. They were executed on a hardware setup consisting of an NVIDIA GEFORCE GTX 3090 GPU and an AMD Ryzen 9 5900X CPU with 64 GB of RAM. The training process utilized the AdamW optimizer [60] with a learning rate of 1e-3 for VG and approximately 1e-4 and 1e-5 for Vineyard models. The Binary Cross-Entropy with Logits Loss (BCEWithLogitsLoss) function was employed to calculate the loss, and the outputs (logits) were passed through a sigmoid activation function to obtain the final probabilities.

4.2.1 Evaluation Metrics

The performance of the segmentation methods was evaluated using several metrics, including pixel accuracy (acc), F_1 score, and Intersection over Union (IoU). These metrics provide insights into the accuracy and quality of the segmentation results.

The pixel accuracy of a segmentation map is measured by the proportion of correctly classified pixels. Even though accuracy is straightforward, it might not be optimal for imbalanced datasets in which one class dominates. This is where the F1 score excels. The F1 score balances precision (correctly predicted positive pixels) and recall (actual positive pixels correctly identified), resulting in a single value that considers both false positives and false negatives.

IoU, also known as the Jaccard index, quantifies the overlap between predicted and ground truth segmentation masks. It calculates the ratio of the intersection of these masks to their union. For semantic segmentation tasks, it is a common metric used to evaluate spatial accuracy.

The pixel accuracy is defined as:

$$\text{Acc} = \frac{TP + TN}{TP + FP + TN + FN}, \tag{4.2}$$

[3] SegNet GitHub Implementation

[4] DeepLabV3 Pytorch

where TP, TN, FP, and FN represent True Positives, True Negatives, False Positives, and False Negatives, respectively. The F_1 score is calculated as:

$$F_1 \text{ score} = \frac{2 \times TP}{2 \times TP + FP + FN}. \tag{4.3}$$

The IoU is computed as:

$$IoU = \frac{\text{Area of Intersection}}{\text{Area of Union}} , \tag{4.4}$$

where *Area of Intersection* refers to the number of overlapping pixels between the predicted mask and ground truth mask: $A \cap B = \{p_{ij} : p_{ij} \in A \text{ and } p_{ij} \in B\}$, where p_{ij} denotes a pixel at coordinate (i, j), while A and B represent the ground truth mask and the predicted mask, respectively. The *Area of Union* represents the total number of pixels encompassed by both prediction and ground truth masks, including the overlapping region: $A \cup B = \{p_{ij} : p_{ij} \in A \text{ or } p_{ij} \in B\}$, where p_{ij} denotes a pixel at coordinate (i, j), while A and B represent the ground truth mask and the predicted mask, respectively.

4.3 Results and Discussion

This section presents the experimental results for both classical and DL-based segmentation methods, comprising both quantitative and qualitative assessments. The qualitative results are organized in Table 4.2, while the visual representations of the segmentation masks are illustrated in Fig 4.1. Each segmentation approach is evaluated in four distinct methods: first, with the RGB and NDVI modalities individually, followed by the modalities fused using early and late fusion techniques, as described in Section 3.3. The results were obtained with the segmentation threshold $T = 0.5$, for the late fusion results, both models were given an equal contribution: *i.e.* $\alpha = 0.5$ and $\beta = 0.5$.

Classical vs DL-based

In this work, classical unsupervised and supervised DL-based segmentation methods were employed. The classical methods demonstrate to perform well on tasks where the primary objective is to separate foreground from background, as is the case of the Vineyard dataset, where the goal is to segment individual plants. In such case, unsupervised approaches are competitive with DL-based approaches, offering the advantage of simplicity and lower complexity. However, in segmentation tasks that involve identifying spatial regions, containing both foreground and background, such as the Maize dataset, where the objective is to detect

		Maize			Vineyard											
Method	Dataset	VG			Qta. Baixo			ESAC			Valdoeiro			Average		
		Acc.	F1	IoU	Acc.	F1	IoU	Acc.	F1	IoU	Acc.	F1	IoU	Acc.	F1	IoU
OTS	RGB	74.4	28.3	16.7	57.6	41.9	27.6	79.1	52.7	40.2	74.5	38.6	26.0	71.4	40.4	27.6
	NDVI	76.1	32.3	19.6	84.1	61.7	46.1	65.6	49.7	34.8	92.4	67.8	56.0	79.6	52.9	39.1
	Early F.	75.6	34.1	20.9	71.6	52.7	37.7	71.4	55.5	41.2	89.5	65.3	52.7	77.1	51.9	38.1
	Late F.	67.5	33.7	20.8	83.0	67.5	44.7	89.5	66.8	42.9	91.6	81.6	56.7	**82.7**	**62.4**	**41.3**
Edge-b.	RGB	76.6	12.9	6.9	69.8	15.5	8.5	78.1	26.3	15.4	86.9	22.7	13.1	77.9	19.4	10.5
	NDVI	75.6	13.0	7.0	70.3	16.8	9.3	61.1	18.7	10.4	94.2	48.6	33.7	75.3	24.3	**15.1**
	Early F.	84.1	21.0	11.9	76.8	16.3	8.9	65.4	14.8	8.0	93.9	39.3	25.8	**80.1**	22.9	13.7
	Late F.	70.1	14.0	7.6	64.8	18.9	10.6	71.1	25.5	14.8	87.6	39.1	25.1	73.5	**24.4**	14.5
Region-b.	RGB	84.1	21.0	11.9	78.3	47.4	33.1	78.9	44.3	33.2	85.3	44.6	31.2	81.7	39.3	27.4
	NDVI	82.2	12.4	6.7	89.0	67.9	52.5	76.0	50.9	37.3	97.2	76.3	63.8	86.1	51.9	40.1
	Early F.	76.7	19.0	10.5	81.3	52.7	37.0	87.6	63.5	49.8	97.3	77.6	65.2	85.8	53.2	**40.6**
	Late F.	81.8	25.3	14.7	93.1	69.3	34.9	92.1	48.9	24.6	98.6	82.7	45.2	**91.4**	**56.6**	29.9
SegNet	RGB	96.2	87.1	78.5	84.9	52.1	35.6	73.1	41.9	27.7	92.1	58.0	41.3	86.6	59.8	45.8
	NDVI	95.6	85.3	76.1	85.9	64.4	48.4	78.5	51.4	36.8	93.9	67.1	51.7	**88.5**	**67.1**	**53.3**
	Early F.	96.8	89.3	80.8	81.1	42.1	26.7	81.4	50.5	33.9	94.3	61.0	43.9	88.4	60.7	46.3
	Late F.	96.1	86.9	78.6	86.2	56.4	40.4	75.9	46.9	33.0	93.4	61.4	45.7	87.9	62.9	49.4
DeeplabV3	RGB	96.5	87.9	79.8	81.8	35.6	21.8	82.0	45.0	30.1	91.0	56.2	39.7	87.9	56.1	42.9
	NDVI	95.9	86.0	77.3	87.3	59.2	42.2	77.7	33.8	21.4	89.1	44.2	28.8	87.5	55.8	42.4
	Early F.	97.3	89.2	81.2	82.1	31.0	18.7	79.9	37.3	23.8	89.9	52.8	36.4	87.3	52.6	40.0
	Late F.	96.6	87.7	80.2	85.3	47.5	32.0	81.0	39.0	25.9	92.5	58.0	42.3	**88.9**	**58.1**	**45.1**

Table 4.2: Segmentation performance on the Maize (VG) and Vineyard (Qta. Baixo, ESAC, and Valdoeiro) datasets, employing classical approaches such as Otsu Threshold (OST), Edge-based, and Region-based, as well as DL-based approaches including SegNet and DeeplabV3. Each method is evaluated with four scores: RGB and NDVI individually, and both modalities fused using early and late fusion techniques. The performance scores are presented in percentage [%], with the **best score** highlighted in bold and the second-best scores underlined.

