# Vision-based Detection, Tracking, and Trait Extraction of Soft-Fruit for Automation in Horticulture

UNIVERSITY OF
LINCOLN

Raymond Kirk

The University of Lincoln Corporate Guidelines 1.0 April 2013

School of Computer Science

College of Science

University of Lincoln

Submitted in partial satisfaction of the requirements for the
Degree of Doctor of Philosophy
in Computer Science

*Supervisor* Dr. Grzegorz Cielniak
*Second Supervisor* Dr. Michael Mangan

January, 2023

# Abstract

This thesis presents methods for automation in horticulture. Focusing on methods to detect, track and extract traits such as volume and mass from fruit non-destructively. A combination of these approaches results in a system that covers practical application requirements for generating data points for horticultural processes such as yield estimation, disease prediction, cultivation management and enabling of robotic applications such as harvesting, data acquisition and autonomous precision farming. Driven by industry challenges, three novel solutions are presented to detect fruit in variable conditions of illumination and viewpoint, a tracking component to re-identify fruit in clusters across image sequences and trait extraction methods that are applied on top of the tracking or detection outputs to non-destructively estimate crop parameters such as size and volume. The motivation of this thesis is twofold: firstly, to demonstrate the capabilities of computer vision techniques applied in horticulture; secondly, the potential for computer vision techniques to facilitate more efficient and accurate crop assessment. The proposed approaches have several advantages compared to some traditional manual approaches: they are non-destructive, fast, achieve state-of-the-art performance, are scalable and are relatively cheap. This thesis demonstrates detection, counting, and analysis of fruit through image-sequences, which makes the solution flexible enough to work on any type of fruit from video feeds, as opposed to static images containing information restricted to one point in time that does not exploit any spatial relationship. The extracted information about the fruit can be used to recommend useful suggestions to a grower, for example reducing the number of required fruit-pickers, estimating harvest yield, reducing collection efforts, or optimising the harvest period for higher market cost. Ultimately, this enables farmers to make better use of their resources while optimising horticultural processes ahead of time, meeting environmental and management targets.

# Acknowledgements

I would like to acknowledge my friends Joseph Donaldson, Catherine Denniss, and Adam Crossan. My family, specifically my mother Josie Angel, for the support and motivation over the years. The great team of individuals at the University of Lincoln and Lincoln Centre For Autonomous Systems for the thought-provoking discussions. The excellent network of researchers at the Collaborative Training Partnership for Fruit Crop Research. The fantastically motivated academics at the Lincoln Institute for Agri-Food Technology. For the many entertaining pub nights and socially distanced quizzes, Jacobus Lock, Halvard Grimstad, and Kristoffer Skarsgård. My supervisors and friends Michael Mangan and Grzegorz Cielniak for support of the highest calibre. Ultimately, Emily Ley, without your support this thesis would not have been half as fun, or half as complete, I am forever grateful for you.

*This work is dedicated to my father, Christopher Martin Angel, who we sadly lost to cancer in the early years of this academic journey. You are forever remembered.*

# Table of Contents

# List of Figures

# List of Tables

# List of Abbreviations

| | |
|---|---|
| ANN | Artifical Neural Network |
| AP | Average Precision |
| AUC | Area Under Curve |
| CHT | Circular Hough Transform |
| CNN | Convolutional Neural Network |
| COCO | Common Objects in Context |
| CPU | Central Processing Unit |
| ExG | Excess Green Vegetation Index |
| ExGExR | Excess Green minus Excess Red Vegetation Index |
| ExR | Excess Red Vegetation Index |
| FPN | Feature Pyramid Network |
| GMM | Gaussian Mixture Models |
| GPU | Graphics Processing Unit |
| GT | ground truth |
| IoU | Intersection Over Union |
| MAE | Mean Absolute Error |
| mAP | Mean Average Precision |
| MOT | Multi-Object Tracking |
| NMS | Non-Maximum Suppression |
| NN | Neural Network |
| PCA | Principal Component Analysis |

| | |
|---|---|
| R-CNN | Region-based Convolutional Neural Network |
| ReLU | Rectified Linear Unit |
| RoI | Region of Interest |
| RPN | Region Proposal Network |
| SfM | Structure From Motion |
| SOT | Single-Object Tracking |
| SOTA | State-Of-The-Art |
| SSD | Single Shot Detector |
| SVM | Support Vector Machine |
| VOC | PASCAL Visual Object Classes Challenge |
| YOLACT | You only look at the coefficients |
| YOLO | You only look once |

# Chapter 1

# Introduction

Agriculture is one of the world's longest standing industries, and it is currently facing some of the most impactful challenges on the planet. From climate change, net-zero carbon pledges to growing more food for more people, on less fertile land, and in less space. Innovation and evolution over recorded history in agriculture and horticulture has largely been driving productivity with larger and more efficient form factor machines. No longer can large-scale mechanical solutions can be depended upon, and technology needs to be investigated to drive innovation in this sector (Tian et al., 2020).

The industry has an unquantifiable source of data obtainable via sensing technology that is either too laborious, too expensive, or too difficult to collect while minimising to a reasonable cost. Imagine a fruit and vegetable farmer who could estimate the biomass, quality and, number of potatoes within a crop without damaging or harvesting a single one. Through data driven applications utilising radar sensors (Konstantinovic et al., 2007) this type of technology is being developed and introduced to the industry (*TuberScan* 2022), providing new-margins to operate within and is an excellent example of the innovation possible when combining sensors and algorithms into data driven applications.

Over the past few decades, technology, and automation has enabled industries to scale in orders of magnitude. Application of these advancements to horticulture is no mean feat, but is one of the final frontiers of modern automation in agriculture. Many of the tasks in other sectors that are easily defined, and are repeatable, expensive or laborious have been successfully replaced with machines and systems (hardware and/or software). A lot of horticultural processes, however, remain somewhat in lagging behind in the adoption of advancements. Research has been scaling in this area at an excellent rate, as seen in Figure 7.1, and adoption lags behind.

This thesis introduces an application and novel approaches of computer vision techniques to detect, track, count and analyse soft-fruits (primarily tested on hand-crafted strawberry datasets) for generating data points for horticultural processes such as yield estimation, disease prediction, cultivation management and enabling of robotic applications such as harvesting, data acquisition and autonomous precision farming. The motivation for this research is twofold: firstly, to demonstrate the capabilities of computer vision techniques applied in horticulture; secondly, the potential for computer vision techniques to facilitate more efficient and accurate crop assessment. Our proposed approaches have several advantages over traditional manual approaches: they are non-destructive (does not require any physical interaction with the fruit), fast, scalable and relatively cheap.

The tone of this thesis heavily directs the motivation of this work towards the future development of yield forecasting systems for soft-fruit enabled and improving on historical forecasting methods via the integration of high fidelity data points of individually scanned fruit on a crop to bolster yield accuracies. This is to demonstrate the capabilities of such a system within horticulture and motivate the adoption of such technologies in society. Computer vision of strawberries in horticulture can provide farmers with information about the estimated yield, which allows them to make informed decisions about harvesting and crop management.

Our approach demonstrates counting fruit through image-sequences (videos), as opposed to single images only containing information restricted to one point in time that does not exploit any spatio-temporal relationship. Additionally, spatial information about the fruit can be used to recommend useful suggestions to a grower, for example reducing the number of required fruit-pickers, estimating harvest yield, reducing collection efforts, or optimising the harvest period for higher market cost. Ultimately, this enables farmers to make better use of their resources while optimising the harvesting process.

This piece of work is first and foremost an application of artificial intelligence and computer vision to the horticultural industry and hopes to be a handrail for development, deployment, and motivation for horticultural/computer vision practitioners to advance adoption of key technologies in this space. Specifically focusing on detection and tracking of multiple fruits for counting, yield forecasting, automated harvesting and quality grading applications.

## 1.1 Motivation

Computer vision is a field of computer science that involves the analysis of digital images and videos to extract information about the physical world. In the field of horticulture, computer vision can be used to assist farm managers and workers in a wide range of processes. Some examples include, cameras being utilised to monitor crops and provide information about crop health, which is then used to make decisions about how best to care for the plants throughout their life cycle. In other cases, cameras could be used in conjunction with other sensors to detect diseases or pests, and provide alerts. In some cases, sensors are utilised to automate processes that would otherwise require human labour, such as detecting when plants need water or nutrients, and delivering the required items. Ultimately, the use of computer vision in agriculture has the potential to increase productivity and quality (Anagnostis et al., 2022).

Our motivation for the research contained in this thesis was to develop a data driven system capable of aiding in critical decision support on the farm. This problem was approached by asking how a system could be developed to collect and process the vast quantity of data available to fruit growers. The solution is broken down into three components, detection, tracking and analysing each individual fruit from video data as shown in Figure 1.1. Augmenting the traditional crop walking process with a low-cost camera capable of collecting the data required to improve efficiency of many of the current practices facing challenges such as, crop management, yield forecasting and labour management. The problem was formulated as such so that growers could easily adapt, implement and use such a system as a tool for forward innovation.

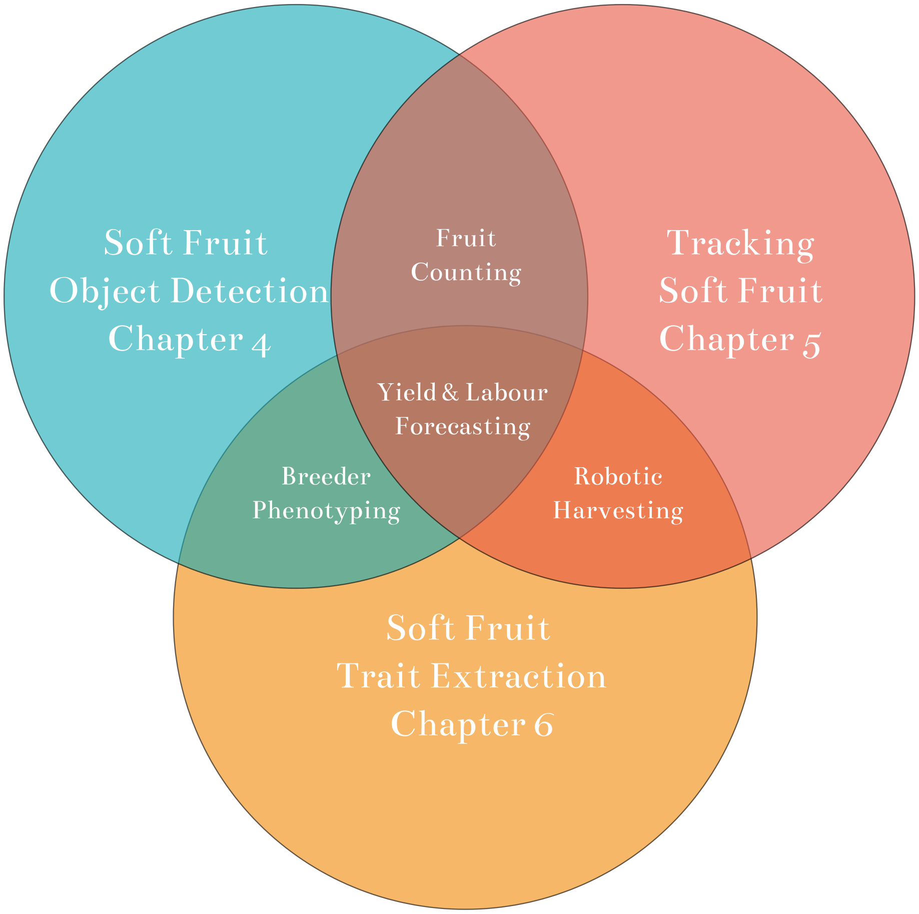


Figure 1.1: Farm and grower ready techniques for automation in horticulture.

As computer vision algorithms become more sophisticated, there is a growing need for their application in new and interesting domains. This thesis provides the motivation for the use of computer vision algorithms within horticulture and provides an overview of the challenges currently faced in the scope of how computer vision algorithms and techniques can be used to overcome them, providing tangible benefits to growers and end-users.

The utilisation of sensing technology, data analysis and robotics are vast within this sector and with the advent of computing power, sophisticated analytical frameworks, and small form factor autonomous robots, frontiers are crossing over into agriculture and horticulture. Adoption of these new technologies brings tangible benefits to everyday operation, notably and specifically to horticultural operations: a) robotics provide an energy and cost-efficient means of covering large areas, b) sensing technology provides a scalable and reliable way to collect useful data points, and c) advancements of learning based algorithms provide generalised methods to analyse the large quantities of data available.

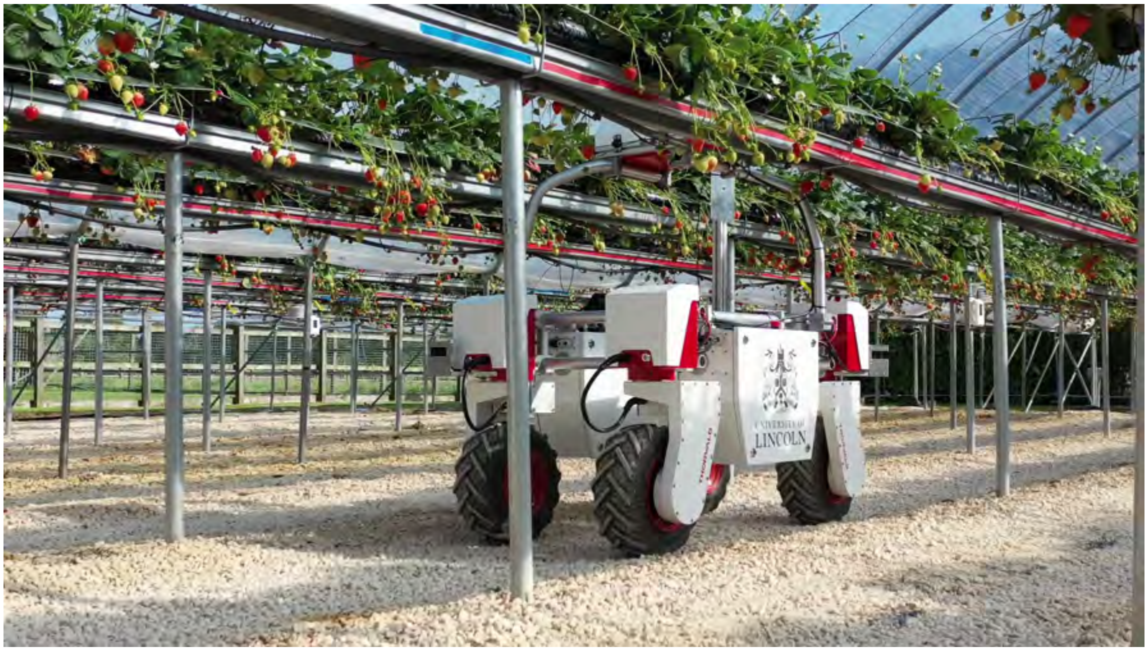


Figure 1.2: Sensors mounted on a robotic platform for data capture in polytunnels.

To maximise adoption in this sector, new technologies have to optimise for multiple objectives. One of the largest constraints is existing infrastructure, the supply-and-demand chain is delicate and margins are tight. New developments will need to ensure they can be deployed on existing infrastructure to maximise the short-term gain and use of expertise in the industry (Tian et al., 2020). Computer vision systems are agnostic to explicit requirements for infrastructure and are a great tool for growers. Many systems are small, power efficient and offer versatile mounting solutions. Further to this, vision systems intrinsically exploit the similar relationship that is shared between historical infrastructure design, and the subsequent labour forces in which they were designed for navigating via visual cues.

Computer vision for fruit systems face many challenges and features, shown in Figure 1.3, which are shared between all applications of vision systems and some unique to different fruit types. The most prevalent issues they currently face in detection are (a) occlusion from the plant canopy, other fruits, shadowing or planting structures, (b) variable illumination from sources such as the sun outdoors and non-uniform lighting indoors, and (c) the computational cost. Moreover, variable weather, seasonal conditions, growth cycles, human induced environmental changes and multiple views are all constraints that detrimentally impact the performance and robustness of fruit detection systems and have been prevalently noted in literature.

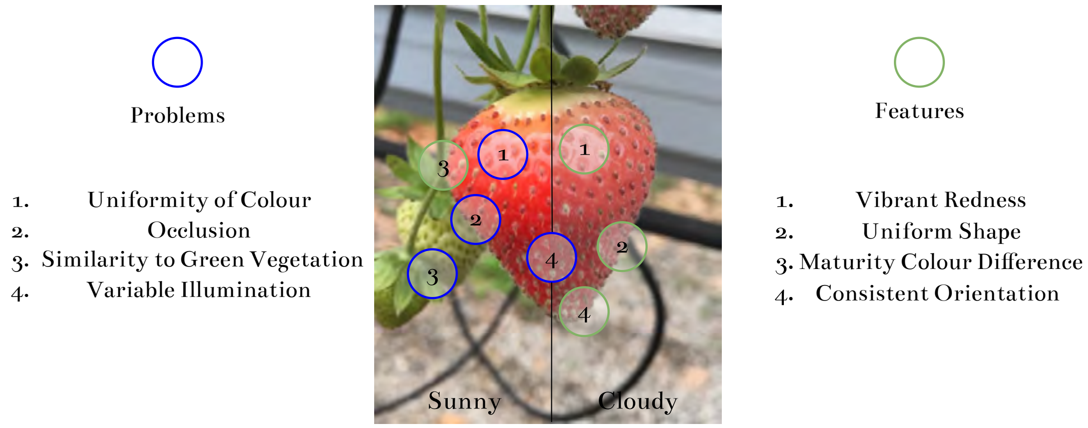


Figure 1.3: Computer Vision for Soft Fruits Challenges and Features

A developed system must be able to (a) detect the produce of interest (b) infer aspects of produce appearance (e.g., size, ripeness, health) and (c) parse guidance information to the physical picking apparatus. Research to date has largely focused on proof-of-principle studies investigating the best combination of sensing hardware and software processing, as discussed in Chapter 3, but in recent years it has accelerated with the advance of deep learning based methods aforementioned in Section 3.2.2. Strawberries present multiple challenges for computer vision approaches, they face many of the classic issues vision systems face, they're difficult to pick due to their softness, occlusion is a major issue due to surrounding vegetation, illumination has a huge impact on colour based approaches and the fruits have multiple stages of maturity that all have different distinguishing features, increasing difficulty of accurate phenotyping. Strawberries are very vibrant when ripe, but when unripe they share similar characteristics to background vegetation as presented in Figure 1.4. Soft-fruit appearance is also temporally unstable, unlike many data sets in computer vision the same object can rapidly change appearance over a short time window (2 days), an example is shown in Figure 1.4. This provides additional challenges for tracking, detection and phenotyping applications deployed in-field. The accuracy of systems could change rapidly over each growing season as the crop develops.

