Tomato robotic harvesting in protected horticulture: Machine Learning techniques for fruit detection and classification

Germano Filipe da Silva Moreira

Mestrado em Engenharia Agronómica

Departamento de Geociências, Ambiente e Ordenamento do Território
2021

Orientador

Mário Campos Cunha, Professor Associado, Faculdade de Ciências da Universidade do Porto, INESCTEC

Coorientador

Filipe Neves dos Santos, Investigador Principal, INESCTEC

Todas as correções determinadas pelo júri, e só essas, foram efetuadas.

O Presidente do Júri,

Porto, _____/_____/_____

Agradecimentos

Em primeiro lugar, um especial agradecimento ao meu orientador, Professor Doutor Mário Cunha, por toda a disponibilidade, compreensão e ensinamentos transmitidos, não só ao longo da dissertação como em todo o meu percurso académico. Foi um prazer imenso trabalhar com o Professor e sinto que não podia ter escolhido um melhor "treinador".

Ao Doutor Filipe Santos, do INESC TEC, por me permitir integrar um projeto como o ROBOCARE e a equipa do CRIIS. Por toda a ajuda, conhecimentos e recursos facultados, que fizeram com que estivesse envolvido numa aprendizagem constante, desenvolvendo novas competências e, sobretudo, a ficar apaixonado pela área. As reuniões de sexta-feira nunca serão esquecidas.

Ao Sandro Magalhães, um encarecido agradecimento, por me ter acompanhado em todo este projeto, com uma disponibilidade enorme, sempre pronto a responder a todas as dúvidas, dando conselhos e transmitindo-me inúmeros conhecimentos que sem eles seria impossível desenvolver este trabalho.

Agradeço ao Professor Doutor André Marçal pela disponibilidade e conhecimento transmitido sobre técnicas de análise e processamento de imagem que se tornaram essenciais para o desenvolvimento substancial desta dissertação.

Um reconhecimento à instituição que me acolheu durante cinco anos, a Faculdade de Ciências da Universidade do Porto, por todo o conhecimento e valores adquiridos que também fazem parte e foram fundamentais para o que sou hoje como pessoa.

À minha família, que me apoiou incansavelmente durante o tempo entregue a esta dissertação. Pela paciência, compreensão e incentivo que me prestaram para ultrapassar todos os momentos, especialmente os menos positivos. Não existem palavras suficientes para agradecer. Sem eles nada seria possível.

Um agradecimento especial ao Leandro Rodrigues, o meu parceiro nesta jornada pelo mundo das tecnologias, por toda a disponibilidade em ajudar, pelas palavras de incentivo e troca de conhecimento. Agradeço ainda ao Tiago Padilha, à Tatiana Pinho e ao José Sarmento, pela colaboração e toda a ajuda prestada.

Por último, não menos importante, um agradecimento aos meus amigos. Ao pessoal de Vairão, pela boa disposição e por toda disponibilidade em ajudar. Um agradecimento muito especial à Patrícia, pela sua motivação, força e por estar sempre do meu lado.

Abstract

Society seeks solutions that reduce the labour required in food production and increase farmers' quality of life. However, these solutions must have high levels of autonomy, precision and intelligence. The harvesting operation is a recurring task in the production of any crop, thus making it an excellent candidate for automation. That said, the development of an accurate fruit detection system is a crucial step towards achieving fully automated robotic harvesting. Most of the strategies used in fruit detection are based on Machine Learning (ML) applications. However, these applications are far from maturity, thus presenting challenges to robotic-assisted harvesting, which motivates their study. Deep Learning (DL), an ML approach, and detection frameworks like Single Shot MultiBox Detector (SSD) or YOLO are more robust and accurate alternatives with better response to highly complex scenarios. The present work proposed the creation of a database of annotated images of tomatoes in greenhouses. Two DL models (SSD MobileNet v2 and YOLOv4) were trained and evaluated for tomato detection using the collected images. Subsequently, their ability to classify the fruits in different classes based on their ripeness stage was evaluated by comparing them with a proposed HSV colour space model. In order to extract more information from the fruits, the correlation between fruit colour and soluble solids content (SSC) was also evaluated with the help of the proposed model. Regardind detection, both models obtained promising results, with the YOLOv4 model standing out with an F1-Score of 86.95%. As for classification, the MobileNetv2 model obtained an Macro F1-Score of 87.27%. The HSV colour space model outperformed the YOLOv4 model, obtaining results similar to the SSD MobileNetv2 model, with a Balanced Accuracy of 79.26%. Regarding the SSC, it was concluded that it is not possible to estimate the Brix degree only through colour. This dissertation is part of the activities of the project ROBOCARE, P2020 developed by INESCTEC.

Keywords

Artificial Intelligence, Computer vision, Deep Learning, Single Shot Multibox Detector, YOLO.

Resumo

A sociedade procura soluções que reduzam a mão de obra necessária na produção de alimentos e que aumentem a qualidade de vida dos agricultores. No entanto, estas soluções devem ter elevados níveis de autonomia, precisão e inteligência. A operação de colheita é uma tarefa recorrente na produção de qualquer cultura, tornando-se assim uma excelente candidata para a automatização. Posto isto, o desenvolvimento de um sistema preciso de deteção de frutos é um passo crucial para que se alcance uma colheita robotizada totalmente automatizada. A maior parte das estratégias utilizadas na deteção de frutos estão assentes em aplicações de Machine Learning (ML). Todavia, estas aplicações estão longe da sua maturidade, apresentando assim desafios à colheita assistida por robótica, o que motiva o seu estudo. Deep Learning (DL), uma abordagem de ML, e frameworks de deteção como Single Shot MultiBox Detector (SSD) ou YOLO são alternativas consideradas mais robustas e precisas com melhor resposta a cenários altamente complexos. O presente trabalho propôs a criação de uma base de dados de imagens anotadas de tomate em estufa. Através das imagens recolhidas, foram treinados e avaliados dois modelos DL (SSD MobileNet v2 e YOLOv4) para deteção de tomate sendo, posteriormente, avaliada a sua capacidade de classificar os frutos em diferentes classes com base no seu estado de maturação, comparando-os com um modelo proposto baseado no espaço de cor HSV. De modo a extrair mais informação dos frutos, foi também avaliada a correlação entre a cor do fruto e o teor de sólidos solúveis (SSC), com a ajuda do modelo proposto. No que diz respeito à deteção, ambos os modelos obtiveram resultados promissores, destacando-se o modelo YOLOv4 com um F1-Score de 86.95%. Já na classificação, o modelo SSD MobileNetv2 obteve um Macro F1-Score de 87.27%. O modelo do espaço de cor HSV superou o modelo YOLOv4, obtendo resultados semelhantes ao modelo MobileNetv2, com uma Balanced Accuracy de 79.26%. Em relação ao SSC, concluiu-se que não é possível estimar o grau Brix apenas através da cor. Esta dissertação está inserida nas atividades do projeto ROBOCARE, P2020 desenvolvido pelo INESCTEC.

Palavras-chave

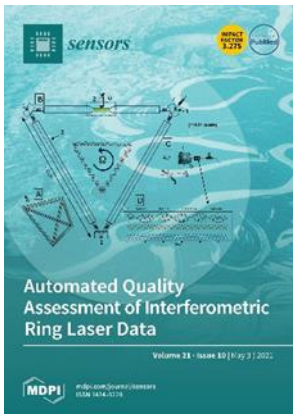
Computer vision, Deep Learning, Inteligência Artificial, Single Shot Multibox Detector, YOLO

Additional Contributions

Parallel to this dissertation, the following contributions were made:

Paper 1

Magalhães SA, Castro L, Moreira G, Dos Santos FN, Cunha M, Dias J, Moreira AP (2021) Evaluating the Single-Shot MultiBox Detector and YOLO Deep Learning Models for the Detection of Tomatoes in a Greenhouse. Sensors 21: 3569



https://doi.org/10.3390/s21103569

Paper available in: 20 May 2021

Classification according to journal: Article

Bibliometric indicators from the Journal Citation Report, Web of Science (JCR – WoS), Clarivate Analytics:

- Journal impact factor (JIF): 3.576 (2020);
- Journal rank: Position 14/64; 1st quartile (Q1).

Paper 2

Padilha TC, Moreira G, Magalhães SA, Dos Santos FN, Cunha M, Oliveira M (2021) Tomato Detection Using Deep Learning for Robotics Application. In. Springer International Publishing, pp 27-38

https://doi.org/10.1007/978-3-030-86230-5_3

Paper available in: 03 September 2021

Classification according to journal: Conference paper

Bibliometric indicators from the Journal Citation Report, Web of Science (JCR – WoS), Clarivate Analytics:

- Journal impact factor (JIF): 0.302 (2005);
- Journal rank: 70/79; 4th quartile (Q4).

Dataset 1

Magalhães, Sandro Augusto, Moreira, Germano, dos Santos, Filipe Neves, & Cunha, Mário. (2021). AgRobTomato Dataset: Greenhouse tomatoes with different ripeness stages [Data set]. Zenodo.

https://doi.org/10.5281/zenodo.5596799

Publication date: 25 October 2021;

License: Creative Commons Attribution 4.0 International.

Dataset 2

Moreira, Germano, Magalhães, Sandro Augusto, Padilha, Tiago, dos Santos, Filipe Neves, & Cunha, Mário. (2021). RpiTomato Dataset: Greenhouse tomatoes with different ripeness stages [Data set]. Zenodo.

https://doi.org/10.5281/zenodo.5596363

Publication date: 25 October 2021;

License: Creative Commons Attribution 4.0 International.

Contents

List of Tables

List of Figures

Acronyms

AI – Artificial Intelligence

ML – Machine Learning

ANN – Artificial Neural Networks

DL – Deep Learning

CNN – Convolutional Neural Networks

SSD – Single Shot Multibox Detector

YOLO – You Only Look Once

ROBOCARE – Intelligent Precision Robotic Platforms for Protected Crops

HSV – Hue, Saturation and Value

HRS – Human-Robot Systems

ARS – Autonomous Robot Systems

DOF – Degrees Of Freedom

RGB – Red, Green and Blue

RoI – Region of Interest

VGG – Visual Geometry Group

RPN – Regional Proposal Network

YIQ – Quadrature-phase

HSI – Hue, Saturation and Intensity

FCM – Fuzzy C-Means

HOG – Histograms of Oriented Gradients

SVM – Support Vector Machine

FCR – False Color Removal

NMS – Non-Maximum Suppression

RVM – Relevance Vector Machine

FPN – Feature Pyramid Network

COCO – Common Objects in Context

OID – Open Image Dataset

PA – Precision Agriculture

TPU – Tensor Processing Unit

GPU – Graphics Processing Unit

IoU – Intersection over Union

TP – True Positives

FN – False Negatives

FP – False Positives

TN – True Negatives

AUC – Area Under the Curve

AP – Average Precision

mAP – Mean Average Precision

SSC – Soluble Solids Content

1. Introduction

Labour is the major cost factor in agriculture, accounting for up to 40% of operational costs in most production systems [1, 2]. The various agricultural tasks require an immense labour force that its necessity ultimately creates bottlenecks, leading to lower productivity and incomes, thus increasing costs. Problems such as ageing or shortage of workers contribute to labour scarcity [3], plus most agricultural activities are unattractive and exclusive, often associated with the news of social discrimination and illegal labour flows. Therefore, the execution of manual labour has given rise to concerns in terms of farm planning and competitiveness.

Finding new solutions is vital, allowing farmers to produce with quality, higher yields and lower costs. One of the solutions involves robotization and automation. Robotic systems can operate in hazardous and challenging farming environments, completing tasks that are often strenuous and physically demanding [4]. Agricultural robots are far from mature, despite all the research and progress in robot technology, as manual work prevails. Production inefficiency and lack of economic payback are the main setbacks of robotics introduction in agriculture [2]. An agricultural robot must face several challenges, presenting difficulties when confronted and implemented in a very dynamic and highly unstructured environment such as the farm environment, characterised by soil variations, different luminosity and visibility caused by quick spatial-temporal changes [4, 5]. The development of advanced technology is essential to achieve high levels of autonomy, precision and intelligence, capable of suppressing the limited performance of today's robots and bringing it up to the standards of manual labour in the near future.

In the greenhouse horticulture sector, where labour accounts for up to 50% of the usual costs [6], the penetration of robots is not yet comparable to the robots developed for open-field farming systems [7]. One of the most important horticultural crops is the tomato. It is the second most harvested vegetable in the world. Still, manual tomato harvesting is associated with low labour productivity, as the harvesting operation absorbed a large part of the labour costs. Thus, since this operation is a recurrent task in the production of any crop, it becomes an excellent candidate for automation [8]. However, automating an operation such as harvesting is not a simple matter, as the robot must be able to detect and manipulate the fruit in an environment that is full of objects of various colours, shapes, sizes, textures and reflective properties, highly unstructured

scenarios with a large degree of uncertainty, constantly changing lighting and shadow conditions as well as severe occlusions [9]. Therefore, the performance of harvesting robots is still limited. There is a need for a clearer understanding of the advances made in harvest robotisation, and a better understanding of this limited performance [8]. State-of-the-art identifies the system's visual perception as one of the leading causes of the poor performance of harvesting robots. Therefore, a highly effective detection system is necessary and crucial to push forward their development [9].

Robotics agricultural applications present a close relationship with image processing and artificial vision techniques, promoting the joint development of these fields. Computer vision and Artificial Intelligence (AI) have attracted growing interest from the agricultural world to optimise several agricultural operations through accurate, robust and automated solutions [10]. Computer vision comprehends methods and techniques that allow the development of systems endowed with artificial vision. These systems involve an image acquisition phase, through cameras or sensors, which will be later processed and analysed. Image analysis refers to the methods used to differentiate a region to be detected, in this case, the fruit from the images acquired [11, 12]. Visual features are used to differentiate this region, mainly colour but also size, shape,or even spectral reflectance. Thresholding the visual features is the most elementary method, but it is less robust, as the high variance of the environment affects the performance of this type of method [9].

Alternatively, Machine Learning (ML) has been increasingly used in fruit detection and classification. ML is an AI discipline, which enables machines to learn for themselves. ML algorithms learn and acquire knowledge through the data they analyse, creating a model capable of predicting or making intelligent decisions [13]. One of the most promising algorithms for its accuracy in complex scenarios is Artificial Neural Networks (ANN). Just like the human brain, these algorithms gather information, processes it and generate an output. They are mostly used in Supervised Learning, meaning that the inputs and outputs are known. The algorithm creates an input-output relationship to generalise and predict results from inputs never seen before [14]. The objects to be detected need to be annotated previously and the models are trained to recognise features from those objects. Then, the model is used to detect the trained objects on new images.

Another strategy is Deep Learning (DL), which is based on ML. It is a modern, more robust and accurate approach with better response to complex scenarios since it has

strong learning capabilities [15]. It is similar to the ANN model, however, it is a "deeper" neural network that provides a hierarchical representation of the data through multiple convolutions [16]. The four main DL models are Unsupervised Pretrained Networks, Convolutional Neural Networks (CNN), Recurrent Neural Networks and Recursive Neural Networks. Since it is the most used in image analysis, the CNN model has been increasingly applied in agriculture and fruit detection [17]. Some detection frameworks have been developed using CNN and have achieved promising results. The most notable are the SSD (Single Shot Multibox Detector) [18] and the YOLO (You Only Look Once) models [19]. These frameworks are composed of a backbone, which is a CNN responsible for extracting the relevant features from the input image, and several convolution filters that detect/classify the objects and estimate their size with a bounding box. The SSD and YOLO models are called one-stage detectors as they are capable of feature extraction and object detection in a single step. This process consumes less time and can therefore be used in real-time applications, such as harvesting robots. Unlike two-stage detectors, these models are generally faster and structurally simpler [20].

However, even with all these advances, robotic application in crop harvesting still presents several challenges. Despite increasing, research on fruit detection using models such as SSD or YOLO is still limited [21]. Better vision systems for all the different operations in the agricultural environment must be developed in parallel with faster and more accurate image processing/algorithm methods [7], demanding research in line with the objectives and the topic proposed by the presented dissertation project.

This study is framed within the activities of the ROBOCARE[1] (Intelligent Precision Robotic Platforms for Protected Crops) project, P2020 developed by INESC TEC and whose research team integrates the orientation of this dissertation. The ROBOCARE project aims to research and develop intelligent precision robotic platforms for protected crops, to decrease the reduction of labour burden and increase the ergonomics of the agricultural operations and the consequent increase in labour productivity and economic profitability of crops. The team leading the project is working on the development of a greenhouse tomato harvesting robot.

This dissertation is focused on the computer vision of the robot. The main objective is to: i) create a dataset of annotated tomato images to train and evaluate two DL models

[1] INESC TEC – ROBOCARE Project (https://www.inesctec.pt/pt/projetos/robocare). Last accessed: 4 November 2021

(SSD MobileNet v2 and YOLOv4) for tomato detection and ii) compare whether the classification of tomatoes into different classes, based on their ripeness, can be done effectively through those same DL models or a proposed model based on HSV (Hue, Saturation and Value) colour space. Furthermore, automatic phenotyping of relevant features for harvesting decisions, such as Brix degree, based on its correlation with fruit colour is sought to be performed with the help of the model mentioned above. In all experiments, tomatoes of the "Plum" variety were used. A meta-analysis also seeks to better understand the landscape of harvesting robots developed specifically for protected horticulture to complement this work.

The lack of access to robust and accurate fruit detection systems has limited the automation of harvesting and the commercialisation of robots for this purpose. It is hoped that this study will help take that final step towards the automation of harvesting and many other tasks in the agricultural sector.

2. Literature Review

2.1 Agricultural Operations

Labour is the main factor contributing to operational costs in agriculture [1]. Despite some differences and great variability in absolute magnitudes, labour generally accounts for about 40% of operational costs in most production systems [2]. The high labour demand for the execution of several agricultural tasks causes bottlenecks within the farm organization with associated efficiency costs, especially in frequent situations of unavailability of labour. Competition for labour between sectors and the ageing or scarcity of workers contribute to labour shortages [3]. In the agricultural industry, the problem is aggravated by the hazardous nature of most farming operations, which makes them unattractive and exclusive, often associated with social discrimination and illegal labour flows. Cost reduction is thus hindered by the vital need for labour power [22].

Horticulture is characterised by a wide range of production systems and methods and a great diversity of herbaceous and woody species for food and/or ornamental production. Within these systems, protected or greenhouse horticulture is one of the most intensive in production inputs and knowledge, focusing on the production of crops with high added value. Its role in regular food production is fundamental, as it is a system that enables the control of environmental factors (temperature, light, etc.), greater efficiency in the use of resources (water, fertilisers, etc.) and the use of high-tech systems leading to higher yields, in a stable and better quality production [7].

Protected horticulture sector has been growing over the years, and in 2019 its world market value was estimated at around 25 million euros, with projections indicating its annual increase of around 9% over the next 5 years [23].

In protected crops, labour represents approximately 50% of operational costs and is considered critical for developing and maintaining this farming system [6]. However, the scarcity and associated costs reduce its economic efficiency, making it difficult to plan operations. This context demands the adoption of new technologies and the search for solutions that improve cost reduction or compensate for the lack of labour to guarantee the success of the most varied production systems.

2.2 Robotization in Agriculture

2.2.1 Opportunities, Limitations and Conditions

The agricultural sector has followed a trend towards valorising production inputs, namely labour and knowledge, to the detriment of the added value of primary production activities, promoting the emergence of technological value chains (generally longer) densified in knowledge aligned with the concept of digitalisation of the sector ("Agriculture digitisation" or "Smart farming"). This orientation towards a more technological side of mechanization and automation systems is due to the revolution that has occurred in recent decades concerning the various technological fronts (computing, sensors, navigation, etc.) [4] (Fig. 1). These advances are mainly related to the necessity to minimise operational and production costs, reduce environmental impacts and optimise production cycles [7].

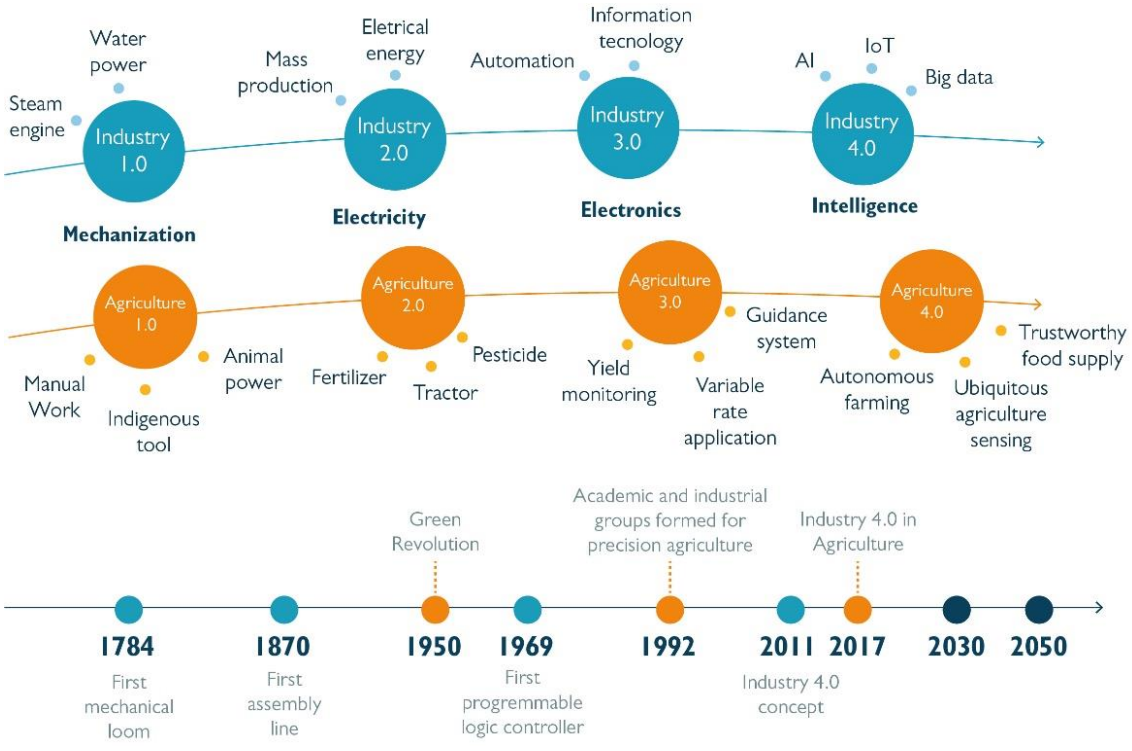


Figure 1 *| Temporal evolution of industry and agriculture with an indication of the common factors of change and the technological contexts that characterize the time frames. Adapted from: Liu, Ma [56].*

The introduction of robotic technology in agriculture could alter productivity, ergonomics and labour hardship. The robots can overcome critical human constraints, such as operating in hazardous and challenging farming environments, ultimately reducing the impact of physically demanding, mundane and repetitive arduous tasks [4]. These new technologies provide high potential for increasing agricultural productivity, which in turn supports the growth and development of the economy in a more sustainable way [24, 25].

Yet, implementing robotic solutions in the agricultural sector is by no means an easy task. The technical viability of agricultural robots for various tasks has been widely validated, however, despite all the research done in the last three decades, very few have commercial applications [26]. The robotic world can be divided into four groups based on the structural characteristics of the environments and objects in which they operate [2]:

1. Environment and objects are structured;
2. Environment is unstructured and objects are structured;
3. Environment is structured and objects are unstructured;
4. Environment and objects are unstructured.

Robots applied in the agricultural branch are associated with the fourth group (Fig. 2), where nothing is structured, making the development of these robotic alternatives challenging [2].

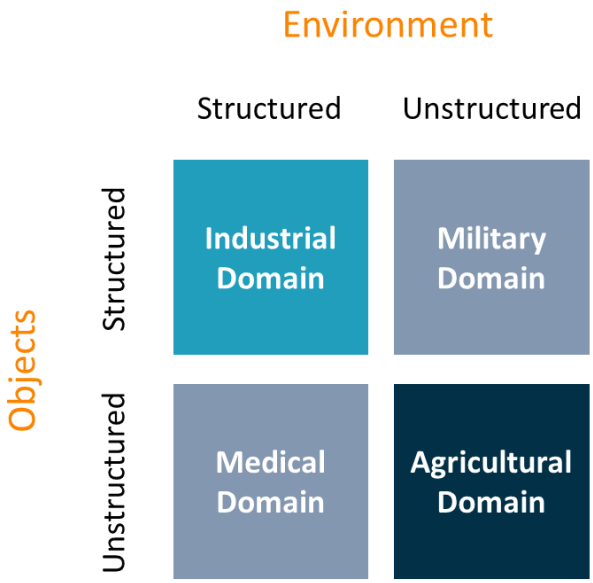


Figure 2 | *The four robotic groups based on the structural characteristics of environments and objects. Adapted from: Bechar and Vigneault [2].*

Unlike industrial applications, which deal with relatively simple, repetitive, well-defined and pre-determined tasks in stable and replicable environments (structured environment), agricultural applications for automation and robotics require advanced technologies to deal with complex and highly diverse environments [27, 28]. Most agricultural operations take place in unstructured environments characterised by quick spatial-temporal changes. Also, factors such as land (slope, shape, obstacles, among others), visibility, lighting and other weather conditions are poorly defined, vary continuously and have inherent uncertainties, creating unpredictable and dynamic situations that the robot will have to manage [5]. In addition, the sector deals with products that are highly sensitive to environmental and physical conditions [29], requiring careful and precise handling to preserve as much as possible the quality of these products along the chain up to the consumer.

That being said, the challenges imposed by the instability of the environment and its objects, require complex technological solutions and, generally, with high specificity which limits its transferability between production systems. Thus, these solutions are not always efficient (especially if compared to industrial solutions) and are generally expensive, which is particularly relevant given the low added value of most agricultural

products. In this context, it can be considered that lower operational efficiency and high costs are the main limitations to the robotic application in agriculture [2].

The implementation of robotics technology in agriculture is feasible if at least one of the following conditions is met [2]:

- The usage cost is lower than the cost of any current method;
- Allows increasing productivity, making the production system more profitable and resilient against competitive market conditions;
- Improves production quality and uniformity;
- Minimises uncertainty and variation in the different production processes;
- Allows the farmer to make decisions and act with higher spatio-temporal resolution;
- Can perform specific tasks defined as being dangerous or that cannot be performed manually;
- It emerges as a response to scenarios with no alternative, such as labour shortages.

The diversity of agricultural operations highlights the usefulness of decomposing the tasks performed by human labour. This decomposition will identify critical factors where there is potential for strong substitution or complementarity and, in turn, identify areas where the introduction of new technologies will have the greatest impact [4]. Therefore, as in other industries and sectors, potentially automatable agricultural tasks can be categorised into four types (Fig. 3), based on their manual or cognitive nature and the execution of standardised and non-standardised tasks.

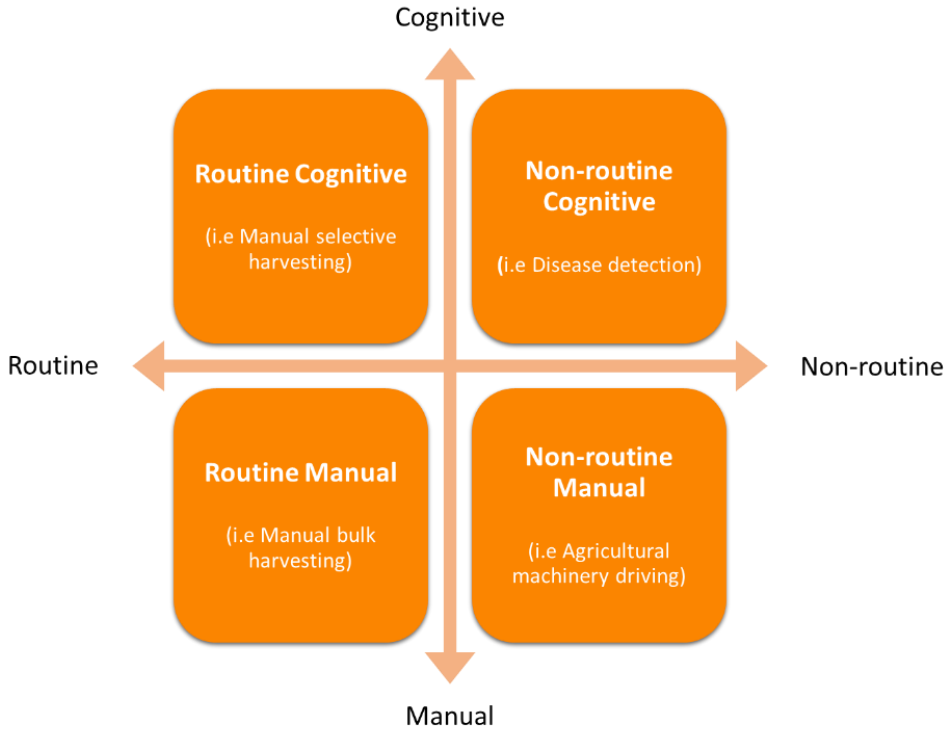


Figure 3 | *Categorization of tasks in terms of cognitive-manual and nonroutine-routine levels. Adapted from: Marinoudi, Sørensen [4].*

2.2.2 Concepts and Required Abilities of Robotic Systems

Robotic solutions can be divided into two concepts [2]:

- Human-Robot Systems (HRS);
- Autonomous Robot Systems (ARS).

As the name implies, HRS are driven by the synergy between human and robotic valences. Incorporating a human operator to interact, rather than just supervise the system, is an advantage as human capabilities in perception, thought and action are unmatched in abnormal and unpredictable environments. [30]. By taking advantage of human sensing capabilities and the accuracy and consistency of a robotic system, HRS become more streamlined solutions, resulting in better performance and reduced costs [2, 31].

On the other hand, ARS are developed to perform tasks, make decisions and act in real time without human dependence. This type of systems are requested in sectors that demand reductions in manpower and workload, being adequate in exigent scenarios with high precision and performance, under stable conditions. However, there is growing research focused on ARS for unstructured environments [2].

Agricultural ARS are composed of numerous subsystems and devices that allow their operation and the execution of various tasks with different degrees of autonomy. They must have the ability to manage unforeseen events, with a certain level of autonomy, where these subsystems and devices deal with various aspects such as: guidance and trajectory planning, mobility and navigation, detection and localization, manipulators and end effectors [32].

Typically, agricultural robots are designed to perform a specific agricultural operation, such as seeding, weed control, pruning, harvesting, among others. For this operation to be executed, ARS needs to perform several supporting tasks that compose it, such as location and navigation, object detection, treatments or actions to be performed, etc. Information and commands are transferred between the various tasks and between the tasks and the main operation. Each task controls one or more subsystems and devices, and one subsystem or device may serve several tasks [2] (Fig. 4).

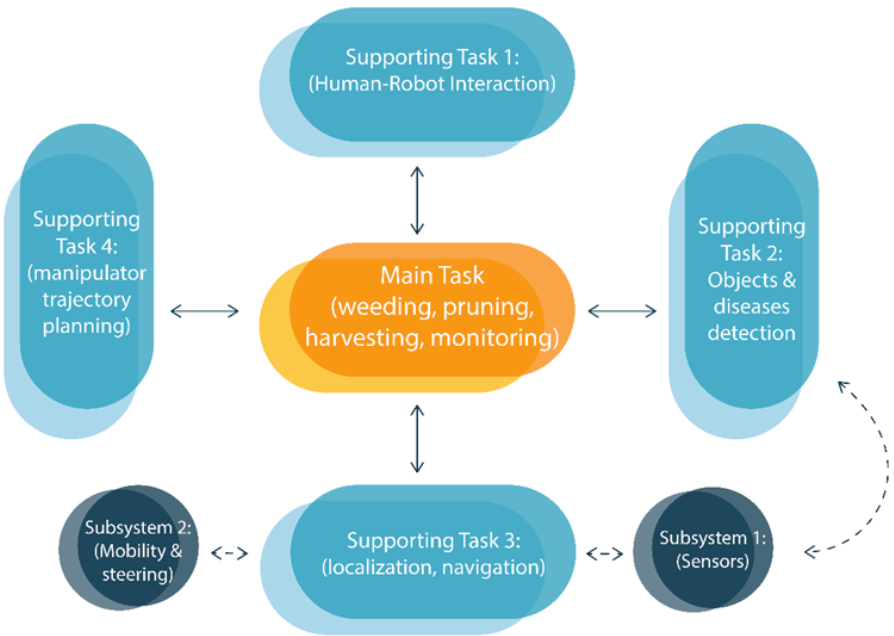


Figure 4 | *Example of the structure of a main task and its support tasks and subsystems for an agricultural robot. Solid arrows represent commands, data and information; dashed arrows represent conceptual connections. Adapted from: Bechar and Vigneault [2].*

In order to achieve an adequate degree of autonomy, the ability to perceive and automate are basic system requirements. Therefore, an ARS should have a high degree of flexibility so that it can be incorporated into constantly changing scenarios as well as the ability to process the information it receives from its sensors. When designing an ARS, two major challenges are quite often imposed. The first is the requirements of non-linear and real-time response underlying the sensor-motor control formulation. The second is how to model and use the approach that a human being would use to solve the problems he faces [33].

These autonomous systems are highly complex as they are made up of several different subsystems, that need to be integrated and correctly synchronised to perform tasks seamlessly and successfully transfer the necessary information [2].

Generally speaking, whether dealing with a HRS or ARS concept, a robot operating in the agricultural environment must have several capabilities [4]:

- The robot should be configurable for the environment in which it operates, for different layouts of the land it moves (i.e. size and shape), soil types, crop parameters (variety, size, maturity), production conditions and systems (open field, greenhouse, with or without soil, etc.) and be adaptable to different crops, in case the farm produces more than one crop or practices crop rotation;

- As far as safety is concerned, it should ensure safe mobility in a dynamic, partially known or completely unknown environment. Furthermore, it must be able to protect the environment from some degradations such as soil compaction;

- For robots intended to manipulate crops, their handling capabilities should adjust to the sensitivity of the products in question. Their sensing capabilities should adjust to the variability of the product, in terms of colour, size, shape, etc.

The development of these robots for integration in agricultural processes should consider: i) Technology and intelligent systems should be developed to overcome the difficulties imposed by the unstructured and complex environments that the sector presents, ii) economic aspects specific to production systems should be addressed in order to understand the effective viability of the various types of robots, iii) safety and reliability are one of the most critical aspects - safeguarding workers, the environment, crop damage (quantity and quality) and machinery is mandatory [2].

Inevitably these systems will become intelligent enough to achieve high levels of autonomy in the near future [34]. However, it is necessary to determine how smart they have to be and define their appropriate behaviour. The increasing labour costs and the demand for differentiated and high quality products as mentioned above, on the one hand, and the decreasing cost of computers, electronics, and progressively more efficient sensors will promote the economic viability of agricultural robots [2].

Since this dissertation is related to the robotic harvesting of tomatoes, all these components and their synergies will be addressed in the following chapters. Special attention will be given to the harvesting operation, in a greenhouse environment, as the main operation and the fruit detection/classification, in this case of tomatoes, as the support task along with all the subsystems and devices needed to perform them.

