Counting and locating high-density objects using convolutional neural network

Mauro dos Santos de Arruda

Universidade Federal de Mato Grosso do Sul
Faculdade de Computação

Mauro dos Santos de Arruda

Advisor: *Prof. Dr. Wesley Nunes Gonçalves*

Doctoral dissertation submitted to the Postgraduate Course in Computer Science at the Universidade Federal de Mato Grosso do Sul, as a partial requirement to obtain the title of Doctor in Computer Science.

UFMS - Campo Grande
August 2022

Acknowledgements

To God for giving me strength and health to overcome all challenges.

To my advisor, Prof. Wesley Nunes Gonçalves, for the support, corrections, incentives and excellent ideas for solving problems and improving the method.

To Prof. José Marcato Junior for the ideas and suggestions with his environmental vision, and colleagues Lucas, Gabriela and Luciene, who contributed to each application in their respective fields.

To the financial support from CAPES and CNPq, and NVIDIA for the donation of the graphic card, and for those involved in the images capture.

To my wife Jéssica, for the support and encouragement during this work. And to all friends and family who, directly or indirectly, were part of this journey, my thanks.

Abstract

Counting and locating objects are essential in different types of applications, as they allow performance improvements in the execution of manual tasks. Deep learning methods are becoming more prominent in this type of application because they can perform good object characterizations. However, challenges such as overlapping, occlusion, scale variations and high density of objects hinder the method's performance, making this problem remains open. Such methods usually use bounding box annotations, which hinder their performance in high-density scenes with adjacent objects. To overcome these limitations, advancing the state-of-the-art, we propose a method for counting and locating objects using confidence maps. The first application allows for the definition of a method based on convolutional neural networks that receive a Multispectral image and detect objects from peaks on the confidence map. In a second application, we insert global and local context information with the Pyramid Pooling Module, to detect different scale objects. In addition we improve the successive refinement of the confidence map with multiple sigma values in the Multi-Sigma Stage phase. In the third application of the method, we propose a band selection module to work with hyperspectral images. In the fourth application, we evaluated the proposed method on high-density objects RGB images and compared it with state-of-the-art methods: YOLO, Faster R-CNN and RetinaNet. Finally, we expanded the method by proposing a two-branched architecture enabling the exchange of information between them. This improvement allows the method to simultaneously detect plants and plantation-rows in different datasets. The results described in this thesis show that the use of convolutional neural networks and confidence maps for counting and locating objects allows high performance. The contributions of this work should support significant advances in the areas of object detection and deep learning.

Keywords: Deep Learning, Object Detection, Convolutional Neural Network, Confidence Map Estimation.

Resumo

Contagem e detecção automática de objetos são essenciais em diferentes tipos de aplicações pois permitem melhorias desempenhos na execução das tarefas manuais. Métodos de aprendizado profundo estão se destacando cada vez mais nesse tipo de aplicação pois conseguem realizar boas caracterizações dos objetos. Entretanto, desafios como a sobreposição, oclusão, diferentes de escalas e alta densidade de objetos atrapalham o desempenho desses métodos, fazendo com que esse problema permaneça aberto. Tais métodos normalmente usam anotações por caixas delimitadoras, o que prejudica seu desempenho em cenas de alta densidade com adjacência de objetos. Para superar tais limitações, avançando o estado da arte, nós propomos um método de contagem e detecção de objetos usando mapas de confiança. A primeira aplicação permitiu definir um método baseado em redes neurais convolucionais que recebem como entrada uma imagem multiespectral e detecta os objetos a partir de picos no mapa de confiança. Em uma segunda aplicação, nós inserimos informações de contexto global e local através do módulo PPM, para a detecção de objetos em diferentes escalas. Além disso, melhoramos o refinamento sucessivo do mapa de confiança com múltiplos valores de sigma na fase MSS. Na terceira aplicação do método, nós propomos um módulo de seleção de bandas para trabalhar com imagens hiperespectrais. Em uma quarta aplicação, nós avaliamos o método proposto em imagens RGB de alta densidade de objetos e comparamos com métodos do estado da arte: YOLO, Faster R-CNN e RetinaNet. Por último, expandimos o método propondo uma arquitetura de duas ramificações permitindo a troca de informações entre eles. Essa melhoria permite que o método detecte simultaneamente plantas e linhas de plantio em diferentes conjuntos de dados. Os resultados descritos nesta tese mostram que a utilização de redes neurais convolucionais e mapas de confiança para a detecção e contagem de objetos permite alto desempenho. As contribuições descritas aqui, devem suportar avanços significativos nas áreas de detecção de objetos e aprendizado profundo.

Palavras-chave: Aprendizado profundo, Detecção de Objetos, Redes Neurais Convolucionais, Estimação por Mapas de Confiança.

Contents

List of Figures

List of Tables

List of Acronyms and Terms

List of Symbols

C Confidence Map

$\hat{\mathrm{C}}$ Ground Truth Confidence Map

$\mathrm{dist}_{\max}$ The maximum distance for a prediction be considered a true positive prediction

F Feature Map

f Overall Loss

fn False Negative Prediction

fp False Positive Prediction

h Height

ha Hectare

L Set of Objects Locations

m Number of Bands

mts Meters

nm Nanometre, a unit of measurement in the International System of Units

p Object Location

sec Seconds

T Number of stages that estimates a confidence map

t Stage in T

tp True Positive Prediction

w Width

δ Minimum value of pixels to separate peaks

$\mu\mathrm{m}$ Micrometre, a unit of measurement in the International System of Units

σ Parameter to controlling the spread of a peak

τ Minimum activation value to be considered as a peak

CHAPTER 1

Introduction

1.1 Context and Motivation

Object counting and locating are decisive in different applications of computer vision [Sindagi and Patel, 2018]. The performance increase has been impressive in areas such as counting and controlling people [Hsieh et al., 2017], wildlife monitoring [dos Santos de Arruda et al., 2018] and support car detections [Hsieh et al., 2017]. The proposed methods vary from solutions with Convolutional Neural Networks (Faster R-CNN [Ren et al., 2017], Mask-RCNN [He et al., 2020], RetinaNet [Lin et al., 2020]), Multi-scale deep feature learning networks [Ma et al., 2020], Gated CNN [Yuan et al., 2019]), Multi-scale variants (Multi-scale Structures [Ohn-Bar and Trivedi, 2017] and ensembles of models [Xu et al., 2020]). Methods like VGG-GAP and VGG-GAP-HR [Aich and Stavness, 2018], Layout Proposal Network (LPN) [Hsieh et al., 2017] and Deep IoU CNN [Goldman et al., 2019] has obtained encouraging results in high-density scenes. However, challenges such as scale variations, overlapping, occlusions, and high-density of objects remain to impair the detection performance.

Currently, many of these methods use Bounding Box (BBox) annotations for counting and locating objects. The work developed by Hsieh et al. [2017] is an example of the use of boxes. They propose spatial kernels with LPNs for counting objects. However, in dense object scenarios, the bounding box overlap makes adjacent regions to be detected as separate objects [Goldman et al., 2019].

In addition, the need for large-scale detailed ground-truths is a challenge

[Russakovsky et al., 2015], as acquisition and annotation are me-consuming preprocess this has led researchers to propose simpler and faster annotation methods [Zhang et al., 2018, Fiaschi et al., 2012]. In this sense, recent studies have implemented point annotations to reduce the supervision task [Aich and Stavness, 2018, Liu et al., 2019]. Besides, the similarity of color, texture and shape of objects are important factors that assist in the object detection tasks, and the use of points reduces the time of the annotation task [Liu et al., 2019, Aich and Stavness, 2018].

Another challenge in object detection field is the high-object density scenarios. Methods of these annotations are not the most suitable for these scenes [Goldman et al., 2019] since the overlap of objects makes it difficult to correctly locate the coordinates of the BBoxes. For this reason, some object counting approaches have been proposed based on density map estimation [Goldman et al., 2019, Aich and Stavness, 2018]. Aich and Stavness [2018] propose the Heatmap Regulation (HR) method that suppresses false detections using points annotations. This strategy regulates the activation maps with coarse ground-thuth in maps generated with Gaussian kernels and point annotations. In [Goldman et al., 2019], they proposed a detection method using CNN and BBox for densely packed scenes. They considered a quality score and an Expectation-Maximization (EM) unit [Moon, 1996] to solve overlapping ambiguities. However, counting and locating objects in high-density images is still an open task.

Object counting applications depend on capturing images that deliver good descriptions of the target object. The use of multispectral and hyperspectral sensors improves the differentiation of vegetation species, health status, and object description [Miyoshi et al., 2020, Takahashi Miyoshi et al., 2020, Csillik et al., 2018, Ozdarici-Ok, 2015]. In this sense, it is common to use higher spectral images in different applications [Takahashi Miyoshi et al., 2020, Hartling et al., 2019, Weinstein et al., 2019]. Multispectral and hyperspectral images have been used in counting methods that combine Light Detection and Ranging (LiDAR) and Red-Green-Blue (RGB) images to detect individual tree-crown with a self-supervised RetinaNet [Weinstein et al., 2019]. In [Hartling et al., 2019], the authors used the Dense Convolutional Network (DenseNet) method on LiDAR and Multispectral images to classify urban tree species. Still, high-resolution RGB sensors have been used in studies to identify vegetation [Weinstein et al., 2019, Berveglieri et al., 2018, Cao et al., 2018, Lobo Torres et al., 2020, Santos et al., 2019]. Despite the low cost of the RGB sensors, they still offer low image information when compared to spectral sensors.

The Hughes phenomenon is another important challenge faced by meth-

ods applied to hyperspectral images. The high dimensionality of features can impact performance by introducing noise and making data more sparser [Hennessy et al., 2020]. For these cases, deep learning methods are usually applied with a dimensionality reduction approach [Alshehhi et al., 2017], such as Principal Component Analysis (PCA) [Richards John and Xiuping, 1999] or mutual information [Audebert et al., 2019]. The band selection technique helps to identify the bands that best characterize objects [Bioucas-Dias et al., 2013]. PCA are a widely used band selection technique that reduces the data dimensionality [Tuominen et al., 2018, Maschler et al., 2018, Liu et al., 2017]. However, in the current scenario of the massive increase of data and spectral bands, more efficient band selection techniques are needed.

In this thesis, we propose a model based on Convolutional Neural Networks (CNNs) and a confidence map estimation for counting and locating objects. The method was proposed and improved throughout real-world applications problems, to address the described problems. For high-object density scenarios we based the method on a density map estimation and use point annotations. Regarding the high dimensionality of hyperspectral images, we proposed a band selection module to identify the best set of spectral bands.

Our method is divided into three phases (see Chapter 2): 1) the feature map generation with the CNN, 2) the confidence map refinement with a Multi-Stage Refinement (MSR) module, and 3) the object detection from confidence map peaks. However, in proposed methods for real-world applications problems, additional phases have been developed to improve the performance: 4) a Pyramid Pooling Module phase to insert invariance to scale (see Chapter 4), 5) a Multi-Sigma Stage phase to improve the MSR phase by inserting a multi-sigma refinement approach along the MSR module (see Chapter 4), 6) a band selection module for hyperspectral images (see Chapter 5), and 7) a Multi-Stage Module (MSM) that's improve the MSS module with a two-branched architecture to allow the simultaneous detection of plants and plantation-rows (see Chapter 6).

1.2 Goals and Contributions

This thesis goal is to propose a method based on a Convolutional Neural Network to locate and count objects using a 2D confidence map. We aim that the proposed method can be applied to different object detection tasks and image types. To reach this goal, we have proposed new methods that apply a baseline method to RGB, Multispectral and Hyperspectral images. In addition, we test the method with different datasets, such as eucalyptus and citrus-trees groves, single tree species (*Syagrus romanzoffiana*), palm trees species

(*Mauritia flexuosa*), cornfields (*Zea mays L.* recently planted, and mature-stage) and vehicle count benchmarks [de Almeida et al., 2015]: Car Parking Lot Dataset (CARPK) [Hsieh et al., 2017] and Pontifical Catholic University of Parana+ Dataset (PUCPR+).

We focus on improving the approach to overcome some common issues in the object detection field: high-dense object conditions, different object scales, overlapping, occlusion, and different types of input images. Current methods do not adapt well to these scenarios and generate missing predictions. Our methods were compared with state-of-the-art and traditional counting and locating methods like: You Only Look Once (YOLO) [Redmon and Farhadi, 2017, Jocher et al., 2022, Wang et al., 2022], Faster R-CNN [Ren et al., 2017], RetinaNet [Lin et al., 2020, Hsieh et al., 2017], LPN [Hsieh et al., 2017], VGG-GAP [Aich and Stavness, 2018]. As is now in the text, besides the better performance compared to traditional methods, the proposed method is suitable for real-world applications.

Some contributions of the proposed method and its applications were:

- Proposal of an object detection method based on confidence map estimation with good performance on regular and high-density images.
- Object detection method applied in different types of input images: RGB, Multispectral and Hyperspectral.
- Detection of objects in different scales with a proposed module that inserts global and local information.
- Improved detection of overlapping and occlusion objects with successive refinement stages.
- Proposal of a two-branch architecture with co-shared Information to improve the object detection task.
- Proposal of a Band Selection phase for learning the best band combination to improve detections.
- Proposal of a method to detect and extract rows based on confidence map estimation.
- Comparison with benchmarks and state-of-the-art methods in the object detection field.
- Development of a multispectral image dataset of citrus trees with $2,389$ images and $37,353$ trees.
- Development of a dataset of RGB images of Eucalyptus with $3,370$ images and approximately $232,000$ eucalyptus trees.

- Development of a dataset of RGB images of corn plantations (V3 and mature) with 564 images and approximately $33,360$ plants and 224 plantation-rows. [1]

1.3 Structure of the Text

The thesis consists of works proposed for real-world applications in the object detection field. The proposed method is improved to adapt to the challenges of each application, while the results influence the next steps of the method. This thesis is divided into seven chapters:

Chapter 2 presents the baseline method modeled as a 2D confidence map estimation problem for object detection applications. The Convolutional Neural Network structure and the confidence map generation are detailed. In addition, we describe the multi-stage refinement phase and the object detection from the confidence map.

Chapter 3 presents the first application of the baseline method over the citrus-trees multi-spectral dataset. We test and evaluate the main parameters of the proposed method and compare it with state-of-the-art object detection methods. The hits and the main challenges are discussed in the results.

In Chapter 4 we improve the baseline method to overcome the challenges found in the citrus-trees application. The baseline multi-stage refinement method is improved by the Multi-Sigma Stage phase. This phase refines the object detections by considering a range of maximum and minimum values of sigma. Besides, we proposed the PPM that inserts global and local context learning. The proposed method is evaluated over three datasets, a proposed dataset of eucalyptus trees and two well-known car datasets: CARPK and PUCPR+.

Chapter 5 apply the improved method on hyperspectral images for single tree species detection. To work with the high number of bands delivered by the hyperspectral images, we propose a band learning module that selects the best bands to characterize objects. The proposed method is compared with the baseline method with two different inputs: all the 25 spectral bands and the bands selected by the well-known technique PCA.

Chapter 6 propose an initial version of the method for line detection. We modify the feature map extraction module with an upsampling step and the PPM module that gives a better feature map for line detection. In addition, we improve the Multi-Sigma Stage phase in an MSM refinement with co-shared Information for lines and object detection. We evaluate the proposed method to

[1] Link to the proposed datasets: https://sites.google.com/view/geomatics-and-computer-vision/home/datasets?authuser=0

detect plants and plantation rows in cornfields and citrus orchards datasets. We compare the method with state-of-the-art methods: HRNet, Faster R-CNN, RetinaNet, YOLOv5 and YOLOv7.

Finally, the conclusion of the thesis and the next phases of the research are presented in Chapter 7.

CHAPTER

2

A New Approach to Object Detection Based on Refinement of Confidence Map using Convolutional Neural Networks

The approach takes an image as input and produces the location of each object. An image has $w \times h$ pixels and m bands, since we can apply the method to different types of images (RGB, Multispectral, Hyperspectral). The problem of object counting was modeled as a 2D confidence map estimation problem, following Cao et al. [2017]. The map is a 2D representation of the confidence that a particular object occurs in each pixel. Our proposed approach uses CNN to estimate the 2D confidence map. We use the ground truth confidence map by placing a 2D Gaussian kernel at each object location (manually labeled) to train the CNN (Sections 2.1 and 2.2). Given the confidence map, predicted and refined by a CNN, the location of each object is obtained from the peaks (local maximum), as described in Section 2.3. If n objects occur in the image, there should be a peak in the 2D confidence map corresponding to each object.

2.1 Generation of the 2D Confidence Map

Given an image with n objects, and locations $L = l_1, l_2, ..., l_n | l_k \in \Re^2$, the ground truth confidence map $\hat{\mathrm{C}}$ is obtained by placing a 2D Gaussian kernel at each center location (l_k) of the labeled objects [Aich and Stavness, 2018]. To

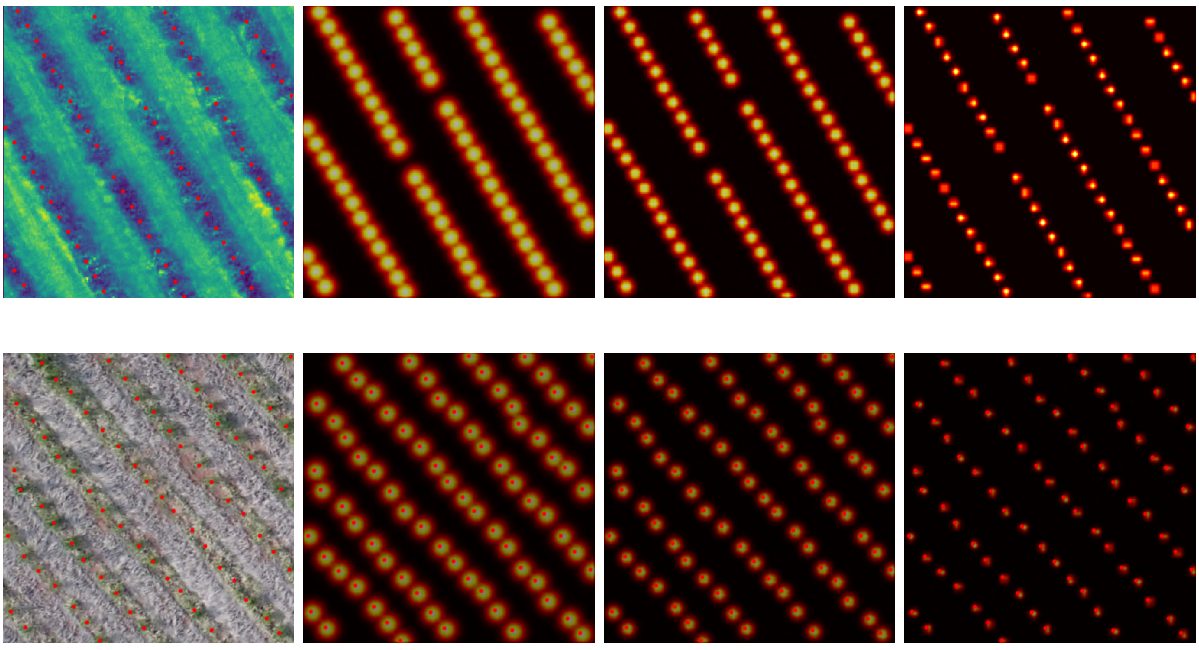
obtain $\hat{C}$, a confidence map C_k is first calculated for each object $k \in [0, n]$. The value of each location $p \in \Re^2$ in C_k is defined by:

$$C_k(p) = exp(-\frac{|p - l_k|_2^2}{\sigma^2}), \tag{2.1}$$

where σ is the important parameter controlling the spread of the peak. Ideally, σ is proportional to the size of the object. The ground truth confidence map $\hat{C}$ is obtained by aggregating the individual maps via a maximum operator (Equation 2.2).

$$\hat{C}(p) = \max_k C_k(p) \tag{2.2}$$

Figure 2.1 illustrates the confidence map for two images and three values of σ. The first column shows the images and locations of plants in red dots. The next three columns present the confidence maps for $\sigma = 1.5, 1.0, 0.5$, respectively. The ground truth confidence map is used to train a CNN.



(a) Input images (b) $\sigma_t = 1.5$ (c) $\sigma_t = 1.0$ (d) $\sigma_t = 0.5$

Figure 2.1: Example of input images of citrus and eucalyptus trees and their corresponding ground truth confidence maps with different σ_t values.

2.2 Confidence Map Estimation and Multi-Stage Refinement (MSR)

The approach uses CNN to learn a regression function that receives an image as input and returns a prediction of the confidence map as shown in

Figure 2.2. The initial part of the CNN (Figure 2.2 (b)) is based on the VGG19 [Simonyan and Zisserman, 2014]. The first two convolutional layers have 64 filters of size 3×3, and they are followed by a 2×2 max-pooling layer. The third and fourth convolutional layers have 128 filters of size 3×3, which are also followed by a 2×2 max-pooling layer. Finally, the last two convolutional layers have 256 filters of size 3×3. All convolutional layers use Rectified Linear Units (ReLU) as the activation function, with a stride of 1 and zero-padding, returning an output with the same resolution as the input. If the first part receives an image with 256×256 pixels with m bands, the model produces a feature map F with a dimension of 64×64 due to the max-pooling layers.

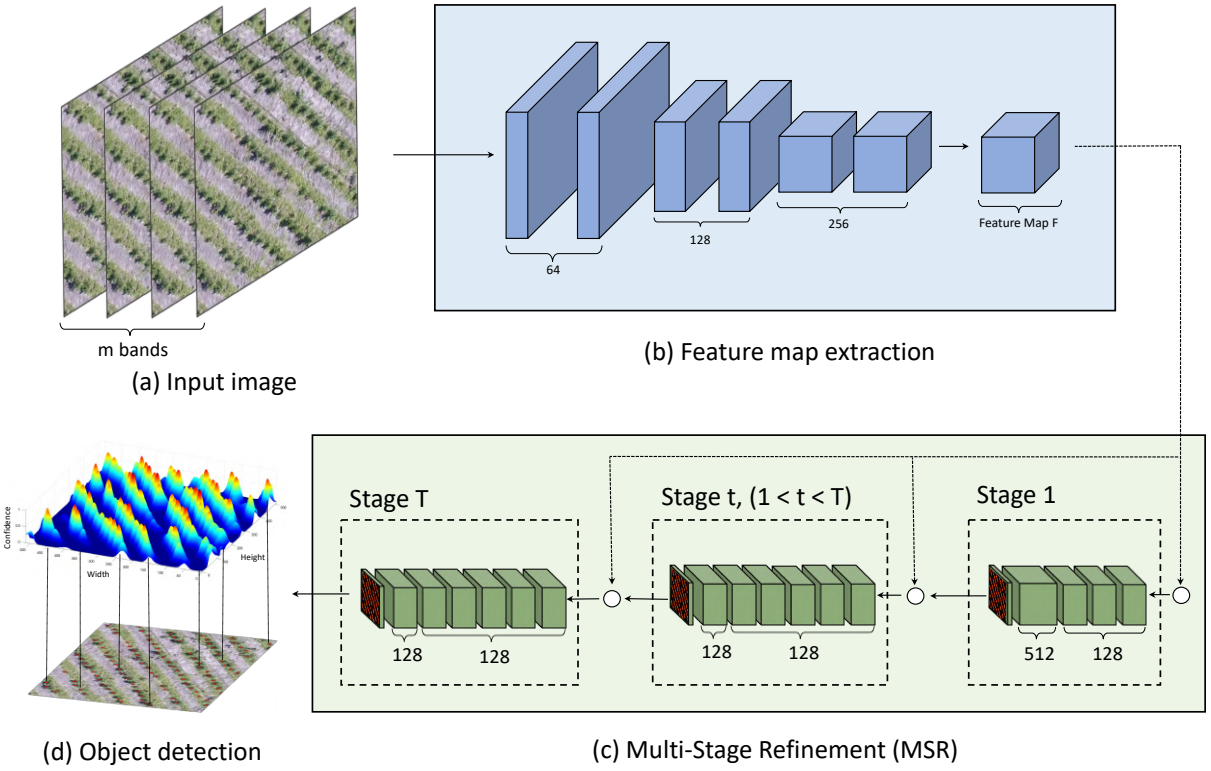


Figure 2.2: The object detection method using a confidence map prediction and a multi-stage refinement process. The initial part of CNN (b) extracts the feature map from the input image (a). The feature map is refined by multiple stages (c) that apply a standard deviation (σ) to the confidence map that is used to locate the objects (d).

The feature map F generated by the first part of the CNN is given as input to T stages of the Multi-Stage Refinement (MSR) that estimate the confidence map. In the first stage (Figure 2.2 (c)), a series of convolutional layers generate the confidence map C_1. The first stage has five convolutional layers: three layers with 128 filters of size 3×3, one layer with 512 filters of size 1×1, and one layer with a single filter that corresponds to the confidence map.

At a subsequent stage t, the prediction returned by the previous stage C_{t-1} and the feature map F are concatenated and used to produce a refined con-

fidence map C_t. The $T-1$ final stage consists of seven convolutional layers: five layers with 128 filters of size 7×7 and one layer with 128 filters with a 1×1 size. The last layer has a sigmoid activation function so that each pixel represents the probability of the occurrence of an object (values between $[0, 1]$). The remaining layers have a ReLU activation function.

To train the CNN, the loss function (Equation 2.3) is applied at the end of each stage. This intermediate supervision addresses the vanishing gradient problem as shown in Cao et al. [2017]. Since the model has max-pooling layers, the predicted confidence map is downsampling across the network and the ground truth confidence map is generated with the output size of the CNN. The loss function is given by:

$$f_t = \sum_p \parallel \hat{C}_t(p) - C_t(p) \parallel_2^2, \tag{2.3}$$

where $\hat{\mathrm{C}}_t$ is the ground truth confidence map of the stage t (Section 2.1). The overall loss f function is given by:

$$f = \sum_{t=1}^{T} f_t \tag{2.4}$$

2.3 Object Localization from the Confidence Map

Object locations are obtained from the confidence map of the last stage (C_T). We estimate the peaks (local maximum) of the confidence map by analyzing the 4-pixel neighborhood of each given location of p. Thus, $p = (x_p, y_p)$ is a local maximum if $C_T(p) > C_T(v)$ for all the neighbors v, where v is given by $(x_p \pm 1, y_p)$ or $(x_p, y_p \pm 1)$. An example of the object location from the confidence map peaks is shown in Figure 2.3 .

To avoid noise or low probability of occurrence of the positions p, the peaks need to be separated by at least δ pixels. This prevents two plants from being detected very close to each other. Also, a peak in the confidence map is considered as an object only if $C_T(p) > \tau$, where τ is a threshold parameter. The values of δ and τ are defined after preliminary experiments for each application.

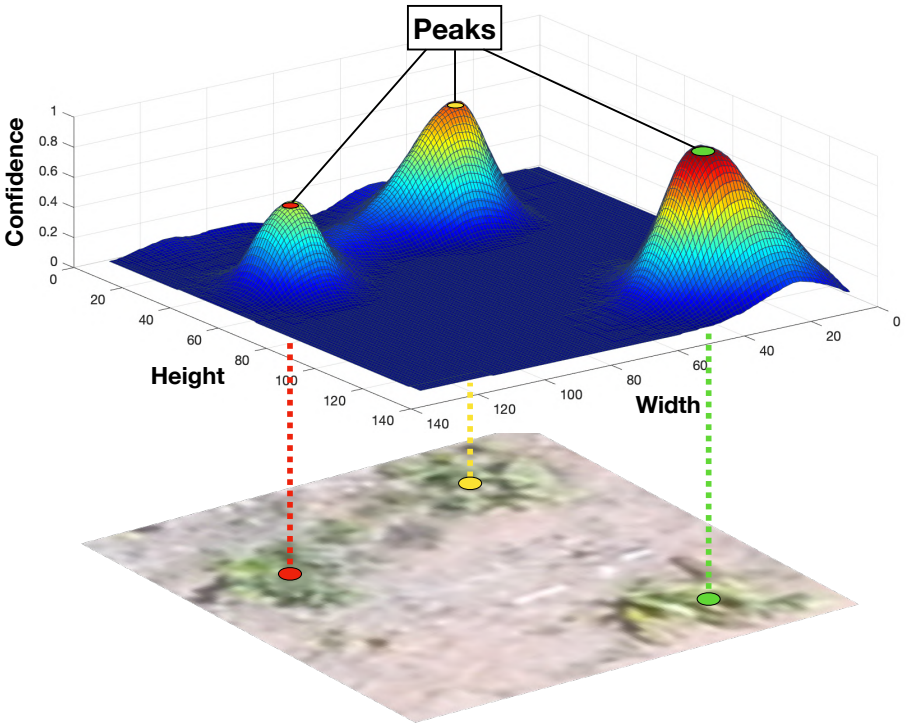


Figure 2.3: Example of the localization of eucalyptus trees from a refined confidence map.

CHAPTER

3

A convolutional neural network approach for counting and geolocating citrus-trees in UAV multispectral imagery

3.1 Introduction

Unmanned aerial vehicle (UAV) platforms allow the capture of high definition images in adverse weather conditions, places of difficult access and periodic capture of the same region [Varela et al., 2018]. Remote sensing has helped farmers maintain their field in addition to other methods in precision agriculture. However, several factors (e.g., plant and ground characteristics, environmental factors) contribute to the increased complexity of the images used in the analysis of plants [Leiva et al., 2017].

Deep Learning (DL) algorithms are benefiting remote sensing applications [Alshehhi et al., 2017, Zhang et al., 2016, Ball et al., 2017, Liu et al., 2018, Liu and Abd-Elrahman, 2018, Paoletti et al., 2018, Ma et al., 2019] and showed to have high performance for different types of application in image data from agricultural fields [Kamilaris and Prenafeta-Boldu, 2018, Wu et al., 2019]. Some applications of this type involve the analysis of wheat spikes [Hasan et al., 2018], wheat-ear density estimation [Madec et al., 2019], rice seedlings in the field [Wu et al., 2019] and the counting of fruits [Chen et al., 2017], plants [Djerriri et al., 2018] and trees [Jiang et al., 2017, Li et al., 2017] in

crop fields.

Plant number information is essential for farmers estimate productivity, evaluate the density of their plantations and errors occurring during the seedling process [Ampatzidis and Partel, 2019]. Several techniques have been proposed to identify and count trees [Goldbergs et al., 2018], since the plant counting process is a labor-intensive and time-consuming task [Leiva et al., 2017]. Some of these research investigates the potential of the CNN approach applied to images obtained from UAV-borne sensors [Djerriri et al., 2018, Onishi and Ise, 2018, Salami et al., 2019]. The automated detection and counting process is being applied in counting trees in agricultural fields, such as citrus plantations [Ozdarici-Ok, 2015]. Recently, the implementation of CNN in UAV image produced high precision results, up to 99.9% [Ampatzidis and Partel, 2019] and 94.59% [Csillik et al., 2018].

Although studies have given high accuracy in counting citrus trees using CNN in UAV multispectral images, the current methodology [Csillik et al., 2018, Ampatzidis and Partel, 2019] is based on object detection CNNs. These CNNs use rectangles to detect each plant individually, but their detection and performance decrease as the image becomes crowded and the plant size decreases [Kang et al., 2019]. In such cases, the boundaries of individual plants may not be sufficiently visible to detect a rectangle, which may increase the difficulty of discriminating individual plants. This chapter presents an application [Osco et al., 2020a] of the proposed method, to cope with the challenge of estimating the number of citrus trees in highly dense orchards from multispectral UAV images.

3.2 Proposed Method

We take as input a UAV multispectral image, with $w \times h$ pixels and m bands, and produces the location of each plant. The proposed method use a 2D confidence map that represent the confidence of a plant occurs in each pixel (see Sections 2.1, 2.2 and 2.3). The plant locations are obtained by the peaks generated in the refined confidence map resulting from the CNN. Figure 3.1 shows the CNN used to generate de confidence map prediction.

To train the approach, we generate a ground truth confidence map $\hat{\mathrm{C}}$ by placing a 2D Gaussian kernel at each plant location. The ground truth generate process is applied for each stage of the network refinement stages. Finally, the ground truth confidence map $\hat{\mathrm{C}}$ is obtained by aggregating the individual maps C_t via the maximum operator (this process are detailed in the Section 2.1).

This work was the first application of the baseline method for object detec-

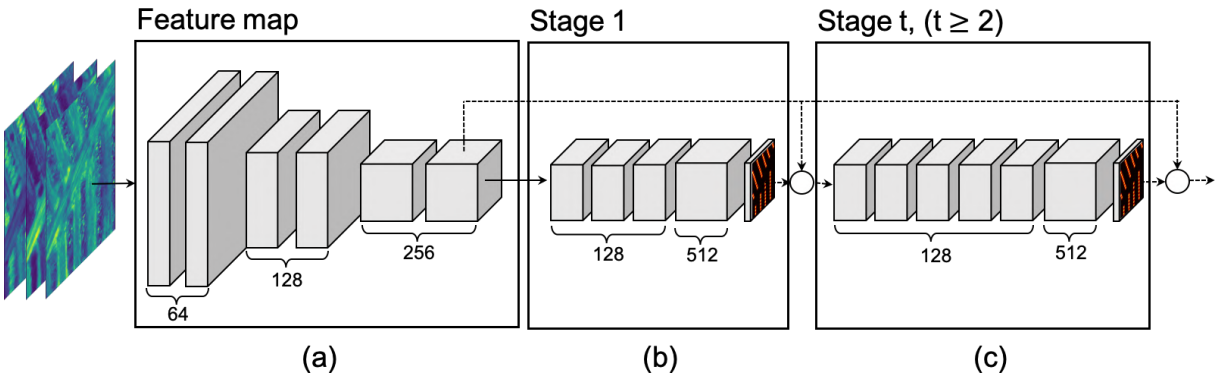


Figure 3.1: CNN used for confidence map prediction. It consists of an initial part (a) to extract a feature map of the input image. This feature map is used as input to the first stage (b). The concatenation of the feature map and the prediction map of the previous stage is used as input for the remaining stages.

tion. The method receives an image with 256×256 pixels and m bands, and produces a feature map F with 64×64 due to the max-pooling layers. Since the size of the predicted confidence map is smaller than the image size, the ground truth confidence map is generated with the output size of the CNN, which in this work was 64×64 pixels.

Besides, for the plant localization in the confidence map, we apply the values of $\delta = 3$ and $\tau = 0.2$ after preliminary experiments. Where δ prevents two plants from being detected very close to each other and the threshold τ defines the minimum confidence to consider a peak as a detected plant. Figure 3.2 shows an example of the confidence map, where the width and height are the image dimensions and the blue peaks represent the regions with local maximum confidence.

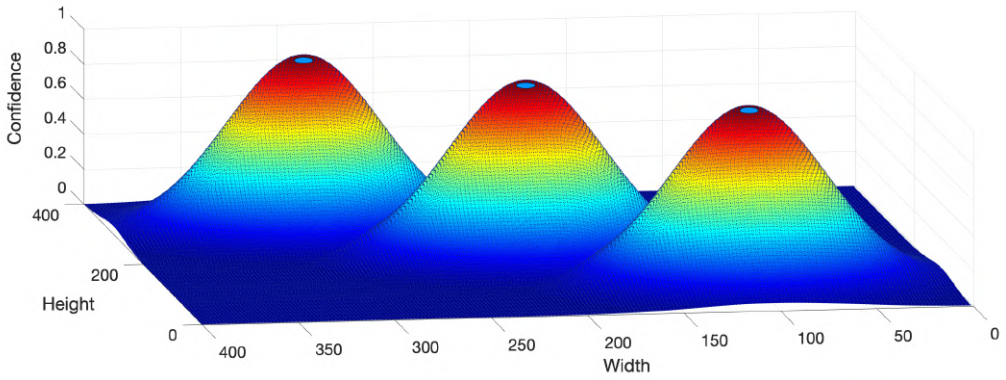


Figure 3.2: Example of the confidence map in three dimensions.

3.3 Experiments

3.3.1 Studied Area

Figure 3.3 shows our studied area with planting lines of a Valencia-orange tree orchard (Citrumelo Swingle rootstock), located in a property in Ubirajara, SP, Brazil. The area has approximately $70\ ha$, with Valencia-orange trees planted at a $7 \times 1.9\ mts$ spacing, with around 752 plants per ha. The UAV flight took place on March 22, 2018, and the trees were in their vegetative state. The trees were approximately 5 years old and about $3\ mts$ high, reaching their maturity and production stages.