Reinforcing the need for stable systems, generalised to changes in appearance. The

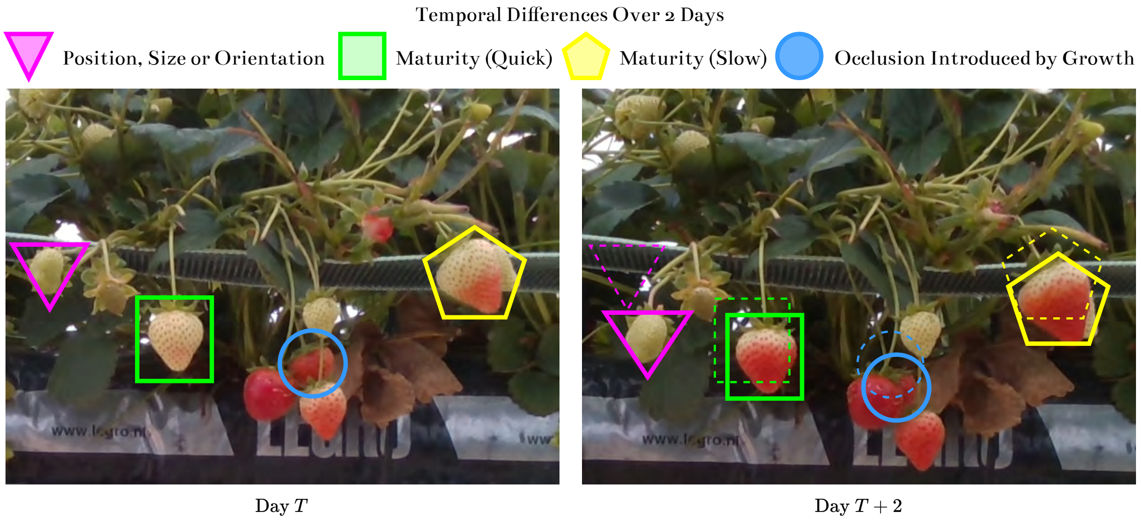

Temporal Differences Over 2 Days
Position, Size or Orientation
Maturity (Quick)
Maturity (Slow)
Occlusion Introduced by Growth
Day $T$
Day $T+2$

Figure 1.4: Computer Vision for Soft Fruits Temporal Stability of Appearance

sensor used to collect data can also vary in quality, however this is largely mitigated with careful consideration of the training dataset samples and the sensor used in production. A comprehensive review of the sensors and systems that have been used in previous research is provided in (Gongal et al., 2015) and includes greyscale, colour, spectral and thermal cameras for fruit detection and single cameras, laser range finders, stereo-vision systems and time of flight systems for fruit localisation. A visual description of the challenges that the application of computer vision systems will face is apparent in Figure 1.5. The intense clustering of the fruits, occlusion from leaves, fruits and other objects as well as the illumination variation caused by the sun results in difficult harvesting conditions. Shown also on the diagram in red are ripe strawberry detections, which simplifies the harvesting operation by ignoring most of the other fruit (maturity classification); this is one way vision systems can help autonomise processed that require the mitigation of monotonous operations such as harvesting. Currently, human pickers can maintain on average 10-16 seconds per strawberry for packing and placing into graded punnets, which gives the image processing algorithms a fairly large window to detect the fruits however for online systems such that of counting fruit real-time (input data is processed within milliseconds so that it is available virtually immediately as feedback to the process from which it is coming) is a requirement.

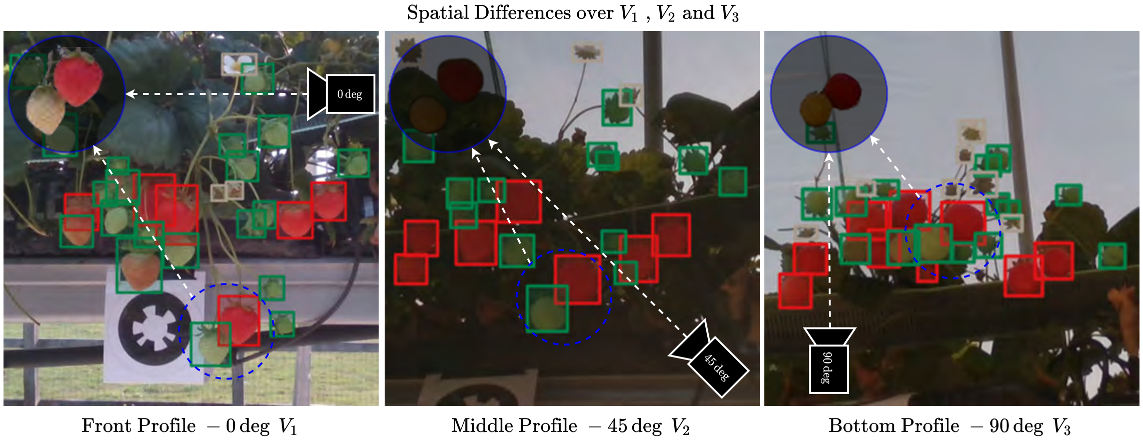

Spatial Differences over $V_1$ , $V_2$ and $V_3$
0 deg
45 deg
90 deg
Front Profile – 0 deg $V_1$
Middle Profile – 45 deg $V_2$
Bottom Profile – 90 deg $V_3$

Figure 1.5: Computer Vision for Soft Fruits Spatial Stability of Appearance

The benefit of detection systems is discussed above, however, in many cases the detection alone is not enough to provide adequate information to improve current horticultural practices. Typically, the detections need to be stitched together or processed in sequences (videos) to provide complete information about the state of a crop only from computer vision based approaches. In some cases, integration of 3D information (RGBD) is also needed to inform end effectors or for mapping an environment accurately. Given this information, further analysis of the results can be performed to generate qualitative systems. This section introduces approaches in literature around computer vision for horticulture, broken down into three sections, detection, tracking and phenotyping. Which when combined can address most of the current horticultural practices, such as harvesting, forecasting and yield estimation. Traditionally growers utilise *crop walking* to perform common growing practices, hence why computer vision lends itself as such a useful tool in this domain, if it can be observed it can be optimised through the advancements made in the computer vision domain.

Below, a series of identified challenges is presented that soft fruit perception systems must overcome:

1. **Real World Conditions** - Some of the current research, developed and tested in limited indoor scenarios, generally are not designed to accommodate the complexity of an agricultural environment.

2. **Multiple Views** - Fruits are not static and change over time, frequently changing orientation and relative size.
3. **Human/Animal Interaction** - The plant structures can change drastically, for example a human picker will physically alter the plant appearance in order to harvest fruit.
4. **Weather** - Directly impacting the quality of data acquisition. Variation in light intensity and weather greatly detriment algorithm reliability and performance.
5. **Growth Cycle** - Dependent on plant species or location, the disparity between expected plant stage and environmental conditions can be great.
6. **Speed** - Horticultural computer vision systems must be able to increase the efficiency in real world conditions, maintaining a profitable throughput in their domain (such as harvesting).

There are endless use cases of computer vision in horticulture and agriculture, the underlying technology provides data on a scale that was previously infeasible to collect manually. State-of-the-art research is showing resounding success of applying these techniques to a wide range of crops. Some robust examples include Williams et al., 2019 who show that up to 70% of kiwi fruits (51% in validation experiments) can be harvested through vision guided robotic arms, and up to 90% of the kiwis are successfully detected by the vision system. Häni, P. Roy and Isler, 2018 introduce a modular end-to-end system for apple orchard yield estimation, utilising a novel semantic segmentation-based fruit detection and counting technique, the deep learning-based approach for fruit counting obtains an accuracy between 95.56% and 97.83% when combined with Gaussian Mixture Models validating applications of apple counting through computer vision and sensing technologies.

One of the limitations of these systems is they require manually labelled images, which is laborious and demanding fiscally. Through simulated learning, Rahnemoonfar and Sheppard, 2017 show that high accuracy systems can be trained for tomatoes when availability of a large number of training samples is not feasible. They obtain test accuracies of 91% when applied to real images, only 2% lower than in simulation, validating the development of horticulture systems for plant disease prevention, labour scheduling and count estimation of fruits, flowers, and trees.

Tripathi and Maktedar, 2020 discuss the power of data-driven applications that utilise computer vision for horticulture in a recent review. They present a review of a wide range of methods currently being developed in this domain. By using images to understand the structure of plants, computer vision can be used to predict the yield of a crop, prevent diseases, inform growing processes, enable automation amongst many more use cases. This information can then be used by farmers and horticulturalists to make informed decisions about how best to care for their crops.

## 1.2 Aims and Objectives

Context is provided in this section to the aims and objectives within the cross-over field between computer science and horticulture to enable the development of applications and advancement of current horticultural practices. The presented work aims to create a system which is capable of addressing some of the key challenges faced by computer vision practitioners when deploying methods within the horticultural/agricultural domain. This work tries to address some limitations faced by current approaches and extend current understanding in this area, specifically:

Objective 1 **Exploration of the problem space within horticulture**

Objective 1.1 Provide a background of the concepts, methods, and frameworks underlying the cross-over between the needs of the horticultural industry and computer vision based solutions, such as machine learning, artificial intelligence and deep learning.

Objective 1.2 To summarise the state-of-the-art in the form of a literature review, and highlight the challenges and opportunities that lie ahead. Discussion of the relevance in the context of horticultural applications, particularly focusing on the application of deep learning to crop detection, tracking and non-destructive phenotyping.

Objective 2 **Accurate and robust detection of soft-fruits in images**

Objective 2.1 A fast fruit detection algorithm is required to ensure data can be consumed as fast as it is captured, to ensure farms that utilise the technologies can implement it within their current practices to augment their capabilities and drive decisions with data.

Objective 2.2 Assessment of the impact many varying factors such as natural or man-made illumination sources relative to the sensors, rigid static structures necessary for plant growth that can block sensor viewports, meteorological factors such as haze, humidity, temperature, and wind as well as occlusion introduced by the plant structure itself can affect accuracy in image data.

Objective 2.3 Robust and accurate systems are required to ensure effective deployment and added benefit to crop processes, this aim explores how deep neural networks can be extended and augmented to coerce models into learning simpler representations of fruits resulting in faster training and greater accuracy in variable conditions.

Objective 3 **Stable re-identification of fruit detections in image sequences**

Objective 3.1 Investigation of limitations of current object tracking frameworks. Object detection algorithms provide a static representation of an environment useful for applications that need no temporal context such as counting the number of fruit in a single image, however to accurately and quickly count the number of fruit in an entire farm from a sequence of images temporal information is required to avoid issues such as double counting an instance.

Objective 3.2 Intrinsically detection systems applied to multiple frames carry no association information with it, techniques exist in the computer vision domain comprised of target detection, appearance models, motion-models and correction for stitching multiple representations of a scene (or image sequences) typically deemed Multi-Object Trackers. An investigation is presented for the application of advancements in the MOT space to create detect multiple instances of the same fruit across multiple time points.

Objective 4 **Non-destructive methods for trait extraction of soft-fruits**

Objective 4.1 Extraction of meaningful patterns from unstructured data such as images or videos is a key component of a computer vision system. Image data and the output of detection and tracking systems is analysed to extract useful structured information describing the soft-fruit.

Objective 4.2 To explore how post-detection and post-tracking methods maximise the value of detection and tracking systems. Localisation and classification of fruit in images and videos provides minimal qualitative information to a grower. Methods are investigated to non-destructively estimate crop traits.

It is worth noting that the application and processes followed in this thesis can be applied in different domains, where perhaps problems are bound by similar constraints. An example is provided of how computer science practitioners can hone their systems to fulfil challenges posed by industry and societal needs to build better and more efficient relationships between research systems and practical deployments.

## 1.3 Contributions

This section outlines and details the main contributions presented in this thesis to the field of Computer Vision curated and applied to the Agri-Food domain. Specifically within horticulture and validated on handcrafted strawberry crop datasets. A system that is capable of localising strawberries within images, tracking their identities across frames and extracting traits such as width, height, and volume non-destructively from each detection or averages of detections across numerous frames is introduced. The contributions are presented against state-of-the-art works and describe the societal benefits by formulating data collection within horticulture via computer vision techniques and methodologies. The presented system is capable of taking unstructured image data from crop rows within horticultural sites and transforming it into structured qualitative information describing the current crop state, load and, when sampled across multiple days, crop performance. The system is formed through the following contributions:

- **Chapter 4** - Coercive and free learning policies to shortcut learning more representative features for combating the transfer of lab based object detection models on curated datasets to unseen outdoor data from multiple view points. In Chapter 4 a system for detection is introduced, *L*a*b*Fruits: A Rapid and Robust Outdoor Fruit Detection System Combining Bio-Inspired Features with One-Stage Deep Learning Networks* Kirk, Cielniak and Mangan (2020b), a neural architecture based on a single-stage detector, RetinaNet (T. Lin et al., 2017). The training is formulated with early fusion of more representative colour space (L*a*b*) for fruits to train faster and increase accuracy with respect to the techniques at the time of publication. Training with these policies is shown to lead to minimal accuracy increase over regular colour spaces when applied to the regular dataset, but when applied to unseen examples from multiple views that contain dramatic appearance changes (such as illumination and colour) that it leads to a much greater accuracy. Further, optimising object detection networks for use within the industry.

- **Chapter 5** Novel extensions of detect-to-track based object tracking frameworks to count soft-fruit in images (detect) and across image-sequences (track). In Chapter 5 a framework is introduced, *Robust Counting of Soft Fruit Through Occlusions with Re-identification* Kirk, Mangan and Cielniak (2021), based on DeepSort (Wojke, Bewley and Paulus, 2017), the de facto state-of-the-art tracker on the MOTA challenge at the time of publication, to count and track strawberry instances across frames addressing the baseline inaccuracy with the standard approach on small homogeneous clustered objects. Our main contributions are (1) a novel first re-identification and label probability based tracking framework, generalising the approach for multiple classes, applied on mobile robots for the purpose of counting fruits (2) extension of a popular re-identification tracking formalisation to embed contextual, shape and class information into association cost (3) four sequences of hand labelled Strawberry data for tracking in complex environments shared for bench-marking with the community and (4) validation of the counting accuracy for the purpose of yield estimation.

- **Chapter 6** Introduction of approaches, for *Non-destructive Soft Fruit Mass and Volume Estimation for Phenotyping in Horticulture* Kirk, Cielniak and Mangan (2021a) to maximise the value of detection and tracking within horticulture to extract phenotypic traits from object detections and tracks in Chapter 6. Destructive phenotyping is an expensive and rarity within the industry due to the time and margin constraints of fruit growers during the season, however the data provided generates critical insights for crop management and breeding policies. Work is presented to transform image data with bounding box or segmentation based detections to non-destruction volume, size, and weight estimations in real-time. Enabling the collection and analysis of millions of samples quickly, foregoing the current constraints. This chapter presents (1) three novel approaches to estimate the phenotypic traits, width, height, cross-section length, volume, and mass from only image segments and depth information of strawberries, (2) a thorough evaluation of the proposed methods in lab conditions against ground truth (GT) data, and, (3) application and validation of the proposed methods in-field from a robotic platform.

## 1.4 Publications

Parts of this thesis have been published in partial satisfaction of the requirements:

**Kirk, R.**, Cielniak, G. en Mangan, M. (2020) *“L* a* b* fruits: A rapid and robust outdoor fruit detection system combining bio-inspired features with one-stage deep learning networks”*, Sensors. Multidisciplinary Digital Publishing Institute, 20(1), bl 275. Kirk, Cielniak and Mangan (2020b)

**Kirk, R.**, Cielniak, G. en Mangan, M. (2020) *“Feasibility Study of In-Field Phenotypic Trait Extraction for Robotic Soft-Fruit Operations”*, UKRAS. Kirk, Cielniak and Mangan (2020a)

Wagner, N., **Kirk, R.**, M Hanheide en Cielniak, G.(2021) *"Efficient and Robust Orientation Estimation of Strawberries for Fruit Picking Applications"*, IEEE International Conference on Robotics and Automation (ICRA). Wagner et al. (2021)

**Kirk, R.**, Cielniak, G. en Mangan, M. (2021) *"Non-destructive Soft Fruit Mass and Volume Estimation for Phenotyping in Horticulture"*, in ICVS - International Conference on Computer Vision Systems. Springer International Publishing, bll 223–233. Kirk, Cielniak and Mangan (2021a)

**Kirk, R.**, Cielniak, G. en Mangan, M. (2021) *"Robust Counting of Soft Fruit Through Occlusions with Re-identification"*, in ICVS - International Conference on Computer Vision Systems. Springer International Publishing (**Best Paper Award**), bll 211–222. Kirk, Cielniak and Mangan (2021b)

## 1.5 Organisation

This thesis is organised as follows. The technical and application domains present in our research are introduced in Chapter 1, mainly noting the motivation and problems faced in research and industry. Summarising the impact of our work through a list of aims and objectives, peer-reviewed publications, overall contributions and dissemination activities. The organisation of this document is introduced below to allow ease of navigation.

In, Chapter 2 an introduction to the research domains present in this thesis and background work relating to each of the contributions is presented. Introducing computer vision techniques, evaluation metrics for our approaches and the underlying algorithms and motivations behind each of the deep learning based methods. Chapter 3 introduces related work and the history of developments that relate to the core contributions of this thesis, specifically the development of object detection and segmentation frameworks introduced with the advent of neural architectures that are responsible for the recent improvements in the object tracking (detect-to-track) paradigm.

Chapter 4 proposes techniques for the application of state-of-the-art object detection frameworks including data acquisition, sensor design, effective data annotation, and present frameworks for fruitful object detection in the horticultural domain. Chapter 5 proceeds to describe the process that was proposed to extend the static (one point in time) detection algorithms for soft fruits to track and count instances over time. Importantly, this section describes the critical process of converting object detections into meaningful data points for further exploratory analysis and generating useful metrics in industry. Tracking instances in image sequences or detecting objects within an image is extremely useful for practical applications within horticulture, in Chapter 6 an introduction to the methods used to extract traits such as width, height, weight, and volume from each object in the images to maximise the value of detected objects is presented. Finally, in Chapter 7 the contributions are concluded and insight into our approach is provided, noting future extensions of our work, contributions of our research and limitations.