2.2.3 Operations, Crops and Environments of Robotic Applications

Agricultural robots have been researched and developed for many operations performed throughout the production cycle, from seeding to harvest. However, the most labour-intensive tasks, such as harvesting, have attracted a greater focus from this type of technology (Fig. 5) [7].

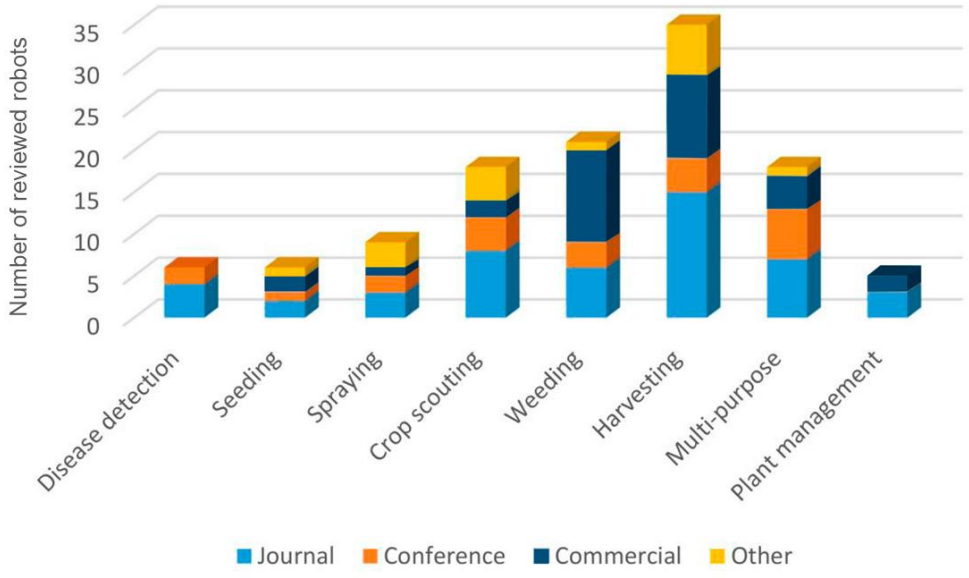


Figure 5 | *Number of reviewed robots per agricultural operation. Adapted from: Fountas, Mylonas [7].*

Strawberries, peppers, grapes and tomatoes have been the most widely promoted crops within the research (Fig. 6) [7].

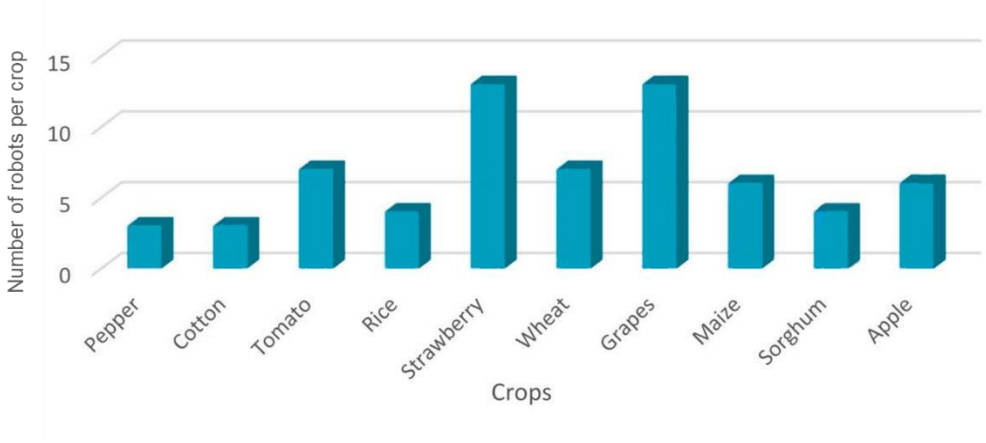


Figure 6 | *Main crops in correlation with the number of robotic systems. Adapted from: Fountas, Mylonas [7].*

As mentioned, robotisation and implementation of automatisms are hampered by the unstructured environment that agricultural systems present. In this context, one would expect that robots operating in semi-structured environments, such as greenhouses, would be in the first line of development.

However, this is not observed, as almost half of the developed robots are allocated to open-field production systems (Fig. 7). This can be explained by the fact that most of the crops around the world are grown in open-field systems and some operations are more associated with this type of production, such as weed control, which is also one of the most studied operations [7].

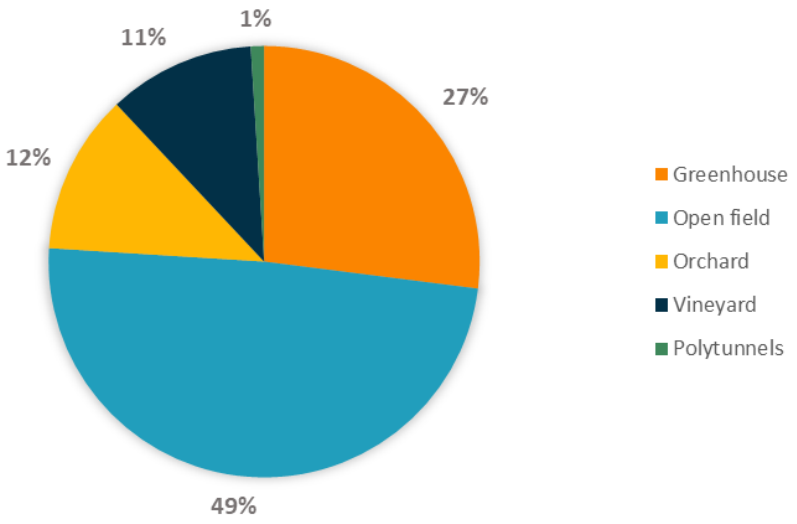


Figure 7 *| Allocation of robots (%) in various agricultural production systems. Adapted from: Fountas, Mylonas [7].*

2.2.4 Tomato Robotic Harvesting in Protected Horticulture

The harvesting operation becomes an excellent candidate for automation due to being recurrent and crucial in the production of high-value crops [8]. Increasing efficiency and reducing labour dependency in this operation could ensure higher yields and competitiveness in high-tech food production, so the development of harvesting robots should be considered as a viable alternative [35].

Since the 1980's this subject has been researched, with Japan, The Netherlands and the USA being the pioneer countries that have made the greatest contribution to its development. However, despite all these advances, robotic harvesting is still far from maturity (Fig. 8), and every year millions of tons of fruit and vegetables are harvested manually in greenhouses. The scarce use of robots can be attributed to their low performance, so it is essential to understand why this limited performance and challenges that can generate a positive trend [8, 35, 36].

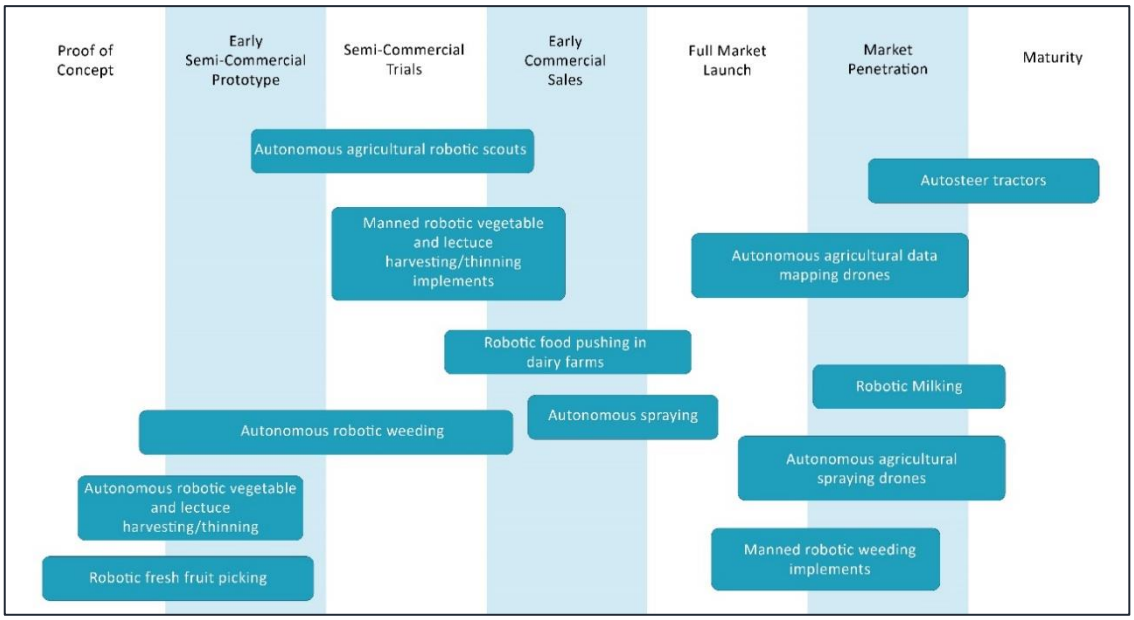


Figure 8 | *Robotic systems: market and technology readiness by agricultural activity. Adapted from: Michael Dent, IDTechEx.*

In protected horticulture, few crops are as important as tomatoes. It is the second most harvested vegetable worldwide and one of the crops with the highest economic value. Between 2003 and 2017, world tomato production increased annually from 124 million tonnes to more than 177 million tonnes, and over the last 15 years, consumption has experienced sustained growth of around 2.5% [37]. This is one of the leaders when it comes to protected horticulture. In the south-east of Spain, Almería, home to the world's largest concentration of greenhouses (over 30,000 hectares), tomatoes are the main crop, accounting for 37.7% of all production [38].

Manual tomato harvesting is associated with low labour productivity because it is sporadic, fatiguing, with high to moderate physical effort and high repeatability by the operator, requiring about 700-900 h/year/ha [39], generating a low labour force attractiveness. Along with the scarcity of labour, the precarious working conditions and increased labour costs constrain the greenhouse harvesting operation. The importance of this crop and the associated high production costs justify, as mentioned above, the fact that this is one of the most common crops in the development of robotic harvesting.

However, robotising tomato picking is not an easy task. The robot must detect and manipulate, in a heterogeneous and unpredictable environment, a fruit that also varies in terms of position, size, shape, colour and even reflectance [8, 9].

The colour is used as an indicator of ripeness and the desired level of ripeness by the producer can vary. As a climacteric fruit, tomatoes can be harvested at the physiological maturity stage (green colour), ripening detached from the plant, or at a more advanced stage, showing a reddish colour (Fig. 9). [40]. The harvest moment can be dictated according to market requirements. If the consumption is local, for proximity markets, the fruit can be harvested later. However, if the fruit needs to be transported over long distances, harvesting in an immature stage would be more appropriate. Therefore, the robot must handle these colour variations in order to achieve a segmented harvest.

Figure 9 | *Different colours that a tomato can present throughout its development.*

Other important aspects of handling the fruit that should be automatically detected are size, shape, and morphological inconformities. Careful handling is crucial, especially with tomatoes because of their poor surface resistance and slippery surfaces. The susceptibility to damage is also a relevant factor. Thus, the development of an end-effector that can handle variations in fruit size and shape and that takes into account the growing environment and the physical properties of the tomato is essential for the prevention of damage during the harvesting phase [41].

Accessibility and visibility of the fruit are two major challenges in the harvesting task [8]. Figure 10 illustrates different lighting conditions that the robot may encounter and scenarios where many fruits are occluded by different parts of the plant which, as mentioned before, end up becoming obstacles that prevent not only their access but also

their visibility. The robotic system must detect less visible fruits and harvest them without damaging other fruits and plant parts.

Figure 10 | *The environment a harvesting robot might encounter in a greenhouse. Fruits with high colour correlation with the background, overlaps and occlusions by different plant structures and different light conditions.*

The age of the plants, pests and diseases and different production methods are other aspects that might also play a role in the variation of the fruits to be detected and harvested. The variability and factors described above are already valid for a single cultivar, but as there are many cultivars of tomato, the variation is even more pronounced. A cultivar has slight genetic differences, which leads to a modification of the fruit in terms of position, shape, size and colour [8].

Despite the difficulties imposed by the set of factors described, there are already some prototypes developed for robotic tomato harvesting. In order to be able to evaluate them, the literature mentions two main performance metrics: speed and harvesting accuracy rate [7].

Instead of developing the entire robotic system, Li, Liu [42] designed an actuating organ by installing it in an industrial manipulator already on the market. The robot is composed of a Motoman-sv3x manipulator, an actuating organ with 3-DOF (Degrees Of Freedom), a camera, a computer, a PMAC controller and a micro servomotor. They analysed the workspace and kinematics of the system and concluded that the harvesting robot meets the operational requirements of a greenhouse, performing the task of separating and harvesting the fruit in only 3 seconds. In order to harvest truss tomatoes,

Ji, Zhang [43] proposed a robotic system, two technologies for the detection of the abscission point and an end effector. For the detection of the tomato and a reference point, an algorithm using a segmentation feature based on the RGB (Red, Green and Blue) colour was used. The approximate fit curve of the stem and the contour of the reference point were extracted to generate the optimal abscission point. Based on a flexible transmission, the end effector allows the robot to perform the harvesting task in 37.2 seconds with a success rate of 88.6%.

Zhao, Gong [44], opted for the HRS concept, to overcome the complexity imposed by the greenhouse environment and developed a modular robotic system of two manipulators with 3-DOF. The tomato detection is made through artificial recognition in which the operator, through an interface, selects the fruit. Two different end effectors were designed and tested, one for each manipulator, but no results were reported. Yasukawa, Li [45], authors of the Tomato-Harvesting Robot Competition, now in its 6th edition [46], designed a robot that moves on rails and is composed of a Kinect v.2 sensor, a USB camera, a computer, a six-axis serially linked manipulator and an end effector. Fruits are detected using infrared images and spectral reflection, with an 88.1% accuracy rate. Targeted for cherry tomatoes, Taqi, Al-Langawi [47] developed a robot that includes a Pixy camera connected to an Arduino Uno microcontroller, an infrared reflection sensor and a Cartesian robotic arm, that operates in X-Y space. The system can detect riped and rotten fruit at an accuracy rate of 100%, which only results from the low harvesting speed of only 2 fruits per minute. Wang, Zhao [48], created a robot composed of an independent four-wheel steering system, a robotic arm with 5-DOF, a navigation system and a stereo binocular vision system. Fruit detection is performed using the Otsu algorithm [49] and the elliptical model method, and the detection and picking is completed in 15 seconds with a success rate of about 86%.

All the projects mentioned point towards the same future goal: to improve the robot's performance. Harvesting a greater number of fruits, in less time, while maintaining high precision becomes imperative. In most cases, the cause of failure is associated with the visual perception of the system, where problems such as light intensity, overlapping and occlusion of the fruits to be detected, due to the different parts of the plant, hinder and end up further delaying the intended goal. Therefore, fruit detection and classification is a critical area capable of dictating the success or failure of robotic systems.

2.3 Computer Vision of Harvesting Robots

2.3.1 Image-based Fruit Detection and Classification

The success of an agricultural robot is directly related to its ability to process information and, in particular, to analyse and interpret visual inputs. For all the difficulties, previously repeated emphatically, that the agricultural environment imposes, the development of an accurate fruit detection system is then a crucial step towards achieving a fully automated robotic harvest, where the main objectives are to [9]:

- detect the presence of individual fruits;
- find them in space;
- discriminate them from their surroundings.

Machine vision has attracted growing interest and is often used to provide accurate, efficient and automated solutions to tasks traditionally performed manually [10]. Their use improves the functionality, intelligence and remote interactivity of harvesting robots. However, it still presents technical difficulties, preventing most robots from reaching commercial use [50]. Even if this failure cannot be exclusively attributed to computer vision, it is undeniable that its success is crucial to achieve high levels of detection, which are mandatory for a robot to be efficient and profitable [9].

Computer vision comprises methods and techniques that allow developing systems endowed with artificial vision, feasibly implementing them in practical applications. These systems can be broken down into three phases [11]:

- image acquisition;
- image processing;
- image analysis.

Systems based on computer vision acquire sensory data through equipment, such as sensors or cameras, in a process defined by transferring electronic signals into a numerical representation [9, 12]. Usually, one or more cameras (monocular or binocular

vision) are used, to which sensors capable of measuring depth or other parameters such as the spectral behaviour of objects (LiDAR or RGB-D cameras) can be attached.

On the other hand, processing encompasses all the tasks that allow the acquired images to be digitally manipulated. Manipulation may refer to more simplistic operations, such as greyscale adjustment, focus corrections, contrast or sharpness improvements and noise reduction. They are used to improve the quality of an image or modify the position of the object of interest through geometric transformations [51] or, at higher levels of processing, to segmentation techniques (partition of images into regions) of the objects present in the images.

Image segmentation is a crucial part when aiming to perform agricultural tasks in an automated way. It results in a set of contours or regions of interest (RoI), which, with the proper extraction of their attributes, can be evaluated for their characteristics [52]. In the scope of segmentation, numerous approaches are proposed that search for several features of the object to be detected, from the most elementary ones, such as colour, size, shape or texture, to the most complex ones, such as spectral reflectance or thermal response, in an attempt to evaluate the relationship between a given set of pixels and those features. Still, most of these approaches involve using confidence thresholds for the feature values, which need to be defined for each image, making the performance dependent on these thresholds. The detection accuracy is therefore highly impacted by the performance of the image segmentation [10].

Finally, image analysis, is related to the recognition and classification of RoI, which are usually performed through the thresholding of visual features, statistical classifiers, or neural networks [11], reviewed in the following chapters.

Several reviews have been elaborated in recent years highlighting the various computer vision techniques and models, specifically for harvesting robots [9, 50, 53]. These systems can present great variability. Different types of sensors can acquire the images and there are many algorithms, models, and features that can be used to process, segment, and classify the fruits to be detected. Thus, the Appendix A serves as a synthesis of these techniques and computational approaches.

2.3.1.1 Machine Learning

With the development of new technologies, the amount of agricultural data has increased dramatically [54]. Thus, in the current era of smart agriculture, more conventional methods fail to ensure the extraction of more valuable information [55]. Therefore, the use of Machine Learning (ML) in agriculture is growing, with several fields of action [56].

ML is a discipline of Artificial Intelligence (AI) that provides computers with the ability to learn without being explicitly programmed [57]. Understands methods and techniques for computer applications, endowing them with the ability to adapt and modify their actions to make them more precise [11]. Optimising a given task is the goal of ML algorithms, which are based on analysing examples and past events. A bit like a human being, who performs a task better and better as he or she gains more experience, the more data used, the better the ML models will be [58]. Algorithms learn and acquire knowledge through the data they analyse, creating a model capable of predicting or making intelligent decisions.

The type of the models' learning is fundamental and defines the different existing ML algorithms. In general, they can be divided into [11, 13, 58, 59] (Fig. 11):

- Supervised learning – is applied to previously annotated data, the inputs and outputs are known, that is, to each input corresponds an output. The algorithm tries to create an input-output relationship based on the annotated dataset, so that it can then generalise and predict outputs from inputs never seen before;

- Unsupervised learning – does not deal with annotated data, which leads to algorithm learning by itself, making it more difficult to implement. This learning is done by comparing inputs to find similarities, not to classify, but mainly to organize or find a structure in the data;

- Reinforcement learning – works through reward and punishment, that is, while the algorithm makes its decisions it is not given any feedback. Only at the end, when it reaches the correct answer it is given positive feedback. Using this process reinforces and consolidates the previous decisions that led it to the correct answer. It is about exploring different answers until the correct ones are found.

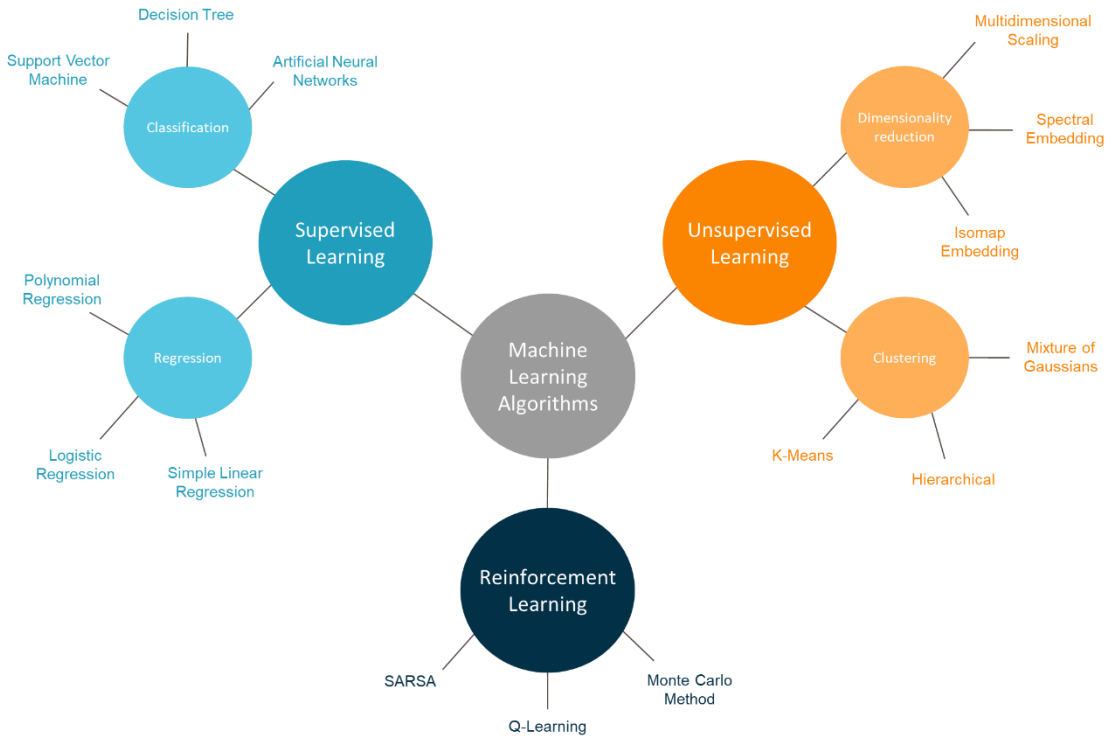


Figure 11 | *Different types of Machine Learning algorithms and their categorization according to their learning mode.*

Currently, there are numerous resources and innovative algorithms associated with ML that enable the resolution of various problems. One of the most promising, due to its accuracy in complex scenarios that require the analysis of a lot of data, is the Artificial Neural Networks (ANN). This is an algorithm mainly used in Supervised Learning, as a classification algorithm, inspired in the functioning of the human neuron, simulating the electrical activity of the brain and nervous system. In the human brain, dendrites are the network that transfers electrical signals to the cell body, which in turn adds and gathers those signals that axons will later transfer to other neurons (Fig. 12 a). ANN are composed of processing elements (neurons) arranged in layers or vectors, with the output of one layer serving as input for the next layer. The structure of an ANN can be divided into 3 main layers [14]:

- Input layer – collects information from the outside world;
- Hidden layer – layer where the neurons that will process the information contained in the inputs are located;
- Output layer – transfers the information from the network to the outside world.

A neuron can be connected to all or a subset of neurons in the next layer, through adaptive weights, in a process similar to synaptic connections in the brain. The knowledge that the algorithm acquires is stored as a set of connection weights, which determine the strength and sign of that connection. These weights can be modified, and this modification allows ANN algorithms to learn [11, 60, 61].

Information processing begins at the input layer, and these inputs are weighed and grouped into the processing neurons via a scalar function vector, such as summation, to produce a single input value. Once the input value is calculated, the neuron uses an activation function to generate the output (Fig. 12 b).

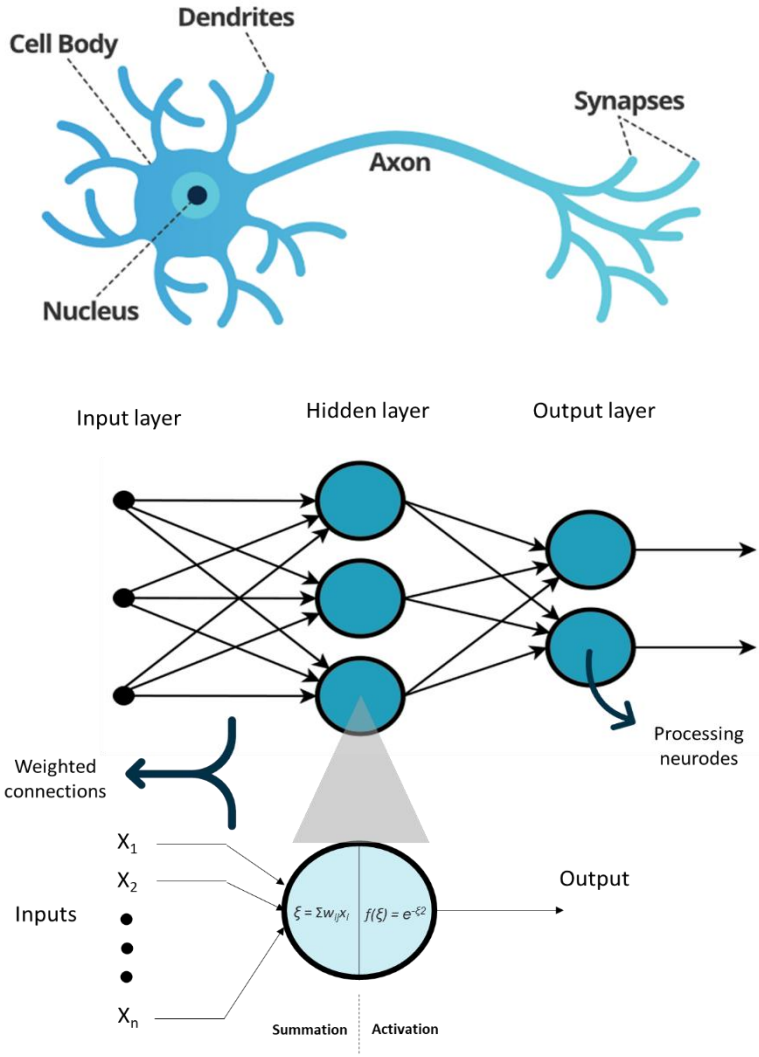


Figure 12 *| Similarity between a human neuron (a) and an ANN (b). Both are composed by processing elements (neurons) and connections between them (weights).*

2.3.1.2 Deep Learning

Deep Learning (DL) [62] is one of the ML-based methods most used nowadays. It is a modern technique for image processing, being very successful in several areas [63, 64], having more recently entered the agricultural domain. The success of DL models is based on the fact that they have high levels of abstraction and the ability to automatically learn complex features present in images [65]. Although similar to ANN, DL consists of a "deeper" neural network, capable of providing a hierarchical representation of the data, which allows equipping these models with strong learning capabilities, quite valuable for answering different types of problems and adapting to their complexity.

The main DL architecture used for image processing are the *Convolutional Neural Networks* (CNN) [17]. This is a type of ANN that makes use of convolution operations in at least one of its layers. The application of DL and CNN in agriculture has been extensively reviewed [16, 65-67], established itself as a promising and efficient approach to overcome several challenges in agriculture related to computer vision, mainly fruit detection and classification.

Unlike conventional ANN, CNN are faster at learning and interpreting complex, large-scale problems due to the sharing of weights and the use of more sophisticated models that allow massive parallelisation [68]. Figure 13 illustrates the architecture of CNN which can be divided into 2 major parts:

- image feature extraction;
- image classification.

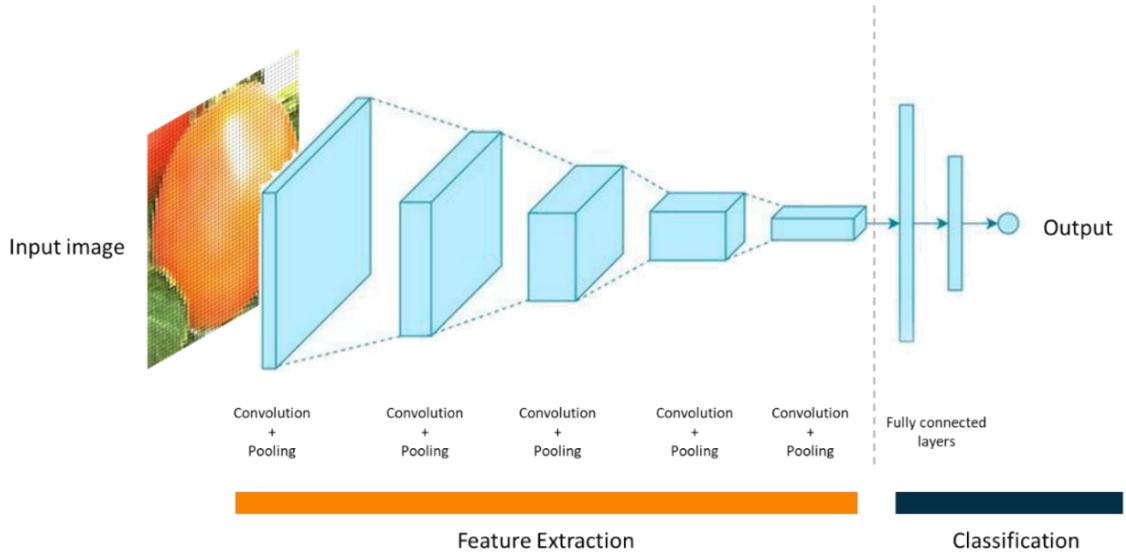


Figure 13 | *Convolutional Neural Network architecture divided into two main phases: feature extraction through convolution layers and classification made by fully connected layers.*

At first, as the name implies, CNN seeks to extract high-level features through convolution/pooling processes. This information is then transferred to the fully connected layers (i.e ANN where all the inputs from one layer are connected to every activation unit of the next layer) responsible for object detection and classification. To better understand all these processes, Appendix B describes all the steps that make up a CNN architecture.

Over the years, several CNN architectures have been successfully developed, making it easier to build models so that they do not have to be created from scratch. Each architecture has its advantages and disadvantages, as well as scenarios where they can be used in a more appropriate way. Some examples of these architectures are AlexNet [69], Visual Geometry Group (VGG) (Fig. 14) [70], Inception [71], ResNet [72] or MobileNet [73].

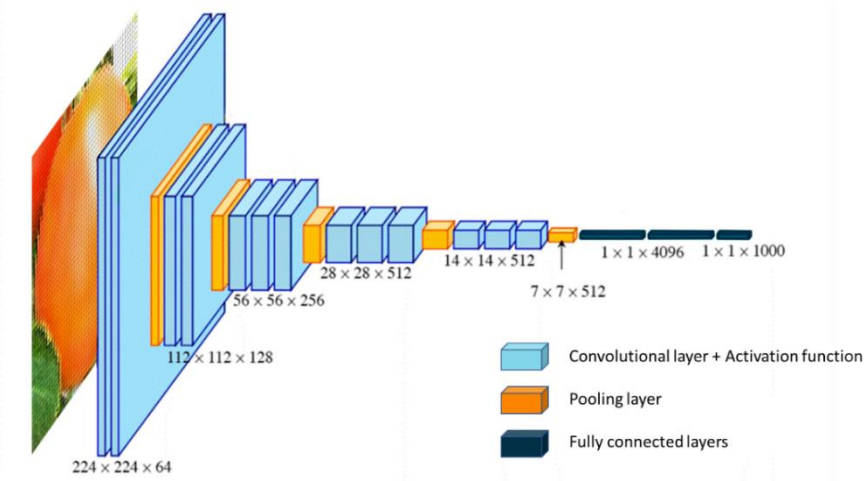


Figure 14 | *CNN VGG16 architecture. Adapted from: Simonyan and Zisserman [70].*

The most elementary feature extraction methods, which rely on the "manual" selection of certain features such as colour, shape or texture, can see their effectiveness dissipate when faced with the agricultural environment's problems (i.e. variations in illumination, occlusions, overlaps, etc.). The great advantage of DL models is that they do not require "manual" feature extraction. However, these can be used as a processing input, automatically selecting and classifying relevant features [66].

However, a CNN can only infer a single class of a given object per image. Thus, several object detection frameworks have been developed over the years, which can locate and classify the presence of multiple objects in a single image [20]. These frameworks usually consist of two parts (Fig. 15): the backbone, which is no more than a CNN responsible for feature extraction, and the head, which predicts the classes and the location of objects. The head can be organised into two types:

- Two-stage detection;
- One-stage detection.

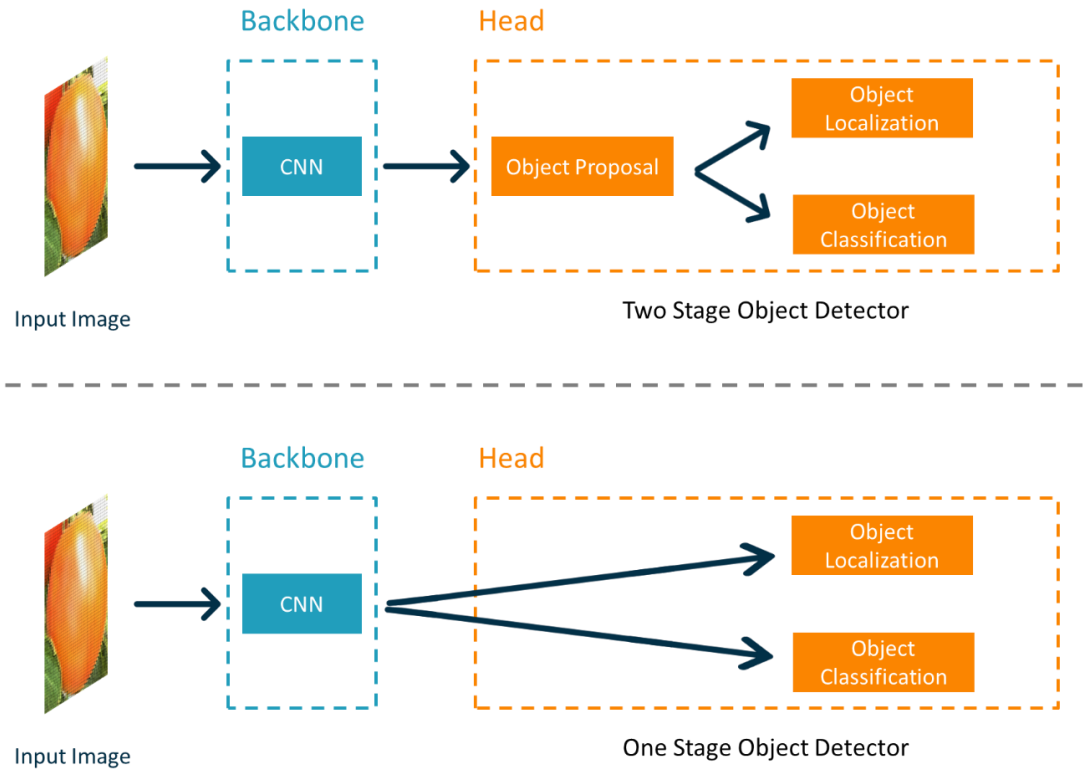


Figure 15 | *Two major types of object detection frameworks: Two-stage detectors and One-stage detectors.*

Two-stage detection frameworks, as the name implies, require two steps to perform object detection and classification. The first is the region proposal step and the second is the detection/classification step. The most notable framework with this type of performance is Faster-RCNN (Regions with CNN's) [74]. In this case, a CNN called Region Proposal Network (RPN) is used, responsible for proposing rectangular regions (bounding boxes) that may contain the object to be detected. Then a region classifier such as Fast-RCNN [75] is used to classify the regions proposed by the RPN [76].