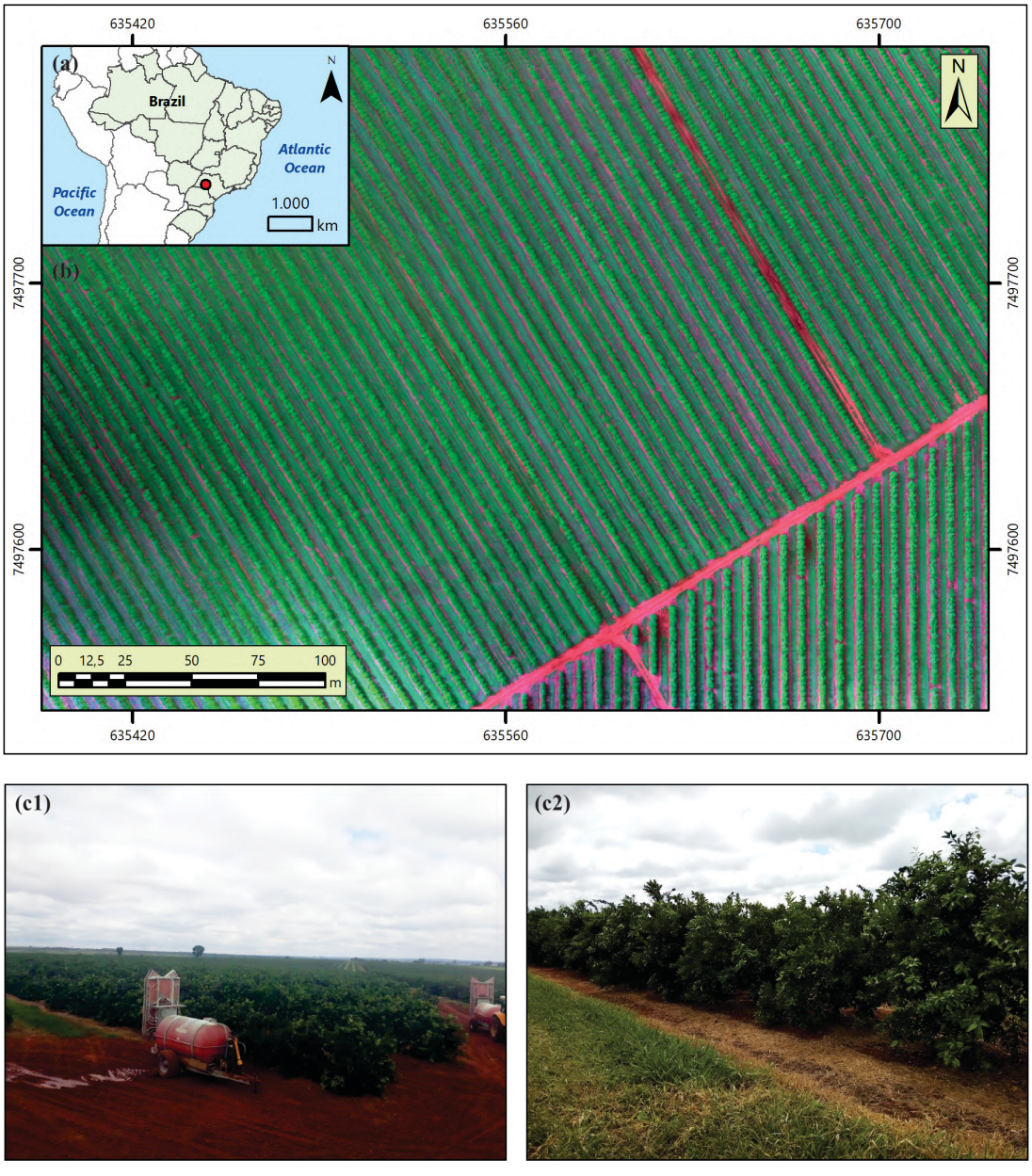


Figure 3.3: Information of the study area: Figure (a) shows its location on the map, the study area is shown through a combination of bands in figure (b), and figures (c1) and (c2) show examples of the planting lines in the ground.

The images were acquired with a Parrot Sequoia camera (©Parrot-Drones SAS, USA) onboard the eBee SenseFly UAV (©SenseFly, Parrot-Group, USA),

which operates in the four spectral bands of green, red, red-edge, and Near-Infrared (NIR), respectively. A total of $37,353$ trees were manually identified in the orthophoto, which was generated using $2,389$ images, acquired in the study area. Details describing the cameras and flight conditions are presented in Table 3.1.

Table 3.1: Parrot Sequoia camera and eBee SenseFly flight details.

Spectral band	**Wavelength**	**Bandwidth**	**Spectral Resolution**	10 bits	**Flight High**	120 mts
Green	550 nm	40	**(GSD) Spatial Resolution**	12.9 cm	**Flight Time**	01:30 P.M.
Red	660 nm	40	**HFOV**	70.6°	**Weather**	cloudy/partially-cloudy
Red-edge	735 nm	10	**VFOV**	52.6°	**Precipitation**	0 mm
Near-Infrared	790 nm	40	**DFOC**	89.6°	**Wind**	at 1 to 2 mts/sec

Ground Sample Distance (GSD); Horizontal Field of View (HFOV); Vertical Field of View (VFOV); Displayed Field of View (DFOC)

The orthorectification was performed with Pix4DMapper software using 9 Ground Control Points (GCPs) surveyed with dual-frequency Global Navigation Satellite System (GNSS) Leica Plus GS15 receiver, in Real-Time Kinematic (RTK) mode. The images were radiometrically corrected using the radiance values of a calibrated reflectance plate, recorded with the camera prior to the flight. An orthorectified surface reflectance image was generated, and the tree locations were generated as point features using the photointerpretation technique.

3.3.2 Experimental Setup

The orthorectified surface reflectance image was split into 562 patches of 256×256 non-overlapping pixels (with approximately 33×33 meters). To evaluate the proposed approach, the patches were randomly divided into training, validation and testing sets made up of 80% (448 patches), 10% (56 patches), and 10% (56 patches), respectively. For training, the Stochastic Gradient Descent (SGD) optimizer was used with a momentum of 0.9. Hyperparameter tuning was performed on the learning rate and the number of epochs, using the validation set to reduce the risk of overfitting. After a minimal hyperparameter tuning, the learning rate was 0.01 and the number of epochs was 300. Instead of training the proposed approach from scratch, the weights of the first part were initialized with pre-trained weights in ImageNet. When the multispectral image had more than three channels, an additional layer with random weights in the first layer was included.

In the experiments, regression metrics are reported measuring the agreement between the number of annotated and predicted plants. The metrics were Mean Absolute Error (MAE), Mean Squared Error (MSE), Coefficient of Determination (R^2), and Normalized Root-Mean-Squared Error (NRMSE). Given the number of annotated y_j and predicted $\hat{y}_j$ plants for patch j, MAE calculates the average of the absolute errors, defined by Equation 3.1

$$MAE = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j| \tag{3.1}$$

Similarly, MSE estimates the average of the squares of the errors, defined by Equation 3.2.

$$MSE = \frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2 \tag{3.2}$$

NRMSE represents the square root of the normalized MSE. This metric facilitates the comparison between methods that work at different scales.

Finally, the R^2 estimates the correlation between the number of annotated and predicted plants. To assess the quality of plant detection, we also used classification metrics such as Precision (P), Recall (R), and F1-Measure (F1) calculated according to Equation 3.3.

$$P = tp/(tp + fp) \qquad R = tp/(tp + fn) \qquad F1 = 2 * \frac{P * R}{(P + R)} \tag{3.3}$$

We defined a true positive (tp) if the predicted and annotated position of the plant is at less than a maximum distance $\mathrm{dist}_{\max}$. False-positive (fp) and false negative (fn) are calculated similarly using the distance $\mathrm{dist}_{\max}$. In this work, the $\mathrm{dist}_{\max}$ was defined as the size of the tree canopy (120 cm). We compared our method to two object detection methods, RetinaNet and Faster R-CNN.

Training and testing were performed using a desktop computer with Intel(R) Xeon(R) CPU $E3 - 1270@3.80$ GHz, 64 GB memory, and NVIDIA Titan V graphics card (5120 Compute Unified Device Architecture (CUDA) cores and 12 GB graphics memory). The methods were implemented using Keras-Tensorflow on the Ubuntu 18.04 operating system. The computational cost for the different number of stages (T) considering this desktop had already been assessed.

3.4 Results and Discussion

3.4.1 Analysis of the Proposed Method Parameters

Table 3.2 presents the results for different bands and combinations among them. The objective is to evaluate which bands are most appropriate for plant counting using the proposed approach. These results were obtained using $T = 6$ stages and $\sigma = 1.0$. Even considering only one spectral band (e.g., green), the proposed approach already presents satisfactory results. However, a performance increase was obtained when combining the green, red and NIR bands, giving an NRMSE of 0.038.

It can also be seen that using the Red-edge band did not imply good results

Table 3.2: Results obtained with different bands and combinations.

Bands	**MAE**	**MSE**	R^2	**NRMSE**
Green	2.51	10.72	0.96	0.039
Red	2.74	13.09	0.95	0.046
Red-edge	3.65	40.56	0.85	0.077
Nir	2.98	18.11	0.93	0.052
Green, Red	2.40	13.32	0.95	0.046
Green, Red-edge	2.67	15.68	0.94	0.050
Green, Nir	2.93	16.09	0.94	0.051
Red, Red-edge	2.37	15.18	0.94	0.050
Red, Nir	2.82	17.35	0.93	0.052
Red-edge, Nir	2.96	17.74	0.93	0.052
Green, Red, Red-edge	2.65	15.67	0.94	0.050
Green, Red, Nir	**2.28**	**9.82**	**0.96**	**0.038**
Green, Red-edge, Nir	2.89	20.09	0.92	0.057
Red, Red-edge, Nir	2.68	14.44	0.95	0.047
Green, Red, Red-edge, Nir	2.56	13.47	0.95	0.046

compared to the other bands. We observed that the Red-edge band does not have sufficient contrast regarding other targets. Red-edge parameters such as curve slope and reflectance can be used to differentiate illuminated from shaded canopies [Xu et al., 2019], and its usage is commonly known in remote sensing applications. However, the evaluated region ($735 \pm 10\ nm$) in this study presented a high similarity between other vegetation targets.

The spectral response from the citrus plants in comparison to other types of land cover (bare soil, shallow grassland, and dense grassland) in the study area is displayed in Figure 3.4. By collecting different samples (one hundred for each land cover type), it could be seen that the orange-trees and dense grassland presented similar surface reflectance at the Red-edge region. This may be indicative of the reduced CNN performance for this band. In general, similar studies implemented common RGB cameras in their analysis [Csillik et al., 2018, Weinstein et al., 2019, Varela et al., 2018, Ampatzidis and Partel, 2019, Fan et al., 2018], so this type of problem was not perceptible. But the CNN struggle in the Red-edge band in this study case is an important finding since it directs towards adversity in using this band for the proposed task.

When increasing to two bands, the use of the Green and Red bands obtained the best result, although it did not surpass the results obtained by the Green band alone. On the other hand, using the Green, Red and NIR bands obtained the best result. These bands achieved MAE, MSE, R^2, and NRMSE of 2.28, 9.82, 0.96 and 0.038, respectively. Considering the four bands as input images, the results were satisfactory although it did not surpass the best result because of the inclusion of the Red-edge band, which does not help in counting the plants.

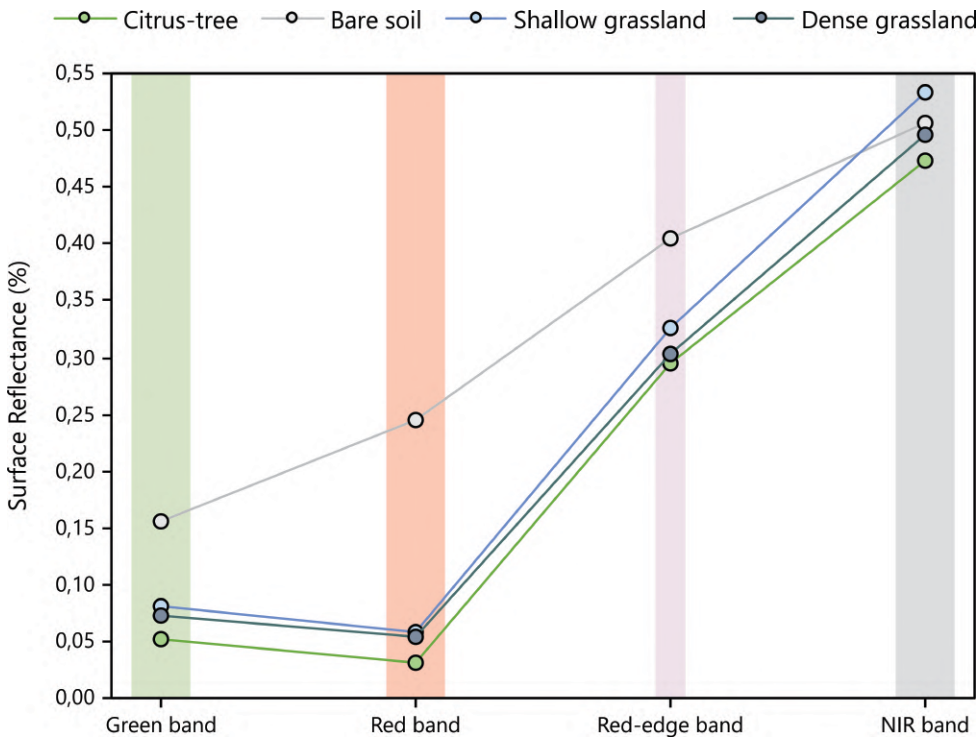


Figure 3.4: Spectral behavior of different types of landcover commonly present in the study area.

The σ, which is responsible for generating the ground truth confidence maps used in the training of the proposed approach, was also evaluated. In these experiments, the green, red and NIR bands that achieved the best results among all bands in the previous experiment were used. σ has a great influence on the results (see Table 3.3) and the best result was obtained for $\sigma = 1.0$, which, in this case, is better fitted to the size of the tree canopy.

Table 3.3: Evaluation of the σ responsible for generating ground truth confidence maps to train the proposed approach.

σ	MAE	MSE	R^2	NRMSE
0.5	5.11	58.86	0.78	0.098
1.0	**2.28**	**9.82**	**0.96**	**0.038**
1.5	3.56	25.63	0.90	0.064

Finally, the number of stages that refine the confidence map predicted by the proposed approach was evaluated (Table 3.4). As expected, the results improve as the number of stages is increased. This shows that the refinement of the confidence map helps in counting the plants. The proposed approach achieved its best result with eight stages ($T = 8$).

The results show that the proposed approach provided accurate results for counting plants and can be used to automate this task. This performance approximates from the accuracy obtained in lesser difficult conditions, such as a high-spaced citrus plantation [Csillik et al., 2018, Ampatzidis and Partel,

Table 3.4: Evaluation of the number of stages T used to refine the confidence map predicted by the proposed approach.

Stages (T)	**MAE**	**MSE**	R^2	**NRMSE**
1	3.61	21.05	0.92	0.057
2	2.86	17.39	0.93	0.052
4	2.56	14.42	0.95	0.047
6	2.28	9.82	0.96	0.038
8	**2.05**	**8.75**	**0.97**	**0.036**
10	2.21	11.79	0.96	0.043

2019]. A visually similar density condition was evaluated in a different crop type [Fan et al., 2018], which achieve 93% accuracy on tobacco plant detection using CNN.

3.4.2 Qualitative Results

To analyze the results qualitatively, a region around the annotated locations was considered to visualize the proximity of the prediction and the center of the plants. Figure 3.5 shows the results using the best configuration (three bands, $\sigma = 1.0$, and $T = 8$). The predicted locations are represented by red dots in this figure and the plant regions are represented by yellow circles whose center is the location annotated by the specialist. It can be seen that the proposed approach can correctly predict most plant locations, with a 2.05 trees error per image, so that they are aligned with the annotated locations and within the plant region.

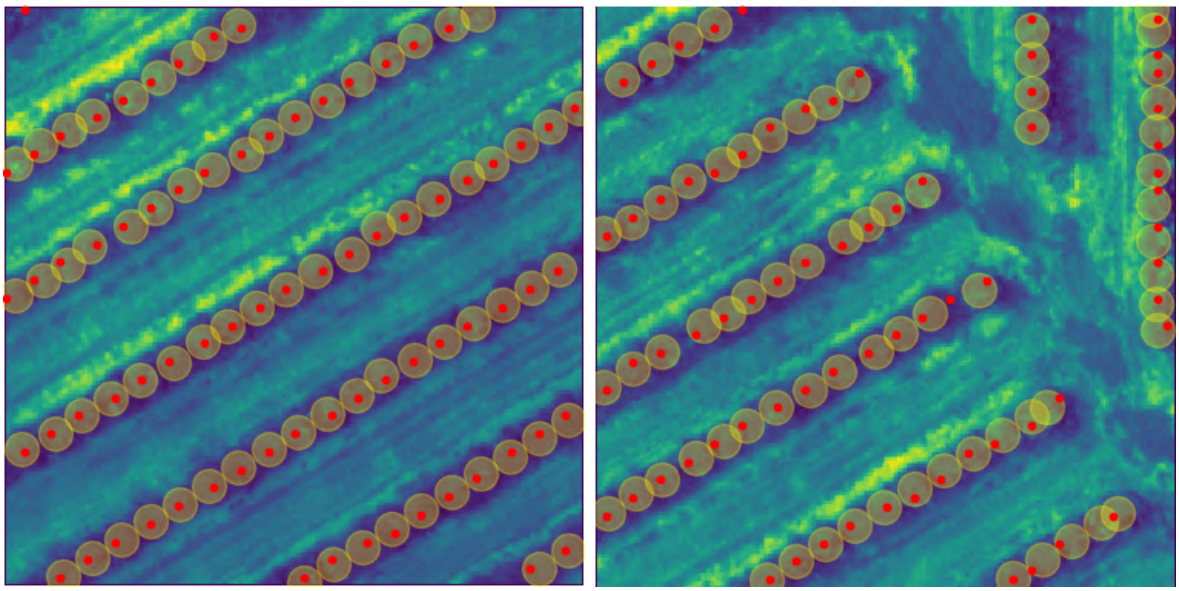


Figure 3.5: Comparison of predicted locations (red dots) and plant regions (yellow circles) in two images.

The results show that planting lines are also identified without the need for

any annotation or additional procedure. Identifying planting lines is also an important feature in remote sensing of agricultural fields since it can easily detect missing trees and help optimize crop management [Bah et al., 2018, Oliveira et al., 2018]. Nonetheless, some difficulties were observed considering the characteristics of the area investigated here. Figure 3.6 shows examples of the main challenges faced by our approach.

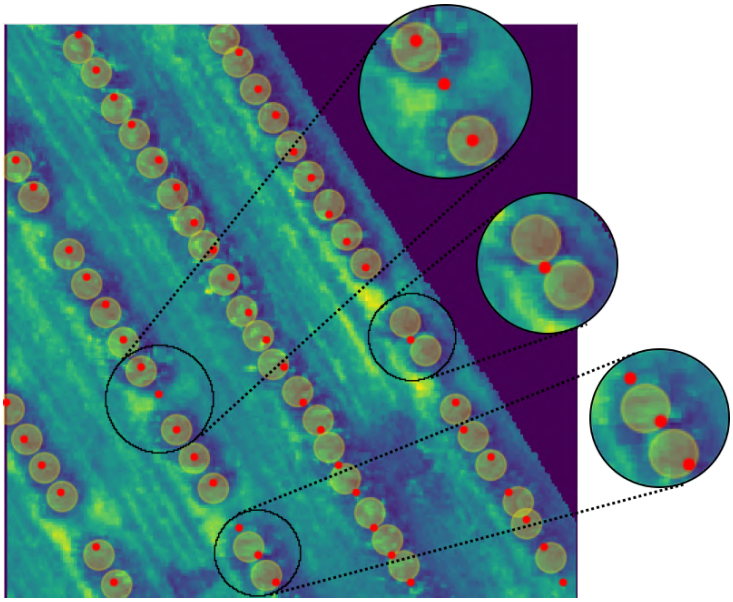


Figure 3.6: Examples of the challenges faced by the proposed approach.

It can be seen that far-center predictions occur in short planting lines (2 to 4 plants) or when much of the plant canopy is occluded. However, even in images where these cases occur, the proposed approach is capable of predicting the location of the vast majority of plants. Besides, different plantation lines with spaced tree locations were identified by the CNN method without difficulty (Figure 3.7). This indicates that our approach is also suitable for estimating isolated trees with different plant spacing.

3.4.3 Comparison with Object Detection Methods

The proposed approach was compared with recent object detection methods such as Faster R-CNN and RetinaNet. To train the object detection methods, we used the plant position (x, y) as the center of the rectangle. The size of the rectangle corresponds to the size of the plant canopy (240 cm). We considered Green, Red, and NIR bands for this comparison. Similarly, an inverse process was used during the testing stage, obtaining the plant position from the center point of the rectangle predicted by the RetinaNet and Faster R-CNN methods.

Table 3.5 shows the results obtained by all methods using MAE, Precision, Recall, and F1 metrics. We can see that the proposed approach achieved better results for all metrics with 0.95 and 0.96 for Precision and Recall, respectively.

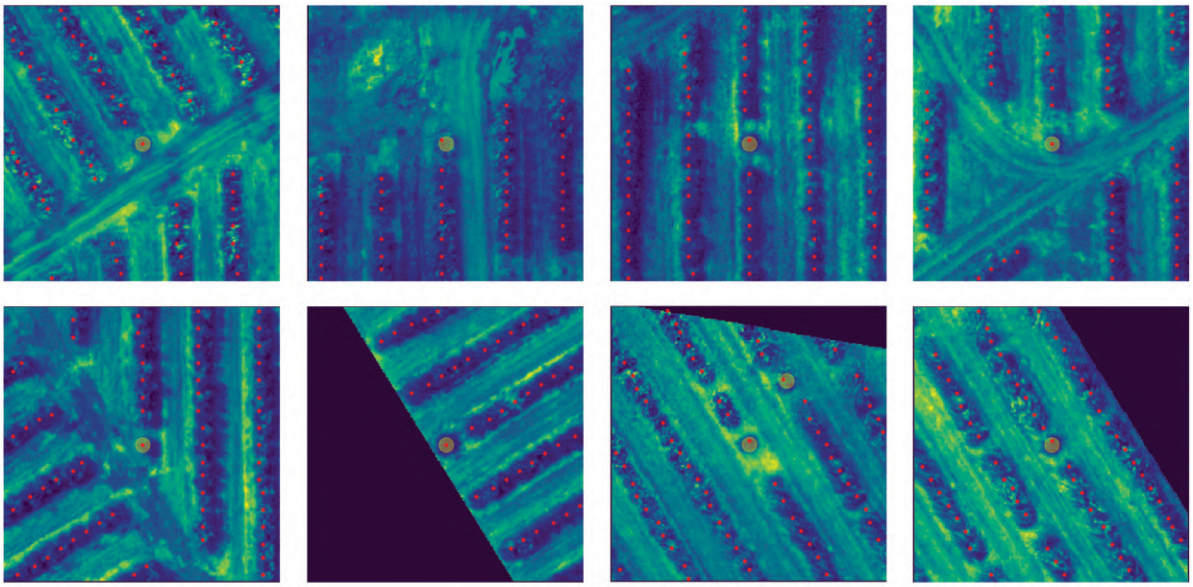


Figure 3.7: Examples of spaced trees correctly identified with the proposed approach.

In addition, the proposed approach achieved an MAE of 2.05 while RetinaNet and Faster R-CNN provided values of 30.87 and 37.85, respectively. RetinaNet and Faster R-CNN achieved only 0.74 and 0.54 for the F1-Measure, against 0.95 of the proposed approach. These results indicate that the proposed approach can predict citrus trees with high precision, having a very low number of false detections.

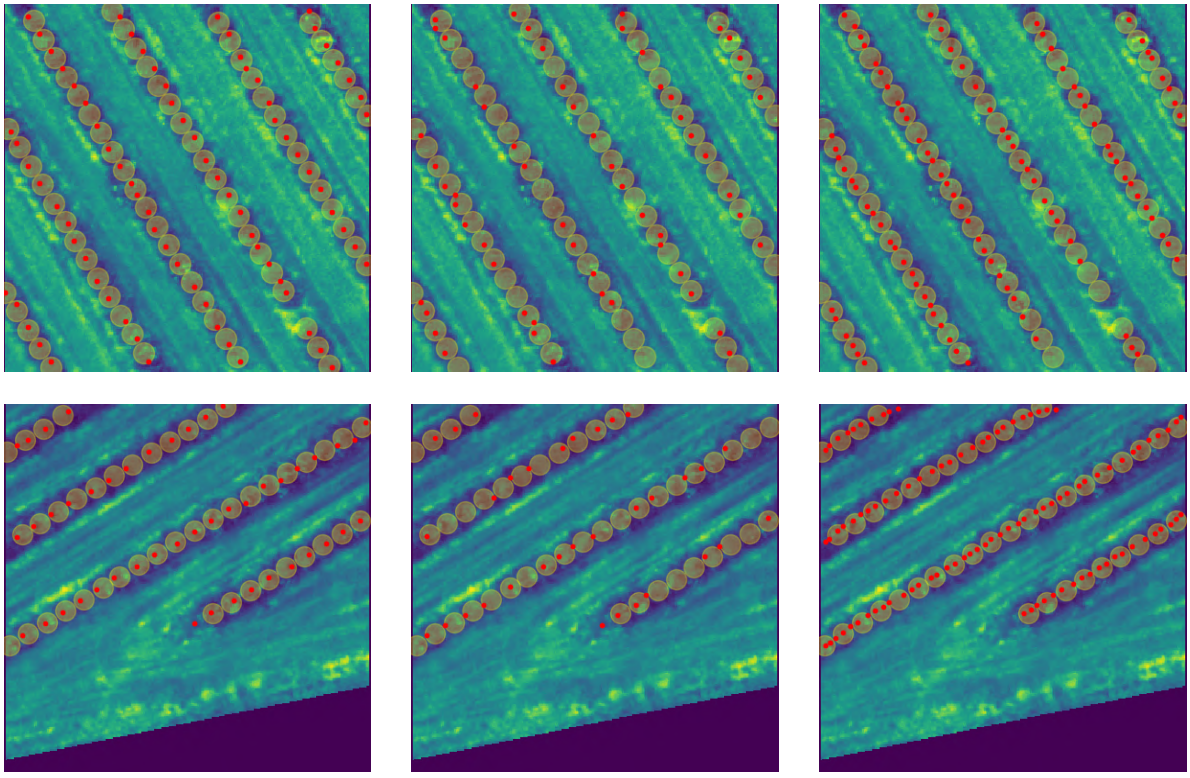
Table 3.5: Comparison of the proposed approach with recent object detection methods.

Methods	**MAE**	**Precision**	**Recall**	**F1-Measure**
RetinaNet	30.87	0.62	0.92	0.74
Faster R-CNN	37.85	0.86	0.39	0.54
Proposed approach	2.05	0.95	0.96	0.95

Figure 3.8 shows the visual results of the predictions generated by the three methods in two images. We can see that our approach has few errors in detecting plants. Faster R-CNN is the most misleading method, failing to identify plants in the images, while RetinaNet predicts more plants than those in the image, generating many false predictions. Note that the Precision reflects this behavior, being lower for RetinaNet than for Faster R-CNN since the number of false positives is higher for RetinaNet.

3.4.4 Computational Cost

Table 3.6 presents the computational cost of the proposed approach using different values for the number of stages, which is the main parameter influ-



(a) Our approach (b) Faster R-CNN (c) RetinaNet

Figure 3.8: Examples of the predictions generated by the three methods: (a) Our approach, (b) Faster R-CNN and (c) RetinaNet. Plant predictions are represented by red dots and plant regions are represented by yellow circles.

encing the size of the CNN. This table presents the average time in seconds (sec) to process an image with 256×256 pixels and three bands. In addition, table shows the estimated number of Frames Per Second (FPS) that the proposed approach is capable of processing.

Table 3.6: Evaluation of the computational cost of the proposed approach for different number of stages.

Stages (T)	**Time (sec)**	**FPS**
1	0.0039 ($\pm$0.0002)	258.26
2	0.0092 ($\pm$0.0006)	108.30
4	0.0215 ($\pm$0.0008)	46.49
6	0.0330 ($\pm$0.0010)	30.31
8	**0.0401 ($\pm$0.0012)**	**24.93**
10	0.0498 ($\pm$0.0015)	20.10

Still, one observation that must be noted is that, by increasing the number of stages, the computational cost also increases. Considering the best result that was obtained with eight stages (Table 3.4), the proposed approach is able to process approximately 25 images per second. The speed/accuracy trade-off

can be considered in the choice of the number of stages. If an application needs to run in real-time with more than 30 images per second, then four or six stages is a good alternative.

3.5 Remarks of the Chapter

In this chapter, we presented a CNN approach to estimate the number and location of citrus trees from UAV multispectral imagery. Our results archived 0.97 for R^2 and 0.036 trees for NRMSE. The combination of the spectral bands green, red and Near-Infrared produced better performance than the use of individual spectral bands. The method also demonstrated reasonable computational cost for embedded real-time applications. One of the advantages of our approach is in estimating a dense map to detect individual trees in high-density plantations, rather than the object-detection approach using rectangles to represent trees. The comparison against object-based methods returned a higher Precision (0.95) and lower MAE (2.05) for our method.

CHAPTER

4

Counting and locating high-density objects using convolutional neural network

4.1 Introduction

High-object density scenes are one of the biggest challenges for counting and locating objects. Object detection methods are, in general, not adequate for high-density scenes [Goldman et al., 2019]. In this scenario, overlapping objects are difficult to analyze due to the size of the instances and the standpoint of the scene. Thus, approaches that model the problem of counting objects with a density estimation has been defined as state-of-the-art solutions, and are providing interesting solutions for dense scenes such as crowds and densely packed objects [Goldman et al., 2019, Aich and Stavness, 2018]. In Goldman et al. [2019], the authors proposed a CNN-based detection method, using the bounding box, to cope with densely packed scenes. They considered a layer to estimate a quality score index and used a novel EM merging unit to solve the overlap ambiguities with this score. However, handling high-density objects in images is still a concerning issue, both in counting and locating objects.

Another problem regarding object count from detection frameworks is the need of detailed ground truth labeled data, which is hard to obtain at large-scales [Russakovsky et al., 2015]. Acquiring a large-scale annotated data is a time-consuming process. Because of that, approaches based on a lighter

weight image label is something that researchers have previously proposed [Zhang et al., 2018, Fiaschi et al., 2012]. Still, recent studies are implementing point annotations to reduce the supervision task [Aich and Stavness, 2018, Liu et al., 2019]. Point annotations are easier to obtain than BBoxs, and many counting and locating approaches do not need to rely on them to identify an object [Liu et al., 2019]. These types of approaches can rely on context information, and, for most problems, object instances will share a similar color, texture, and shape; meaning that the method will learn how to recognize them even if only using dot annotations [Aich and Stavness, 2018].

Recently, state-of-the-art methods to count objects include the VGG-GAP and VGG-GAP-HR [Aich and Stavness, 2018] approaches, LPN [Hsieh et al., 2017] and Deep IoU CNN [Goldman et al., 2019]. These methods were applied in counting and locating cars, crowds, biological cells and products from supermarket shelves, returning impressive performances in high-density scenes. Despite the promising results, scale variations, clutter background, occlusions, and especially high-density of objects are still challenges that hinder methods of providing high-quality predictions. That way, in previous work, we developed an initial model for the location and counting of Citrus-trees in UAV multispectral images [Osco et al., 2020a]. This initial model significantly surpassed methods for detecting objects such as RetinaNet and Faster-RCNN.

This chapter present a method for counting and locating objects based on convolutional neural networks [de Arruda et al., 2022]. The method is based on a density estimation map with the confidence that an object occurs in each pixel, following [Aich and Stavness, 2018]. Unlike previous works that estimate a bounding box for each object, the estimation of a density map allows a better refinement of the occurrence of objects in each pixel of the image. Different from previous work (Chapter 3), this method uses a feature map enhancement with a PPM [Zhao et al., 2017] that allows to incorporate global information at different scales. Consequently, the proposed method incorporates sufficient global context information for a good characterization of objects similarly to Zhang et al. [2019] with its hierarchical context module. Thus, in this chapter, we hypothesize that this approach is most suitable for situations of high object density, since it incorporates detection information in each pixel with the density map and improves this learning with regional information provided by the PPM module.

Another potential pitfall of previous methods is the missed detections due to object occlusion and high-density scenes. To compensate for these problems, and produce the correct predictions, we also propose a Multi-Sigma Stage (MSS) refinement over the ground truth to provide hierarchical learning of the object positions. The MSS refinement phase starts from a initial predic-

tion of the object position to a more refined on of the center of the object. Our hypothesis is that this refinement allows the method to provide more assertive predictions, decreasing the number of missed detections caused by occlusion and high-density scenes.

To verify the performance of the proposed approach, we performed experiments in three image datasets in two challenging applications. First, we perform a parameter evaluation in a tree counting dataset containing $3,370$ images and approximately $232,000$ objects. This dataset presents trees with irregular distribution and different growth stages, different from our previous research (Chapter 3). Once the best parameters were defined, we evaluated the generalization of the method in two car-counting benchmarks: CARPK and PUCPR+. For that, we evaluated the proposed method with 13 other state-of-the-art object detection methods.

4.2 Proposed Method

This is a improved approach for the baseline method described in Chapter 2 that uses a three-channel image, with $w \times h$ pixels, as input.The object counting and location is modeled after a 2D confidence map estimation, following the procedures presented in [Aich and Stavness, 2018]. We improved the confidence map estimation by including global and local information through a PPM [Zhao et al., 2017]. We also proposed a multi-sigma prediction phase to refine the confidence map to a more accurate prediction of the center of the objects.

Figure 4.1 illustrates the phases of the proposed method, which are detailed in the following section. Our approach is divided into four main phases: 1) feature map generation with a CNN (Section 4.2.1); 2) feature map enhancement with the PPM (Section 4.2.2); 3) Multi-Sigma Stage refinement of the confidence map (Sections 4.2.3 and 4.2.4); and, 4) object position obtention by peaks in the confidence map (Section 4.2.4).

4.2.1 Feature Map using CNN

The first part of the proposed approach extract a feature map F with a CNN based on the VGG19 [Simonyan and Zisserman, 2014], from a RGB input image (Figure 4.1 (a)), following Sections 2.1 and 2.2. The feature map is used to characterize the input image and allow the confidence map estimation for the object detection task.

In this application, we evaluated two variations of our method for different input images dimensions. The first variation receives an input image with 512×512 resolution and produces a feature map in the final layer with 64×64

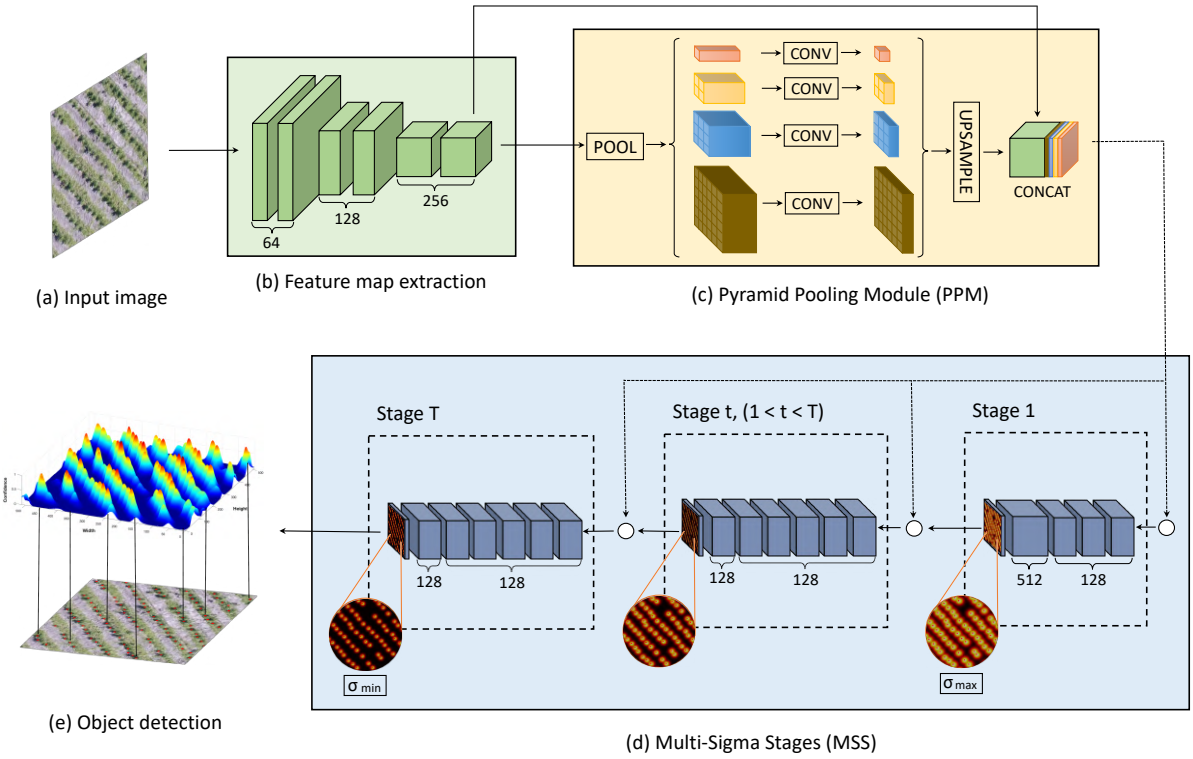


Figure 4.1: Our method for the confidence map prediction using the PPM and the MSS refinement approach. The initial part (b), based on VGG19 [Simonyan and Zisserman, 2014], extracts a feature map from the input image (a). This feature map is used as input for the PPM (c) [Zhao et al., 2017]. The resulting volume is then used as input to the first stage of a MSS phase (d) [Aich and Stavness, 2018]. The concatenation of the PPM and the prediction map of the previous stage is used as input for the remaining stages. The T stages apply a standard deviation (σ) for the confidence map peak, starting at maximum-to-minimum so that values are spaced equally.

resolution. Proportionally, the second variation receives an input images with 1024×1024 pixels, and the output feature map has a resolution of 128×128. Despite the low resolution, this map can describe relevant features extracted from the image.

4.2.2 Improving Feature Map with Pyramid Pooling Module

Many CNN cannot incorporate sufficient global context information to ensure a good performance in characterizing high-density objects. To solve this issue, our method adopts a global and subregional context module called PPM [Zhao et al., 2017]. This module allows CNN to be invariant to scale since it associates subregional and global information in the feature map. Figure 4.1 (c) illustrates the PPM that combines the features of four pyramid scales, with resolutions of 1×1, 2×2, 3×3 and 6×6, respectively.

The highest general level, shown in orange, applies a global max pooling which creates a 1×1 feature map to describe the global image context, such as the number of detected objects in the image. The other levels divide the input map into subregions, forming a grouped representation of the image with their subcontext information, as dense or sparse regions.