# Chapter 2

# Deep Learning for Computer Vision

In this chapter, the research domains present in this thesis and background work relating to each of the contributions are introduced. Introducing computer vision techniques, evaluation metrics for our approaches and the underlying algorithms and motivations behind each of the deep learning based methods. Artificially intelligent systems have seen a boom in recent years. With the advent of modern Graphics Processing Unit (GPU) based machines and algorithmic development, systems are easier to train, take less time to train and have more predictive power than previously capable. The number of applications is very broad and spans across almost all sectors, such as medicine and agriculture. The ability to traverse through large quantities of data and noise to arrive at a solution is accelerating innovation in all areas. Traditionally, artificial intelligence was used to describe intelligent machines, more recently it is used as an umbrella term to encompass many algorithms and architectures such as those contained within the machine and deep learning domains. Generally, learning based systems can be split into three categories:

1. Supervised learning (Learning outputs) - Data with corresponding labels is passed to the learning system to teach it to predict the correct labels. Through many iterations the models try to converge to an optimal solution minimising the error between the initial labels and predicted ones. This type of learning is to calculate outputs for a given input.

2. Unsupervised learning (Learning patterns) - Unlike supervised approaches, the corresponding labels are hidden to the learning system. The aim of the unsupervised approach is to find patterns useful for clustering or associating

the data with other examples. This type of learning is to discover patterns for or within a given input.

3. Reinforcement Learning (Learning actions) - Algorithms try to minimise the cost of reaching an end goal given an initial starting state that may have multiple viable solutions. For each transition within a state, the algorithm can be awarded when it guesses the correct answer and subsequently can optimise to learn a general solution for any given state. This type of learning is to learn actions and responses in series to reach a goal.

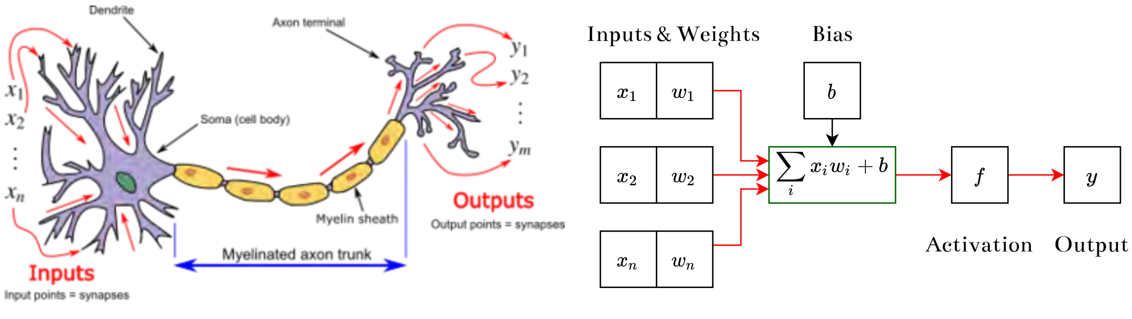
Beyond these three definitions exist hybrid modals such as semi-supervised learning, that is a hybrid of unsupervised and supervised learning problems, typically in this paradigm few labelled examples and many unlabelled examples exist, with the objective being to label the unlabelled data from the few examples given and unlabelled inputs. In the following section, only the application of supervised approaches is considered. Supervised learning problems are data driven and given one architecture the data can easily be modified to change the application domain. This has enabled single advancements in object detection to benefit a wide variety of industries and ensures results model data not experience. A few common concepts, algorithms, and architectures are introduced below frequently used in Convolutional Neural Network (CNN) based object detectors and some common baseline datasets used to compare them.

The organisation for the section is as follows. The basis of deep learning for computer vision is introduced, including the terminology, common algorithms and concepts utilised in most of the approaches and evaluation metrics used to benchmark state-of-the-art research against previous works. The datasets used in the computer vision domain are then introduced, that have been instrumental in driving the development of generalised computer vision detector and tracking frameworks. Each dataset within the computer vision community is specific in how they structure files, how they formulate each problem, and what evaluation metrics they use to best gauge overall performance.

The core components of this thesis are, detection, tracking, and analysis through visual inspection (phenotyping) of fruit for the benefit of applied horticultural systems. To provide a base of all knowledge relating to this domain, the subsequent sections are broken up as stated. First, an introduction to the State-Of-The-Art (SOTA) advancements made by deep learning based object detectors and the key insights at each stage in Section 2.6 and Section 2.7. Next an extension of the detection work for tracking detections across multiple frames is presented in Section 2.8, this enables the use of systems online and development of many more systems that detection alone cannot benefit. Key papers in the space are introduced that have achieved state-of-the-art and are commonly accredited with some of the most impactful advancements in the research domain. Note that all illustrations in the following sections focus on specific architecture contributions rather than specific convolutional network designs, for accurate network diagrams, layer sizes, and implementation details, please refer to each paper directly cited.

## 2.1 Neural Networks

Neural Networks (NNs) are a family of functions that are parameterised by the weights of the network and are interconnected to multiple other neurons. NNs with at least one hidden layer are universal approximators (Cybenko, 1989), any continuous function can be modelled as a combination of the family of neurons weights and connectivity. In practice, there are NNs that are much deeper than a single or multi-neuron architecture. Intuitively, this is due to the ability to express functions as a combination of compact, independent, and smooth neurons in a network, which subsequently also benefits optimisation methods such as gradient descent making problems easier for the networks to learn and represent. Deeper or wider networks, however, do not change the representational power (ability to model input data) of a network, nor do they ensure better training or higher accuracies; the more weights in a network, the more capacity it has to overfit to a dataset expressing the noise instead of the relationship in the actual data. One of the most powerful features of NNs is the ability to model data transformations in a functional manner, converting floating point numbers well understood by machines via a combination of weights and activations for nonlinearity to the desired output. A single neuron, as shown in Figure 2.1, consists of inputs $x$, weights $w$, bias $b$, an activation function $f$ and finally the output function $y$.



Biological Neuron

Artificial Neuron

Figure 2.1: A drawing of a biological neuron (left) and an artificial neuron used in machine learning (right).

While the anecdote of comparing biological neurons and artificial ones used an Artifical Neural Network (ANN) is largely incorrect due to the complexity and additional processes at play with the former, it intuitively describes a similar process of data flow. In a biological neuron, the cell body receives inputs from its dendrites and produces outputs along its myelinated axon trunk to the axon terminal. Similarly, an artificial neuron takes a digital input $x_n$, which is typically represented as a floating point value such that of image data or tabular features, and interacts multiplicatively with the weight $w_n$ (dot product), combines them linearly with the bias $b$ and applies the nonlinearity $f$ to get the output $y$. The mathematical representation of an artificial neuron output is given in Equation 2.1.

$$y = f\left(\sum_i x_i w_i + b\right) \tag{2.1}$$

The activation functions are key to introduce nonlinearity to the neurons and when connected ensures problems such that of the XOR problem (a nonlinear problem where output given two binary inputs should be false if equal and true if not equal) is solvable. Something that had previously led to a period after 1969 which was named the *AI-Winter*, when the authors Minsky and Papert (1969) had shown that two neurons could not solve the XOR problem given unlimited training time. ANNs are architectures of these neurons, shown in Figure 2.2.

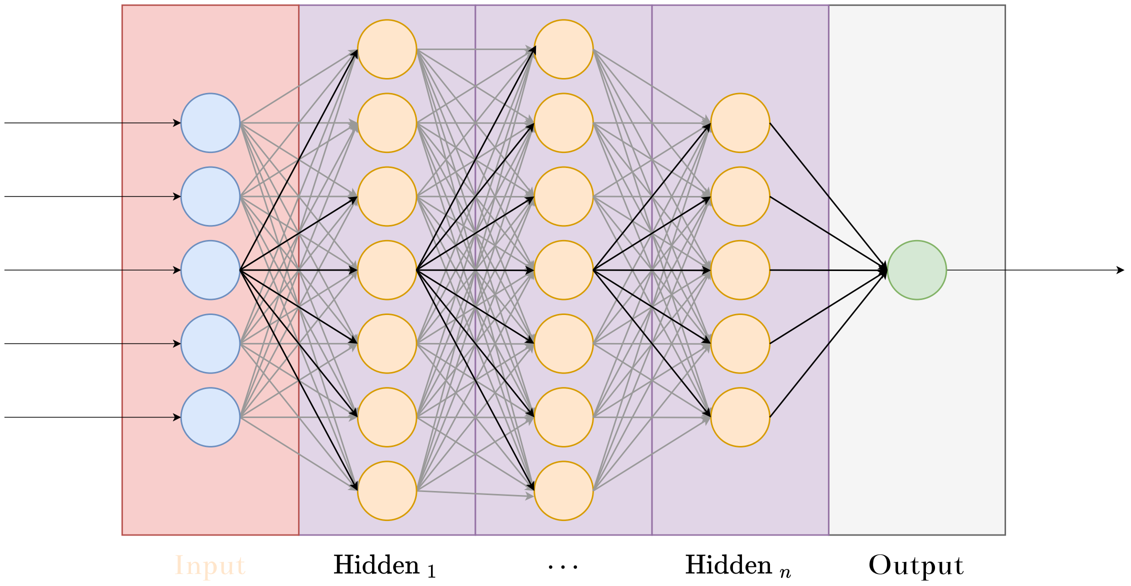


Figure 2.2: Artificial neural network (architecture of neurons).

The neurons are organised in layers, with any number of them chosen as a hyperparameter, usually tied to the problem requirements. The input layer (red) is the first layer that is directly connected to the input data. The last one, usually named the output layer (grey), is the final step in generating the network prediction. The hidden layers (purple) are in-between the two layers and produce the signal necessary to generate the output signal. Each neuron in each layer has its own weight $w_i$, which represents the contribution to the approximated total function. The learning process is designed to fine-tune each neuron weight in order to achieve the desired result as indicated by the training data. In this basic example, the layers have full connectivity but, in practise, can be connected by any arbitrary design.

## 2.2 Activation Functions

As mentioned above, a breakthrough for NNs was the introduction of activation functions $f$, enabling non-linearity in a NN. The activation function defines the output of any given node in an NN architecture, essentially deciding if the output should be used or not, likewise to a digital signal of 0 (off) to 1 (on), suppressing irrelevant data and presenting useful data. Without activation functions, NNs are akin to linear regressors. Common activation functions are presented below in context as to the usage and innovation, allowing deep networks to be trained and simply defined with multiple outputs.

### 2.2.1 Linear

The linear activation function, shown in Equation 2.2 and Figure 2.3, is commonly referred to as an identity function.

$$f(x) = x \tag{2.2}$$

Its use in NNs is limited due to the input being mapped to the output continuously, the derivative is a constant and has no discriminatory power to the input. The size or depth of a network of nodes is irrelevant if the activation function is always linear, as each node is a linear function of the last, essentially reducing the number of nodes to one.

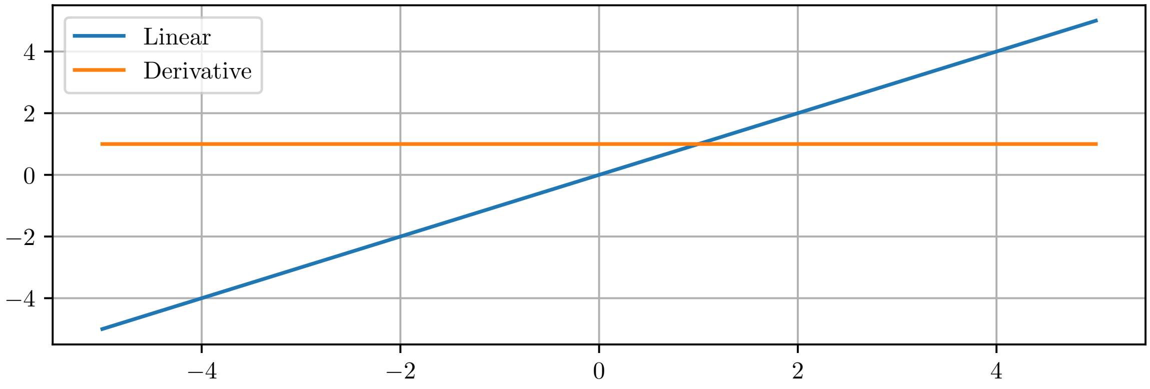


Figure 2.3: Linear activation function (identity) and its derivative.

### 2.2.2 Logistic

The logistic activation function, shown in Equation 2.3 and Figure 2.4, is commonly referred to as the sigmoid activation function.

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

It maps any real number input to between 0 and 1, where the larger the input, the closer the output to 1 and vice-versa. Unlike the linear activation function, the derivative is now related to the input value, enabling back propagation and training of the neural networks. Consequent nodes can now formulate and minimise the error for nonlinear (non-trivial) problems. It is use case in models is typically related to probability as an output due to the range.

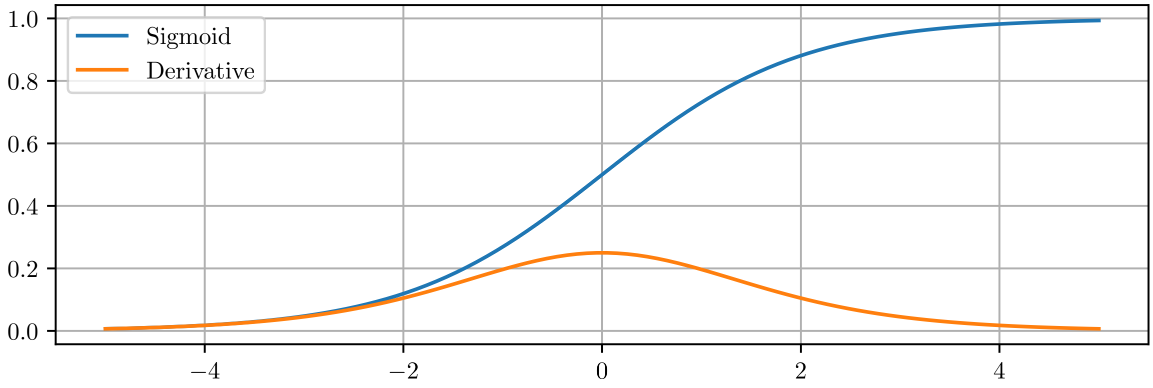


Figure 2.4: Logistic activation function (sigmoid) and its derivative.

The limitations of sigmoid are that the gradient less than $-3$ or greater than 3 is small, therefore training can suffer from the vanishing gradient issue, where large variations of the input yield small changes in the activation function output. When many nodes are stacked, by the chain rule the derivatives of each layer are multiplied, resulting in an exponentially decreasing (vanishing) gradient.

### 2.2.3 Softmax

The softmax activation function, shown in Equation 2.4 and Figure 2.5, is similar to the logistic activation function, and is often referred to as a combinatoric sigmoid activation function.

$$f(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_j}} \text{ for } i = 1, \ldots, K \tag{2.4}$$

The main difference being, logistic activation functions map any real number input to between 0 and 1, resulting in a total sum of probabilities that can exceed 1, whereas softmax ensures all of the probabilities total 1. The relative probabilities are used to classify examples, providing the probabilities of belonging to multiple classes.

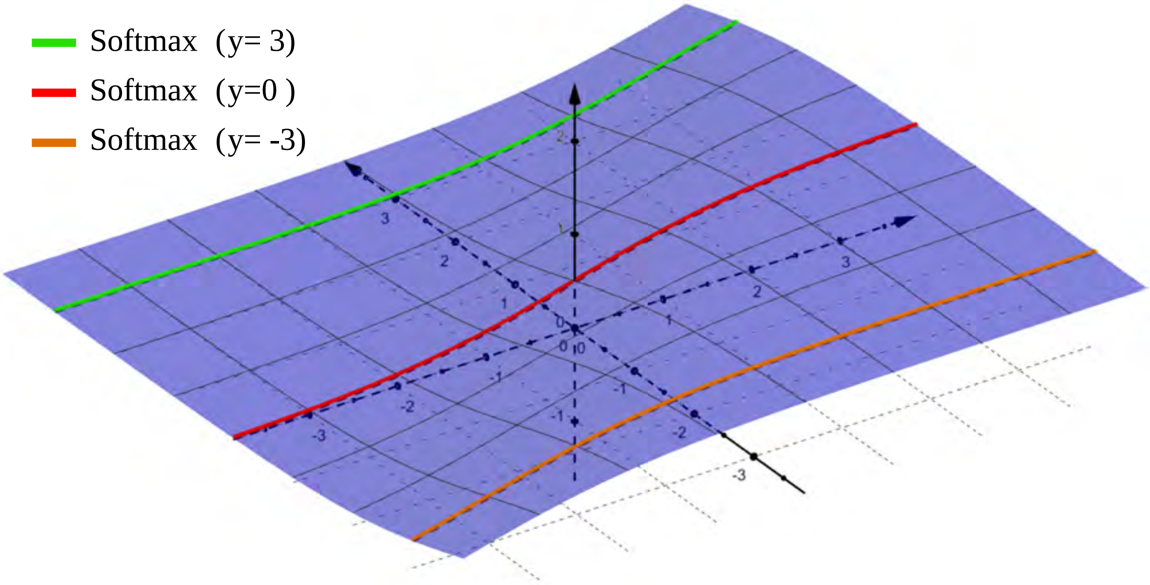


Figure 2.5: Softmax activation function with shifted $y$ intercept.

From Figure 2.5 it can be seen that for $y = 0$ the intercept is the original logistic activation function, however in the negative direction larger input values are required to result in the same output, and for the positive direction smaller input values result in the same output values. Unlike a logistic function where the midpoint 0 is shifted depending on the magnitude of other values in the input vector.

### 2.2.4 Rectified Linear Unit (ReLU)

The Rectified Linear Unit (ReLU) activation function shown in Equation 2.5 and Figure 2.6, despite its linearity for positive values it has a derivative function, remains non-linear and due to its simplicity it is computationally efficient to implement.

$$f(x) = \max(0, x) \tag{2.5}$$

ReLU is commonly used in deep neural network applications, returns 0 for negative inputs and returns the identity mapping likewise to the linear activation function for positive inputs. For values at 0 or less than 0 it is non-differentiable.

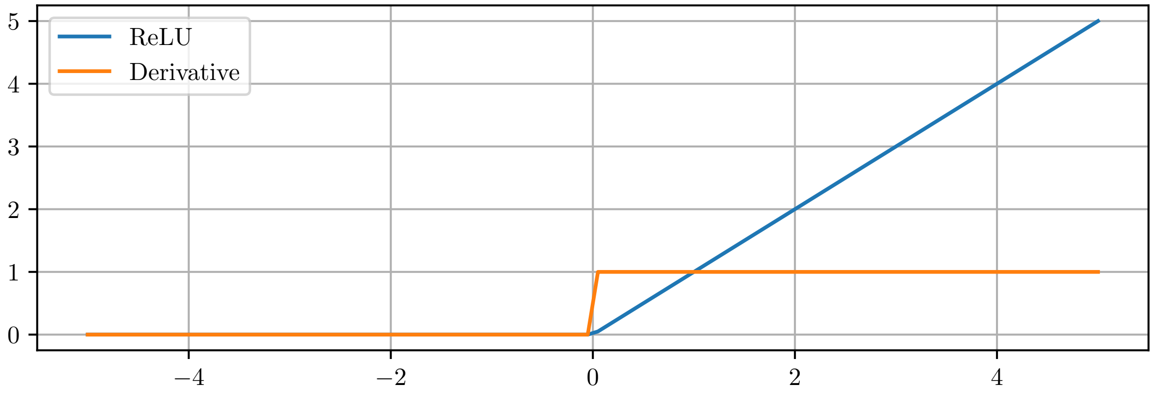


Figure 2.6: ReLU and its derivative.