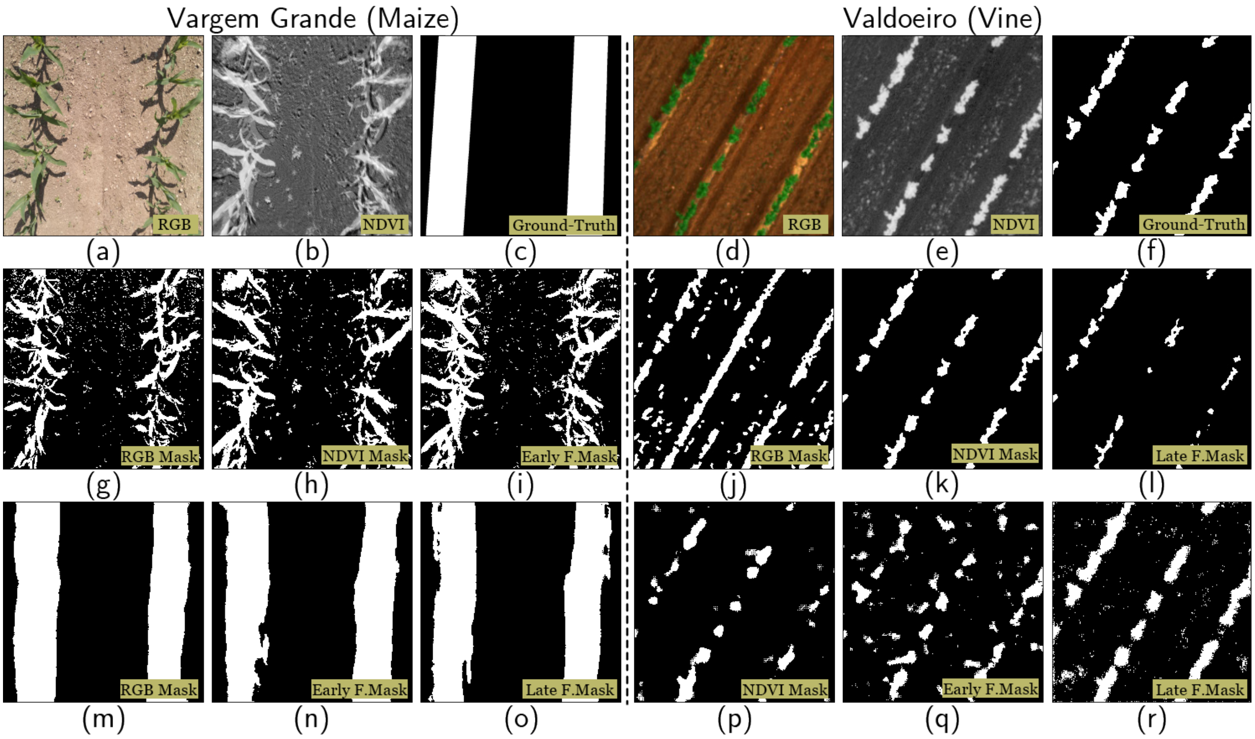


Figure 4.1: Qualitative segmentation results of both VG and vineyard dataset. The images (a) to (f) (top row), represent respectively the RGB, NDVI and ground-truth masks. Images (g) to (l) (middle row) represent segmentation masks generated by classical approaches. And finally, images (m) to (r) (bottom row) represent segmentation masks generated by SegNet. More specifically, images (g) to (i) were generated by Otsu, while images (j) to (i) were generated with a region-based method.

the plant rows, supervised DL-based approaches show a clear advantage due to their ability to learn spatial information. The results obtained in our experiments consistently confirm this, as depicted in Table 4.2 and Fig 4.1.

Fusion vs No-Fusion

The results consistently show that late fusion either achieves the best performance or ranks a close second, distinctly outperforming early fusion. This superiority means that, on average, extracting features from individual modalities first and then fusing them at a later stage yields better results compared to one model from both modalities combined.

Upon analyzing the average results, it becomes evident that late fusion capitalizes on the model with the highest performance. By averaging the outputs of both models, late fusion is able to reduce the noise associated with the lesser-performing model. However, this method also has a downside: valuable information from the best-performing model may be diluted or lost. Thus, while late fusion leverages the strengths of both models to enhance overall robustness, finding the right balance in the contributions of each model becomes crucial. One potential approach to achieve this balance is to weight the contributions based on their respective performance. Investigating this weighted fusion strategy offers an interesting avenue for future work.

MS vs RGB

To compare MS and RGB imagery, we can based on the results summarized in Table 4.2, with the representative of MS data being the NDVI. The results clearly indicate that NDVI outperformed RGB in most of the classic segmentation methods, showcasing its effectiveness in capturing relevant information for vegetation-related tasks. This observation aligns with our expectations, given that NDVI is specifically designed to highlight vegetation and its health.

A closer examination of the results reveals that NDVI not only outperformed RGB in classic methods but also demonstrated strong performance in DL-based segmentation methods, such as Segnet.

In the case of DeeplabV3, NDVI achieved results that were nearly on par with RGB. This result is intriguing and suggests that for this particular DL architecture, both NDVI and RGB can be equally effective.

Runtime Analysis

In terms of computational performance, DL methods demand a considerable amount of time to execute due to the intensive computations involved. In our case, the maximum runtime reached approximately twenty-five minutes for the entire training process, specifically during late fusion, where the batch size supported by the hardware was limited to 32 (VG) and 16 (Vineyard). In contrast, classical methods demonstrate the opposite behavior, being significantly faster and achieving results within a minute.

5 Conclusions

5.1 Final Dissertation Remarks

This work aimed to study the impact of fusion (combining) approaches of multispectral data in segmentation tasks applied domains related to digital-precision agriculture and agricultural robotics. The study was conducted on both classical and DL-based segmentation methods, where the experimental part is supported by two datasets: a dataset of vineyards and a dataset of maize crops, recorded and curated specifically for this study. In addition, there was an intention to conduct comparisons between fusion and non-fusion methodologies, as well as to evaluate both DL and traditional segmentation techniques.

The experimental findings show two principal observations: First, classical segmentation methods, utilizing techniques like thresholding and edge detection, are competitive against DL-based approaches in tasks requiring foreground-background separation. This highlights their continued applicability in specialized scenarios. Second, late fusion, where individual modalities are processed and then fused, emerges as the most robust approach, demonstrating its superior adaptability across various experimental conditions. These insights offer valuable guidance for both current applications and future research in segmentation algorithms.

5.2 Future Work

Regarding future work, there are some potential changes to improve the current research, which are listed below:

- Investigate different fusion methods, such as middle fusion, and compare their outcomes to the existing results.

- Incorporate additional spectral bands as inputs, such as the RE, to improve the current results.

6 Bibliography

[1] Farmonaut Support. Applications of satellite imagery bands - part 2: Vegetataion red edge (b5, b6, b7, b8a), Jan 2019. URL `https://medium.com/@farmonaut/applications-of-satellite-imagery-bands-part-2-vegetataion-red-edge-b5-b6-b7-b8a-2a`

[2] Hmrishav Bandyopadhyay. Image segmentation: Deep learning vs traditional [guide], Aug 2021. URL `https://www.v7labs.com/blog/image-segmentation-guide`.

[3] Rikiya Yamashita, Mizuho Nishio, Richard Kinh Gian Do, and Kaori Togashi. Convolutional neural networks: an overview and application in radiology. *Insights into imaging*, 9:611–629, 2018.

[4] K Lorenz and R Lal. Environmental impact of organic agriculture. *Advances in agronomy*, 139:99–152, 2016.