In order to increase the speed of all these processes, frameworks that perform localization and classification in a single step were developed. In this case, the region proposal step has been removed and CNNs are used that consider a dense sampling of possible locations of the objects to detect. This process is less time consuming and can therefore be used in real-time applications. Although some one-stage detection frameworks do not perform as well as two-stage frameworks, they are much faster [77]. However, by prioritising inference speed, they end up having some disadvantages, especially when detecting objects with irregular shapes or small sizes [20]. The most popular approaches are the SDD (Single Shot Multibox Detector) [18] and YOLO (You

Only Look Once) [19], which, as they are used within the scope of this work, are described in more detail in Appendix C.

In order to implement these DL frameworks in object detection, the processes required can be broken down into 3 fundamental parts [13]:

- data input;
- model building;
- generalization.

The initial step consists in obtaining a set of images that contain features to be considered in model learning. This dataset must be carefully chosen so that it is relevant to the problem at hand and that it has variations consistent with the implementation context [66]. To facilitate this process, numerous public datasets of already annotated images are available, including for the agricultural context [78], which can be used to compare already trained models or to increase the dataset. Generally, this dataset is divided into training, validation and test sets, as well as the respective annotations of those images and the classes of objects to be detected. The training set is used to build the models. The validation set is used to fine-tune some model parameters, such as confidence or overlap thresholds, before being applied to the test set to achieve generalization [66].

Model building is when models learn to recognise a given object through the relevant features they extract from the images in the training set. A common practice in model training is called transfer learning (referred to as fine-tuning). This is a technique used to train DL models more efficiently and stably, as it allows the reuse of existing parameters (convolution weights) of a pre-trained model on large datasets. Basically, the weights of the initial layers of the pre-trained model are copied to the new model. Still, the final classification layer, responsible for classifying objects, is not transferred, and the new model is in charge of that part, being trained for the new classes [66].

Finally, the test set provides an unbiased evaluation of a final model fit on the training data set. The model needs to be assessed for its accuracy relative to the data it was trained on, so that it can then predict the outputs of inputs it has never been trained on (generalisation). This evaluation can be done using several metrics, as indicated by Padilla, Netto [79] or Kamilaris and Prenafeta-Boldú [80].

2.3.1.3 Computer Vision for Tomato Detection and Classification

This section presents some algorithms, methods and techniques proposed by different authors regarding fruit detection and classification, more specifically in tomato (Tab. 1).

In the segmentation and detection of tomatoes, still in the plant, a RoI in the canopy is often used, which may include, besides the fruit, other structures, especially the leaves that may difficult the detection of the fruit (occlusion or overlap) mainly in the early stages of ripening (Fig. 16). Therefore, the colour is a feature used recurrently to differentiate the object to be detected, in this case, the fruit from everything external to it and from the background that, can be very complex at the crop level. Several colour spaces such as HSV, HSI (Hue, Saturation and Intensity), L*a*b* and RGB, among others, are used to extract this feature. Besides, mathematical morphology approaches [81] combined with Machine Learning techniques have also been used in fruit detection in occlusion and overlap situations.

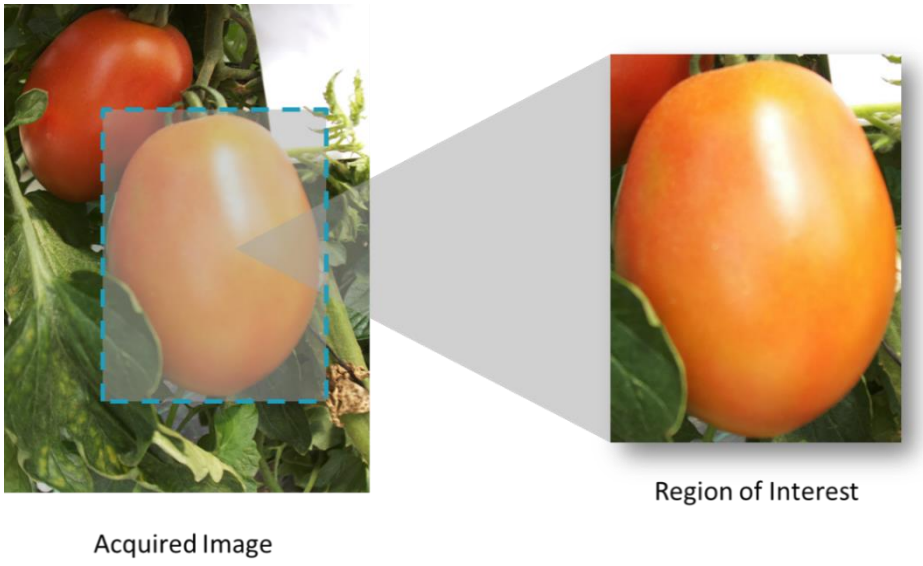


Figure 16 | *Example of an original image and the Region of Interest to be detected.*

In order to develop a harvesting robot in greenhouses, Yin, Chai [82] segmented riped tomatoes through K-means clustering using the colour space L*a*b*, recording an average task execution time of 10.14 seconds. Huang, Yang [83] used the colour space L*a*b* to segment and localize riped tomatoes in a greenhouse and bi-level partition fuzzy logic entropy to discriminate the fruits from the background, but the results were

not quantified. Zhao, Gong [84], developed a detection algorithm capable of recognizing green, intermediate, and riped tomatoes. First, images of component a* and images of component L* were extracted from the colour space L*a*b* and the luminance of the Quadrature-phase (YIQ) colour space, respectively. Then, wavelet transformation was adopted to merge the images at a pixel level, which combined the information from the two original images. Finally, to differentiate the fruit from the background, an adaptive threshold algorithm was used to obtain the optimal threshold. When testing, 93% of the tomatoes were detected. Arefi, Motlagh [85] proposed an algorithm for recognising riped tomatoes through a combination of RGB, HSI and YIQ colour spaces and morphological characteristics of the image. The algorithm obtained a total accuracy of 96.36% when tested in a greenhouse with artificial lighting. Qingchun, Wang [86] developed a riped tomato harvesting robot for a greenhouse, whose identification and location of fruits consist of transforming RGB colour space images into a HIS colour model to identify and locate the fruits. The robot performs this task in 4 seconds, and the harvest success rate is 83.9%. Zhang [87], aiming to detect riped tomatoes, also converted the RGB colour space into an HSI colour space. The riped tomato region was cut based on the grey distribution of the H component using the threshold method. The Canny operator [88] was used to detect the edges, and after a corrosive expansion, the coordinates of the center of the tomato were marked. The results were not quantified.

Benavides, Cantón-Garbín [89] designed a computer vision system for the detection of riped tomatoes in greenhouses. The segmentation of the fruit was mainly done based on the colour and edges of the fruit, using the R component of the RGB images and the Sobel operator [90], respectively. Clustered tomatoes were detected with a precision of 87.5% and beef tomatoes with 80.8%. Malik, Zhang [91] presented a riped tomato detection algorithm based on HSV colour space and the watershed segmentation method. In order to remove the background and detect only riped tomatoes, the HSV colour space was used, and through morphological operations it was possible to modify the detected fruits. The watershed segmentation algorithm was implemented to "separate" the clustered fruits. The combination of these two methods led to a precision of 81.6%. Zhu, Yang [92] combined mathematical morphology with a Fuzzy C-Means (FCM) based method for detecting riped tomatoes in a greenhouse, with no results reported. Again, based on mathematical morphology, Xiang, Ying [93] tested a riped cluster tomato recognition algorithm. The algorithm is divided into 4 fundamental steps: tomato image segmentation, performed based on a normalized colour difference; recognition of the clustered region according to the length of the longest edge of the

minimum enclosing rectangle of the tomato region; clustered regions, in a binary image, were processed by an iterative erosion course to separate each tomato in this clustered region and every seed region in the clustered region acquired by the iterative erosion was restored using a circulatory dilation operation. At a distance of 500 mm, they achieved a detection rate of 87.5%, while between 300 and 700 mm the rate dropped to 58.4%.

Yamamoto, Guo [94] used different Machine Learning techniques to detect and distinguish the different stages of tomato ripeness. The proposed method consists of 3 steps: pixel-based segmentation conducted to roughly segment the pixels of the images into classes composed of fruits, leaves, stems and background; Blob-based segmentation to eliminate the wrong classifications generated in the first step, and finally X-means clustering was applied to detect fruits individually in a fruit cluster. The results indicated a precision of 88%. Zhao, Gong [95], to detect riped tomatoes, extracted the Haar-like features of grey-scale image, classifying them with AdaBoost classifier. The false negatives derived from this classification were eliminated using a colour analysis approach based on the average pixel value. The results showed that the combination of AdaBoost classification with the colour analysis allowed a 96% detection rate, although 10% were false negatives and 3.5% of the fruits were not detected. Liu, Mao [96], proposed an algorithm for the detection of greenhouse riped tomatoes, where the Histograms of Oriented Gradients (HOG) descriptor was used to train a Support Vector Machine (SVM) classifier. A coarse-to-fine scanning method was developed to detect the fruit, followed by a proposed False Color Removal (FCR) method to eliminate false-positive detections. The Non-Maximum Suppression (NMS) method was finally used to merge the overlapping results. The algorithm was able to detect the fruits with an accuracy of 94.41%. Wu, Zhang [97] developed a greenhouse riped tomato detection algorithm for a harvesting robot, through a method that combines analysis and selection of multiple features, a Relevance Vector Machine (RVM) classifier and a bi-layer classification strategy. The algorithm demonstrated an accuracy of 94.90%. Wang, Zhao [48], developing a greenhouse harvest robot for tomatoes, used the Otsu segmentation algorithm [49] to automatically detect riped tomatoes, obtaining success rates of 99.3%.

In recent years, the use of ML and especially DL techniques in fruit detection has been increasingly tested and used. Unlike conventional methods, it is a more robust and accurate alternative with better response to occlusion and green tomato detection

problems. This problem is rarely studied due to the difficulty of segmentation and differentiating it from the background, as it has similar colours (Fig. 17).

Figure 17 | *The great colour correlation between the green tomatoes and the background, which makes their detection and subsequent harvesting very difficult.*

The comparison made by Alam Siddiquee, Islam [98] can observe this, who compared a ML method, known as "Cascaded Object Detector" with a system that combines more traditional methods of image processing, named "Colour Transformation", "Colour Segmentation" and "Circular Hough Transformation", in the detection of riped tomatoes. The results showed that the accuracy of the ML method is 95% better than conventional methods.

Xu, Jia [99], have improved the YOLOv3-tiny method to obtain a faster and more accurate detection of riped tomatoes. The model's accuracy was increased by enhancing the backbone network, and the image enhancement allowed better detection in more complex scenarios. The results show that the F1-score of the proposed model is 91.92%, which is 12% higher than the unmodified YOLOv3-tiny method. Liu, Nouaze [100] used the YOLOv3 detection model to create the YOLO-Tomato model, which was possible to achieve due to the incorporation of dense architecture for feature extraction and the replacement of the traditional R-box by the proposed C-box. In scenarios with moderate occlusions, the model obtained a detection rate of 94.58%, 4% more than in scenarios with severe occlusions. In order to overcome overlaps and occlusions, Sun, He [101] developed a detection method based on Convolutional Neural Network (CNN), more

specifically the Feature Pyramid Network (FPN) method. The proposed method has improved the detection rate from 90.7% to 99.5% by comparing this method with traditional Faster R-CNN models. Mu, Chen [102] built a tomato detection model capable of detecting green tomatoes in greenhouses, regardless of possible occlusions. The model uses a pre-trained Faster R-CNN structure with the deep CNN Resnet-101 based on the Common Objects in Context (COCO) dataset, which was then fine-tunned for tomato detection, reaching an accuracy of 87.83%.

As mentioned before, the SSD model promises a substantial improvement in fruit detection and therefore has been increasingly studied, since it can capture the information of an object and its anti-interference and directly complete the localization and the classification task in just one step. This improvement is demonstrated by de Luna, Dadios [103], who designed a computer visualization system to evaluate the growth of tomato plants through the detection of fruits and flowers. Two DL models were used: R-CNN and SSD. The fruit detection accuracy of the R-CNN model was only 19.48%, while the SSD model showed a much higher detection rate of 95.99%. Yuan, Lv [21] developed an SSD-based algorithm to detect cherry tomatoes in greenhouses, whether riped, green or intermediate. After creating the datasets, they were used to train and develop network models. To study the effect of the base network, one of the experiments was tested on different networks, such as VGG16, MobileNet, Inception V2. The results indicated that the Inception V2 network obtained the best performance with an accuracy of 98.85%.

Table 1 | *Algorithms, methods and techniques proposed by different authors regarding tomato detection at different ripeness levels.*

Method	Tomato Ripeness	Inference time (s) or Accuracy (%)	Authors
L*a*b* color space and K-means clustering	Ripe	10.14 s	Yin, Chai [82]
L*a*b color space and Bi-level partition fuzzy logic entropy	Ripe	—	Huang, Yang [83]
L*a*b color space and Threshold algorithm	Green, Intermediate and Ripe	93%	Zhao, Gong [84]
RGB, HSI, and YIQ color spaces and Morphological characteristics	Ripe	96.36%	Arefi, Motlagh [85]
RGB color space images into a HIS color model	Ripe	4 s and 83.90%	Qingchun, Wang [86]
RGB color space into an HSI color space, treshhold method and Cany operator	Ripe	—	Zhang [87]
R component of the RGB images and Sobel operator	Ripe	Clustred tomatoes - 87.50% Beef tomatoes - 80.80%	Benavides, Cantón-Garbín [89]
HSV color space and Watershed segmentation method	Ripe	81.60%	Malik, Zhang [91]
Mathematical morphology and Fuzzy C-Means base method	Ripe	—	Zhu, Yang [92]
Mathematical morphology, Normalized color difference and Iterative erosion course	Ripe	At 500 mm distance - 87.50% Between 300 and 700 mm - 58.40%	Xiang, Ying [93]
Pixel-based segmentation, Blob-based segmentation and X-means clustering	Green, Intermediate and Ripe	88%	Yamamoto, Guo [94]
Haar-like features of gray scale image and AdaBoost classifier	Ripe	96%	Zhao, Gong [95]
Histograms of Oriented Gradients and Support Vector Machine	Ripe	94.41%	Liu, Mao [96]
Selection of multiple features; Relevance Vector Machine and bi-layer classification strategy	Ripe	94.90%	Wu, Zhang [97]
Otsu segmentation algorithm	Ripe	99.30%	Wang, Zhao [48]
Improved YOLOv3-tiny method	Ripe	91.92% (F1 score)	Xu, Jia [99]
YOLOv3 detection model to create the proposed YOLO-Tomato model	Green, Intermediate and Ripe	94.58%	Liu, Nouaze [100]
Feature Pyramid Network	Green, Intermediate and Ripe	99.50%	Sun, He [101]
Faster R-CNN structure with the deep convolutional neutral network Resnet-101	Green	87.83%	Mu, Chen [102]
Comparation: R-CNN vs SSD	Green, Intermediate and Ripe	R-CNN: 19.48% SSD: 95.99%	de Luna, Dadios [103]
SSD network models such as VGG16, MobileNet, Inception V2	Green, Intermediate and Ripe	Best performance: Inception V2 network with 98.85%	Yuan, Lv [21]

3. Materials and Methods

3.1 Harvesting robots in protected horticulture: Systematic Review

A systematic review is presented to answer the following research question: What is the academic overview of the advances in automated harvesting, in the protected horticulture sector?

Harvesting robots can be divided into two types: bulk harvesting (all fruits/vegetables are harvested) or selective harvesting (only riped or ready-to-harvest fruits are collected) [7]. This review focuses mainly on robots for selective harvesting, as they are of greater interest in the research world. The publications concerning this type of robots may cover complete systems, i.e., the development of an entire robot, or support tasks in the robotic harvesting aid, such as fruit detection or manipulation, system localization and navigation.

Therefore, the following specific questions were answered to answer the above question: i) How is the distribution of publications by crop? ii) From which countries are the authors that publish the most about robotic harvesting? iii) Are the articles related to complete systems or to support tasks? Furthermore, iv) What are the main support tasks?

The Web of Science database*,* “Principle Collection”, was used to access bibliographic records around the theme "harvesting robots in the protected horticulture" from 1990-2021. The Web of Science tool was used because its citation analysis provides better graphs and is more detailed than the citation analysis of the Scopus tool [104]. Also, it does not present results of inconsistent accuracy, as happens with Google Scholar [105].

In order to identify relevant publications on the subject, the following keywords were chosen in the search fields combining title, abstract and the author's keywords: "Robo* harv* OR Mechanic* harv*" AND "Greenhouse OR Greenhouse horticulture". The key: “AND Tomato*" was added to focus the research on the tomato crop. It is important to note that, due to the limitations of these databases, some sources with adherence to the theme may be missing in the results obtained.

This review will enable researchers to identify niches of opportunity to do research, and know the main topics and the emerging issues of harvesting automatization in a greenhouse environment, so that it can be fully achieved in the near future.

3.2 Dataset Acquisition

Recently, there has been an increasing proliferation of public datasets containing large amounts of annotated images, such as the COCO [106], PASCAL VOC [107] or Open Image (OID) [108] datasets. However, the nature of the images that compose them cannot directly translate to Precision Agriculture (PA) applications.

This can be evidenced by one of the papers developed in parallel with this dissertation, where the OIDv6 dataset was benchmarked against an acquired dataset inside greenhouse for tomato detection, using four DL object detectors [109]. The results highlight the benefit of using self-acquired datasets to detect tomatoes because the state-of-the-art datasets lack some relevant features of the fruits in the agricultural environment, such as the shape or colour. Most of these datasets have few annotations per image and the tomato is generally riped.

Thus, specialized datasets for PA tasks have emerged as Lu and Young [78] survey shows. However, despite the datasets reviewed targeting fruit detection, none of them represents the type of data intended to be detected and classified in this study, as they all lack images of tomatoes, specifically in a greenhouse environment.

To overcome this bottleneck, two image datasets of tomatoes of the “Plum” variety at different ripeness stages were collected in greenhouses. Both datasets were made publicly available at the open-access digital repository Zenodo:

- AgRobTomato Dataset [110];
- RpiTomato Dataset [111].

AgRobTomato Dataset images were collected on two different days (August 6 and 8, 2020) at a greenhouse in Barroselas, Viana do Castelo, Portugal (Fig. 18 a). To increase the representativeness of the data, the mobile robot AgRob v16 (Fig. 18 b), controlled by a human operator, was guided through the greenhouse inter-rows and captured RGB

images of the tomato plants using a ZED camera[2], recording them as a video in a single ROSBag file. The camera was mounted on an anthropomorphic manipulator, which remained in the rest position, looking sideward, towards the tomato plants, during the whole acquisition process. The video was converted into images by sampling a frame every 3 seconds to reduce the correlation between images, ensuring an overlapping ratio of about 60%. The images collected on the two days were merged, resulting in a dataset of 449 images with a resolution of 1280x720 px each.

(a) (b)

Figure 18 | *Barroselas Greenhouse configuration (a) and the AgRob v16 robot used for image collection (b). Source: INESC TEC.*

To collect more information from the fruits, namely the Brix degree as presented in section 3.4.1, a total of 60 tomatoes from the same variety were collected from a different greenhouse located in Amorosa, Viana do Castelo, Portugal, on June 15, 2021 (Fig. 19 a). Before being collected, RGB images of each fruit were captured from different perspectives. The images were taken with a Raspberry Pi Computer Model B[3] with 4GB RAM, connected to a Raspberry Pi High Quality Camera[4] (12.3 MP and 7.9 mm diagonal image size) with a 6 mm (wide angle) CS-mount lens with 3 MP (Fig. 19 b). A total of 258 images were obtained, which made the RpiTomato Dataset.

[2] ZED Dual Camera (https://www.stereolabs.com/zed/). Last accessed: 15 August 2021
[3] Raspberry Pi Computer Model B (https://www.raspberrypi.com/products/raspberry-pi-4-model-b/). Last accessed: 15 August 2021
[4] Raspberry Pi High Quality Camera (https://www.raspberrypi.com/products/raspberry-pi-high-quality-camera/). Last accessed: 15 August 2021

(a) (b)

Figure 19 *| Amorosa Greenhouse configuration (a) and the Raspberry Pi high quality camera attached to a Raspberry Pi Computer Model B used for image collection (b).*

3.3 Tomato Detection and Classification

3.3.1 Classes

The general focus of the ML field is to predict an outcome using the available data. The prediction task can be called a "detection problem" when the outcome represents a single class. On the other hand, if the outcome represents different classes, it means a "classification problem". Through the acquired datasets, two one-stage object detection frameworks (SSD and YOLO) were evaluated in tomato detection and compared with a novel HSV Colour Space model in tomato classification.

When it comes to classification, this study aims to differentiate the fruits according to their ripeness stage. Considering the colected images, 4 classes were defined based on the USDA colour chart for fresh tomatoes [112] (Fig. 20):

- Green (a) – More than 90% of the surface is green;
- Turning (b) – 10 to 30% of the surface is yellow;
- Light Red (c) – Between 60 to 90% of the surface is red;
- Red (d) – 90 to 100% surface is red.

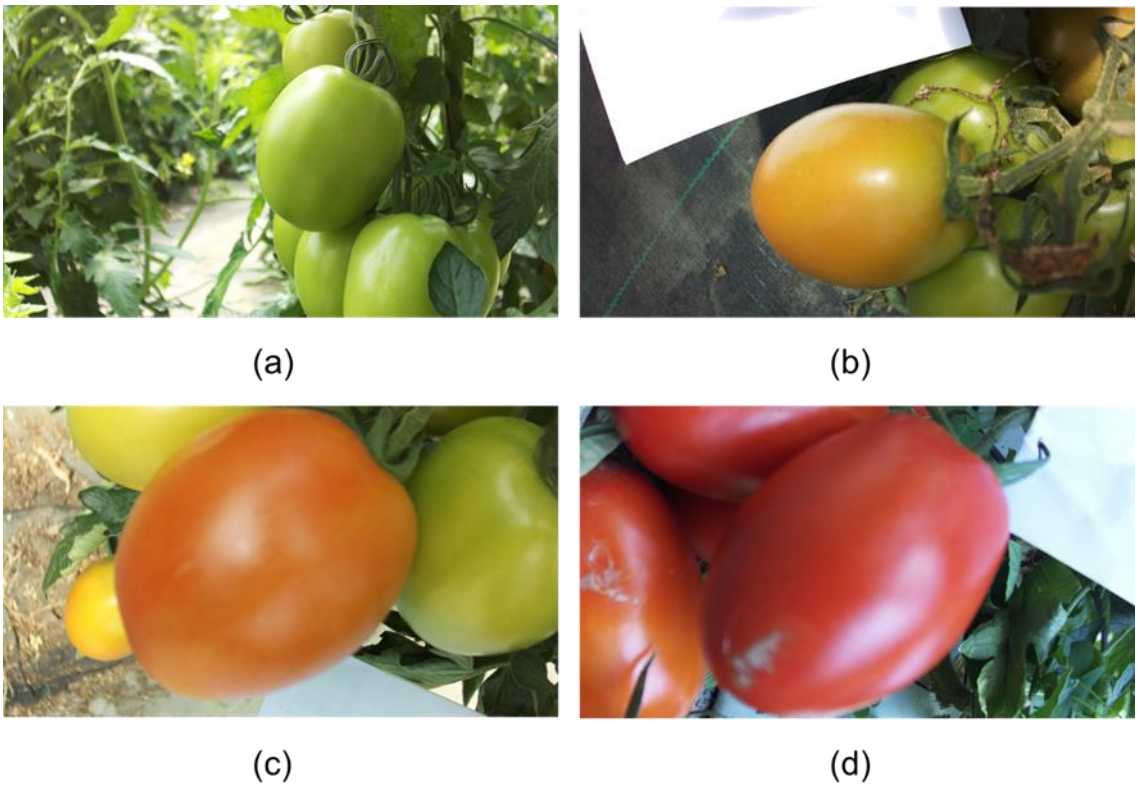


Figure 20 | *Classification classes defined according to the colour of a tomato during ripening: Green (a); Turning (b); Light Red (c); Red (d).*

3.3.2 Data Processing

Since it involves supervised learning, the models need to be provided with an annotated dataset. Thus, the images from the AgRob Dataset were manually annotated using the open-source annotation tool CVAT[5] [113], indicating by rectangular bounding boxes the position and class of each plant (Fig. 21). Regarding the detection, the images were annotated considering only the class "tomato", as the goal is that the fruits are detected regardless of their ripeness. For the classification, the images were annotated with the 4 chosen maturity classes. In essence, two independent annotated datasets were obtained, one to train and evaluate the models for tomato detection and the other for tomato classification.

[5] Computer Vision Annotation Tool (CVAT) (https://cvat.org). Last accessed: 22 October 2021

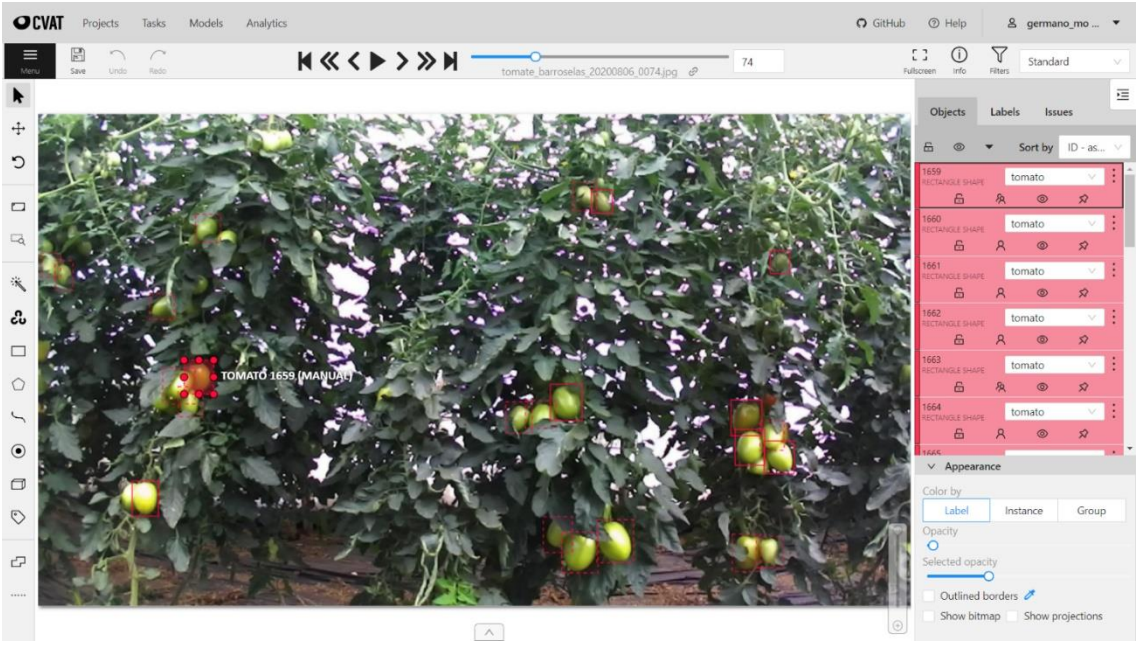


Figure 21 | *Image annotation performed through the CVAT tool.*

After annotating, the images of both datasets were exported under the Pascal VOC format [107] and the YOLO format to train the SSD and YOLO frameworks, respectively. The Pascal VOC format resumes the annotations for each image in a single .xml file. Each annotation identifies its class, size and position, and include some additional features of the annotations as whether the target object is difficult to detect, occluded or truncated. On the other hand, the YOLO format comprises a .txt file where each line represents an annotation and, besides the numerical identifier of the corresponding class, it contains the coordinates of the bounding box. This format requires an additional file where each numerical identifier corresponds to a class [114].

High-resolution DL models are time and computationally consuming and cannot process full-sized images, considering the input of square images, thus rescaling them before processing. For this reason, to avoid distortion, the original images were split into images with a resolution of 720x720 px. Thus, the number of images in the datasets was doubled to 898 images. Yet, some images contained few annotations, and the splitting resulted in non-annotated images. These images were then removed from the datasets, being left with 849 images.

To train and validate the different models, the datasets were divided into 3 sets:

- Training set (60% of the data);
- Validation set (20% of the data);
- Test set (20% of the data).

Some studies reported the use of data augmentation techniques. Data augmentation can artificially increase the dataset, improving the overall learning procedure and performance by inputting varied data into the model [80]. In this case, transformations were only applied to the training and validation sets. The transformations were carefully chosen, applying those that could happen in an actual situation, that is, the ones that the robot's vision could be confronted with when performing the harvesting task in a greenhouse environment. The transformations were applied with a random factor and are as displayed in Figure 22.

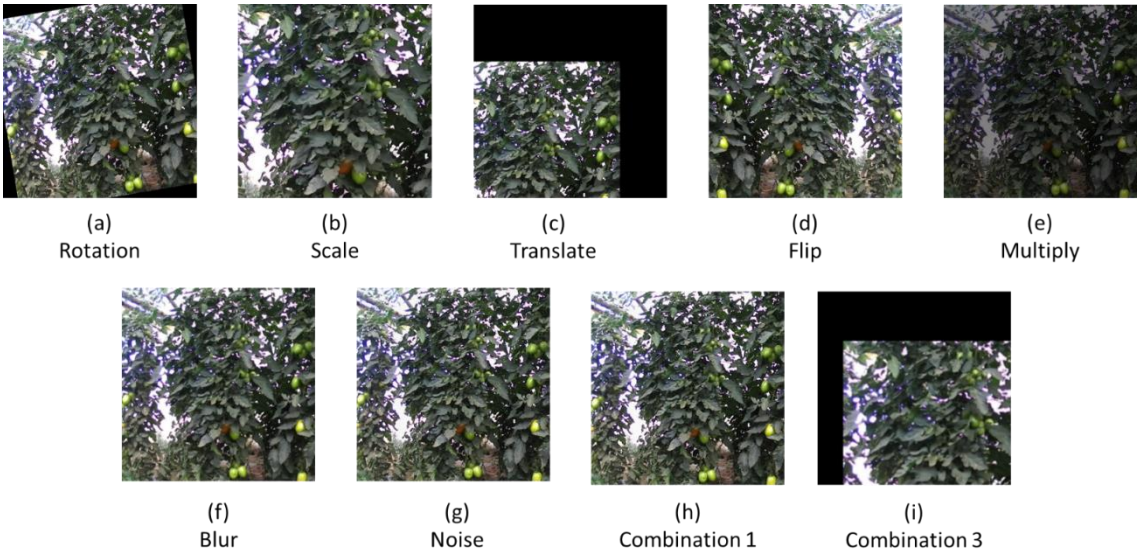


Figure 22 *| Different types of transformation applied to the AgRob Dataset images: Rotation (a); Scale (b); Translate (c); Flip (d); Multiply (e); Blur (f); Noise (g); Combination1 (h) and Combination3 (i), which are a random combination of 1 or 3 of the previous transformations.*

The data augmentation led to 7,608 annotated images. The training and validation sets contained 5,590 and 1,849 images respectively, while the test set was composed of 169 images.

3.3.3 Deep Learning Models Training

The literature refers to several ML frameworks, an interface, library, or tool that easily creates ML models [65]. Since it is desired that the robot uses a TPU (Tensor Processing Unit), the choice of framework falls on TensorFlow[6] [115], an open source easily scalable ML library developed by Google, which provides a collection of workflows to develop and train models using Python, C++, JavaScript, or Java.

By the time all these processes had been performed, only TensorFlow 1 had fully compatible tools to train and compile the models to the TPU. Then, TensorFLow r.1.15.0 was used for the training and inference scripts, which run on Google Collaboratory (Colab) notebooks[7] that give free access to powerful GPU's (Graphics Processing Unit) and TPU's to develop DL models. Although the GPU's available may vary for each Colab session, in general an NVIDIA Tesla T4 with a VRAM of 12 GB and a computation capability between 3.5 and 7.5 was assigned to all sessions.

Based on the additional contribution in two scientific articles [116] [109], the best performing models of each article were chosen for benchmarking purposes. Therefore, one pre-trained SSD MobileNet v2 model from the TensorFlow database[8] and one YOLOv4 model from the Darknet database[9] were considered. Both models were pre-trained with Google's COCO dataset[10] [106] with an input size of 640x640 px (SSD MobileNet v2) and 416x416 px (YOLOv4).

Through transfer learning, a fine-tune was performed to the pre-trained models to detect and classify tomatoes. Slight changes to the default training pipeline were made, such as adjusting the batch size for each model (24 to the SSD MobileNet v2 model and 64 to the YOLOv4 model) and removing data augmentation from the pipeline. The SSD MobileNetv2 model training sessions ran for 50 000 epochs, while the YOLOv4 model training was much faster, requiring only 10 000 epochs. The number of epochs may vary from model to model, but in this case was chosen based on the suggestion given by the literature, but mainly taking into account the "average loss" training metric, selecting the number of epochs that would be sufficient to converge. As far as the MobileNet v2 SSD

[6] TensorFlow (https://www.tensorflow.org/). Last accessed: 3 November 2021
[7] Google Colab (https://colab.research.google.com/). Last accessed: 25 September 2021
[8] SSD MobileNet v2 model (https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf1_detection_zoo.md). Last accessed: 3 November 2021
[9] YOLOv4 model (https://github.com/zauberzeug/darknet_alexeyAB). Last accessed: 3 November 2021
[10] COCO Dataset (https://cocodataset.org/#home). Last accessed: 3 November 2021

model is concerned, an evaluation session occurred at every 50 epochs, following the standard value used by the pre-trained models. Since Darknet had no available validation sessions, it was not considered for the YOLOv4 model. These evaluation sessions are quite useful, since they allow monitoring the evolution of the training, meaning if the evaluation loss started to increase while the training loss decreased or remained constant, the deep learning model was over-fit to the training data.

Figure 23 reports an overview of all the required steps used to reach the trained DL.