The levels of the PPM contain feature maps with various sizes. Because of this, we used a 1×1 convolution layer with 512 filters after each level. The feature maps are normalize with a Batch Normalization (BN) [Ioffe and Szegedy, 2015] step before the upsampling layer. We upsampled the feature maps to the same size as the input map with bilinear interpolation. Lastly, these feature maps are concatenated with the input map to form an improved description of the image. This step ensures that small object information is not lost in the PPM phase.

Although this module is proposed for semantic segmentation, it has proven to be a robust method for counting objects according to our experiments. The module allowed image information at different scales and its global context to be grouped with the feature map for a better description of the input image, improving the detection performance.

4.2.3 Refinement with Multi-Sigma Stages (MSS)

The Multi-Sigma Stage phase is based of the Section 2.2, where the improved feature map obtained by PPM is used as input for the T stages that estimates the confidence map. The first stage (Figure 4.1 (d)) receives the feature map and generates the confidence map C_1 by using five convolutional layers (described in Section 2.2). At a subsequent stage t (Figure 4.1 (d)), the prediction returned by the previous stage C_{t-1} and the feature map from the

PPM phase are concatenated. They are used to produce a refined confidence map C_t. The $T-1$ final stages consist of seven convolutional layers (described in Section 2.2).

Different from the baseline method, we proposed hierarchical learning of the center of the object. For that we use different σ values for each stage. The σ values varing from a maximum to a minimum value across the T stages (described in Section 4.2.4). In this way, the first stage generates initial position predictions, while the other stages refine the predictions (Figure 4.3). To avoid the vanishing gradient problem during the training phase, we adopted a loss function at the end of each stage as described in Section 2.2.

4.2.4 Generation of Confidence Maps and Object Localization

As mentioned in Sections 2.2 and 2.3, to train our method, a ground truth confidence map $\hat{\mathrm{C}}_t$ is generated by placing a 2D Gaussian kernel at each center of the labeled objects [Aich and Stavness, 2018]. Besides, the Gaussian kernel has a standard deviation (σ) that controls the spread of the confidence map peak, as shown in Figure 2.1.

Different from our previous research (Chapter 3), this approach uses different values of σ_t for each stage t to refine the object center prediction during each stage. The σ_1 of the first stage is set to a maximum value (σ_{max}) while the σ_T of the last stage is set to a minimum value (σ_{min}). The appropriate values of σ_{max} and σ_{min} are evaluated in the experiments. The σ_t for each intermediate stage is equally spaced between $[\sigma_{max}, \sigma_{min}]$. The early stages should return a gross prediction of the center of the objects, and this prediction is refined in the subsequent stages. In our experiment, the usage of different σ helped refine the confidence map, improving its robustness.

Finally, the object locations are obtained from the confidence map of the last stage (C_T). The peaks are estimate analyzing the 4-pixel neighborhood of each given location in the confidence map (described in Section 2.1). After preliminary experiment, we used $\tau = 0.35$ and $\delta = 1$ pixel, that's allows the detection of objects from two pixels of distances.

4.3 Experiments

4.3.1 Image Datasets

To test the robustness of our method, we evaluated it in a new and challenging dataset of eucalyptus tree images. We used this image dataset because there are different tree plantation densities, ranging from extreme cases to more sparsed trees (Figure 4.2). This variation in density is a challenge for

counting and locating objects. The trees were also at different growth stages. This permitted to evaluate the proposed method in different scales (tree size) and changes in appearance.

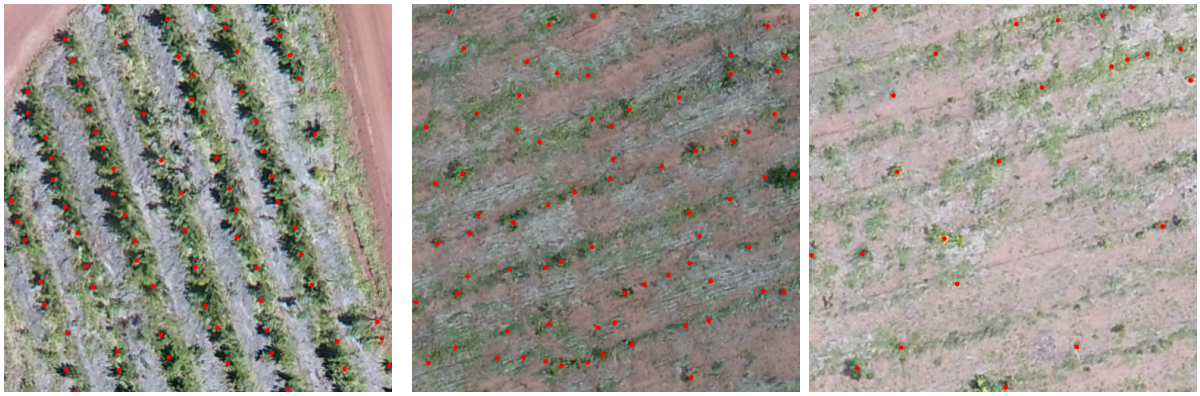


Figure 4.2: Examples of the tree dataset. The eucalyptus trees are at different growth stages and plantation densities.

The images were captured by an UAV in a rural property in Mato Grosso do Sul, Brazil, over four different areas of approximately $40\ ha$ each. The eucalyptus trees were planted at different spacing, the densest being at 1.25 meters from each other, with an average of $1,750$ trees per hectare. These trees were at different growth stages, variating between high and canopy areas. The images were acquired with an RGB sensor, which produced a pixel size of 4.15 cm. A total of four orthomosaic were generated from the area of interest. Approximately $232,000$ eucalyptus trees were labeled as a point feature by a specialist.

To evaluate the robustness and generability of the proposed approach, we also compared the performance of our method in two well-known image datasets for counting cars: CARPK and PUCPR+ benchmarks [Hsieh et al., 2017]. We compare the prediction metrics with state-of-the-art methods such One-Look Regression [Mundhenk et al., 2016], IEP Counting [Stahl et al., 2019], YOLO [Redmon et al., 2016], YOLO9000 [Redmon and Farhadi, 2017], Faster R-CNN [Ren et al., 2017], RetinaNet [Lin et al., 2020, Hsieh et al., 2017], LPN [Hsieh et al., 2017], VGG-GAP [Aich and Stavness, 2018], VGG-GAP-HR [Aich and Stavness, 2018] and Deep IoU CNN [Goldman et al., 2019].

4.3.2 Experimental Setup

The four orthomosaics were split into $3,370$ patches with 512×512 pixels without overlapping. These patches were randomly divided into training ($n = 2,870$), validation ($n = 250$) and testing ($n = 250$) sets. For training the CNN, we applied a Stochastic Gradient Descent optimizer with a momentum

of 0.9. To reduce the risk of overfitting, we used the validation set for the hyperparameter tuning on the learning rate and the number of epochs. After minimal hyperparameter tuning, the learning rate was 0.01 and the number of epochs was equal to 100. Instead of training the proposed approach from scratch, we initialized the weights of the first part with pre-trained weights in ImageNet. Six regression metrics, the MAE [Wackerly et al., 2014, Chai and Draxler, 2014], Root Mean Squared Error (RMSE) [Wackerly et al., 2014, Chai and Draxler, 2014], the R^2 [Draper and Smith, 1998], the Precision, Recall, and the F1-Measure, were used to measure the performance. Training and testing were perfomed in a deskop computer with Intel(R) Xeon(R) CPU $E3-1270@3.80$ GHz, 64 GB memory, and NVIDIA Titan V Graphics Card (5120 CUDA cores and 12 GB graphics memory). The methods were implemented using Keras-Tensorflow on the Ubuntu 18.04 operating system.

4.4 Results and Discussion

This section presents and discusses the results obtained by the proposed method while comparing it with state-of-the-art methods. First, we demonstrate the influence of different parameters, which includes the σ to generate the ground truth confidence maps, the number of stages necessary to refine the prediction, and the usage of PPM [Zhao et al., 2017] to include context information based on multiple scales. Second, we compare the results with a baseline of the proposed method. For this, we used the tree counting dataset and the car counting datasets (CARPK and PUCPR+).

4.4.1 Parameter Analysis

We present the results of the proposed method in the validation set for a different number of stages on the tree counting dataset. These stages are responsible for refining the confidence map. We observed that by using two stages ($T = 2$), the proposed method already returned satisfactory results (Table 4.1). When increasing to $T = 4$ stages, we obtained the best result, with MAE, RMSE, R^2, Precision, Recall and F1-Measure of 2.69, 3.57, 0.977, 0.817, 0.831, and 0.823, respectively. These results indicate the multi-sigma refinement affect the object counting tasks significantly. This is because the confidence map is refined in later stages, increasing the chance of objects be detect in high-density regions. Thus, we verified that the increase in the number of stages is decisive for a good refinement of the predictions. With $T = 6$ or more stages we see that the performance stabilizes and begins to decrease, due to the deepening of the layers.

We evaluated the σ_{min} and σ_{max} responsible for generating the ground truth

Table 4.1: Evaluation of the number of stages (T) on the validation set of the tree counting dataset using $\sigma_{min} = 1$ and $\sigma_{max} = 3$.

Stages (T)	**MAE**	**RMSE**	R^2	**Precision**	**Recall**	**F1-Measure**
2	2.86	3.82	0.974	0.809	0.825	0.816
4	**2.69**	**3.57**	**0.977**	**0.817**	**0.831**	**0.823**
6	3.48	4.61	0.962	0.805	0.836	0.819
8	2.90	3.79	0.974	0.816	0.823	0.818
10	3.32	4.25	0.967	0.789	0.796	0.790

confidence maps implemented in the T stages. In this experiment, we adopt $T = 4$ stages that achieved the best results from the previous experiment. The confidence map from the first stage is generated using σ_{max}, while the last stage uses σ_{min}, and the intermediate stages are constructed from values equally spaced between $[\sigma_{max}, \sigma_{min}]$. A low σ, relative to the object area (e.g., tree canopy) provides a confidence map without correctly covering the object's area. However, a high σ generates a confidence map that, while fully covers the object, may include nearby objects in high-density conditions. These conditions make it difficult to spatially locate objects in the image.

The evaluation for σ_{max} is presented in Table 4.2. The highest result was obtained with $\sigma_{max} = 3$, which best covers the tree-canopies without overlapping them. Still, we observed that other values for σ_{max} also returned good results. Since σ_{max} is used in the first stage, it does a small influence over the final result, since the confidence map is refined in subsequent stages.

Table 4.2: Evaluation of the σ_{max} in the validation set of the tree counting dataset. We adopted the $\sigma_{min} = 1$ and stages $T = 4$.

σ_{max}	**MAE**	**RMSE**	R^2	**Precision**	**Recall**	**F1-Measure**
2	3.31	4.31	0.966	0.811	0.837	0.822
3	**2.69**	**3.57**	**0.977**	**0.817**	**0.831**	**0.823**
4	3.21	4.24	0.968	0.804	0.816	0.809

The results for the σ_{min} are summarized in Table 4.3. The σ_{min} has great influence over the final result since it is responsible for the last confidence map. The overall best result was obtained with a $\sigma_{min} = 1.0$, which achieved a MAE, RMSE, R^2, Precision, Recall and F1-Measure of 2.69, 3.57, 0.977, 0.817, 0.831 and 0.823, respectively. This shows that the $\sigma_{min} = 1.0$ is the best fit for the size of the tree canopy. The conducted experiments showed that, with appropriate values of $\sigma_{max} = 3$ and $\sigma_{min} = 1$, high performance for counting trees can be obtained (Table 4.3).

To verify the potential of our method in real-time processing, we perform a comparison of the processing time performance for different amounts of

Table 4.3: Evaluation of the σ_{min} in the validation set of the tree counting dataset. We used $\sigma_{max} = 3$ and stages $T = 4$.

σ_{min}	**MAE**	**RMSE**	R^2	**Precision**	**Recall**	**F1-Measure**
0.5	11.01	13.77	0.658	0.868	0.721	0.783
0.75	2.93	3.89	0.972	0.820	0.831	0.824
1	**2.69**	**3.57**	**0.977**	**0.817**	**0.831**	**0.823**
1.25	3.05	4.01	0.970	0.815	0.822	0.817
1.5	2.94	3.73	0.975	0.818	0.810	0.813

stages (T). Table 4.4 shows the processing time of the proposed method for values of $T = 2$, 4, 6, 8 and 10. For this, we used 100 images from the tree test set and extracted the average processing time and standard deviation. We used the values of $\sigma_{min} = 1$ and $\sigma_{max} = 3$ that obtained the best performance in the previous tests. The results showed that the proposed approach can achieve real-time processing. For the best configuration with stages $T = 4$ the approach can deliver an image detection in 1.42 seconds with a standard deviation of 0.028.

Table 4.4: Processing time evaluation of the proposed approach for different amounts of T.

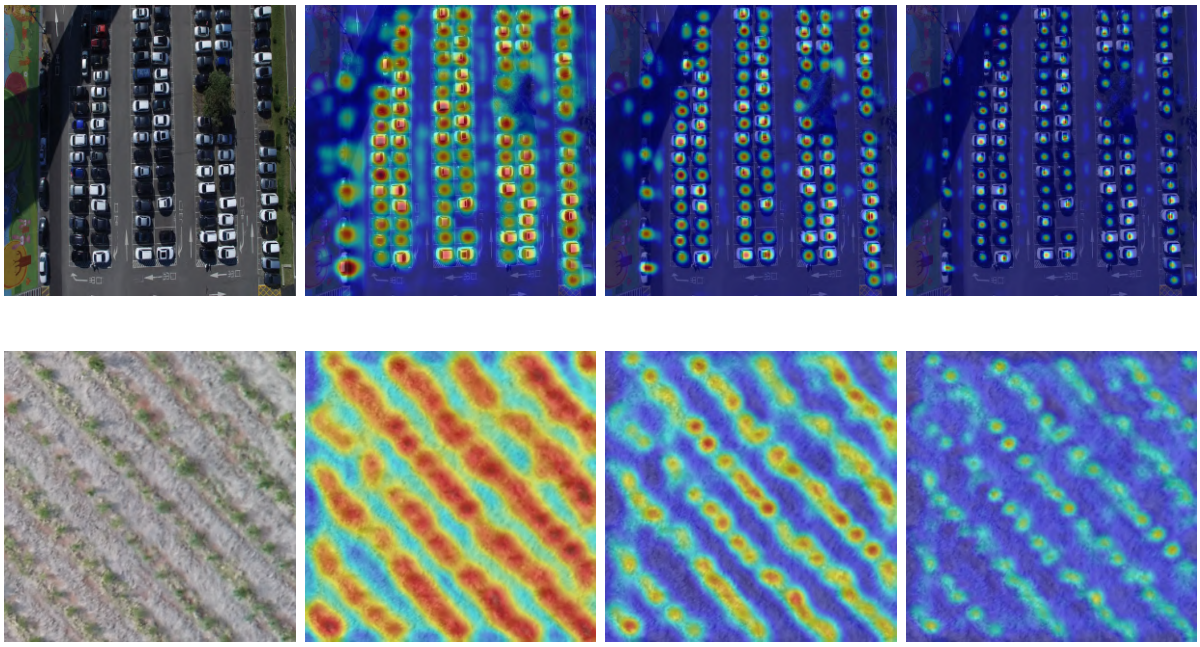
Stages (T)	**Average Time (sec)**	**Standard deviation**
2	0.802	0.022
4	1.426	0.028
6	2.063	0.058
8	2.675	0.059
10	3.373	0.100

4.4.2 Tree Counting

To analyse the design of the proposed architecture, we compared it with a baseline model that does not include the PPM and the MSS modules on tree-counting dataset. The overall best result with just the baseline of the CNN was obtained with a $\sigma = 1$, returning an MAE, RMSE, R^2, Precision, Recall, and F1-Measure equal to 2.85, 3.72, 0.977, 0.814, 0.833 and 0.822, respectively.

A gain in performance is observable when analyzing the results from the inclusion of the PPM and MSS in the baseline (Table 4.5). The inclusion of the PPM has no significant improvement for the results, while the baseline with multi-sigma refinement achieves better results. One explanation for this is that multiple stages provide hierarchical learning of the object position, refining the prediction of the center of the object across stages. Examples of the confidence map refinement across the stages are shown in Figure 4.3.

Besides, when we implemented both these two modules, it outperformed all the baselines results (Table 4.5). This performance gain can be explained by the sharing of the benefits that the two modules deliver, on the one hand the PPM module delivers subregional and global information in the feature map and the multi-sigma refinement uses this information to refine the objects predictions throughout the stages. The results shows that the combination of these two modules is essential to object counting.



(a) RGB Image (b) Stage 1 (c) Stage 3 (d) Stage 4

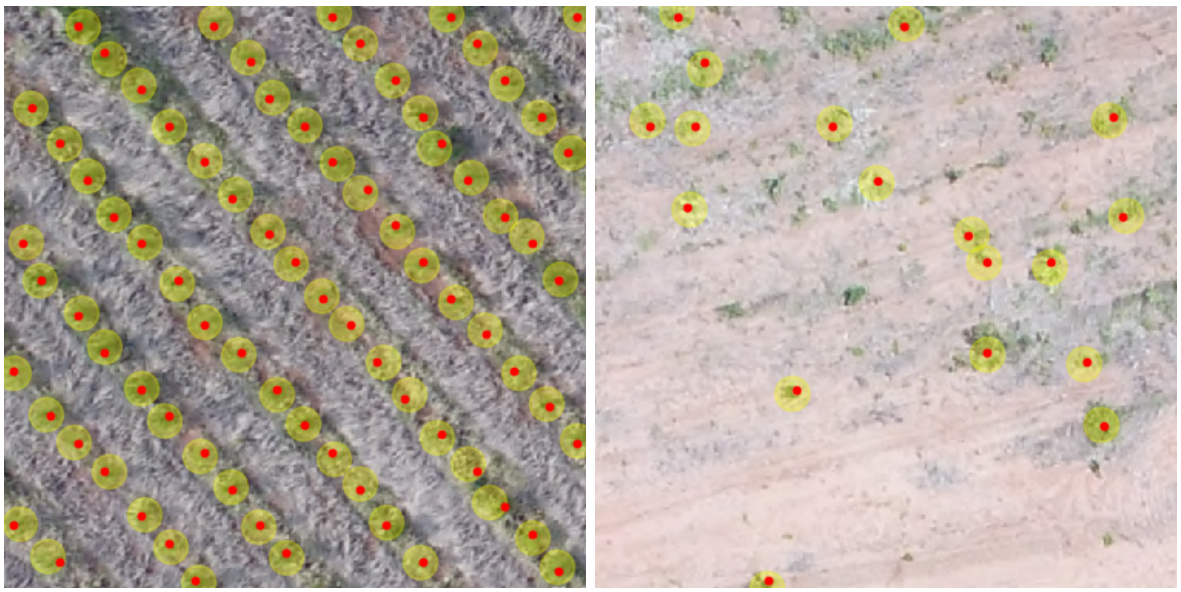
Figure 4.3: Example of two images showing the confidence map refinement by our method. The first column presents the Input image and the other columns present the activation map obtained in Stages 1, 3 and 4, respectively.

Table 4.5: Results of the proposed method and its baseline for the tree couting dataset.

Method	**MAE**	**RMSE**	R^2	**Precision**	**Recall**	**F1-Measure**
Baseline (σ = 0.5)	11.97	15.10	0.62	0.861	0.709	0.772
Baseline (σ = 1.0)	2.85	3.72	0.977	0.814	0.833	0.822
Baseline (σ = 2.0)	3.07	4.37	0.968	0.822	0.805	0.812
Baseline + PPM	2.44	3.38	0.981	0.825	0.836	0.829
Baseline + multi-sigma	2.78	3.64	0.978	0.808	0.833	0.819
Proposed Method	**2.05**	**2.87**	**0.986**	**0.822**	**0.834**	**0.827**

We considered a region around the labeled object position to analyze qualitatively the proximity of the prediction with the center of the object. The results using the best configuration ($\sigma_{min} = 1.0$, $\sigma_{max} = 3.0$, and $T = 4$) is displayed in Figure 4.4. The predicted positions are represented by red dots, and the tree-canopies regions are represented by yellow circles whose center

is the labeled position. The proposed method can correctly predict most of the tree positions. Another important contribution is that planting-lines are also learned without the need for annotation or additional procedure (see Figure 4.4 (a) and the expanded research in Chapter 6). Furthermore, the proposed method can correctly identify trees even outside the planting lines, in a non-regular distribution (Figure 4.4 (b)).

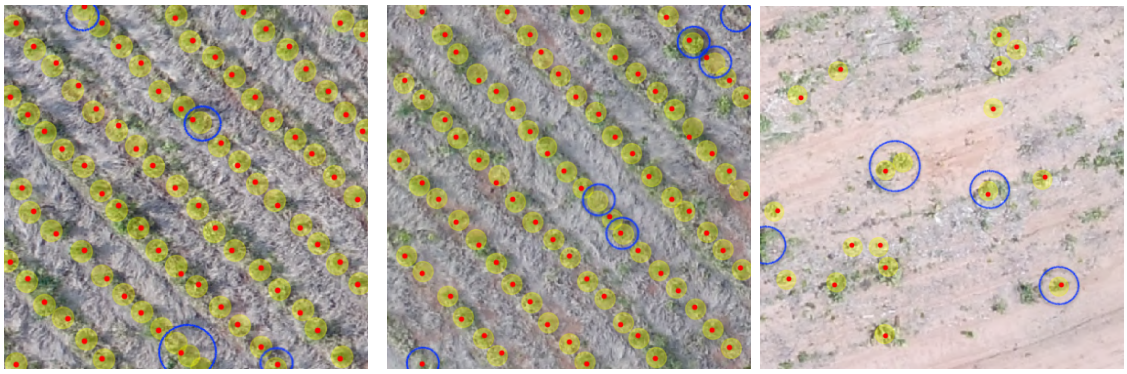


(a) Planting Lines (b) Non-regular Planting

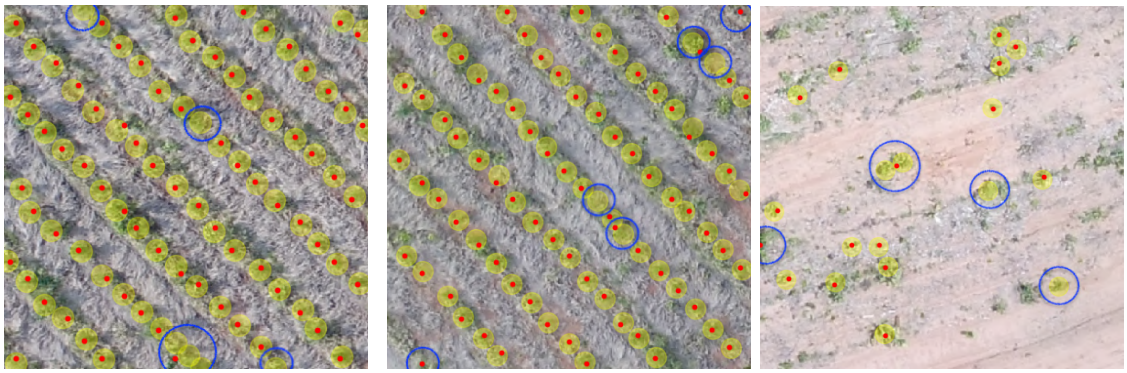
Figure 4.4: Comparison of predicted positions (red dots) in two images with different tree density. Predicted positions are represent by red dots while the tree-canopies regions are represent by yellow circles (centered in the labeled position).

A comparison of the proposed method with both PPM and MSS modules against the baseline is displayed in Figure 4.5. The baseline fails to detect some trees while returning some false-positives. The proposed method is capable of detecting more difficult true-positives, not detected by the baseline methods, with fewer false-negatives.

Although the proposed method returned a good performance for the tree counting dataset, it also had some challenges (Figure 4.6). The "far-from-center" predictions occurred in short planting-lines (Figure 4.6 (a)) or in disperse vegetation. This also happened in highly dense areas (Figure 4.6 (b)), although in fewer occurrences. Still, the proposed method was capable of predicting the correct position of the majority of trees.

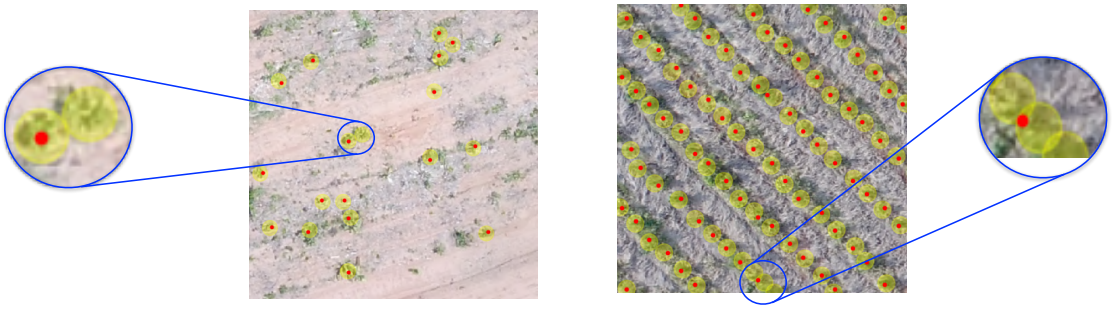


(a) Proposed Method



(b) Baseline

Figure 4.5: Comparison of the predicted positions of (a) the proposed method and (b) the baseline. Predicted positions are shown by red dots while tree-canopies are represented by yellow circles. Blue circles show the challenges faced by the methods.



(a) Short Planting Lines

(b) Canopy Overlap

Figure 4.6: Examples of the challenges faced by the proposed method.

4.4.3 Density analysis

To verify the performance of the proposed approach for object detection in different types of densities, we divided the tree dataset of 250 images into three density groups: low, medium and high. For this, the images were ordered ac-

cording to the number of trees annotated, then the three groups were defined based on the quantities of trees in a balanced way. The low corresponds to the images that have up to 52 plants, the medium between 53 and 78 plants, and the high above 78 plants. Thus, the sets of low, medium and high test images were left with 83, 90 and 77, respectively.

Table 4.6 presents the results obtained by the proposed approach at the three density levels. We can see that the approach does equally well at each density level, obtaining better results at the low level achieved an MAE, RMSE, R^2, Precision, Recall, and F1-Measure equal to 1.70, 2.34, 0.966, 0.818, 0.846 and 0.829, respectively.

Table 4.6: Results of the proposed method for different object densities.

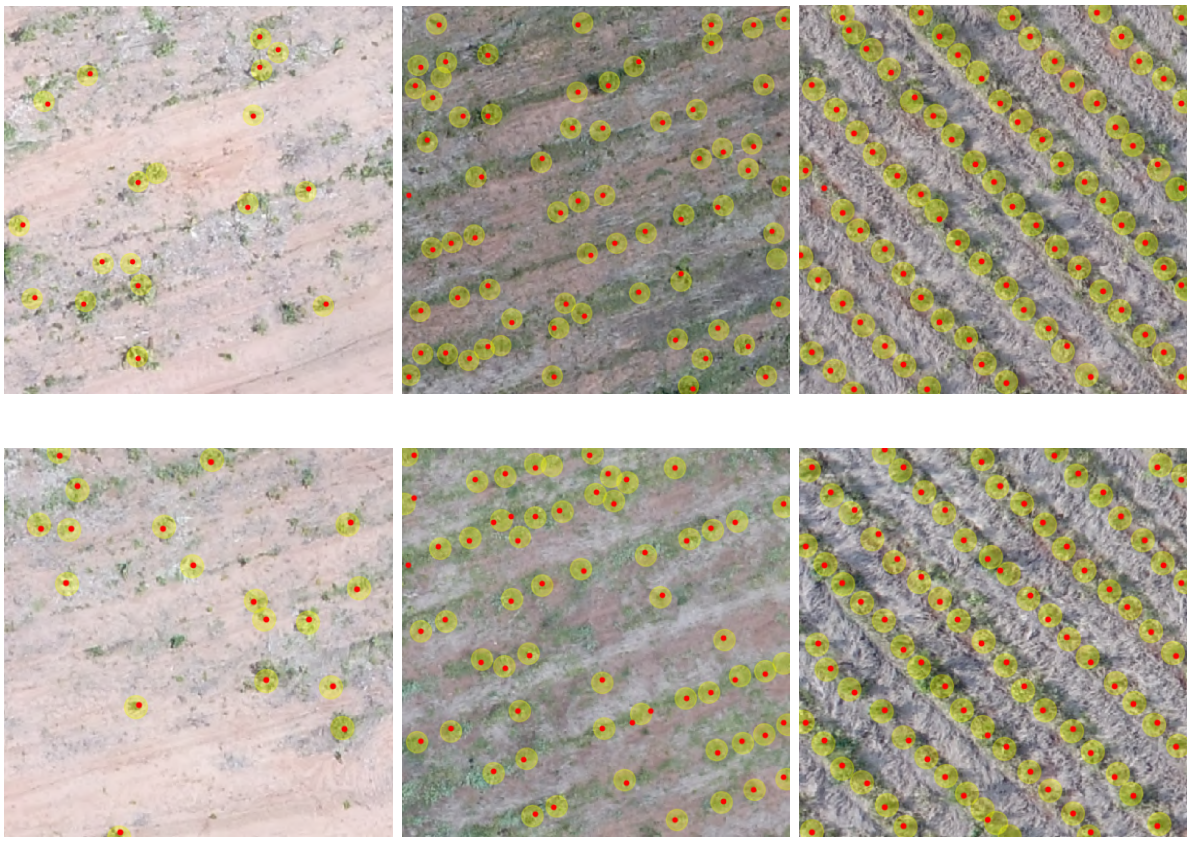
Density Level	**MAE**	**RMSE**	R^2	**Precision**	**Recall**	**F1-Measure**
Low	1.70	2.34	0.966	0.818	0.846	0.829
Medium	2.10	2.85	0.865	0.824	0.829	0.826
High	2.38	3.36	0.843	0.823	0.826	0.824

Figure 4.7 shows the visual results for plant detection at the three density levels. We can see that the proposed approach is able to correctly detect the centers of the plants, even in irregular plantings (see Figure 4.7 (a) and (b)). In addition, as shown in Table 4.6 we can see that at the low level the approach detects the plants positions more easily, since there is not much overlap of the tree-canopies.

4.4.4 Experiments on Cars Datasets

To generalize the proposed approach while comparing its robustness against other state-of-the-art methods, we evaluated its performance in two well-known benchmarks: CARPK and PUCPR+ [Hsieh et al., 2017]. These benchmarks provide a large-scale aerial dataset for counting cars in parking lots. We adopted the same protocols for the training and testing sets. The images have been resized from 1280×720 pixels to 1024×1024 pixels since we obtained similar performance when using full-resolution images in our approach.

To perform these experiments, we compare the proposed approach with state-of-the-art methods: One-Look Regression [Mundhenk et al., 2016], IEP Counting [Stahl et al., 2019], YOLO and YOLO9000 [Redmon et al., 2016, Redmon and Farhadi, 2017], Faster R-CNN [Ren et al., 2017], RetinaNet [Lin et al., 2020, Hsieh et al., 2017], LPN [Hsieh et al., 2017], VGG-GAP and VGG-GAP-HR [Aich and Stavness, 2018], Deep IoU CNN [Goldman et al., 2019], GSP [Aich and Stavness, 2019], Crowd-SDNet [Wang et al., 2021] and GAnet [YuanQiang et al., 2020].



(a) Low (b) Medium (c) High

Figure 4.7: Examples of the performance of the proposed approach at different levels of object densities. Column (a) shows the results for low densities, (b) for medium densities and (c) for high densities.

Experiments on CARPK dataset

The CARPK dataset [Hsieh et al., 2017] is composed of 989 training images ($42,274$ cars) and 459 test images ($47,500$ cars). The number of cars per image ranges from 1 to 87 in training images, and from 2 to 188 in test images.

Unlike the images of trees that we seek to cover its canopy, in the car images the confidence map seeks to cover the surface of the vehicle to correctly identify the objects. Table 4.7 presents the comparison with state-of-the-art methods. We can see that recent approaches such as Crowd-SDNet [Wang et al., 2021] and GAnet [YuanQiang et al., 2020] reached a MAE of 4.95 and 4.61, and an RMSE of 7.09 and 6.55, respectively. Traditional approaches such as Faster R-CNN, YOLO and RetinaNet achieved a MAE of 24.32, 45.36 and 16.62, and an RMSE of 37.62, 52.02 and 22.30. The proposed approach reached a MAE and an RMSE of 4.45 and 6.18, in addition it had a Precision, Recall and F1-Measure of 0.767, 0.765 and 0.763, respectively.

Similar to this work, Global Sum Pooling (GSP) [Aich and Stavness, 2019] also estimates an activation map indicating the positions of the objects. Al-

Table 4.7: CARPK comparative results.

Method	MAE	RMSE	R^2	Precision	Recall	F1-Measure
One-Look Regression	59.46	66.84	-	-	-	-
IEP Counting	51.83	-	-	-	-	-
YOLO v1	48.89	57.55	-	-	-	-
YOLO9000	45.36	52.02	-	-	-	-
Faster R-CNN	24.32	37.62	-	-	-	-
RetinaNet	16.62	22.30	-	-	-	-
LPN	13.72	21.77	-	-	-	-
VGG-GAP	10.33	12.89	-	-	-	-
VGG-GAP-HR	7.88	9.30	-	-	-	-
Deep IoU CNN	6.77	8.52	-	-	-	-
GSP	5.46	8.09	-	-	-	-
Crowd-SDNet	4.95	7.09	-	-	-	-
GAnet	4.61	6.55	-	-	-	-
Proposed Method	4.45	6.18	0.975	0.767	0.765	0.763

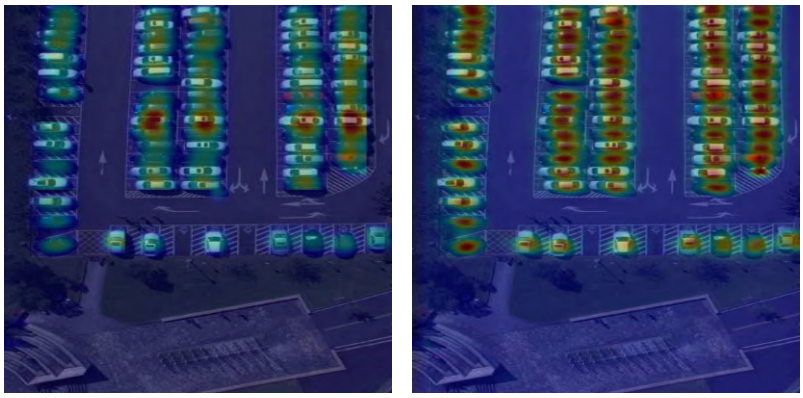
though it obtains relevant results, the proposed method delivers a gain of 1.01 and 1.91 for MAE and RMSE, respectively. In Figure 4.8 the visual comparison of the activations generated by the GSP and the proposed approach with its refinement in multiple stages is presented. We can observe that following the quantitative results the proposed approach delivers more refined predictions, achieving greater performance.

We observed that the proposed method achieved state-of-the-art performance in counting cars. As shown in Figure 4.9, the proposed method improves the results by detecting more difficult true-positives. Some cars are partially covered by trees or shadows (Figure 4.9 (a)) while others are partially occluded (Figure 4.9 (b)) at the edge of the images. Our method was able to detect such cases. The PPM helped improve the object representation, while the MSS module refinement provided a better position in the center of the objects. These features, incorporated in our approach, provide to be important additions for the detection of objects in these challenging scenarios.

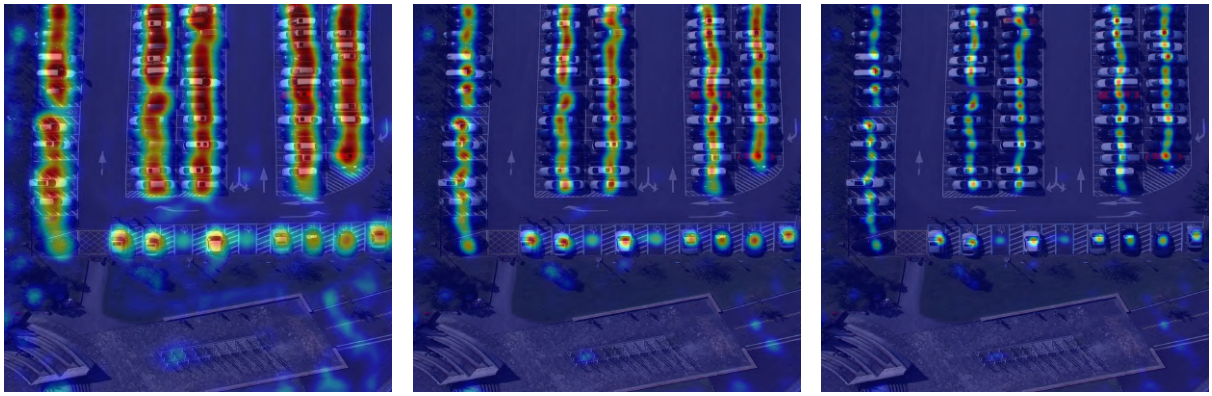
Experiments on PUCPR+ dataset

PUCPR+ [Hsieh et al., 2017] is a subset of the PUCPR dataset [de Almeida et al., 2015], and it is composed of 100 training images and 25 test images. The training and test images contain respectively $12,995$ and $3,920$ car instances.

Table 4.8 presents the comparison with 12 state-of-the-art methods for the PUCPR+ dataset. Again, we note that the approaches GAnet [YuanQiang et al., 2020] and Crowd-SDNet [Wang et al., 2021] reached a MAE of 3.28 and 3.20,



(a) GAP (b) GSP



(c) Ours (Stage 1) (d) Ours (Stage 3) (e) Ours (Stage 4)

Figure 4.8: Comparison of the activations generated by GAP and GSP approaches (first row) adapted from [Aich and Stavness, 2019], and by the multiple stages of refinement of the proposed approach (second row).

and an RMSE of 4.96 and 4.83, respectively. In the same way as observed for the CARPK dataset, the traditional approaches Faster R-CNN, YOLO and RetinaNet achieved intermediate performances with MAE of 39.88, 130.40 and 24.58, and an RMSE of 47.67, 172.46 and 4.58. This shows that traditional methods of object detection are not suitable for dense scenes. The proposed approach reached a MAE and an RMSE of 3.16 and 4.39, and obtained a Precision, Recall and F1-Measure of 0.832, 0.829 and 0.830, respectively.