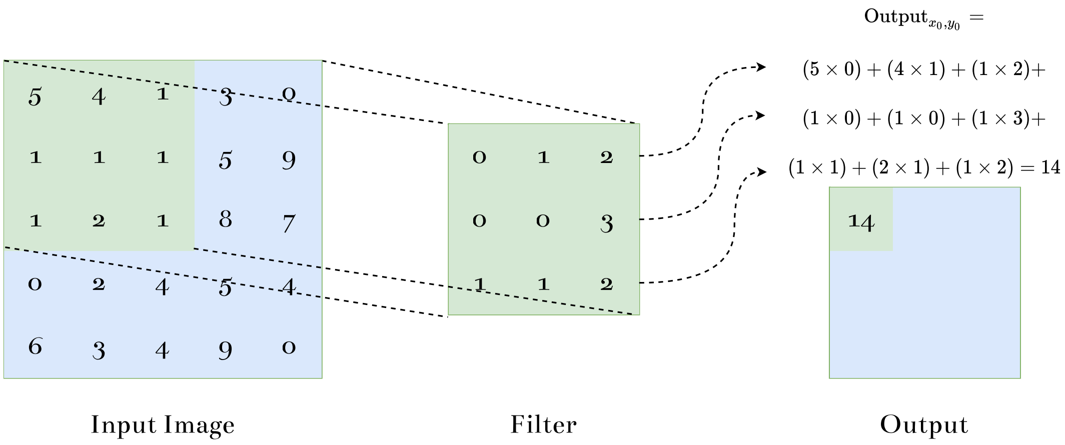
This characteristic of the ReLU activation function can result in some *dead* neurons in a NN, due to the gradients where input is negative the activation function is equal to 0. Resulting in no contribution to the NN architecture. An alternative is commonly used that is named Leaky ReLU, where negative values are mapped by $0.01x$, resulting in small gradients but ones that are still optimised.

## 2.3 Convolutional Neural Networks

A CNN is a type of deep learning model that typically comprises convolutional layers, pooling layers, activation functions, fully connected layers, normalisation, regularisation and loss functions. In a typical CNN architecture, each convolutional layer is followed by an activation function that increases the nonlinearity of the output, such as that of the ReLU activation function (Agarap, 2018), then a pooling layer, pooling areas via operations such as max, min, average or sum, then one or more convolutional layers, and finally one or more fully connected layer. These components are typically stacked horizontally, decreasing the resolution of the output at each stage.

A characteristic that sets apart CNNs from a regular, fully connected neural network (specifically, for image data) is taking into account the structure and spatial relationship of the images and pixels while processing them. To apply a standard NN architecture to images would result in a huge number of parameters, for a single RGB image with a width of 1920 and height 1080 results in $1920 \times 1080 \times 3 = 6,220,800$

input parameters, with little benefit as each neuron would only be connected to a small part of the input. The receptive field of an NN is defined as the proportion of input space passed to a node. CNNs introduced filters which slide over the input image (stride), shown in Figure 2.7, to reduce the spatial dimensions and parameters while keeping features from the original image. Reducing the number of parameters required in the network without loosing high-level features, each layer node now shares multiple parts of the input, and the receptive field essentially allows it to ignore all pixels that are outside the sliding window. Unlike classical approaches the filters are not hardcoded to extract texture or edges, but they are weights (nodes) that are learned, and the weights are shared between nodes as shown in Figure 2.7, as the weights of the filter remain unchanged when sliding over the input (parameter sharing).



Output$_{x_0,y_0}$ =
$(5 \times 0) + (4 \times 1) + (1 \times 2) +$
$(1 \times 0) + (1 \times 0) + (1 \times 3) +$
$(1 \times 1) + (2 \times 1) + (1 \times 2) = 14$
5 4 1 3 0
1 1 1 5 9
1 2 1 8 7
0 2 4 5 4
6 3 4 9 0
0 1 2
0 0 3
1 1 2
14
Input Image
Filter
Output

Figure 2.7: CNN 3 × 3 filter convolution with a stride of 1.

### 2.3.1 Bounding Box

Bounding boxes are used in object detectors and tracking systems to define where an object resides in an image, what class it belongs to, and for predictions, the confidence of the network that the object resides at that location. Bounding boxes are typically represented as four values, dependent on the technique used, traditionally the coordinates $x, y, w, h$ represent the object. Where $x$ and $y$ are the coordinates of the top left bounding box of the object and $w$ and $h$ are the width and height of the box. For classification networks patches of image are passed as an input, however for object detection architectures the whole of the image is passed and thus bounding boxes are required to define the position of the object in the image, for prediction purposes and for calculating the loss to the GT data object. Bounding boxes are the method used for the extension from single-object predictions to multi-object per-image and to enable the location of fruit to be calculated for automated harvesting.

### 2.3.2 Instance Segmentation

Bounding boxes are one of the methods to represent an object occupancy within an image, they generalise the area in which an object is present by bounding it with a rectangular box. Segmentation based approaches go one step further and mask the input image with a integer mask, where a 0 value represents background and positive integers represent the presence of an object. Typically, the segmentation approaches can be split into two categories, semantic and instance. Semantic approaches provide all pixels within an image with a categorical pixel value from 0 to the number of classes. Whereas instance segmentation, approaches cover a subset of the number of pixels in an image and can overlap one another. In practise, the image masks can be represented in several ways, from closed polygons of $x$ and $y$ corners, or as binary images. It is typical that object detection frameworks formulate instance segmentation and bounding box approaches separate to semantic segmentation problems.

## 2.4 Objective Loss

So far, the inner processes in architectures based on NNs and CNNs have been detailed. The objective of each is to minimise the prediction error output from the networks to the GT examples, through backgropagation shifting the learnt weights of a network by a magnitude and direction closer to an optimal solution. Supervised learning attempts to predict the transformation from an input to an output. This section introduces the loss functions at the core of the measurement of the error within CNNs.

### 2.4.1 L$_1$ Loss

The L$_1$ Loss Function in Equation 2.6 is used to minimise the error between the network predictions and the GT examples. It is defined as the sum of all absolute differences between the real (GT) and predicted (network output) values.

$$\text{L}_1 \text{ Loss} = \sum_{i=1}^{n} |y_t - \hat{y}| \tag{2.6}$$

### 2.4.2 L$_2$ Loss

The L$_2$ Loss Function in Equation 2.7 is used to minimise the error between network predictions and ground-truth examples. It is defined as the sum of all the squared differences between the real values (GT) and the predicted values (network output). L$_2$ loss is much more susceptible to outliers due to the squared error of $y_t - \hat{y}$.

$$\text{L}_2 \text{ Loss} = \sum_{i=1}^{n} (y_t - \hat{y})^2 \tag{2.7}$$

### 2.4.3 Cross Entropy Loss

Cross entropy loss criterion, shown in Equation 2.8 (binary), where $p \in [0, 1]$ is the estimated probability of a class, $y \in \{\pm 1\}$ denotes negative (background) and positive examples, $\alpha_t \in [0, 1]$ is a weighting factor (hyperparameter), and $p_t$ simplifies the definition of cross entropy. Note that the weighting factor is an extension of the typical binary Cross Entropy loss introduced in a later section 2.7.3 and commonly in practise. Cross entropy loss compares each prediction class probability from an activation function such that of logistic or softmax and penalises prediction probabilities that are far from the GT observation (typically 0 or 1, background or foreground). A lower value denotes a better fit to the model.

$$
\begin{aligned}
CrossEntropy(p, y) &= \begin{cases} -\alpha_t log(p) & \text{if } y = 1 \\ -\alpha_t log(1 - p) & \text{otherwise.} \end{cases} \\
p_t &= \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{otherwise.} \end{cases} \\
CrossEntropy(p_t) &= -\alpha_t log(p_t)
\end{aligned} \tag{2.8}
$$

### 2.4.4 Focal Loss

In some network architectures, there are substantially more background (negative) examples than that of foreground (positive) examples in the object predictions. With Cross Entropy loss, easily classified negative examples comprise the majority of the loss function and over influence gradient while training. The authors of T. Lin et al. (2017) introduced Focal Loss to address this problem, shown in Equation 2.9, Focal Loss adds a factor $(1 - p_t)^\gamma$ to the standard Cross Entropy criterion to reduce loss for well classified examples for higher hyperparameter values of $\gamma$. This ensures that the loss of misclassified examples has a higher contribution to the overall loss of the network, resulting in better class balance and performance for positive objects.

$$FocalLoss(p_t) = -\alpha_t(1 - p_t)^\gamma log(p_t) \tag{2.9}$$

## 2.5 Evaluation Methods

This section introduces common methods and metrics to evaluate object detection tasks in each of the later chapters.

### 2.5.1 Intersection Over Union

Intersection Over Union (IoU) is an evaluation metric for measuring the overlap between two boxes, as shown in Figure 2.8. IoU is typically used in object detection technologies for multiple reasons. It can be used as a measure of fitness in loss functions, to evaluate predictions compared to GT bounding boxes or to reduce the computational complexity of forward passes by removing highly overlapping proposals to name a few.

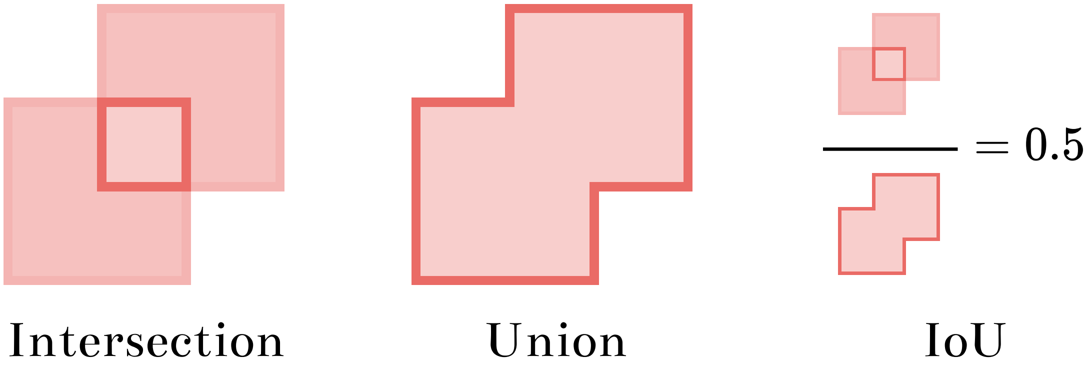


Figure 2.8: Intersection over union (IoU) of two bounding boxes.

### 2.5.2 Non-maximum Supression

Non-Maximum Suppression (NMS) is a technique that filters out superfluous bounding box proposals generated by a feedforward convolutional network. Typically, CNNs can generate thousands of predictions for images that contain a few examples. The reason why so many proposals are generated varies from one architecture to another, but usually it stems from the representation (such as bounding boxes) being valid over multiple areas of the image; for example, two bounding boxes over a single object could equally be as valid as one another, as shown in Figure 2.9.

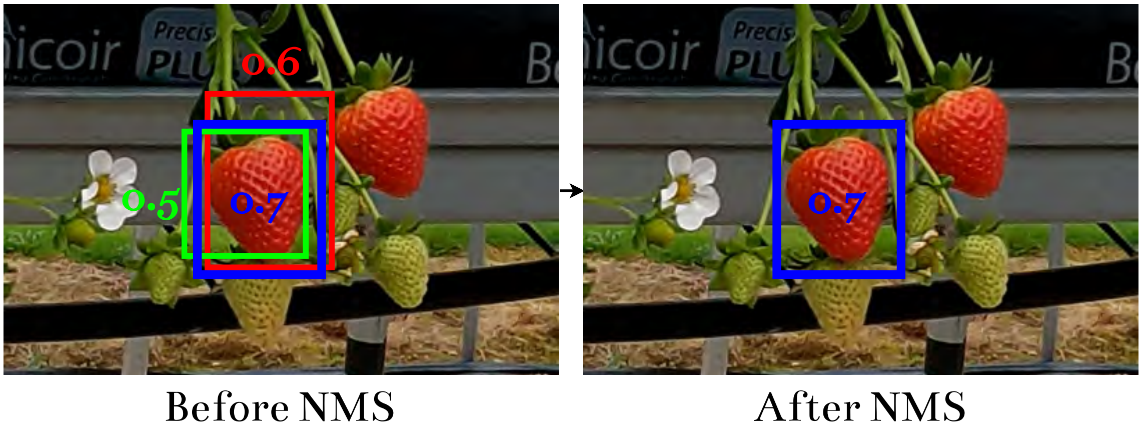


Figure 2.9: Non-maximum suppression (NMS) of three bounding boxes.

NMS is a bounding-box processing method that is used to reduce the number of proposals and improve the speed of detection. The basic NMS algorithm selects the highest confidence bounding box, adds it to a list, then compares the intersection over union with all other proposals, if the IoU exceeds a predefined threshold the compared proposal is deleted, this process is repeated until all proposals have been either removed or deleted.

### 2.5.3 Precision, Recall and $\mathbf{F_1}$

Most methods for object detection and tracking use either accuracy rate, precision, recall, $\mathrm{F}_1$, Average Precision (AP) or Mean Average Precision (mAP). The $\mathrm{F}_1$ score is the harmonic average of precision $P$ and recall, $R$ where precision is the number of true positives $T_P$ divided by the sum of true and false positives $F_P$, and recall is the number of true positives $T_P$ divided by the sum of true positives $T_P$ and false negatives $F_N$. True positives $T_P$ are correct detections, false positives $F_P$ are incorrect, false negatives $F_N$ denote when GT instances are not detected and finally true negatives $T_N$ are object instances in an image not labelled in the GT, for this reason many evaluation metrics do not consider it in the evaluation metric formularisation. With this definition, precision $P$ is the percentage of correct positive detections and recall $R$ is the percentage of true positives among the GT instances. The equation to compute the $\mathrm{F}_1$ score using precision and recall is presented in Equation 2.10. An object is considered correctly detected in the results when the predicted bounding boxes have an IoU of at least 0.5 (50% is a typical hyperparameter) with the GT annotation. Some older methods provide results using a value of 0.4 (40%). To allow a more accurate comparison to the experiments in this thesis we also present this value for comparison to works such as (Sa et al., 2016). Justification for the smaller IoU value is that objects in the respective data sets require less overlap than that of Common Objects in Context (COCO) and ImageNet.

$$P = \frac{T_P}{T_P + F_P}, R = \frac{T_P}{T_P + F_N}, F_1 = 2 \cdot \frac{P \cdot R}{P + R} \tag{2.10}$$

### 2.5.4 Average and Mean Average Precision

The AP and mAP evaluation metrics are the average precision per class (also referred to as Area Under Curve (AUC) in other domains of the precision and recall curve) and over all classes, respectively. For a IoU threshold, AP  summarises the corresponding precision recall curve, where higher values represent better detectors. The precision at each recall level $p(r)$ is interpolated $pinterp(r)$ by taking the maximum precision measured where the corresponding recall exceeds, $r$ shown in Equation 2.11. In the COCO (T.-Y. Lin, Maire et al., 2014a) challenge, a 101-point interpolated AP definition is used for a given IoU threshold, shown in Equation 2.12. COCO uses the average mAP  of IoU thresholds as the primary evaluation metric. The calculation of this metric is usually done in two steps: first as shown in Equation 2.13 AP  is calculated at different thresholds of IoU (the default is 10 thresholds from 0.5 to 0.95 with a step size of 0.05) and their average is used to get the AP  of that class, second as in Equation 2.14 calculate mAP  by taking an average over all classes (default is 2 for our classes). The average of classes over averages of AP  at different IoU thresholds is done, so the final single value metric is more robust to localisation errors. The metrics $\mathrm{mAP}_{.4}$  and $\mathrm{mAP}_{.5}$  are used in our experiments to compare against other works and measure the accuracy of our detector.

$$\mathrm{pinterp}(r) = \max_{\tilde{r}:\tilde{r}\geq r} p(r) \tag{2.11}$$

$$\mathrm{AP}[\mathrm{class}, \mathrm{iou}] = \frac{1}{101} \sum_{r\in(0,0.01,\ldots,1)} \mathrm{pinterp}(r) \tag{2.12}$$

$$\mathrm{AP}[\mathrm{class}] = \frac{1}{10} \sum_{\mathrm{iou}\in(0.5,0.05,\ldots,0.95)} AP[\mathrm{class}, \mathrm{iou}] \tag{2.13}$$

$$\mathrm{mAP} = \frac{1}{2} \sum_{\mathrm{class}\in(0,1)} AP[class] \tag{2.14}$$

### 2.5.5 Computer Vision Benchmarks

This section focuses on the data that has driven advancements of deep learning based object detectors and trackers. The datasets are provided as open source for validation and to benchmark progress against the state-of-the-art. The application of agnostic approaches is the main focus in this section as they have the widest use in research, however it is worthwhile noting that many more datasets are available. The open-source datasets do not only provide images with annotations, but they have also defined evaluation metrics and data formats that have since become an open standard for object detectors and trackers.

The PASCAL Visual Object Classes Challenge (VOC) was introduced in 2005 (Mark Everingham et al., 2010) and had updates applied to it throughout the years 2006 to 2012 with the latest version consisting of 20 classes and 11,540 images containing 27,450 ROIs annotated objects and 6,929 segmentations. This data set has been widely used as a reference for object detection, semantic segmentation, and classification tasks, divided into 5717 training images and 5823 validation images containing roughly the same number of objects. It contains classes spanning over multiple object types from people, animals (birds, cats, cows, dogs, horses, sheep) to vehicles (aeroplanes, bicycles, boats, buses, cars, motorbikes, trains) and indoor objects (bottles, chairs, dining tables, potted plants, sofas, televisions). The interpolated AP evaluation metric was introduced in VOC, it is a single number used to summarise the Precision-Recall curve, similar to AUC and approximately equal if precision is interpolated by constant segments. It does this by averaging precision at 11 recall levels from 0 to 1 with a step size of 0.1. Instead of directly using the precision at each recall point, AP is calculated by taking the maximum precision where the recall is greater than the current recall step. For all of the predictions made they are only considered in the evaluation metric if the IoU exceeds 0.5, where overlaps occur the highest IoU is used. The mean AP metric $\mathrm{mAP}_{0.5}$ is used to describe the overall performance over all classes.