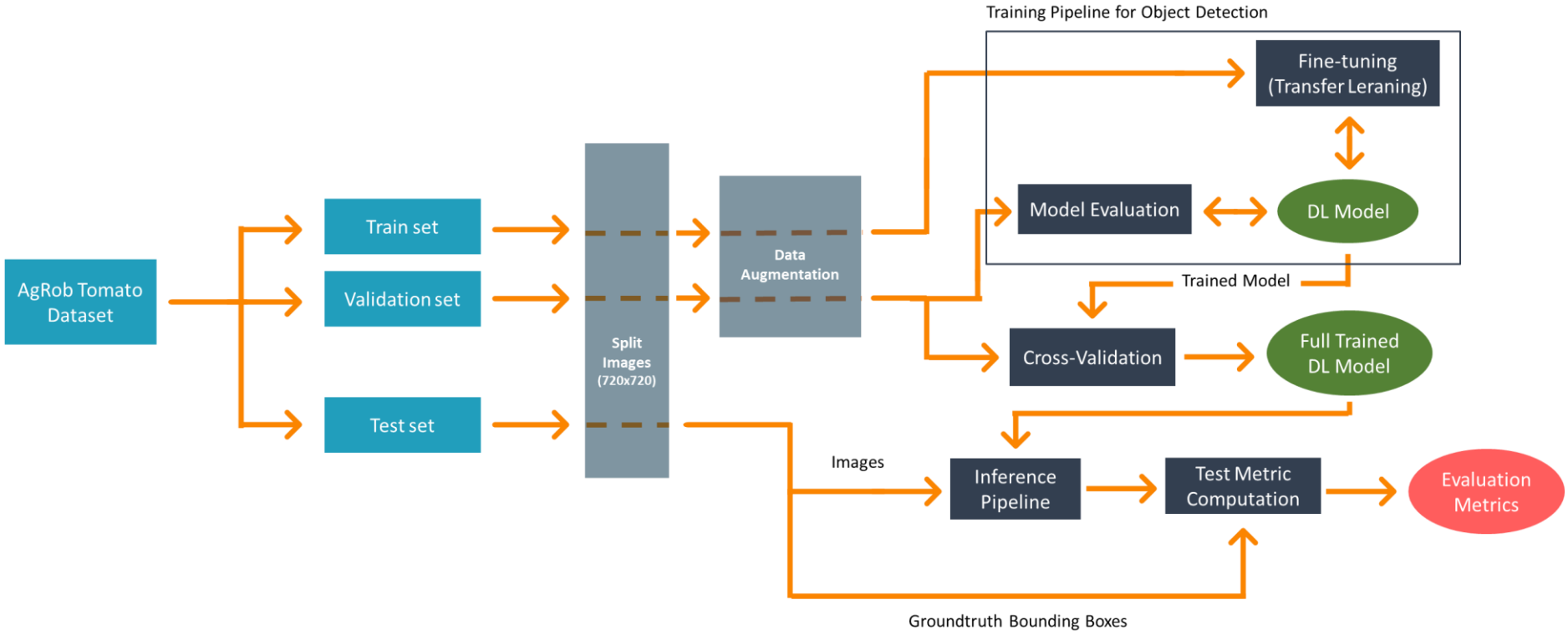


Figure 23 | *Workflow of the performed methods to reach the trained DL models.*

3.3.4 HSV Colour Space Model Development

An approach based on histograms from the HSV color space was developed as an alternative to DL models for tomato classification. All the scripts used throughout this process are authorship and were created from scratch through Spyder[11], an open-source cross-platform integrated development environment for scientific computing in the Python language, provided by Anaconda software[12]. The final HSV Colour Space model and the scripts can be found in the following GitHub repository:

https://github.com/gerfsm/HSV_Colour_Space_Model

In order to build the model, images of 10 tomatoes from each ripeness class were selected. To add some variability, half of the images come from the AgRobTomato Dataset and the other from the RpiTomato Dataset, as they present different perspectives of the fruits. The AgRobTomato Dataset offers a farther perspective, while in the RpiTomato Dataset the fruits are closer.

The first step was to extract the RoI from the images[13]. All the images were labelled using the annotation tool CVAT and the coordinates of the annotation bounding box were used to segment the image and extract the RoI (Fig. 24).

Figure 24 | *Segmentation of the image RoI to be classified via the coordinates of the annotation bounding box.*

[11] Spyder (https://www.spyder-ide.org/). Last accessed: 25 October 2021
[12] Anaconda (https://www.anaconda.com/). Last accessed: 25 October 2021
[13] Script: ROI's_crops.py (https://github.com/gerfsm/HSV_Colour_Space_Model/blob/Scripts/ROI's_crops.py). Last accessed: 12 June 2021

The next step was to convert the RoI images from RGB to HSV colour space[14]. The RGB colour information is usually much noisier than the HSV information. Thus, using only the Hue channel makes a computer vision algorithm less sensitive, if not invariant, to problems like lighting variations. For example, a green tomato can be exposed to different lighting conditions; in both conditions the tomato has exactly the same Hue value, but widely different RGB values.

The image's colour space conversion (Fig. 25) was performed through the function "cv.cvtColor()"[15] from OpenCV, a real-time optimized Computer Vision library [117].

Figure 25 | *Conversion of a RoI's RGB colour space to HSV colour space.*

For each HSV image, a colour histogram was generated focusing only on the Hue channel[16]. OpenCV was used to extract the colorimetric data from the RoI. Different applications use different scales to represent the HSV colour space. For the Hue values, OpenCV uses a scale ranging between 0-179. Since the interest is focused on analyzing the region of colours that a tomato can display, the entire colour spectrum is unnecessary. Therefore, the location of the origin for the Hue parameter was changed, giving the histogram a normal distribution.

[14] Script: RGB_to_HSV.py (https://github.com/gerfsm/HSV_Colour_Space_Model/blob/Scripts/RGB_to_HSV.py). Last accessed: 13 June 2021
[15] Changing Colorspaces – cv.cvtColor() (https://docs.opencv.org/4.5.2/df/d9d/tutorial_py_colorspaces.html). Last accessed: 13 June 2021
[16] Script: HSV_Histogram.py (https://github.com/gerfsm/HSV_Colour_Space_Model/blob/Scripts/HSV_Histogram.py). Last accessed: 22 July 2021

Through Matplotlib [118], a comprehensive library for creating static, animated, and interactive visualizations in Python, the function “matplotlib.pyplot.hist”[17] was used to plot the histogram (Fig. 26). In this case, the function parameter "density" was set to "True", which causes a probability density to be drawn and returned. It was preferred to use all the bins in the range to get as accurate a model as possible. Each bin displays the bin's raw count divided by the total number of counts and the bin width, so that the area under the histogram integrates to 1.

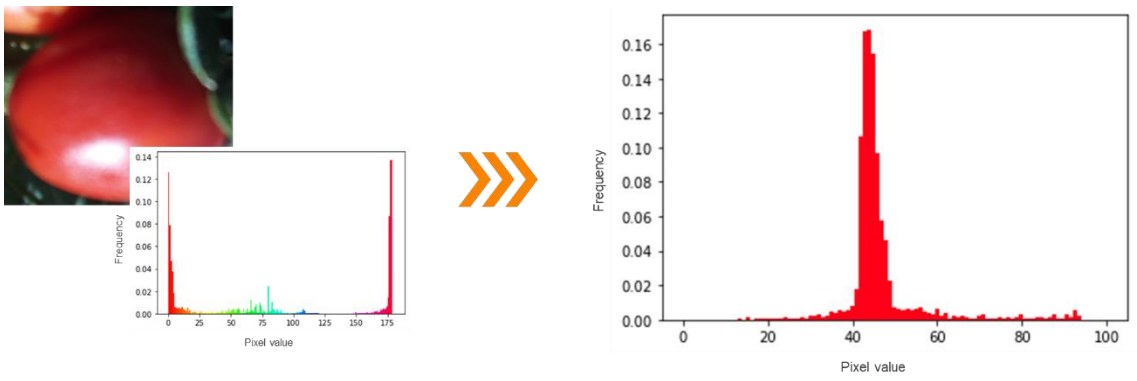


Figure 26 | *Example of the histogram plot with normal distribution, based on the Hue values of the HSV colour space of a Red tomato.*

Although the segmentation technique used is faster and easier to implement, it can be noted that the RoI covers the object to be classified and some of the background, which slightly affects the results obtained. In some cases, the background makes the data look multimodal (Fig. 27), i.e. there is more than one "peak" data distribution. Trying to fit a multimodal distribution with a unimodal (one "peak") model will generally give a poor fit and lead to incorrect classifications.

[17] Plot a histogram – matplotlib.pyplot.hist (https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.hist.html). Last accessed: 22 July 2021

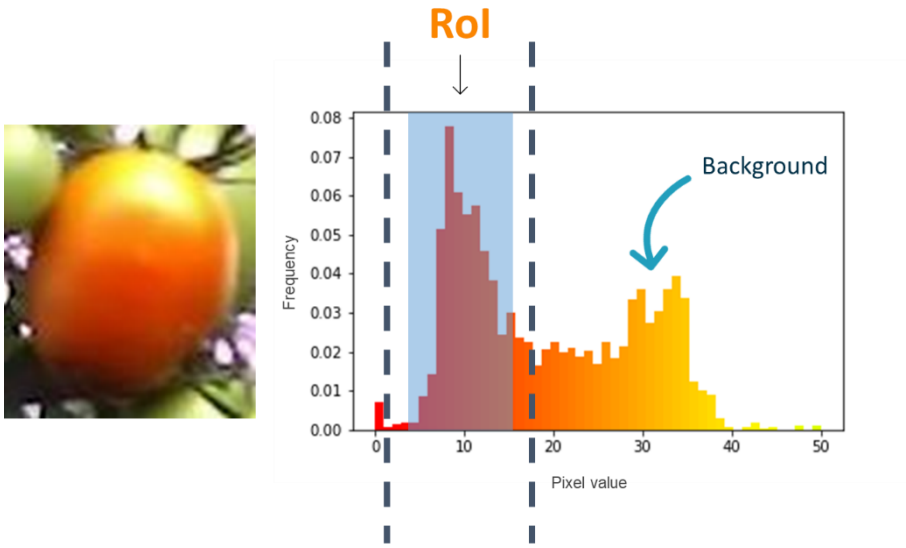


Figure 27 | *Histogram affected by background colorimetric information. The green colour of the tomatoes in the background is displayed in the histogram and makes the data distribution bimodal.*

A Gaussian mixture model was used to overcome this problem. This function is a probabilistic model for representing normally distributed subpopulations within an overall population (Fig. 28) and was applied to the data using the function "sklearn.mixture.GaussianMixture"[18], from the module sklearn.mixture that implements mixture modeling algorithms. Sklearn (or scikit-learn) [119] is a useful library for ML in Python, which contains a lot of efficient tools for statistical modelling.

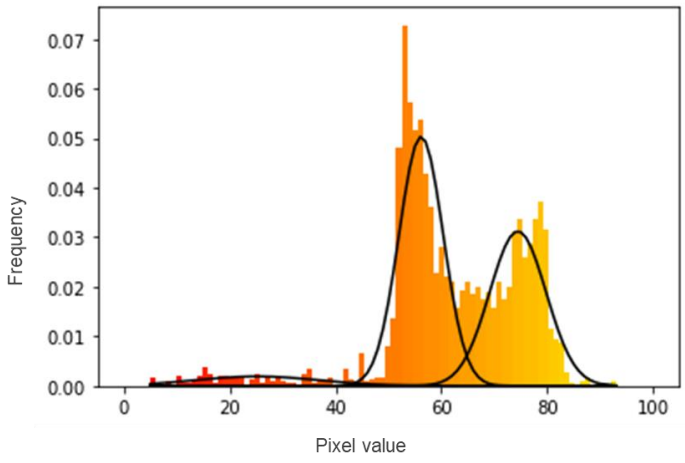


Figure 28 | *Representation of a Gaussian mixture model probability distribution.*

[18] Gaussian Mixture – sklearn.mixture.GaussianMixture (https://scikit-learn.org/stable/modules/generated/sklearn.mixture.GaussianMixture.html). Last accessed: 29 July 2021

The next step was to choose the Gaussian with the highest peak[19], which corresponds to the RoI, and ignore the rest (Fig. 29 a). The curve was selected according to the Gaussian mixture weights. These weights are normalized to 1, motivated by the assumption that the model must explain all the data, then using the law of total probability. So, in that sense, they are the probabilities of the point being part of the cluster. In other words, the weights are the estimated probability of a draw (i.e distribution curve) belonging to each respective normal distribution. Even with the background noise, the data distribution in the RoI zone is well defined. The probability that this region is a normal distribution is much higher, meaning that the weight is higher. Selecting the higher weights leads to the RoI Gaussian.

For a more careful analysis, a boxplot was also generated for each RoI, through the function "matplotlib.pyplot.boxplot"[20]. The boxplots represent the values within 3 standard deviations of the mean, corresponding to 99.7% of the Gaussian (Fig. 30 b).

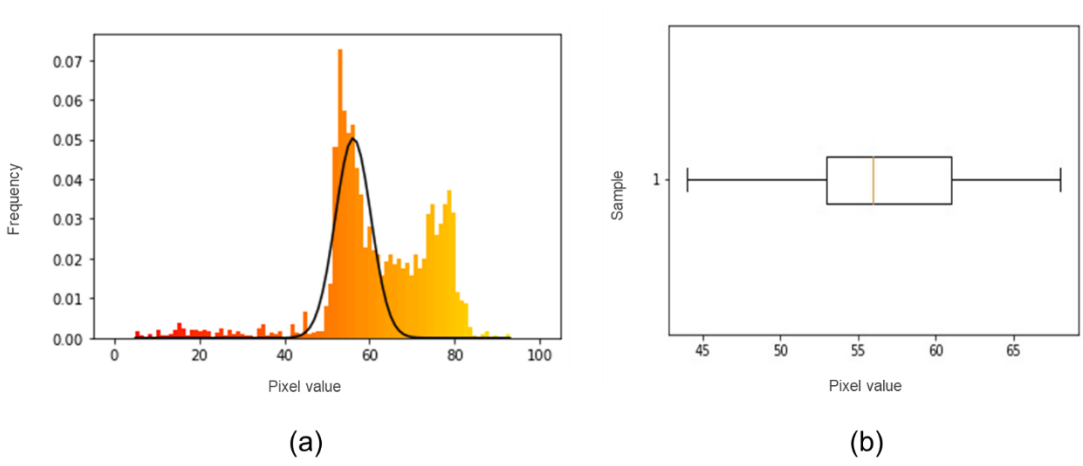


Figure 29 | *Final representation of the histogram with the Gaussian corresponding to the fruit to be classified (a) and its boxplot (b).*

[19] Scripts: HSV_Histogram_GaussianMix.py (https://github.com/gerfsm/HSV_Colour_Space_Model/blob/Scripts/HSV_Histogram_GaussianMix.py). Last accessed: 29 July 2021

[20] Box and whisker plot – matplotlib.pyplot.boxplot (https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.boxplot.html). Last accessed: 29 July 2021

Based on the results obtained, correlating the mean histogram of each sample with its respective class, a statistical classifier was reached. These results will be presented and exploited later in the respective Results and Discussion section.

Achieving the classifier culminated in the ultimate model. For a specific image, given the bounding boxes coordinates of the fruits to be classified (input), in a single pass, the HSV Colour Space model segments the RoI's, converts them to the HSV colour space, through the colorimetric information. Also, the Gaussian Mixture probabilistic model generates a histogram and calculates it's mean that through the statistical classifier generates an output. The model returns the class to which that fruit belongs.

Figure 30 reports an overview of all the required steps used to reach the HSV Colour Space model.

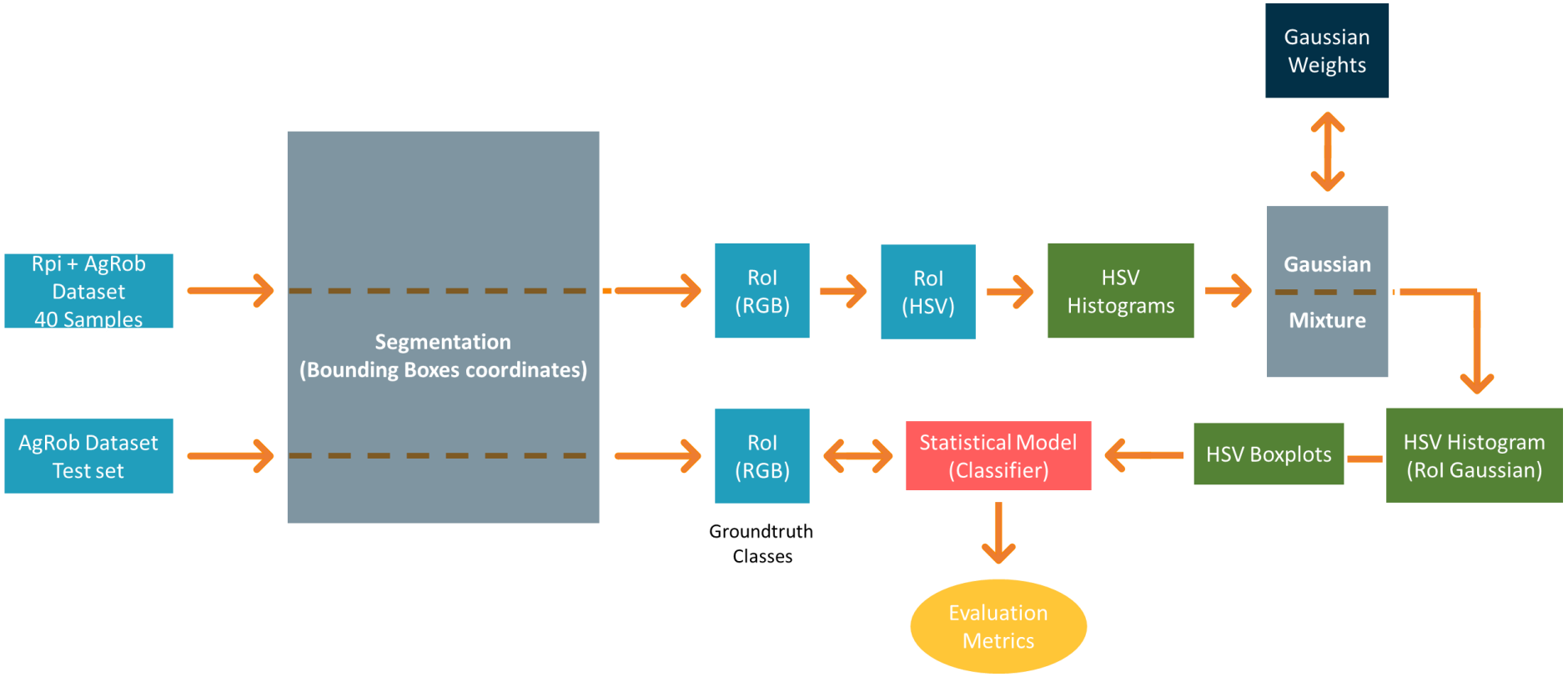


Figure 30 | *Workflow of the performed methods to reach the developed and evaluated HSV Colour Space model.*

3.3.5 Evaluation Metrics

All models used were evaluated with the images from the AgRobTomato Dataset. DL models trained with a single class were logically evaluated on the detection problem. However, the DL models trained with all 4 ripeness classes in addition to the classification were also evaluated on the detection problem, as it is still necessary to compare the detection ability of models trained with a single class with models trained with multiple classes. For this purpose, the 4 classes were considered as one. The HSV Colour Space model was only evaluated for the classification problem.

A “correct detection” is commonly established through the Intersection over Union (IoU) metric when it comes to the detection problem. The IoU measures the overlapping area between the predicted bounding box (B_p) and the groundtruth bounding box (B_{gt}) divided by the area of union between them (eq. 1), as illustrated in Figure 31. In this case, a correct detection was considered if IoU ≥ 50%.

$$IoU = \frac{Area\left(B_p \cap B_g\right)}{Area\left(B_p \cup B_g\right)} \quad (1)$$

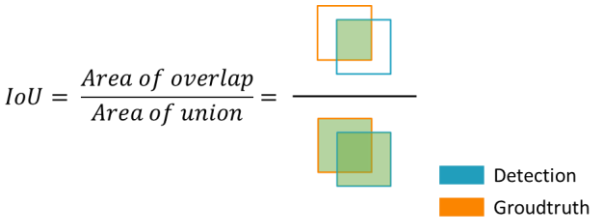


Figure 31 | *Representation of the Intersection Over Union (IoU) metric.*

To better benchmark the two DL models, the metrics used by the Pascal VOC challenge [107] (Precision x Recall curve and Mean Average Precision) were chosen, with the addition of the following metrics:

- Recall;
- Precision;
- F1-Score.

Recall (eq. 1) is the ability of the model to detect all the relevant objects (i.e all groundtruth bounding boxes). Precision (eq. 2) is the ability to identify only the relevant objects and the F1-Score (eq. 3) is the first harmonic mean between Recall and Precision. The number of groundtruths (relevant objects) can be computed by the sum of the True Positives and False Negatives (TP+FN) and the number of detections is the sum of the TP's and False Positives (TP+FP). TP are the correct detections of the groundtruths, FP are improperly detected objects and FN are undetected.

$$Recall = \frac{TP}{TP + FN} = \frac{TP}{All\ groundtruths} \quad (1)$$

$$Precision = \frac{TP}{TP + FP} = \frac{TP}{All\ detections} \quad (2)$$

$$F1\text{-}Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (3)$$

All the detections performed by a DL model have a confidence rate associated with them. It is a value that features the certainty in the performed prediction (i.e. a confidence rate of 50 % determines that the network is 50% sure of the detected or classified object). Graphically representing the ratio between Precision and Recall (Precision x Recall curve) can be seen as a trade-off between Precision and Recall for different confidence values associated with the bounding boxes generated by a detector. The higher the confidence, the higher the Precision of the model (low FP's). However, many groundtruths may be missed, yielding high FN's, and thus a low Recall rate. A great object detector is one that keeps its Precision rates high, while its Recall increase. Thus, a high Area Under the Curve (AUC) tends to indicate both high Precision and Recall.

However, it is difficult to accurately measure the AUC, as the Precision x Recall curve is often a zigzag-like curve. To overcome this problem is often calculated the Average Precision (AP) metric. Since the Pascal VOC challenge metrics [107] are considered in this study, AP was calculated by the all-point interpolation approach. In this case, the AP

(eq. 4) is obtained by interpolating the Precision at each level, taking the maximum Precision ($P_{interp}(R)$) whose Recall value is greater or equal than R_{n+1}.

$$AP = \sum_{n} (R_{n+1} - R_n) P_{interp}(R_{n+1}) \quad (4)$$

If there is more than one class to detect, the Mean Average Precision (mAP) metric is used, simply the average AP over all classes (eq. 5). AP_i represents the AP of class i and NC is the number of classes evaluated.

$$mAP = \frac{\sum_{i=1}^{NC} AP_i}{NC} \quad (5)$$

The final step of the inference was to optimize the confidence score, using the cross-validation technique: AgRobTomato's validation set augmentations were removed, and the F1-Score was computed for all the confidence thresholds from 0% to 100%, into steps of 1%. The confidence threshold that optimises the F1-Score was selected for the model's normal operation. The AgRobTomato's test set was used to evaluate both models, and the whole inference process occurred on the Google Colab server, using a Tesla T4 GPU.

To assess the classification ability, based on an overview by Grandini, Bagli [120], the main evaluation metric chosen was a confusion matrix, along with the Precision and Recall for each class that will act as building blocks for the Macro F1-Score and Balanced Accuracy metrics.

A confusion matrix (Fig. 32) gives a simple yet efficient performance measures for a classification model. Each entry denotes the number of predictions made by the model where it classified the classes correctly or incorrectly. The rows allow inferring about the Precision and the columns about the Recall of each class. Along with the TP's, FP's and FN's, the True Negatives (TN) are also considered, referring to the number of predictions where the classifier correctly predicts the negative class as negative. In this case the number of groundtruths becomes the sum of these four indicators.

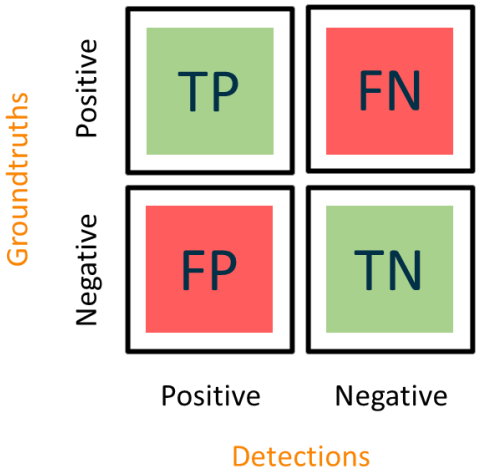


Figure 32 | *Example of a confusion matrix for Binary Classification*

The choice of using Macro F1-Score and Balanced Accuracy, instead of single F1-Score and Accuracy, comes from the fact that the test set, and the whole dataset, are quite unbalanced. The test set contains a lot of green tomatoes (over 1000 samples), unlike the other classes, where the Red class has the poorest representation, with only 4 tomatoes. The Balanced Accuracy metric is a simple arithmetic mean of Recall of each class, so every class has the same weight and importance, therefore being balanced (eq. 6). To achieve the Macro F1-Score, it is necessary to compute Macro-Precision and Macro-Recall, computed as the arithmetic means of the metrics for single classes. Again, each class has the same weight in the average, so that there is no distinction between highly and poorly populated classes. Macro F1-Score is the harmonic mean of Macro-Precision and Macro-Recall (eq. 7).

$$Balanced\ Accuracy\ = \frac{\sum_{1}^{No.Classes} \frac{TP}{Total\ Groundtruths}}{No.Classes} \tag{6}$$

$$Macro\ F1\text{-}Score\ = 2 \times \frac{Macro - Precision \times Macro - Recall}{Macro - Precision + Macro - Recall} \tag{7}$$

3.4 Tomato Phenotyping

3.4.1 Brix Degree Measurement and Prediction

Assessing the fruit quality for fresh consumption includes different aspects, which can also help establish the harvesting moment. One of the methods used is the approximate measurement of the fruit's sugar content, through the Brix degree (Brixº). Brixº values are important because they can be measured objectively and they relate to a subjective criterion that consumers use to assess fruit quality: flavor or sweetness.

Still, it is rarely used before harvesting because it is a destructive method. The ability to extract other relevant information, in addition to detecting and classifying, could increase the quality of automated harvesting, making it more selective and adaptable. Taking this into consideration, one of this study's goals is to understand if there is any relationship between fruit colour (ripeness stage) and the Soluble Solids Content (SSC), so that it can be estimated in a simple and non-destructive way.

The SSC of the 60 tomato samples was measured by a handheld Milwaukee MR32ATC refractometer (Milwaukee, USA), inside the greenhouse on the same day the fruits were collected. The tomatoes were cut in half, and a few drops were squeezed into the detection window to record the data. The average value of each sample repeated 3 times was the final SSC value of the sample.

The correlation between fruit colour and SSC was performed by averaging the results obtained for each class and comparing them. However, since assigning a class to a fruit based on human visual perception is an empirical and somewhat subjective task, another way to understand the colour-Brix correlation was to use the HSV Colour Space model. The model assigns a "value" to the colour, making interpretations more reliable and accurate.

As previously mentioned, some samples used to construct the HSV Colour Space model came from the Raspberry Dataset, collected during the Brix measurement. HSV histograms were generated for these samples and the average of each was compared with its measured SSC value.

4. Results & Discussion

4.1 Meta-analysis: Harvesting robots in protected horticulture

Table 2 compiles all the results obtained, presenting them by crop and type of robot system (harvesting robot or support task), describing and associating them with the respective authors and countries of origin. Note that some articles refer to the same project.

Table 2 | *Description of harvesting robots and support tasks applied in protected horticulture, based on the crops, authors and countries of origin.*

Crop	System	Description	Author	Country
Asparagus	Harvesting Robot	Harvesting robot coordinated with 3D vision sensor	Irie, Taguchi [142]	Japan
Cucumber	Harvesting Robot [1]	Modular harvesting robot	Van Henten, Hemming [160]	The Netherlands
Cucumber	Support Task	Detection: Dynamic threshold segmentation algorithm	Qi, Yang [155]	China
Cucumber	Support Task	Detection: Fusion method (colour and texture features) based on HIS, MSER and HOG	Li, Zhao [149]	China
Cucumber	Support Task	Detection: Image segmentation algorithm based on rough set theory	Qing-Hua, Li-Yong [156]	China
Cucumber	Support Task	Detection: Machine vision algorithm based on near-infrared spectral imaging	Yuan, Xu [166]	China
Cucumber	Support Task [1]	Manipulation: Kinematic structure of a manipulator (four link PPRR type)	Van Henten, Slot [162]	The Netherlands
Cucumber	Support Task [1]	Manipulation: Inverse kinematics algorithm	Van Henten, Schenk [161]	The Netherlands
Green Perilla	Support Task	Detection: Leaf recognition method based on DNN techniques	Masuzawa, Miura [152]	Japan
Pea	Harvesting Robot	Harvesting robot based on VIS–NIR reflection analysis, global thresholding, texture and shape modelling	Tejada, Stoelen [159]	Norway
Saffron	Harvesting Robot	Harvesting robot developed using scalability properties and computer vision	Perez-Vidal and Gracia [154]	Spain
Strawberry	Harvesting Robot	Harvesting robot mounted on a travel platform	Hayashi, Yamamoto [138]	Japan
Strawberry	Harvesting Robot	Harvesting robot based on 3D vision due to three RGB cameras	De Preter, Anthonis [133]	Belgium
Strawberry	Support Task	Manipulation: End-effector for application to an elevated-substrate culture	Yamamoto, Hayashi [163]	Japan

(Cont.)

Sweet Pepper	Harvesting Robot [2]	Harvesting robot with 6 DOF, custom designed end effector and a RGB-D camera	Arad, Balendonck [121]	The Netherlands
Sweet Pepper	Harvesting Robot	Harvesting robot with 3 DOF, cylindrical end effector and three color CCD cameras	Lee, Kam [148]	South Korea
Sweet Pepper	Harvesting Robot [3]	Harvesting robot with 9 DOF, color cameras and a ToF camera	Bac, Hemming [123]	The Netherlands
Sweet Pepper	Support Task	Detection: 3D pose estimation using a model matching algorithm	Eizentals and Oka [134]	Japan
Sweet Pepper	Support Task [2]	Detection: Adaptive image-dependent thresholding method using reinforcement learning	Ostovar, Ringdahl [153]	Sweden
Sweet Pepper	Support Task [2]	Detection: Flash-no-Flash approach comparing a simple detection algorithm and a deep learning model	Arad, Kurtser [122]	The Netherlands
Sweet Pepper	Support Task [2]	Detection: Large-scale semantic image segmentation datasets based on empirical data	Barth, Ijsselmuiden [127]	The Netherlands
Sweet Pepper	Support Task [2]	Detection: Modular software framework design to implement an eye-in-hand sensing and motion control	Barth, Hemming [126]	The Netherlands
Sweet Pepper	Support Task [2]	Detection: Statistical models for fruit detectability	Kurtser and Edan [146]	The Netherlands
Sweet Pepper	Support Task [2]	Detection: Dynamic sensing algorithm to select the best-fit viewpoint location	Kurtser and Edan [147]	The Netherlands
Sweet Pepper	Support Task [3]	Detection: Effect of multiple camera positions and viewing angles on fruit detectability	Hemming, Ruizendaal [139]	The Netherlands
Sweet Pepper	Support Task [3]	Detection: Stem localization with a developed algorithm using a support wire as a visual cue and stereo-images	Bac, Hemming [124]	The Netherlands
Sweet Pepper	Support Task	Detection: CDD LED lighting system	Kitamura and Oka [144]	Japan
Sweet Pepper	Support Task	Detection: Multi-target positioning approach based on deep CNN	Chen, Li [130]	China
Sweet Pepper	Support Task	Detection: Least-squares SMV optimized by the improved particle swarm optimization (IPSO-LSSVM)	Ji, Chen [143]	China
Sweet Pepper	Support Task [3]	Manipulation: Azimuth angle of the end-effector and sensitivity analysis for five paremeters	Bac, Roorda [125]	The Netherlands
Sweet Pepper	Support Task [3]	Manipulation: Efficient two-stage trajectory planning approach for a redundant harvesting manipulator	Schuetz, Baur [157]	Germany
Sweet Pepper	Support Task [3]	Manipulation: Design and testing of two end-effectors (Fin Ray and Lip type)	Hemming, van Tuijl [140]	The Netherlands
Sweet Pepper	Support Task [2]	Phenotyping: Camera viewpoint and fruit orientation on the performance of a maturity level classification algorithm	Harel, van Essen [137]	The Netherlands
Tomato	Harvesting Robot	Truss tomato harvesting robot and two key technologies of picking-point recognition and end-effector design	Ji, Zhang [43]	China
Tomato	Harvesting Robot [4]	Dual-arm harvesting robot with two 3 DOF manipulators	Zhao, Gong [44]	China

(Cont.)

Tomato	Harvesting Robot	Cherry tomato harvesting robot composed by a IR reflective sensor and a Pixy camera	Taqi, Al-Langawi [158]	Kuwait
Tomato	Harvesting Robot [5]	Harvesting robot using infrared image and specular reflection	Yasukawa, Li [164]	Japan
Tomato	Harvesting Robot	Harvesting robot with 5 DOF and a binocular stereo vision system	Wang, Zhao [48]	China
Tomato	Harvesting Robot	Harvesting robot with 3 DOF (Analysis of workspace and kinematics)	Li, Liu [42]	China
Tomato	Support Task	Detection: K-means clustering using the L*a*b* color space	Yin, Chai [165]	China
Tomato	Support Task	Detection: RGB, HSI, and YIQ colour spaces and morphological characteristics	Arefi, Motlagh [85]	Iran
Tomato	Support Task	Detection: Fuzzy C-Means based method combined with mathematical morphology	Zhu, Yang [92]	China
Tomato	Support Task	Detection: L*a*b colour space and Bi-level partition fuzzy logic entropy	Huang, Yang [83]	China
Tomato	Support Task [4]	Detection: Haar-like features of gray scale image and AdaBoost classifier	Zhao, Gong [95]	China
Tomato	Support Task [5]	Detection: Machine learning using infrared image and specular reflection	Fujinaga, Yasukawa [136]	Japan
Tomato	Support Task	Detection: Histograms of Oriented Gradients and SVM	Liu, Mao [151]	South Korea
Tomato	Support Task	Detection: Faster R-CNN structure with the deep CNN Resnet 101, Resnet 50 and Inception-Resnet v2	Mu, Chen [102]	Japan
Tomato	Support Task	Detection: SSD-based algorithm to train and develop network models such as VGG16, MobileNet, Inception V2	Yuan, Lv [21]	China
Tomato	Support Task	Detection: R component of the RGB images and Sobel operator	Benavides, Cantón-Garbín [89]	Spain
Tomato	Support Task	Manipulation: Two end-effectors for petty-tomato developed using a pneumatic tube	Kondo, Shibano [145]	Japan
Tomato	Support Task	Manipulation: Design of an end-effector with 4 DOF and his workspace through the Monte Carlo method	Cui, Hua [132]	China
Tomato	Support Task	Manipulation: End effector with four fingers and a centrally located fruit suction device	Chiu, Yang [131]	Taiwan
Tomato	Support Task	Manipulation: PR-APT method for planning a trajectory of the manipulator end-effector	Boryga, Graboś [129]	Poland
Tomato	Support Task	Manipulation: Dual-arm cooperative approach using a binocular vision sensor	Ling, Zhao [150]	China
Tomato	Support Task [5]	Phenotyping: method of generating a map of the tomato growth states	Fujinaga, Yasukawa [135]	Japan
Watermelon	Harvesting Robot	Multi-functional tele-operative modular robotic system	Heon and Si-Chan [141]	South Korea

In total, 56 articles were obtained [21, 42-44, 48, 83, 85, 92, 95, 102, 121-166] and Figure 33 shows the main crops studied. It can be seen that harvest robotisation is a studied and sought-after topic in several crops, however, crops such as tomato (22 articles) or sweet-pepper (19 articles) appear at the leading edge when it comes to research, representing about 73% of the results obtained.