Figure 4.10 presents the detections obtained by the proposed approach on the PUCPR+ dataset. Due to the point of view of the camera, the cars appear closer and distant in the same image. Thus, the results help to assess the generalization of the approach to recognize objects at different scales and with overlap (Figure 4.10 (a)). Since PPM adds multi-scale information to objects and multi-sigma refines detections, especially in highly dense areas, we see that the proposed approach achieves good detections even in these challenging scenes. Following the results in the CARPK dataset, the proposed approach achieves good performance in occlusion situations (Figure 4.10 (b)).

(a) Occlusion by trees and shadows

(b) Partial occlusion

Figure 4.9: Car detection by the proposed method on the CARPK dataset. Figure (a) shows the detections in scenarios of occlusions by trees and shadows, while figure (b) shows the cars partially hidden at the end of the image. Orange circles highlight challenging cases.

Table 4.8: PUCPR+ comparative results.

Method	**MAE**	**RMSE**	R^2	**Precision**	**Recall**	**F1-Measure**
YOLO v1	156.00	200.42	-	-	-	-
YOLO9000	130.40	172.46	-	-	-	-
Faster R-CNN	39.88	47.67	-	-	-	-
RetinaNet	24.58	33.12	-	-	-	-
One-Look Regression	21.88	36.73	-	-	-	-
IEP Counting	15.17	-	-	-	-	-
VGG-GAP	8.24	11.38	-	-	-	-
LPN	8.04	12.06	-	-	-	-
Deep IoU CNN	7.16	12.00	-	-	-	-
VGG-GAP-HR	5.24	6.67	-	-	-	-
GAnet	3.28	4.96	-	-	-	-
Crowd-SDNet	3.20	4.83	-	-	-	-
Proposed Method	3.16	4.39	0.999	0.832	0.829	0.830

4.5 Remarks of the Chapter

In this chapter, we proposed a new method based on a CNN which returned state-of-the-art performance for counting and locating objects with a high-

(a) Multiple distances (b) Partial occlusion

Figure 4.10: Car detection by the proposed method on the PUCPR+ dataset. Figure (a) shows the detections in scenarios from multiple distances between overlapping objects and figure (b) shows the cars partially hidden by trees and at the end of the image. Orange circles highlight challenging cases.

density in images. The proposed approach is based on a density estimation map with the confidence that an object occurs in each pixel. For this, our approach produces a feature map generated by a CNN, and then apply an enhancement with the PPM. To improve the predictions of each object, it uses a Multi-Sigma Stage refinement process, and the object position is calculated from the peaks of the refined confidence maps.

Experiments were performed in three datasets with images containing eucalyptus trees and cars. Despite the challenges, the proposed method obtained better results than previous methods. Experimental results on the CARPK and PUCPR+ indicate that the proposed method improves MAE, e.g., from 6.77 to 4.45 on CARPK and 5.24 to 3.16 on database PUCPR+. The proposed method is suitable for dealing with high object-density in images, returning a state-of-the-art performance for counting and locating objects.

CHAPTER

5

A Novel Deep Learning Method to Identify Single Tree Species in UAV-Based Hyperspectral Images

5.1 Introduction

The rapid development of lightweight sensors has contributed to the development of faster and more accurate techniques for the acquisition of surface information [Aasen et al., 2018]. In the last years, UAV platforms have been widely used for investigating forest health and monitoring [Nasi et al., 2015], biodiversity [Saarinen et al., 2018], resource management [Reis et al., 2019], and have become important tools for monitoring regions, improving flexibility and cost compared to spaceborne and airborne platforms. Besides, a recent review in forest remote sensing from UAV-based images showed that only 7% of the reviewed studies applied hyperspectral sensors in their analysis [Guimarães et al., 2020]. In the same study, the authors estimated that just 5% of the revised documents did make use of the spectral information of their data. Obtaining images with high-spectral resolution allows better monitoring of tree species, but can be challenging with individual trees, because adjacent branches and leaves can affect the individual tree recognition and their spectral signatures [Colgan et al., 2012].

The feature extraction in hyperspectral data was performed with conventional and machine learning algorithms like the Random Forest (RF), Decision Trees (DT), Support Vector Machine (SVM), Artificial Neural Network (ANN),

K-Nearest Neighbor (KNN), among others [Nevalainen et al., 2017, Xie et al., 2019]. The performance of these techniques has been evaluated in several studies and, for vegetation analysis, some achieved interesting results with a combination between them and remote sensing data [Maxwell et al., 2018, Pham et al., 2019]. For the individual tree detection and classification, a study was able to provide accuracies up to 95% using only shallow learners (i.e., conventional machine learning algorithms) and a combination of point-clouds with hyperspectral data [Nevalainen et al., 2017]. Another paper adopted object-based classification models like SVM and KNN to map mangrove species in hyperspectral and digital surface models achieving the best accuracy of almost 89% with SVM [Cao et al., 2018]. The CNN based approaches were recently applied to classify tree species using hyperspectral and RGB images [Nezami et al., 2020, Sothe et al., 2020]. Nezami et al. [2020] achieved 97.6% of accuracy in detecting the three tree species most common in Finish forests using CNN with hyperspectral, RGB, and structural data. Sothe et al. [2020] showed accuracies of almost 84% in detecting tree species in Brazilian ombrophilous forest with CNN and hyperspectral images only.

As the objects of interest in hyperspectral images usually have a greater complexity of characteristics, the parametric or conventional machine learning algorithms may not be the most suitable option. Thus, some researches started to implement deep learning in the remote sensing field [Nezami et al., 2020, Safonova et al., 2019, Li et al., 2017]. Deep learning-based methods are quickly gaining momentum in remote sensing approaches involving image segmentation and classification, change and object detection [Ma et al., 2019]. Generally, deep learning provided more accurate results when compared to traditional or shallow methods in situations in which a significant amount of data is available [Sothe et al., 2020, Khamparia and Singh, 2019]. Deep neural networks have been applied in environmental studies, some of which included single-tree species identification. Recently, published studies investigated state-of-the-art networks like YOLO v3 [Redmon and Farhadi, 2018], RetinaNet [Lin et al., 2017], and Faster-RCNN [Ren et al., 2015] to detect and segment tree-species in RGB imagery [Lobo Torres et al., 2020, Santos et al., 2019].

Despite the good performance of the methods mentioned above, some challenges of using hyperspectral images still remain. One of them is the Hughes phenomenon (the curse of dimensionality), which persists when dealing with small sample sizes [Belgiu and Dragut, 2016]. The high dimensionality of data could be problematic because an increased number of features may decrease its performance, as it introduces noise and sparsity in the feature space [Hennessy et al., 2020]. When applying a CNN, which is one of the most commonly

used deep learning architectures for image and pattern recognition [Alshehhi et al., 2017], data dimensionality reduction approaches are sure to be expected. For this purpose, either a PCA or mutual information is normally used [Audebert et al., 2019].

The hyperspectral data can deliver highly detailed views of objects according to their response to the analyzed spectral band. In many cases, it is common to use a band selection step to identify the bands that best characterize the object of interest [Bioucas-Dias et al., 2013]. A PCA [Richards John and Xiuping, 1999] is a common example of a band selection technique widely used in data analysis [Tuominen et al., 2018, Maschler et al., 2018, Liu et al., 2017]. The PCA is a linear scheme for reducing the dimensionality of high-dimensional data [Johnson and Wichern, 2015]. Still, PCA learns to reduce the spectral bands without considering the target position such as individual trees or any other information in a supervised manner. Therefore, with the growth in data volumes due to the large increase of spectral bands, more efficient methods are needed.

Another challenge related to remote sensing images of forested areas comes from the high-density of their environment. Most of the spectral divergences between trees and non-trees pixels are important because the brighter pixels are often recognized as the tree-crown, while darker pixels are viewed as indicative of their boundary [Ozcan et al., 2017, Csillik et al., 2018]. In highly-dense areas, this type of differentiation could be difficult, even for deep neural network-based approaches as some of them rely on bounding box [Santos et al., 2019, Ampatzidis and Partel, 2019]. In this manner, in a previous study, we developed a CNN based method to deal with highly-dense vegetation [Osco et al., 2020a]. In this study, however, we evaluated the performance of a primary version of our network to identify citrus-trees in an orchard. This method, implemented with data captured by a multispectral sensor in the UAV platform, significantly outperformed object detection methods based on Bounding Box estimation like RetinaNet and Faster-RCNN.

To fill part of the gap and challenges aforementioned, this chapter presents, a variation of the propose method [Miyoshi et al., 2020] for detect and geolocate single-tree species in a tropical forest with hyperspectral imagery. The approach was constructed to cope with a highly-dense scene while implementing a strategy to deal with the Hughes phenomenon. Differently from a PCA, which is considered a pre-processing step, we aim to estimate a combination of hyperspectral bands that most contribute to the mentioned task within the network's architecture. For this, we included the band selection phase as the initial step of our network. The phase learns from multiple combinations between bands which contributed the most for the tree identification task.

This is followed by a feature map extraction and the MSS module to refine the confidence map to produce an accurate result of the tree geolocation in a highly-dense scene.

5.2 Proposed Method

The proposed CNN method takes a hyperspectral image as input and computes the individual tree positions. The hyperspectral image has 25 bands with $w \times h$ pixels each. The tree identification and location are modeled as a 2D confidence map estimation, following the procedures related in [Aich and Stavness, 2018]. The confidence map is a 2D representation of the likelihood of a tree occurring in each pixel of the image (describes in Section 2.1). First, the hyperspectral images go through a band learning process before extracting the feature map. This allows the method to improve its accuracy by learning the best band combination for the trees detection. We included the Pyramid Pooling Module [Zhao et al., 2017] that uses global and local information to improve the estimation of the confidence map (describe in Section 4.2.2). Besides, we implemented a Multi-Sigma Stage prediction that refines the confidence map to a more accurate prediction of the center of the trees (describe in Section 4.2.3).

Figure 5.1 presents the approach for tree detection and geolocation. The method starts with a band-learning module that is responsible for learning m new bands from the hyperspectral image (Figure 5.1 (b)). Additionally, a feature map (Figure 5.1 (c)) is extracted using the output volume of the band-learning module. This feature map obtains global and local neighborhood information when passing through the PPM (Figure 5.1 (d)). The volume is then processed by a Multi-Sigma Stage module (Figure 5.1 (e)) with T stages to refine the tree detection. Finally, we obtain the tree's positions (Figure 5.1 (f)) at the end of the method.

5.2.1 Band learning machine module

To improve the band selection process of our network, we propose an end-to-end band learning module. This module receives a hyperspectral image with $w \times h$ pixels and 25 bands and learns m filters with size $1 \times 1 \times 25$ to generate an output image with dimensions $w \times h \times m$. Figure 5.2 illustrates an example of the application of the last filter, represented by the yellow color. Each filter is convolved through the input image (Figure 5.2 (a)) with a stride of 1 pixel, creating a corresponding output volume (Figure 5.2 (c)). During training, each filter has its weights adjusted to combine the bands that have more influence on the single-tree detection task. In this way, the layers that

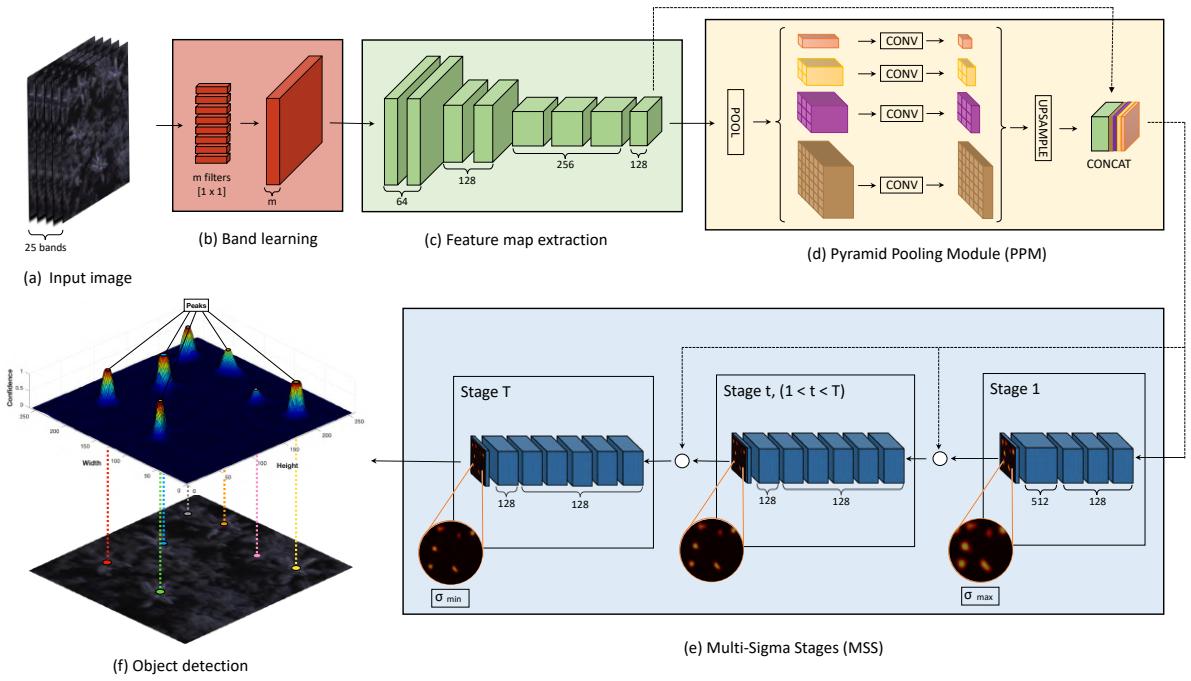


Figure 5.1: Proposed method for tree detection. The first layer (b) is responsible for the selection of the m best bands of the input image (a). The initial part of CNN (c) obtains the feature map of the input image. The feature map is used as an input to the PPM enhancement module (d). The resulting volume is used as input in the initial stages of the MSS phase (e). The T stages refine prediction positions until objects (f) are detected.

have more response in detecting objects will be enhanced, while the others will be discarded in the process.

5.2.2 Feature map extraction and tree localization

The Feature map extraction and tree localization follow the base method describe in Section 2.2 and 2.3. In this way, we firts extract the feature map using a CNN, based on VGG19, from the hyperspectral image. Then, we applied the PPM module (Section 4.2.2) that is responsible for characterize global and local information from the image. We use this module to make our method invariant to scale, because we expect improve the detection of the plants at different growth stages.

The proposed approach estimates the confidence map from the improved feature map generate by the PPM module. For that, the MSS phase refines the feature map for T stages (describe in Section 4.2.3). The goal is refine the confidence map with different values of σ in the T stages and improve the robustness. For that we use values of σ ranging from a maximum σ_{max} to a minimum value σ_{min}. In this aplication we addopted the minimum distance

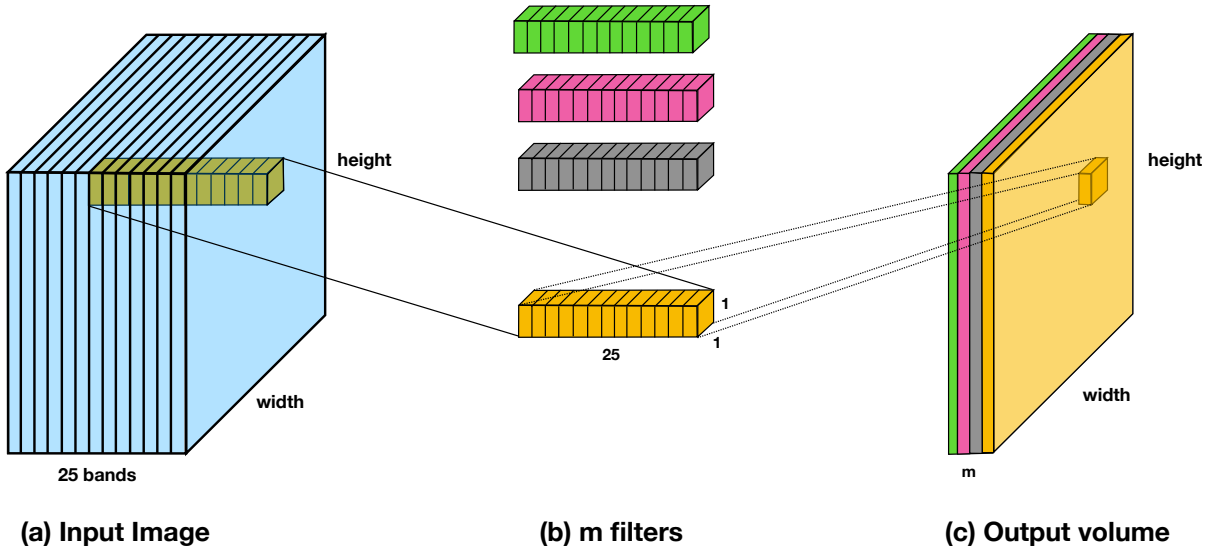


Figure 5.2: Band learning module structure. The multispectral image (a) is convolved with m filters with size $1 \times 1 \times 25$ (b) that generate an output volume (c) with m bands.

$\delta = 1$ pixel and the threshould $\tau = 0.35$, after preliminary experiments.

5.3 Experiments

5.3.1 Studied Area

To assess the proposed method, we used a transect area inside a forest fragment known as Ponte Branca (Figure 5.3). The Ponte Branca fragment is composed of a submontane semideciduous forest, which is part of the Black-Lion-Tamarin Ecological Station, in the countryside of the western region of the São Paulo state, in Brazil. The area has been protected by governmental laws since 2002 [Brasil, 2002, 2004] and suffered illegal logging until the end of the 1970s [Berveglieri et al., 2016]. From the 1970s to the 2000s, forest degradation was noticed in the northern part of Ponte Branca [Berveglieri et al., 2018], where the transect is located. In the transect area, more than 20 tree species were encountered [Takahashi Miyoshi et al., 2020, Berveglieri et al., 2018, 2016]. These species are considered as pioneers and secondaries tree species, with their majority considered within the primary degree of a regeneration state [Berveglieri et al., 2018].

From the tree species present in this area, *Syagrus romanzoffiana* is one key species since it is one of the most common palm trees in the Brazilian Atlantic forest [Giombini et al., 2017]. Palm trees can be considered as a key species in tropical forests because of its abundance of fruits and seeds and its importance for contributing to the forest structure [Elias et al., 2019, da Silva

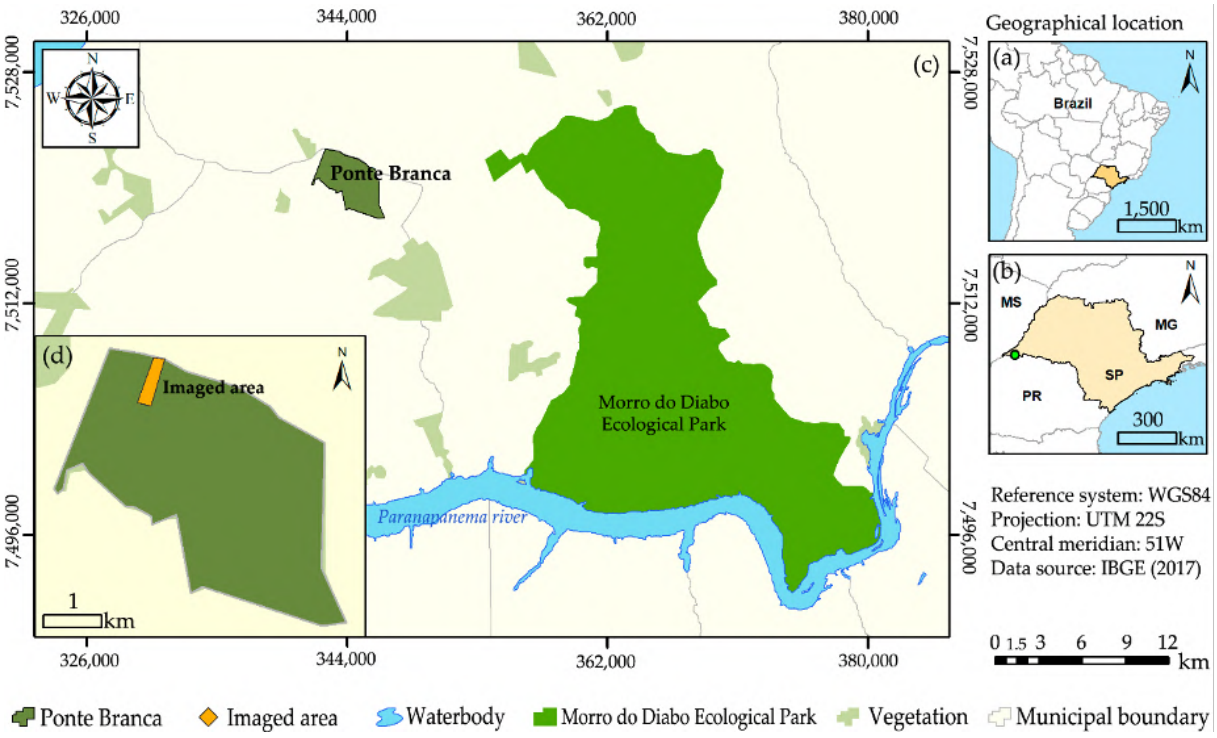


Figure 5.3: Study area in (a) Brazil, (b) São Paulo, (c) the western region of the São Paulo state, and (d) Ponte Branca forest fragment.

et al., 2011]. *Syagrus romanzoffiana* is an evergreen tree, tolerant to shadows, with great potential to be used for fauna restoration and conservation [Vieira et al., 2019], as blooms it produces fruits almost the entire year [Giombini et al., 2017, Lorenzi et al., 1992]. Its fruits are consumed by at least 60 different vertebrate species [Mendes et al., 2016], which may be an important factor in its high productivity. Besides, *Syagrus romanzoffiana* density can be related to the successional stage of forests in the area. According to the Brazilian Ministry of the Environment [Vieira et al., 2019], there is a higher number of *Syagrus romanzoffiana* samples in early secondary forests than in late secondary forests. In this manner, this tree species can be used as an indicator of forest regeneration. Aside from that, a higher frequency of *Syagrus romanzoffiana* indicates that the Atlantic forest in the initial stage of regeneration, where a lower frequency indicates a more preserved forest.

5.3.2 Image acquisition

The images that composed the dataset used were acquired on 16 August 2016, 01 July 2017, and 16 June 2018. They were acquired during the winter and dry season using a Rikola hyperspectral camera (Senop Oy, Oulu, Finland). The Rikola camera was onboard a UX4 UAV quadcopter (Nuvem UAV, Presidente Prudente, Brazil). This camera produces 25 spectral bands ranging from 506 nm to 820 nm, which were acquired over a transect area, presented

in Section 5.3.1 (Figure 5.3). Each image datacube is acquired by the two Complementary Metal Oxide Semiconductor (CMOS) sensors of the camera, both with 5.5μm of pixel size and frame format with 1017×648 pixels.

The flights were conducted 160 meters (mts) high above the ground with a speed of $4\ mts \cdot sec^{-1}$, providing images with a Ground Sample Distance equal to 10 cm, and forward and side overlaps higher than 70% and 50%, respectively. After the image acquisition, the dark current correction was performed with a dark image acquired before the flight campaign. In sequence, geometric processing was carried out in the Agisoft PhotoScan software (version 1.3) (Agisoft LLC, St. Petersburg, Russia) using initial Interior Orientation Parameters (IOPs) and Exterior Orientation Parameters (EOPs) from the Global Position Navigation (GPS) receiver of the camera. Additionally, during the bundle block adjustment process, three GCPs were used for each flight. The geometric process was carried out for the bands centered at $550.39\ nm$, $609.00\ nm$, $679.84\ nm$, and $769.89\ nm$ of each dataset, being the remaining ones estimated by the method developed in [Honkavaara et al., 2013, 2017]. The following products were created during this process: refined EOPs and IOPs; a sparse point cloud and a Digital Surface Model (DSM) of the area.

In a subsequent step, we used the EOPs, IOPs, sparse point cloud and DSM of the area for the radiometric block adjustment. This step is based on the methodology developed by [Honkavaara et al., 2013, Honkavaara and Khoramshahi, 2018] and aims to reduce illumination differences among images and to correct them from the Bidirectional Reflectance Distribution Function (BRDF) effects. The radiometric process was carried out in the radBA software [Honkavaara et al., 2013, Honkavaara and Khoramshahi, 2018] and uses common points among the images, the Sun position (i.e., the Sun zenithal and azimuthal angles), and the incident and reflected angles of each pixel. As the final product, we obtained the orthomosaics of each year radiometrically corrected. Moreover, the empirical line [Smith and Milton, 1999] was applied to transform the Digital Numbers (DN) into reflectance factor values. The empirical line parameters were calculated using three radiometric reference targets colored in light-grey, grey and black.

5.3.3 Experimental setup

The images were split into patches with 256×256 pixels without overlapping. The patches were randomly divided into training, validation and testing sets, in a proportion of 50%, 25%, and 25%, respectively. Figure 5.4 shows the images used to extract the training, validation and test sets in each year (2016, 2017, and 2018) and Table 5.1 shows the number of samples. It is noted the different number of samples for each year because of slight differences in

Table 5.1: Number of training, validation and test samples used in each experiment.

	2016	**2017**	**2018**	**Total**
Training	106	174	175	455
Validation	81	112	112	305
Test	79	116	116	311

the images acquisition. For training, we initialized the first part weights of our network with pre-trained weights on ImageNet and applied a stochastic gradient descent optimizer with a moment of 0.9. The validation set was used to adjust the learning rate and the number of epochs, reducing the risk of overfitting in our method. After the adjustments, the learning rate was set to 0.001 and the number of epochs was set to 100. The proposed approach was implemented in Python on Ubuntu 18.04 operating system and used the Keras-Tensorflow API. The workstation used for both training and testing has an Intel (R) Xeon (E) $E3-1270@3.80$ GHz CPU, 64 GB memory and an NVIDIA Titan V graphics card, that includes a 5120 CUDA cores and 12 GB of graphics memory. Lastly, to evaluate the performance of the approaches, we adopted three metrics: Precision, Recall, and F1-Measure [Story and Congalton, 1986].

5.4 Results and Discussion

5.4.1 Validation of the parameters

We first evaluate the influence of the proposed method parameters using only the validation images and reported the average F1-Measure of the three years. Parameters σ_{min}, σ_{max} and the number of stages, responsible for the refinement task in the density map prediction, were evaluated in the data displayed in Figure 5.5. From the F1-Measure shown in Figure 5.5 (a), $\sigma_{min} = 1$ obtained the best result. As show in Figure 5.5 (b), the best result for σ_{max} was 3, that is larger because it determines the density map of the first stage that is refined in the subsequent stages. The number of stages T ranged from 2 to 8 as shown in Figure 5.5 (c). We found that $T = 6$ achieved the highest overall F1-Measure. In this manner, the refinement step of our network used the following parameters: $\sigma_{min} = 1$, $\sigma_{max} = 3$, and $T = 6$.

The input images in the experiment have a total of 25 spectral bands. Our method can detect the combination of them contributed effectively to the tree detection task. We then evaluated the proposed convolutional layer for learning m linear band combinations in Figure 5.6. The experiment showed that the number of band combinations $m = 5$ reached the best F1-Measure of 0.939 against 0.892 when considering all the 25 spectral bands. The data shows that

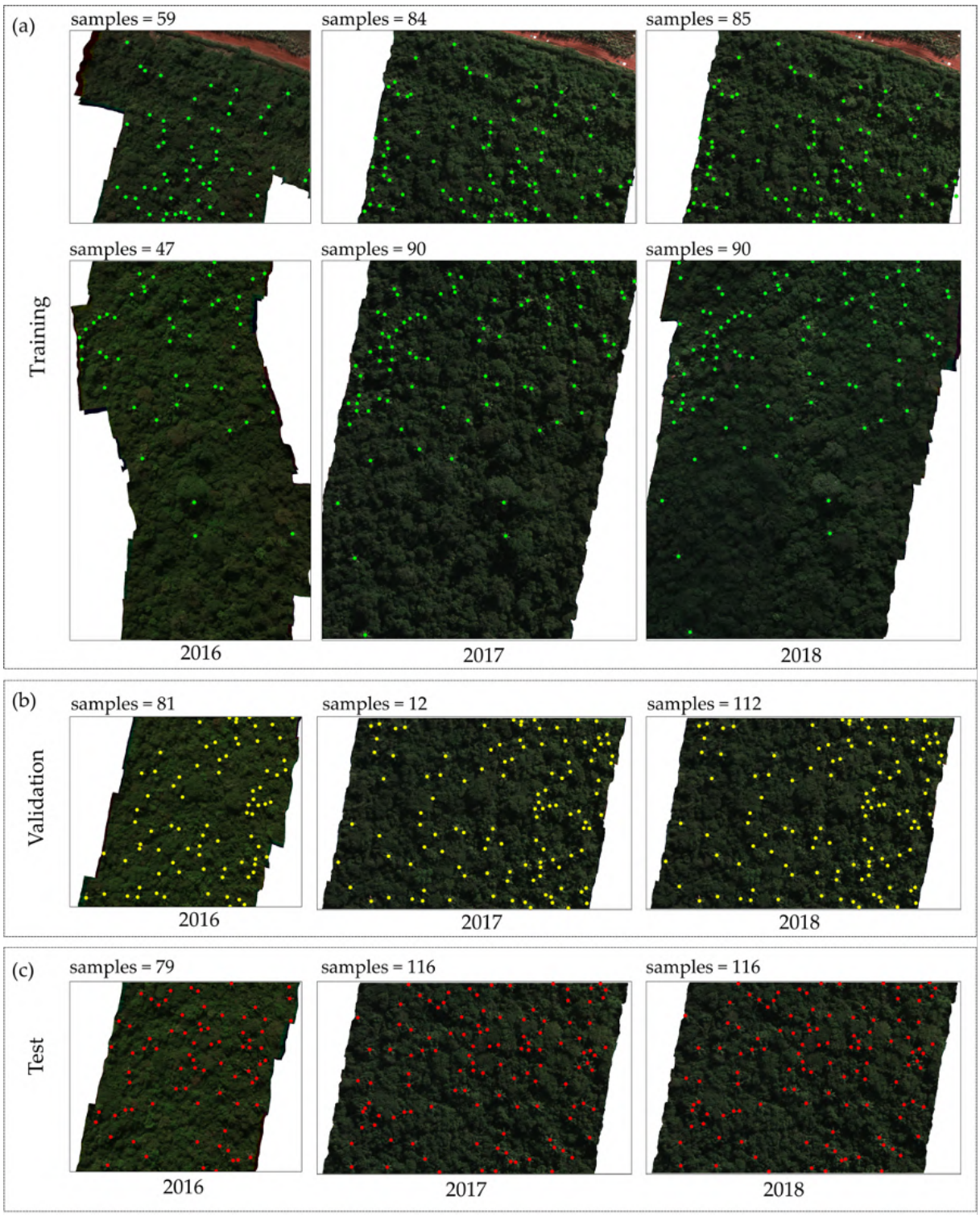
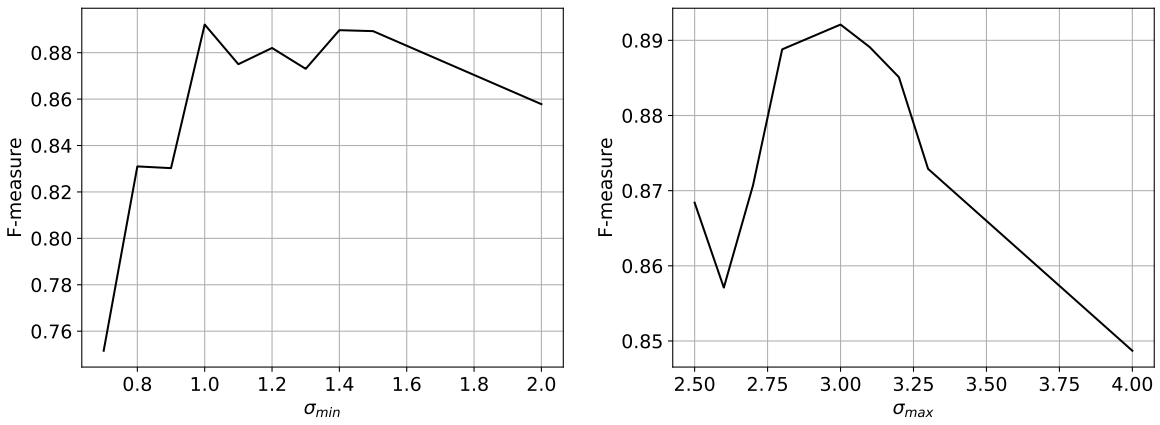


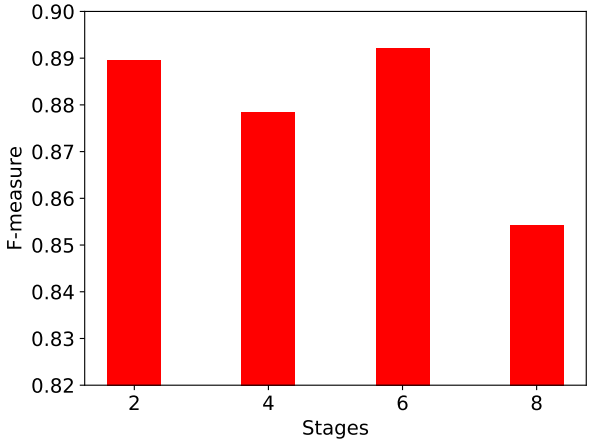
Figure 5.4: Image parts used for (a) training, (b) validation and (c) test in each year (2016, 2017, and 2018). The green, yellow and red dots represent the tree locations in the Train, Validation and Test dataset.

adding more linear combinations does not improve the results. These results confirm that the proposed layer appropriately combines which bands should be considered while avoiding the correlation and the scarcity that hinder most deep learning methods.

Figure 5.7 shows an example of the $m = 5$ linear band combinations. As



(a) (b)



(c)

Figure 5.5: Evaluation of (a) σ_{min}, (b) σ_{max}, and (c) number of stages T responsible for the refinement task in density map prediction.

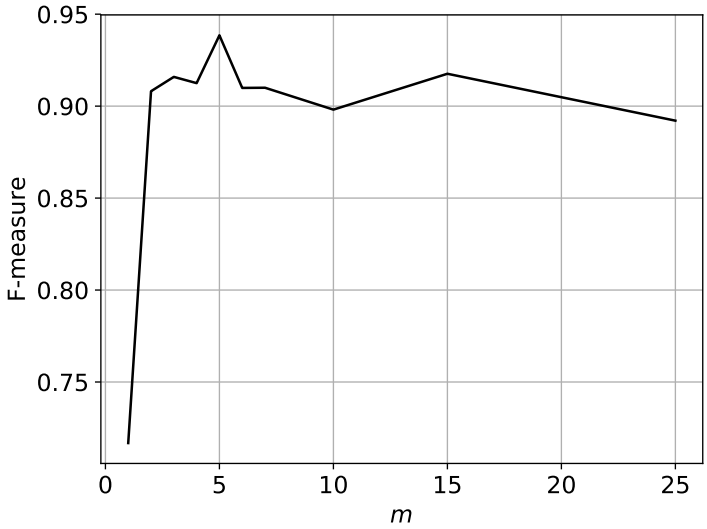


Figure 5.6: Evaluation of the number of linear band combination m.

displayed, these 5 new bands highlighted in blue the target of interest. The point in red represents the labeled ground truth. The values range from yellow to blue in the colormap, and our object of interest presents the highest values.

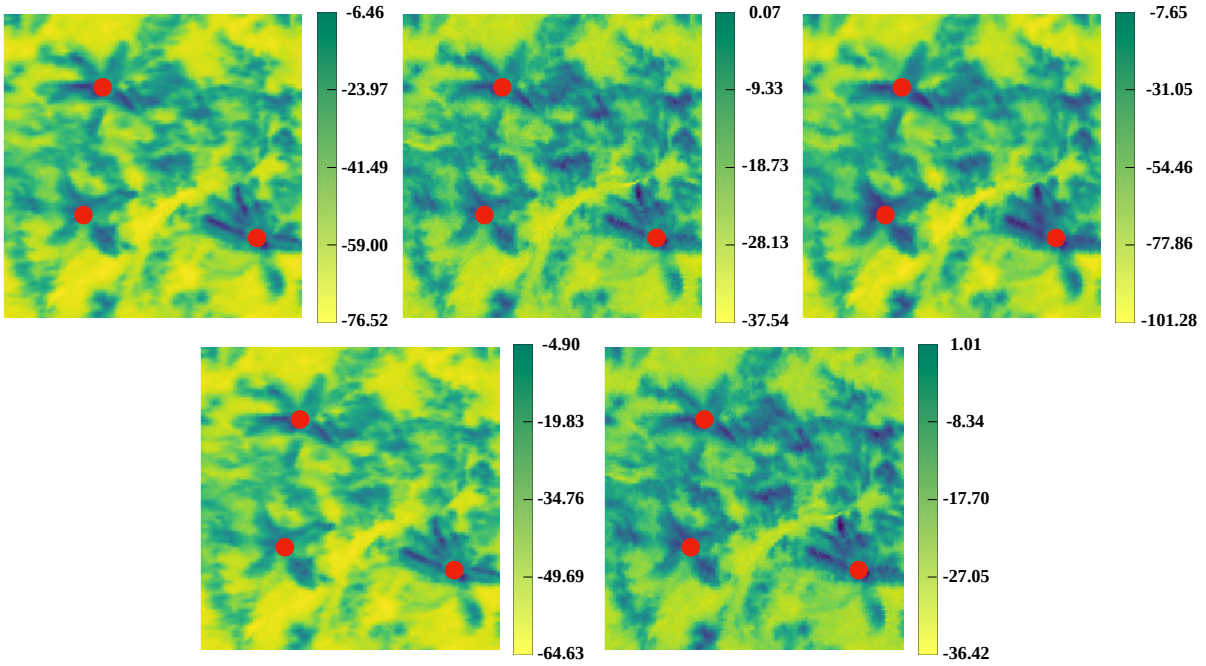


Figure 5.7: Example of the five linear band combinations obtained by the proposed method. The red dots represent the annotated trees.