[5] Stavros G. Vougioukas. Agricultural robotics. *Annual Review of Control, Robotics, and Autonomous Systems*, 2(1):365–392, 2019. doi: 10.1146/annurev-control-053018-023617. URL `https://doi.org/10.1146/annurev-control-053018-023617`.

[6] Mohd Saiful Azimi Mahmud, Mohamad Shukri Zainal Abidin, Abioye Abiodun Emmanuel, and Hameedah Sahib Hasan. Robotics and automation in agriculture: present and future applications. *Applications of Modelling and Simulation*, 4:130–140, 2020.

[7] Shveta Mahajan, Amitava Das, and Harish Kumar Sardana. Image acquisition techniques for assessment of legume quality. *Trends in Food Science & Technology*, 42(2): 116–133, 2015. ISSN 0924-2244. doi: https://doi.org/10.1016/j.tifs.2015.01.001. URL `https://www.sciencedirect.com/science/article/pii/S0924224415000023`.

[8] Richard Szeliski. *Computer vision: algorithms and applications*. Springer Nature, 2022.

[9] Man Zhang, Yuhan Ji, Shichao Li, et al. Research progress of agricultural machinery navigation technology [j/ol]. *Transactions of the Chinese Society for Agricultural Machinery*, 51(4):1–18, 2020.

[10] Josep Peñuelas and Iolanda Filella. Visible and near-infrared reflectance techniques for diagnosing plant physiological status. *Trends in plant science*, 3(4):151–156, 1998.

[11] Syed Muslim Jameel, Abdul Rehman Gilal, Syed Sajjad Hussain Rizvi, Mobashar Rehman, and Manzoor Ahmed Hashmani. Practical implications and challenges of multispectral image analysis. In *2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET)*, pages 1–5. IEEE, 2020.

[12] Jie You, Wei Liu, and Joonwhoan Lee. A dnn-based semantic segmentation for detecting weed and crop. *Computers and Electronics in Agriculture*, 178:105750, 2020.

[13] Zhenrong Du, Jianyu Yang, Cong Ou, and Tingting Zhang. Smallholder crop area mapped with a semantic segmentation deep learning method. *Remote Sensing*, 11(7): 888, 2019.

[14] Liangxiu Han, Muhammad Salman Haleem, and Moray Taylor. A novel computer vision-based approach to automatic detection and severity assessment of crop diseases. In *2015 Science and Information Conference (SAI)*, pages 638–644. IEEE, 2015.

[15] Martin Thoma. A survey of semantic segmentation. *arXiv preprint arXiv:1602.06541*, 2016.

[16] Shijie Hao, Yuan Zhou, and Yanrong Guo. A brief survey on semantic segmentation with deep learning. *Neurocomputing*, 406:302–321, 2020.

[17] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[18] Nuno Cunha, Tiago Barros, Mário Reis, Tiago Marta, Cristiano Premebida, and Urbano J Nunes. Multispectral image segmentation in agriculture: A comprehensive study on fusion approaches. *ROBOT2023: Sixth Iberian Robotics Conference*, 2023.

[19] T. Barros, P. Conde, G. Gonçalves, C. Premebida, M. Monteiro, C.S.S. Ferreira, and U.J. Nunes. Multispectral vineyard segmentation: A deep learning comparison study.

Computers and Electronics in Agriculture, 195:106782, 2022. ISSN 0168-1699. doi: https://doi.org/10.1016/j.compag.2022.106782. URL https://www.sciencedirect.com/science/article/pii/S0168169922000990.

[20] Shervin Minaee, Yuri Y Boykov, Fatih Porikli, Antonio J Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2021.

[21] X Cufi, X Munoz, J Freixenet, and J Marti. A review of image segmentation techniques integrating region and boundary information. *Advances in imaging and electron physics*, 120:1–39, 2003.

[22] B Basavaprasad and S Hegadi Ravindra. A survey on traditional and graph theoretical techniques for image segmentation. *Int. J. Comput. Appl*, 975:8887, 2014.

[23] Canny edge detection, 2013. URL https://docs.opencv.org/3.4/da/d22/tutorial_py_canny.html.

[24] Allan Hanbury. Image segmentation by region based and watershed algorithms. *Wiley Encyclopedia of Computer Science and Engineering*, pages 1543–1552, 2007.

[25] SV Kasmir Raja, A SHAIK ABDUL KHADIR, and SS Riaz Ahamed. Moving toward region-based image segmentation techniques: A study. *Journal of Theoretical & Applied Information Technology*, 5(1), 2009.

[26] Pushkar Pradham, Nicolas H Younan, and Roger L King. *Concepts of image fusion in remote sensing applications.* Academic, 2008.

[27] Di Feng, Christian Haase-Schütz, Lars Rosenbaum, Heinz Hertlein, Claudius Gläser, Fabian Timm, Werner Wiesbeck, and Klaus Dietmayer. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(3):1341–1360, 2021.

[28] Yifei Zhang, Désiré Sidibé, Olivier Morel, and Fabrice Mériaudeau. Deep multimodal fusion for semantic image segmentation: A survey. *Image and Vision Computing*, 105: 104042, 2021. ISSN 0262-8856.

[29] David L Hall and James Llinas. An introduction to multisensor data fusion. *Proceedings of the IEEE*, 85(1):6–23, 1997.

[30] Abhinav Valada, Gabriel Oliveira, Thomas Brox, and Wolfram Burgard. Towards robust semantic segmentation using deep fusion. In *Robotics: Science and systems (RSS 2016) workshop, are the sceptics right*, 2016.

[31] Xing Hao, Guigang Zhang, and Shang Ma. Deep learning. *International Journal of Semantic Computing*, 10(03):417–439, 2016.

[32] Laith Alzubaidi, Jinglan Zhang, Amjad J Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, José Santamaría, Mohammed A Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions. *Journal of big Data*, 8:1–74, 2021.

[33] Seldon. Supervised vs unsupervised learning explained, Jun 2023. URL `https://www.seldon.io/supervised-vs-unsupervised-learning-explained`.

[34] Guangle Yao, Tao Lei, and Jiandan Zhong. A review of convolutional-neural-network-based action recognition. *Pattern Recognition Letters*, 118:14–22, 2019.

[35] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553): 436–444, 2015.

[36] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

[37] Neena Aloysius and M Geetha. A review on deep convolutional neural networks. In *2017 international conference on communication and signal processing (ICCSP)*, pages 0588–0592. IEEE, 2017.

[38] Barry J Wythoff. Backpropagation neural networks: a tutorial. *Chemometrics and Intelligent Laboratory Systems*, 18(2):115–155, 1993.

[39] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[40] Geoffrey Hinton, Nitish Srivastava, and Kevin Swersky. Neural networks for machine learning lecture 6a overview of mini-batch gradient descent. *Cited on*, 14(8):2, 2012.

[41] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 07 2011.

[42] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[43] Xiaoqing Liu, Kunlun Gao, Bo Liu, Chengwei Pan, Kongming Liang, Lifeng Yan, Jiechao Ma, Fujin He, Shu Zhang, Siyuan Pan, et al. Advances in deep learning-based medical image analysis. *Health Data Science*, 2021, 2021.

[44] Çağrı Kaymak and Ayşegül Uçar. A brief survey and an application of semantic image segmentation for autonomous driving. *Handbook of Deep Learning Applications*, pages 161–200, 2019.

[45] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[46] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[47] Ashfaq Hussain and Ajay Khunteta. Semantic segmentation of brain tumor from mri images and svm classification using glcm features. In *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, pages 38–43, 2020. doi: 10.1109/ICIRCA48905.2020.9183385.