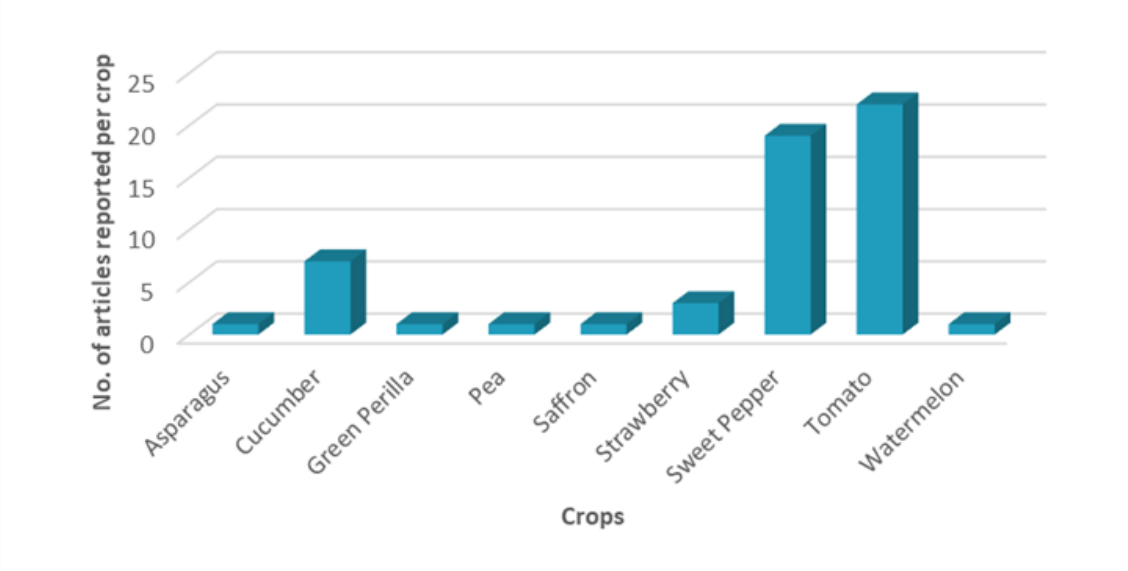


Figure 33 | *The main crops studied in robotic harvesting for protected horticulture based on the number of published articles.*

Regarding the countries, China (17 articles; 30%), the Netherlands (15 articles; 27%) and Japan (11 articles; 20%) are the ones leading (77%) the research on harvesting robots (Fig. 34). Although it presents 15 articles, roughly half of the Dutch articles belong to a single project - SWEEPER, a European Union project to create a greenhouse pepper harvesting robot, which ended in the year 2018 [121]. Generally speaking, Asian countries have a slight advantage over European ones.

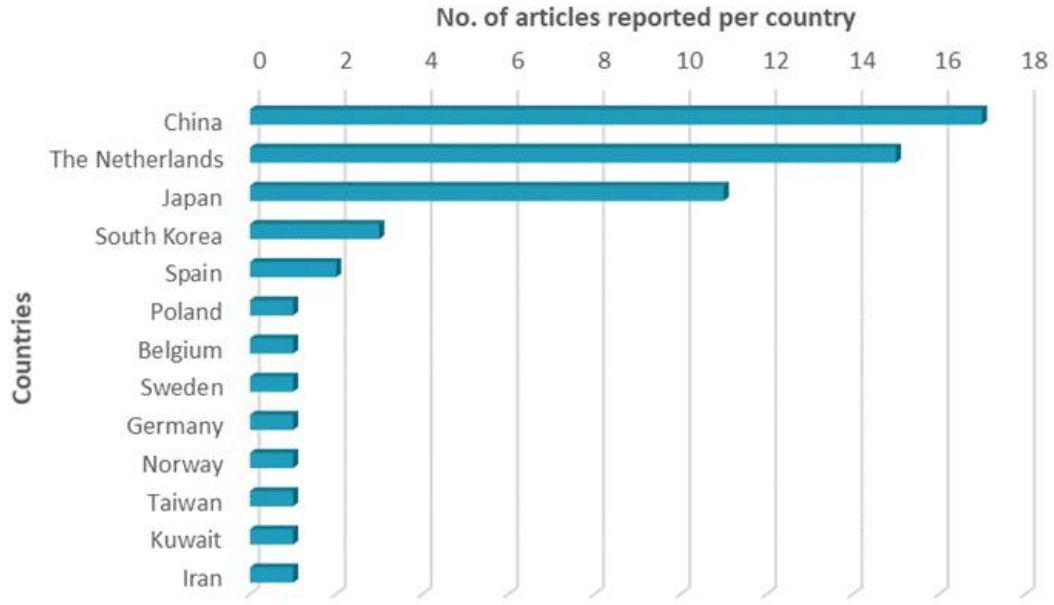


Figure 34 | *Countries with most published articles on robotic harvesting in protected horticulture.*

Despite the high number of results, only 16 articles (28%) are related to complete systems, with emphasis on tomato (6 articles), strawberry (3 articles) and sweet-pepper (2 articles) crops. The remaining projects are related to support tasks (Fig. 35), which corroborates why few solutions have reached commercialisation phases and how difficult it is to design these systems. A robot, for whatever operation, is composed of different support tasks. All of them need to be robust and efficient enough for the system to work perfectly. With high performance levels, justifying its acquisition and consequent use - for example, the fruit detection can be highly precise and exact in a harvesting robot. Still, suppose the manipulator does not correspond and fails to pick the fruit from the plant or causes damage to it. In that case, the system as a whole automatically ceases to have any relevance, being unable to perform the task at hand, in this case, the fruit harvesting.

For all this, it is clear that the research is in a development phase in the attempt to improve the support tasks so that, in the future, a complete solution can be reached more quickly.

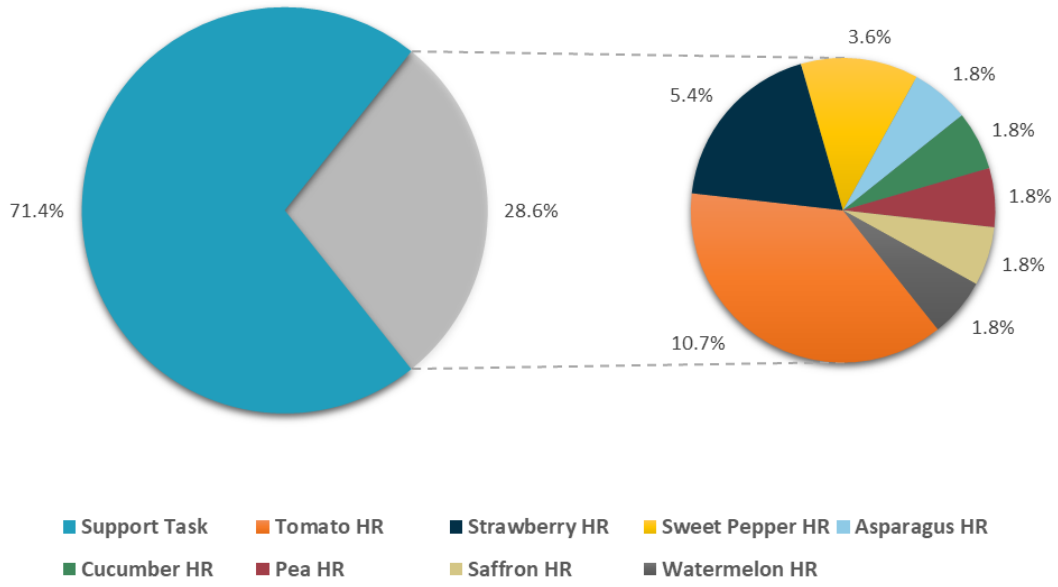


Figure 35 | *Percentage of articles related to support tasks and harvesting robots according to the crops for which they were developed. Abbreviations: HR = Harvesting Robot.*

Among the support tasks, detection stands out (27 articles), followed by manipulation (11 articles) and phenotyping (2 articles) (Fig. 36). The large number of studies directed towards fruit detection is, to some extent, easy to explain as it is essential to obtain a robotic harvest. Logically, the fruit will only be harvested or even phenotyped, if detected, showing that detection is one of the first steps to be taken towards automating the harvesting task.

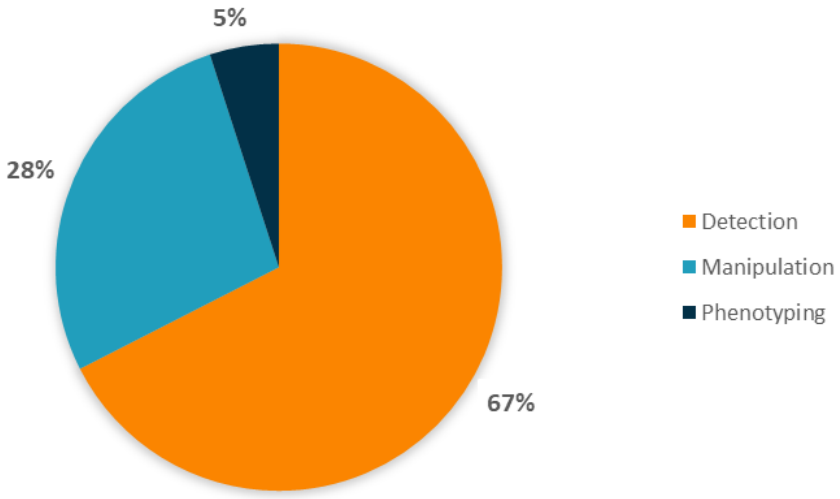


Figure 36 | *Percentage of articles assigned to different support tasks.*

Overall, the performance of harvesting robots is far from being on par with the human workforce. From the results, the detection of the objects to be harvested and their manipulation are the two major obstacles on the path to automation. The first is related to the unstructured environment, where light conditions and obstructions by different plant parts (leaves, stems or even other fruits by overlapping) make the robot's vision extremely difficult. The second is due to the sensitive nature of agricultural products, which require careful handling. Developing and optimising these systems becomes vital, as they should be equipped with better software and hardware. Software to deal with problems associated with detection and the robot's visual perception and hardware to ensure that harvesting is done safely without damaging fruits and plants [7].

4.2 Single Class Tomato Detection

As mentioned, the models required defining the best confidence threshold before proceeding to evaluate their performance. Table 3 indicates the value of the confidence threshold that maximises the F1-Score for each model, finding the best balance between the Precision and Recall, optimising the number of TP's while avoiding the FP's and FN's. Both models found their best F1-Score at similar confidence thresholds, however the YOLOv4 model achieved a better F1-Score of about 87%.

Table 3 | *Confidence threshold for each DL model (1 class) that optimises the F1-score metric.*

DL Model	Confidence Threshold ≥	F1-Score
SSD MobileNet v2	63%	76.23%
YOLOv4	65%	87.23%

Figure 37 reports the evolution of the F1-score with the variation of the confidence threshold for cross-validation. It is possible to infer that the models behave slightly different. Models with flattened curves indicate higher confidence in their predictions and a low amount of FPs and FNs. Such is the case with the SSD MobileNet v2 model, which despite having a lower F1-Score, is more consistent, as it can maintain essentially the same F1-Score value over the threshold of confidence.

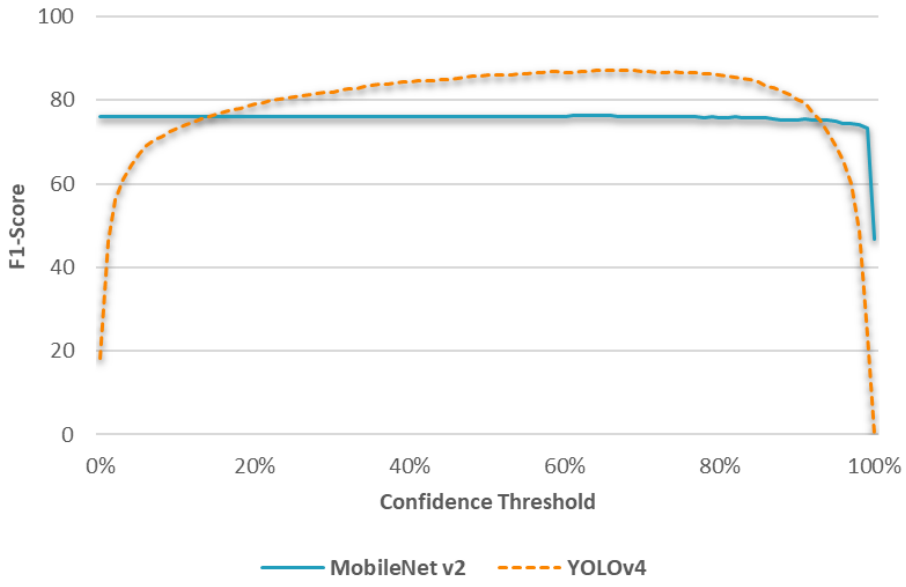
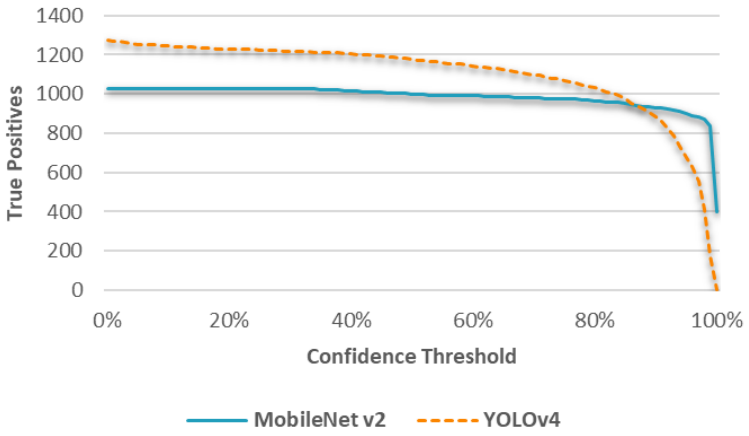
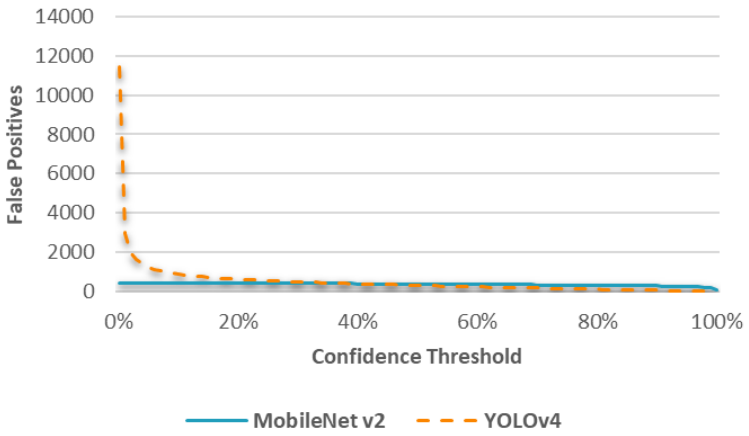


Figure 37 | *Evolution of the F1-score with the variation of the confidence threshold for both DL models (1 class) in the validation set without augmentation.*

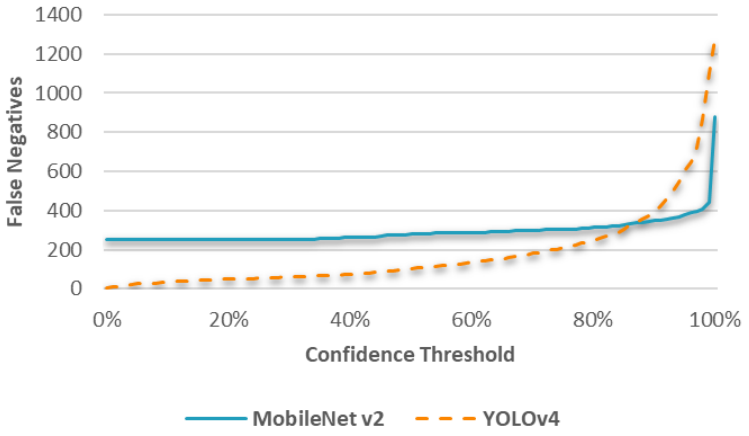
Figure 38 shows the number of TP's, FP's and FN's across the confidence threshold for each model. Once again, it is possible to verify the consistency of the SSD MobileNet v2 model. Although the YOLOv4 model manages to have more TP's and less FN's, with the increase of the confidence threshold these values, tend to vary more than the values obtained by the SSD MobileNet v2 model (Fig. 38 a and Fig. 38 c). It is worth highlighting that the SSD MobileNet v2 model almost had no FP's (Fig. 38 b). This is essential to avoid harvesting non-fruits and consequently damage the plants or the robot itself.



(a)



(b)



(c)

Figure 38 | *Evolution of the number of TP's (a), FP's (b), and FN's (c) in both DL models (1 class) with the increase of the confidence threshold.*

The previous analysis provided the benchmark in the validation set. The results presented below refer to the benchmarking of the test set that allows understanding the generalisation capacity of the trained DL models.

Table 4 shows the results across the different metrics, considering all the predictions and the best-computed confidence threshold. Lower confidence rates tend to have lower Precision but a higher Recall rate. Hence, limiting the confidence threshold can become an advantage. This can be especially verified considering the results obtained by the YOLOv4 model. When the model has full freedom to make predictions (confidence threshold ≥ 0), it presents a Recall close to 100%, but an inferior Precision, only around 10%, drastically affecting the F1-Score. However, by limiting the confidence threshold, the model could obtain a higher Precision without harming the Recall rate too much, thus obtaining an excellent F1-Score.

Table 4 *| Detection results of the DL models (1 class) over the evaluation metrics, considering all the predictions and the best computed confidence threshold.*

DL Model	Confidence Threshold ≥	mAP	Precision	Recall	F1-Score
MobileNet v2	0%	73.07%	74.43%	79.21%	76.75%
YOLOv4	0%	94.48%	11.07%	99.56%	19.92%
MobileNet v2	63%	71.46%	77.66%	77.10%	77.38%
YOLOv4	65%	84.36%	87.92%	86.00%	86.95%

Figure 39 shows a Precision x Recall curve that was built using all the predictions. This curve establishes the compromise between the Recall rate and the Precision rate, with the evolution of the prediction confidence score. The best performing model has the highest AUC [132], therefore the YOLOv4 model. However, the low Precision at higher Recall rates and the lower F1-Score indicates that the model has much prediction noise and false positives. Thus, considering all the model predictions, using the F1-score as a balanced metric between the Recall and Precision, SSD MobileNet v2 was the best performing model, with an F1-Score of 76.75%.

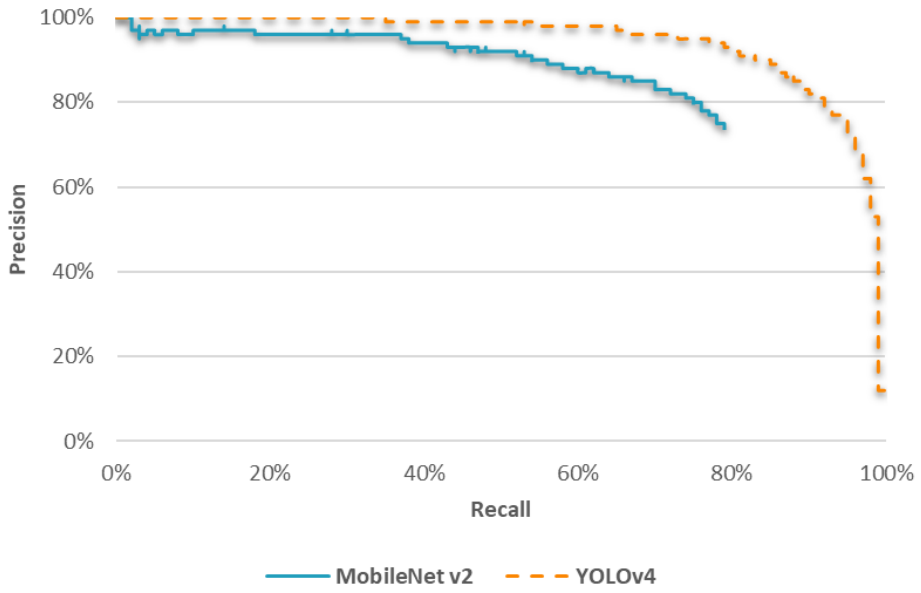


Figure 39 | *Precision × Recall curve for both DL models (1 class) in the test set considering all the predictions.*

Performing an additional filtering process on the predictions, the Precision increased considering the best-computed confidence threshold. The Recall x Precision curve was now transformed through a truncation process (Fig. 40). Both models had a Precision rate higher than 75%, with the YOLOv4 model almost achieving 88%. Recall and Precision rates were similar for both models, which shows that the models are well balanced. Also, the models had a high confidence rate in their predictions, with the YOLOv4 model standing out, reaching an F1-Score close to 87%, meaning that it possesses the ability to detect almost all groundtruths, without neglecting Precision, i.e. having few FP's.

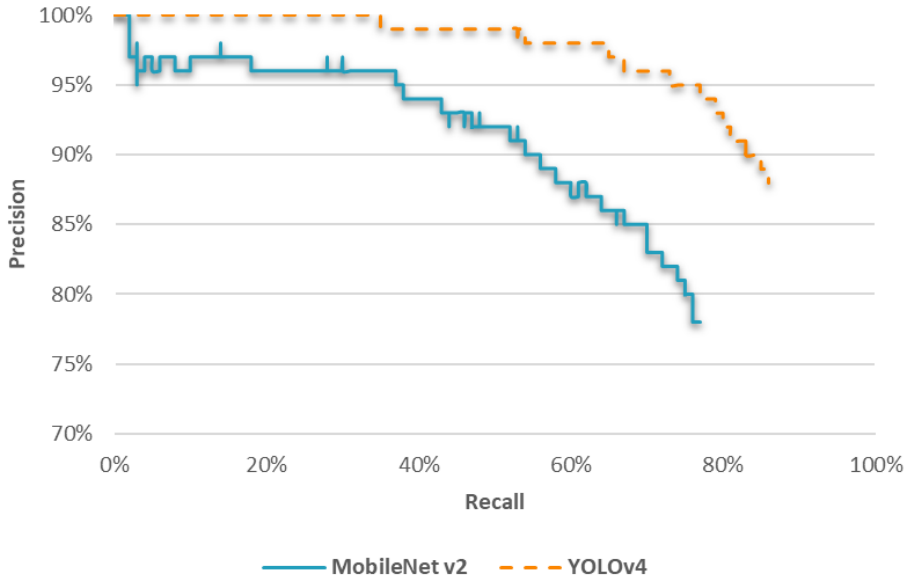


Figure 40 | *Precision × Recall curve for both DL models (1 class) in the test set using the calibrated confidence threshold.*

Overall, both models were generic enough to characterise the class tomato to detect all the tomatoes successfully. The results were similar between the validation set and the test set, with the YOLOv4 model obtaining promising results, being the best performing model. Interestingly, the use of filtered results by a threshold was only significant to the YOLOv4 model. The SSD MobileNet v2 model obtained identical results regardless of the confidence threshold, which means that it can be used without any filtering process without compromising the results. This can be clearly understood from Figure 41. In the YOLOv4 model, using all predictions resulted in many FP's.

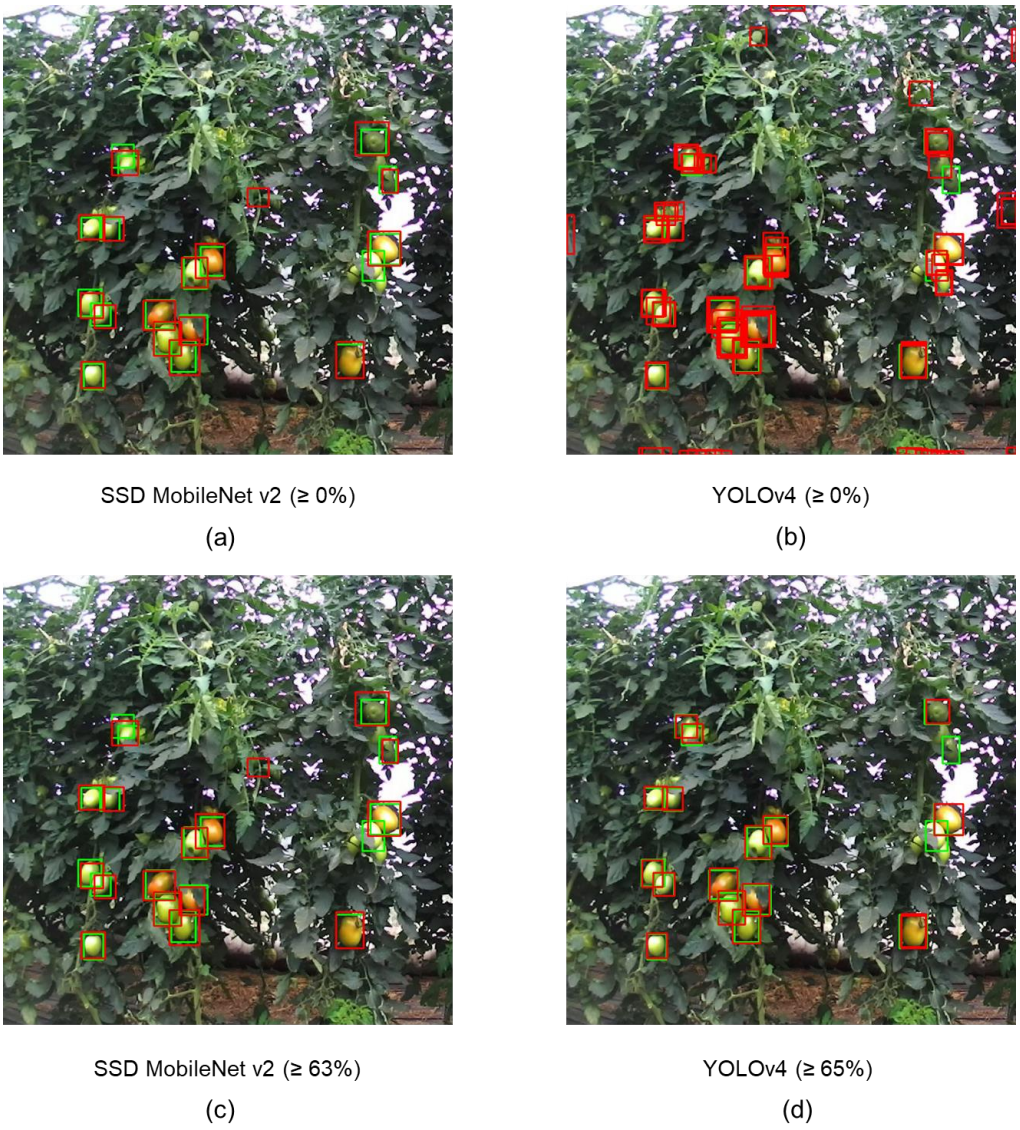


Figure 41 | *Comparison between using unfiltered images (a and b) and filtered images through the best confidence threshold (c and d) for the DL models (1 class). Green bounding boxes = groundtruth annotations; Red bounding boxes = model detections.*

Additionally, to better understand the capabilities of the models, situations of darkened and occluded/overlapped tomatoes were analysed, considering representative images from the dataset as presented in Figure 42. In poor lighting situations, both models performed well, showing robustness and capability to deal successfully with problems posed by different lighting conditions. In cases where tomatoes are occluded by branches, stems, leaves or other tomatoes (overlapped), the models showed a great performance. An interesting detail is that, in both situations, the models were able to detect tomatoes that were not annotated as groundtruth correctly. This image analysis

is important because cases like these show that the models can be better than what the results indicate. Therefore re-annotating the dataset can be an advantage.

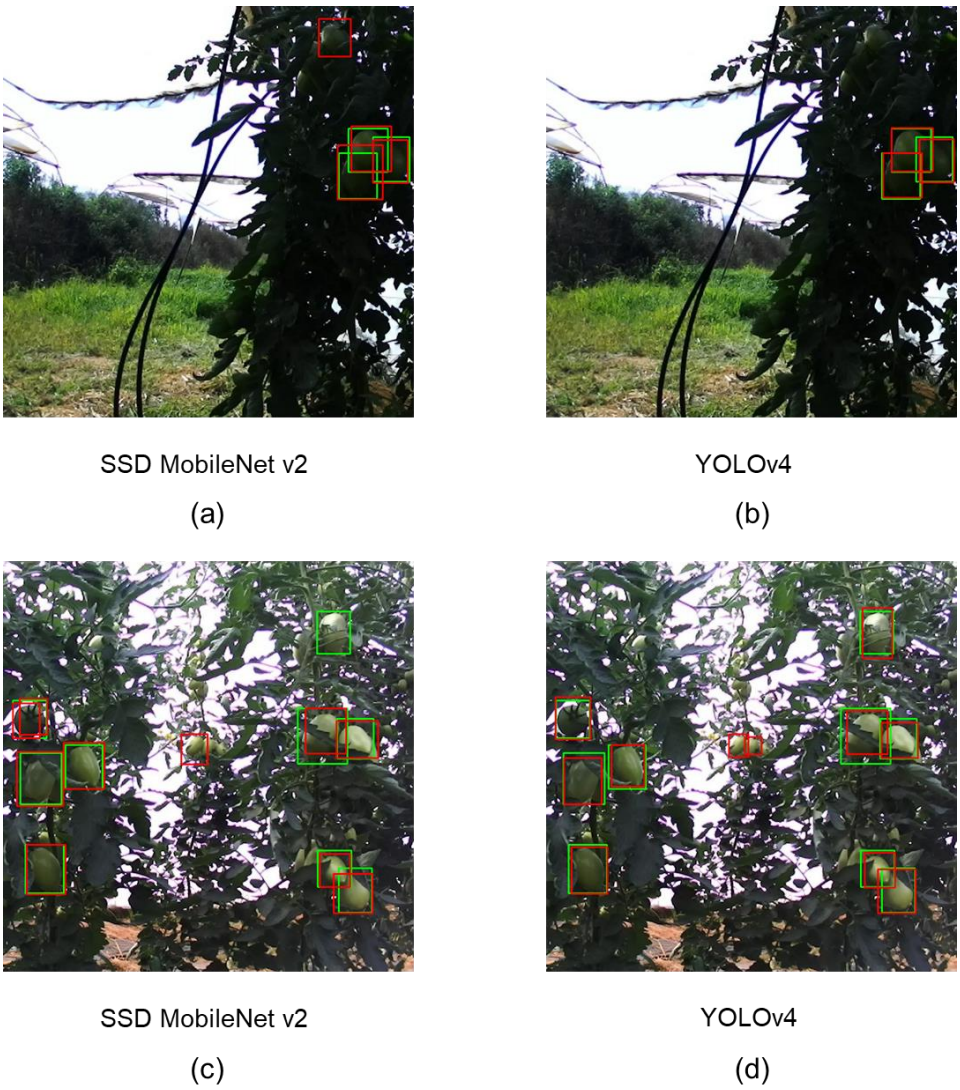


Figure 42 | *Result comparison for darkened (a and b) and occluded/overlaped images (c and d) for the DL models detection (1 class). Green bounding boxes = groundtruth annotations; Red bounding boxes = model detections.*

As mentioned earlier, the use of DL models for tomato detection has grown. Comparing these results with different authors is essential to understand the relevance of the results obtained and potential aspects that could be improved. However, a robust comparison is often hindered due to several of factors such as: i) Most methodologies are applied to the detection of riped tomatoes, which is somewhat easier due to the higher colour contrast between the fruit and the background, thus leading to better results; ii) even though they are targeted for greenhouse crops, in some research

studies, the DL models are trained and evaluated with a set of images taken in an artificial environment, with solid and stable background; iii) how these models are evaluated is not standardised, numerous metrics may vary from one paper to another. Most papers lack the F1-Score, often presenting the accuracy (same as Recall) or the AP/mAP of each model. Table 5 compiles the results of other authors, indicating the DL models used and considering the context in which they were applied: type of environment and fruit ripeness.

Table 5 | *Results of different papers regarding single class tomato detection through DL models.*

DL Model	Method	Results	Author
R-CNN with VGG16	Different ripeness stages Artificial environment	Recall: 19.48%	de Luna, Dadios [103]
Mask-RCNN with ResNet50 and ResNet101	Different ripeness stages Artificial environment	mAP: 90.13% and 93.30% (ResNet50 and ResNet101)	Lee, Nazki [167]
Faster R-CNN with Resnet 101	Green/Turning ripeness stages Greenhouse environment	F1-Score: 83.67%	Mu, Chen [102]
4 SSD and 1 YOLO models	Different ripeness stages Greenhouse environment	SSD MobileNet v2 (Best) F1-Score: 66.15 %	Magalhães, Castro [116]
3 SSD and 1 YOLO models	Different ripeness stages Greenhouse environment	YOLOv4 (Best) F1-Score: 61.16%	Padilha, Moreira [109]
3 SSD models	Different ripeness stages Greenhouse environment	SSD Inception v2 (Best) AP: 98.85%	Yuan, Lv [21]
Improved YOLOv3	Different ripeness stages Greenhouse environment	F1-Score: 94.18%	Chen, Wang [170]
Improved YOLOv3	Different ripeness stages Greenhouse environment	mAP: 76.90%	Zhang, Chen [172]
YOLO-Tomato	Different ripeness stages Greenhouse environment	F1-Score: 93.91%	Liu, Nouaze [100]
TD-Net	Different ripeness stages Greenhouse environment	AP: 81.64%	Zhou, Xu [166]
Improved DenseNet	Red ripeness stage Greenhouse environment	Recall: 91.26%	Liu, Pi [169]
Improved YOLOv3 Tiny	Red ripeness stage Greenhouse environment	F1-Score: 91.92%	Xu, Jia [171]

As they are one-stage detectors, the SSD MobileNetv2 and YOLOv4 models, although faster at detecting, could lack accuracy. This is not overly verified when compared to two-stage detection frameworks. Both models obtained an Recall well above the R-CNN used by de Luna, Dadios [103] and the YOLOv4 model achieved a higher F1-Score than the Faster R-CNN with ResNet 101 model implemented by Mu, Chen [102]. The Mask RCNN models with ResNet50 and ResNet101 used by Lee, Nazki [167] obtained a higher mAP of 90.13% and 93.30%, respectively. However it is still important to note that the models were evaluated in a stable environment where the fruits were detached from the plant.

As mentioned before, along with the DL models presented in this dissertation, other one-stage detection frameworks were trained – SSD Mobilenet v2, SSD Inception v2, SSD ResNet50, SSD ResNet101, YOLOv4 and YOLOv4 Tiny – and their benchmark can be accessed in the papers mentioned in the Additional Contributions section [109, 116]. Both the SSD MobileNet v2 and YOLOv4 models outperformed all the frameworks used on those studies, which obtained quite high Precision rates but ended up failing in their ability to detect all relevant objects, causing the overall F1-Score to vary only between 50-60%.

To detect cherry tomatoes, Yuan, Lv [21] evaluated 3 SSD models and the SSD Inception v2 model obtained an almost flawless AP of 98.85%. Still, in this case, the annotation was done by tomato clusters and not for each fruit, which may facilitate the detection and, consequently, high results. Zhou, Xu [168] developed TD-Net, a model derived from Fast R-CNN, and obtained a PA of 81.64%, still lower than that obtained by the YOLOv4 model. Liu, Pi [169] improved the DenseNet model in detecting riped tomatoes in different datasets. The dataset variation affected the Recall values, which found its best at 91.26% but its worst at 59.78%.