5.4.2 Band Analysis

To determine the robustness of the band selection module as an initial step of our network, we performed a comparison with our network baseline (i.e., every step beyond the feature map extraction, Figure 5.1) and different inputs. One input consisted of all the 25 spectral bands, whereas the other input was composed of spectral bands obtained through a PCA approach.

Table 5.2 displays the overall precision, recall, and F1-Measure for the test images in the different scenarios described in the previous paragraph. By analyzing the precision values, it is evident that the baseline of our method in conjunction with the PCA spectral bands returned higher values when in comparison with the baseline plus all 25 bands. These precision values indicate that they do not have many false positives (i.e., do not detect trees incorrectly). When the recall values are analyzed, the proposed method with the band selection module is better than both approaches.

When considering the F1-Measure, viewed as the harmonic mean of precision and recall, it is observed that the use of all 25 bands was exceeded by the PCA (from 0.889 to 0.921). Compared to the baseline with the 25 spectral bands, the proposed method using five linear band combinations significantly

Table 5.2: Comparative results between the proposed method and PCA in the test images.

Method	Precision	Recall	F1-Measure
Proposed method (25 bands)	0.898	0.881	0.889
Proposed method + PCA	0.979	0.871	0.921
Proposed method	**0.973**	**0.945**	**0.959**

improved the F1-Measure; from 0.889 to 0.956. Besides, the supervised reduction of bands proposed here proved to be superior to the PCA method, with an increase of 3.8% in F1-Measure (from 0.921 to 0.959) and 7.4% in recall (from 0.871 to 0.945).

Figure 5.8 shows a qualitative results where the detected trees have a yellow circle (meaning true-positive) while undetected trees have a red circle (false-negative). The yellow dots indicate incorrect detection by both methods (false-positive). By implementing all bands, the network returned the worst results due to the redundancy of spectral information; corroborating with the Hughes phenomenon. The PCA improved the detection of trees (Figure 5.8 (b)) although it failed to detect a portion of them, which explains the low recall values when compared to the proposed method. As showcased here, the proposed method was able to detect the majority of trees correctly (Figure 5.8 (c)).

5.4.3 Discussion

The methodological contribution of our CNN based method is evident when comparisons, both quantitatively and qualitatively, are made (Figure 5.8 and Table 5.2). The implementation of a band selection module within our network's architecture not only reduces the amount of noise provoked by the dimensionality of hyperspectral data but also achieved better performance in the proposed task. A comparison with the PCA method, which is a common practice to reduce the number of bands needed, demonstrates the importance of adopting a method that considers the spectral information of the labeled object to select the right number of bands. This feature is not a common procedure for deep neural networks to consider within their architectures, and future methods could benefit from the module proposed here.

Although we have already applied the proposed approach on high-density scene [Osco et al., 2020a] this was the first time that we have used a heavily-dense forested environment and hyperspectral data. The PPM module and the MSS stage refinement are important phases since they produce a high-quality density map containing the object's location. This returns high predictions even when trees are located near each other. In this sense, these modules are

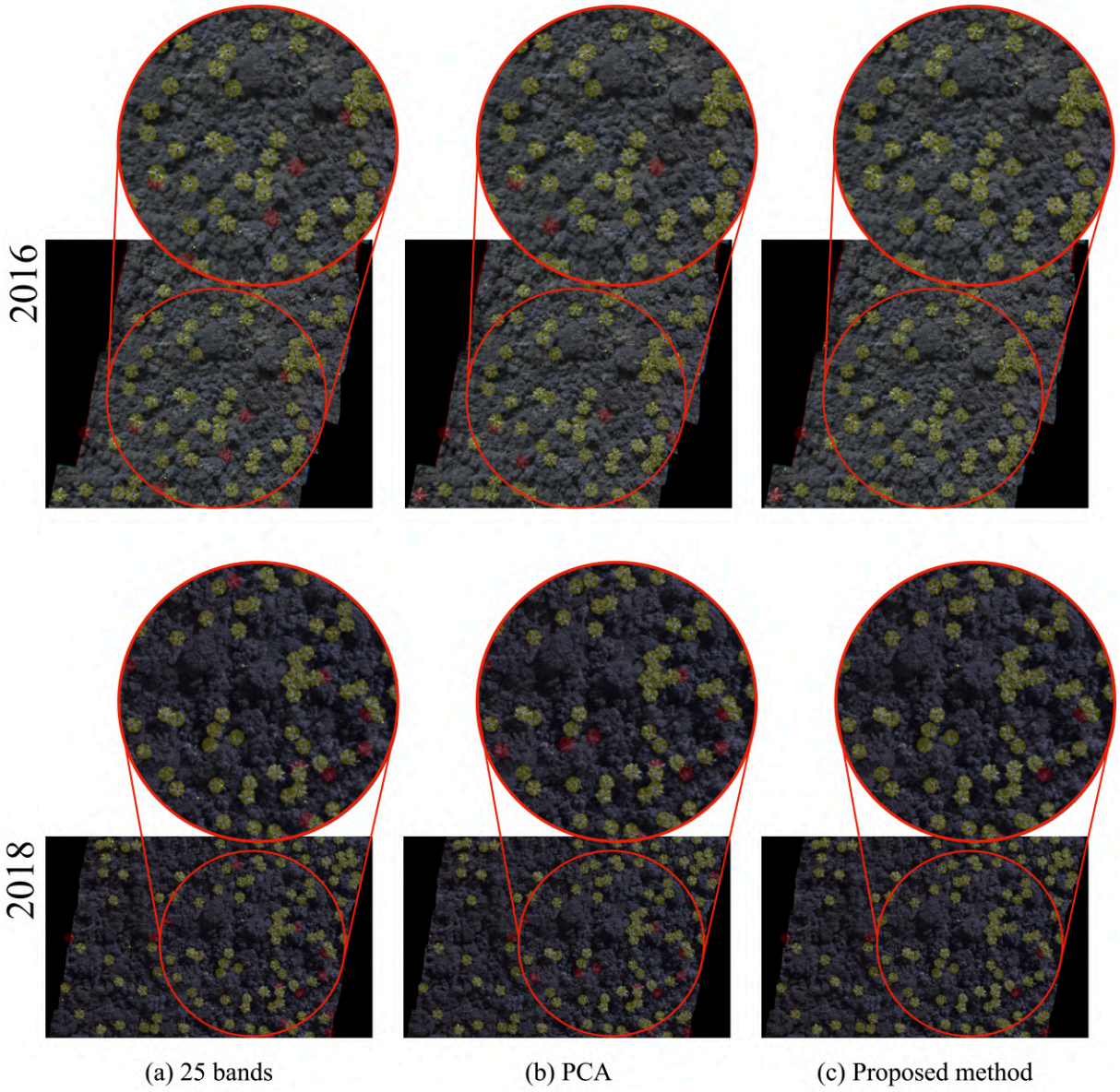


Figure 5.8: Qualitative results of tree detection using (a) all 25 bands, (b) PCA, and (c) proposed method for the years 2016 and 2018. The yellow circle indicates a true-positive detected tree, while undetected trees have a red circle (false-negative), and yellow dots indicate incorrect detection (false-positive).

important as they enable our method to predict both overlapping and isolated trees (Figure 5.8 (c)).

Bearing the results of the proposed network baseline in the detecting *Syagrus romanzoffiana*, it is highlighted the high F1-Measure value achieved (0.959 as shown in Table 5.2). Moreover, besides the developed method, the *Syagrus romanzoffiana* characteristics may assist this tree species identification. Results from Takahashi Miyoshi et al. [2020] showed the higher reflectance factor of this tree species when compared with the other seven tree species belonging to the transect area, especially in the near-infrared region of the electromagnetic spectrum. In this region, the vegetation response is mainly

affected by the leaf's cell structure [Jensen, 2000] and is an important region to tree species identification [Clark and Roberts, 2012, Dalponte et al., 2012]. Beyond that, there is the unique crown spatial distribution of *Syagrus romanzoffiana*. Its crown shape is like a star, while the other tree species has umbrella, oval, broad, or irregular shapes among others, not counting the difference in the existence of different layers in these crowns [Takahashi Miyoshi et al., 2020].

Lastly, when comparing the results with different researches that applied deep learning, it is noticed that they are consistent with ours. Sothe et al. [2020] showed a better performance of CNN than SVM and RF when identifying tree species from the ombrophilous dense forest. Safonova et al. [2019] found values of F1-Measures up to almost 93% when applying data augmentation and CNN in RGB images. Furthermore, Nezami et al. [2020] also achieved high precision and recall values (i.e., higher than 0.9) when identifying three tree species using a 3D-CNN. Using the Residual Neural Network (ResNet) and RGB images acquired with UAV over three years, Natesan et al. [2019] achieved an average F1-Measure value of 80% to identify three types of pine trees. The use of deep learning in RGB images is also shown by Santos et al. [2019] achieving an average precision of 92% in *Dipteryx alata* tree species identification. These accuracies demonstrate that our method, with an F1-Measure equal to 0.959 (Table 5.2), was also able to return state-of-the-art performance for the detection of tree species in a forest environment.

5.5 Remarks of the Chapter

In this chapter we presented a deep learning method, based upon a CNN architecture, to deal with high dimensionality data of hyperspectral UAV-based images to detect single-tree species. Our approach was constructed with a band selection feature in its initial step. This implementation within the network proved to be appropriate to deal with high dimensionality and was superior when compared with the baseline method considering all the 25 spectral bands and the PCA approach. Our CNN architecture is also followed by a feature map extraction and a MSS model refinement of the confidence map. The constructed architecture considers the possibility of every pixel in the image to be correspondent with an actual tree-species. This was important to produce accurate results in a highly-dense scene.

The proposed method returned a state-of-the-art performance for detecting and geolocating trees in UAV-based hyperspectral images, with an F1-Measure, Precision and Recall values equal to 0.959, 0.973, and 0.945 respectively. Differently from other current deep neural networks, our method esti-

mates a combination of hyperspectral bands that most contribute to the mentioned task within the network's architecture. The approach demonstrated here is important to deal with forest environment monitoring while providing accurate identification of single-trees.

CHAPTER

6

A CNN approach to simultaneously count plants and detect plantation-rows from UAV imagery

6.1 Introduction

Advances in both remote sensing and computational vision areas are improving agricultural landscape mapping in the past years [Weiss et al., 2020]. This integration is benefiting precision farming in several applications, such as environment control [Jr. and Daughtry, 2018]; phenology characterization [Wang et al., 2019a] nutrition evaluation [Delloye et al., 2018, Prado Osco et al., 2019, Osco et al., 2020b], yield-prediction [Chen et al., 2017, Hunt et al., 2019, Jin et al., 2018, 2019, Sun et al., 2019]; temporal analysis [Zhong et al., 2019], crop-management [Wang et al., 2019b] and others. With the intensification of food demand around the world, farmers are required to increase their efficiency. However, this increase in productivity must come from technological advances and optimization of the production areas instead of their expansion. An accurate estimation of the number of plants in crop fields is important to predict the amount of yield while monitoring growth status [Kitano et al., 2019]. Likewise, the detection of plantation-rows is essential since this information can be used by a specialist to evaluate the number of missed plants in each plantation-row and, consequently, the production rate of a crop [Oliveira et al., 2018]. These practices can help improve precision farming applications, resulting in better management of the agricultural system.

The visual inspection of plants in agricultural fields is a complicated task because it can be challenging and biased [Leiva et al., 2017]. Recently, data obtained from UAV-based sensors have been used to assist its management [Jiang et al., 2017, 2019, Deng et al., 2018]. Different sensing systems are used to map plants in high-resolution images, like UAV-based RGB, multi and hyperspectral cameras [Surovỳ et al., 2018, Ozdarici-Ok, 2015, Paoletti et al., 2018], LiDAR [Verma et al., 2016, Hartling et al., 2019], Synthetic Aperture Radar (SAR) [Ndikumana et al., 2018, Ho Tong Minh et al., 2018] and airborne imagery [Li et al., 2016]. Sensors such as LiDAR and SAR, although returns a high performance in plant detection [Jakubowski et al., 2013, Tao et al., 2015], are high-priced and difficult to reproduce in low-budget models. To circumvent this, recent studies regarding plant or tree density estimation have implemented RGB-based sensors in their applications [Weinstein et al., 2019, Csillik et al., 2018, Fan et al., 2018, Varela et al., 2018, Ampatzidis and Partel, 2019]. The low cost and high market availability associated with them may justify this preference. Furthermore, in the computational vision context, RGB images are enough for straightforward identification tasks such as plant detection [Wu et al., 2019, Hassanein et al., 2019].

The automatic identification of plants is generally divided into two categories: detection and delineation [Ozcan et al., 2017]. For detection purposes, both plant-size and spatial resolution of the image can be considered enough features [Csillik et al., 2018]. Delineation, however, may require information regarding spectral heterogeneity between the scene's objects, shadow complexity, and background effects (e.g. soil brightness) [Nevalainen et al., 2017]. In the past years, morphological operations and segmentation algorithms like "Watershed", "Valley Following", and "Region Growing" were used to count plants in both forested [Larsen et al., 2011] and cultivated areas [Ozcan et al., 2017]. The aforementioned techniques rely mostly on the spectral divergence between the pixels (plant and non-plant), indicating that a brighter pixel is recognized as the plant, while dark pixels (viewed as shadows) represent their boundary. In such cases, an Excess Greenness Index (ExG) could be applied to individualize the green pixels with high saturation from the background [Varela et al., 2018], or Object-Based Approaches (OBIA) [Hussain et al., 2013] or the use of Fourier-transformations [Jensen, 1996] and Gray Level Co-Occurrence Matrix (GLCM) textural metrics [Huang et al., 2014]. Another technique could be the conversion from RGB to grayscale Hue-Saturation-Value (HSV) image data [Oliveira et al., 2018]. These methodologies obtained interesting results in the last decade. Still, in recent years, more robust and intelligent algorithms are being created and tested in these applications, like DL-based models to promote a more generalized approach.

DL is one type of Machine Learning (ML) technique, based on ANN, adopting a deep strategy for data representation [Ghamisi et al., 2017, Badrinarayanan et al., 2015], resulting in a large learning capability and improved performance [Ball et al., 2017], in which several components and types of layers constitute a DL architecture [Kamilaris and Prenafeta-Boldu, 2018]. The most frequently used architectures in the past years are Unsupervised Pre-Trained Networks (UPNs), Recurrent Neural Networkss (RNNs), and Convolutional Neural Networks [LeCun et al., 2015, Khamparia and Singh, 2019]. In recent years, the CNN architectures presented great performances for image and pattern recognition, especially in remote sensing approaches [Alshehhi et al., 2017]. These approaches can be majorly separated into spectral, spatial, and spectral-spatial information extraction [Ghamisi et al., 2017, Li et al., 2017, Zhang et al., 2017]. When considering both spectral and spatial information, the model accuracy can significantly improve [Zhang et al., 2017], which is important since it helps to address the appropriate approach to solve specific problems.

In vegetation detection and delineation, DL models have been used to identify weed in bean and spinach fields [Bah et al., 2018], count palm trees in plantation areas [Djerriri et al., 2018, Li et al., 2017], classify urban-trees species [Santos et al., 2019, Hartling et al., 2019], tree crown prediction in forest areas [Weinstein et al., 2019], counting of rice seedlings [Wu et al., 2019], identification of citrus-tree crowns [Osco et al., 2020a, Csillik et al., 2018], tobacco plant detection [Fan et al., 2018], fir-trees insect-damage detection [Safonova et al., 2019], and others. The models implemented in these studies were mostly derived from the RNN and CNN architectures, some with modified versions of previously published algorithms, while others presenting an entirely new model. Regardless, a recent revision paper indicated that proximally 42% of the implemented architectures in agricultural studies were based on CNNs, as AlexNet, VGG16, and Inception-ResNet, compared to other deep learning architectures, like Recurrent Neural Networks and Recursive Neural Networks [Kamilaris and Prenafeta-Boldu, 2018].

For the detection of plants, different architectures and modifications were applied in state-of-the-art studies. CNNs adopting the architectures AlexNet [Krizhevsky, 2014] and Inception (v2, v3, and v4) [Szegedy et al., 2015, 2016, Szegedy et al., 2017] had been recently used to count sorghum plants [Ribera et al., 2017]. The modification of the mentioned architectures allowed them to estimate the number of plants using regression instead of classification. A modified version of the VGG16 model [Simonyan and Zisserman, 2014] was used to identify tree health status [Sylvain et al., 2019]. Likewise, a U-Net [Ronneberger et al., 2015] modification was used to detect palm

trees [Freudenberg et al., 2019]. A CNN region-based, YOLOv3 [Redmon and Farhadi, 2018], was used to recognize citrus trees and classify its crown [Ampatzidis and Partel, 2019]. The YOLOv3 architecture, alongside the RetinaNet [Lin et al., 2020] and Faster R-CNN [Ren et al., 2015] were also used to classify tree species [Santos et al., 2019]. Lastly, a modified Deep Convolutional Network (DCN), considering morphological operations and watershed segmentation, was efficiently used to detect tobacco plants [Fan et al., 2018].

The aforementioned methods returned important information regarding the tested approaches in a diverse number of agricultural fields. One type of crop that is still not benefited by these methods is corn. Corn (*Zea mays L.*) is an important crop and it is largely cultivated in countries like the United States of America, Brazil, China, Canada, and others [Mohanty and Swain, 2019]. For the detection of corn-plants in UAV imagery, few studies have been conducted with this computer vision approach [Mochida et al., 2018]. An early-season uniformity detection with a decision tree algorithm in an object-detection approach was used to identify corn plants in ultra-high-resolution imagery acquired with UAVs [Varela et al., 2018]. Experiments related to drought stress were evaluated through a DCN with images obtained from a stationary station [An et al., 2019]. Regarding the use of CNN in UAV imagery, a study applied the U-Net architecture to segment corn-plants from other field objects [Kitano et al., 2019]. This study, however, comments on issues that should be addressed in future research, such as earlier growth stages and higher plant density. Another unexplored issue, although not mentioned in these studies, is the detection of plantation-rows.

Another type of agronomic culture that could benefit, both from counting plants and detecting plantation-rows, is Citrus. Citrus tree detection is an important prerequisite for farmers and technicians to estimate yield and even, in some cases, compensate plantation gaps. Recently, studies already discussed the importance of DL in citrus tree counting and area estimation [Osco et al., 2020a, Ampatzidis and Partel, 2019, Csillik et al., 2018]. Our previous research (described in Chapter 3), conducted with UAV-based multispectral imagery, as well as the others in RGB imagery, returned similar outcomes in this task. However, to the best of our knowledge, a deep learning architecture capable of counting plants and mapping plantation-rows simultaneously for different cultivars is, still, a challenging and unproposed-task. A model with these capabilities can be used as an alternative to the visual interpretation task of crops-fields and could contribute to the sustainable management of agricultural systems. Some crops, such as citrus plants, corn, and many others, have a limited capacity to compensate for missing areas within a row since they cannot occupy those areas or at least lean towards them, and

this negatively impacts the yield per unit land area during the harvest season [Primicerio et al., 2017, Varela et al., 2018, Oliveira et al., 2018, Hassanein et al., 2019].

Additionally, performing the counting plant task in high-density areas is a problem even more challenging for both visual inspection and automatic analysis. Few investigations were conducted to solve counting plant tasks in high-density plantations [Osco et al., 2020a, Fan et al., 2018], and in most of these investigations, only one cultivar has been considered. It should also be noted that the detection of plants and plantation-rows consists of an important metric for the assessment of agricultural fields [Primicerio et al., 2017, Hassanein et al., 2019]. The number of plants helps farmers and rural technicians to estimate the yield at the end of the crop cycle [Oliveira et al., 2018]. This type of assessment, when performed in the early stages of planting, is important for rapid decision making. For corn and other types of cultivars, the decision window is brief, and a rapid detection may help to mitigate or prevent problems with its production. In citrus orchards, the counting of trees is also used to estimate yield and can help farmers to better monitor gaps in their plantation-rows.

In this regard, aiming to contribute to the aforementioned issues, this chapter present a new deep learning architecture to simultaneously count plants and detect plantation-rows for distinct cultivars from UAV imagery [Osco et al., 2021]. Our approach is based on a CNN in which its architecture is formed by two processing branches that share information (concatenated) for counting plants and detecting plantation-rows. This allows the refinement of plant detections to the regions where the plantation lines were detected, similarly, the plantation lines learn and adjust to the positions in which the plants are. In our approach, we used RGB imagery to compose a dataset since it is a lower-priced solution than most of the remaining remote sensing systems, being easily replicable in other situations. The framework discussed demonstrates a viable solution with computer vision assistance to count and detect plants and plantation-rows for different types of crops (i.e. corn and citrus) and plantation densities while preserving their geolocation information in UAV-based RGB imagery.

6.2 Materials and Method

The framework proposed in this chapter was composed of five steps: (1) The acquisition of RGB images from corn and citrus fields with a camera embedded in a UAV platform. (2) The pre-processing of images and their labeling by a specialist in a Geographical Information System (GIS) environment. Here,

the specialist defined the plantation-rows using a line-feature and the corn-plants and the citrus-trees with point-features. (3) The data splitting into training, validation, and testing datasets using the hold-out method. (4) The model's performance evaluation with the detection of the plantation-rows and the number of plants. (5) The error metrics calculated for each task performed by our CNN method.

6.2.1 Study Area and Data

The study was firstly conducted with corn (*Zea mays L.*) plants in an experimental area undertaken at "Fazenda Escola" at the Federal University of Mato Grosso do Sul, in Campo Grande, MS, Brazil. The evaluated area has approximately $7,435\ mts^2$, with corn-plants (both young and mature) planted at a 30×50 cm spacing, resulting in 4-to-5 plants per square meter. The plantation-rows consist of two different lengths and directions. We considered plants in two growth stages: Corn Recently Planted (V3) and; mature-stage with cobs. The processed image was labeled by a specialist in the QGIS 3.10 open-source software. Firstly, the plantation-rows were detected using a line feature. Secondly, each line was scanned by the specialist and the corn plants were manually identified by a point feature. Figure 6.1 shows a high-scale example of the resulting process.

We collected the corn images with a Phantom 4 Advanced (ADV) UAV for two days, using an RGB camera equipped with a 1-inch 20-megapixel CMOS sensor. The images were obtained with 80% longitudinal and 60% lateral overlaps, with a GSD of 1.55 cm. The images were processed with Pix4D commercial software. We optimized the interior and exterior parameters of the acquired images and generated the sparse point cloud based on the Structure-From-Motion (SfM) method. We then create the dense point clouds using the Multi-View Stereo (MVS) technique. The UAV flight was approved by the Department of Airspace Control (DECEA), which is responsible for Brazilian airspace.

The second experiment was conducted in a citrus orchard (*Citrus Sinensis Pera*), located in the countryside at the Boa Esperança do Sul municipality, SP, Brazil. The area is composed of citrus-trees at their maturity phase, in which the spacing in-line was initially around 3 meters from each-others and in the later years, more recent trees were planted at a more approximate area; thus, returning different spacing in-line and densities. The area has approximately $10.000\ mts^2$. We used an X7 - Spire II UAV embedded with an RGB sensor at 80 meters flight altitude, returning a GSD equal to 2.28 cm. To preprocess these images, we adopted the same strategy mentioned previously using the Pix4DMapper software. Figure 6.2 illustrates both areas (corn and citrus) with their respective locations and RGB imagery examples.

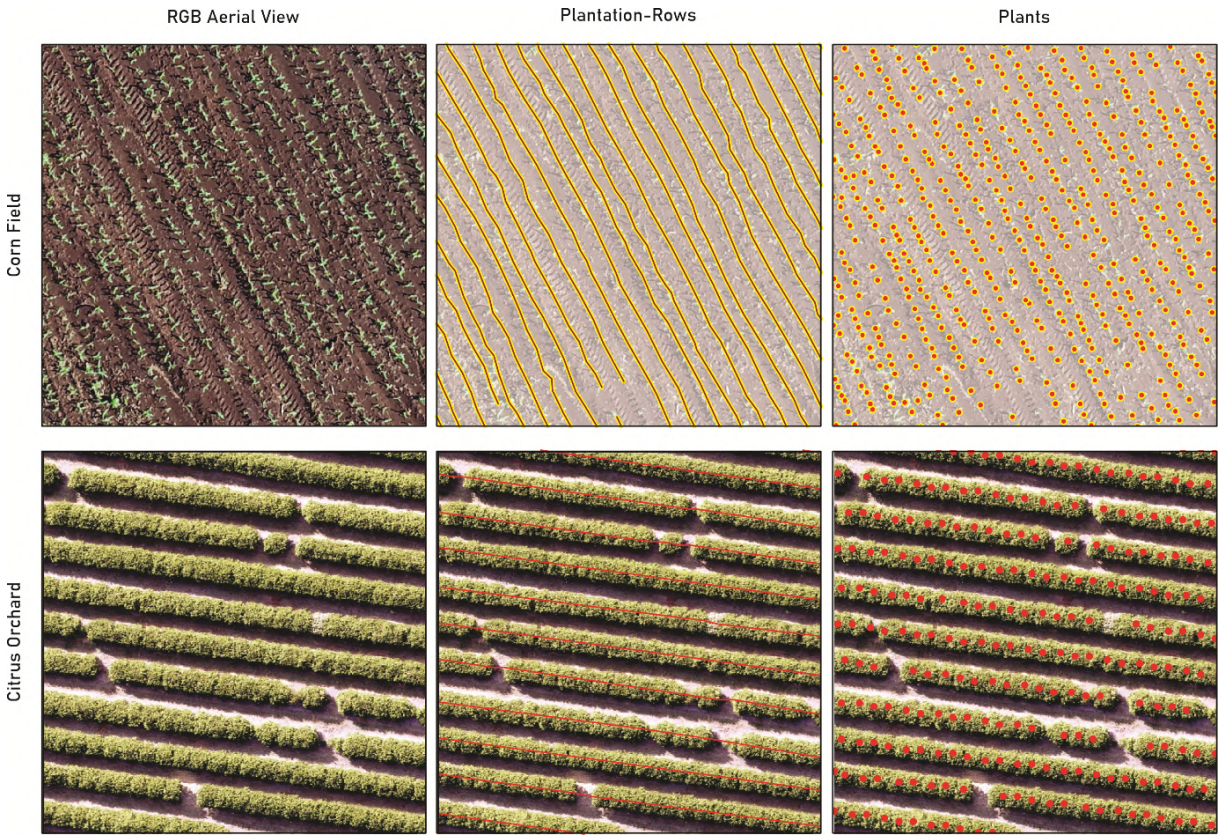


Figure 6.1: High-scale examples of the RGB images used displaying the plantation-rows, corn-plants, and citrus-trees that were manually identified.

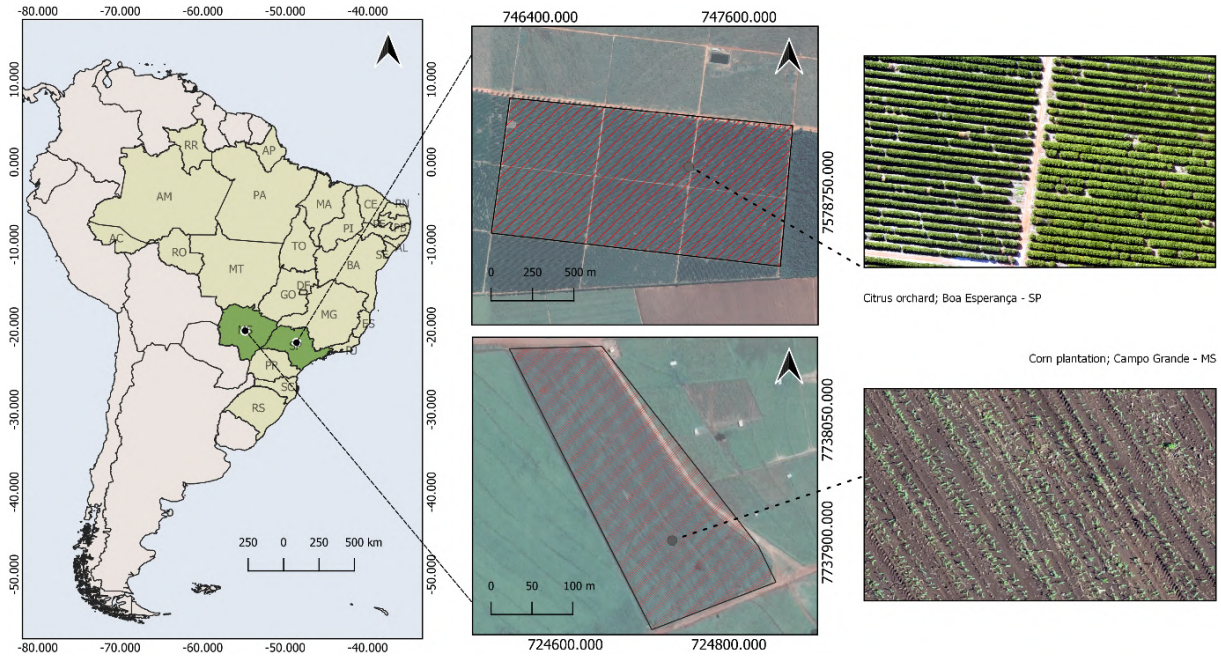


Figure 6.2: Overall visualization of the study area. The cornfield (row-below) is located at Campo Grande, MS, and the citrus orchard (row-above) is in Boa Esperança, SP.

6.2.2 Convolutional Neural Network

The UAV images with $w \times h$ pixels were processed using the proposed CNN model to compute the positions of individual plants and plantation-rows. The object counting and geolocation were modeled after a 2D confidence map estimation using the method presented in both [Aich and Stavness, 2018] and our previous work (described in Chapter 4). The confidence map in this case is a 2D representation of the likelihood of an object occurring in each pixel. The PPM [Zhao et al., 2017] that inserts global and local neighborhood information was included in the model to improve the estimation of the confidence map. A MSM prediction that refines the confidence map was used to a more accurate prediction of the center of the objects, similar to our previous work to detect tree species in a national park with hyperspectral imagery [Miyoshi et al., 2020].

Our latest addition to the aforementioned architecture was two detection-branches inside the MSM. This addition was necessary to our model to understand how plantation-rows are displayed in the image and how they are related to the plant's position, and vice-versa. This construction permitted our deep network to return both lines and point features simultaneously with their respective geolocation information. It should be mentioned, however, that since our method stores the coordinates of the Geographic Tagged Image File Format (GeoTIFF) image format in a separated file, it does not necessarily require this geographic information to perform the detection when analyzing the input image. Regardless, this information is then added later at the final prediction map, where it incorporates the coordinates X, Y from the separated file and creates a geolocated predicted map. The concatenation process, as well as the information exchanged between the two-branches during the multiple stages, allows for the refinement of both confidence maps generated ("object-plant" detection and "line-plantation" detection). With this refinement, the line or point feature is extracted from the center's position of the highest peaks on the map.

Figure 6.3 presents our method for detecting plants and plantation-rows. The method starts by extracting the feature map as shown in Figure 6.3 (b) from an RGB input image as viewed in Figure 6.3 (a). The feature map obtains global and local neighborhood information when passing through the PPM (see Figure 6.3 (c)). The volume is then processed by an MSM, see Figure 6.3 (d), with T stages, and is refined to detect plants and plantation-rows in two branches. For this, the volume obtained from the PPM module, as shown in Figure 6.3 (c), is used as an input for the T stages of MSM. Also, both branches share their volumes between each stage of the MSM, obtaining a more precise identification of the plants and plantation-rows. Finally, we

obtain the detection of plants as shown in Figure 6.3 (e) and rows as shown in Figure 6.3 (f) at the end of the processing of each branch.

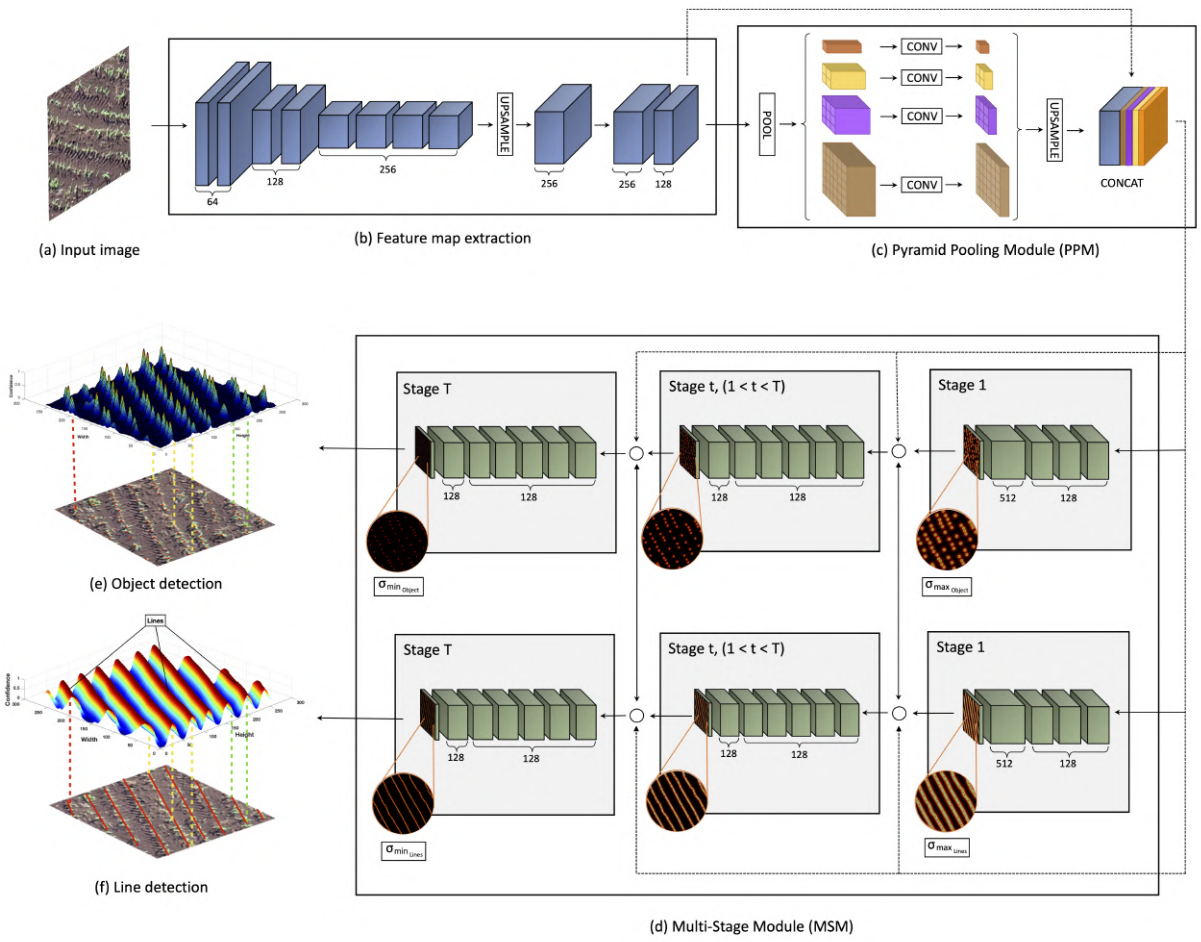


Figure 6.3: Our method proposed for detecting objects and plant-rows: (a) input UAV image, (b) the feature map obtained by CNN, (3) the PPM enhancement module with the feature map as an input, (d) the two detection branches of the MSM, (e) object detection (plants) and (f) line detection (plantation-rows).

The following subsections detail the four main phases of the proposed CNN: (1) the generation of the feature map with CNN; (2) the feature map enhancement with the PPM module; (3) The refinement of the confidence map by the MSM module; and (4) how we obtain the positions of objects and rows through peaks in the confidence map.

Feature Map Extraction

The feature map was extracted from an RGB input image with 256×256 pixels using a CNN based on the VGG19 [Simonyan and Zisserman, 2014] as a feature extractor. The CNN has eight convolutional layers, two maximum pooling layers, and one upsampling layer. The convolutional layers have 64, 128, and 256 filters, all with a size 3×3. The two max-pooling layers are inserted after the second and fourth convolutional layers and use a window of 2×2.

Also, after each convolutional layer, we have ReLU function. Finally, the last layer consists of an upsampling layer that delivers an extracted map with 128×128 resolution that can describe relevant features from the image.

Pyramid Pooling Module - PPM

The global and local image properties allow for the identification of the plant's position to be more accurate in high-density situations. On the other hand, challenges in identifying plants at different scales and stages of growth are very common in various applications. Thus, our method adopts the global and local context module called PPM [Zhao et al., 2017], which allows it to be scale-invariant and helps the network to deal with multiple sizes of the canopy. The PPM module receives as input the feature map generated in the previous step (Figure 6.3 (b)) and applies four parallel pooling layers (Figure 6.3 (c)), following the described in Section 4.2.2. Finally, the PPM delivers a improved feature map of the image with global and sub-regional information.

Multi-Stage Module Refinement and Co-Shared Information

The MSM refinement phase estimates a confidence map from the improved feature map obtained by the PPM. This phase includes two branches of detection with T refinement stages; the first is for plant detection and the second is for plantation-row detection. The first stage contains five convolutional layers and receives as input the improved feature map of the PPM module. The first three layers have 128 filters with 3×3 sizes, the fourth layer has 512 filters with 1×1 size, and the last layer is composed of a single filter that corresponds to the confidence map generated by the first stage of each branch, C_1^{plant} and C_1^{row}, respectively.