[48] Mohamed A Hamada, Yeleussiz Kanat, and Adejor Egahi Abiche. Multi-spectral image segmentation based on the k-means clustering. *Int. J. Innov. Technol. Explor. Eng*, 9: 1016–1019, 2019.

[49] Sue Han Lee, Chee Seng Chan, Paul Wilkin, and Paolo Remagnino. Deep-plant: Plant identification with convolutional neural networks. In *2015 IEEE international conference on image processing (ICIP)*, pages 452–456. IEEE, 2015.

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

[51] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous con-

volution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[52] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6230–6239, 2017. doi: 10.1109/CVPR.2017.660.

[53] Rajitha de Silva, Grzegorz Cielniak, and Junfeng Gao. Towards agricultural autonomy: crop row detection under varying field conditions using deep learning. *arXiv preprint arXiv:2109.08247*, 2021.

[54] Dongyan Zhang, Yang Ding, Pengfei Chen, Xiangqian Zhang, Zhenggao Pan, and Dong Liang. Automatic extraction of wheat lodging area based on transfer learning method and deeplabv3+ network. *Computers and Electronics in Agriculture*, 179:105845, 2020. ISSN 0168-1699. doi: https://doi.org/10.1016/j.compag.2020.105845. URL `https://www.sciencedirect.com/science/article/pii/S0168169920307341`.

[55] S Tejaswi Digumarti, Lukas Maximilian Schmid, Giuseppe Maria Rizzi, Juan Nieto, Roland Siegwart, Paul Beardsley, and Cesar Cadena. An approach for semantic segmentation of tree-like vegetation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1801–1807. IEEE, 2019.

[56] Inkyu Sa, Zetao Chen, Marija Popović, Raghav Khanna, Frank Liebisch, Juan Nieto, and Roland Siegwart. weednet: Dense semantic weed classification using multispectral images and mav for smart farming. *IEEE robotics and automation letters*, 3(1):588–595, 2017.

[57] Zhishuang Song, Zhitao Zhang, Shuqin Yang, Dianyuan Ding, and Jifeng Ning. Identifying sunflower lodging based on image fusion and deep semantic segmentation with uav remote sensing imaging. *Computers and Electronics in Agriculture*, 179:105812, 2020.

[58] Dongyan Zhang, Yang Ding, Pengfei Chen, Xiangqian Zhang, Zhenggao Pan, and Dong Liang. Automatic extraction of wheat lodging area based on transfer learning method and deeplabv3+ network. *Computers and Electronics in Agriculture*, 179:105845, 2020.

[59] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[60] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Appendix A

Scientific Paper - ROBOT2023: Sixth Iberian Robotics Conference

Multispectral Image Segmentation in Agriculture: A Comprehensive Study on Fusion Approaches

Nuno Cunha, Tiago Barros, Mário Reis, Tiago Marta, Cristiano Premebida, and Urbano J. Nunes

University of Coimbra, Institute of Systems and Robotics (ISR), Department of Electrical and Computer Engineering (DEEC), Coimbra - Portugal.
{nuno.cunha, tiagobarros, mario.reis, tiago.marta, cpremebida, urbano}@isr.uc.pt

Abstract. Multispectral imagery is frequently incorporated into agricultural tasks, providing valuable support for applications such as image segmentation, crop monitoring, field robotics, and yield estimation. From an image segmentation perspective, multispectral cameras can provide rich spectral information, helping with noise reduction and feature extraction. As such, this paper concentrates on the use of fusion approaches to enhance the segmentation process in agricultural applications. More specifically, in this work, we compare different fusion approaches by combining RGB and NDVI as inputs for crop row detection, which can be useful in autonomous robots operating in the field. The inputs are used individually as well as combined at different times of the process (early and late fusion) to perform classical and DL-based semantic segmentation. In this study, two agriculture-related datasets are subjected to analysis using both deep learning (DL)-based and classical segmentation methodologies. The experiments reveal that classical segmentation methods, utilizing techniques such as edge detection and thresholding, can effectively compete with DL-based algorithms, particularly in tasks requiring precise foreground-background separation. This suggests that traditional methods retain their efficacy in certain specialized applications within the agricultural domain. Moreover, among the fusion strategies examined, late fusion emerges as the most robust approach, demonstrating superiority in adaptability and effectiveness across varying segmentation scenarios. The dataset and code is available at https://github.com/Cybonic/MISAgriculture.git

1 INTRODUCTION

In agriculture, autonomous robots are becoming increasingly popular because of the potential benefits they may have on food security, sustainability, resource-use efficiency, reduction of chemical treatments, and optimization of human effort and yield [14]. Alongside this trend, the utilization of multispectral imagery in agricultural applications, including AgRA (Agricultural Robotics and Automation), has become increasingly significant in recent years. Some notable applications of these images include plant disease detection, fruit maturity, and crop production analysis [5].

Certain bands, captured at specific frequencies across the electromagnetic spectrum, have the ability to reveal distinct information about plants. Among these bands, the near-infrared (NIR) band holds significance in agricultural tasks (e.g., assessing crop health) as it can effectively highlight chlorophyll absorption and water content in plants. One widely used index that relies on the NIR band is the Normalized Difference Vegetation Index (NDVI), which provides a quantitative measure of vegetation greenness and density. Compared with RGB-only data, incorporating this additional spectral information can enhance the discrimination of different objects and features within images. This enables more accurate identification and classification of crops, improving the process of image segmentation [19].

This work focuses on assessing the applicability of fusion approaches using multispectral data for segmentation-related agricultural tasks. Specifically, we investigate two fusion approaches: early fusion and late fusion. Early fusion involves combining the information from multiple sources at the input level before the segmentation process. This means that the data from different sources are merged into a single representation prior to segmentation. On the other hand, late fusion occurs after the segmentation process has been applied to each individual image. The segmentations are obtained independently, and then ”fused,” or combined, at a later stage. By exploring both early and late fusion techniques, we aim to assess their impact on image segmentation performance and determine which fusion approach yields superior results for the specific objectives of this work.

Through a comprehensive comparative analysis, the aim of this work is to make significant progress in automatic crop-row detection by studying early and late fusion of multispectral data using classical and DL-based segmentation approaches. To accomplish this, this paper brings two key contributions:

- A curated multispectral dataset collected on maize crops using a robotic platform, with crop row annotations;
- An extensive comparison study conducted on both deep learning (DL)-based and classical segmentation methods, focusing on early and late fusion techniques across two distinct datasets. The findings reveal two key insights: First, classical segmentation approaches prove to be competitive with DL-based methods in tasks that involve foreground-background separation, demonstrating their continued relevance in certain applications. Second, late fusion emerges as the most robust fusion approach, showcasing its superior adaptability and effectiveness across various scenarios.

2 RELATED WORK

Image segmentation is a fundamental task in computer vision, which involves the division of an image into meaningful regions or objects to understand the scene [11][18][4]. In the past, semantic segmentation relied on methodsusing thresholding [15], edge-based [12] and region-based [6] . These methods have the advantage of simplicity and low computational cost.

On the other hand, convolutional neural networks (CNNs) have revolutionized the field in recent years and are now the most effective technique in pattern recognition application [8]. One of the strongest advantages of using DL in image processing is the reduced need for handcrafted features. These improvements helped agricultural tasks such as disease detection in vines [7], identification of crops, weeds, and soil [10] through architectures such as encoder-decoder SegNet and Mask R-CNN respectively.