It is worth noting that many papers report the modification and improvement of the YOLOv3 model [100, 170-172]. The YOLOv3 model improved by Zhang, Chen [172] obtained a mAP of 76.90%, below the YOLOv4 model. Still, the improvement performed by the remaining authors led to an F1-Score always higher than 90%, even when detecting tomatoes with different ripeness stages [100, 170, 171]. These results show that improving and specifying the state-of-the art DL models for a particular task can be a great advantage and lead to substantially better results.

Based on the literature reviewed, the SSD MobileNet v2 and YOLOv4 model results are still promising, taking into account that they were not fed with a well-balanced dataset, were lightly modified and were trained to detect any type of tomato in a real greenhouse environment.

4.3 Multi-Class Tomato Detection

In order to classify the tomatoes according to their ripeness, the DL models were trained with the 4 classes defined. First, the models were evaluated considering the detection problem to understand the difference between training with 1 or 4 classes.

Through the cross-validation technique, the best confidence threshold was defined, as shown in Table 6. The F1-Score values were similar to those of the models trained with 1 class, but with a lower optimal confidence threshold, which means that the models are less confident in their predictions.

Table 6 | *Confidence threshold for each DL model (4 classes) that optimises the F1-score metric.*

DL Model	Confidence Threshold ≥	F1-Score
SSD MobileNet v2	34%	76.25%
YOLOv4	52%	86.15%

The graphs obtained from the F1-Score along the confidence threshold (Fig. 43) were identical to the ones previously obtained. The SSD MobileNet v2 was again the best model, being more constant regarding the F1-Score value.

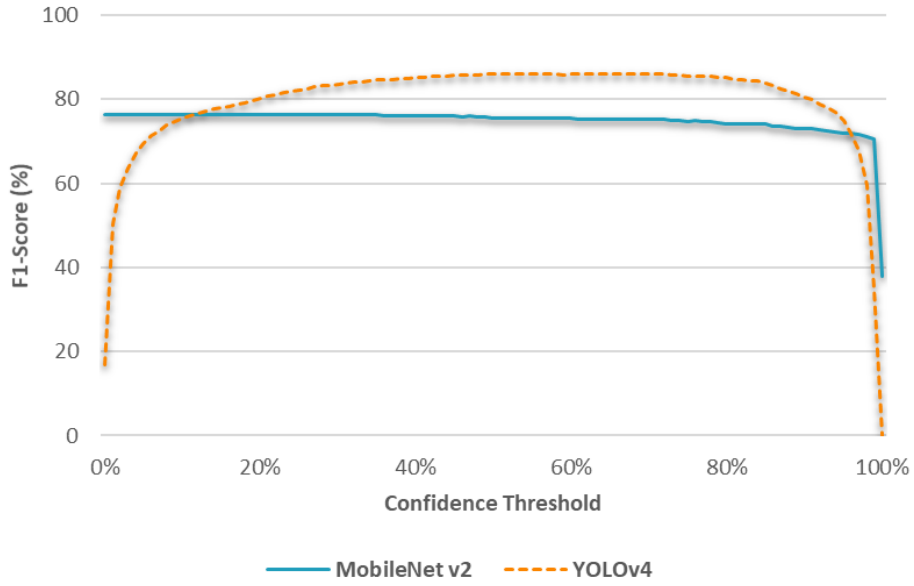
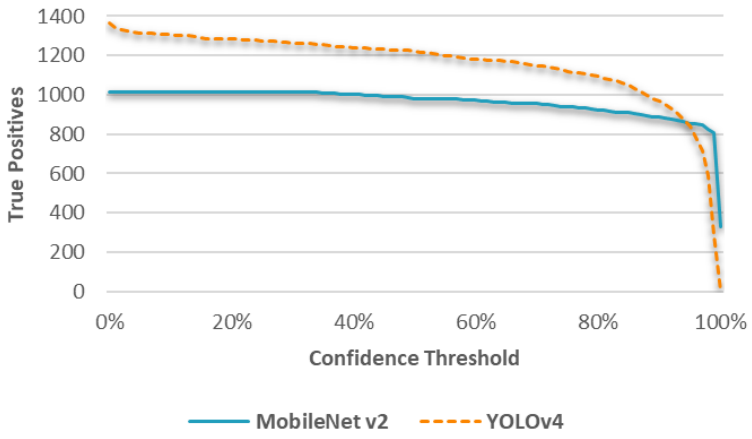


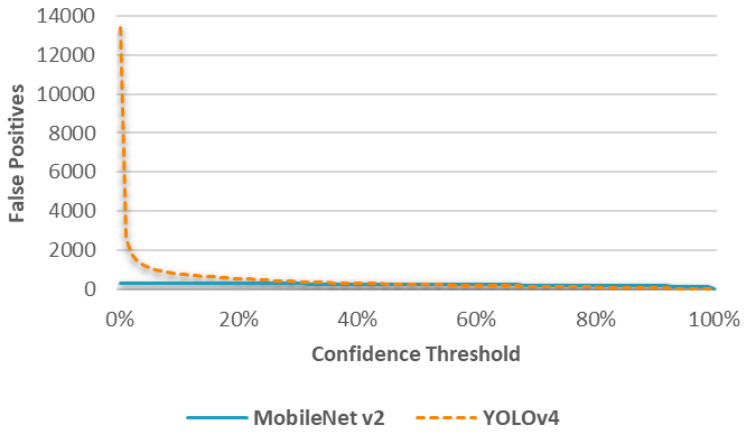
Figure 43 | *Evolution of the F1-score with the variation of the confidence threshold for both DL models (4 classes) in the validation set without augmentation.*

As shown in Figure 44, the variation of TP's, FP's and FN's was also similar: very low amount of FP's (Fig. 44 b) regarding the SSD MobileNet v2 model, in contrast to the higher, but more inconstant, TP's (Fig. 44 a) and FN's (Fig. 44 c) of the YOLOv4 model.

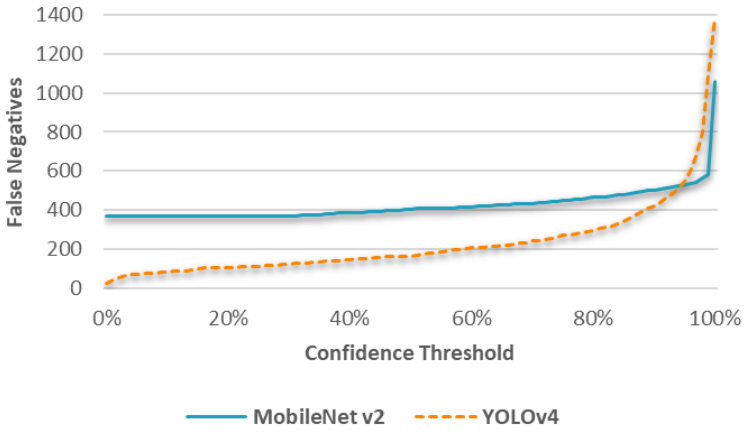


(a)

(Cont.)



(b)



(c)

Figure 44 *| Evolution of the number of TP's (a), FP's (b), and FN's (c) in both DL models (4 classes) with the increase of the confidence threshold.*

As shown in Table 7, limiting the confidence threshold led to similar results regarding the F1-Score with the test set.

Table 7 *| Detection results of the DL models (4 classes) over the evaluation metrics, considering all the predictions and the best computed confidence threshold.*

DL Model	Confidence Threshold ≥	mAP	Precision	Recall	F1-Score
SSD MobileNet v2	0%	64.25%	88.26%	66.23%	75.67%
YOLOv4	0%	91.20%	10.62%	96.29%	19.14%
SSD MobileNet v2	34%	63.92%	88.73%	65.85%	75.60%
YOLOv4	52%	80.69%	96.70%	81.01%	88.16%

Considering all predictions, the SSD MobileNet v2 model performed better, showing a higher balance between the Precision and Recall, despite the YOLOv4 model having a higher AUC, as Figure 45 shows.

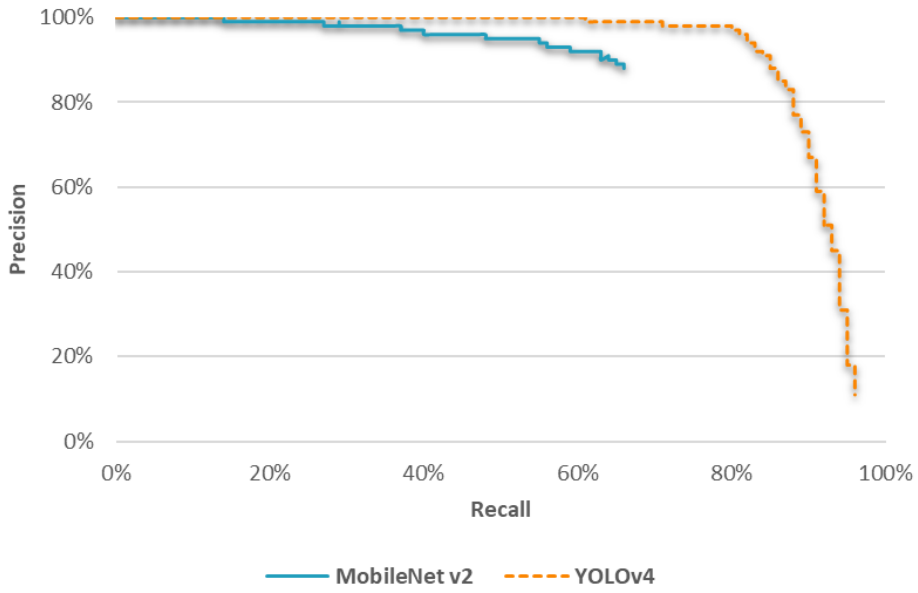


Figure 45 | *Precision × Recall curve for both DL models (4 classes) in the test set considering all the predictions.*

Looking at the best threshold, the YOLOv4 model was again sensitive to the filtering process, becoming the model with the best performance with an F1-Score close to 90%. In both models, all the predictions had a Precision rate higher than 88%, but the Recall rates have fallen, by 15-20%, as illustrated in Figure 46.

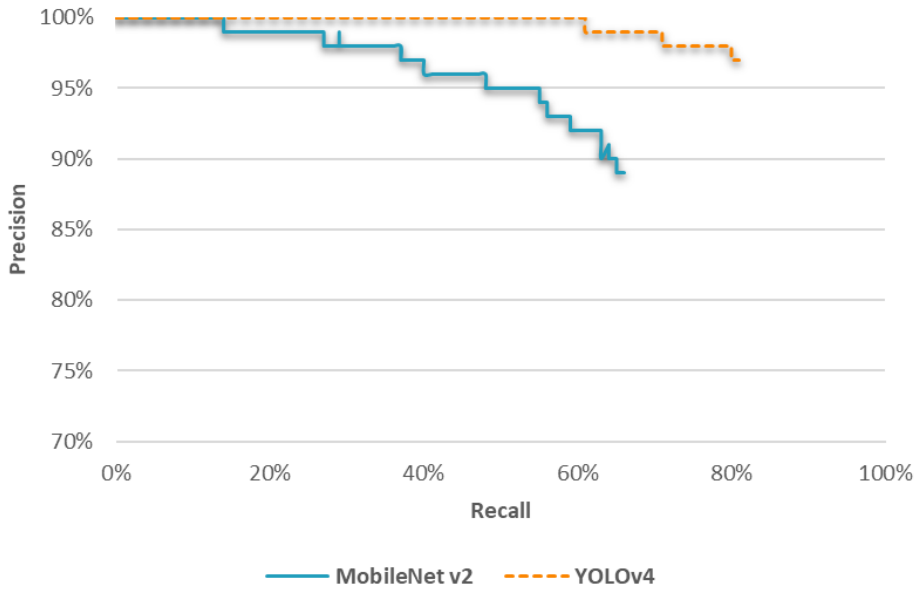


Figure 46 | *Precision × Recall curve for both DL models (4 classes) in the test set using the calibrated confidence threshold.*

In the first impression, the models trained to detect a single tomato class, compared with those trained to detect 4 classes are identical, especially concerning the F1-Score. The critical difference lies in the Recall metric. Models trained with one class are considerably more balanced, presenting identical Precision and Recall values. By having more classes to detect, models need to be more precise, which ultimately affects Recall, i.e. their ability to detect all groundtruths. In practical terms, this can be a disadvantage for a harvesting robot, that besides being precise, it is desirable that it can detect as many tomatoes as possible. This problem is most evident in the SSD MobileNet v2 model, which had a Precision of approximately 89%, but only detected 66% of the tomatoes. While in the YOLOv4 model, the Recall rate was above 80%, but still decreased compared to the same model trained with one class.

Image analysis (Fig. 47) again demonstrates the importance of limiting predictions, especially in the YOLOv4 model. Beyond that, it can be seen that the frameworks can no longer detect as many tomatoes (low Recall rate).

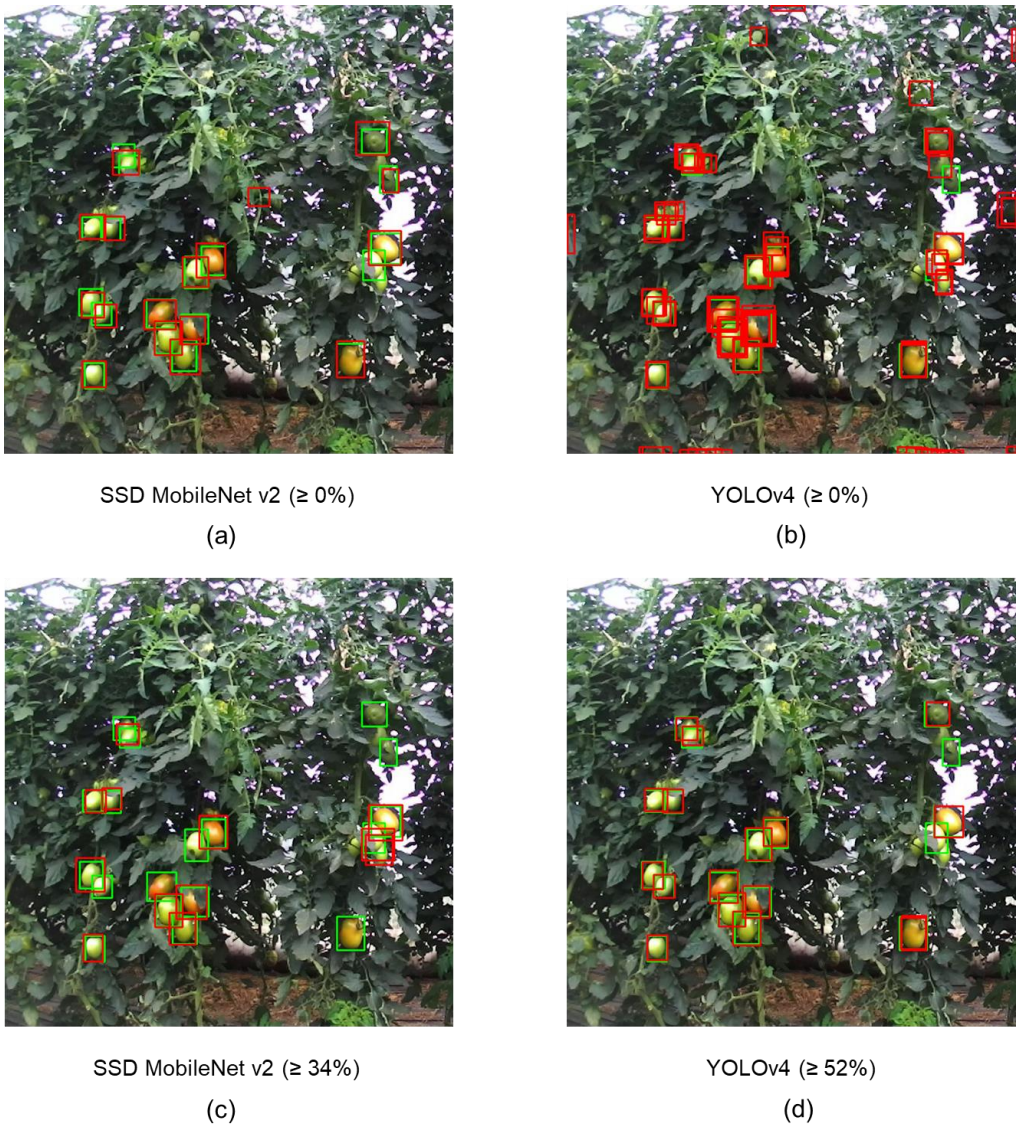


Figure 47 | *Comparison between using unfiltered images (a and b) and filtered images through the best confidence threshold (c and d) for the DL models (4 class). Green bounding boxes = groundtruth annotations; Red bounding boxes = model detections.*

Regarding occlusion/overlap and poor lighting situations, both models, achieved a solid performance despite being trained with more classes and their lower Recall rate (Fig. 48).

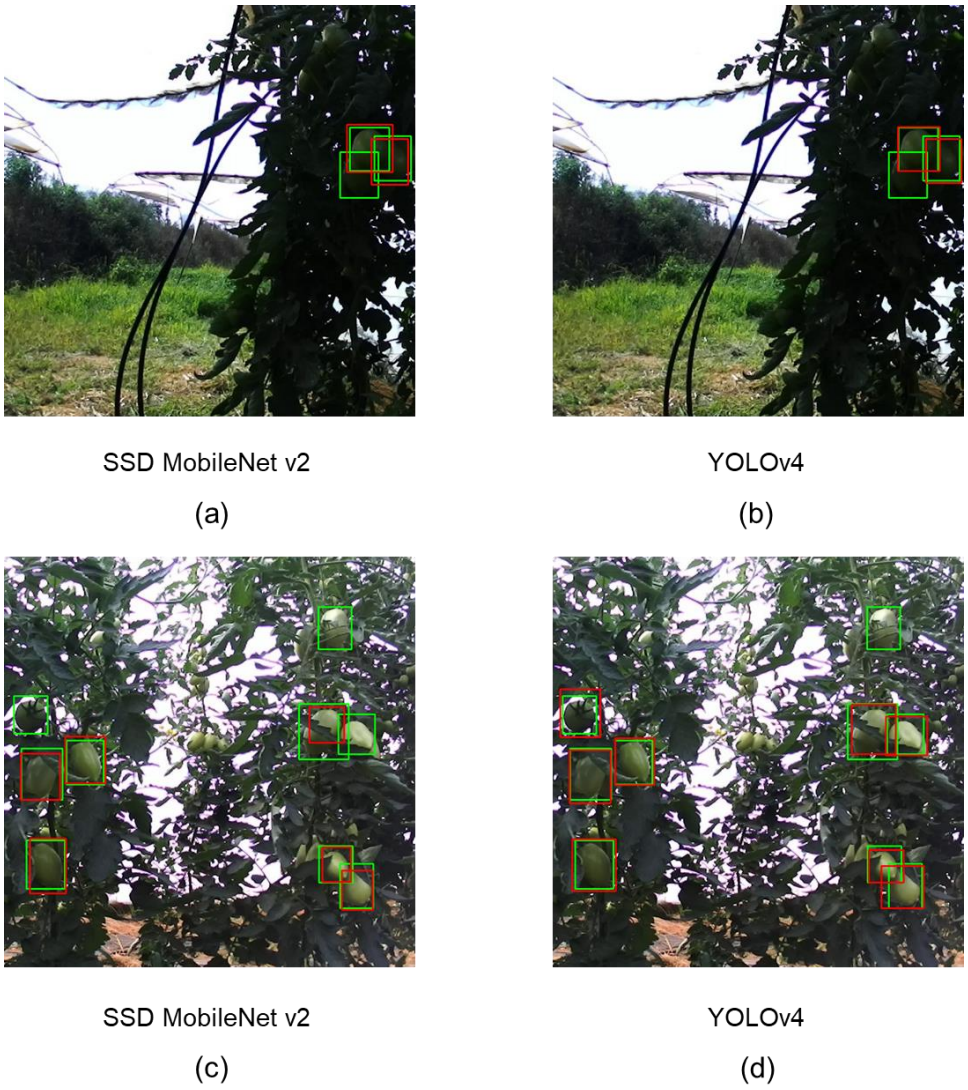


Figure 48 | *Result comparison for darkened (a and b) and occluded/overlaped images (c and d) for the DL models detection (4 class). Green bounding boxes = groundtruth annotations; Red bounding boxes = model detections.*

The results obtained are still interesting, especially considering that the models were trained to detect 4 classes. Most research works focuses on detecting 1 class and when several classes are employed, only 2 are usually considered just to distinguish unriped from riped tomatoes. Another major gap is that when analysing DL models trained with multiple classes, most authors rarely decompose their evaluation into a detection and classification problem, looking only at one of these two, mainly to the classification problem. Table 8 presents the results of other studies considering the DL models, the number of classes and the environment where they were applied.

Table 8 | *Results of different papers regarding multi-class tomato detection through DL models.*

DL Model	Method	Results	Author
Mask-RCNN with ResNet50, ResNet101 and ResNext	2 classes Greenhouse environment	F1-Score: 80.00%	Afonso, Fonteijn [173]
Different YOLO architectures	2 classes Greenhouse environment	YOLOv4 (Best) F1-Score: 66.00%	Ruparelia, Jethva [174]
Modified YOLO-Tomato models	2 classes Greenhouse environment	YOLO-Tomato-C (Best) F1-Score: 97.90%	Lawal [175]
Feature Pyramid Network	3 classes Greenhouse environment	mAP: 99.50%	Sun, He [101]
Different DL models	3 classes Greenhouse environment	RetinaNet (Best) mAP: 74.51%	Tsironis, Bourou [176]

Afonso, Fonteijn [173] used the Mask-RCNN framework with different backbones and the Mask-RCNN with ResNext model obtained the best performance with an F1-Score of 80%, still lower than the one obtained by the YOLOv4 model and only 5% higher than the SSD MobileNetv2 model studied in this dissertation. Both models in this work outperformed the different YOLO architectures Ruparelia, Jethva [174] evaluated in detecting unriped and riped tomatoes, which had YOLOv4 as the best model with an F1-Score of 66%. Still, in the detection of 2 classes, there are studies with excellent results and which surpass the ones presented in this dissertation. Lawal [175] proposed fusing YOLO-Tomato models with different activation functions, achieving an F1-Score of 97.90% through the YOLO-Tomato-C framework. Some studies evaluate DL models in detecting more than two classes. These are the cases of Sun, He [101], who through a proposed Feature Pyramid Network model achieved an mAP of 99.50% in detecting flowers, green and red tomatoes and Tsironis, Bourou [176] who created a dataset with 3 classes (unriped, semi-riped and fully-riped) and evaluated different DL models, but only two outperformed the SSD MobileNet v2 model and none outperformed the YOLOv4 model, with the best result being obtained by the RetinaNet model with a mAP of 74.51%.

Thus, considering the existing literature, especially the YOLOv4 model obtained excellent results considering that it was trained to detect 4 tomato classes, something not often studied.

4.4 Tomato Classification Based on Ripeness Stage

4.4.1 Deep Learning Models Approach

Regarding the classification problem, Figure 49 and 50 represent the confusion matrix for the SSD MobileNet v2 and YOLOv4 models, respectively. As corroborated by the detection problem, the SSD MobileNet v2 model detected fewer tomatoes (1190) than the YOLOv4 model (1323) and, notably, the classification is unbalanced towards the Green class. Nevertheless, SSD MobileNet v2 model had an excellent performance, with a Precision higher than 86% in all classes. The YOLOv4 model had even better results for the Green class, but weaker results for the other classes (≤ 75%), failing even to detect, and therefore classify, any Red tomato.

Looking at the Recall rates, it can be observed that the SSD MobileNet v2 model scored lowest in the Turning class due to some difficulty in distinguishing Green with Turning class tomatoes. The low Recall in the Red class may be misleading and somewhat inaccurate due to the low representativeness of the class, which means that no firm conclusions can be drawn about the performance of the model in classifying tomatoes of this class.

The YOLOv4 model had a slight difficulty distinguishing Turning tomatoes from Light Red ones, but nothing that affects the Recall values to any great extent since it is greater than or equal to 80% for both classes. Both models had nearly the same occurrence of cases where they classified the background (leaves and stems) as an Green tomato, certainly to blame for the large colour correlation between them.

		Groundtruth				
	n = 1190	Unriped	Breaking Stage	Reddish	Riped	N/ Annotated	Precision
Predicted	Unriped	1081	10	2	–	18	97.30%
	Breaking Stage	2	40	3	1	–	86.96%
	Reddish	–	3	28	–	–	90.32%
	Riped	–	–	–	2	–	100%
	Recall	99.82%	75.47%	84.85%	66.67%		

Figure 49 | *Confusion matrix of the SSD MobileNet v2 model, providing the number of predictions made by the model where it classified the classes correctly or incorrectly and the Precision and Recall rates for each class.*

		Groundtruth				
	n = 1323	Green	Turning	Light Red	Red	N/ Annotated	Precision
Predicted	Green	1187	2	–	–	16	98.50%
	Turning	10	48	8	–	–	72.73%
	Light Red	–	10	39	3	–	75.00%
	Red	–	–	–	0	–	0.00%
	Recall	99.16%	80.00%	82.98%	0.00%		

Figure 50 | *Confusion matrix of the YOLOv4 model, providing the number of predictions made by the model where it classified the classes correctly or incorrectly and the Precision and Recall rates for each class.*

Table 9 provides a general and better understanding of the confusion matrix, through Macro F1-Score and Balanced Accuracy metrics. The Macro F1-Score metric implies that the most extensive classes have the same importance as small ones have. Thus, high Macro-F1 values indicate that the algorithm has good performance on all the classes, whereas low Macro F1-Score values refer to poorly predicted classes [120]. SSD MobileNet v2 model outperformed the YOLOv4 model with a Macro F1-Score of 87.27%. The low value obtained by the YOLOv4 model was highly affected by its inability to classify any Red tomato.

Regarding the Balanced Accuracy, smaller classes eventually have a more than a proportional influence on the formula, although their size is reduced in terms of the number of units. The SSD MobileNet v2 model was much better at classifying the fruits, with 81.70% Balanced Accuracy. Although it succeeded in detecting more tomatoes, the YOLOv4 model had considerable difficulties when it comes to classifying, with a Balanced Accuracy of only 65.54%, again affected by the Red class.

Table 9 | *Classification results of the DL models (4 classes) over the evaluation metrics, considering the best computed confidence threshold.*

Model	Confidence Threshold	Macro F1-Score	Balanced Accuracy
SSD MobileNet v2	34%	87.27%	81.70%
YOLOv4	52%	63.37%	65.54%

The sorting of fruits based on their ripeness stage is an operation much more associated with post-harvest. For this reason, in the overwhelming majority of studies, especially on DL models, the classification of fruits is done in a structured environment. This is similar to that found in processing industries after the fruits are harvested, which in a certain way makes this work groundbreaking, since it was carried out with images of a real environment in a greenhouse. Another critical factor refers to the metrics used to evaluate the models. Most authors only use the accuracy (average Recall of each class), which may not reflect the true quality of the models. Table 10 presents some of the works carried out in the tomato classification scope using DL models, concerning the number of classes, type of environment and results obtained.

Table 10 | *Results of different papers regarding tomato classification describing the DL models and methodology used.*

DL Model	No. Classes	Results	Author
ANN	3 classes	Accuracy: 99.32%	de Luna, Dadios [103]
Proposed classification model with CNN	5 classes	Accuracy: 91.90%	Zhang, Jia [177]
Proposed classification model with CNN	2 classes	Accuracy: 98.78%	Toon, Zakaria [178]
VGG16, VGG19, and ResNet101	3 classes	VGG19 (Best) Accuracy: 94.14%	Huynh, Vo [179]
AlexNet	3 classes	Accuracy: 100%	Das, Yadav [180]
CNN and YOLOv3	3 classes	YOLOv3 (Best) Accuracy: 94.67%	Mutha, Shah [181]
SDF-ConvNets	5 classes	F1-Score: 96.50%	Ko, Jang [182]

de Luna, Dadios [103] used an ANN to classify tomatoes into 3 distinct classes and obtained an accuracy of 99.32%. Both Zhang, Jia [177] and Toon, Zakaria [178] proposed CNN-based models to classify tomatoes into 5 and 2 ripeness classes, respectively, achieving an accuracy of 91.90% and 98.78%. Huynh, Vo [179] tested different CNN, with VGG19 getting the best result with an accuracy of 94.14% in classifying tomatoes into 3 ripeness classes. Also, for 3 classes, the AlexNet model evaluated by Das, Yadav [180] and the YOLOv3 model of Mutha, Shah [181], obtained an accuracy of 100% and 94.67%, respectively. Ko, Jang [182] developed a novel model called SDF-ConvNets that relies on multiple streams of CNN and stochastic decision fusion methodology, obtaining an F1-Score of 96.50%.

All the studies mentioned obtained results far superior to those obtained by the DL models studied in this dissertation. Nevertheless, considering that they were trained with images from a greenhouse environment and that the dataset used is unbalanced, the results can certainly be improved. For instance, both models obtained excellent results in the Green tomato classification (Recall higher than 99%), so one can get identical results for the remaining classes by balancing the dataset.

4.4.2 HSV Colour Space Model Approach

Based on the Hue histogram mean of each sample used to build the model and its correlation with the respective class (Appendix D), a quadratic function was obtained as the statistical classifier (Figure 51).

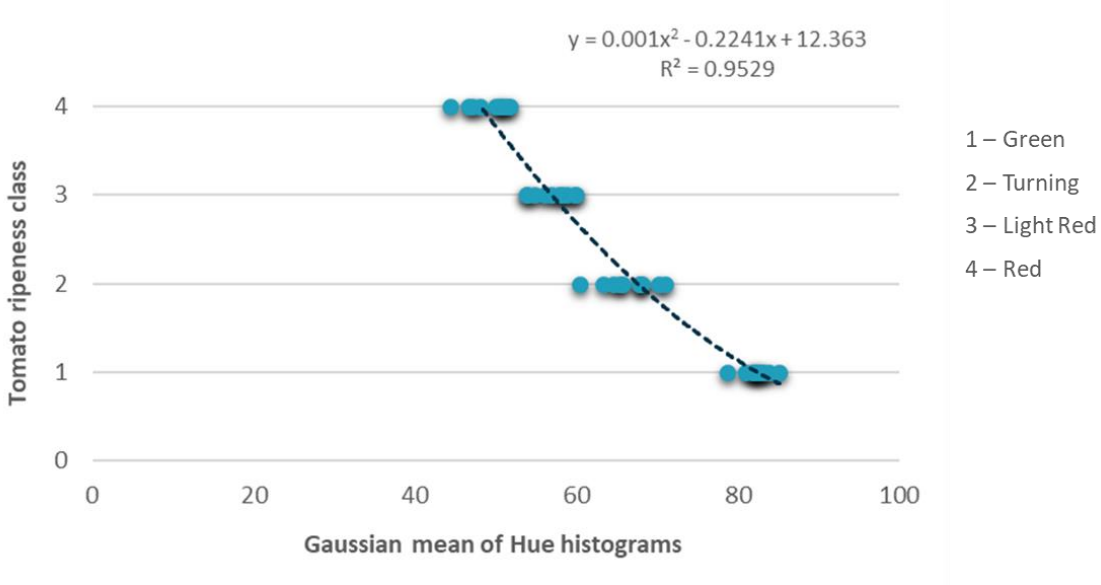


Figure 51 | *Correlation between the Hue histograma Gaussian mean of each sample with its respective class, along with the plot of the tendency line, equation and R^2 of the quadratic function obtain.*

In order to classify the tomatoes, it was necessary to define the thresholds for each class. By fine-tuning the equation, namely adding 0.25 to the independent term, the following was established:

- Green:
  - $y \leq 1.5$
- Turning:
  - $1.5 < y \leq 2.5$
- Light Red:
  - $2.5 < y \leq 3.5$
- Red:
  - $y > 3.5$

All groundtruths in the AgRob's test set were used to benchmark the HSV colour space model. Figure 52 shows the model's confusion matrix and the Precision and Recall rates for each class. The model achieved interesting results, beating the DL models in the classification of Green tomatoes and showing great Precision in the classification of Light Red tomatoes. The Recall values of the Light Red class and the Precision values of the Turning class were affected by the difficulty of the model to distinguish tomatoes from these two classes since roughly one-third of Light Red tomatoes were classified as Turning.

		Groundtruth				
	n = 1590	**Green**	**Turning**	**Light Red**	**Red**	**Precision**
Predicted	**Green**	1470	6	–	–	99.59%
	Turning	2	50	19	–	70.42%
	Light Red	–	4	32	1	86.49%
	Red	–	2	1	3	50%
	Recall	99.86%	80.65%	61.54%	75.00%	

Figure 52 | *Confusion matrix of the HSV Colour Space model, providing the number of predictions made by the model where it classified the classes correctly or incorrectly and the Precision and Recall rates for each of the classes.*

It is possible to verify from Table 11 that, as far as the Macro F1-Score is concerned, the model performed better than the YOLOv4 model and only lagged behind the SSD MobileNet v2 model by around 10%, mainly due to the difficulty mentioned above. The value of the Balanced Accuracy confirms that the HSV Colour Space model outperformed the YOLOv4 model, achieving a similar result to the SSD MobileNet v2 model.

Table 11 | *Classification results of the HSV Colour Space Model over the evaluation metrics.*

Model	Macro F1-Score	Balanced Accuracy
HSV Color Histogram Model	77.92%	79.26%

Overall, the YOLOv4 model, despite being able to detect more tomatoes, fell significantly short in the classification. In contrast, the SSD MobileNet v2 model had identical results to the proposed HSV colour space model, especially concerning Balanced Accuracy.

Before the emergence and application of DL models, the classification of fruits according to their ripeness stage was already studied through more elementary image processing techniques. Most of these techniques rely on the analysis and thresholding of several colour spaces. However, similarly to DL models, the studies very rarely consider the problems imposed by the agricultural environment. Besides being done in a controlled environment, to obtain better results are often applied segmentation techniques and algorithms to remove the background. In addition, testing is often done with a small number of samples. Table 12 presents some of these studies, indicating the techniques and number of classes as well as the results obtained.

Table 12 | *Results of different papers regarding tomato classification describing the colour-based models and methodology used.*

Model	No. Classes	Results	Author
Aggregated percent surface area below certain Hue angles	6 classes	Accuracy: 77.00%	Choi, Lee [183]
HSV colour histogram matching	5 classes	Accuracy: 97.20%	Li, Cao [184]
K-Nearest Neighbour based on GLCM and HSV colour space	5 classes	Accuracy: 100%	Indriani, Kusuma [185]
Fuzzy Rule-Based classification based on RGB colour space	6 classes	Accuracy: 94.29%	Goel and Sehgal [186]
YCbCr colour histogram	6 classes	Accuracy: 98.00%	Rupanagudi, Ranjani [187]
Multiplication of V and Cb colour channel using Otsu thresholding	6 classes	Mean Square Error: 3.14	Sari, Adinugroho [188]

The proposed HSV colour space model achieved better results than the approach by Choi, Lee [183], which, through the aggregated percent surface area below certain Hue angles, achieved an accuracy of 77% when classifying tomatoes in 6 different ripening stages. Using the HSV colour space, Li, Cao [184] applied a dominant colour histogram

matching method, achieving 97.20% accuracy. Indriani, Kusuma [185] combined the colour space with Gray Level Co-occurrence Matrix and K-Nearest Neighbour classification which led to a state-of-the art accuracy of 100%. Goel and Sehgal [186] applied Fuzzy Rule-Based classification through the RGB colour space obtaining an accuracy of 94.29%. Rupanagudi, Ranjani [187] classified tomatoes into 6 classes with an accuracy of 98.00% through histograms. Also through the YCbCr colour space, with addition of the YUV colour space, Sari, Adinugroho [188] multiplied the Cb and V channels and achieved a Mean Square Error of only 3.14 through the Otsu segmentation algorithm.