ImageNet, introduced in 2009 (Deng et al., 2009), introduced an image classification data set based on the WordNet lexical database for English words (Miller, 1995).

The original revision of the dataset took the approach to populate each synset of the WordNet database with 50-100 images of the observation. The most recent and widely used revision is the 2012-2017 ImageNet Large Scale Visual Recognition Challenge dataset (ILSVRC), which spans 1000 object classes and contains 1,281,167 training images, 50,000 validation images, and 100,000 test images. Transfer learning is a field within deep learning where the objective is to distil knowledge from one network with a large dataset into another network. ImageNet provided a dataset containing a huge amount of variation that has been utilised in practise to provide a set of stable and generalised weights for training new neural algorithms since its inception in 2009.

COCO is a large-scale object detection, segmentation, and captioning data set (T.-Y. Lin, Maire et al., 2014a). It has multiple challenges with a common dataset format for training object detection (bounding box and keypoint), object segmentation (instance), and captioning networks. The detection dataset contains more than 200,000 images with 80 classes split over training and validation containing more than 500,000 annotated objects. Similar to VOC there is an evaluation metric for benchmarking trained networks on the dataset. mAP was calculated by averaging AP over the 80 categories of objects and the 10 IoU thresholds from 0.5 to 0.95 with a step size of 0.05 for the challenge COCO 2017, extending the evaluation metric VOC which used a single threshold. The authors of the challenge believed that this approach to evaluation rewards detectors with better localisation. Summarised, the AP is calculated for a IoU threshold of 0.5 for each class i (the precision at every recall value , 0 to 1 in steps of 0.01), then it is repeated for IoU thresholds of 0.55 to 0.9 in 0.05 steps, and finally an average is taken over all the 80 classes and all the 10 thresholds to get the mAP value used to benchmark the challenge predictions. The authors also introduce this metric for multiple scales, for evaluating detections based on the object size inside the image to observe the AP for objects of varying sizes. The authors have defined small objects with those having an area less than $32^2$ pixels, medium objects as those having an area between $32^2$ and $96^2$ and large objects with an area greater than $96^2$.

With the advent of object classification and detection dataset challenges posed to researchers for the advancement of neural architectures, data was needed for the tracking domain, as it requires a different formulation. Tracking identities across images is the high-level task of object tracking frameworks, and the open datasets described thus far have disjoint, unrelated identities in almost all examples. Multi-Object Tracking (MOT) is a dataset challenge that was introduced in 2015 (Leal-Taixé, Anton Milan, Reid et al., 2015) with the aim of addressing this problem by introducing a dataset to benchmark tracking methods and providing a set of evaluation methods. The dataset has received updates since the inception in 2015, 2016, 2017 and 2020. Each dataset in the challenge is a sequence of images taken from videos, with GT bounding boxes with identities.

## 2.6 Multi Stage Architectures

CNNs translate the image information into feature maps that can model different representations of the data given appropriate loss functions. In order to detect bounding boxes or segment pixels belonging to a specific instance, the network needs the context of separable regions. In this section, multistage architectures are presented that have proposed regions via methods such as selective search or dedicated region proposal networks and perform convolutions on the proposed regions to classify, further localise, and segment each region independently, resulting in detected objects. The multistage refers to the fact that they CNNs cannot directly infer separable regions on their own. End-to-end solutions are introduced in Section 2.7 that can directly separate regions within feature maps that subsequently simplify training and inference.

### 2.6.1 R-CNN

Region-based Convolutional Neural Networks (R-CNNs) were introduced in 2013 (Ross B. Girshick et al., 2013) which used CNNs to extract features from region proposals instead of relying on hand-crafted low-level features such as edges, gradients, and corners to detect objects. There are three main components to the R-CNN algorithm: (1) Region Proposals, (2) Convolutional Neural Network and, (3) Region classification, shown in Figure 2.10.

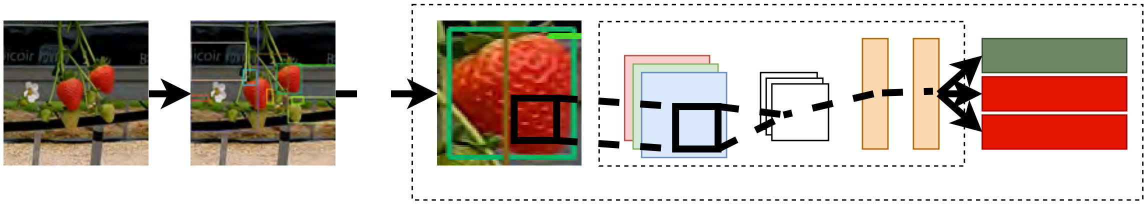
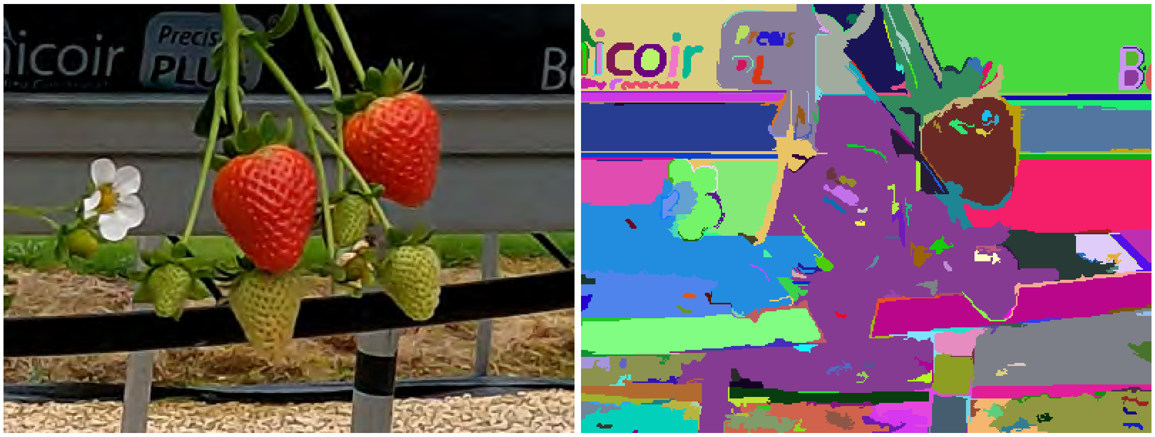


Figure 2.10: R-CNN extracting CNN features from Selective Search region proposals and classifying with support vector machines.

Region proposals could be generated many ways, the purpose of which is to take an image and predict likely regions that contain objects of interest, either in the form of a bounding box or per-pixel. The R-CNN paper utilises Selective Search, a greedy per-pixel superpixel region proposal algorithm based on merging superpixels generated from low-level features such as edges, to generate  2000 proposals. An example is shown in Figure 2.11. Selective search merges superpixels in a hierarchy of similarity measures, (1) colour similarity, (2) texture similarity (Gaussian derivatives), (3) size similarity, (4) shape similarity, and (5) a combination of the aforementioned measures. Each proposal is then set to a fixed input size and an CNN is used to extract fixed length feature vectors. For each feature, vector classification is run in the form of category specific linear Support Vector Machines (SVMs) to generate the final predictions.



(**a**) Input

(**b**) Selective Search Segmentation

Figure 2.11: Selective Search algorithm applied to the Strawberry in image (2.11a) and visualised in (2.11b) with the parameters $\sigma = 0.5$, $K = 500$, $min = 50$.

### 2.6.2 Fast R-CNN

The problem with R-CNN is that it is slow to train and slow to test, despite using the method deemed fast mode of selective search. Fast R-CNN was introduced in 2015 (Ross B. Girshick, 2015) to address this issue. It focused on methods to improve the overall speed of the algorithm. Mainly, reducing the number of forward passes of the feature extractor to a single pass and instead extracting the region-specific features through a process called Region of Interest (RoI) pooling and, training with multitask loss for localisation and classification of predictions against GT. RoI pooling is the process of cropping regions of interest from the convolutional feature network, which is usually a collection of deep vertically connected convolutions and pooling operations such as VGG (Simonyan and Zisserman, 2015), and converting them into a fixed-length feature map as shown in Figure 2.12. Usually RoI pooling does this by taking the region proposal, scaling it to the size of the feature map, and divides the segment of the RoI feature map into $W \times H$ blocks (hyperparameters of the pooling layer RoI) taking the maximum value of each block from the segment.

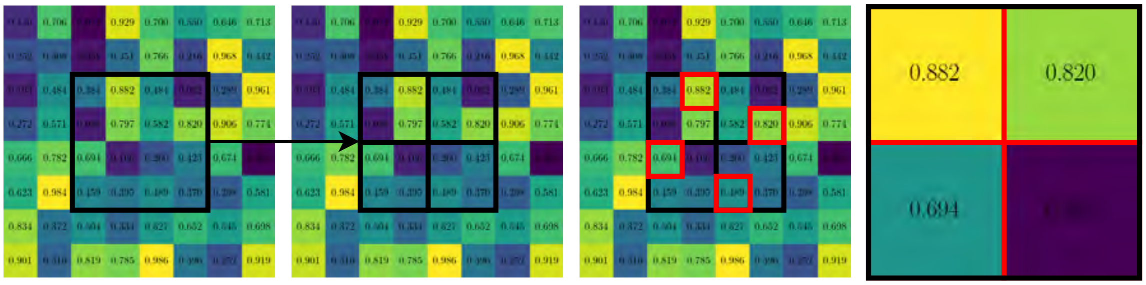


Figure 2.12: RoI Pooling generating fixed size feature maps from region proposals.

The way it achieved the speed and accuracy increase was two-fold, firstly, instead of doing multiple forward passes of the CNN to generate the fixed length feature vectors for each proposal SVM classification, they restructured the network to perform a single forward pass. This reduced the complexity of the train and test time from $O(N)$ to $O(1)$. The single forward pass is run on the entire image and a fixed length feature map is generated; at this stage, the fixed length feature vectors can be extracted from the feature map for each region proposal. At this point, each proposal is still generated with the selective search algorithm. However, the final predicted bounding boxes are generated by taking the fixed-length feature maps for each location and passing them through fully connected layers (network heads), soft max classification, and bounding box regression for each bounding box corner. They also adjusted the loss function of the network to multitask loss (L1 loss) to regress to the final candidate locations, simplifying the end-to-end training. The architecture changes of the Fast R-CNN compared to the R-CNN are shown in Figure 2.13. Similarly to R-CNN, the performance was constrained by the initial region proposals, in this case selective search.

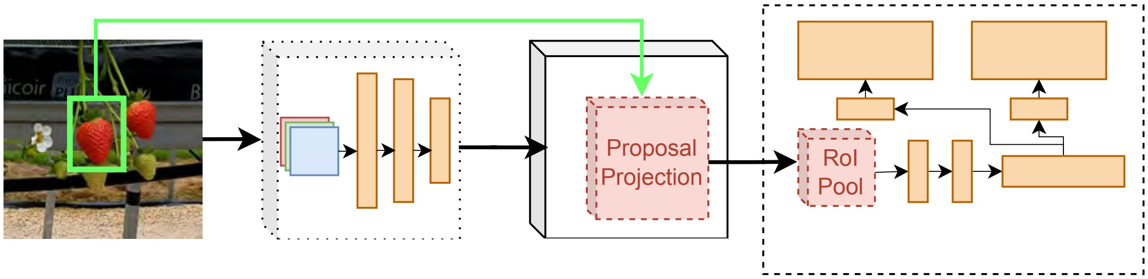


Figure 2.13: Fast R-CNN extracting RoI features from an image feature map, pooling and classifying with multitask loss.

### 2.6.3 Faster R-CNN

Several improvements were made over the next two years, from the R-CNN base to the introduction of Faster R-CNN (Ren, K. He, Ross B. Girshick et al., 2015b), which focused on improving the accuracy and speed of R-CNN. Faster R-CNN adopts the improvements made by Fast R-CNN and adds a Region Proposal Network (RPN) to extract object regions from the feature map instead of using the results of algorithms such as EdgeBoxes (Zitnick and Dollár, 2014) or Selective Search (Uijlings et al., 2013) shown in Figure 2.14.

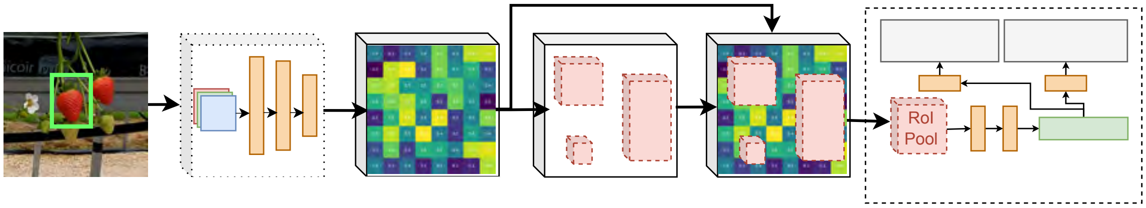


Figure 2.14: Faster R-CNN generating regions of interest from a separate network head, pooling and classifying with multitask loss.

Selective search greedily merged superpixels based on low-level features and at the time EdgeBoxes (n approach for generating object bounding box proposals directly from edges) provided a better tradeoff between proposal quality and speed. As both region proposal algorithms are based on Central Processing Unit (CPU) implementations, Faster R-CNN looks to remove the bottleneck and encompass region proposals as part of the network. With the introduction of the RPN in Faster R-CNN, proposals are obtained nearly free (10ms per image) and are highly accurate proposals that outperforms methods based on CPU due to the implementation on GPU. Anchors were typically from this point defined at 3 different aspect ratios (wide, tall, square) at 3 different scales (small, medium, large) resulting in 9 anchors per window location, shown in Figure 2.15. Instead of using the precomputed region proposals, anchors were used in the second stage with an objectness score to learn the object positions for each possible location.

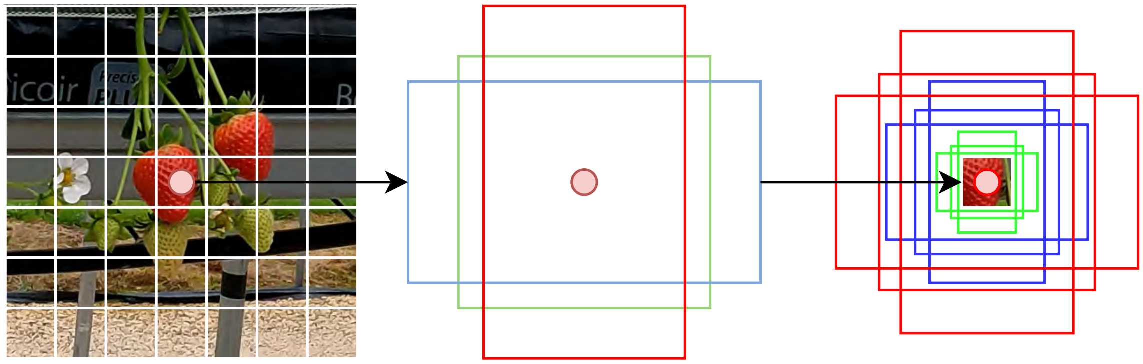


Figure 2.15: Anchor box region proposals applied over a sliding window. For visualisation an image is used in place of the feature map used in the RPN.

The RPN slides an $n \times n$ window over the feature map to predict object bounds and *objectness* (probability of the object belonging to the background or foreground) scores for each location and offset parameters to reshape each anchor to the final bounding box; the best candidates are the handcrafted anchor boxes with the highest IoU.

### 2.6.4 Mask R-CNN

R-CNN approaches to bounding-box object detection were introduced to address a reasonable number of candidate object regions. Mask R-CNN (K. He, Gkioxari et al., 2017) extended Faster R-CNN to allow attending to RoI on the feature maps more accurately using a new region proposal layer deemed RoIAlign, and further added a mask prediction branch for each RoI, leading to high-throughput instance segmentation and better accuracy. RoIPool first introduced in Fast R-CNN (Ross B. Girshick, 2015) quantises the regions of interest, rounding floating point values to decimal values in the result feature map (bounding boxes constrained to image coordinates); RoIAlign instead performs bilinear interpolation to compute exact values of input features instead; skipping quantisation all together (floating point image coordinates), a crucial architecture change that closed the gap between bounding box detection and the much harder task of instance segmentation. The difference between RoI pooling shown in Figure 2.12 and RoI align shown in Figure 2.16 ensures that no data is lost (red) from the region proposal and no additional data is gained (green).

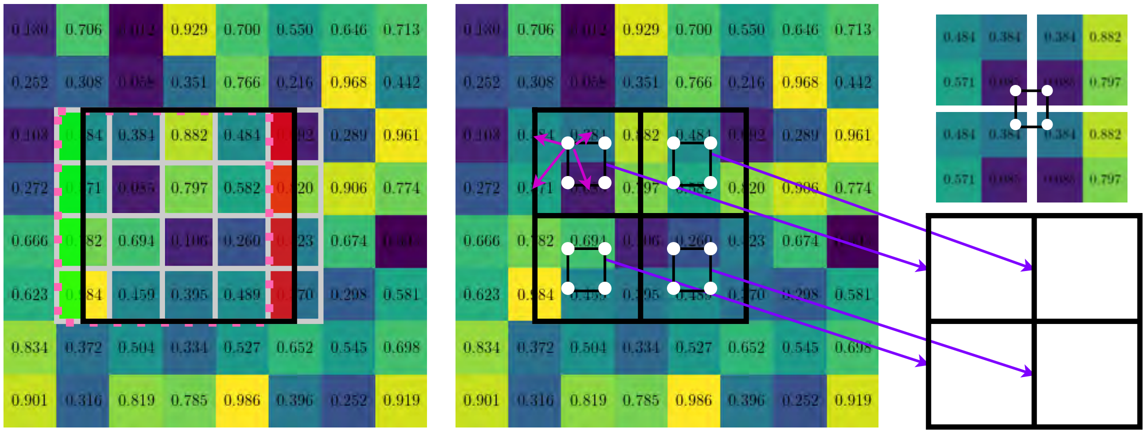


Figure 2.16: RoI pooling in Faster R-CNN vs RoI align in Mask R-CNN. RoI Pooling quantises the coordinates, whereas RoI align performs bilinear interpolation of the closest neighbours (pink) for four points within the proposal (white) which does not face the data loss (red) or gain (green) issue of quantised proposals in RoI pooling.

Generating instance segmentation masks for each proposal is performed in Mask R-CNN by another network consisting of convolutional layers after the RoI align stage, similarly to the classification and bounding box regression network heads in Faster

R-CNN. The architecture otherwise remains the same and is shown in Figure 2.17. In the original paper, they propose multiple instance segmentation heads consisting of different combinations of convolutional layers, and to better represent objects at multiple scales, but the premise remains the same.