The $T-1$ final stages refine the positions predicted in the first stage, forming one type of hierarchical learning of the object positions. Because of that, in the stage t, where $t = [2, 3, ..., T]$, the prediction returned by the previous stage of each branch (C_{t-1}^{plant}, C_{t-1}^{row}) and the feature map from the PPM process are concatenated. Later, they are used to produce a refined confidence map for each branch of the stage t (C_t^{plant} and C_1^{row}). These stages have seven convolutional layers, in which: five layers with 128 filters with a 7×7 size; and one layer with 128 filters with a 1×1 size. The last layer has a sigmoid activation function so that each pixel represents the probability of the occurrence of an object (values between $[0, 1]$). The remaining layers have a ReLU activation function.

Sharing volumes between the branches at the end of a stage t allows the learning of the plantation-rows, from the previous stage C_{t-1}^{row}, to influence the plant prediction in the current stage of C_t^{plant}, refining object predictions for

regions where plantation-rows have been identified. Similarly, learning the positions of plants from a previous stage C_{t-1}^{plant} helps the row detection branch to predict more accurate plantation-rows in the current stage, C_t^{row}, because they consider the object's predictions for defining these rows. Also, the use of the improved feature map obtained in the PPM phase at the entrance of each stage allows for multi-scale features, obtained from both the global and local context information, to be incorporated into the refinement process.

Lastly, to avoid the vanishing gradient problem during the training phase, we adopted loss functions to be applied at the end of each stage of the branches. Each branch (i.e. C_{t-1}^{plant}, C_{t-1}^{row}) has its loss function (similar to Equation 2.3); so while f_t^{plant} represents the loss function of the plant detection, f_t^{row} represents the loss function of the plantation-row itself (similar to Equation 2.4). By the end of it, the general loss functions of each branch are also calculated.

Confidence Map

To train our approach, two confidence maps, C_t^{plant} and C_t^{row} are generated as ground truth for each stage t by using annotations of the plants and plantation-rows in the image (see Section 6.2.1). The confidence map is generated by placing a 2D Gaussian kernel at the labeled plants and plantation-rows. Thus, in the object detection (plant), we have high responses in their centers, while in the row detection, we have high responses over the entire extension of the plantation-rows. The Gaussian kernel has a standard deviation (σ_t) that controls the spread of the confidence map peak, as shown in Section 2.1.

Our approach uses different values of σ_t for each stage t to refine both the plant and the row predictions during each stage. The σ_1 of the first stage is set to a maximum value (σ_{max}) while the σ_T of the last stage is set to a minimum value (σ_{min}), following Section 4.2.3. Unlike previous research, this approach has two branches and the values of σ_{max} and σ_{min} for each branch are independent and could be different for plant and row detection. This generates more accurate maps for each task, depending on the characteristics of the crop and plantation pattern itself.

Figure 6.4 illustrates two examples of ground truth confidence maps with three values of σ for the V3 corn dataset. The top line of the image shows the confidence map for plant detection while the bottom line presents the confidence map for row detection. Column (a) shows the RGB images and the locations of each plant and row marked by red dots and lines, respectively. The columns (b, c, and d) present the ground truth confidence maps for $\sigma = 0.5$, 1.0, and 1.5, respectively; which are responsible for controlling the spread in the top values of our confidence maps. During our experiment, the usage of

different σ helped us to refine the confidence map, improving its robustness.

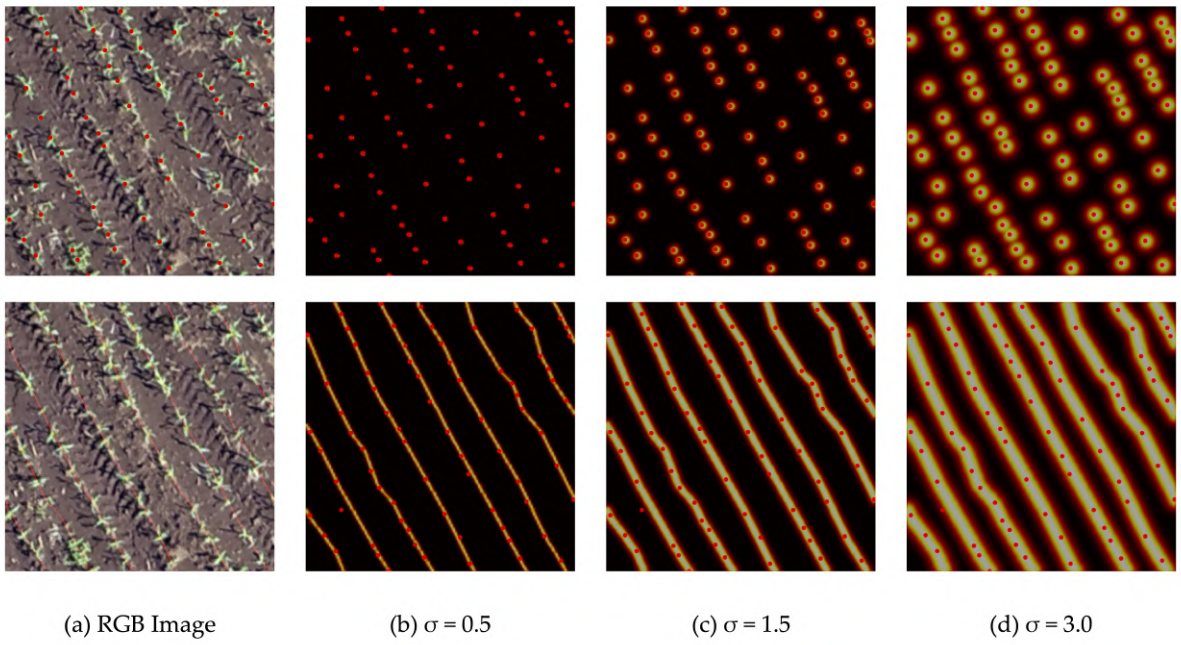


Figure 6.4: Example of an RGB image and its corresponding ground truth confidence maps for object and row detection with different σ values.

Object and Row Localization and Extraction

The location of plants and plantation-rows is obtained from the last stage of each branch (C_t^{plant}, C_t^{row}) of the MSM module. For the location of the objects, we estimate the peaks (local maximum) of the confidence map by analyzing the 4-pixel neighborhood of each given location of p, following Section 2.3.

To avoid noise or low probability of occurrence of the positions of p, a peak in the confidence map is considered as a plant or plantation-row only if $C_T^{plant-plantation-row}(p) > \tau$. We set a minimum distance δ to prevent the detection of plants and rows very close to each other. After conducting a preliminary experiment, we used as a minimum $\delta = 1$ pixel and $\tau = 0.35$.

To detect rows, we use the skeleton topological algorithm [Quan et al., 2019] on the row confidence map, C_t^{row}, to obtain the central activations that represent the plantation-rows. The skeletonization algorithm makes successive passes over the map, removing the activation borders. This process is repeated until there is no border to be removed, and only the skeleton (centerline) of the confidence map remains. Thus, the algorithm delivers a thin version of the shape that is equidistant to its boundaries. Figure 6.5 shows an example of the confidence map and the skeleton generated by this approach.

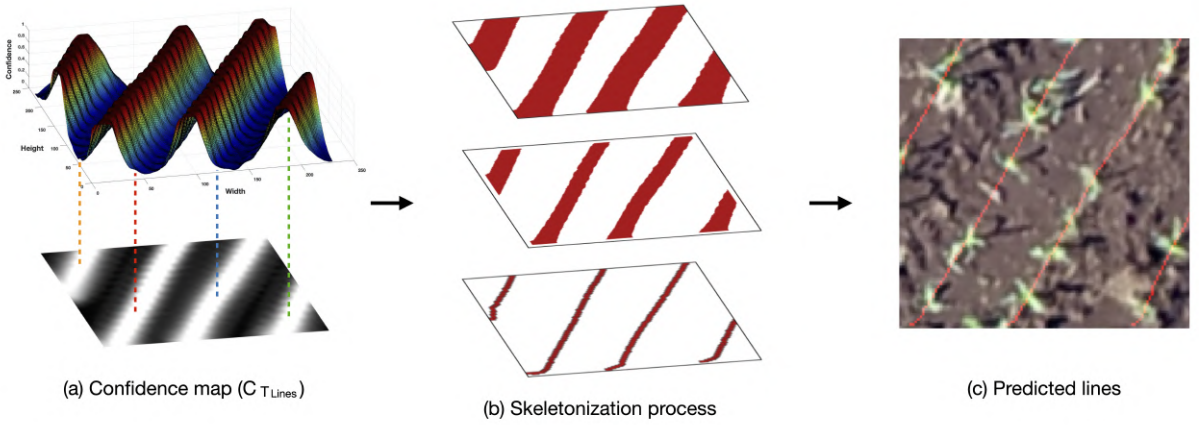


Figure 6.5: Example of skeletonization process on the confidence map. (a) confidence map C_t^{row} and its 3D representation, (b) skeletonization process over the confidence map, and (c) predicted rows over the image.

6.2.3 Experimental Setup

The collected corn images described in Section 6.2.1 were split into 1157 patches with 256×256 pixels without overlapping, were 564 and 593 patches corresponding to V3 and mature corn plantations, respectively. The citrus plantation images were split into 635 patches with 256×256 pixels without overlapping. The point and line features identified as image-samples were also split between both (corn and citrus) patches. For the corn plantations (V3 and mature), $33,360$ plants and 224 plantation-rows were considered in the training phase. As for the citrus-orchard, $14,810$ trees and 213 citrus plantation-rows were used in our experiment. As previously stated, this type of characterization with different plant phonologies, sites, and sensor characteristics was essential to ascertain the generalization and robustness of our approach.

The training, validation, and testing sets of each dataset were formed through a random division of the patches in 60%, 20%, and 20%, respectively. This allows the network to not consider the same patch for any of the subsets. For training, we initialize our network with the weights of the first part pre-trained on ImageNet, and apply a SGD optimizer with a moment of 0.9. The validation set was used to adjust the learning rate and the number of epochs, reducing the risk of overfitting. After the initial adjustments, the learning rate was set to 0.001 and the number of epochs was set to 100. We also performed additional comparisons with our CNN against state-of-the-art deep neural networks, like High-Resolution Network (HRNet), Faster R-CNN, RetinaNet, YOLOv5 and YOLOv7. These networks were implemented considering the same dataset characteristics and sampling conditions.

Both our CNN and the other deep networks methods were implemented in Python language on the Ubuntu 18.04 operating system and used the Keras-

TensorFlow API. The computer used for training and testing has an Intel (R) Xeon (E) $E3-1270@3.80$ GHz CPU, 64 GB memory, and an NVIDIA Titan V graphics card, that includes a 5120 CUDA cores and 12 GB of graphics memory. Finally, to evaluate the performance of our approach, we adopted five regression metrics: MAE, Mean Relative Error (MRE), the MSE, P, R, and F1. The regression metrics, MAE, MRE, and MSE, were also adopted in this problem since they estimate the comparison between a given number of labeled corn plants and citrus trees against the predicted positions returned by the network for each image patch. This strategy was also adopted in related work [de Arruda et al., 2022] and helped to better explain the estimative and performance of the method.

6.3 Results

This section is organized as follows: First, we present the results from an analysis of the parameters to refine the prediction, following the analysis of the σ values for the generation of the ground truth confidence maps of each detection branch. Later, we compare the results from our corn plantation dataset with a baseline method of our CNN as well as other deep learning architectures. Finally, we explore the generalization of our method by evaluating its performance on an entirely different agricultural crop (citrus).

6.3.1 Corn Plantation Dataset

Parameter Analysis in Plant and Plantation-Row Detection

Here, we present the results of the proposed method in the validation set for a different number of stages on the corn-crop dataset. These stages are responsible for refining the confidence map. We observed that, by using two stages ($T = 2$), the proposed method already returned the following results (Table 6.1). When increasing to $T = 6$ stages, we obtained the following results in detecting plants: MAE $= 7.778$, MRE $= 0.1360$, MSE $= 100.132$, P $= 0.865$, R $= 0.937$, and F1 $= 0.894$. These results indicate that the MSM phase affects the plant detection/counting tasks significantly. This is because the confidence map is refined in later stages, increasing the chance of plants to be detected in hard-to-detect regions of the image, associated with a highly-dense plantation. Similarly, we can observe that the performance in predicting plantation-rows is improved with $T = 6$ stages, reaching P, R, and F1 of 0.934, 0.983, and 0.956, respectively. Although the result for eight stages ($T = 8$) is slightly better than $T = 6$ (Table 6.1), the computational cost significantly increases, not justifying its adoption in later experiments.

Table 6.1: Evaluation of the number of stages T on the validation set using $\sigma_{min} = 1$ and $\sigma_{max} = 3$ for both branches.

Stages (T)	**Plant**						**Row**		
	MAE	**MRE**	**MSE**	**P**	**R**	**F1**	**P**	**R**	**F 1**
2	7.991	0.1398	102.769	0.862	0.928	0.888	0.926	0.985	0.952
4	7.672	0.1342	98.89	0.866	0.931	0.892	0.924	0.985	0.951
6	**7.778**	**0.1360**	**100.132**	**0.865**	**0.937**	**0.894**	**0.934**	**0.983**	**0.956**
8	7.867	0.1376	100.575	0.866	0.936	0.895	0.936	0.984	0.957

We also evaluated the σ_{min} and σ_{max} responsible for generating the ground truth confidence maps implemented in the T stages of the MSM phase. In the first stage, the confidence map is generated with σ_{max}, while the last stage uses σ_{min} and the intermediate stages adopt values equally spaced between [σ_{max}, σ_{min}]. The σ value concerning the plant influences the predicted location of their locations by the model. A low σ provides a confidence map that does not correctly cover the plant's whole area, while a σ too high can include nearby plants in high-density conditions. In both cases, these conditions make it difficult to spatially locate the plants in the image.

To evaluate σ_{min} and σ_{max} we adopted the stages $T = 6$ that obtained the best results in the previous phase of the experiment, and we use the same value of σ_{min} and σ_{max} in the two detection branches (plant and plantation-row). Table 6.2 shows the evaluation of σ_{max} in the validation set. The highest result was obtained with $\sigma_{max} = 3$, indicating that the confidence map peak, by adopting this value, covers correctly each plant without overlapping nearby plants. On the other hand, as the confidence map is refined by the T stages, and σ_{max} is applied in the first stage, it has a small influence on the final result, evidenced by the results of the other σ_{max}.

Table 6.2: Evaluation of the σ_{max} in the validation set. We adopted stages $T = 6$ and $\sigma_{min} = 1$ for both branches.

σ_{max}	**Plant**						**Row**		
	MAE	**MRE**	**MSE**	**P**	**R**	**F1**	**P**	**R**	**F1**
2	7.867	0.1376	100.592	0.863	0.934	0.892	**0.939**	**0.984**	**0.959**
3	**7.778**	**0.1360**	**100.132**	**0.865**	**0.937**	**0.894**	0.934	0.983	0.956
4	7.734	0.1353	96.867	0.865	0.931	0.891	0.935	0.984	0.957

Table 6.3 presents the results of the σ_{min} evaluation with the validation set. In this experiment, we set the $\sigma_{max} = 3$ for the two branches and the stages $T = 6$ because they obtained the best performances in the previous experiments. In this phase, we observe that σ_{min} has a great influence on the method performance since it is used to generate the ground truth confidence maps at the last stage of the MSM phase. The best result was obtained with $\sigma_{min} = 1.0$, indicating a better fit for the plant canopy size. The experiments showed that the best results for counting plants were achieved with $\sigma_{max} = 3$

and $\sigma_{min} = 1$, delivering F1-Measure of 0.894 and MAE of 7.778 plants per patch, respectively.

Table 6.3: Evaluation of the σ_{min} in the validation set. We used stages $T = 6$ and $\sigma_{max} = 3$ for both branches.

σ_{min}	Plant						Row		
	MAE	**MRE**	**MSE**	**P**	**R**	**F1**	**P**	**R**	**F1**
0.5	21.362	0.3737	552.070	0.949	0.597	0.723	**0.944**	**0.984**	**0.962**
1	**7.778**	**0.1360**	**100.132**	**0.865**	**0.937**	**0.894**	0.934	0.983	0.956
1.5	8.230	0.1440	111.115	0.857	0.938	0.890	0.921	0.981	0.948
2	7.840	0.1371	98.902	0.860	0.929	0.888	0.900	0.963	0.928

Regardless of the aforementioned observations, we noticed that the plantation-row detection performance was better with $\sigma_{max} = 2$ and $\sigma_{min} = 0.5$, reaching P, R and F1 of 0.939, 0.984 and 0.959, respectively, for $\sigma_{max} = 2$ (see Table 6.2) and 0.944, 0.984 and 0.962, respectively, for $\sigma_{min} = 0.5$ (see Table 6.3). Therefore, we evaluated the variation of σ between the two branches, which so far were considered the same. To solve this, we defined a σ pair for each branch, σ_{min}^{plant}, σ_{max}^{plant} and σ_{min}^{row}, σ_{max}^{row} for plant and plantation-row detection branches, respectively. For the plant detection branch, we set $\sigma_{min}^{plant} = 1.0$ and $\sigma_{max}^{plant} = 3.0$ that had the best results in the previous experiments, and we varied the σ of the plantation-row detection branch according to Table 6.4. The experiments showed that the best results were obtained with $\sigma_{min}^{row} = 0.5$ and $\sigma_{max}^{row} = 3.0$, reaching P, R, and $F1$ of 0.950, 0.983, and 0.965, respectively.

Table 6.4: Evaluation of σ for planting row detection. We adopted the $\sigma_{min}^{plant} = 1$, $\sigma_{max}^{plant} = 3$ and stages $T = 6$.

σ_{min}^{row}	σ_{max}^{row}	Plant						Row		
		MAE	**MRE**	**MSE**	**P**	**R**	**F1**	**P**	**R**	**F1**
1	3	7.778	0.1360	100.132	0.865	0.937	0.894	0.934	0.983	0.956
0.5	**3**	**7.672**	**0.1342**	**97.283**	**0.866**	**0.935**	**0.894**	**0.950**	**0.983**	**0.965**
0.5	2	7.831	0.1370	101.725	0.864	0.934	0.893	0.945	0.983	0.962

Plant and Plantation-Rows Extraction

To analyze the design of the proposed architecture, we compared it with a baseline model that does not include the plantation-row detection branch. The overall best result with the baseline method was obtained with $\sigma = 1.0$, returning an MAE, MRE, MSE, P, R, and F1 equal to 6.345, 0.1051, 71.637, 0.870, 0.940, and 0.899, respectively (Table 6.5). Although the result of $\sigma = 0.5$ and 2.0 has higher precision value, we observed that in the first case ($\sigma = 0.5$), has a very low recall value, indicating that the number of false-negative predictions is high. Still, for $\sigma = 2.0$, we observe that the recall value decreases while the F1-Measure maintains the same, indicating a stabilization in the results.

We analyzed the plant and plantation-row detection branches independently to evaluate their performance over the complete method. In both cases, the results of the proposed approach were better, mainly in the precision metric. This indicates that the proposed approach benefits from information exchange between the branches in the MSM phase. Regardless, these results also indicate that both branches can be applied independently, without much loss of performance, as in situations where the plantation-rows are not well defined or in other problems involving a row or centerline detection (e.g., roads).

Table 6.5: Results of the proposed method and its baselines.

Methods	**Plant**						**Row**		
	MAE	**MRE**	**MSE**	**P**	**R**	**F1**	**P**	**R**	**F1**
Baseline ($\sigma = 0.5$)	23.849	0.3950	738.876	0.946	0.536	0.685			
Baseline ($\sigma = 1.0$)	6.345	0.1051	71.637	0.870	0.940	0.899			
Baseline ($\sigma = 2.0$)	5.991	0.0992	68.132	0.872	0.938	0.899			
Proposed Approach	5.486	0.0908	55.982	0.878	0.934	0.901	0.950	0.979	0.963
$\sigma_{min}^{plant} = 1$ $\sigma_{max}^{plant} = 3$	5.778	0.0957	61.619	0.872	0.939	0.901			
$\sigma_{min}^{row} = 0.5$ $\sigma_{max}^{row} = 3$							0.945	0.980	0.961

Regarding the different growth periods of the corn plants (V3 and mature), our neural network produced similar performance metrics (Table 6.6). The V3 detection was slightly better than mature plant's detection when evaluating the classification metrics (P, R, and F1). The V3 detection was also better in plantation-row detection. This could be related to the smaller size of V3 plants, and therefore the occurrence of fewer occlusions in the neural network's prediction. As for the better plantation-row identification, since the information is exchanged between both CNNs branches (Figure 6.3 (d)), where the concatenation from the multi-stage feature extraction is refined with information from both plant and row locations. So, an improved plant detection will often result in an improved plantation row detection. It should be noted that the MAE, MRE, and MSE metrics, which indicate the amount of error produced in each image patch, were slightly worse for the V3 plants than mature plants. It should be highlighted that is hard-to-detect plants located mostly at the edges of the patches (Figures 6.9 and 6.10), implying the increase of MAE, MSR, and MSE values.

Table 6.6: Performance of the proposed CNN according to the different growth periods of the corn plantation

Trained over:	**Tested over:**	**Plant**						**Row**		
		MAE	**MRE**	**MSE**	**P**	**R**	**F1**	**P**	**R**	**F1**
V3 + Mature	V3 + Mature	6.224	0.1038	66.706	0.856	0.905	0.876	0.914	0.941	0.926
V3 + Mature	V3	7.504	0.1243	92.530	0.870	0.924	0.891	0.947	0.980	0.961
V3 + Mature	Mature	5.008	0.0840	42.184	0.843	0.887	0.862	0.884	0.904	0.893
V3	V3	5.486	0.0908	55.982	0.878	0.934	0.901	0.950	0.979	0.963
Mature	Mature	4.378	0.0734	36.848	0.872	0.872	0.870	0.887	0.918	0.901

Figure 6.6 presents some examples of the results obtained with the proposed method for the V3 corn-crop dataset when adopting the best configu-

ration predefined during the phase of the experiment, $\sigma_{min}^{plant} = 1.0$, $\sigma_{max}^{plant} = 3.0$, $\sigma_{min}^{row} = 0.5$, $\sigma_{max}^{row} = 3.0$, and $T = 6$. We considered a region around the labeled plant position to analyze qualitatively the prediction within the plant center. The correctly predicted positions are represented by blue dots in the image, and the regions are represented by see-through yellow-circles whose center is the labeled position of the plant. The blue and red dots represent the true and false plant predictions, respectively, while yellow and red circles identify whether the annotated plants were detected or not by the method. The proposed CNN can correctly predict most of the plant's positions. Even in overlapping plants, the proposed approach can correctly identify the plant's position. We also observed that the method achieves high performance in plantations at different density conditions.

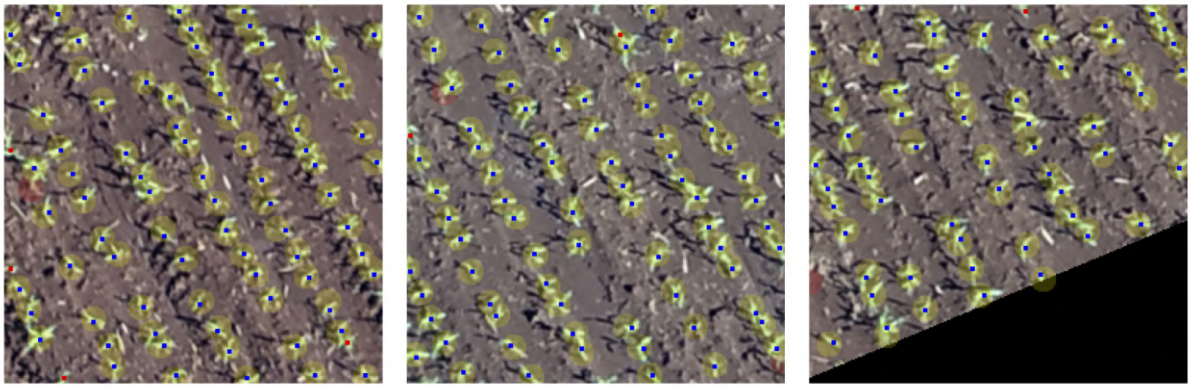
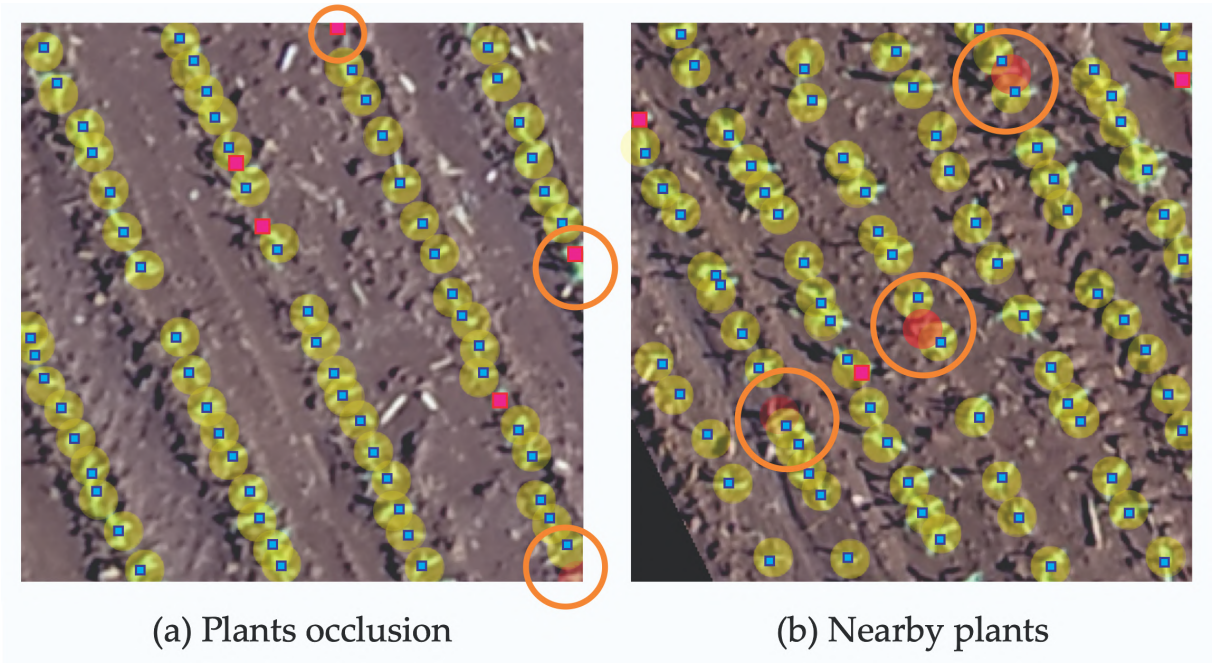


Figure 6.6: Visual results of the proposed method for object detection. Predicted positions are shown by dots while tree-canopies are represented by circles.

Although the proposed method is appropriate for most of the corn plant detection and counting tasks, it also faces some challenges (Figure 6.7). The two main challenges are the plant detection at the borders of the image-patch (see Figure 6.7 (a)) when most of the plant is occluded, and when we have high-density regions with plants overlapping each other in the plantation-rows (see Figure 6.7 (b)). Nonetheless, even in these cases, we observed that our method can correctly predict the position of the majority of the plants. Moreover, the predicted positions of the plants have a high level of accuracy, with most of the predictions (blue-dots) close to the center of the annotations (center of the yellowish circles).

Figure 6.8 shows the performance of the proposed approach in detecting plant-rows. The predicted rows (yellow) fits with the annotated rows extension (red). The three main challenges in the detection of planting rows are the prediction of rows that correctly adapt the curves of the plantation, the identification of rows with large spacing between plants, and the identification of isolated plants outside the correct plant-row.



(a) Plants occlusion (b) Nearby plants

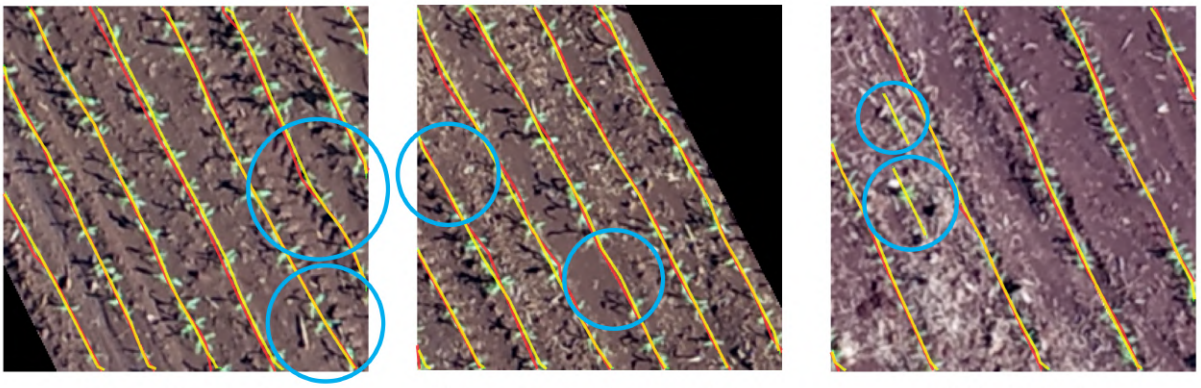
Figure 6.7: Examples of the challenges faced by the proposed method for plant detection. The orange circles show the challenges faced by the method. The blue dots and the yellow circles represent a correct prediction and the tree-canopies of the annotated plants. The red-pink dots and red circles represent the false-positives detections and the missing annotated plants, respectively.

As shown in Figure 6.8 (a), the approach can correctly identify the curves of the plant-rows, highlighted by the blue circles. This happens because when exchanging information between the detection branches, the positions of the detected plants influence the shape of predicted rows. Also, Figure 6.8 (b) shows that the approach predicted plantation-rows with large spacing between the plants (blue circles). Finally, the identification of outside plants in the plant-rows is a challenge, because they may define plant-rows incorrectly. In this case, the proposed approach can identify them (see Figure 6.8 (c)) without attributing them to a true plant-row definition.

Comparative Results with State-of-the-Art Deep Networks

Our CNN method obtained better performance when compared to some state-of-the-art object detection methods, like HRNet, Faster R-CNN, RetinaNet, YOLOv5 [Jocher et al., 2022] and YOLOv7 [Wang et al., 2022]. In YOLO comparison we adopted two subvariations and the use of transfer learning. For YOLOv5 we use the Large (YOLOv5l) and the xLarge (YOLOv5x) model, and for YOLOv7 we use the models for normal GPU (YOLOv7) and Cloud GPU (YOLOv7-W6) proposed by the authors. In additional, we evaluate the method training from scrat and using transfer learning, represented by "*".

Table 6.7 shows the results of this comparison for the MAE, MRE, MSE, P, R, and F1 metrics. We observe that the best variations of YOLOv5 and



(a) Curve adaptation (b) Large spacing between plants (c) Single plants

Figure 6.8: Examples of planting row detection by proposed method and its challenges. The blue circles highlight the challenges described. Yellow lines correspond to the lines identified by the network, while red lines under it correspond to the labeled example.

YOLOv7 methods were YOLOv5x and YOLOv7*, reaching a 6.590 and 7.017 for MAE and 0.855 and 0.850 for F1. Our approach obtained better values of MAE, MRE, MSE, P, and F1 when compared with the evaluated methods, reaching important differences in the P ($+9.5\%$ of HRNet, $+12\%$ of Faster R-CNN, $+16.3\%$ of RetinaNet, $+1.3\%$ of YOLOv5x and $+2,6\%$ of YOLOv7*) and MAE (-8.665 from HRNet, -11.021 from Faster R-CNN, -14.026 from RetinaNet, -0.366 from YOLOv5x and -0.773 from YOLOv7*) metrics. This indicates that the proposed method delivers more accurate detections than the other evaluated deep networks while generating fewer false detections.

Table 6.7: Results of the proposed method and the object detection methods HRNet, Faster R-CNN, RetinaNet, YOLOv5 and YOLOv7 for the corn plantation (V3 and matures) datasets. The "*" indicates the use of transfer learning in YOLO methods.

Methods	**Plant**						**Row**		
	MAE	**MRE**	**MSE**	**P**	**R**	**F1**	**P**	**R**	**F1**
HRNet	14.879	0.2481	319.258	0.761	0.955	0.840			
Faster R-CNN	17.245	0.2876	392.754	0.736	0.952	0.825			
RetinaNet	20.250	0.3377	558.025	0.693	0.940	0.786			
YOLOv5l	7.896	0.1407	92.258	0.832	0.868	0.849			
YOLOv5l*	8.633	0.1516	99.021	0.822	0.856	0.838			
YOLOv5x	6.590	0.1176	65.702	0.843	0.867	0.855			
YOLOv5x*	8.323	0.1463	97.107	0.823	0.853	0.837			
YOLOv7	7.948	0.1418	100.224	0.832	0.867	0.849			
YOLOv7*	7.017	0.1247	72.189	0.826	0.876	0.850			
YOLOv7-W6	8.952	0.1693	111.314	0.627	0.834	0.716			
YOLOv7-W6*	6.905	0.1239	76.000	0.821	0.880	0.849			
Proposed Approach	6.224	0.1038	66.706	0.856	0.905	0.876	0.914	0.941	0.926

Despite the proposed method obtaining slightly lower results for R, when we analyze the F1, which considers P and R, we observed that the approach still obtains better performance with a difference of $+3.6\%$ from HRNet, $+5.1\%$ from Faster R-CNN, $+9\%$ from RetinaNet, $+2.1\%$ from YOLOv5x and $+2.6\%$ from YOLOv7*. Since the HRNet, Faster R-CNN, RetinaNet and YOLO baseline

methods implemented in this study require Bounding Box instead of line or point features, we were not able to evaluate its metrics into the plantation-rows properly. Nonetheless, our method achieves high-performance metrics in the detection of plantation-rows, reaching 0.914, 0.941, 0.926 for P, R, and F1, respectively. This is an important observation since our method can perform this type of analysis (simultaneously counting and detecting plants and plantation-rows) within a one-step architecture.

To verify the potential of the proposed approach in real-time processing, we compared its performance with the other investigated methods in terms of computational cost. Table 6.8 shows the average processing time and standard deviation for the detection in the test set. The values of $\sigma_{min}^{plant} = 1$, $\sigma_{max}^{plant} = 3$, $\sigma_{min}^{row} = 0.5$ and $\sigma_{max}^{row} = 3$ and $T = 6$, which obtained the best performance in previous experiments, were used in this evaluation. The results show that the approach can achieve real-time processing, offering image detection in 0.582 seconds with a standard deviation of 0.001. Similarly, the HRNet, Faster R-CNN, RetinaNet, and the best variations of YOLOv5 and YOLOv7 methods (showed in Table 6.7) obtained an average detection time and standard deviation of 0.070, 0.053, 0.050, 0.013, 0.012, and 0.009, 0.010, 0.010, 0.0002, 0.005, respectively. The processing time was calculated considering the workstation described previously (see section 6.2.3).

Table 6.8: Processing time evaluation of the compared approaches.

Method	**Average Time (*sec*)**	**Standard deviation**
HRNet	0.070949	0.009826
Faster R-CNN	0.053500	0.010663
RetinaNet	0.050697	0.010225
YOLOv5x	0.013122	0.000297
YOLOv7*	0.012188	0.005262
Proposed Approach	0.582698	0.001175

Figure 6.9 shows the visual comparison of object detection methods applied here to the corn plants dataset in the advanced growth stage (mature stage with cobs). The proposed approach also returns more accurate detection than the compared methods. The orange circles show the areas where our method stands out in comparison to the other CNNs. We can observe that the evaluated methods generate false detections in denser planting regions. Regardless, in some cases, they detect corn plants outside the planting lines or in empty regions. On the other hand, even in challenging situations highlighted by the blue circles, we notice that our approach delivers more accurate detection than the evaluated methods, with a lower number of false detections (represented by the red dots).