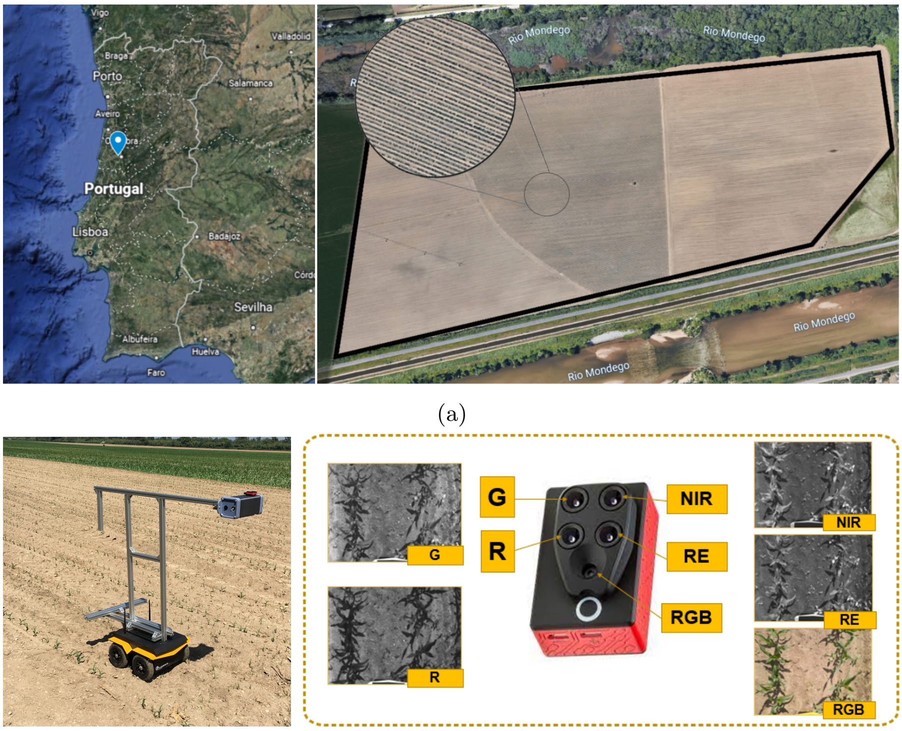
Image Segmentation can improve scene understanding however, complex environments require complementary information that multiple modalities can give to better understand the scene [1]. To achieve this goal, fusion methods can be applied which encompass, usually, three steps. First, it is necessary to understand which modalities should be fused, then what method should be applied to fuse the information, techniques like addition or average mean, concatenation or ensemble, and finally where should the information be fused along the network [3][20]. Focusing on 'where' the information is fused, we highlighted two stages, (i) the early fusion which consists of combining (merging) the data at the input layer , and (ii) the late fusion which consists of training features separately for each modality and merging them at later layers using methods such as element-wise summation [17].

3 MATERIALS AND METHODS

This section outlines the methods, tools, and processes employed to conduct the experiments of this work. Firstly, we provide a comprehensive characterization of the study sites and present the technical details of the recorded maize data. Secondly, we formulate the segmentation problem in generic terms and then in a multispectral fusion context by focusing, specifically, on early and late fusion techniques of two distinct information sources.

3.1 Study Site and Materials

The study was conducted using data collected from a maize crop known as Vargem Grande (VG) located in the Coimbra region, situated in the center of mainland Portugal (see Fig. 1a). The data collection took place during July of 2022, specifically during the early growth stage of the plants. To ensure optimal lighting conditions and minimize shadow interference, the data was collected around midday under sunny weather conditions.



(a)

(b)

(c)

Fig. 1: Study site and material used to record the dataset, where (a) illustrates the studied maize crop denominated Vargem Grande, (b) is the recording setup with which the dataset was recorded, and (c) is the multispectral sensor with its five sensors.

The multispectral dataset was captured using a Parrot Sequoia multispectral camera[1]. This camera consists of four monochrome sensors (Green, Red, Red Edge, and Near Infrared) along with an RGB sensor (see Fig. 1c). To facilitate the data collection process, the camera was mounted on a mobile platform known as the Jackal from Clearpath[2]. The camera was positioned 1.2 meters above the ground, with the sensors facing downward (see Fig. 1b).

To gather the data, the robot was teleoperated in-between the crop rows. Images from all five sensors were captured every two seconds, ensuring a comprehensive dataset for analysis.

Table 1: Specifications of the sensor. Field of View (FoV)

Sensors	Band: Center wavelength (width) [nm]	Resolution [px]	Focal Length [mm]	HFoV [º]	VFoV [º]
Mono- -chrome	G:550(40); R:660(40); RE:735(10); NIR:790(40)	1280×960	3.98	62	49
RGB	R,G,B	4068×3456	4.88	64	50

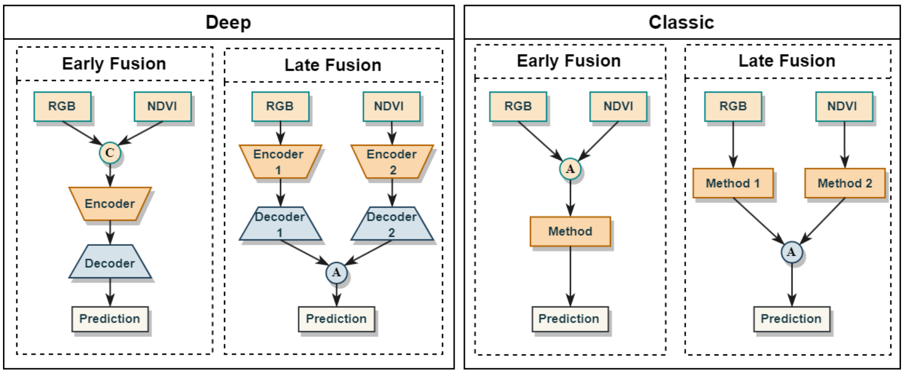


Fig. 2: Simplified approach of early and late fusion using RGB and NDVI as inputs on deep and classic methods.

3.2 Problem Formulation

Image segmentation involves the task of dividing an image into regions, or objects, based on their shared characteristics. Mathematically, image segmentation can be defined as a function that maps an input image to a class likelihood mask. Thus, let I represent the input image, defined as a three-dimensional array $I = [p_{ijk}]_{h \times w \times b}$, where $p_{ijk} \in [0, ..., 255]$ denotes the pixel intensity at coordinates (i, j, k). The image dimensions are given by h (height), w (width), and b (number of spectral bands), with $i \in [1, h]$, $j \in [1, w]$, and $k \in [1, b]$. To perform image segmentation, we aim to obtain a class likelihood mask Q, represented by $Q = [q_{ijk}]_{h \times w \times c}$. Here, $q_{ijk} \in [0, 1]$ indicates the likelihood of the pixel at coordinates (i, j, k) belonging to each of the $C = \{1, ..., c\}$ segmentation classes, constrained by $\sum_{k=1}^{c} q_{ijk} = 1$.

In the specific context of this study, we focus on binary segmentation. This means that only one class is considered, resulting in a single-channel likelihood matrix $Q = [q_{ij}]_{h \times w \times 1}$. Hence, the final segmentation mask with a class per pixel $M = [m_{ij}]_{h \times w} \in \{0, 1\}$, is obtained through a threshold-based approach:

[1] Parrot Sequoia User Guide

[2] Jackal Homepage

$$m_{ij} = \begin{cases} 1 & \text{if } q_{ij} \geq T \\ 0 & \text{if } q_{ij} < T \end{cases} \tag{1}$$

where T is a threshold value chosen to distinguish between the positive and negative classes in the segmentation.

The binary segmentation framework is used to compare classical methods with deep learning (DL)-based approaches using two input modalities: RGB (I^{RGB}) and NDVI (I^{N}). The RGB image I^{RGB} is defined as a tree-dimensional array $I^{RGB} = [p_{ijk}^{RGB}]_{h \times w \times 3}$, capturing the visible spectrum (400-700 nm) with the Red, Green, and Blue bands. On the other hand, the NDVI image I^{N} is a two-dimensional array $I^{N} = [p_{ij}^{N}]_{h \times w}$, representing the Normalized Difference Vegetation Index. The NDVI is calculated as:

$$I^{N} = \frac{NIR - Red}{NIR + Red} \, , \tag{2}$$

where Red and NIR correspond to specific spectral bands. The Red band lies within the visible spectrum, while the NIR band extends beyond the visible range (700 to 1100 nm). These bands are particularly valuable for agricultural monitoring, capturing the absorption of chlorophyll in visible light and its reflection in the NIR spectrum.