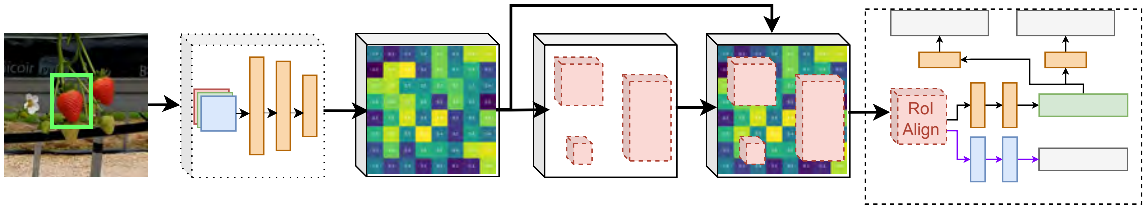


Figure 2.17: Mask R-CNN extension of Faster R-CNN adding an instance segmentation head for proposals and RoI align better capturing feature map data.

## 2.7 Single Stage Architectures

The above section focused on multistage architectures. Multi-stage architectures for object detection typically have two stages. The first stage is responsible for narrowing down the total number of proposed object locations, this is to filter out the majority of the background object proposals. Faster-RCNN introduced anchor boxes and used a RPN to filter out background samples. The second stage is usually comprised of a number of network heads. Separate networks that take each candidate location and feature map and either classify, regress bounding box offsets from the anchors or, in the case of Mask R-CNN, predict the binary mask for each candidate. Both of these stages ensure a good balance between the foreground (positive) examples while training/testing and the background (negative) samples. In single stage architectures, the first stage is removed and instead a larger set of initial candidate locations is used. In the following section, the advancements made by single-stage detectors are discussed, aiming to improve on accuracy, model simplicity, and inference speed.

### 2.7.1 YOLO

Removing the entire region proposal stage from the object detection pipeline was motivated by the need to simplify the implementation of deep neural networks for object detection, improve the inference speed, and increase the overall accuracy deriving complete object detection from the feature map. Similarly to the improvements seen in two-stage architectures from R-CNN to Faster R-CNN, working directly with the computed feature maps of convolutional network heads resulted in almost free (computationally) performance increases and simplified the overall network architecture, reducing the total number of steps required. You only look once (YOLO) (Redmon, Divvala et al., 2016) took this design pattern and restructured the problem from classification to regression.

Instead of classifying region proposals from a common feature extraction network such as VGG16, YOLO divides each image into grid cells and predicts the probabilities of objects that occupy the cell along with their respective bounding boxes with a custom combination of convolutional layers to reshape the input into the desired final shape as shown in Figure 2.18. Dividing the input image into an $S \times S$ grid of cells, where each cell predicts a single object and outputs $B$ boundary boxes $(x, y, w, h, c)$ with confidence scores $c$ (later referred to as objectness, that is 0 when no object exists otherwise confidence equivalence trained to IoU cost between predicted and GT bounding box) and $C$ the class probability that a class occupies the grid cell. Despite any number of bounding boxes chosen as a hyperparameter, only one object is predicted for each location. The resultant shape of the prediction YOLOs is $S \times S \times (B \times 4 + 1 + N)$ where $4 + 1$ denotes the vector of the boundary box $B$ and $N$ reflects the number of possible classes.

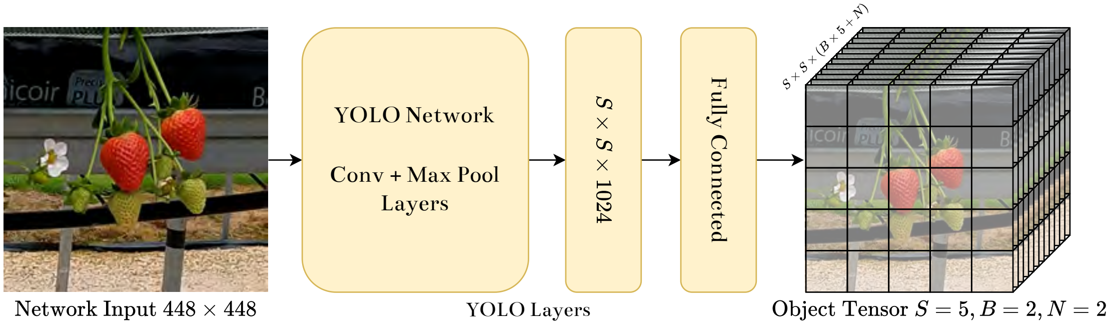
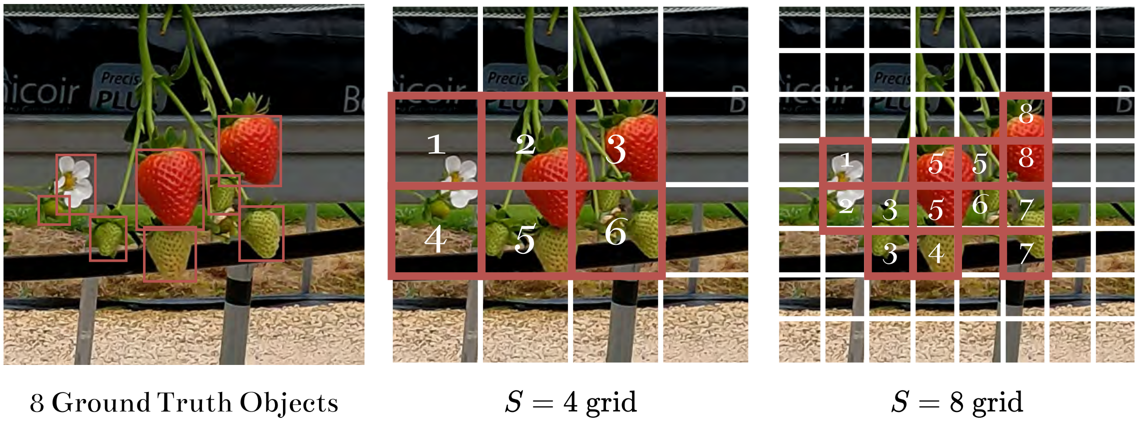


Figure 2.18: YOLO object detection formulisation, multiple convolutional and max pool layers are stacked to predict a $S \times S \times (B \times 4 + 1 + N)$ vector for multiple classes.

If the centre of a GT bounding box falls within one of the $S \times S$ grid boxes, then that cell is responsible for predicting $B$ bounding boxes offset from the cell position. Compared to two-stage architectures, this is a much simpler approach to region proposals than that of a dedicated network to filter out negative examples. A limitation of this YOLO architecture is that due to each grid cell only predicting a single object, any clustered objects with high intersection union or objects that are close to each other can only be detected by a very fine grid size and in some cases it is impossible to detect both objects. Figure 2.19 illustrates this concept. For each bounding box predicted per grid cell, the one that has the highest intersection union with the GT bounding box is selected as the candidate and the sum of the squared error between all overlapping predictions is the final loss. The sum of squared error loss is composed of the classification, localisation and finally the objectness score. In most cases the majority of cells contain negative examples, to counter this the authors use a weighting factor $\gamma$ to reduce the prediction power of negative examples.



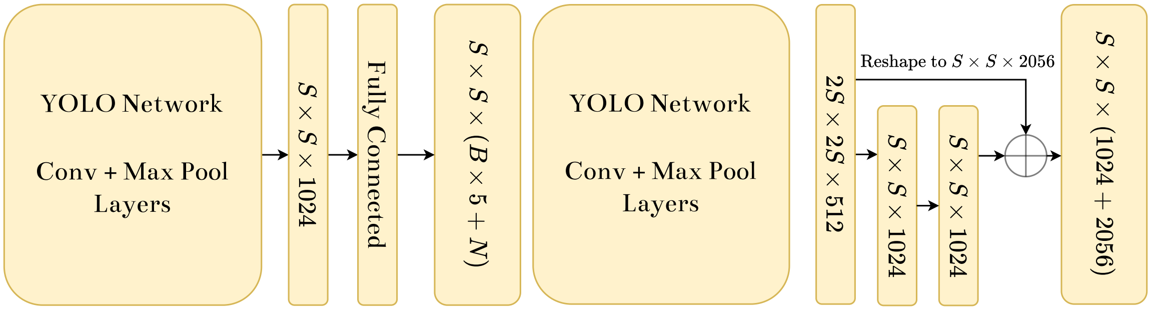
8 Ground Truth Objects

$S = 4$ grid

$S = 8$ grid

Figure 2.19: Limitations of YOLO, 8 GT objects exist, but depending on the grid size ($S \times S$) only 6 ($S = 4$) objects can be detected.

Compared to Faster R-CNN, both the bounding box and class are predicted in a single-stage and due to the simpler network architecture, YOLO can run much faster on the same hardware. Also, due to the end-to-end network, any final prediction can utilise the global image context rather than region specific areas pooled by RoI pool or RoI align, the authors attribute fewer false positives due to this. The paper notes that training YOLO was unstable due to the bounding box proposal strategy (naive guess for each grid cell), when applying to problems with multiple classes, it is likely the final bounding boxes will not share similar shapes. YOLO (v2) was introduced to address some of these concerns.



(**a**) YOLO Network Tail (v1)

(**b**) YOLO Network Tail (v2)

Figure 2.20: Difference in YOLO network version tails, introduced as pass-through in v2, the second iteration reshapes an earlier convolutional layer and concatenates with the last layer to increase spatial resolution of predictions.

One problem with anchor boxes is that a new hyperparameter is introduced, and when any new hyperparameter is introduced, there is room for optimisation. The authors note that the selection process for a number of anchor boxes at specific ratios is difficult. In the first version of the YOLO network early optimised specific boundary shapes that best fit objects in the training data but would have a steep gradient change when a dissimilar bounding shape was trained on resulting in unstable training. In Faster R-CNN, 9 anchor boxes were selected for 3 aspect ratios (tall, wide, square) and 3 scales (small, medium, large). However, the more anchor shapes that are present, the higher probability that the object shapes cannot converge to a single shape and the fewer dissimilar anchor box shapes in total results in diminishing accuracy returns when fitting to the training data. In the YOLO v2 paper, the authors use K-Means clustering to select both hyperparameters, and it converges to 5 anchor boxes of similar aspect ratios to that of other SOTA papers used at the time. Finally, the authors shuffle the input image size every 10 batches in multiples of 32 width and height values as a data augmentation approach. Application of all the mentioned approaches took the mean average precision from 63.4 to 78.6, maintaining real-time computation and outperforming previous approaches.

YOLO v3 introduced in (Redmon and Farhadi, 2018) brought many improvements to the overall architecture. The majority of the mentioned architectures, thus far, utilise a variant of a softmax classifier to predict class labels. This results in a single-class prediction for each location, grid cell, or region proposal depending on the architecture. YOLO v3 instead, separates the class predictions using a softmax function with multiple logistic regressors to predict the probability for all classes independently, allowing the overall class estimates to sum up to an arbitrarily large value compared to the softmax approach which bounds the total probability of each class to one. Following on from YOLO v2, this allowed multiple class labels to persist for each prediction and in the case of similar labels such as bus and truck allows better convergence.

### 2.7.2 Single Shot Detector (SSD)

Similarly to YOLO v1, the Single Shot Detector (SSD) was introduced in 2015 (W. Liu et al., 2015) to further simplify the architecture by removing the region proposal network stage and to work with the feature map more directly when assessing candidate locations, shown in Figure 2.21. Introduced after Faster R-CNN, it has accuracy similar to some two-stage detectors, outperforming the base Faster R-CNN configuration. The removal of the region proposal stage allowed the single shot detector model to run at real-time speeds compared to 7 frames per second of the standard, Faster R-CNN model and comparable to YOLO. Real-time is defined as any system capable of computing data faster than the rate data would be made available in a real scenario, in the case of object detection the real-time speed is 30 frames per second in line with the standard video format data rate. At the time of publication, it was the first real-time detection method that exceeded the 70% mean average precision benchmark on the VOC (2007) data set (M. Everingham et al., n.d.).

Overall, the SSD uses a similar architecture to how the RPN assessed candidate regions, a fixed number of anchor boxes is used for prediction but instead of the anchor boxes being used to pool features and evaluate a classifier in the SSD paper they simultaneously produce a score for each object category. Avoiding the requirement of embedding an RPN in the overall architecture. Region proposals are generated by dividing the image space into a grid $N$ by $M$ with $K$ total anchor boxes per

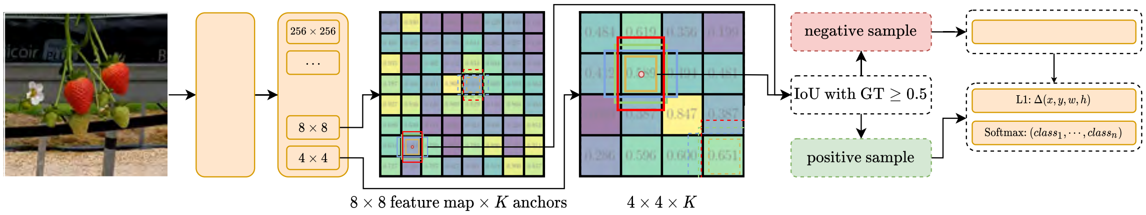


Figure 2.21: SSD feature map (from VGG-16) is distilled into several maps with gradually lower spatial dimensions for multiscale object detection. At each stage $K$ anchor boxes are considered in a $M \times N$ grid, ones that overlap a GT sample are propagated into the loss at a maximum 3:1 ratio with top performing negative samples. Note, feature map sizes are for illustration only.

grid centre, allowing the use of multiple anchor aspect ratios and sizes. As the SSD convolutional layers get deeper, the spatial resolution becomes smaller, and the resultant feature maps are used to detect larger objects, the process is shown in Figure 2.21, for example a feature map $16 \times 16$ with 3 different proposal anchors would equate to $16 \times 16 \times 3 = 768$ possible objects in the image (smaller objects) whereas a deeper feature map with a $4 \times 4$ resolution would equate to $4 \times 4 \times 3 = 48$ possible candidate covering a larger receptive region in the original image and subsequently capable of capturing much larger objects.

Each $M \times N$ feature map has a $3 \times 3$ classification convolutional filter applied to each anchor in each cell for predictions, for a total of $M \times N \times (K \times (C + 4))$ predictions where $C$ is the number of classes plus one for the negative examples and 4 denotes the length of bounding box deltas from static anchors to GT predictions. The hard negative mining stage ensures that an appropriate class balance is maintained in the loss function, otherwise the detector would be coerced into over-optimising for negative examples, resulting in poor performance for the target objectives. Unlike, YOLO which used K-means clustering to select initial anchor boxes per grid location, SSD instead selects 4 arbitrary shapes of different width and height ratios for the anchors. It also introduces some simple data augmentation approaches, shown in Figure 2.22, chosen randomly from (1) using the entire image, (2) a random intersection over union patch (10% to 90% in 20% steps) or (3) a random image patch and combined with horizontal flipping and photometric distortions, they obtain higher accuracy and more stable training with these methods.

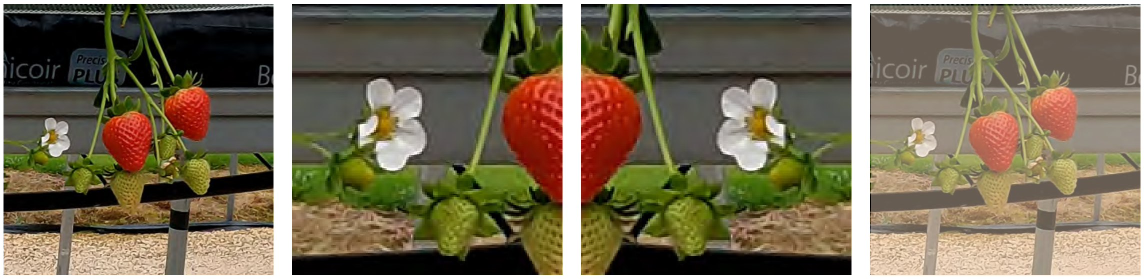


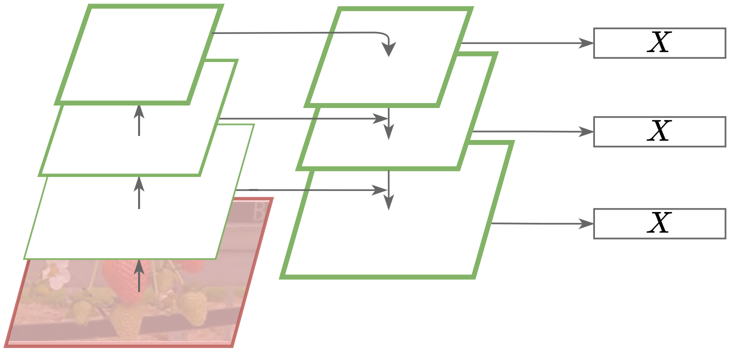
Figure 2.22: Any combination of the above four augmentation methods resulted in the final image and GT bounding boxes in single shot detector.

SSD is a fast single-stage object detector. The key contribution is the utilisation of multiple feature maps, an approach improved upon in future works discussed below in Section 2.7.3, with multiscale convolutional bounding box outputs skipping the region proposal network requirement all together. Their approach handles the class imbalance issue by rejecting numerous negative examples in the training stage, maintaining a hyperparameter ratio of samples at any given time. The approach is an efficient and simple use of convolutional layers within the network, and they add a minimal number of extra layers for semantic representation of the input image at different scales.

### 2.7.3 RetinaNet

RetinaNet was introduced in 2017 (T.-Y. Lin, Goyal et al., 2017), and achieved SOTA results on the COCO challenge, foregoing the two-stage architecture of Faster-RCNN for a single-stage architecture while maintaining high accuracy but improving the inference speed. RetinaNet achieves this by utilising a Feature Pyramid Network (FPN), originally introduced in T.-Y. Lin, Dollár et al. (n.d.) extracting multiple feature maps at different scales from the CNN backbone in a top-down pathway to construct semantically rich feature maps combined with the original feature map to add more meaningful localisation information, shown in Figure 2.23, and by introducing a new loss function deemed Focal Loss.