Figure 6.10 shows the performance of the four methods in detecting corn

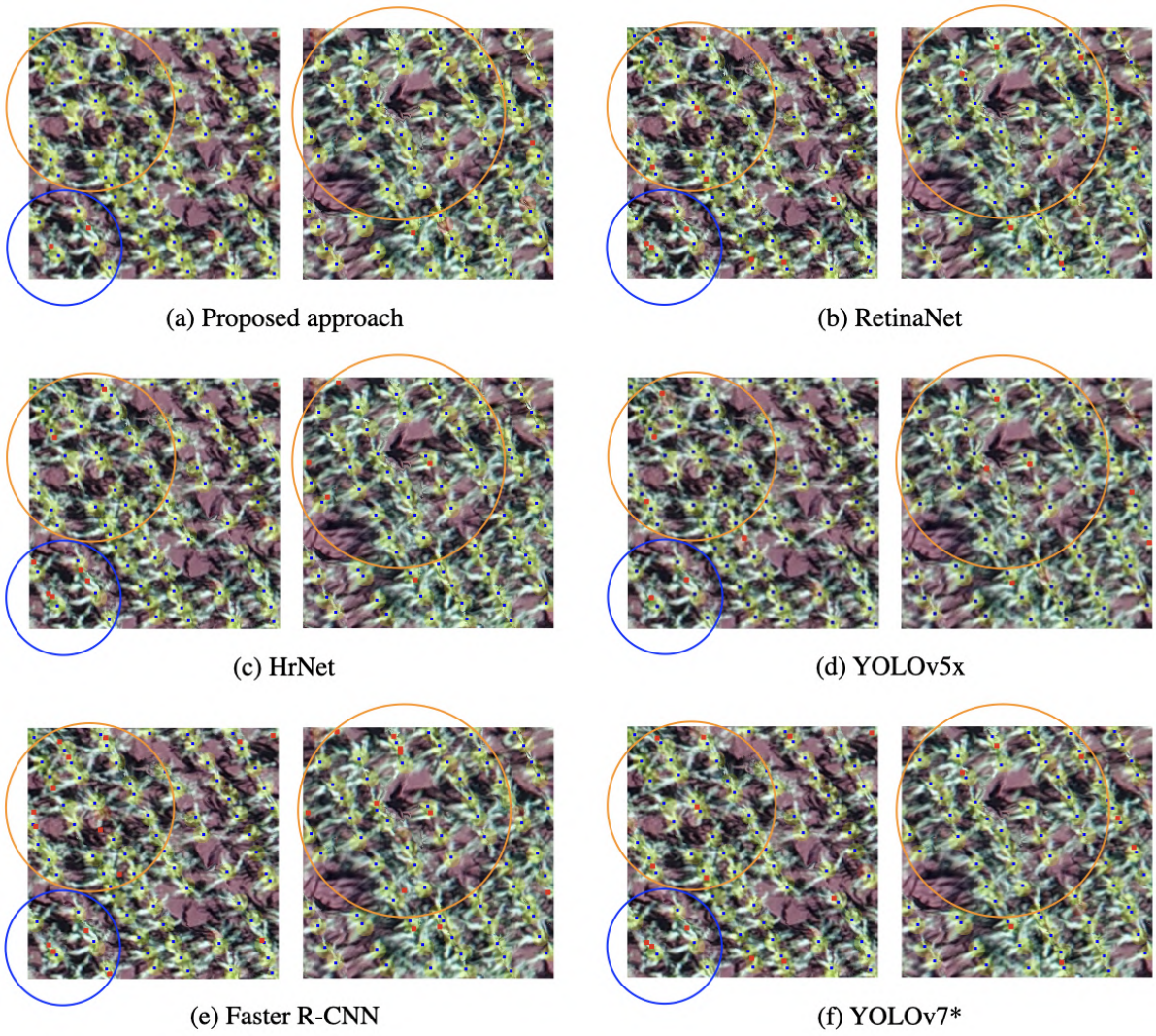


Figure 6.9: Comparison of the object detection methods HRNet, Faster R-CNN, RetinaNet, YOLOv5x and YOLOv7* in the corn plants dataset with a higher growth stage (mature with cobs). The orange and blue circles highlight usual and challenging detections, respectively.

plants in their initial growth stage (V3). In the usual detection situations (highlighted by orange circles), where there is no overlap and occlusion, our method outperforms the compared methods. Also, the compared methods (Figure 6.10 (b), (c), (d), (e) and (f)) fail more at the limits of the planting lines and generate nearby false detections. In challenging detections (highlighted by blue circles), our approach delivers more accurate detections than the evaluated methods.

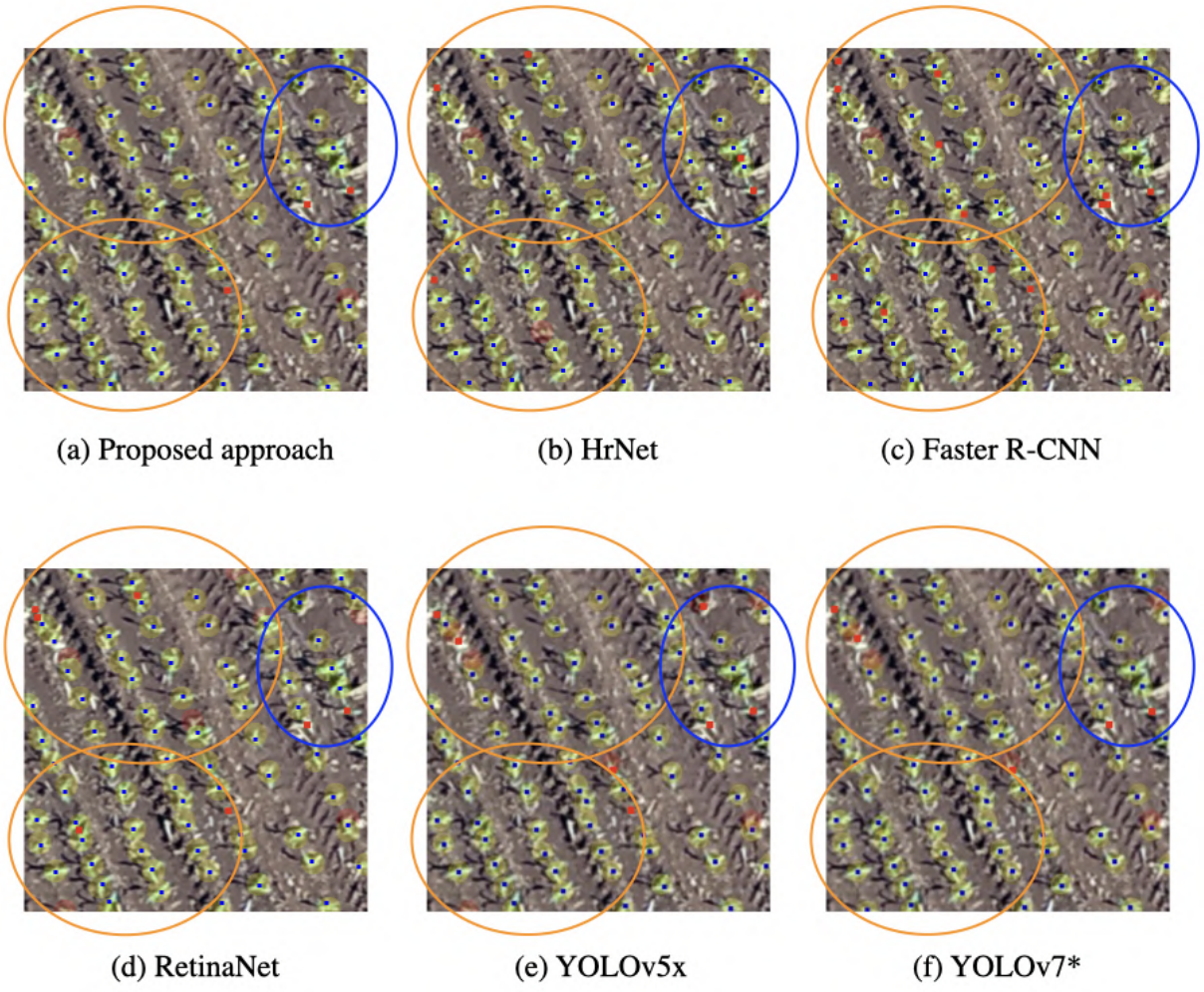


Figure 6.10: Comparison of the object detection methods: HRNet, Faster R-CNN, RetinaNet YOLOv5x and YOLOv7* in the corn plants dataset with an earlier growth stage (V3). The orange and blue circles highlight usual and challenging detections, respectively.

6.3.2 Experiments in the Citrus Plantation Dataset

The parameters used for the citrus plants are the same as the ones adopted for the corn plants ($\sigma_{min}^{plant} = 1$, $\sigma_{max}^{plant} = 3$, $\sigma_{min}^{row} = 0.5$ and $\sigma_{max}^{row} = 3$ and $T = 6$), except for the maximum pixel distance between the prediction and the ground truth annotation (15 pixels for the citrus radius, while in corn we used 8 pixels), which is justified due to the difference in size between the corn plants and citrus trees canopies areas. The achieved results show that our

approach can be generalized into different types of plantations with minimal adjustments to the model (Table 6.9). Also, even in high-density plantations such as citrus, our method maintains high performance, delivering highly accurate predictions. This shows that the MSM module helps not only by learning information related to plant growth in the multiple stages but also by learning other plantation types with more challenging canopies (high-density).

Table 6.9: Results of the proposed method for the citrus orchard dataset.

Method	**Plant**						**Row**		
	MAE	**MRE**	**MSE**	**P**	**R**	**F1**	**P**	**R**	**F1**
Proposed Approach	1.409	0.0615	3.724	0.922	0.905	0.911	0.965	0.970	0.964

Figure 6.11 shows the performance of the proposed method on the citrus dataset. Different from the corn dataset, these plants have a much more complex delimitation, with a large canopy and a lot of overlap between more than one plant. However, following the quantitative results, we found that the proposed approach detections are accurate, delivering centralized detections to the plantation-rows. In the plant detection task (Figure 6.11, top row) the blue-dots and yellow-circles represent the correct detections and the tree-canopies of the labeled plants, annotated by a specialist. In the plantation-rows detection (Figure 6.11, bottom row) the red-lines represent the annotations by the specialist and the green lines represent the detections by the proposed method. Here, our method overcomes different challenges such as the highly vegetated cover area, that generates overlapping trees canopies (Figure 6.11 (a)), the detection with plants and plantation-rows occlusion (Figure 6.11 (b)), and the detection of single plants in the limits of the plantation-rows (Figure 6.11 (c)).

6.4 Discussion

The contribution of our study is to demonstrate a feasible alternative to correctly predict the actual number of plants while simultaneously detecting plantation-rows in UAV-based RGB imagery. Upon our evaluation with different sets, the method presented here for estimating plants and plantation-rows may be replicated in other crops, not being only restricted to the ones presented here. Another important contribution of this method is the detection of high-dense canopies plantations. The usage of a confidence map boosted by the refinement between the two architecture branches helped our network to better detect both overlapped plants and individual plants with high accuracy. Also, another contribution of our method is the plantation-rows refinement, which can help farms to correct problems that occurred during the seedling process at early stages in corn, or compensate for plantation gaps in its area.

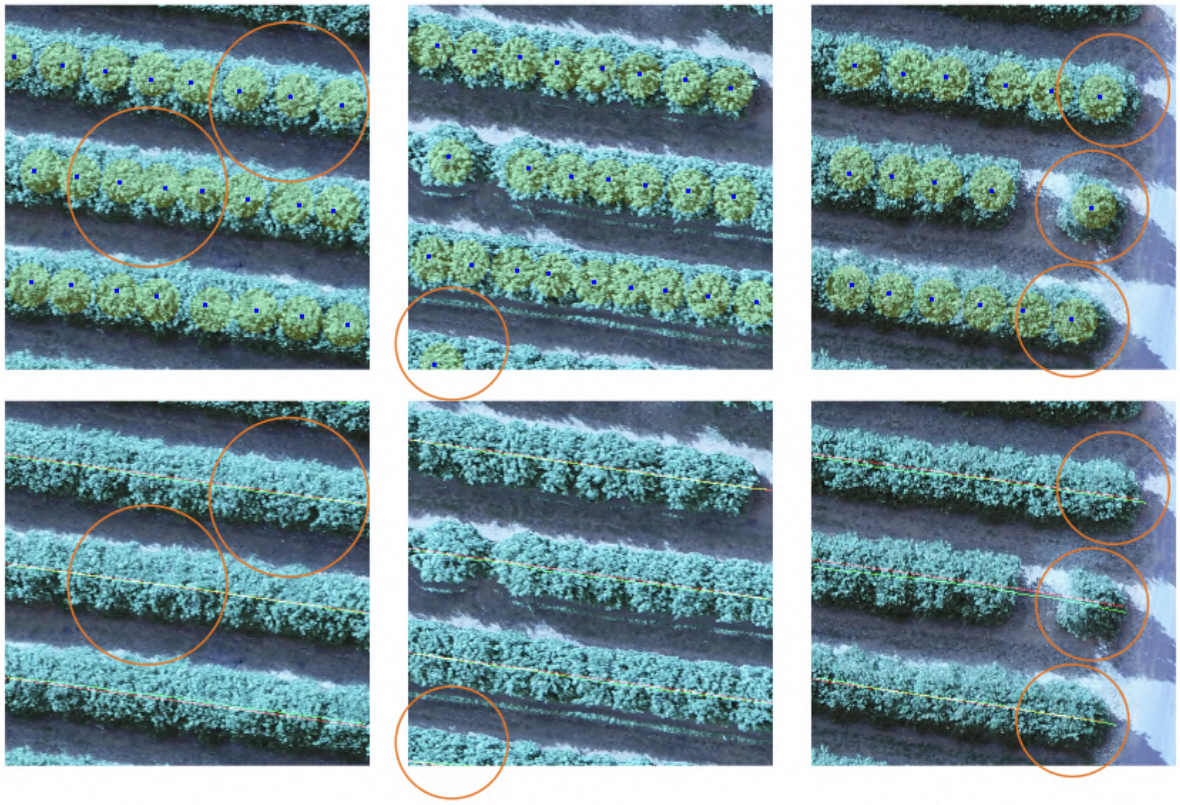


Figure 6.11: Examples of plant and plantation-rows detections in the citrus dataset. Plant and plantation-rows detections are shown in the top and bottom row of the image, respectively. The blue-dots and the yellow-circles represent a correct prediction and the tree-canopies of the labeled plants. The red and green lines represent the annotated and detected plantation-rows. Orange circles highlight the challenges overcome by the approach in each scene.

This form of contribution (plantation-row detection) is also viewed as feasible by similar studies [Primicerio et al., 2017, Fan et al., 2018, Kitano et al., 2019, Salami et al., 2019, Ampatzidis and Partel, 2019].

In our corn-field experimental dataset, up to 4 or 5 plants were estimated per square meter (mts^2) through the photo-interpretation point feature extraction, and our method was capable of detecting these plants with high accuracy levels (F1-Measure of 0.876 and MAE of 6.224). An important issue is that the proposed network only needs points and line features, as opposed to the bounding box type of label needed for the other neural networks. This decreases the time and effort required for the annotation task. Also, our CNN method achieved interesting results with RGB imagery. This tendency to migrate to RGB sensors achieved important outcomes, enough to be a reliable alternative from more expensive equipment such as LiDAR or multi and hyperspectral sensors. RGB images have a relatively low cost to obtain since most conventional UAVs come equipped with them. The tendency to adopt RGB images is not unique to our study [Chen et al., 2017, Wu et al., 2019, Safonova et al., 2019, Salami et al., 2019]. Even so, we intend to implement other types of sensors and methods into our research to take advantage of the proposed method. We believe that this could assist our network in discriminable count plants and detect plantation-rows in other types of environments. Regardless, both datasets evaluated here (corn and citrus) can be considered a high challenge, as they not only are highly-dense types of plantations, with different canopies sizes and pixel-types, but they also represent different plant characteristics.

In cornfields, studies contributed to plant detection in remote sensing imagery. In early-season detection, a decision tree algorithm was able to detect with 0.93 accuracy corn plants with two-to-three leaves [Varela et al., 2018]. The closest study to our approach comes from Kitano et al. [2019]. Their method used a U-Net architecture modification capable of counting corn plants in different growth stages and flying heights. The best overall result had a 2.6% residual percentage, while the worst scenario resulted in a 53.3% residual percentage. Another paper [Gnädinger and Schmidhalter, 2017] performed a digital counting in maize cultivars in aerial images from UAVs, achieving correlations up to 0.89 (R^2) and demonstrating the feasibility of their method. One type of deep neural network (DeepSeedling) was also recently developed based upon a Faster R-CNN model and achieved F1 scores of 0.727 (at IOU_{all}) and 0.969 (at $IOU_{0.5}$). In this regard, another research also implemented the Faster R-CNN model modification to detect maize in terrestrial imagery, achieving similar scores [Quan et al., 2019]. In this manner, we also adopted the Faster R-CNN into our model's comparison; but it returned an inferior

F1-Measure than the proposed CNN for the same dataset. Our approach, however, presents a lower R value, which may be associated with the presence of False-Negatives in our method's prediction. But, since it resulted in higher P values than the others, the harmonic metric F1 was overall better. Lastly, a study that aimed at segmenting individually maize plants with also the Faster R-CNN and LiDAR imagery returned an accurate model also identifying the measured height with impressive results (R^2 higher than 0.9). In this manner, another related work developed a method based on an integrated skeleton extraction and pruning approach, also resulting in interesting outcomes [Zhou et al., 2018].

To show the generalization capability of our model, we also performed additional experiments in a citrus orchard. Citrus-tree crowns, in contrast to the corn plants, are planted in a much denser condition. This highly-density system is commonly implemented in multiple regions around Brazil and other countries, as it helps farmers to maintain a high production set and still not expand their farmlands to new areas. Regardless, this type of system is imposing a new difficulty for the deep learning object detection approach, and our CNN method, based upon a confidence map extraction, appears to be suitable to deal with this condition. As not only citrus orchards possess the mentioned characteristic, other types of crops are also planted at much denser states. Thus, it is important to consider this dataset as an interesting challenge to the proposed approach. Regardless, our method was able to return high accuracies even when considering these characteristics (Table 6.9).

Still, regarding citrus orchards, our previous research (described in Chapter 3), with a simpler conception of a CNN, was able to predict citrus-trees in a different dataset composed only from multispectral imagery; returning an MAE equal to 2.28 trees and an F1-Measure score of 0.950. Other studies conducted in RGB based imagery [Ampatzidis and Partel, 2019, Csillik et al., 2018] were also able to perform well with CNN based architectures (YOLOv3 and a basic CNN), returning classification metrics above 90.0%. Still, since they evaluated datasets with different characteristics than ours, it is difficult to perform such comparisons. Besides, plantation-row identification is also a not commonly found task in the literature. Recently, a novel deep neural network (CRowNet) was proposed for this task alone [Bah et al., 2020], where the authors showed their approach to detect rows in different crops.

Another issue that could be further explored with our network is the addition of datasets with a high variety of weeds and other related problems into our plantation-rows. In our area, weeds were not mainly a concern. However, as we concatenated these branches at the end of each stage, the network can handle the target plant detections with more precision, since it determines

that, for a plant to exist within a certain position, it needs to share information with its plantation-line. This reduces the possibility of the neural network detecting additional objects with similar spectral information (like weeds) aside from the plantation pattern. Besides, the given patterns of weeds are spatially different from the target plants, so this information is also considered by the CNN when learning the labeled data. Still, the performance of a similar network was evaluated in a previous study with a much higher weed density plantation [Osco et al., 2020a], and returned a similar performance. In this manner, another study demonstrated how an advanced encoder-decoder network was able to outperform other approaches into automatically detecting both crop-rows and weed within the lines [Adhikari et al., 2019]. This type of approach could be also considered in our network. Regardless, our neural network, by accurately identifying plants and plantation-rows in both dense and sparse environments, with a one-step type of approach, may still help future research to solve part of this generalization problem when considering both plant and plantation-row identification in one single step.

As previously stated, other crops may benefit from the approach presented here. Whether in counting plants, as well as detecting existing plantation-rows. Since it uses a confidence map where it is calculated the probability that the plant or tree will occur at each pixel, it differs from common object detection deep learning methods that need labeling rectangles to detect a target. In densified plantations, this characteristic of a common object detection deep neural network can be problematic since overlapping plants may reduce the performance of the used model [Ampatzidis and Partel, 2019]. Regardless, the highest benefit from the method presented here is the incorporation of a two-branched architecture to deal with plant and plantation-row detection simultaneously on a one-step basis. Since the multiple stage refinement branches are concatenated with each other, both detection approaches (plant and plantation-row) are beneficiated from the knowledge extracted in the other counterpart. In short, as the networks update the branch of the plantation-row with information from the plant branch, the plant branch predictions are refined with information from the plantation-row branch; and vice-versa. Although we also proved, by evaluating each branch prediction individually during our experimental phase in our results section, the overall accuracy of both predictions was higher when considering this strategy.

The proposed CNN is modified to return a prediction map instead of classification. For this reason, the R^2, MAE, and MSE metrics differ from the commonly found metrics used in this situation. However, studies that approached plant detection and counting as a classification problem obtained 93.3% accuracy for rice seedlings using a combined deep network [Wu et al.,

2019], 96.2% for citrus-tree detection [Csillik et al., 2018], and more than 96.0% to count oil palm trees [Li et al., 2017]. In a study that addressed this problem as a prediction, the authors evaluated palm-trees and stated that their AlexNet CNN architecture returned 0.99 R^2 predictions, with 2.6% to 9.2% relative error, depending on the evaluated dataset [Djerriri et al., 2018]. Thus, it is evident that the CNN proposed in our study achieved results similar to or better than those existing in the current literature. It should also be considered the adverse situations evaluated here; as it presented more plants per area than the usual. Also, none of these studies implemented a plantation-row detection in their methods; which is another differential of our approach. Although many object detection deep networks can be used to detect plants and rows, they require several modifications to simultaneously perform both tasks. As mentioned, our approach uses a two-branched architecture, and one branch benefits from the other. This interaction between both branches is an important feature and does come in handy when both problems are being considered.

Currently, applications involving UAVs, RGB sensors, and deep learning models are contributing to addressing the aforementioned issues here discussed [Weinstein et al., 2019, Csillik et al., 2018, Fan et al., 2018, Ampatzidis and Partel, 2019]. But a disadvantage of deep learning, in general, is the need to label thousands of plants for the training process, as well as a high-end computer to process these data [Goldman et al., 2019]. One of the reasons for reducing accuracy in our approach was plant-occlusion and high-proximity between plants (Figure 6.7) and the occurrence of single-plants outside the plantation-rows, an uncommon spacing between plants, and some "curves" in the plantation-rows (Figure 6.8). Another issue is related to the consumed time of our approach concerning other methods (Table 6.8). This occurred mainly because our neural network computes both plants and plantation-lines, while the other compared Bounding Box methods only account for plant detection. Also, on this matter, the usage of object detectors architectures is an alternative to the adoption of confidence maps and point-labeling over the input image, but the annotation of the plants with bounding boxes, as required by neural networks such as Faster R-CNN, is more laborious when compared to a single point annotation, as required by our approach. Besides, object detectors have higher difficulties in detecting plants in dense areas as shown in previous works [Osco et al., 2020b]. Also, the method presented here can be fed with other data sources, as demonstrated with the variated dataset tested, and will require less prior information than before. In this way, where new information is incorporated into the method, more accuracy and new learning patterns can be expected to be achieved.

6.5 Remarks of the Chapter

This Chapter introduces a CNN approach to simultaneously detect plants and plantation-rows in different datasets (cornfields and citrus orchards), derived from RGB imagery acquired with a UAV-based remote system. The presented method is a feasible alternative from visual inspection and should assist in precision farming practices. It proved highly accurate results, achieving a MAE of 6.224 plants per image patch, MRE of 0.1038, Precision and Recall values of 0.856 and 0.905, respectively, and an F1-Measure equal to 0.876. These results were superior to the results from other deep networks (HRNet, Faster R-CNN, RetinaNet, YIOLOv5 and YOLOv7) evaluated with the same task and dataset. For the plantation-row detection, our approach returned precision, recall, and F1-Measure scores of 0.914, 0.941, and 0.926, respectively. To test the robustness of our model with a different type of crop, we performed the same task in the citrus orchard dataset. It returned an MAE equal to 1.409 citrus-trees per image patch, MRE of 0.0615, Precision of 0.922, Recall of 0.911, and F1-Measure of 0.965. For the citrus plantation-row detection, our approach resulted in Precision, Recall, and F1-Measure scores equal to 0.965, 0.970, and 0.964, respectively. The proposed method also has a reasonable cost alternative, since it uses an RGB-based sensor.

Another contribution of our CNN approach is that, by applying a two-branched architecture and enabling information to be exchanged between them, our approach can benefit from the results of one detection to the other. Besides, instead of using a common bounding box object-detection approach, it estimates a confidence map to detect individual plants. This presents an advantage when evaluating high-density plantations, since it does not rely on target boundaries, but it uses the probability of a unique pixel being recognized as the plant. This architecture is also beneficiated from the two branche's approach as their refinement is linked to the information exchanged between them. As some plants are naturally limited to compensate for missing areas, we recommend the method here to count and detect plantation-rows simultaneously, in a one-step architecture, since it helps to estimate plantation patterns and errors. We trust that, in the current state, the method provides an enhancement in decision-making tasks while contributing to the more sustainable management of agricultural areas by remote sensing systems. We hope that the approach presented here may assist in research regarding remote sensing technologies and precision farming applications.

CHAPTER

7

Conclusion

In this work we propose a computer vision approach to locate and count objects using a 2D confidence map. We present different applications of the method in object detection tasks and image types. We proposed a baseline method for working with RGB, Multispectral and Hyperspectral images. The method was tested with different datasets, such as eucalyptus and citrus-trees orchards, single tree species, palm trees species, cornfields (recently planted and mature-stage) and vehicle count benchmarks. Datasets have different conditions like low and high-density objects, uniform and sparse locations, and different sizes and shapes. In addition, the different types of images test the approach performance to work with different amounts of data across channels.

These applications bring improved features to the model to extract better performance in each object detection task. The first application (Chapter 3) shows that the method can estimate the number and location of citrus trees from UAV multispectral imagery. In addition, the method demonstrated reasonable computational costs for embedded real-time applications. Finally, the method was capable to detect individual trees in high-density plantations achieving a higher Precision (0.95) and a lower MAE (2.05) than object-based methods.

In the second application (Chapter 4) we present an improved model for counting and locating objects with high-density in images. Unlike the first application, we have introduced a feature map enhancement (PPM) to provide a better description of the image with global and sub-regional information. Furthermore, we propose a Multi-Sigma Stage refinement module to improve the object prediction. Different from the first application, we tested the method in

two challenging applications (tree and car counting tasks) with RGB images. The results show that the proposed method generalizes well when applied for different object detection tasks and is suitable for dealing with high object density in images, as it achieved an MAE of 4.45 and 3.16 on CARPK and PUCPR+ datasets.

The third application (Chapter 5) shows the detection of single tree species in dense scenarious. The application works with hyperspectral UAV-based images and the fourth application uses aerial RGB imagery. To handle hyperspectral images of 25 bans, we proposed a band selection module in the first stage of the network. We show that the proposed approach can handle with high (hyperspectral) amount of information and returns state-of-the-art performance to detect and locate single trees in dense scenarios.

In the most recent application (Chapter 6), we introduce a model improvement for the object counting task. We propose a CNN to simultaneously detect plants and plantation-rows in different RGB datasets (cornfields and citrus orchards). The approach presents a two-branch refinement phase with co-sharing information for the detection task, in a one-step architecture. This model shows that the information exchanged between them improves performance in object and line detection individually. The results show that the methods are suitable for counting and detecting plantation-rows. We evaluate the method with state-of-the-art methods: HRNet, Faster R-CNN, RetinaNet, YOLOv5 and YOLOv7. In the object detection task, the method achieves a MAE of 6.224 plants per image patch and a F1-Measure equal to 0.876. In the plantation row detection the method achieves a F1-Measure scores of 0.925.

In addition to the applications and improvements presented in this thesis, the proposed method was used in other counting applications such as Arce et al. [2021]. In this paper the proposed method with the PPM and MSS modules is applied over RGB images of palm trees. The method was able to identify and geolocate single species in a high-complex forested environment. The method was evaluated with counting methods: Faster R-CNN and RetinaNet, and returned an MAE of 0.75 trees and F1-Measure of 86.9%.

In future steps we intend to improve the object detection method, inserting features to assist more real applications tasks. In addition, since this thesis unifies the applications published over the doctorate, we intend to expand comparision with other recent CNNs and arquitectures. For this, we are working in adapting the proposed method to use the open-source object detection toolbox MMDetection [Chen et al., 2019], to test more backbones in the Feature Map extraction phase.

Finally, we intent to verify the impact of noise and reduced annotations for training networks. For this, we will systematically reduce the number of

images and the number of annotations for object detection problems, following the previous work by [Rolnick et al., 2017, Thulasidasan et al., 2019].

Bibliography

H. Aasen, E. Honkavaara, A. Lucieer, and P. J. Zarco-Tejada. Quantitative remote sensing at ultra-high resolution with uav spectroscopy: A review of sensor technology, measurement procedures, and data correction workflows. *Remote Sensing*, 10(7), 2018. ISSN 2072-4292. doi: 10.3390/rs10071091. Cited on page 47.

S. P. Adhikari, H. Yang, and H. Kim. Learning semantic graphics using convolutional encoder-decoder network for autonomous weeding in paddy. *Frontiers in Plant Science*, 10, 2019. ISSN 1664-462X. doi: 10.3389/fpls.2019.01404. Cited on page 90.

S. Aich and I. Stavness. Improving object counting with heatmap regulation, 2018. Cited on pages xvi, 1, 2, 4, 7, 27, 28, 29, 30, 32, 33, 40, 50, and 70.

S. Aich and I. Stavness. Global sum pooling: A generalization trick for object counting with small datasets of large images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019. Cited on pages xvi, 40, 41, and 43.

R. Alshehhi, P. R. Marpu, W. L. Woon, and M. D. Mura. Simultaneous extraction of roads and buildings in remote sensing imagery with convolutional neural networks. *ISPRS Journal of Photogrammetry and Remote Sensing*, 130:139–149, 2017. ISSN 0924-2716. doi: https://doi.org/10.1016/j.isprsjprs.2017.05.002. Cited on pages 3, 13, 49, and 65.

Y. Ampatzidis and V. Partel. Uav-based high throughput phenotyping in citrus utilizing multispectral imaging and artificial intelligence. *Remote Sensing*, 11 (4), 2019. ISSN 2072-4292. doi: 10.3390/rs11040410. Cited on pages 14, 19, 20, 49, 64, 66, 88, 89, 90, and 91.

J. An, W. Li, M. Li, S. Cui, and H. Yue. Identification and classification of maize drought stress using deep convolutional neural network. *Symmetry*, 11(2), 2019. ISSN 2073-8994. doi: 10.3390/sym11020256. Cited on page 66.

L. S. D. Arce, L. P. Osco, M. d. S. d. Arruda, D. E. G. Furuya, A. P. M. Ramos, C. Aoki, A. Pott, S. Fatholahi, J. Li, F. F. d. Araújo, et al. Mauritia flexuosa palm trees airborne mapping with deep convolutional neural network. *Scientific Reports*, 11(1):1–13, 2021. Cited on page 94.

N. Audebert, B. Le Saux, and S. Lefevre. Deep learning for classification of hyperspectral data: A comparative review. *IEEE Geoscience and Remote Sensing Magazine*, 7(2):159–173, June 2019. ISSN 2168-6831. doi: 10.1109/MGRS.2019.2912563. Cited on pages 3 and 49.

V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015. Cited on page 65.

M. D. Bah, A. Hafiane, and R. Canals. Deep learning with unsupervised data labeling for weed detection in line crops in uav images. *Remote Sensing*, 10 (11), 2018. ISSN 2072-4292. doi: 10.3390/rs10111690. Cited on pages 22 and 65.

M. D. Bah, A. Hafiane, and R. Canals. Crownet: Deep network for crop row detection in uav images. *IEEE Access*, 8:5189–5200, 2020. doi: 10.1109/ACCESS.2019.2960873. Cited on page 89.

J. E. Ball, D. T. Anderson, and C. S. C. Sr. Comprehensive survey of deep learning in remote sensing: theories, tools, and challenges for the community. *Journal of Applied Remote Sensing*, 11(4):1–54, 2017. doi: 10.1117/1.JRS.11.042609. Cited on pages 13 and 65.

M. Belgiu and L. Dragut. Random forest in remote sensing: A review of applications and future directions. *ISPRS Journal of Photogrammetry and Remote Sensing*, 114:24–31, 2016. ISSN 0924-2716. doi: https://doi.org/10.1016/j.isprsjprs.2016.01.011. Cited on page 48.

A. Berveglieri, A. M. G. Tommaselli, N. N. Imai, E. A. W. Ribeiro, R. B. Guimaraes, and E. Honkavaara. Identification of successional stages and cover changes of tropical forest based on digital surface model analysis. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(12):5385–5397, 2016. doi: 10.1109/JSTARS.2016.2606320. Cited on page 52.

A. Berveglieri, N. N. Imai, A. M. Tommaselli, B. Casagrande, and E. Honkavaara. Successional stages and their evolution in tropical forests using multi-temporal photogrammetric surface models and superpixels. *ISPRS Journal of Photogrammetry and Remote Sensing*, 146:548–558, 2018.

ISSN 0924-2716. doi: https://doi.org/10.1016/j.isprsjprs.2018.11.002. Cited on pages 2 and 52.

J. M. Bioucas-Dias, A. Plaza, G. Camps-Valls, P. Scheunders, N. Nasrabadi, and J. Chanussot. Hyperspectral remote sensing data analysis and future challenges. *IEEE Geoscience and Remote Sensing Magazine*, 1(2):6–36, 2013. doi: 10.1109/MGRS.2013.2244672. Cited on pages 3 and 49.

Brasil. Descreto s/n de 16 de julho de 2002, Jul 2002. URL http://www.planalto.gov.br/ccivil_03/dnn/2002/Dnn9609.htm. accessed on 25 October 2020. Cited on page 52.

Brasil. Descreto s/n de 14 de maio de 2004, May 2004. URL http://www.planalto.gov.br/CCIVIL_03/_Ato2004-2006/2004/Decreto/_quadro.htm. accessed on 25 October 2020. Cited on page 52.

J. Cao, W. Leng, K. Liu, L. Liu, Z. He, and Y. Zhu. Object-based mangrove species classification using unmanned aerial vehicle hyperspectral images and digital surface models. *Remote Sensing*, 10(1), 2018. ISSN 2072-4292. doi: 10.3390/rs10010089. Cited on pages 2 and 48.

Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1302–1310, 2017. doi: 10.1109/CVPR.2017.143. Cited on pages 7 and 10.

T. Chai and R. R. Draxler. Root mean square error (rmse) or mean absolute error (mae)?-arguments against avoiding rmse in the literature. *Geoscientific model development*, 7(3):1247–1250, 2014. Cited on page 34.

K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin. Mmdetection: Open mmlab detection toolbox and benchmark. *CoRR*, abs/1906.07155, 2019. Cited on page 94.

S. W. Chen, S. S. Shivakumar, S. Dcunha, J. Das, E. Okon, C. Qu, C. J. Taylor, and V. Kumar. Counting apples and oranges with deep learning: A data-driven approach. *IEEE Robotics and Automation Letters*, 2(2):781–788, 2017. doi: 10.1109/LRA.2017.2651944. Cited on pages 13, 63, and 88.

M. L. Clark and D. A. Roberts. Species-level differences in hyperspectral metrics among tropical rainforest trees as determined by a tree-based classifier. *Remote Sensing*, 4(6):1820–1855, 2012. ISSN 2072-4292. doi: 10.3390/rs4061820. Cited on page 61.

M. S. Colgan, C. A. Baldeck, J.-B. Féret, and G. P. Asner. Mapping savanna tree species at ecosystem scales using support vector machine classification and brdf correction on airborne hyperspectral and lidar data. *Remote Sensing*, 4(11):3462–3480, 2012. ISSN 2072-4292. doi: 10.3390/rs4113462. Cited on page 47.

O. Csillik, J. Cherbini, R. Johnson, A. Lyons, and M. Kelly. Identification of citrus trees from unmanned aerial vehicle imagery using convolutional neural networks. *Drones*, 2(4), 2018. ISSN 2504-446X. doi: 10.3390/drones2040039. Cited on pages 2, 14, 19, 20, 49, 64, 65, 66, 89, and 91.

F. R. da Silva, R. M. Begnini, B. C. Lopes, and T. T. Castellani. Seed dispersal and predation in the palm syagrus romanzoffiana on two islands with different faunal richness, southern brazil. *Studies on Neotropical Fauna and Environment*, 46(3):163–171, 2011. doi: 10.1080/01650521.2011.617065. Cited on page 52.

M. Dalponte, L. Bruzzone, and D. Gianelle. Tree species classification in the southern alps based on the fusion of very high geometrical resolution multispectral/hyperspectral images and lidar data. *Remote Sensing of Environment*, 123:258–270, 2012. ISSN 0034-4257. doi: https://doi.org/10.1016/j.rse.2012.03.013. Cited on page 61.

P. R. de Almeida, L. S. Oliveira, A. S. Britto, E. J. Silva, and A. L. Koerich. Pklot - a robust dataset for parking lot classification. *Expert Systems with Applications*, 42(11):4937 – 4949, 2015. ISSN 0957-4174. doi: https://doi.org/10.1016/j/eswa.2015.02.009. Cited on pages 4 and 42.

M. d. S. de Arruda, L. P. Osco, P. R. Acosta, D. N. Gonçalves, J. Marcato Junior, A. P. M. Ramos, E. T. Matsubara, Z. Luo, J. Li, J. de Andrade Silva, and W. N. Gonçalves. Counting and locating high-density objects using convolutional neural network. *Expert Systems with Applications*, 195:116555, 2022. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2022.116555. Cited on pages 28 and 76.

C. Delloye, M. Weiss, and P. Defourny. Retrieval of the canopy chlorophyll content from sentinel-2 spectral bands to estimate nitrogen uptake in intensive winter wheat cropping systems. *Remote Sensing of Environment*, 216:245–261, 2018. ISSN 0034-4257. doi: https://doi.org/10.1016/j.rse.2018.06.037. Cited on page 63.

L. Deng, Z. Mao, X. Li, Z. Hu, F. Duan, and Y. Yan. Uav-based multispectral remote sensing for precision agriculture: A comparison between different cameras. *ISPRS Journal of Photogrammetry and Remote Sensing*, 146:

124–136, 2018. ISSN 0924-2716. doi: https://doi.org/10.1016/j.isprsjprs.2018.09.008. Cited on page 64.

K. Djerriri, M. Ghabi, M. S. Karoui, and R. Adjoudj. Palm trees counting in remote sensing imagery using regression convolutional neural network. In *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, pages 2627–2630, 2018. doi: 10.1109/IGARSS.2018.8519188. Cited on pages 13, 14, 65, and 91.

M. dos Santos de Arruda, G. Spadon, J. F. Rodrigues, W. N. Gonçalves, and B. B. Machado. Recognition of endangered pantanal animal species using deep learning methods. In *International Joint Conference on Neural Networks*, pages 1–8, July 2018. doi: 10.1109/IJCNN.2018.8489369. Cited on page 1.