4.5 Brix Degree Prediction

Appendix E displays the results obtained from the SSC measurement for the 60 samples collected. Brixº is one of the important fruit quality indicators and measures the total SSC present in the fruit, mainly organic sugars. It is often assumed that SSC increases as the fruits mature, but this is not always the case. As Dai, Wu [189] study indicates, unlike other fruits such as grapes, tomatoes have small sugar accumulation fluctuations and even decrease in soluble sugar concentration over fruit development. Figure 53 fully corroborates this fact, being possible to observe that SSC percentage does not vary significantly throughout the four classes. Even the Red class, as well as the other classes, on average, presented a lower SSC value than the Turning class.

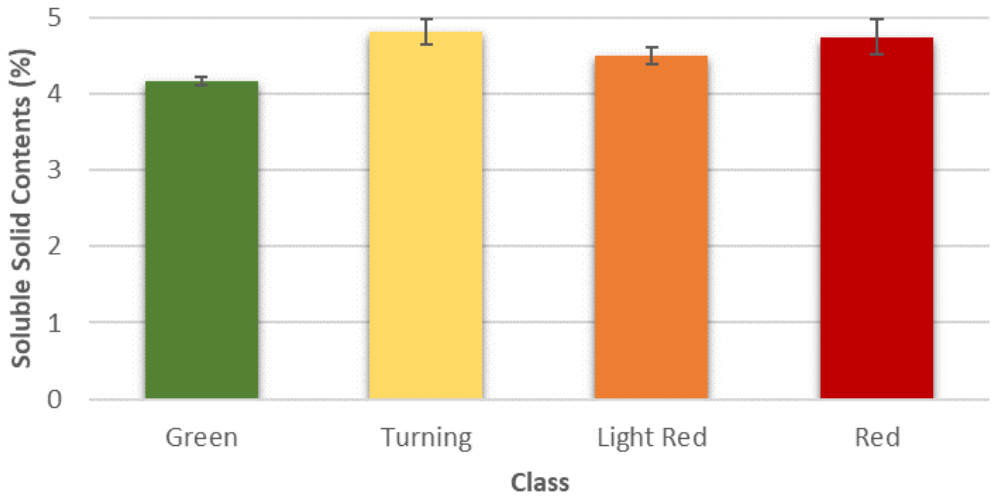


Figure 53 | *Mean value and standard error of the SSC measured for each ripeness class.*

Figure 54 allows a better insight into the subject. Through the Raspberry Dataset samples used for building the HSV Colour Space model and comparing the numerical value assigned to the colour of each sample with its measured SSC, it can be observed that there is no straightforward relationship between colour (ripeness) and SSC. The results are very scattered, most of the Red tomatoes analysed have the same SSC as the Green tomatoes and tomatoes with roughly the same histogram mean have different SSC. The results show that it is not possible to predict Brix by colour information alone.

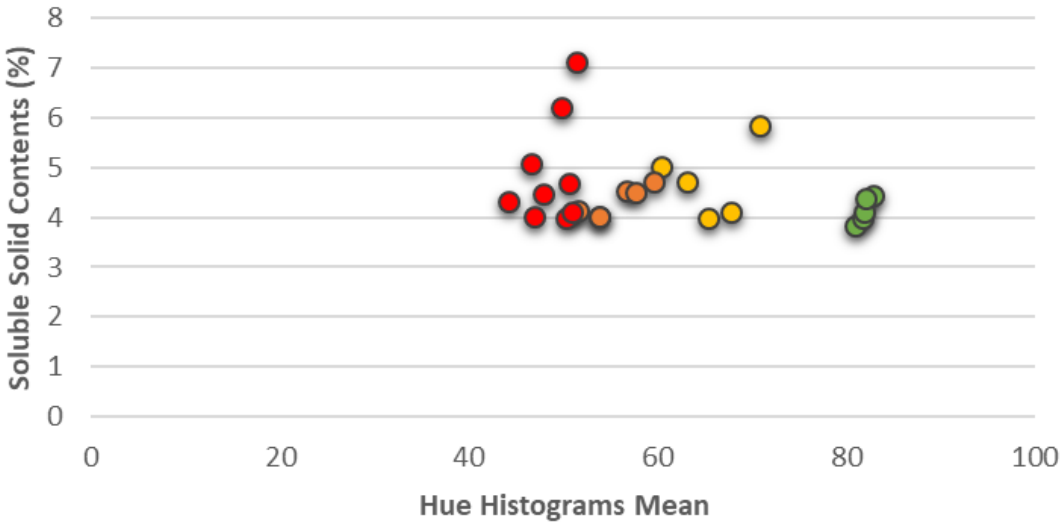


Figure 54 | *Correlation between the Hue histogram average of each sample with its measured SSC. Different colours represent the class to which each sample belongs.*

Conclusions

All the objectives proposed in this dissertation were accomplished. Through the results obtained by the meta-analysis, it is noticeable that there are very few robots developed for protected crops. Most of the reviewed articles are directed to support tasks, mainly about crop detection and classification (67%), where tomato and pepper crops stand out. The great need for research on this topic shows why this is one of the biggest gaps and obstacles in developing robots. More research needs to be done to overcome the immaturity of machine vision and image analysis algorithms.

To answer this problem, two DL models were trained and evaluated for tomato detection and classification. In detecting tomatoes regardless of their ripeness both DL assessed models had a very balanced performance, regarding Precision and Recall rates. The best performing model was the YOLOv4 model, obtaining a strong F1-Score of about 87%. The results also showed that some models, in this case the YOLOv4 model, perform better by tuning the confidence threshold.

In the case of tomato classification according to ripeness based on colour, the models need to be evaluated for their classification and detection performance. Regarding the detection problem, the DL models obtained similar results. The big difference comes with the Recall rate. The fact that they are trained with multiple classes means that the models have to be more specific and accurate, which affects their ability to detect all the groundrtuth tomatoes (Recall). Both models achieved high Precision rates, over 88%, but the Recall rate dropped, especially in the SSD MobileNet v2 model, only 65%. In practical cases, this can be a problem when developing a harvesting robot. A robot that detects a few tomatoes, no matter how precise, will not help and benefit the harvesting operation. Therefore, it can be concluded that training DL models with more than one class can lead to an unbalance performance, particularly concerning their Precision and Recall rates. The more precise a DL model, the less effective it will be in detecting all groundtruths.

When it comes to the ability to classify, the models behaved quite differently. If it was the worst at detecting, the SSD MobileNet v2 model was the best at classifying, achieving a Macro F1-Score of about 87% and a Balanced Accuracy of almost 82%. The YOLOv4 model had disappointing results, having the most difficulty distinguishing between Turning and Light Red tomatoes and was unable to detect any Red tomatoes, getting

results of approximately 63% and 65% on the Macro F1-Score and Balanced Accuracy metrics, respectively. The classification results allow concluding that none of the two models were good at both detecting and classifying tomatoes. The YOLOv4 model correctly detected more tomatoes, but the classification was poor, while the SSD MobileNet v2 model detected fewer tomatoes but was more accurate in classification. The results obtained by the YOLOv4 model in the classification of Red class tomatoes suggest that when dealing with multi-classes, this model needs to be fed with a more balanced dataset, unlike the SSD MobileNet v2 model, which, despite the unbalanced dataset, was able to classify Red tomatoes, even with an Precision rate of 100%.

As an alternative in classification, an HSV Colour Space model was proposed. The model achieved great results in the Green tomatoes classification but struggled to distinguish tomatoes from the Turning class with the Light Red class. Still, it outperformed the YOLOv4 model with distinction, and came close to the SSD MobileNet v2 model, especially regarding the Balanced Accuracy, around 80%. The results become more interesting when one realises that to achieve these results, the DL models had to be trained with a large number of images, specifically 511 images. In contrast, the HSV Colour Space model was developed and trained with only 40 images (10 from each class). On a theoretical basis, increasing the number of images to train and test the HSV Colour Space model, the results could be even better, outperforming both DL models. Another advantage of the proposed model is its simplicity, making it more intuitive and accepted: adjusting the number of classes required and changing the confidence thresholds for each class.

According to the results of the multi-class detection and classification problems, the solution may involve the use of DL models just to detect tomatoes regardless of their ripeness stage, since they are more balanced and show a good compromise between the Precision and Recall rates. The classification task can be performed by the HSV Colour Space model, since it obtained similar results as the DL models, especially regarding the Balanced Accuracy. Getting the best out of each model, by modifying and fusing a DL model with the proposed HSV Colour Space model, it is possible to create a framework capable of detecting a greater number of fruits, and classify them correctly without much loss in both moments.

At the phenotyping level, through the Brix degree data collected, and with the help of the HSV Colour Space model, it is possible to conclude that the SSC of the tomatoes can not be estimated only through their colour information.

In perspective, putative future work should go through:

- Enlarge the dataset, balancing it with more images with Red tomatoes;
- Evaluating the performance of these models in on-time conditions, inside the greenhouses;
- Modify and improve DL models for the detection and classification of tomatoes in specific;
- Improve the HSV Color Space Model by testing with more images;
- Seek to develop techniques to extract other relevant information from the fruit to achieve an automated and differentiated harvest.

Bibliographic References

1. Bechar, A. and M. Eben-Chaime, *Hand-held computers to increase accuracy and productivity in agricultural work study.* International Journal of Productivity and Performance Management, 2014. **63**(2): p. 194-208.DOI: 10.1108/IJPPM-03-2013-0040.
2. Bechar, A. and C. Vigneault, *Agricultural robots for field operations: Concepts and components.* Biosystems Engineering, 2016. **149**: p. 94-111.DOI: https://doi.org/10.1016/j.biosystemseng.2016.06.014.
3. Iida, M., et al., *Advanced Harvesting System by using a Combine Robot.* IFAC Proceedings Volumes, 2013. **46**(4): p. 40-44.DOI: https://doi.org/10.3182/20130327-3-JP-3017.00012.
4. Marinoudi, V., et al., *Robotics and labour in agriculture. A context consideration.* Biosystems Engineering, 2019. **184**: p. 111-121.DOI: https://doi.org/10.1016/j.biosystemseng.2019.06.013.
5. Bechar, A., *Robotics in horticultural field production.* Stewart Postharvest Review, 2010. **6**(3): p. 1-11.DOI: 10.2212/spr.2010.3.11.
6. Manzano-Agugliaro, F. and A. García-Cruz, *Time study techniques applied to labor management in greenhouse tomato (Solanum lycopersicum L.) cultivation.* Agrociencia, 2009. **43**(3): p. 267-277.
7. Fountas, S., et al., *Agricultural Robotics for Field Operations.* 2020. **20**(9): p. 2672.
8. Bac, C.W., et al., *Harvesting Robots for High-Value Crops: State-of-the-Art Review and Challenges Ahead.* Journal of Field Robotics, 2014. **31**.DOI: 10.1002/rob.21525.
9. Kapach, K., et al., *Computer vision for fruit harvesting robots—State of the art and challenges ahead.* International Journal of Computational Vision and Robotics, 2012. **3**: p. 4-34.DOI: 10.1504/IJCVR.2012.046419.
10. Mavridou, E., et al., *Machine Vision Systems in Precision Agriculture for Crop Farming.* Journal of Imaging, 2019. **5**: p. 89.DOI: 10.3390/jimaging5120089.
11. Patrício, D. and R. Rieder, *Computer vision and artificial intelligence in precision agriculture for grain crops: A systematic review.* Computers and Electronics in Agriculture, 2018. **153**: p. 69-81.DOI: 10.1016/j.compag.2018.08.001.

12. Zareiforoush, H., et al., *Potential Applications of Computer Vision in Quality Inspection of Rice: A Review.* Food Engineering Reviews, 2015. **7**(3): p. 321-345.DOI: 10.1007/s12393-014-9101-z.
13. Sharma, A., et al., *Machine Learning Applications for Precision Agriculture: A Comprehensive Review.* IEEE Access, 2020. **PP**: p. 1-1.DOI: 10.1109/ACCESS.2020.3048415.
14. Shanmuganathan, S., *Artificial Neural Network Modelling: An Introduction*. 2016, Springer International Publishing. p. 1-14.DOI: 10.1007/978-3-319-28495-8_1.
15. Schmidhuber, J., *Deep learning in neural networks: An overview.* Neural Networks, 2015. **61**: p. 85-117.DOI: https://doi.org/10.1016/j.neunet.2014.09.003.
16. Kamilaris, A. and F. Prenafeta Boldú, *A review of the use of convolutional neural networks in agriculture.* The Journal of Agricultural Science, 2018. **156**: p. 1-11.DOI: 10.1017/S0021859618000436.
17. Lecun, Y., et al., *Gradient-based learning applied to document recognition.* Proceedings of the IEEE, 1998. **86**(11): p. 2278-2324.DOI: 10.1109/5.726791.
18. Liu, W., et al., *SSD: Single Shot MultiBox Detector*. 2016, Springer International Publishing. p. 21-37.DOI: 10.1007/978-3-319-46448-0_2.
19. Redmon, J., et al., *You Only Look Once: Unified, Real-Time Object Detection.* arXiv pre-print server, 2016.DOI: None. arxiv:1506.02640.
20. Agarwal, S., Jean, and F.e.e. Jurie, *Recent Advances in Object Detection in the Age of Deep Convolutional Neural Networks.* arXiv pre-print server, 2019.DOI: None. arxiv:1809.03193.
21. Yuan, T., et al., *Robust Cherry Tomatoes Detection Algorithm in Greenhouse Scene Based on SSD.* Agriculture, 2020. **10**: p. 160.DOI: 10.3390/agriculture10050160.
22. Bechar, A., et al., *Improvement of work methods in tomato greenhouses using simulation.* Transactions of the ASABE, 2007. **50**(2): p. 331-338.
23. Market Data Forecast. *Greenhouse Horticulture Market*. 2020.
24. Bochtis, D.D., C.G.C. Sørensen, and P. Busato, *Advances in agricultural machinery management: A review.* Biosystems Engineering, 2014. **126**: p. 69-81.DOI: https://doi.org/10.1016/j.biosystemseng.2014.07.012.

25. Eberhardt, M. and D. Vollrath, *The Effect of Agricultural Technology on the Speed of Development.* World Development, 2018. **109**: p. 483-496.DOI: https://doi.org/10.1016/j.worlddev.2016.03.017.
26. Urrea, C. and J. Muñoz, *Path Tracking of Mobile Robot in Crops.* Journal of Intelligent & Robotic Systems, 2015. **80**(2): p. 193-205.DOI: 10.1007/s10846-013-9989-1.
27. Hiremath, S.A., et al., *Laser range finder model for autonomous navigation of a robot in a maize field using a particle filter.* Computers and Electronics in Agriculture, 2014. **100**: p. 41-50.DOI: https://doi.org/10.1016/j.compag.2013.10.005.
28. Nof, S.Y., *Springer Handbook of Automation*. 2009: Springer Publishing Company, Incorporated.
29. Eizicovits, D. and S. Berman, *Efficient sensory-grounded grasp pose quality mapping for gripper design and online grasp planning.* Robotics and Autonomous Systems, 2014. **62**(8): p. 1208-1219.DOI: https://doi.org/10.1016/j.robot.2014.03.011.
30. Tervo, K. and H.N. Koivo, *Adaptation of the Human-Machine Interface to the Human Skill and Dynamic Characteristics.* IFAC Proceedings Volumes, 2014. **47**(3): p. 3539-3544.DOI: https://doi.org/10.3182/20140824-6-ZA-1003.02614.
31. Parasuraman, R., T.B. Sheridan, and C.D. Wickens, *A model for types and levels of human interaction with automation.* IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, 2000. **30**(3): p. 286-297.DOI: 10.1109/3468.844354.
32. van Henten, E.J., et al., *Robotics in protected cultivation.* IFAC Proceedings Volumes, 2013. **46**(18): p. 170-177.DOI: https://doi.org/10.3182/20130828-2-SF-3019.00070.
33. Ng, K.C. and M.M. Trivedi, *A neuro-fuzzy controller for mobile robot navigation and multirobot convoying.* 1998. **28**(6 %J Trans. Sys. Man Cyber. Part B): p. 829–840.DOI: 10.1109/3477.735392.
34. Blackmore, B., et al. *A specification for an autonomous crop production mechanization system*. in *International Symposium on Application of Precision Agriculture for Fruits and Vegetables 824*. 2008.
35. Shamshiri, R., et al., *Research and development in agricultural robotics: A perspective of digital farming.* International Journal of Agricultural and Biological Engineering, 2018. **11**: p. 1-14.DOI: 10.25165/j.ijabe.20181104.4278.

36. Grift, T., et al., *Review of Automation and Robotics for the BioIndustry.* Journal of Biomechatronics Engineering, 2008. **1**.
37. FAOSTAT, *Food and Agriculture Organization of the United Nations*. 2020: http://www.fao.org/faostat/en/#data/QC.
38. Valera, D., et al., *The greenhouses of Almería, Spain: technological analysis and profitability.* Acta Horticulturae, 2017: p. 219-226.DOI: 10.17660/ActaHortic.2017.1170.25.
39. Ferreira, V.S., *A cultura do tomate em estufa. Avaliação das consições climáticas em dois tipos de estufa e sua influência na produtividade e nos custos de produção do tomate na região do Oeste.Relatório de estágio*.Mestrado em Engenharia Agronómica.2017, Universidade de Lisboa, Lisboa.
40. Giovannoni, J., *MOLECULAR BIOLOGY OF FRUIT MATURATION AND RIPENING.* 2001. **52**(1): p. 725-749.DOI: 10.1146/annurev.arplant.52.1.725.
41. Jun, J., et al., *Towards an Efficient Tomato Harvesting Robot: 3D Perception, Manipulation, and End-Effector.* IEEE Access, 2021. **9**: p. 17631-17640.DOI: 10.1109/ACCESS.2021.3052240.
42. Li, Z., et al., *Analysis of Workspace and Kinematics for a Tomato Harvesting Robot.* Intelligent Computation Technology and Automation, International Conference on, 2008. **1**: p. 823-827.DOI: 10.1109/ICICTA.2008.138.
43. Ji, C., et al., *Research on Key Technology of Truss Tomato Harvesting Robot in Greenhouse.* Applied Mechanics and Materials, 2014. **442**: p. 480-486.DOI: 10.4028/www.scientific.net/AMM.442.480.
44. Zhao, Y., et al., *Dual-arm Robot Design and Testing for Harvesting Tomato in Greenhouse*. Vol. 49. 2016. 161-165.DOI: 10.1016/j.ifacol.2016.10.030.
45. Yasukawa, S., et al., *Development of a Tomato Harvesting Robot.* Proceedings of International Conference on Artificial Life and Robotics, 2017. **22**: p. 408-411.DOI: 10.5954/ICAROB.2017.OS22-1.
46. Matsuo, T., et al., *Tomato-Harvesting Robot Competition: Aims and Developed Robot of 6th Competitions.* Proceedings of International Conference on Artificial Life and Robotics, 2021. **26**: p. 397-400.DOI: 10.5954/ICAROB.2021.OS22-2.
47. Taqi, F., et al. *A cherry-tomato harvesting robot*. in *2017 18th International Conference on Advanced Robotics (ICAR)*. 2017.DOI: 10.1109/ICAR.2017.8023650.

48. Wang, L.L., et al., *Development of a tomato harvesting robot used in greenhouse.* International Journal of Agricultural and Biological Engineering, 2017. **10**: p. 140-149.DOI: 10.25165/j.ijabe.20171004.3204.
49. Otsu, N., *A Threshold Selection Method from Gray-Level Histograms.* IEEE Transactions on Systems, Man, and Cybernetics, 1979. **9**(1): p. 62-66.DOI: 10.1109/tsmc.1979.4310076.
50. Tang, Y., et al., *Recognition and Localization Methods for Vision-Based Fruit Picking Robots: A Review.* 2020. **11**(510).DOI: 10.3389/fpls.2020.00510.
51. Hornberg, A., *Front Matter*, in *Handbook of Machine and Computer Vision*, Wiley-VCH, Editor. 2017. p. i-xxvii.DOI: https://doi.org/10.1002/9783527413409.fmatter.
52. Schaeffel, F., *Processing of Information in the Human Visual System*, in *Handbook of Machine and Computer Vision*, Wiley-VCH, Editor. 2017. p. 1-29.DOI: https://doi.org/10.1002/9783527413409.ch1.
53. Zhao, Y., et al., *A review of key techniques of vision-based control for harvesting robot.* Computers and Electronics in Agriculture, 2016. **127**: p. 311-323.DOI: 10.1016/j.compag.2016.06.022.
54. Yang, X., et al., *A Survey on Smart Agriculture: Development Modes, Technologies, and Security and Privacy Challenges.* IEEE/CAA Journal of Automatica Sinica, 2020. **8**: p. 273-302.DOI: 10.1109/JAS.2020.1003536.
55. Evstatiev, B.I. and K.G. Gabrovska-Evstatieva, *A review on the methods for big data analysis in agriculture.* IOP Conference Series: Materials Science and Engineering, 2021. **1032**: p. 012053.DOI: 10.1088/1757-899x/1032/1/012053.
56. Liakos, K., et al., *Machine Learning in Agriculture: A Review.* Sensors, 2018. **18**(8): p. 2674.DOI: 10.3390/s18082674.
57. Samuel, A.L., *Some Studies in Machine Learning Using the Game of Checkers.* IBM Journal of Research and Development, 1959. **3**(3): p. 210-229.DOI: 10.1147/rd.33.0210.
58. Benos, L., et al., *Machine Learning in Agriculture: A Comprehensive Updated Review.* Sensors, 2021. **21**(11): p. 3758.DOI: 10.3390/s21113758.
59. Welling, M. and D. Bren. *A First Encounter with Machine Learning*. 2010.
60. Escamilla-García, A., et al., *Applications of Artificial Neural Networks in Greenhouse Technology and Overview for Smart Agriculture Development.* Applied Sciences, 2020. **10**: p. 3835.DOI: 10.3390/app10113835.

61. Walczak, S. and N. Cerpa, *Artificial Neural Networks*, in *Encyclopedia of Physical Science and Technology (Third Edition)*, R.A. Meyers, Editor. 2003, Academic Press: New York. p. 631-645.DOI: https://doi.org/10.1016/B0-12-227410-5/00837-1.
62. LeCun, Y., Y. Bengio, and G. Hinton, *Deep Learning.* Nature, 2015. **521**: p. 436-44.DOI: 10.1038/nature14539.
63. Najafabadi, M.M., et al., *Deep learning applications and challenges in big data analytics.* 2014. **2**: p. 1-21.
64. Wan, J., et al., *Deep Learning for Content-Based Image Retrieval: A Comprehensive Study*. 2014.
65. Naranjo Torres, J., et al., *A Review of Convolutional Neural Network Applied to Fruit Image Processing.* Applied Sciences, 2020. **10**: p. 3443.DOI: 10.3390/app10103443.
66. Koirala, A., *Deep learning – method overview and review of use for fruit detection and yield estimation https://authors.elsevier.com/a/1YvMhcFCSJKOi.* Computers and Electronics in Agriculture, 2019. **162**.DOI: 10.1016/j.compag.2019.04.017.
67. Zhang, Q., et al., *Applications of Deep Learning for Dense Scenes Analysis in Agriculture: A Review.* Sensors, 2020. **20**(5): p. 1520.DOI: 10.3390/s20051520.
68. Pan, S.J., Q.J.I.T.o.K. Yang, and D. Engineering, *A Survey on Transfer Learning.* 2010. **22**: p. 1345-1359.
69. Krizhevsky, A., I. Sutskever, and G.E. Hinton, *ImageNet classification with deep convolutional neural networks*, in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*. 2012, Curran Associates Inc.: Lake Tahoe, Nevada. p. 1097–1105.
70. Simonyan, K. and A. Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition.* arXiv pre-print server, 2015.DOI: None. arxiv:1409.1556.
71. Szegedy, C., et al., *Going Deeper with Convolutions.* arXiv pre-print server, 2014.DOI: None. arxiv:1409.4842.
72. He, K., et al., *Deep Residual Learning for Image Recognition.* arXiv pre-print server, 2015.DOI: None. arxiv:1512.03385.
73. Andrew, et al., *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.* arXiv pre-print server, 2017.DOI: None. arxiv:1704.04861.

74. Ren, S., et al., *Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.* arXiv pre-print server, 2016.DOI: None

arxiv:1506.01497.

75. Girshick, R. *Fast R-CNN*. in *2015 IEEE International Conference on Computer Vision (ICCV)*. 2015.DOI: 10.1109/ICCV.2015.169.

76. Vasconez, J.P., et al., *Comparison of convolutional neural networks in fruit detection and counting: A comprehensive evaluation.* Computers and Electronics in Agriculture, 2020. **173**: p. 105348.DOI: https://doi.org/10.1016/j.compag.2020.105348.

77. Huang, J., et al., *Speed/accuracy trade-offs for modern convolutional object detectors.* arXiv pre-print server, 2017.DOI: None. arxiv:1611.10012.

78. Lu, Y. and S. Young, *A Survey of Public Datasets for Computer Vision Tasks in Precision Agriculture.* Computers and Electronics in Agriculture, 2020. **178**.DOI: 10.1016/j.compag.2020.105760.

79. Padilla, R., S. Netto, and E. da Silva, *A Survey on Performance Metrics for Object-Detection Algorithms*. 2020.DOI: 10.1109/IWSSIP48289.2020.

80. Kamilaris, A. and F.J.C.E.A. Prenafeta-Boldú, *Deep learning in agriculture: A survey.* 2018. **147**: p. 70-90.

81. Dougherty, E.R., *Optimal Binary Morphological Bandpass Filters Induced by Granulometric Spectral Representation.* Journal of Mathematical Imaging and Vision, 1997. **7**(2): p. 175-192.DOI: 10.1023/A:1008209706862.

82. Yin, H., et al., *Ripe Tomato Recognition and Localization for a Tomato Harvesting Robotic System.* Soft Computing and Pattern Recognition, International Conference of, 2009. **0**: p. 557-562.DOI: 10.1109/SoCPaR.2009.111.

83. Huang, L., S. Yang, and D. He, *Abscission Point Extraction for Ripe Tomato Harvesting Robots.* Intelligent Automation & Soft Computing, 2012. **18**.DOI: 10.1080/10798587.2012.10643285.

84. Zhao, Y., et al., *Robust Tomato Recognition for Robotic Harvesting Using Feature Images Fusion.* Sensors, 2016. **16**: p. 173.DOI: 10.3390/s16020173.

85. Arefi, A., et al., *Recognition and localization of ripen tomato based on machine vision.* Australian Journal of Crop Science, 2011. **5**.

86. Qingchun, F., et al., *Design and test of tomatoes harvesting robot*. 2015. 949-952.DOI: 10.1109/ICInfA.2015.7279423.

87. Zhang, F., *Ripe Tomato Recognition with Computer Vision*. 2015.DOI: 10.2991/iiicec-15.2015.107.
88. Canny, J., *A Computational Approach to Edge Detection.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1986. **PAMI-8**(6): p. 679-698.DOI: 10.1109/TPAMI.1986.4767851.
89. Benavides, M., et al., *Automatic Tomato and Peduncle Location System Based on Computer Vision for Use in Robotized Harvesting.* Applied Sciences, 2020. **10**(17): p. 5887.DOI: 10.3390/app10175887.
90. Gupta, S., S.G. Mazumdar, and M.T. Student, *Sobel Edge Detection Algorithm.* International Journal of Computer Science and Management Research, 2013. **2**(2).
91. Malik, M.H., et al., *Mature Tomato Fruit Detection Algorithm Based on improved HSV and Watershed Algorithm.* IFAC-PapersOnLine, 2018. **51**: p. 431-436.DOI: 10.1016/j.ifacol.2018.08.183.
92. Zhu, A., L. Yang, and Y. Chen, *An FCM-based method to recognize and extract ripe tomato for harvesting robotic system*. 2012. 533-538.DOI: 10.1109/ICAL.2012.6308135.
93. Xiang, R., Y. Ying, and H. Jiang. *Tests of a recognition algorithm for clustered tomatoes based on mathematical morphology*. in *2013 6th International Congress on Image and Signal Processing (CISP)*. 2013.DOI: 10.1109/CISP.2013.6744040.
94. Yamamoto, K., et al., *On Plant Detection of Intact Tomato Fruits Using Image Analysis and Machine Learning Methods.* Sensors (Basel, Switzerland), 2014. **14**: p. 12191-12206.DOI: 10.3390/s140712191.
95. Zhao, Y., et al., *Detecting tomatoes in greenhouse scenes by combining AdaBoost classifier and colour analysis.* Biosystems Engineering, 2016. **148**: p. 127-137.DOI: 10.1016/j.biosystemseng.2016.05.001.
96. Liu, G., S. Mao, and J. Kim, *A Mature-Tomato Detection Algorithm Using Machine Learning and Color Analysis.* Sensors, 2019. **19**: p. 2023.DOI: 10.3390/s19092023.
97. Wu, J., et al., *Automatic Recognition of Ripening Tomatoes by Combining Multi-Feature Fusion with a Bi-Layer Classification Strategy for Harvesting Robots.* Sensors (Basel, Switzerland), 2019. **19**(3): p. 612.DOI: 10.3390/s19030612.
98. Alam Siddiquee, K.N.E., et al., *Detection, quantification and classification of ripened tomatoes: a comparative analysis of image processing and machine*

learning. IET Image Processing, 2020. **14**(11): p. 2442-2456.DOI: 10.1049/iet-ipr.2019.0738.

99. Xu, Z., et al., *Fast Method of Detecting Tomatoes in a Complex Scene for Picking Robots.* 2020. **8**: p. 55289-55299.
100. Liu, G., et al., *YOLO-Tomato: A Robust Algorithm for Tomato Detection based on YOLOv3.* Sensors, 2020. **20**.DOI: 10.3390/s20072145.
101. Sun, J., et al., *Detection of tomato organs based on convolutional neural network under the overlap and occlusion backgrounds.* Machine Vision and Applications, 2020. **31**.DOI: 10.1007/s00138-020-01081-6.
102. Mu, Y., et al., *Intact Detection of Highly Occluded Immature Tomatoes on Plants Using Deep Learning Techniques.* Sensors, 2020. **20**: p. 2984.DOI: 10.3390/s20102984.
103. de Luna, R., et al., *Tomato Growth Stage Monitoring for Smart Farm Using Deep Transfer Learning with Machine Learning-based Maturity Grading.* AGRIVITA Journal of Agricultural Science, 2020. **42**.DOI: 10.17503/agrivita.v42i1.2499.
104. Falagas, M.E., et al., *Comparison of PubMed, Scopus, Web of Science, and Google Scholar: strengths and weaknesses.* The FASEB Journal, 2008. **22**(2): p. 338-342.DOI: 10.1096/fj.07-9492lsf.
105. Aguillo, I.F., *Is Google Scholar useful for bibliometrics? A webometric analysis.* Scientometrics, 2012. **91**(2): p. 343-351.DOI: 10.1007/s11192-011-0582-8.
106. Lin, T.-Y., et al., *Microsoft COCO: Common Objects in Context.* arXiv pre-print server, 2015.DOI: None. arxiv:1405.0312.
107. Everingham, M., et al., *The Pascal Visual Object Classes (VOC) Challenge.* International Journal of Computer Vision, 2010. **88**(2): p. 303-338.DOI: 10.1007/s11263-009-0275-4.
108. Kuznetsova, A., et al., *The Open Images Dataset V4.* International Journal of Computer Vision, 2020. **128**(7): p. 1956-1981.DOI: 10.1007/s11263-020-01316-z.
109. Padilha, T.C., et al., *Tomato Detection Using Deep Learning for Robotics Application*. 2021, Springer International Publishing. p. 27-38.DOI: 10.1007/978-3-030-86230-5_3.
110. Magalhães, S.A., et al., *AgRobTomato Dataset: Greenhouse tomatoes with different ripeness stages [Data set].* Zenodo, 2021.DOI: https://doi.org/10.5281/zenodo.5596799.

111. Moreira, G., et al., *RpiTomato Dataset: Greenhouse tomatoes with different ripeness stages [Data set].* Zenodo, 2021.DOI: https://doi.org/10.5281/zenodo.5596363.
112. USA. Agricultural Marketing Service. Fruit and Vegetable Division. Fresh Products Branch, *United States Standards for Grades of Fresh Tomatoes: Effective October 1, 1991*. 1997, [Washington, D.C.] : U.S. Dept. of Agriculture, Agricultural Marketing Service, Fruits and Vegetable Division, Fresh Products Branch, [1997 printing].
113. Sekachev, B., et al., *opencv/cvat: v1.1.0*. 2020, Zenodo.DOI: 10.5281/zenodo.4009388.
114. Padilla, R., et al., *A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit.* Electronics, 2021. **10**(3): p. 279.DOI: 10.3390/electronics10030279.
115. Abadi, M.i., et al., *TensorFlow: A system for large-scale machine learning.* arXiv pre-print server, 2016.DOI: None arxiv:1605.08695.
116. Magalhães, S.A., et al., *Evaluating the Single-Shot MultiBox Detector and YOLO Deep Learning Models for the Detection of Tomatoes in a Greenhouse.* Sensors, 2021. **21**(10): p. 3569.DOI: 10.3390/s21103569.
117. Bradski, G., *The openCV library.* Dr. Dobb's Journal of Software Tools, 2000. **25**.
118. Hunter, J.D., *Matplotlib: A 2D Graphics Environment.* Computing in Science & Engineering, 2007. **9**(3): p. 90-95.DOI: 10.1109/MCSE.2007.55.
119. Pedregosa, F., et al., *Scikit-learn: Machine Learning in Python.* Journal of Machine Learning Research, 2012. **12**.
120. Grandini, M., E. Bagli, and G. Visani, *Metrics for Multi-Class Classification: an Overview.* arXiv pre-print server, 2020.DOI: None arxiv:2008.05756.
121. Arad, B., et al., *Development of a sweet pepper harvesting robot.* Journal of Field Robotics, 2020. **37**(6): p. 1027-1039.DOI: 10.1002/rob.21937.
122. Arad, B., et al., *Controlled Lighting and Illumination-Independent Target Detection for Real-Time Cost-Efficient Applications. The Case Study of Sweet Pepper Robotic Harvesting.* Sensors, 2019. **19**: p. 1390.DOI: 10.3390/s19061390.
123. Bac, C.W., et al., *Performance Evaluation of a Harvesting Robot for Sweet Pepper.* Journal of Field Robotics, 2017. **34**.DOI: 10.1002/rob.21709.