This addresses some problems faced in Faster R-CNN and Mask R-CNN with the RPN. in these architectures. The RPN only generates object proposals for a single



(**b**) Top Down Feature Maps

Figure 2.23: FPN where $X$ denotes neural network heads such as classification and regression of anchors and thicker outlines on the convolutional layers and $M$ denote semantic strength. A single scale is used in 2.23a from a pyramidal feature hierarchy (Fast, Semantic Information Lost), this is to speed up performance, and some approaches also use the output from other scales to maximise captured semantic information. The architecture design used in T.-Y. Lin, Dollár et al. (n.d.) (Fast, Accurate) is shown in 2.23b, using lateral connections and a top-down path with skip connections to increase the semantic value of the layers used.

feature map. The authors noticed that feature maps at multiple scales were already being computed in the network backbone and sought to exploit multiple scales, predicting region proposals for each and combining at a later stage with NMS. This simplifies the network by removing the need for a dedicated RPN. The architecture of RetinaNet, shown in Figure 2.24, therefore no longer needed to have multiple scales of anchor boxes and instead combined the proposals from multiple scales with different anchor sizes per scale.

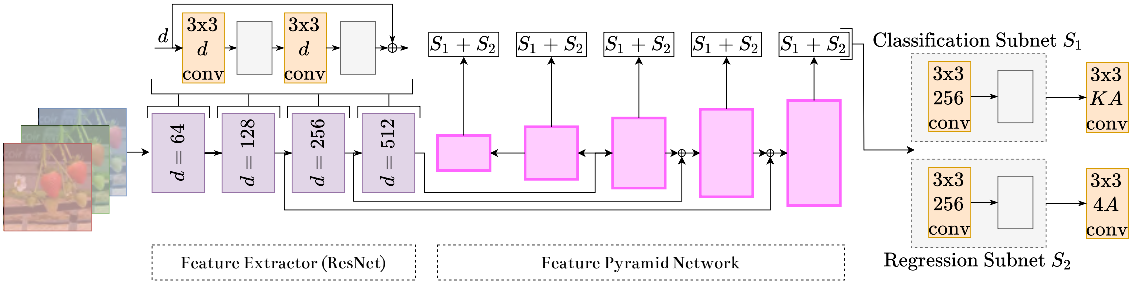


Figure 2.24: RetinaNet architecture with a ResNet 18 feature extractor backbone and FPN for multiscale anchor box regression and classification.

Two subnets are applied to the resultant feature map scales from the FPN, a classification subnet of shape $W \times H \times KA$ and regression subnet of shape $W \times H \times 4A$ where $W, H$ are the respective width and height of the feature map applied to, $K$ is the number of possible object classes and $A$ is the number of handcrafted anchors selected for each spatial position. The top $N$ scoring candidates from these two subnets are post-processed, typically with NMS to remove overlapping predictions for the same spatial locations, to output the final model predictions. Since thousands of anchor boxes are generated over each feature map and most candidate positions do not contain an object, there are substantially more background (negative) examples than that of foreground (positive) examples. With Cross Entropy loss, easily classified negative examples comprise the majority of the loss function and over influence gradient while training. The authors utilise Focal Loss to address this problem, shown in Equation 2.9, Focal Loss adds a factor $(1 - p_t)^\gamma$ to the standard cross-entropy criterion to reduce loss for well-classified examples for higher hyperparameter values of $\gamma$. This ensures that the loss of misclassified examples has a higher contribution to the overall loss of the network, resulting in better class balance and performance for positive objects.

In summary RetinaNet introduced semantically rich feature pyramid networks with skip connections, a class balanced loss function, and per-scale classification and regression of anchor boxes in a single network and at the time of publication it outperformed two-stage architectures such as Faster RCNN and single-stage architectures such as SSD (W. Liu et al., 2015) and YOLO (V2) (Redmon and Farhadi, 2016). While maintaining a simple architecture that was easily defined.

### 2.7.4 You only look at the coefficients (YOLACT)

Following the trend of adding instance segmentation capabilities to detection networks, like that of Mask R-CNN, following Faster R-CNN, one-stage architectures RetinaNet, SSD, and YOLO had instance segmentation capabilities added by RetinaMask (Fu, Shvets and Berg, 2019), Mask SSD (H. Zhang et al., 2020), and Insta-YOLO (Mohamed et al., 2021) respectively. Instance segmentation algorithms do not depend on bounding box architectures, but SOTA methods highly couple the two, as the joint training problem is effective in learning accurate object bounds and pixel relationships. The direction of research has shown that the features extracted within the network for bounding box prediction activate regions containing object pixels (positive), more so than background pixels (negative). At the time of publication Mask R-CNN had an inference speed of 7 frames per second on a TITAN X Nvidia GPU (high specification for the time). The addition of instance segmentation capabilities to the one-stage architecture was motivated by the search for more performant networks such as YOLO capable of running $\geq$ 30fps.

You only look at the coefficients (YOLACT), was introduced in 2019 (Bolya et al., 2019), and simplifies the object detection similarly to RetinaNet, removing the region proposal network and integrating an FPN for multiscale, resolution and semantic feature maps. The authors split instance segmentation into two distinct tasks, (1) creating a set of prototype masks and (2) predicting per-instance mask coefficients. This is shown in Figure 2.25 where $K$ refers to the number of classes, $P$ the number of mask coefficients, and $A$ the number of anchors.

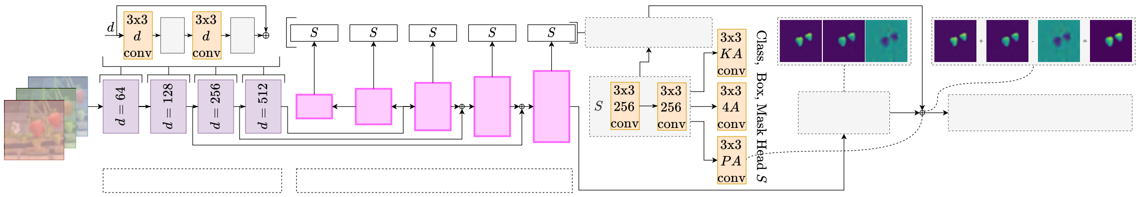


Figure 2.25: YOLACT architecture (RetinaNet based) combining a prototypic network head (Snell, Swersky and Zemel, 2017) with an extra mask coefficient head attached to the anchor box regression and classification heads for global mask extraction.

The prototype and mask coefficients are then combined linearly to create instance masks. The combination of both results in high-quality masks comparable to those of Mask R-CNN and exhibits temporal stability due to the lack of repooling. Each prototype mask is global and predicts a mask over the whole input image, and the network learns optimal combinations of the activations to produce a high-quality map based on the region-specific candidates detected from the FPN. Although YOLACT is less accurate overall for mean average precision compared to Mask R-CNN, it has a much lower inference speed. Improving upon Mask R-CNN at 7 frames per second to over 40 frames per second for the fastest variant. This is mostly due to the efficient use of feature maps and semantic information collated in the resultant top-down FPN feature maps. Similarly to RetinaNet, YOLACT presented a simple alternative to two-stage architectures for instance segmentation, removing the need for separate RPNs.

## 2.8 Object Tracking

Recent object tracking frameworks follow the detect-to-track paradigm (Wojke, Bewley and Paulus, 2017; Bewley et al., 2016; Bochinski, Eiselein and Sikora, 2017a; Ning et al., 2016) following the success of deep object detectors. R-CNNs were introduced in 2013 (Ross B. Girshick et al., 2013) which used convolutional networks to extract features from region proposals instead of relying on handcrafted low-level features such as edges, gradients and corners to detect objects. More recently, approaches have moved from two-stage architectures (CNN and RPN) to single-stage architectures (CNN) such as RetinaNet (T. Lin et al., 2017) and YOLO (v3) (Redmon and Farhadi, 2018) to improve the speed of the network while maintaining similar architectures. Due to the advent of these accurate and fast object detectors, tracking objects through detection has become a well established method for tracking and counting objects. Most approaches are generally formulated by combining object detectors, motion models, appearance models and data association algorithms.

Separation of target objects from distractors such as false positives is a challenging part of the association problem in the detect-to-track paradigm. At the time of publication, previous work focusing on simple tracking architectures have achieved SOTA results on the  dataset challenge.

### 2.8.1 Detect, Associate and Update

The detect-to-track (or tracking-by-detection) paradigm can usually be split into two categories, single- and multi-object tracking. Single-Object Tracking (SOT) aims at tracking an object of a single type rather than multiple objects. Some key papers in research refer to SOT as visual object tracking. A detected object can be either derived from an object detection NN or manually labelled in an image frame. Once labelled, the target bounding box is the initial state of a track. The purpose of SOT algorithms is to then detect the same object in subsequent frames or until the object disappears. This formulation of the problem usually does not require further updates of the current object detection, but instead relies on the algorithm to predict future states.

MOT on the other hand, tries to estimate the state of multiple objects, including objects with differing classes. For MOT based trackers, the goal is to track all of the detections in the first frame in multiple frames, regularly receiving updates on the current detection state associated with its predicted object states. MOT has the task of tracking any number of objects in an image frame until they are no longer present, even over periods of occlusion it should maintain the same identity for tracked bounding boxes.

Due to the nature of detecting, tracking, and measuring fruits in this thesis, the focus falls within the MOT paradigm. Tracking systems discussed in this section at the time of writing achieved class leading results on the MOT challenge, and are all that utilise a detect-to-track framework. Similarly, the principal technique of each algorithm is to detect objects of interest, associate with the estimated tracks, and update the internal tracking model such that of appearance or motion models, in each frame is represented. The motion model is responsible for predicting the future observations state and for appearance models to re-identify object instances over a long temporal range (Bochinski, Senst and Sikora, 2018) or for performing complex global optimisations to compute the tracks for each object. The commonality between SOTA detect-to-track methods used in the MOT based trackers is summarised below:

- **Detection Model** - Detect objects of interest in the current frame.
- **Motion Model** - Track objects through prediction of the estimated position, velocity, and shape (aspect) of observations.
- **Appearance Model** - Track objects through visual cues, such as generation of a unique identification feature vector that represents each observation of separate objects.
- **Association Model** - Build an association matrix between previous and

current observations, which can be based on the motion model, the appearance model or other combinations of detection and track association costs.

Most of the MOT SOTA trackers are build from a combination of these components and regularly perform with best results on the public benchmark. Three simple, SOTA methods are summarised below that utilise different combinations of the above techniques.

### 2.8.2 Tracking through Detection

Intersection over Union Tracker (Bochinski, Eiselein and Sikora, 2017b) uses simple IoU matching as the cost for matching objects across frames, however, it requires an extremely high frame rate detector. The authors note that with the advent of high frame rate object detectors, trackers such as a simple IoU association between frames is sufficient to reach SOTA results on the MOT dataset challenge. The approach makes two assumptions, (1) all frames containing an object contain a detection for that object with minimal gaps or occlusions and (2) that detections between frames are highly overlapping, as shown in 2.26.

When both assumptions hold, the authors show that the tracking formulation is trivial and only requires that tracks are maintained for a minimum number of frames and that at least one observation in each track is from a high scoring detection. Unlike SORT that uses the Hungarian algorithm for the association of detection and target observation cost, they note that in practice it is unlikely that detection and observation with a high IoU value will have many possible matches. Further, simplifying the tracking formulation by utilising the pre-computed IoU values from the detector. This simplicity makes the overall tracking framework almost free of overhead for extending the detector framework.

### 2.8.3 Detection and Appearance

Bochinski, Senst and Sikora (2018) extend the Intersection over Union Tracker (Bochinski, Eiselein and Sikora, 2017b) approach by noting that their previous assumptions do not hold and object detectors generally can miss detection between

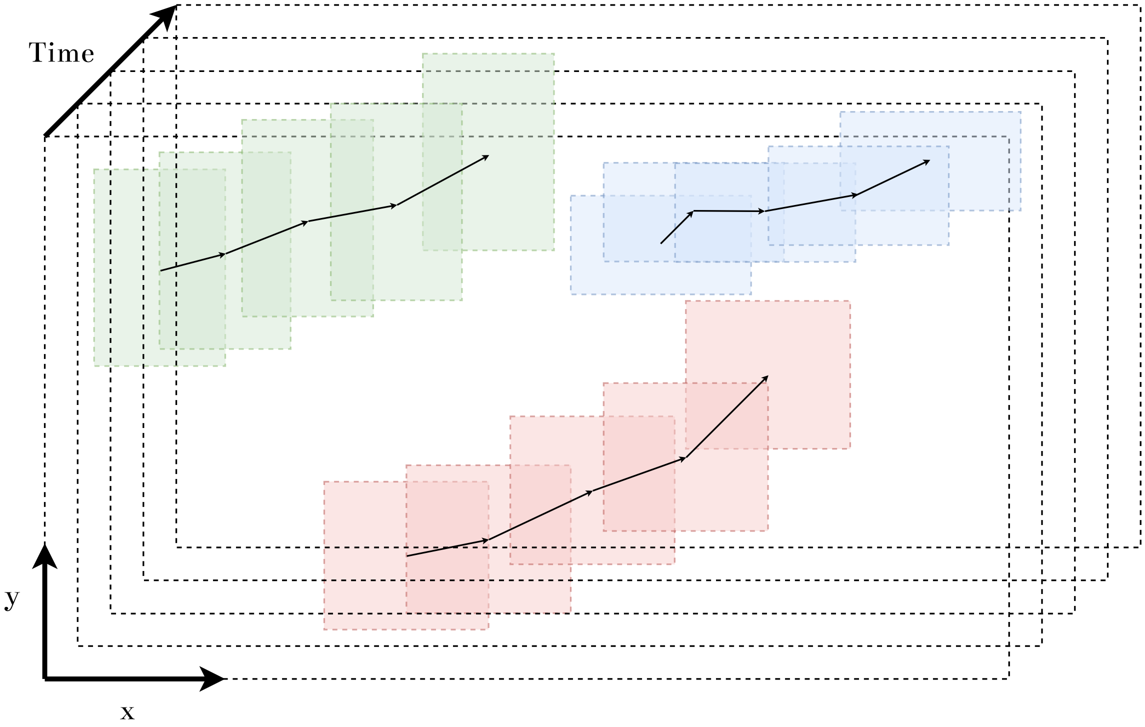


Figure 2.26: Detection based Intersection over Union Tracker for high frame rate image sequences with high IoU between objects

frames. Missed detection can arise for many reasons, such as occluding objects or poor detector accuracy or generalisation to different viewpoints. In this approach, the authors utilise visual information (i.e. an appearance model) to track objects when detections become unstable. By falling back to a visual model, the tracking formulation attempts to account for these cases. When a target detection does not have an associated detection in the current frame, the visual tracker is initialised with the last known location and is used to track an object for a set number of frames. If any new detection satisfies the IoU criteria on the visual tracker, then both tracks are joined and the visual tracker is terminated at the last known location. For any new tracks, a similar visual tracker is applied through the last number of set frames to check if it was actually connected to a lost track to ensure consistency in both directions.

### 2.8.4 Detection, Motion and Association

Simple Online Real-time Tracking (SORT) (Bewley et al., 2016) used the Hungarian data association algorithm with a Kalman filter cost to model the motion of detections and was at the time the best in class online tracking method, outperforming methods on the MOT dataset with much more complex architectures. Due to the simplicity, the method can operate at real-time speeds, which were over 20x faster than other approaches in the MOT leader-board at the time of publication. They estimate the motion of each object by approximating it with a linear constant velocity model, which is independent of other objects and camera motion. The target state in each frame is modelled with $[u, v, s, r, \dot{u}, \dot{v}, \dot{s}]$, where $u$ and $v$ are the target horizontal and vertical location, $s$ and $r$ are the scale and aspect ratio, and subsequent terms are components of velocity. This state space is the representation used in the motion model to propagate a target observation into subsequent frames. The Kalman filter framework is used to solve the velocity state components, where each detection is associated to a target observation. If no association is made the components are solved using the linear velocity model, the formulation in shown in 2.27.

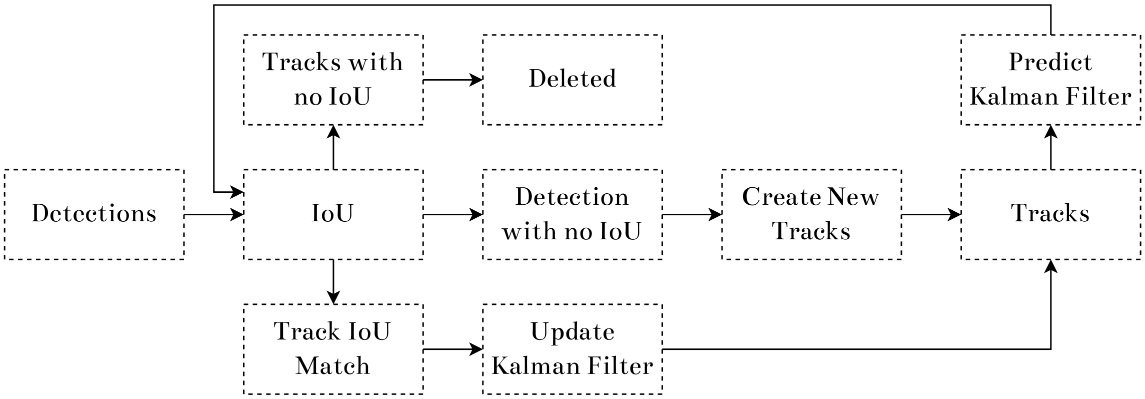


Figure 2.27: Detection, Motion and Association SORT Tracker.

The data association step takes a list of current targets (the predicted object states) and tries to associate them with the current observations (detections). In SORT, each of the target new states is estimated by predicting the new location in the current frame and the IoU cost is calculated for existing observations to associate detections and targets, where an association is made when the IoU cost is over a specified threshold. They solve the assignment optimally by the Hungarian algorithm for data association. Where the IoU threshold is not met, the object in question is created as a new identity with no-velocity and high covariance, denoting the large uncertainty. This created track has to meet a hard-coded number of associations before it will be fully instantiated as a new track. Track deletion criteria is the opposite. If no observations have been made for an identity for a specified number of time steps, the track is deleted, preventing performance degradation due to an unbounded number of untracked targets at any given time step.

Consequent to the track deletion step in the original papers, the SORT method suffers from objects that have missed-detections or are occluded for a short period of time, and they note this as outside the scope of the application. However, the harsh track deletion criteria aids in the overall efficiency of the model by considering a minimal number of targets at a given time step.

### 2.8.5 Detection, Motion, Appearance and Association

Similarly to the improvements made to the IoU based tracker by adding visual cues via Kernelized Correlation Filter (KCF) or Median-Flow, the authors in Wojke, Bewley and Paulus (2017) introduced Simple Online and Real-time Tracking With a Deep Association Metric which improved upon SORT by integrating an appearance model based on re-identification feature vectors. With the motivation to reduce the number of overall identify switches and lost tracks due to occlusion. SORT utilised a Kalman filter, IoU cost and the Hungarian algorithm in their object tracking formulation. DeepSort looks to extend this by adding an appearance model. Track handling and deletion criteria are mostly identical to the original formulation. With a state space of $[u, v, \lambda, h, \dot{x}, \dot{y}, \dot{\lambda}, \dot{h}]$ denoting the bounding box centre $u, v$, aspect ratio $\lambda$, height $h$ and respective velocities. For each track, a counter is instantiated to count the number of frames since a successful association. Tracks that exceed a predefined maximum age are deleted from the track set. New track hypotheses are initiated for each detection that cannot be associated to an existing track, and that pass a probation period of association for a minimum number of frames, tracks that do not associate within this period are deleted, as shown by the data flow in Figure 2.28.