N. R. Draper and H. Smith. *Applied regression analysis*, volume 326. John Wiley & Sons, 1998. Cited on page 34.

G. A. Elias, R. Colares, A. R. Antunes, P. T. Padilha, J. M. T. Lima, and R. Santos. Palm (arecaceae) communities in the brazilian atlantic forest: a phytosociological study. *Floresta e Ambiente*, 26, 2019. ISSN 2179-8087. Cited on page 52.

Z. Fan, J. Lu, M. Gong, H. Xie, and E. D. Goodman. Automatic tobacco plant detection in uav images via deep neural networks. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(3):876–887, 2018. doi: 10.1109/JSTARS.2018.2793849. Cited on pages 19, 21, 64, 65, 66, 67, 88, and 91.

L. Fiaschi, U. Koethe, R. Nair, and F. A. Hamprecht. Learning to count with regression forest and structured labels. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 2685–2688, 2012. Cited on pages 2 and 28.

M. Freudenberg, N. Nölke, A. Agostini, K. Urban, F. Wörgötter, and C. Kleinn. Large scale palm tree detection in high resolution satellite images using u-net. *Remote Sensing*, 11(3), 2019. ISSN 2072-4292. doi: 10.3390/rs11030312. Cited on page 66.

P. Ghamisi, J. Plaza, Y. Chen, J. Li, and A. J. Plaza. Advanced spectral classifiers for hyperspectral images: A review. *IEEE Geoscience and Remote Sensing Magazine*, 5(1):8–32, 2017. doi: 10.1109/MGRS.2016.2616418. Cited on page 65.

M. I. Giombini, S. P. Bravo, Y. V. Sica, and D. S. Tosto. Early genetic consequences of defaunation in a large-seeded vertebrate-dispersed palm (syagrus romanzoffiana). *Heredity*, 118(6):568–577, 2017. Cited on pages 52 and 53.

F. Gnädinger and U. Schmidhalter. Digital counts of maize plants by unmanned aerial vehicles (uavs). *Remote Sensing*, 9(6), 2017. ISSN 2072-4292. doi: 10.3390/rs9060544. Cited on page 88.

G. Goldbergs, S. W. Maier, S. R. Levick, and A. Edwards. Efficiency of individual tree detection approaches based on light-weight and low-cost uas imagery in australian savannas. *Remote Sensing*, 10(2), 2018. ISSN 2072-4292. doi: 10.3390/rs10020161. Cited on page 14.

E. Goldman, R. Herzig, A. Eisenschtat, J. Goldberger, and T. Hassner. Precise detection in densely packed scenes. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 5227–5236, June 2019. Cited on pages 1, 2, 27, 28, 33, 40, and 91.

N. Guimarães, L. Pádua, P. Marques, N. Silva, E. Peres, and J. J. Sousa. Forestry remote sensing from unmanned aerial vehicles: A review focusing on the data, processing and potentialities. *Remote Sensing*, 12(6), 2020. ISSN 2072-4292. doi: 10.3390/rs12061046. Cited on page 47.

S. Hartling, V. Sagan, P. Sidike, M. Maimaitijiang, and J. Carron. Urban tree species classification using a worldview-2/3 and lidar data fusion approach and deep learning. *Sensors*, 19(6), 2019. ISSN 1424-8220. doi: 10.3390/s19061284. Cited on pages 2, 64, and 65.

M. M. Hasan, J. P. Chopin, H. Laga, and S. J. Miklavcic. Detection and analysis of wheat spikes using convolutional neural networks. *Plant Methods*, 14 (1):1–13, 2018. Cited on page 13.

M. Hassanein, M. Khedr, and N. El-Sheimy. Crop row detection procedure using low-cost uav imagery system. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 2019. Cited on pages 64 and 67.

K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):386–397, 2020. ISSN 1939-3539. Cited on page 1.

A. Hennessy, K. Clarke, and M. Lewis. Hyperspectral classification of plants: A review of waveband selection generalisability. *Remote Sensing*, 12(1), 2020. ISSN 2072-4292. doi: 10.3390/rs12010113. Cited on pages 3 and 48.

D. Ho Tong Minh, D. Ienco, R. Gaetano, N. Lalande, E. Ndikumana, F. Osman, and P. Maurel. Deep recurrent neural networks for winter vegetation quality mapping via multitemporal sar sentinel-1. *IEEE Geoscience and Remote Sensing Letters*, 15(3):464–468, 2018. doi: 10.1109/LGRS.2018.2794581. Cited on page 64.

E. Honkavaara and E. Khoramshahi. Radiometric correction of close-range spectral image blocks captured using an unmanned aerial vehicle with a radiometric block adjustment. *Remote Sensing*, 10(2), 2018. ISSN 2072-4292. doi: 10.3390/rs10020256. Cited on page 54.

E. Honkavaara, H. Saari, J. Kaivosoja, I. Polonen, T. Hakala, P. Litkey, J. Makynen, and L. Pesonen. Processing and assessment of spectrometric, stereoscopic imagery collected using a lightweight uav spectral camera for precision agriculture. *Remote Sensing*, 5(10):5006–5039, 2013. ISSN 2072-4292. doi: 10.3390/rs5105006. Cited on page 54.

E. Honkavaara, T. Rosnell, R. Oliveira, and A. Tommaselli. Band registration of tuneable frame format hyperspectral uav imagers in complex scenes. *ISPRS Journal of Photogrammetry and Remote Sensing*, 134:96–109, 2017. ISSN 0924-2716. doi: https://doi.org/10.1016/j.isprsjprs.2017.10.014. Cited on page 54.

M. Hsieh, Y. Lin, and W. H. Hsu. Drone-based object counting by spatially regularized regional proposal network. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 4165–4173, 2017. doi: 10.1109/ICCV.2017.446. Cited on pages 1, 4, 28, 33, 40, 41, and 42.

X. Huang, X. Liu, and L. Zhang. A multichannel gray level co-occurrence matrix for multi/hyperspectral image texture representation. *Remote Sensing*, 6(9):8424–8445, 2014. ISSN 2072-4292. doi: 10.3390/rs6098424. Cited on page 64.

M. L. Hunt, G. A. Blackburn, L. Carrasco, J. W. Redhead, and C. S. Rowland. High resolution wheat yield mapping using sentinel-2. *Remote Sensing of Environment*, 233:111410, 2019. ISSN 0034-4257. doi: https://doi.org/10.1016/j.rse.2019.111410. Cited on page 63.

M. Hussain, D. Chen, A. Cheng, H. Wei, and D. Stanley. Change detection from remotely sensed images: From pixel-based to object-based approaches. *ISPRS Journal of Photogrammetry and Remote Sensing*, 80:91–106, 2013. ISSN 0924-2716. doi: https://doi.org/10.1016/j.isprsjprs.2013.03.006. Cited on page 64.

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *ArXiv e-prints*, 2015. Cited on page 31.

M. K. Jakubowski, W. Li, Q. Guo, and M. Kelly. Delineating individual trees from lidar data: A comparison of vector- and raster-based segmentation approaches. *Remote Sensing*, 5(9):4163–4186, 2013. ISSN 2072-4292. doi: 10.3390/rs5094163. Cited on page 64.

J. R. Jensen. *Introductory digital image processing: a remote sensing perspective.* Number Ed.2 in 2. Prentice-Hall Inc., Upper Saddle River, USA, 1996. Cited on page 64.

J. R. Jensen. Remote sensing of the environment an earth resource perspective. *Upper Saddle River (NJ), USA*, 2000. Cited on page 61.

H. Jiang, S. Chen, D. Li, C. Wang, and J. Yang. Papaya tree detection with uav images using a gpu-accelerated scale-space filtering method. *Remote Sensing*, 9(7), 2017. ISSN 2072-4292. doi: 10.3390/rs9070721. Cited on pages 13 and 64.

Y. Jiang, C. Li, A. H. Paterson, and J. S. Robertson. Deepseedling: Deep convolutional network and kalman filter for plant seedling detection and counting in the field. *Plant methods*, 15(1):1–19, 2019. Cited on page 64.

S. Jin, Y. Su, S. Gao, F. Wu, T. Hu, J. Liu, W. Li, D. Wang, S. Chen, Y. Jiang, S. Pang, and Q. Guo. Deep learning: Individual maize segmentation from terrestrial lidar data using faster r-cnn and regional growth algorithms. *Frontiers in Plant Science*, 9, 2018. ISSN 1664-462X. doi: 10.3389/fpls.2018.00866. Cited on page 63.

Z. Jin, G. Azzari, C. You, S. Di Tommaso, S. Aston, M. Burke, and D. B. Lobell. Smallholder maize area and yield mapping at national scales with google earth engine. *Remote Sensing of Environment*, 228:115–128, 2019. ISSN 0034-4257. doi: https://doi.org/10.1016/j.rse.2019.04.016. Cited on page 63.

G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, NanoCode012, Y. Kwon, TaoXie, and K. Michael. ultralytics/yolov5: v6.2 - YOLOv5 Classification Models, Apple M1, Reproducibility, ClearML and Deci.ai integrations, Aug. 2022. URL https://doi.org/10.5281/zenodo.7002879. Cited on pages 4 and 81.

R. A. Johnson and D. Wichern. Applied multivariate statistical analysis. *Statistics*, 6215(10):10, 2015. Cited on page 49.

E. R. H. Jr. and C. S. T. Daughtry. What good are unmanned aircraft systems for agricultural remote sensing and precision agriculture? *International Journal of Remote Sensing*, 39(15-16):5345–5376, 2018. doi: 10.1080/01431161.2017.1410300. Cited on page 63.

A. Kamilaris and F. X. Prenafeta-Boldu. Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, 147:70–90, 2018. ISSN 0168-1699. doi: https://doi.org/10.1016/j.compag.2018.02.016. Cited on pages 13 and 65.

D. Kang, Z. Ma, and A. B. Chan. Beyond counting: Comparisons of density maps for crowd analysis tasks-counting, detection, and tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(5):1408–1422, 2019. doi: 10.1109/TCSVT.2018.2837153. Cited on page 14.

A. Khamparia and K. M. Singh. A systematic review on deep learning architectures and applications. *Expert Systems*, 36(3):e12400, 2019. doi: https://doi.org/10.1111/exsy.12400. e12400 EXSY-Jul-18-241.R3. Cited on pages 48 and 65.

B. T. Kitano, C. C. T. Mendes, A. R. Geus, H. C. Oliveira, and J. R. Souza. Corn plant counting using deep learning and uav images. *IEEE Geoscience and Remote Sensing Letters*, pages 1–5, 2019. doi: 10.1109/LGRS.2019.2930549. Cited on pages 63, 66, and 88.

A. Krizhevsky. One weird trick for parallelizing convolutional neural networks, 2014. Cited on page 65.

M. Larsen, M. Eriksson, X. Descombes, G. Perrin, T. Brandtberg, and F. A. Gougeon. Comparison of six individual tree crown detection algorithms evaluated under varying forest conditions. *International Journal of Remote Sensing*, 32(20):5827–5852, 2011. doi: 10.1080/01431161.2010.507790. Cited on page 64.

Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436–444, 2015. Cited on page 65.

J. N. Leiva, J. Robbins, D. Saraswat, Y. She, and R. J. Ehsani. Evaluating remotely sensed plant count accuracy with differing unmanned aircraft system altitudes, physical canopy separations, and ground covers. *Journal of Applied Remote Sensing*, 11(3):1–15, 2017. doi: 10.1117/1.JRS.11.036003. Cited on pages 13, 14, and 64.

D. Li, H. Guo, C. Wang, W. Li, H. Chen, and Z. Zuo. Individual tree delineation in windbreaks using airborne-laser-scanning data and unmanned aerial vehicle stereo images. *IEEE Geoscience and Remote Sensing Letters*, 13(9): 1330–1334, 2016. doi: 10.1109/LGRS.2016.2584109. Cited on page 64.

W. Li, H. Fu, L. Yu, and A. Cracknell. Deep learning based oil palm tree detection and counting for high-resolution remote sensing images. *Remote Sensing*, 9(1), 2017. ISSN 2072-4292. doi: 10.3390/rs9010022. Cited on pages 13, 48, 65, and 91.

T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(2):318–327, 2020. ISSN 1939-3539. doi: 10.1109/TPAMI.2018.2858826. Cited on pages 1, 4, 33, 40, and 66.

T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. Cited on page 48.

L. Liu, B. Song, S. Zhang, and X. Liu. A novel principal component analysis method for the reconstruction of leaf reflectance spectra and retrieval of leaf biochemical contents. *Remote Sensing*, 9(11), 2017. ISSN 2072-4292. doi: 10.3390/rs9111113. Cited on pages 3 and 49.

T. Liu and A. Abd-Elrahman. Deep convolutional neural network training enrichment using multi-view object-based analysis of unmanned aerial systems imagery for wetlands classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 139:154–170, 2018. ISSN 0924-2716. doi: https://doi.org/10.1016/j.isprsjprs.2018.03.006. Cited on page 13.

T. Liu, A. Abd-Elrahman, J. Morton, and V. L. Wilhelm. Comparing fully convolutional networks, random forest, support vector machine, and patch-based deep convolutional neural networks for object-based wetland mapping using images from small unmanned aircraft system. *GIScience & Remote Sensing*, 55(2):243–264, 2018. doi: 10.1080/15481603.2018.1426091. Cited on page 13.

Y. Liu, M. Shi, Q. Zhao, and X. Wang. Point in, box out: Beyond counting persons in crowds. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. Cited on pages 2 and 28.

D. Lobo Torres, R. Queiroz Feitosa, P. Nigri Happ, L. Elena Cué La Rosa, J. Marcato Junior, J. Martins, P. Olã Bressan, W. N. Gonçalves, and

V. Liesenberg. Applying fully convolutional architectures for semantic segmentation of a single tree species in urban environment on high resolution uav optical imagery. *Sensors*, 20(2), 2020. ISSN 1424-8220. doi: 10.3390/s20020563. Cited on pages 2 and 48.

H. Lorenzi et al. Arvores brasileiras. *Instituto Plantarum de Estudos da Flora*, 1992. Cited on page 53.

L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson. Deep learning in remote sensing applications: A meta-analysis and review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 152:166–177, 2019. ISSN 0924-2716. doi: https://doi.org/10.1016/j.isprsjprs.2019.04.015. Cited on pages 13 and 48.

W. Ma, Y. Wu, F. Cen, and G. Wang. Mdfn: Multi-scale deep feature learning network for object detection. *Pattern Recognition*, 100:107149, 2020. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2019.107149. Cited on page 1.

S. Madec, X. Jin, H. Lu, B. De Solan, S. Liu, F. Duyme, E. Heritier, and F. Baret. Ear density estimation from high resolution rgb imagery using deep learning technique. *Agricultural and Forest Meteorology*, 264:225–234, 2019. ISSN 0168-1923. doi: https://doi.org/10.1016/j.agrformet.2018.10.013. Cited on page 13.

J. Maschler, C. Atzberger, and M. Immitzer. Individual tree crown segmentation and classification of 13 tree species using airborne hyperspectral data. *Remote Sensing*, 10(8), 2018. ISSN 2072-4292. doi: 10.3390/rs10081218. Cited on pages 3 and 49.

A. E. Maxwell, T. A. Warner, and F. Fang. Implementation of machine-learning classification in remote sensing: an applied review. *International Journal of Remote Sensing*, 39(9):2784–2817, 2018. doi: 10.1080/01431161.2018.1433343. Cited on page 48.

C. P. Mendes, M. C. Ribeiro, and M. Galetti. Patch size, shape and edge distance influence seed predation on a palm species in the atlantic forest. *Ecography*, 39(5):465–475, 2016. doi: https://doi.org/10.1111/ecog.01592. Cited on page 53.

G. T. Miyoshi, M. d. S. de Arruda, L. P. Osco, J. Marcato Junior, D. N. Gonçalves, N. N. Imai, A. M. G. Tommaselli, E. Honkavaara, and W. N. Gonçalves. A novel deep learning method to identify single tree species in uav-based hyperspectral images. *Remote Sensing*, 12(8), 2020. ISSN 2072-4292. doi: 10.3390/rs12081294. Cited on pages 2, 49, and 70.

K. Mochida, S. Koda, K. Inoue, T. Hirayama, S. Tanaka, R. Nishii, and F. Melgani. Computer vision-based phenotyping for improvement of plant productivity: a machine learning perspective. *GigaScience*, 8(1), 12 2018. ISSN 2047-217X. doi: 10.1093/gigascience/giy153. Cited on page 66.

S. K. Mohanty and M. R. Swain. Bioethanol production from corn and wheat: Food, fuel, and future. In R. C. Ray and S. Ramachandran, editors, *Bioethanol Production from Food Crops*, pages 45–59. Academic Press, 2019. ISBN 978-0-12-813766-6. doi: https://doi.org/10.1016/B978-0-12-813766-6.00003-5. Cited on page 66.

T. Moon. The expectation-maximization algorithm. *IEEE Signal Processing Magazine*, 13(6):47–60, 1996. doi: 10.1109/79.543975. Cited on page 2.

T. N. Mundhenk, G. Konjevod, W. A. Sakla, and K. Boakye. A large contextual dataset for classification, detection and counting of cars with deep learning. In B. Leibe, J. Matas, N. Sebe, and M. Welling, editors, *Computer Vision - ECCV 2016*, pages 785–800, Cham, 2016. Springer International Publishing. ISBN 978-3-319-46487-9. Cited on pages 33 and 40.

R. Nasi, E. Honkavaara, P. Lyytikainen-Saarenmaa, M. Blomqvist, P. Litkey, T. Hakala, N. Viljanen, T. Kantola, T. Tanhuanpaa, and M. Holopainen. Using uav-based photogrammetry and hyperspectral imaging for mapping bark beetle damage at tree-level. *Remote Sensing*, 7(11):15467–15493, 2015. ISSN 2072-4292. doi: 10.3390/rs71115467. Cited on page 47.

S. Natesan, C. Armenakis, and U. Vepakomma. Resnet-based tree species classification using uav images. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 2019. Cited on page 61.

E. Ndikumana, D. Ho Tong Minh, N. Baghdadi, D. Courault, and L. Hossard. Deep recurrent neural network for agricultural classification using multitemporal sar sentinel-1 for camargue, france. *Remote Sensing*, 10(8), 2018. ISSN 2072-4292. doi: 10.3390/rs10081217. Cited on page 64.

O. Nevalainen, E. Honkavaara, S. Tuominen, N. Viljanen, T. Hakala, X. Yu, J. Hyyppa, H. Saari, I. Polonen, N. N. Imai, and A. M. G. Tommaselli. Individual tree detection and classification with uav-based photogrammetric point clouds and hyperspectral imaging. *Remote Sensing*, 9(3), 2017. ISSN 2072-4292. doi: 10.3390/rs9030185. Cited on pages 48 and 64.

S. Nezami, E. Khoramshahi, O. Nevalainen, I. Polonen, and E. Honkavaara. Tree species classification of drone hyperspectral and rgb imagery with deep learning convolutional neural networks. *Remote Sensing*, 12(7), 2020. ISSN 2072-4292. doi: 10.3390/rs12071070. Cited on pages 48 and 61.

E. Ohn-Bar and M. M. Trivedi. Multi-scale volumes for deep object detection and localization. *Pattern Recognition*, 61:557 – 572, 2017. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2016.06.002. Cited on page 1.

H. C. Oliveira, V. C. Guizilini, I. P. Nunes, and J. R. Souza. Failure detection in row crops from uav images using morphological operators. *IEEE Geoscience and Remote Sensing Letters*, 15(7):991–995, 2018. doi: 10.1109/LGRS.2018.2819944. Cited on pages 22, 63, 64, and 67.

M. Onishi and T. Ise. Automatic classification of trees using a UAV onboard camera and deep learning. *CoRR*, abs/1804.10390, 2018. Cited on page 14.

L. P. Osco, M. dos Santos de Arruda, J. M. Junior, N. B. da Silva, A. P. M. Ramos, E. A. S. Moryia, N. N. Imai, D. R. Pereira, J. E. Creste, E. T. Matsubara, J. Li, and W. N. Gonçalves. A convolutional neural network approach for counting and geolocating citrus-trees in uav multispectral imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 160:97 – 106, 2020a. ISSN 0924-2716. doi: https://doi.org/10.1016/j.isprsjprs.2019.12.010. Cited on pages 14, 28, 49, 59, 65, 66, 67, and 90.

L. P. Osco, A. P. M. Ramos, M. M. Faita Pinheiro, É. A. S. Moriya, N. N. Imai, N. Estrabis, F. Ianczyk, F. F. d. Araújo, V. Liesenberg, L. A. d. C. Jorge, J. Li, L. Ma, W. N. Gonçalves, J. Marcato Junior, and J. Eduardo Creste. A machine learning framework to predict nutrient content in valencia-orange leaf hyperspectral measurements. *Remote Sensing*, 12(6), 2020b. ISSN 2072-4292. doi: 10.3390/rs12060906. Cited on pages 63 and 91.

L. P. Osco, M. d. S. de Arruda, D. N. Gonçalves, A. Dias, J. Batistoti, M. de Souza, F. D. G. Gomes, A. P. M. Ramos, L. A. de Castro Jorge, V. Liesenberg, and et al. A cnn approach to simultaneously count plants and detect plantation-rows from uav imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 174:1–17, February 2021. ISSN 0924-2716. doi: 10.1016/j.isprsjprs.2021.01.024. Cited on page 67.

A. H. Ozcan, D. Hisar, Y. Sayar, and C. Unsalan. Tree crown detection and delineation in satellite images using probabilistic voting. *Remote Sensing Letters*, 8(8):761–770, 2017. doi: 10.1080/2150704X.2017.1322733. Cited on pages 49 and 64.

A. Ozdarici-Ok. Automatic detection and delineation of citrus trees from vhr satellite imagery. *International Journal of Remote Sensing*, 36(17):4275–4296, 2015. doi: 10.1080/01431161.2015.1079663. Cited on pages 2, 14, and 64.

M. Paoletti, J. Haut, J. Plaza, and A. Plaza. A new deep convolutional neural network for fast hyperspectral image classification. *ISPRS Journal of Photogrammetry and Remote Sensing*, 145:120–147, 2018. ISSN 0924-2716. doi: https://doi.org/10.1016/j.isprsjprs.2017.11.021. Deep Learning RS Data. Cited on pages 13 and 64.

T. D. Pham, N. Yokoya, D. T. Bui, K. Yoshino, and D. A. Friess. Remote sensing approaches for monitoring mangrove species, structure, and biomass: Opportunities and challenges. *Remote Sensing*, 11(3), 2019. ISSN 2072-4292. doi: 10.3390/rs11030230. Cited on page 48.

L. Prado Osco, A. P. Marques Ramos, D. Roberto Pereira, É. Akemi Saito Moriya, N. Nobuhiro Imai, E. Takashi Matsubara, N. Estrabis, M. de Souza, J. Marcato Junior, W. N. Gonçalves, J. Li, V. Liesenberg, and J. Eduardo Creste. Predicting canopy nitrogen content in citrus-trees using random forest algorithm associated to spectral vegetation indices from uav-imagery. *Remote Sensing*, 11(24), 2019. ISSN 2072-4292. doi: 10.3390/rs11242925. Cited on page 63.

J. Primicerio, G. Caruso, L. Comba, A. Crisci, P. Gay, S. Guidoni, L. Genesio, D. R. Aimonino, and F. P. Vaccari. Individual plant definition and missing plant characterization in vineyards from high-resolution uav imagery. *European Journal of Remote Sensing*, 50(1):179–186, 2017. doi: 10.1080/22797254.2017.1308234. Cited on pages 67 and 88.

L. Quan, H. Feng, Y. Lv, Q. Wang, C. Zhang, J. Liu, and Z. Yuan. Maize seedling detection under different growth stages and complex field environments based on an improved faster r-cnn. *Biosystems Engineering*, 184:1–23, 2019. ISSN 1537-5110. doi: https://doi.org/10.1016/j.biosystemseng.2019.05.002. Cited on pages 74 and 88.

J. Redmon and A. Farhadi. Yolo9000: Better, faster, stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525, 2017. doi: 10.1109/CVPR.2017.690. Cited on pages 4, 33, and 40.

J. Redmon and A. Farhadi. Yolov3: An incremental improvement, 2018. Cited on pages 48 and 66.

J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016. doi: 10.1109/CVPR.2016.91. Cited on pages 33 and 40.

B. P. Reis, S. a. V. Martins, E. I. Fernandes Filho, T. S. Sarcinelli, J. M. Gleriani, G. E. Marcatti, H. G. Leite, and M. Halassy. Management recommendation generation for areas under forest restoration process through images obtained by uav and lidar. *Remote Sensing*, 11(13), 2019. ISSN 2072-4292. doi: 10.3390/rs11131508. Cited on page 47.

S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 91–99. Curran Associates, Inc., 2015. Cited on pages 48 and 66.

S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 39(6):1137–1149, June 2017. ISSN 0162-8828. doi: 10.1109/TPAMI.2016.2577031. Cited on pages 1, 4, 33, and 40.

J. Ribera, Y. Chen, C. Boomsma, and E. J. Delp. Counting plants using deep learning. In *2017 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 1344–1348, 2017. doi: 10.1109/GlobalSIP.2017.8309180. Cited on page 65.

A. Richards John and J. Xiuping. Remote sensing digital image analysis: an introduction, 1999. Cited on pages 3 and 49.

D. Rolnick, A. Veit, S. J. Belongie, and N. Shavit. Deep learning is robust to massive label noise. *CoRR*, abs/1705.10694, 2017. Cited on page 95.

O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, and A. F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention - MICCAI*, pages 234–241, Cham, 2015. Springer International Publishing. ISBN 978-3-319-24574-4. Cited on page 65.

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y. Cited on pages 2 and 27.

N. Saarinen, M. Vastaranta, R. Nasi, T. Rosnell, T. Hakala, E. Honkavaara, M. A. Wulder, V. Luoma, A. M. G. Tommaselli, N. N. Imai, E. A. W. Ribeiro, R. B. Guimaraes, M. Holopainen, and J. Hyyppa. Assessing biodiversity

in boreal forests with uav-based photogrammetric point clouds and hyperspectral imaging. *Remote Sensing*, 10(2), 2018. ISSN 2072-4292. doi: 10.3390/rs10020338. Cited on page 47.

A. Safonova, S. Tabik, D. Alcaraz-Segura, A. Rubtsov, Y. Maglinets, and F. Herrera. Detection of fir trees (abies sibirica) damaged by the bark beetle in unmanned aerial vehicle images with deep learning. *Remote Sensing*, 11 (6), 2019. ISSN 2072-4292. doi: 10.3390/rs11060643. Cited on pages 48, 61, 65, and 88.

E. Salami, A. Gallardo, G. Skorobogatov, and C. Barrado. On-the-fly olive tree counting using a uas and cloud services. *Remote Sensing*, 11(3), 2019. ISSN 2072-4292. doi: 10.3390/rs11030316. Cited on pages 14 and 88.

A. A. d. Santos, J. Marcato Junior, M. S. Araújo, D. R. Di Martini, E. C. a. Tetila, H. L. Siqueira, C. Aoki, A. Eltner, E. T. Matsubara, H. Pistori, R. Q. Feitosa, V. Liesenberg, and W. N. Gonçalves. Assessment of cnn-based methods for individual tree detection on images captured by rgb cameras attached to uavs. *Sensors*, 19(16), 2019. ISSN 1424-8220. doi: 10.3390/s19163595. Cited on pages 2, 48, 49, 61, 65, and 66.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *ArXiv e-prints*, September 2014. Cited on pages xvi, 9, 29, 30, 65, and 71.

V. A. Sindagi and V. M. Patel. A survey of recent advances in cnn-based single image crowd counting and density estimation. *Pattern Recognition Letters*, 107:3 – 16, 2018. ISSN 0167-8655. doi: https://doi.org/10.1016/j.patrec.2017.07.007. Cited on page 1.

G. M. Smith and E. J. Milton. The use of the empirical line method to calibrate remotely sensed data to reflectance. *International Journal of Remote Sensing*, 20(13):2653–2662, 1999. doi: 10.1080/014311699211994. Cited on page 54.

C. Sothe, C. M. D. Almeida, M. B. Schimalski, L. E. C. L. Rosa, J. D. B. Castro, R. Q. Feitosa, M. Dalponte, C. L. Lima, V. Liesenberg, G. T. Miyoshi, and A. M. G. Tommaselli. Comparative performance of convolutional neural network, weighted and conventional support vector machine and random forest for classifying tree species using hyperspectral and photogrammetric data. *GIScience & Remote Sensing*, 57(3):369–394, 2020. doi: 10.1080/15481603.2020.1712102. Cited on pages 48 and 61.

T. Stahl, S. L. Pintea, and J. C. van Gemert. Divide and count: Generic object counting by image divisions. *IEEE Transactions on Image Processing*, 28

(2):1035–1044, 2019. ISSN 1057-7149. doi: 10.1109/TIP.2018.2875353. Cited on pages 33 and 40.

M. Story and R. G. Congalton. Accuracy assessment: a users perspective. *Photogrammetric Engineering and remote sensing*, 52(3):397–399, 1986. Cited on page 55.

J. Sun, L. Di, Z. Sun, Y. Shen, and Z. Lai. County-level soybean yield prediction using deep cnn-lstm model. *Sensors*, 19(20), 2019. ISSN 1424-8220. doi: 10.3390/s19204363. Cited on page 63.

P. Surovỳ, N. A. Ribeiro, and D. Panagiotidis. Estimation of positions and heights from uav-sensed imagery in tree plantations in agrosilvopastoral systems. *International Journal of Remote Sensing*, 39(14):4786–4800, 2018. doi: 10.1080/01431161.2018.1434329. Cited on page 64.

J.-D. Sylvain, G. Drolet, and N. Brown. Mapping dead forest cover using a deep convolutional neural network and digital aerial photography. *ISPRS Journal of Photogrammetry and Remote Sensing*, 156:14–26, 2019. ISSN 0924-2716. doi: https://doi.org/10.1016/j.isprsjprs.2019.07.010. Cited on page 65.

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. Cited on page 65.

C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. Cited on page 65.

C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. *ArXiv e-prints*, 2017. Cited on page 65.

G. Takahashi Miyoshi, N. N. Imai, A. M. Garcia Tommaselli, M. V. Antunes de Moraes, and E. Honkavaara. Evaluation of hyperspectral multitemporal information to improve tree species identification in the highly diverse atlantic forest. *Remote Sensing*, 12(2), 2020. ISSN 2072-4292. doi: 10.3390/rs12020244. Cited on pages 2, 52, 60, and 61.

S. Tao, F. Wu, Q. Guo, Y. Wang, W. Li, B. Xue, X. Hu, P. Li, D. Tian, C. Li, H. Yao, Y. Li, G. Xu, and J. Fang. Segmenting tree crowns from terrestrial and mobile lidar data by exploring ecological theories. *ISPRS Journal of*

Photogrammetry and Remote Sensing, 110:66–76, 2015. ISSN 0924-2716. doi: https://doi.org/10.1016/j.isprsjprs.2015.10.007. Cited on page 64.

S. Thulasidasan, T. Bhattacharya, J. Bilmes, G. Chennupati, and J. Mohd-Yusof. Combating label noise in deep learning using abstention, 2019. Cited on page 95.

S. Tuominen, R. Nasi, E. Honkavaara, A. Balazs, T. Hakala, N. Viljanen, I. Polonen, H. Saari, and H. Ojanen. Assessment of classifiers and remote sensing features of hyperspectral imagery and stereo-photogrammetric point clouds for recognition of tree species in a forest area of high species diversity. *Remote Sensing*, 10(5), 2018. ISSN 2072-4292. doi: 10.3390/rs10050714. Cited on pages 3 and 49.

S. Varela, P. R. Dhodda, W. H. Hsu, P. V. V. Prasad, Y. Assefa, N. R. Peralta, T. Griffin, A. Sharda, A. Ferguson, and I. A. Ciampitti. Early-season stand count determination in corn via integration of imagery from unmanned aerial systems (uas) and supervised learning techniques. *Remote Sensing*, 10(2), 2018. ISSN 2072-4292. doi: 10.3390/rs10020343. Cited on pages 13, 19, 64, 66, 67, and 88.

N. K. Verma, D. W. Lamb, N. Reid, and B. Wilson. Comparison of canopy volume measurements of scattered eucalypt farm trees derived from high spatial resolution imagery and lidar. *Remote Sensing*, 8(5), 2016. ISSN 2072-4292. doi: 10.3390/rs8050388. Cited on page 64.

R. F. Vieira, J. Camillo, and L. Coradin. Espécies nativas da flora brasileira de valor econômico atual ou potencial: plantas para o futuro: Região centro-oeste., Mar 2019. accessed on 14 January 2021. Cited on page 53.

D. Wackerly, W. Mendenhall, and R. L. Scheaffer. *Mathematical statistics with applications*. Cengage Learning, 2014. Cited on page 34.

C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, 2022. URL https://arxiv.org/abs/2207.02696. Cited on pages 4 and 81.

H. Wang, R. Magagi, K. Goïta, M. Trudel, H. McNairn, and J. Powers. Crop phenology retrieval via polarimetric sar decomposition and random forest algorithm. *Remote Sensing of Environment*, 231:111234, 2019a. ISSN 0034-4257. doi: https://doi.org/10.1016/j.rse.2019.111234. Cited on page 63.

S. Wang, G. Azzari, and D. B. Lobell. Crop type mapping without field-level labels: Random forest transfer and unsupervised clustering techniques. *Re-*

mote Sensing of Environment, 222:303–317, 2019b. ISSN 0034-4257. doi: https://doi.org/10.1016/j.rse.2018.12.026. Cited on page 63.

Y. Wang, J. Hou, X. Hou, and L. P. Chau. A self-training approach for point-supervised object detection and counting in crowds. *IEEE Transactions on Image Processing*, 30:2876–2887, 2021. doi: 10.1109/TIP.2021.3055632. Cited on pages 40, 41, and 42.

B. G. Weinstein, S. Marconi, S. Bohlman, A. Zare, and E. White. Individual tree-crown detection in rgb imagery using semi-supervised deep learning neural networks. *Remote Sensing*, 11(11), 2019. ISSN 2072-4292. doi: 10.3390/rs11111309. Cited on pages 2, 19, 64, 65, and 91.

M. Weiss, F. Jacob, and G. Duveiller. Remote sensing for agricultural applications: A meta-review. *Remote Sensing of Environment*, 236:111402, 2020. ISSN 0034-4257. doi: https://doi.org/10.1016/j.rse.2019.111402. Cited on page 63.

J. Wu, G. Yang, X. Yang, B. Xu, L. Han, and Y. Zhu. Automatic counting of in situ rice seedlings from uav images based on a deep fully convolutional neural network. *Remote Sensing*, 11(6), 2019. ISSN 2072-4292. doi: 10.3390/rs11060691. Cited on pages 13, 64, 65, 88, and 90.

Z. Xie, Y. Chen, D. Lu, G. Li, and E. Chen. Classification of land cover, forest, and tree species classes with ziyuan-3 multispectral and stereo data. *Remote Sensing*, 11(2), 2019. ISSN 2072-4292. doi: 10.3390/rs11020164. Cited on page 48.

J. Xu, W. Wang, H. Wang, and J. Guo. Multi-model ensemble with rich spatial information for object detection. *Pattern Recognition*, 99:107098, 2020. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2019.107098. Cited on page 1.

N. Xu, J. Tian, Q. Tian, K. Xu, and S. Tang. Analysis of vegetation red edge with different illuminated/shaded canopy proportions and to construct normalized difference canopy shadow index. *Remote Sensing*, 11(10), 2019. ISSN 2072-4292. doi: 10.3390/rs11101192. Cited on page 19.

J. Yuan, H.-C. Xiong, Y. Xiao, W. Guan, M. Wang, R. Hong, and Z.-Y. Li. Gated cnn: Integrating multi-scale feature layers for object detection. *Pattern Recognition*, page 107131, 2019. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2019.107131. Cited on page 1.

C. YuanQiang, D. Du, L. Zhang, L. Wen, W. Wang, Y. Wu, and S. Lyu. Guided attention network for object detection and counting on drones. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 709–717. Association for Computing Machinery, 2020. ISBN 9781450379885. doi: 10.1145/3394171.3413816. Cited on pages 40, 41, and 42.

H. Zhang, Y. Li, Y. Zhang, and Q. Shen. Spectral-spatial classification of hyperspectral imagery using a dual-channel convolutional neural network. *Remote Sensing Letters*, 8(5):438–447, 2017. doi: 10.1080/2150704X.2017.1280200. Cited on page 65.

L. Zhang, L. Zhang, and B. Du. Deep learning for remote sensing data: A technical tutorial on the state of the art. *IEEE Geoscience and Remote Sensing Magazine*, 4(2):22–40, 2016. doi: 10.1109/MGRS.2016.2540798. Cited on page 13.

S. Zhang, H. Li, W. Kong, L. Wang, and X. Niu. An object counting network based on hierarchical context and feature fusion. *Journal of Visual Communication and Image Representation*, 62:166 – 173, 2019. ISSN 1047-3203. doi: https://doi.org/10.1016/j.jvcir.2019.05.003. Cited on page 28.

Y. Zhang, Y. Bai, M. Ding, Y. Li, and B. Ghanem. Weakly-supervised object detection via mining pseudo ground truth bounding-boxes. *Pattern Recognition*, 84:68 – 81, 2018. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2018.07.005. Cited on pages 2 and 28.

H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. Cited on pages xvi, 28, 29, 30, 31, 34, 50, 70, and 72.

L. Zhong, L. Hu, and H. Zhou. Deep learning based multi-temporal crop classification. *Remote Sensing of Environment*, 221:430–443, 2019. ISSN 0034-4257. doi: https://doi.org/10.1016/j.rse.2018.11.032. Cited on page 63.

C. Zhou, G. Yang, D. Liang, X. Yang, and B. Xu. An integrated skeleton extraction and pruning method for spatial recognition of maize seedlings in mgv and uav remote images. *IEEE Transactions on Geoscience and Remote Sensing*, 56(8):4618–4632, 2018. doi: 10.1109/TGRS.2018.2830823. Cited on page 89.