3.3 Image Fusion

Fusion, in the context of image segmentation, refers to the integration of information derived from diverse sources into a unified representation. The fusion process can be applied at various stages, depending on the segmentation methods employed [16]. In this study, we specifically investigate two fusion approaches: early fusion and late fusion.

Early Fusion In the context of image processing, early fusion involves the merging of information at the input level, specifically within the pixel space. In this study, early fusion is employed using two different approaches: classical segmentation methods and DL-based segmentation methods.

In the comparison between classical and DL-based methods, the representation of early fusion varies depending on the approach used. Specifically, when employing classical approaches, the RGB image I^{RGB} is transformed into a grayscale representation denoted as $I^{Gr} = [p_{ij}^{Gr}]_{h \times w}$. This conversion is achieved using the standard formula:

$$p_{ij}^{Gr} = 0.299 \, p_{ij}^{R} + 0.587 \, p_{ij}^{G} + 0.114 \, p_{ij}^{B} \, , \tag{3}$$

where p_{ij}^{R}, p_{ij}^{G}, and p_{ij}^{B} represent the pixel intensities of the Red, Green, and Blue bands at the coordinate (i, j), respectively, with $i \in [1, h]$ and $j \in [1, w]$. The resulting grayscale image I^{Gr} has dimensions given by $h \times w$.

For classical approaches, the fused representation I^{Ec} is obtained by computing the pixel-wise mean between the NDVI image I^N and the grayscale image I^{Gr}:

$$p_{ij}^{Ec} = \frac{p_{ij}^N + p_{ij}^{Gr}}{2} \ , \tag{4}$$

here, p_{ij}^N and p_{ij}^{Gr} represent the pixel intensities of the NDVI and grayscale images at the (i, j) coordinate, respectively. The resulting fused representation I^{Ec} is an image of dimensions $h \times w$. On the other hand, when employing DL-based segmentation methods, the fused representation I^{Ed} is obtained by channel-wise concatenation of the RGB image I^{RGB} and the NDVI image I^N. This is represented as:

$$I^{Ed} = [I^{RGB}, I^N] = \left[p_{ijk}^{RGB} \,|\, p_{ijk}^N\right]_{h\times w\times 4} \tag{5}$$

where p_{ijk}^{RGB} and p_{ijk}^N represent the pixel intensities of the RGB and NDVI images at the (i, j, k) coordinate, respectively. The resulting fused representation I^{Ed} is a tensor with dimensions $h \times w \times 4$, where the first three channels correspond to the RGB image and the fourth channel corresponds to the NDVI image.

Late Fusion Early fusion involves merging information at the input space, while late fusion performs the merging at the output space. In this study, late fusion is achieved by computing the pixel-wise weighted sum of the class likelihoods of each model before the final class decision.

In the context of a late fusion framework, the segmentation process involves two input images: I^N and I^{RGB}. Each image is individually processed through a segmentation model, generating respective output likelihood masks: $Q^N = [q_{ij}^N]_{h\times w\times 1}$ and $Q^{RGB} = [q_{ij}^{RGB}]_{h\times w\times 1}$, where, q_{ij}^N and $q_{ij}^{RGB} \in [0, 1]$ represent the likelihood of the positive class at the pixel coordinates (i, j).

The fused representation is obtained by computing a pixel-wise weighted sum of the likelihoods from both segmentation models. Hence, the fused likelihood q_{ij}^L at the pixel coordinates (i, j) is calculated using the following formula:

$$q_{ij}^L = \alpha \cdot q_{ij}^N + \beta \cdot q_{ij}^{RGB} \ , \tag{6}$$

where α and β are weights that can be adjusted to balance the contribution of each likelihood according to the models' performance. By controlling the values of α and β, the fusion process can be fine-tuned to achieve optimal segmentation results based on the strengths of the individual models.

4 EXPERIMENTAL EVALUATION

The evaluation section in this study provides a comprehensive assessment of early and late fusion techniques within a multispectral image segmentation framework

Table 2: Dataset information, where B,G,R,RE and NIR represent Blue, Green, Red, Red-edge and Near-infrared, respectively.

Dataset	Vargem Grande	Qta Baixo	ESAC	Valdoeiro
Sample Size (Train/Test)	532 (425/107)	150	189	120
Bands	R, G, RE, NIR, RGB	B, G, R, RE, NIR, Thermal		
Dimensions Fusion	1100×825	240×240		

applied to the AgRA domain. The section outlines the datasets used for evaluation, describes the implementation details and evaluation metrics employed, and presents a thorough discussion of the quantitative and qualitative results obtained.

4.1 Datasets

The proposed approaches undergo evaluation using primarily the maize crop dataset (referred to as VG) described in Section 3.1. Complementary, a dataset collected from vineyards are used to assess cross-domain generalization capability. For the VG dataset, a total of 532 images were recorded for each of the five sensors (R, G, RE, NIR, and RGB). The images were aligned and cropped to a final size of 1100 × 825, and for evaluation purposes, they were resized to 240 × 240. The dataset was then split into an 80/20 ratio for training and testing, respectively. Regarding the vineyard data, the dataset encompasses images of 240 × 240 from three distinct vineyards. The evaluation follows the approach proposed in [2], employing a cross-validation method that involves training on data from two vineyards and testing on the third. Relevant information about the datasets can be found in Table 2.

4.2 Implementation Details

This section outlines the implementation details of both the classical and DL-based segmentation approaches.

Classical Approach Three classical segmentation methods were employed: Otsu's thresholding[3], edge-based[4], and region-based[4] techniques. For Otsu's thresholding, the opencv *threshold* function with an automatic threshold value was utilized to perform the segmentation. In the case of edge-based segmentation, the Canny edge detector was employed to detect the edges of the objects, with further processing to fill the contours and remove small objects from the segmented image. Lastly, a region-based segmentation was performed by generating

[3] OpenCV Image Thresholding - Otsu's thresholding.

[4] Edge-based and Region-based segmentation - Canny edge-detector and Watershed transform.

an elevation map using the Sobel gradient, determining markers for background and plants based on gray value histograms, and then applying the watershed transform to fill regions of the elevation map with those markers.

Deep Learning Approaches In this study, two distinct DL-based segmentation models were utilized: SegNet[5] and DeepLabV3[6]. SegNet employs an encoder-decoder architecture, where the input is gradually encoded to a latent space and then gradually decoded to an output mask. In contrast, DeepLabV3 upsamples the latent representation in fewer steps.

Both models were implemented using the PyTorch [13] framework and executed on a hardware setup consisting of an NVIDIA GEFORCE GTX 3090 GPU and an AMD Ryzen 9 5900X CPU with 64 GB of RAM. The training process utilized the AdamW optimizer [9] with a learning rate of 1e-3 for VG and approximately 1e-4 and 1e-5 for vine models. The Binary Cross-Entropy with Logits Loss (BCEWithLogitsLoss) function was employed to calculate the loss, and the outputs (logits) were passed through a sigmoid activation function to obtain the final probabilities.