124. Bac, C.W., J. Hemming, and E.J. Van Henten, *Stem localization of sweet-pepper plants using the support wire as a visual cue.* Computers and Electronics in Agriculture, 2014. **105**: p. 111–120.DOI: 10.1016/j.compag.2014.04.011.
125. Bac, C.W., et al., *Analysis of a motion planning problem for sweet-pepper harvesting in a dense obstacle environment.* Biosystems Engineering, 2015. **146**.DOI: 10.1016/j.biosystemseng.2015.07.004.
126. Barth, R., J. Hemming, and E.J. Van Henten, *Design of an eye-in-hand sensing and servo control framework for harvesting robotics in dense vegetation.* Biosystems Engineering, 2016. **146**.DOI: 10.1016/j.biosystemseng.2015.12.001.
127. Barth, R., et al., *Data synthesis methods for semantic segmentation in agriculture: A Capsicum annuum dataset.* Computers and Electronics in Agriculture, 2018. **144**: p. 284-296.DOI: 10.1016/j.compag.2017.12.001.
128. Benavides, M., et al., *Automatic Tomato and Peduncle Location System Based on Computer Vision for Use in Robotized Harvesting.* Applied Sciences, 2020. **10**: p. 5887.DOI: 10.3390/app10175887.
129. Boryga, M., et al., *Trajectory Planning with Obstacles on the Example of Tomato Harvest.* Agriculture and Agricultural Science Procedia, 2015. **7**: p. 27-34.DOI: 10.1016/j.aaspro.2015.12.026.
130. Chen, C., et al., *Monocular positioning of sweet peppers: An instance segmentation approach for harvest robots.* Biosystems Engineering, 2020. **196**: p. 15-28.DOI: https://doi.org/10.1016/j.biosystemseng.2020.05.005.
131. Chiu, Y.C., P.Y. Yang, and S. Chen, *Development of the end-effector of a picking robot for greenhouse-grown tomatoes.* Applied Engineering in Agriculture, 2013. **29**: p. 1001-1009.DOI: 10.13031/aea.29.9913.
132. Cui, Y.J., J.N. Hua, and P. Shi, *Optimal Design and Simulation on the Major Linkage Parameters of a Harvesting Manipulator.* Applied Mechanics and Materials, 2011. **44-47**: p. 651-655.DOI: 10.4028/www.scientific.net/AMM.44-47.651.
133. De Preter, A., J. Anthonis, and J. Baerdemaeker, *Development of a Robot for Harvesting Strawberries.* IFAC-PapersOnLine, 2018. **51**: p. 14-19.DOI: 10.1016/j.ifacol.2018.08.054.
134. Eizentals, P. and K. Oka, *3D pose estimation of green pepper fruit for automated harvesting.* Computers and Electronics in Agriculture, 2016. **128**: p. 127-140.DOI: 10.1016/j.compag.2016.08.024.

135. Fujinaga, T., S. Yasukawa, and K. Ishii, *Tomato Growth State Map for the Automation of Monitoring and Harvesting.* Journal of Robotics and Mechatronics, 2020. **32**: p. 1279-1291.DOI: 10.20965/jrm.2020.p1279.
136. Fujinaga, T., et al., *Recognition of Tomato Fruit Regardless of Maturity by Machine Learning Using Infrared Image and Specular Reflection.* Proceedings of International Conference on Artificial Life and Robotics, 2018. **23**: p. 761-766.DOI: 10.5954/ICAROB.2018.OS21-5.
137. Harel, B., et al., *Viewpoint Analysis for Maturity Classification of Sweet Peppers.* Sensors, 2020. **20**: p. 3783.DOI: 10.3390/s20133783.
138. Hayashi, S., et al., *Field Operation of a Movable Strawberry-harvesting Robot using a Travel Platform.* Japan Agricultural Research Quarterly: JARQ, 2014. **48**: p. 307-316.DOI: 10.6090/jarq.48.307.
139. Hemming, J., et al., *Fruit Detectability Analysis for Different Camera Positions in Sweet-Pepper.* Sensors (Basel, Switzerland), 2014. **14**: p. 6032-6044.DOI: 10.3390/s140406032.
140. Hemming, J., et al. *Field test of different end-effectors for robotic harvesting of sweet-pepper*. 2016. International Society for Horticultural Science (ISHS), Leuven, Belgium.DOI: 10.17660/ActaHortic.2016.1130.85.
141. Heon, H. and K. Si-Chan. *Development of multi-functional tele-operative modular robotic system for greenhouse watermelon*. in *Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003)*. 2003.DOI: 10.1109/AIM.2003.1225538.
142. Irie, N., et al., *Asparagus harvesting robot coordinated with 3-D vision sensor*. 2009. 1-6.DOI: 10.1109/ICIT.2009.4939556.
143. Ji, W., et al., *Recognition Method of Green Pepper in Greenhouse Based on Least-Squares Support Vector Machine Optimized by the Improved Particle Swarm Optimization.* IEEE Access, 2019. **7**: p. 119742-119754.
144. Kitamura, S. and K. Oka, *Improvement of the Ability to Recognize Sweet Peppers for Picking Robot in Greenhouse Horticulture*. 2006. 353-356.DOI: 10.1109/SICE.2006.315789.
145. Kondo, N., et al. *END-EFFECTORS FOR PETTY-TOMATO HARVESTING ROBOT*. 1995. International Society for Horticultural Science (ISHS), Leuven, Belgium.DOI: 10.17660/ActaHortic.1995.399.28.

146. Kurtser, P. and Y. Edan, *Statistical models for fruit detectability: spatial and temporal analyses of sweet peppers.* Biosystems Engineering, 2018. **171**: p. 272-289.DOI: 10.1016/j.biosystemseng.2018.04.017.

147. Kurtser, P. and Y. Edan, *The Use of Dynamic Sensing Strategies to Improve Detection for a Pepper Harvesting Robot*. 2018. 8286-8293.DOI: 10.1109/IROS.2018.8593746.

148. Lee, B., et al., *A Vision Servo System for Automated Harvest of Sweet Pepper in Korean Greenhouse Environment.* Applied Sciences, 2019. **9**: p. 2395.DOI: 10.3390/app9122395.

149. Li, D., et al., *Cucumber Detection Based on Texture and Color in Greenhouse.* International Journal of Pattern Recognition and Artificial Intelligence, 2017. **31**.DOI: 10.1142/S0218001417540167.

150. Ling, X., et al., *Dual-arm cooperation and implementing for robotic harvesting tomato using binocular vision.* Robotics and Autonomous Systems, 2019.DOI: 10.1016/j.robot.2019.01.019.

151. Liu, G., et al., *A Robust Mature Tomato Detection in Greenhouse Scenes Using Machine Learning and Color Analysis*. 2019. 17-21.DOI: 10.1145/3318299.3318338.

152. Masuzawa, H., J. Miura, and S. Oishi, *Image based recognition of Green Perilla Leaves using Deep Neural Network for a Harvesting Support Robot.* The Proceedings of JSME annual Conference on Robotics and Mechatronics (Robomec), 2018. **2018**: p. 1P1-A02.DOI: 10.1299/jsmermd.2018.1P1-A02.

153. Ostovar, A., O. Ringdahl, and T. Hellström, *Adaptive Image Thresholding of Yellow Peppers for a Harvesting Robot.* Robotics, 2018. **7**.DOI: 10.3390/robotics7010011.

154. Perez-Vidal, C. and L. Gracia, *Computer based production of Saffron (Crocus sativus L.): From mechanical design to electronic control.* Computers and Electronics in Agriculture, 2020. **169**: p. 105198.DOI: 10.1016/j.compag.2019.105198.

155. Qi, L., et al. *A Dynamic Threshold Segmentation Algorithm for Cucumber Identification in Greenhouse*. in *2009 2nd International Congress on Image and Signal Processing*. 2009.DOI: 10.1109/CISP.2009.5304301.

156. Qing-Hua, Y., et al., *Cucumber image segmentation algorithm based on rough set theory.* New Zealand Journal of Agricultural Research, 2007. **50**(5): p. 989-996.DOI: 10.1080/00288230709510377.

157. Schuetz, C., et al., *Evaluation of a direct optimization method for trajectory planning of a 9-DOF redundant fruit-picking manipulator.* Proceedings - IEEE International Conference on Robotics and Automation, 2015. **2015**: p. 2660-2666.DOI: 10.1109/ICRA.2015.7139558.
158. Taqi, F., et al., *A cherry-tomato harvesting robot*. 2017. 463-468.DOI: 10.1109/ICAR.2017.8023650.
159. Tejada, V.F., et al., *Proof-of-concept robot platform for exploring automated harvesting of sugar snap peas.* Precision Agriculture, 2017. **18**(6): p. 952-972.DOI: 10.1007/s11119-017-9538-1.
160. Van Henten, E.J., et al., *An Autonomous Robot for Harvesting Cucumbers in Greenhouses.* Auton. Robots, 2002. **13**: p. 241-258.DOI: 10.1023/A:1020568125418.
161. Van Henten, E.J., et al., *Collision-free inverse kinematics of the redundant seven-link manipulator used in a cucumber picking robot.* Biosystems Engineering, 2010. **106**.DOI: 10.1016/j.biosystemseng.2010.01.007.
162. Van Henten, E.J., et al., *Optimal manipulator design for a cucumber harvesting robot.* Computers and Electronics in Agriculture, 2009. **65**: p. 247-257.DOI: 10.1016/j.compag.2008.11.004.
163. Yamamoto, S., et al., *Development of an end effector for a strawberry-harvesting robot.* Acta Horticulturae, 2008. **801**: p. 565-571.DOI: 10.17660/ActaHortic.2008.801.63.
164. Yasukawa, S., et al., *Development of a Tomato Harvesting Robot*. Icarob 2017: Proceedings of the 2017 International Conference on Artificial Life and Robotics, ed. M. Sugisaka, et al. 2017, Shimohanda: Alife Robotics Co, Ltd. P408-P411.
165. Yin, H., et al., *Technical Note: Ripe Tomato Detection for Robotic Vision Harvesting Systems in Greenhouses.* Transactions of the ASABE, 2011. **54**(4): p. 1539-1546.DOI: https://doi.org/10.13031/2013.39005.
166. Yuan, T., et al., *Detecting the information of cucumber in greenhouse for picking based on NIR image.* Guang Pu Xue Yu Guang Pu Fen Xi, 2009. **29**(8): p. 2054-8.
167. Lee, J., et al., *Artificial Intelligence Approach for Tomato Detection and Mass Estimation in Precision Agriculture.* Sustainability, 2020. **12**(21): p. 9138.DOI: 10.3390/su12219138.
168. Zhou, Y., et al., *Classification and recognition approaches of tomato main organs based on DCNN.* Nongye Gongcheng Xuebao/Transactions of the Chinese

Society of Agricultural Engineering, 2017. **33**: p. 219-226.DOI: 10.11975/j.issn.1002-6819.2017.15.028.

169. Liu, J., J. Pi, and L. Xia, *A novel and high precision tomato maturity recognition algorithm based on multi-level deep residual network.* Multimedia Tools and Applications, 2020. **79**(13-14): p. 9403-9417.DOI: 10.1007/s11042-019-7648-7.
170. Chen, J., et al., *An improved Yolov3 based on dual path network for cherry tomatoes detection.* Journal of Food Process Engineering, 2021. **44**(10).DOI: 10.1111/jfpe.13803.
171. Xu, Z.-F., et al., *Fast Method of Detecting Tomatoes in a Complex Scene for Picking Robots.* IEEE Access, 2020. **PP**: p. 1-1.DOI: 10.1109/ACCESS.2020.2981823.
172. Zhang, W., et al., *Easy domain adaptation method for filling the species gap in deep learning-based fruit detection.* Horticulture Research, 2021. **8**(1).DOI: 10.1038/s41438-021-00553-8.
173. Afonso, M., et al., *Tomato Fruit Detection and Counting in Greenhouses Using Deep Learning.* Frontiers in Plant Science, 2020. **11**.DOI: 10.3389/fpls.2020.571299.
174. Ruparelia, S., M. Jethva, and R. Gajjar, *Real-Time Tomato Detection, Classification, and Counting System Using Deep Learning and Embedded Systems*. 2022, Springer Singapore. p. 511-522.DOI: 10.1007/978-981-16-2123-9_39.
175. Lawal, M.O., *Tomato detection based on modified YOLOv3 framework.* Scientific Reports, 2021. **11**(1).DOI: 10.1038/s41598-021-81216-5.
176. Tsironis, V., S. Bourou, and C. Stentoumis, *TOMATOD: EVALUATION OF OBJECT DETECTION ALGORITHMS ON A NEW REAL-WORLD TOMATO DATASET.* The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2020. **XLIII-B3-2020**: p. 1077-1084.DOI: 10.5194/isprs-archives-xliii-b3-2020-1077-2020.
177. Zhang, L., et al., *Deep Learning Based Improved Classification System for Designing Tomato Harvesting Robot.* IEEE Access, 2018. **6**: p. 67940-67950.DOI: 10.1109/access.2018.2879324.
178. Toon, O.P., et al., *Autonomous Tomato Harvesting Robotic System in Greenhouses: Deep Learning Classification.* MEKATRONIKA, 2019. **1**(1): p. 80-86.DOI: 10.15282/mekatronika.v1i1.1148.

179. Huynh, D.P., et al., *Classifying maturity of cherry tomatoes using Deep Transfer Learning techniques.* IOP Conference Series: Materials Science and Engineering, 2021. **1109**(1): p. 012058.DOI: 10.1088/1757-899x/1109/1/012058.
180. Das, P., J.K.P.S. Yadav, and A.K. Yadav, *An Automated Tomato Maturity Grading System Using Transfer Learning Based AlexNet.* Ingénierie des systèmes d information, 2021. **26**(2): p. 191-200.DOI: 10.18280/isi.260206.
181. Mutha, S.A., A.M. Shah, and M.Z. Ahmed, *Maturity Detection of Tomatoes Using Deep Learning.* SN Computer Science, 2021. **2**(6).DOI: 10.1007/s42979-021-00837-9.
182. Ko, K., et al., *Stochastic Decision Fusion of Convolutional Neural Networks for Tomato Ripeness Detection in Agricultural Sorting Systems.* Sensors, 2021. **21**(3): p. 917.DOI: 10.3390/s21030917.
183. Choi, K., et al., *Tomato Maturity Evaluation Using Color Image Analysis.* Transactions of the ASAE, 1995. **38**(1): p. 171-176.DOI: https://doi.org/10.13031/2013.27827.
184. Li, C., Q. Cao, and F. Guo, *A method for color classification of fruits based on machine vision.* WSEAS TRANSACTIONS on SYSTEMS, 2009. **8**: p. 312-321.
185. Indriani, O.R., et al. *Tomatoes classification using K-NN based on GLCM and HSV color space*. IEEE.DOI: 10.1109/innocit.2017.8319133.
186. Goel, N. and P. Sehgal, *Fuzzy classification of pre-harvest tomatoes for ripeness estimation – An approach based on automatic rule learning using decision tree.* Applied Soft Computing, 2015. **36**: p. 45-56.DOI: 10.1016/j.asoc.2015.07.009.
187. Rupanagudi, S.R., et al. *A cost effective tomato maturity grading system using image processing for farmers*. IEEE.DOI: 10.1109/ic3i.2014.7019591.
188. Sari, Y.A., et al. *Multiplication of V and Cb color channel using Otsu thresholding for tomato maturity clustering*. IEEE.DOI: 10.1109/siet.2017.8304136.
189. Dai, Z., et al., *Inter-Species Comparative Analysis of Components of Soluble Sugar Concentration in Fleshy Fruits.* 2016. **7**(649).DOI: 10.3389/fpls.2016.00649.
190. Heaton, J., *Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning: The MIT Press, 2016, 800 pp, ISBN: 0262035618.* Genetic Programming and Evolvable Machines, 2017. **19**.DOI: 10.1007/s10710-017-9314-z.
191. Dumoulin, V. and F. Visin, *A guide to convolution arithmetic for deep learning.* 2016.

192. Lee, C.-Y., P. Gallagher, and Z. Tu, *Generalizing Pooling Functions in Convolutional Neural Networks: Mixed, Gated, and Tree.* 2015.

193. Scherer, D., A. Müller, and S. Behnke, *Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition*. 2010. 92-101.DOI: 10.1007/978-3-642-15825-4_10.

194. Zhao, Z.-Q., et al., *Object Detection with Deep Learning: A Review.* arXiv pre-print server, 2019.DOI: None. arxiv:1807.05511.

195. Wang, C.-Y., et al., *CSPNet: A New Backbone that can Enhance Learning Capability of CNN.* arXiv pre-print server, 2019.DOI: None. arxiv:1911.11929.

196. Redmon, J. and A. Farhadi, *YOLO9000: Better, Faster, Stronger.* arXiv pre-print server, 2016.DOI: None. arxiv:1612.08242.

197. Redmon, J. and A. Farhadi, *YOLOv3: An Incremental Improvement.* arXiv pre-print server, 2018.DOI: None. arxiv:1804.02767.

Appendix A

Table A1 | *Imaging sensors, visual features and image analysis methods and techniques that can be used to develop a computer vision system.*

	Imaging Sensors	
	Principles	**Pros & Cons**
Monocular vision	Consists of a standard BW (Black and White) or colour camera and a CCD (Charge Coupled Device) or a CMOS (Complementary Metal Oxide Semiconductor)	Simplest and lowest cost system, but only provides 2D information; Problems like light change can influence the imaging results
Binocular Vision	Two cameras separated in a certain distance with a specific angle that allows, through triangulation, to obtain depth information	Most common approach to obtain the 3D position of detected fruit; Image matching is time consuming and the cameras need calibration
Vision and Range Sensors	To acquire depth information in a more direct way, sensors that measure depth, such as LiDAR or RGB-D cameras, can be attached to the cameras	Alternative to obtain the 3D position in the condition of light changing and background clustering; The imaging processing is also a challenge
Spectral Imaging	Recognition of objects based on their different reflectance in selected wavelengths using spectral cameras, that integrates both spectroscopic and imaging techniques	Great advantage when the target and background have the same colors; Imaging processing is very time consuming and the sensor cost is high
Hyperspectral Imaging	Emerging technology that provides the complete spectral signature for each pixel in the visual field of the camera	Brings an overwhelming amount of additional information that leads to better decisions; Costly price, both in acquisition and processing time

(Cont.)

	Visual Features	
	Principles	**Pros & Cons**
Color	The most significant visual feature used in harvesting robots, mainly in the RGB representation. Other color spaces such as HIS, HSV, CIE or L*a*b are also used	Good performance on invariance of size and view point change; Vulnerable to illumination change and color correlation between the fruit and the background
Texture	Texture is perhaps the first visual feature that goes beyond purely local features, like color or spectral reflectance	Effective feature when colour is not discriminatory enough, and it is usually more stable than reflective properties under illumination variations
Shape	Shape implies a particular spatial relationship between the geometrical atoms like points, occluding contours and surfaces that make up a coherent physical object	Shape-based analysis approach is not affected by varying illuminations; Computationally demanding to extract and analyse
Spectral reflectance	Spectral reflectance signatures result from the presence or absence, as well as the position and shape of specific absorption features, of the surface	Effective discriminatory factor when fruits and background have the same color; Sensitive to illumination and cannot resolve issues like occlusions
Thermal response	Thermal response of objects is related to their emitted radiation in the infrared range where it is strongly affected by both the temperature and the emissivity of materials	Usually the background accumulate significantly less heat and emit it for a shorter time, making thermography an excellent approach; Sensitive to the illumination and heat accumulation
Multi-Feature fusion	A single feature rarely represents the target object in a satisfactory manner so it is reasonable to think that one visual feature could compensate for the limitations of the others. Hence, multi-features may provide increased performance	Can improve the recognition rate of uneven illumination conditions, partially occluded surfaces, and similar background features; Still prone to light changes

(Cont.)

Image Analysis		
	Principles	**Pros & Cons**
Elementary methods	Approaching the problem via thresholding the visual feature	Simpler and easier to process models; Less robust, as the high variance of the environments makes algorithms little more than coarse and inaccurate segmentation
Clustering	Form of unsupervised learning approach to partition the image into targets (fruits) and background. Some of the most clustering methods used are the K-means clustering, X-means clustering or Fuzzy Cmeans	Useful when multiple visual cues are merged; Exhibits similar drawbacks as the elementary methods: sensitive to illumination conditions and needs to cope with feature points that do not separate well into clusters
Shape inference	Method of finding a shape that best fits the geometric evidence measured from the image. Inference process can involve a variety of mechanisms such as voting, statistical inference, or optimisation	Works better with spherical fruits; It is not easy to build good models and some suffer from the computational cost
Template matching	Technique for recognising portions of a given image that match with a specific template pattern. Based on similarity measures such as cross-correlation and sum of squared differences	Useful in contexts where the diversity of the target object is small enough; It does not work as well on harvesting robots because the agricultural environment is extremely variable
Voting	Computational technique in which each local visual evidence in the image votes for all possible global interpretations it could arise from. Two of the most poupular models used are Hough transform and circular Hough transform	Good technique for detecting shapes and patterns; Expensive in computation
Machine/Deep Learning	Design and analysis of algorithms that improve their performance based on observable data. Main models used: Artificial Neural Networks or Support Vector Machines	Promising technique that can produce a higher fruit recognition rates; Requires a longer training time and may not deal well with scenarios that are too complex
3D Reconstruction	Establishment of a mathematical model suitable for computer representation and processing of spatial objects. Process of reversing the 3D information collected by visual sensors	Technique able to gather more information; Often requires the use of empirical knowledge that can lead to unsatisfactory results; Sensitive to changes in illumination and some parts of the background

Appendix B

Convolutional Neural Networks Architecture

CNN can be defined as "deeper" ANNs that, unlike conventional ones, are faster at learning and interpreting complex, large-scale problems due to the sharing of weights and the use of more sophisticated models that allow immense parallelisation [68]. Figure B1 illustrates the architecture of CNNs that may include several convolution stages composed of 4 main componentes [65]:

- Kernels (filter bank);
- Convolution layer;
- Non-linear activation function;
- Pooling layer.

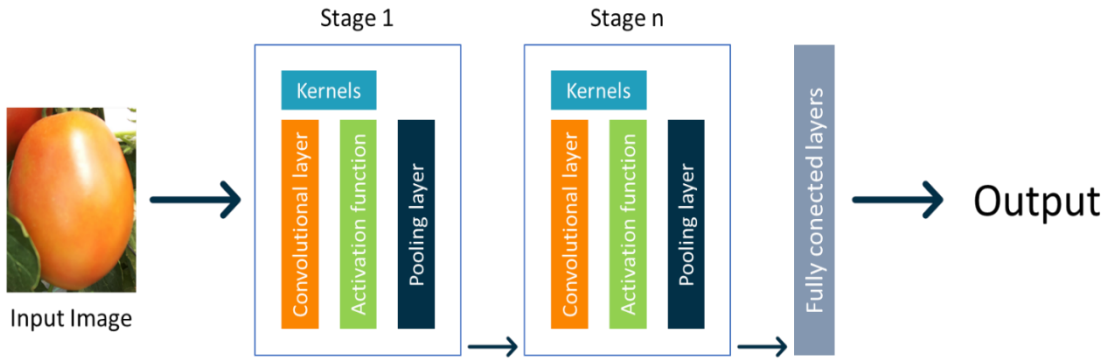


Figure B1 | *Architecture and the main components that make up a CNN. Adapted from: Naranjo Torres, Mora [64].*

Given a certain input image, kernels are responsible for detecting particular features at each point of that input, and thus its spatial translation will be transferred to the next layer without being altered. In the convolution layer, the 2D matrix representing the image undergoes a convolution, resulting in a smaller 2D kernel matrix. In this process a small filter operates from left to right in the image, from the top to the bottom, where at each location the sum of the products between each kernel and its input element is computed (Fig. B2) [65, 190]. The process is repeated using different kernel filters to obtain as many output feature maps as desired [191].

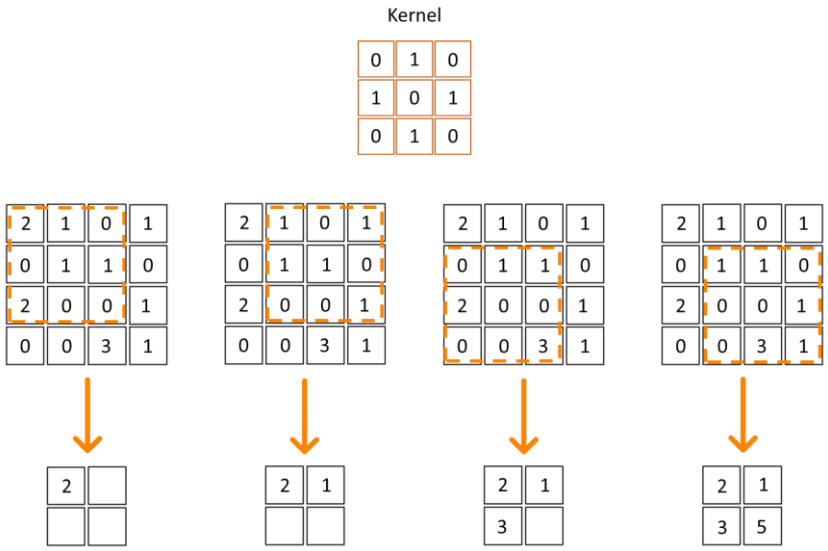


Figure B2 | *Convolution operation with an input image (4x4) and a 3x3 kernel. Adapted from: Naranjo Torres, Mora [64].*

Then, an activation function is applied to the output generated by the kernel filter, which determines the behaviour of the output neuron. In CNNs, the most commonly used activation functions are: Rectified linear unit (ReLu), sigmoid and hyperbolic tangent [65]. The pooling layer, on the other hand, lowers the number of network parameters, reducing the spatial size of the convolution outputs. Two types of pooling can be used for this purpose: max pooling, which calculates the maximum value of each input field, or average pooling, which calculates the average (Fig. B3) [192, 193].

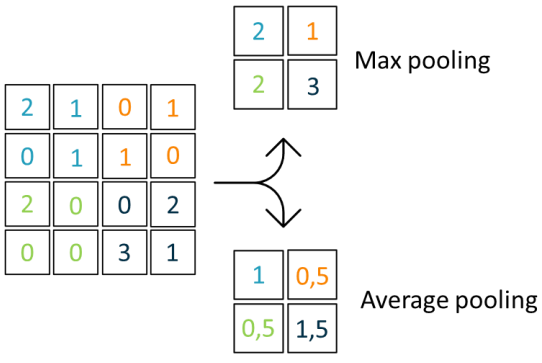


Figure B3 | *Pooling operations by using a 2x2 filters applied with a stride of 2.*

Finally, the final output of all these convolution processes is converted to a 1D matrix and the fully connected layer uses them to classify the input image, just like a traditional ANN [65].

Appendix C

One-Stage Object Detectors

Single Shot Multibox Detector

The SSD approach is much faster than Faster-RCNN because it can simultaneously predict the object class and its bounding box. The input image goes through a series of convolution and pooling layers to generate feature maps at different scales. A 3x3 convolution window evaluates, at each location of the feature maps, a small predefined set of anchor boxes, with different aspect ratios and scales, for which the model simultaneously predicts the probabilities of the class and the boundaries of the bounding boxes for different scales [66, 194] (Fig. C1).

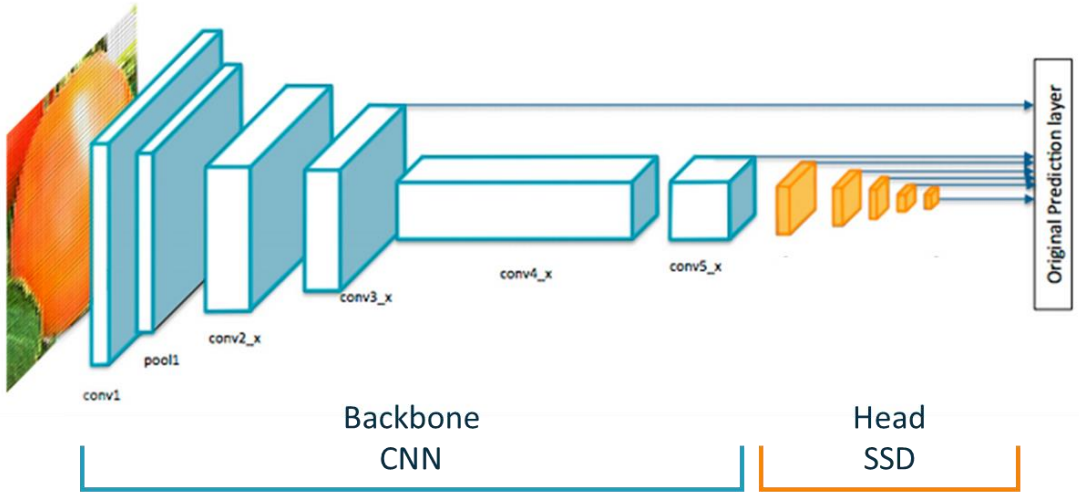


Figure C1 | *SSD architecture, composed of a CNN as backbone and the convolution layers as head.*

You Only Look Once

For YOLO framework, a CNN converts the input image into a dimensional data structure (tensor) of scores for object detection. The image is divided into a grid of cells (each cell has 1/32 of the resolution of the network input) and each cell is responsible for object detection [66] (Fig. C2). YOLO models have evolved over time and are now in their fifth documented version, the YOLOv5[21]. The backbone used in these models belongs to Darknet, an open-source neural network [195]. In its first version (YOLOv1), unlike the SSD framework, no bounding boxes are used. Instead, the model directly predicts two bounding boxes and one class per grid cell [66]. The following versions underwent slight modifications, such as the implementation of anchor boxes (YOLOv2 [196]), the addition of more convolution layers to the backbone or modifications at the head level, such as the multi-scale detector present in the YOLOv3 model [197], making the YOLO approach increasingly fast, accurate and robust to problems such as small object detection.

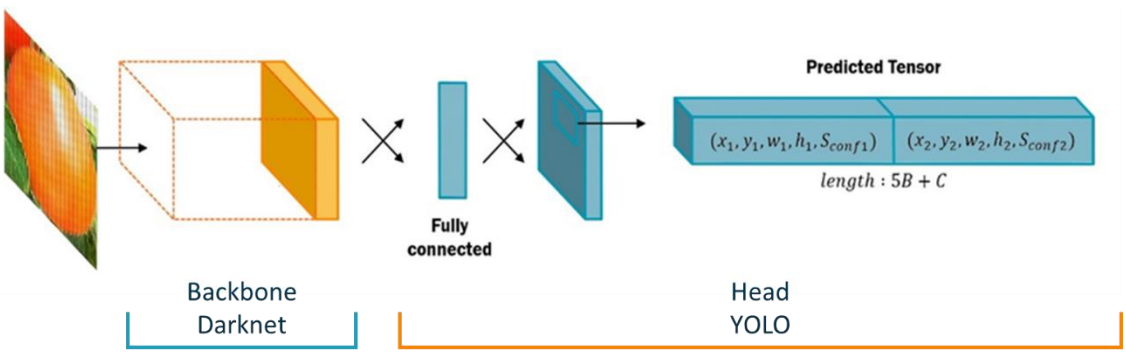


Figure C2 | *YOLO architecture, composed by a Darknet neural network as backbone and the predicted tensor as head.*

[21] YOLOv5 open source code (https://github.com/ultralytics/yolov5). Last accessed: 3 November 2021

Appendix D

Table D1 | *Hue histogram mean of each sample used to build the model and its correlation with the respective class*

Class	Sample	Hue Mean
Green	1	80.9
	2	81.8
	3	81.9
	4	82.9
	5	82.1
	6	82.9
	7	85.1
	8	83.7
	9	82.4
	10	78.7
Turning	1	70.9
	2	63.2
	3	65.4
	4	67.9
	5	60.4
	6	64.5
	7	70.1
	8	68
	9	65.7
	10	67.7
Light Red	1	59.7
	2	56.8
	3	53.8
	4	53.8
	5	57.8
	6	58.7
	7	59.9
	8	56.1
	9	58.2
	10	54.8
Red	1	50.6
	2	48
	3	49.9
	4	51.7
	5	44.3
	6	46.6
	7	50.4
	8	51
	9	47
	10	51.4

Appendix E

Table E1 | *SSC measurement for the 60 tomato samples collected.*

Sample	Class	% Brix	% Brix Mean	Sample	Class	% Brix	% Brix Mean
1	Green	3.5 4 4	3.83	16	Turning	3.9 4 4	3.97
2	Green	3.8 4 4.1	3.97	17	Turning	6 6.1 6	6.03
3	Green	4.1 4.2 4.1	4.13	18	Turning	5.8 5.9 5.8	5.83
4	Green	4.2 4 4.3	4.17	19	Turning	4.6 4.5 5	4.70
5	Green	4 4.3 4.5	4.27	20	Turning	4 4 3.9	3.97
6	Green	3.9 4 4	3.97	21	Turning	5 5.1 5	5.03
7	Green	4 4 4.2	4.07	22	Turning	4.5 4.2 4.3	4.33
8	Green	4 4 4.1	4.03	23	Turning	4.5 4.7 4.8	4.67
9	Green	4 4.1 4	4.03	24	Turning	3.9 4.1 4.3	4.10
10	Green	4.1 4.1 4.1	4.10	25	Turning	5 5 5	5.00
11	Green	4.5 4.3 4.5	4.43	26	Turning	4.5 4.2 4.3	4.33
12	Green	4.4 4.3 4.5	4.40	27	Turning	4.5 4.6 4.9	4.67
13	Green	4.5 4.5 4.3	4.43	28	Turning	5.8 5.7 5.6	5.70
14	Green	4.2 4.4 4.5	4.37	29	Turning	5 4.9 5	4.97
15	Green	4.2 4.4 4	4.20	30	Turning	4.8 4.7 4.7	4.73

(Cont.)

Sample	Class	% Brix	% Brix Mean	Sample	Class	% Brix	% Brix Mean
31	Light Red	4.8 4.3 5	4.70	46	Red	4.5 4.9 4.9	4.77
32	Light Red	4.2 4.5 4.8	4.50	47	Red	5 5 5	5.00
33	Light Red	4 4.5 4.2	4.23	48	Red	4.9 4.5 4.6	4.67
34	Light Red	4.2 4.8 4.4	4.47	49	Red	4.4 4.5 4.5	4.47
35	Light Red	5 4.6 4.9	4.83	50	Red	6.2 6.1 6.3	6.20
36	Light Red	4.5 4.6 4.9	4.67	51	Red	4.2 4.1 4.1	4.13
37	Light Red	5.7 5.8 5.8	5.77	52	Red	5 5 5	5.00
38	Light Red	4.3 4.6 4.7	4.53	53	Red	4.3 4.3 4.3	4.30
39	Light Red	4.6 4.7 4.1	4.47	54	Red	5 5 5.2	5.07
40	Light Red	4 4 4.4	4.13	55	Red	4 4 3.9	3.97
41	Light Red	4 4 4.2	4.07	56	Red	4 4.2 4.1	4.10
42	Light Red	3.9 4.1 4	4.00	57	Red	3.9 3.9 4	3.93
43	Light Red	4.5 4.1 4	4.20	58	Red	4 3.9 4.1	4.00
44	Light Red	4.6 5 5	4.87	59	Red	4.4 4.2 4.5	4.37
45	Light Red	3.8 4.2 4	4.00	60	Red	7.1 7 7.2	7.10