4.3 Evaluation Metrics

The performance of the segmentation methods was evaluated using several metrics, including pixel accuracy (acc), F_1 score, and Intersection over Union (IoU). These metrics provide insights into the accuracy and quality of the segmentation results. The pixel accuracy is defined as:

$$\text{acc} = \frac{TP + TN}{TP + FP + TN + FN}, \tag{7}$$

where TP, TN, FP, and FN represent True Positives, True Negatives, False Positives, and False Negatives, respectively. The F_1 score is calculated as:

$$F_1 \text{ score} = \frac{2 \times TP}{2 \times TP + FP + FN}. \tag{8}$$

The IoU is computed as :

$$IoU = \frac{\text{Area of Intersection}}{\text{Area of Union}} , \tag{9}$$

where *Area of Intersection* refers to the number of overlapping pixels between the predicted mask and ground truth mask: $A \cap B = \{p_{ij} : p_{ij} \in A \text{ and } p_{ij} \in B\}$, where p_{ij} denotes a pixel at coordinate (i, j), while A and B represent the ground truth mask and the predicted mask, respectively. The *Area of Union* represents the total number of pixels encompassed by both prediction and ground truth masks, including the overlapping region: $A \cup B = \{p_{ij} : p_{ij} \in A \text{ or } p_{ij} \in B\}$, where p_{ij} denotes a pixel at coordinate (i, j), while A and B represent the ground truth mask and the predicted mask, respectively.

[5] SegNet GitHub Implementation

[6] DeepLabV3 Pytorch

Table 3: Segmentation performance on the Maize (VG) and Vine (Qta. Baixo, ESAC, and Valdoeiro) datasets, employing classical approaches such as Otsu Threshold (OST), Edge-based, and Region-based, as well as DL-based approaches including SegNet and DeeplabV3. Each method is evaluated with four scores: RGB and NDVI individually, and both modalities fused using early and late fusion techniques. The performance scores are presented in percentage [%], with the **best score** highlighted in bold and the second-best scores underlined.

Dataset / Method		Maize			Vine									Average		
		VG			Qta. Baixo			ESAC			Valdoeiro					
		Acc.	F1	IoU	Acc.	F1	IoU	Acc.	F1	IoU	Acc.	F1	IoU	Acc.	F1	IoU
OTS	RGB	74.4	28.3	16.7	57.6	41.9	27.6	79.1	52.7	40.2	74.5	38.6	26.0	71.4	40.4	27.6
	NDVI	76.1	32.3	19.6	84.1	61.7	46.1	65.6	49.7	34.8	92.4	67.8	56.0	79.6	52.9	39.1
	Early F.	75.6	34.1	20.9	71.6	52.7	37.7	71.4	55.5	41.2	89.5	65.3	52.7	77.1	51.9	38.1
	Late F.	67.5	33.7	20.8	83.0	67.5	44.7	89.5	66.8	42.9	91.6	81.6	56.7	**82.7**	**62.4**	**41.3**
Edge-b.	RGB	76.6	12.9	6.9	69.8	15.5	8.5	78.1	26.3	15.4	86.9	22.7	13.1	77.9	19.4	10.5
	NDVI	75.6	13.0	7.0	70.3	16.8	9.3	61.1	18.7	10.4	94.2	48.6	33.7	75.3	24.3	**15.1**
	Early F.	84.1	21.0	11.9	76.8	16.3	8.9	65.4	14.8	8.0	93.9	39.3	25.8	**80.1**	22.9	13.7
	Late F.	70.1	14.0	7.6	64.8	18.9	10.6	71.1	25.5	14.8	87.6	39.1	25.1	73.5	**24.4**	14.5
Region-b.	RGB	84.1	21.0	11.9	78.3	47.4	33.1	78.9	44.3	33.2	85.3	44.6	31.2	81.7	39.3	27.4
	NDVI	82.2	12.4	6.7	89.0	67.9	52.5	76.0	50.9	37.3	97.2	76.3	63.8	86.1	51.9	40.1
	Early F.	76.7	19.0	10.5	81.3	52.7	37.0	87.6	63.5	49.8	97.3	77.6	65.2	85.8	53.2	**40.6**
	Late F.	81.8	25.3	14.7	93.1	69.3	34.9	92.1	48.9	24.6	98.6	82.7	45.2	**91.4**	**56.6**	29.9
SegNet	RGB	96.2	87.1	78.5	84.9	52.1	35.6	73.1	41.9	27.7	92.1	58.0	41.3	86.6	59.8	45.8
	NDVI	95.6	85.3	76.1	85.9	64.4	48.4	78.5	51.4	36.8	93.9	67.1	51.7	**88.5**	**67.1**	**53.3**
	Early F.	96.8	89.3	80.8	81.1	42.1	26.7	81.4	50.5	33.9	94.3	61.0	43.9	88.4	60.7	46.3
	Late F.	96.1	86.9	78.6	86.2	56.4	40.4	75.9	46.9	33.0	93.4	61.4	45.7	87.9	62.9	49.4
DeeplabV3	RGB	96.5	87.9	79.8	81.8	35.6	21.8	82.0	45.0	30.1	91.0	56.2	39.7	87.9	56.1	42.9
	NDVI	95.9	86.0	77.3	87.3	59.2	42.2	77.7	33.8	21.4	89.1	44.2	28.8	87.5	55.8	42.4
	Early F.	97.3	89.2	81.2	82.1	31.0	18.7	79.9	37.3	23.8	89.9	52.8	36.4	87.3	52.6	40.0
	Late F.	96.6	87.7	80.2	85.3	47.5	32.0	81.0	39.0	25.9	92.5	58.0	42.3	**88.9**	**58.1**	**45.1**

4.4 Results and Discussion

This section presents the experimental results for both classical and DL-based segmentation methods, comprising both quantitative and qualitative assessments. The qualitative results are organized in Table 3, while the visual representations of the segmentation masks are illustrated in Fig 3. Each segmentation approach is evaluated in four distinct methods: first, with the RGB and NDVI modalities individually, followed by the modalities fused using early and late fusion techniques, as described in Section 3.3. The results were obtained with the segmentation threshold $T = 0.5$, for the late fusion results, both models were given an equal contribution: *i.e.* $\alpha = 0.5$ and $\beta = 0.5$.

Classical vs DL-based In this work, we employ classical unsupervised and supervised DL-based segmentation methods. The classical methods demonstrate to perform well on tasks where the primary objective is to separate foreground from background, as is the case of the Vineyard dataset, where the goal is to segment individual plans. In such case, unsupervised approaches are competitive with DL-based approaches, offering the advantage of simplicity and lower complexity. However, in segmentation tasks that involve identifying spatial regions, containing both foreground and background, such as the Maize dataset, where the objective is to detect the plant rows, supervised DL-based approaches show a clear advantage due to their ability to learn spatial information. The results obtained in our experiments consistently confirm this, as depicted in Table 3 and Fig 3.

Fusion vs No-Fusion The results consistently show that late fusion either achieves the best performance or ranks a close second, distinctly outperforming early fusion. This superiority means that, on average, extracting features from individual modalities first and then fusing them at a later stage yields better results compared to one model from both modalities combined.

Upon analyzing the average results, it becomes evident that late fusion capitalizes on the model with the highest performance. By averaging the outputs of both models, late fusion is able to reduce the noise associated with the lesser-performing model. However, this method also has a downside: valuable information from the best-performing model may be diluted or lost. Thus, while late fusion leverages the strengths of both models to enhance overall robustness, finding the right balance in the contributions of each model becomes crucial. One potential approach to achieve this balance is to weight the contributions based on their respective performance. Investigating this weighted fusion strategy offers an interesting avenue for future work.

Runtime Analysis In terms of computational performance, DL methods demand a considerable amount of time to execute due to the intensive computations involved. In our case, the maximum runtime reached approximately twenty-five minutes for the entire training process, specifically during late fusion, where the

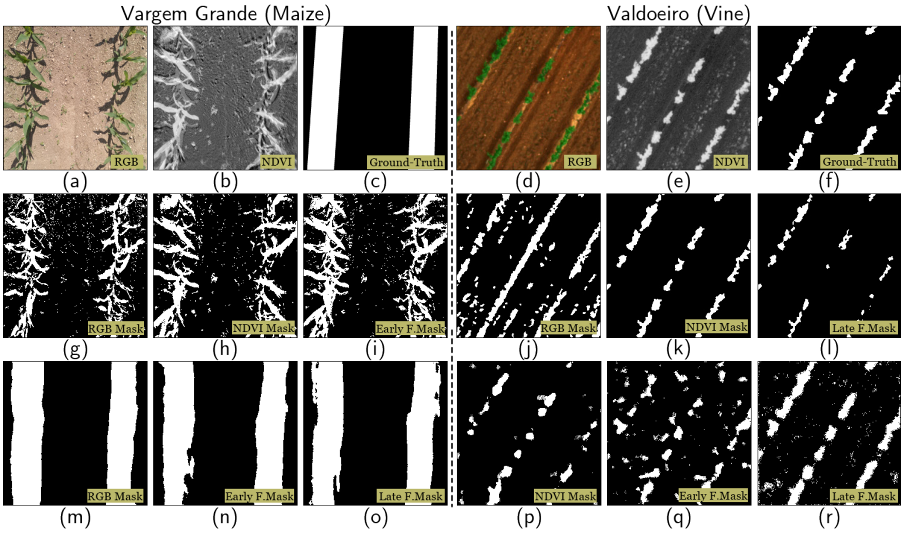


Fig. 3: Qualitative segmentation results of both VG and vineyard dataset. The images (a) to (f) (top row), represent respectively the RGB, NDVI and ground-truth masks. Images (g) to (l) (middle row) represent segmentation masks generated by classical approaches. And finally, images (m) to (r) (bottom row) represent segmentation masks generated by SegNet. More specifically, images (g) to (i) were generated by Otsu, while images (j) to (l) were generated with a region-based method.

batch size supported by the hardware was limited to 32 (VG) and 16 (Vine). In contrast, classical methods demonstrate the opposite behavior, being significantly faster and achieving results within a minute.

5 CONCLUSIONS

This work studies the impact of fusion (combining) approaches of multispectral data in segmentation tasks applied domains related to digital-precision agriculture and agricultural robotics. The study was conducted on both classical and DL-based segmentation methods, where the experimental part is supported by two datasets : a dataset of vineyards and a dataset of maize crops, recorded and curated specifically for this study.

The experimental findings show two principal observations: First, classical segmentation methods, utilizing techniques like thresholding and edge detection, are competitive against DL-based approaches in tasks requiring foreground-background separation. This highlights their continued applicability in specialized scenarios. Second, late fusion, where individual modalities are processed and then fused, emerges as the most robust approach, demonstrating its superior adaptability across various experimental conditions. These insights offer valuable guidance for both current applications and future research in segmentation algorithms.

ACKNOWLEDGMENTS

This work has been supported by the project GreenBotics (ref. PTDC/EEI-ROB/2459/2021), founded by Fundação para a Ciência e a Tecnologia (FCT), Portugal. It was also partially supported by FCT through grant UIDP/00048/2020 and under the PhD grant with reference 2021.06492.BD.

References

1. Asvadi, A., Garrote, L., Premebida, C., Peixoto, P., J. Nunes, U.: Multimodal vehicle detection: fusing 3d-lidar and color camera data. Pattern Recognition Letters **115**, 20–29 (2018)
2. Barros, T., Conde, P., Gonçalves, G., Premebida, C., Monteiro, M., Ferreira, C., Nunes, U.: Multispectral vineyard segmentation: A deep learning comparison study. Computers and Electronics in Agriculture **195**, 106,782 (2022)
3. Feng, D., Haase-Schütz, C., Rosenbaum, L., Hertlein, H., Gläser, C., Timm, F., Wiesbeck, W., Dietmayer, K.: Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. IEEE Transactions on Intelligent Transportation Systems **22**(3), 1341–1360 (2021)
4. Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., Garcia-Rodriguez, J.: A review on deep learning techniques applied to semantic segmentation (2017)
5. Jameel, S.M., Gilal, A.R., Rizvi, S.S.H., Rehman, M., Hashmani, M.A.: Practical implications and challenges of multispectral image analysis. In: 2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), pp. 1–5. IEEE (2020)
6. Karthick, S., Sathiyasekar, K., Puraneeswari, A.: A survey based on region based segmentation. International Journal of Engineering Trends and Technology **7**(3), 143–147 (2014)
7. Kerkech, M., Hafiane, A., Canals, R.: Vine disease detection in uav multispectral images using optimized image registration and deep learning segmentation approach. Computers and Electronics in Agriculture **174**, 105,446 (2020)
8. Lee, S.H., Chan, C.S., Wilkin, P., Remagnino, P.: Deep-plant: Plant identification with convolutional neural networks. In: 2015 IEEE international conference on image processing (ICIP), pp. 452–456. IEEE (2015)
9. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
10. Milioto, A., Lottes, P., Stachniss, C.: Real-time semantic segmentation of crop and weed for precision agriculture robots leveraging background knowledge in cnns. In: 2018 IEEE International Conference on Robotics and Automation (ICRA), pp. 2229–2235 (2018)
11. Minaee, S., Boykov, Y., Porikli, F., Plaza, A., Kehtarnavaz, N., Terzopoulos, D.: Image segmentation using deep learning: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(7), 3523–3542 (2022)

12. Muthukrishnan, R., Radha, M.: Edge detection techniques for image segmentation. International Journal of Computer Science & Information Technology **3**(6), 259 (2011)
13. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019)
14. Pretto, A., Aravecchia, S., Burgard, W., Chebrolu, N., Dornhege, C., Falck, T., Fleckenstein, F., Fontenla, A., Imperoli, M., Khanna, R., Liebisch, F., Lottes, P., Milioto, A., Nardi, D., Nardi, S., Pfeifer, J., Popović, M., Potena, C., Pradalier, C., Rothacker-Feder, E., Sa, I., Schaefer, A., Siegwart, R., Stachniss, C., Walter, A., Winterhalter, W., Wu, X., Nieto, J.: Building an aerial–ground robotics system for precision farming: An adaptable solution. IEEE Robotics & Automation Magazine **28**(3), 29–49 (2021). DOI 10.1109/MRA.2020.3012492
15. Sahoo, P., Soltani, S., Wong, A.: A survey of thresholding techniques. Computer Vision, Graphics, and Image Processing **41**(2), 233–260 (1988)
16. Valada, A., Oliveira, G., Brox, T., Burgard, W.: Towards robust semantic segmentation using deep fusion. In: Robotics: Science and systems (RSS 2016) workshop, are the sceptics right? Limits and potentials of deep learning in robotics, vol. 114 (2016)
17. Valada, A., Oliveira, G.L., Brox, T., Burgard, W.: Deep multispectral semantic scene understanding of forested environments using multimodal fusion. In: 2016 international symposium on experimental robotics, pp. 465–477. Springer (2017)
18. Yu, H., Yang, Z., Tan, L., Wang, Y., Sun, W., Sun, M., Tang, Y.: Methods and datasets on semantic segmentation: A review. Neurocomputing **304**, 82–103 (2018)
19. Yuan, K., Zhuang, X., Schaefer, G., Feng, J., Guan, L., Fang, H.: Deep-learning-based multispectral satellite image segmentation for water body detection. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing **14**, 7422–7434 (2021)
20. Zhang, Y., Sidibé, D., Morel, O., Mériaudeau, F.: Deep multimodal fusion for semantic image segmentation: A survey. Image and Vision Computing **105**, 104,042 (2021)