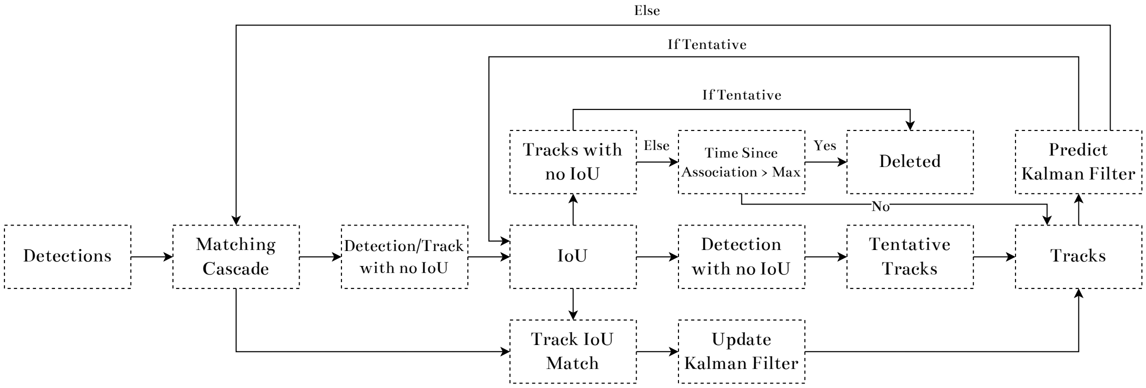


Figure 2.28: Detection, Motion, Appearance and Association DeepSORT Tracker.

The appearance model is created by training a CNN to classify examples into identities, a form of classification commonly referred to as re-identification. The re-identification network is trained with the last layer outputting a 128 length feature vector before the classification head. The objective of the network is to minimise the classification loss of each input identity. When trained, the feature vector output is used, with the motivation that the network has learned separable features for each differing identities and similar features for the same identities. For the Hungarian algorithm, the matrix is formulated of weighted costs of the predicted Kalman states and new observations and also the appearance model cost. Where the appearance model cost is a cosine similarity cost between appearance vectors from the trained re-identification CNN. The matching step occurs in the matching cascade that performs minimal cost matching for associations and attempts to use IoU to recover unmatched tracks and detections.

Utilisation of the re-identification feature embeddings ensures that DeepSORT includes appearance information and is able to track for longer periods of occlusion, making it a strong competitor to the SOTA algorithms at the time of publication. While similar to the single-stage architectures mentioned above, remaining simple to implement and running in real-time.

# Chapter 3

# Related Work

This chapter introduces the related work and background of research for the contributions noted in this thesis and its publications. It focuses on introducing the domain of machine and deep learning for computer vision within agriculture. Subsequently, introducing the application and success within current fruit detection and tracking research within horticulture. A system that can detect, track, and then further analyse fruit from images will benefit multiple current agricultural practices.

A further analysis is then given in Section 3.1 explaining the bridge gap between current research in deep learning based detectors and trackers and the specific application and modification of each to horticulture. Section 3.2 starts by then introducing both classical approaches (machine learning) and static image processing systems that have been developed for fruit detection, and ends with the summary of current deep learning based fruit detectors. Following is a summary of the current literature for fruit tracking in Section 3.3, an essential approach for systems that need to maintain detection identities over time, such that of counting systems or systems which need to perform visual inspection of a fruit.

The final section in Section 3.4 of the related work is fittingly related to fruit phenotyping, utilising multiple sensor types from CCD and CMOS cameras to hyperspectral imaging systems capable of multi-spectra image capture. The potential of non-destructive analysis of fruit is shown through a review of current work. The development of accurate detector and tracking systems in combination with non-destructive fruit phenotyping closes the loop (crop digitalisation from visual data) for the visual inspection of fruit and enables applications from destructive and labour-intensive harvesting to non-destructive phenotyping for estimating yield.

## 3.1 Automation in Horticulture

Many current horticultural practices such as harvest require monotonous, repetitive manoeuvres from a human picker, which are labour intensive for many fruit types. Labour is fast becoming an expensive commodity, with farmer population also falling, the end cost for the general population is rising (Si, G. Liu and J. Feng, 2015; Sa et al., 2016). By 2050 the United Nations Food and Agriculture Organisation (FAO) has stated the world needs to produce 70% more food to accommodate for the increase in population (Food and United Nations, 2009), with 80-90% of the crop production growth expected to come from higher yields and increased cropping intensity. In the UK specifically, labour shortages are more present in 2021 than previous years, due to numerous factors. Machine vision systems are being developed to help increase crop yield, harvest efficiency, effective cost while minimising resource waste (Teixidó et al., 2012). In, (S. Liu and Whitty, 2015) they estimate that precise yield estimation in grape vineyards will save one hundred million US dollars per year in the wine industry, further justifying the need and potential for computer vision systems in agri/horticulture. Capturing and analysing spatio-temporal attributes of crops for use in precise forecasting, automated disease detection, optimised agrochemical application and harvesting will become increasingly important in future agricultural strategies.

Moreover, autonomous systems could possibly reduce the impact/use of materials that currently pose a risk to the environment such as: fuel, water, herbicides, and pesticides amongst other materials (Hertwich, Voet and Tukker, 2010) and reduce worker fatigue caused by the increase of industry demand and uncomfortable harvesting conditions (Sa et al., 2016). Development of accurate fruit recognition systems is a crucial step towards increasing harvest efficiency (Kaczmarek, 2017); utilising the cheaper and larger endurance autonomous harvesting robots can provide. Strategies that support this advance have already been implemented in industry; for strawberries, tabletop culture greenhouses have been employed to reduce worker fatigue and support robotic harvesting by providing easy access to the fruits (Rajendra et al., 2009).

Some notable works have used SVM classifiers (S. Liu and Whitty, 2015), Bas-relief representations (Maldonado and Barbosa, 2016), and morphological processes analysing reflections caused by artificial lighting (Font, T. Pallejà et al., 2014) for yield estimation. SIFT descriptors and bag of word histogram super pixel classification (W S Qureshi et al., 2014) has been used for pineapple classification. Bag of words and statistical clustering has been used for pepper recognition and yield estimation (Song et al., 2014). K-Means clustering has been utilised for pomegranate detection and tracking (A. Roy et al., 2011). Texture and colour filtering with an edge fusion step has been used to detect count apples (Linker, Cohen and Naor, 2012) and utilising the OHTA colour space with principal axis of inertia to detect strawberries and their orientation (G. Feng, Qixin and Masateru, 2008). Similar work has also been attempted in 3D (L. Sun, Cai and Zhao, 2015; Chaivivatrakul et al., 2014; Kusumam et al., 2017; Rajendra et al., 2009; Hayashi et al., 2010; Scarfe, 2012; Font, Tomàs Pallejà et al., 2014; Kaczmarek, 2017), such as range and amplitude filtering of depth signals for citrus harvesting (L. Sun, Cai and Zhao, 2015), a combination of Viewpoint Feature Histograms, SVM classifiers and temporal filters for detecting broccoli heads (Kusumam et al., 2017) and simple colour and depth filtering in (Hayashi et al., 2010) for strawberry detection/localisation.

Novel digital technologies including vision systems, robotics and autonomous systems are seen as potential game changers for the horticulture sector. Visions systems can be used to assess and to sense the crop to enable better decision support; robotics and autonomous systems offer new means to drive productivity. These issues apply to all soft and top fruits, but also more widely across the whole fresh produce sector. However, all picking and vision systems are dependent on the development of complex algorithms developed to identify, measure and locate fruit in real time. The development of these systems is not trivial, especially in outdoor environments where the background light level and quality can change within an instant. This thesis aims to demonstrate real applications of soft fruit computer vision systems using off the shelf cameras, motivated by applications in yield estimation, forecasting and harvesting of soft fruit crops within the horticultural industry. In the following sections, the findings of current research and structure the impact of our contributions in the subsequent chapters is presented. A lot of current research looks at green vegetation classification, for example rice, pineapples, cotton, and apples. One reason for this is that they share a larger part of the market in most countries, especially due to issues such as weed control. However, the focus of this research is applied to soft-fruits, specifically Strawberries. Strawberries are interesting compared to some other fruits due to the change in appearance from white flowers to white then green unripe and finally red mature strawberries.

## 3.2 Fruit Detection

In the following section, a review of relevant and significant research is presented relating to detection of fruit within images. The literature review spans multiple sensor types, from colour and depth camera sensors to multi-spectra cameras, with a summary of progression shown in Figure 3.1. It also covers many image processing techniques, from support vector machines to deep learning. Classical approaches are presented first which is defined as a combination of image processing and machine learning and later deep learning with the advancement of the convolutional neural networks is introduced. Notably papers in this domain use conflicting definitions for some terms, when referring to fruit counting relating to detection it is the number of detections in an image, not the number across a sequence of images. A problem later solved by some detect to track based methods in Section 3.3.

### 3.2.1 Handcrafted Techniques

Approaches to the automation of harvesting, yield, maturity and quality estimation of fruit have been widely varied in the last 20 years (Nguyen et al., 2016). The appearance of fruits in natural conditions changes rapidly and is perceptually challenging to distinguish from different viewpoints due to reasons such as shadowing from surrounding objects. The shape, colour, texture, fruit attributes and environmental affordances all change unpredictably. The work presented in this thesis

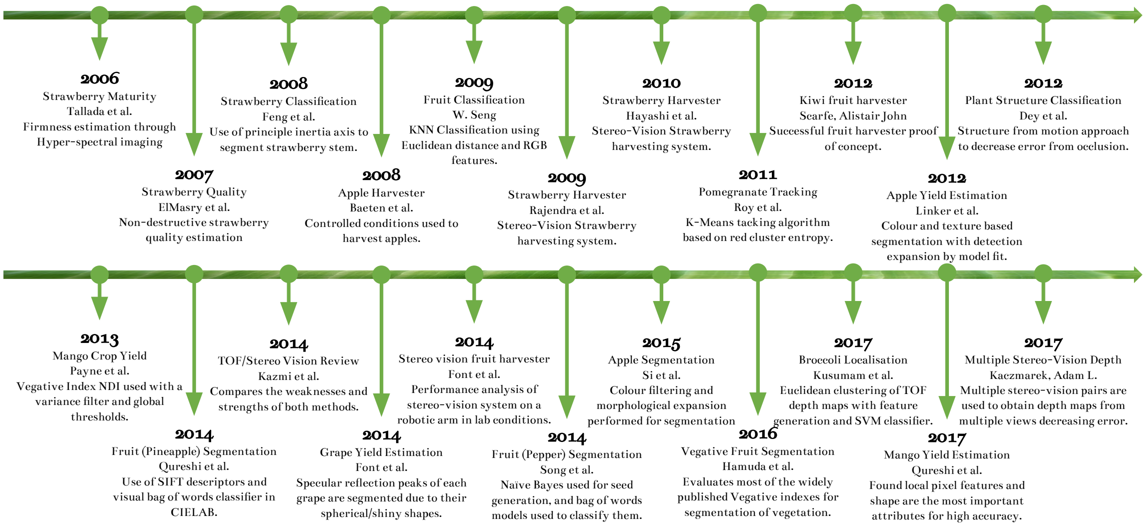

2006
Strawberry Maturity
Tallada et al.
Firmness estimation through
Hyper-spectral imaging
2007
Strawberry Quality
ElMasry et al.
Non-destructive strawberry
quality estimation
2008
Strawberry Classification
Feng et al.
Use of principle inertia axis to
segment strawberry stem.
2008
Apple Harvester
Baeten et al.
Controlled conditions used to
harvest apples.
2009
Fruit Classification
W. Seng
KNN Classification using
Euclidean distance and RGB
features.
2009
Strawberry Harvester
Rajendra et al.
Stereo-Vision Strawberry
harvesting system.
2010
Strawberry Harvester
Hayashi et al.
Stereo-Vision Strawberry
harvesting system.
2011
Pomegranate Tracking
Roy et al.
K-Means tacking algorithm
based on red cluster entropy.
2012
Kiwi fruit harvester
Scarfe, Alistair John
Successful fruit harvester proof
of concept.
2012
Apple Yield Estimation
Linker et al.
Colour and texture based
segmentation with detection
expansion by model fit.
2012
Plant Structure Classification
Dey et al.
Structure from motion approach
to decrease error from occlusion.
2013
Mango Crop Yield
Payne et al.
Vegative Index NDI used with a
variance filter and global
thresholds.
2014
Fruit (Pineapple) Segmentation
Qureshi et al.
Use of SIFT descriptors and
visual bag of words classifier in
CIELAB.
2014
TOF/Stereo Vision Review
Kazmi et al.
Compares the weaknesses and
strengths of both methods.
2014
Grape Yield Estimation
Font et al.
Specular reflection peaks of each
grape are segmented due to their
spherical/shiny shapes.
2014
Stereo vision fruit harvester
Font et al.
Performance analysis of
stereo-vision system on a
robotic arm in lab conditions.
2014
Fruit (Pepper) Segmentation
Song et al.
Naïve Bayes used for seed
generation, and bag of words
models used to classify them.
2015
Apple Segmentation
Si et al.
Colour filtering and
morphological expansion
performed for segmentation
2016
Vegative Fruit Segmentation
Hamuda et al.
Evaluates most of the widely
published Vegative indexes for
segmentation of vegetation.
2017
Broccoli Localisation
Kusumam et al.
Euclidean clustering of TOF
depth maps with feature
generation and SVM classifier.
2017
Mango Yield Estimation
Qureshi et al.
Found local pixel features and
shape are the most important
attributes for high accuracy.
2017
Multiple Stereo-Vision Depth
Kaczmarek, Adam L.
Multiple stereo-vision pairs are
used to obtain depth maps from
multiple views decreasing error.

Figure 3.1: Summary of key historical publications in computer vision for agriculture.

attempts to address these issues rather than focusing solely on detection of the fruit from a representative data set, which is what much other work does. Numerous problems exist in this research space and many approaches have shown promising results for classification, segmentation and localisation of crops (Kaczmarek, 2017; Maldonado and Barbosa, 2016; Dey, Mummert and Sukthankar, 2012; Song et al., 2014; W S Qureshi et al., 2014). However, as noted in, Sa et al. (2016) the problem of creating a fast and reliable fruit detection system still persists. Primarily due to many aforementioned reasons such as variable weather, crop illumination, seasonal condition, growth cycle, human induced scene changes, speed and multiple views.

In their proposal for tracking pomegranates over multiple frames in (A. Roy et al., 2011), they note two distinct approaches to automatic robot harvesting, spectral based and shape based. Stating that spectral based approaches are fast but weak to occlusions and inconsistent illumination, whereas shape based are computationally expensive but more robust to these limitations. They obtain 96.6% accuracy with a 25% and 11.3% false positive and false negative error rate respectively by using K-Means clustering and morphological operations. Concavity is used in (B. Zhang et al., 2015) to detect the stem and calyx of apples using 3D reconstruction techniques. A multi-spectral and a CCD camera are used for surface reconstruction of the apples, which are modelled as approximately spherical objects. The ratio between the reconstructed surfaces and that of a standard spherical object of the same size would have maximal intensity change, indicating a concave region that has a high probability of containing the stem. The stem is accurately detected in most of the trials using low cost cameras that do not require complex calibration, although a major limitation of this approach is data acquisition in non-laboratory conditions due to occlusion and visible defects. In (W S Qureshi et al., 2014) they utilise super-pixel over-segmentation, dense SIFT descriptors and a bag of words histogram to classify fruits in images; achieving an accuracy of 97.657% for pineapples. A bag of words models was also used in Song et al. (2014) to find peppers in images in their two-step automated fruit counting approach. Simple colour transformations and a naive Bayes classifier are used to detect initial regions of interest, which are then in-turn used to train the bag of words models, which uses texture and Maximally Stable Colour Region feature sets (Forssén, 2007). The estimates from multiple images are aggregated, limiting the impact of occluded fruits, to calculate the final fruit count. They note that a more comprehensive solution could have been achieved with 3D data.

Colour can be used for analysis as well as using texture and shape features, in G. Feng, Qixin and Masateru (2008) they use the HSI colour space to segment and calculate maturity of strawberries. Maturity is calculated based on the ratio of red pixels (ripe) to green in laboratory conditions. The principle axis of moment of inertia is then used for pose estimation and stem segmentation. Normalised green-red difference index (NGRDI) (Hunt et al., 2005) is used in An et al. (2016) to distinguish the background and vegetation. In, (Linker, Cohen and Naor, 2012) they integrate both texture and colour features to increase segmentation and classification accuracy. K-d-trees are used to increase the system performance, they organise the pixels through binary space partitioning, allowing for faster search of neighbouring points of interest. The partitioning subdivides the space into convex sets by hyperplanes so that smaller values will be on the left and greater values on the right. Smoothness and colour features are used to find fruit pixels, these sets of pixels are then expanded, using K-d-trees for efficiency, to any surrounding pixels sharing similar features. Contours are then segmented to form arcs; arcs in close proximity are joined. The joined arcs are compared against an ideal model of an apple to calculate the probability of it being a fruit. This was an effective way of segmenting the apples, an 85% and a 95% accuracy were obtained when capturing the images in natural lighting and underexposing the images respectively.

In, W. S. Qureshi et al. (2017) they propose a method for counting fruit on mango tree canopies using texture based dense segmentation and shape based. They use a high resolution CMOS camera and compare four different image analysis approaches for counting the mangos. They conclude by recommending two approaches, one an extension of work in Linker, Cohen and Naor (2012) and the other an extension of earlier work in W S Qureshi et al. (2014); Respectively they were, a K-nearest neighbour classification approach based on colour and smoothness with contour segmentation using elliptical shape models and a super-pixel over-segmentation approach with a bag of words models based on clustering dense SIFT features combined with colour filtering and morphological processes. Payne et al. (2013) also proposes a method to estimate mango crop yield. Normalised difference index (NDI) (Stajnko, Lakota and Hoevar, 2004) is used to reduce illumination variability and selecting regions where green is dominated by red colours. A variance filter then removes pixels in densely edged areas (background/plant structures). Finally a threshold is applied to the $Cr$ and $Cb$ channels of a $YCbCr$ version of the image and all of the previous steps are collated into a binary image. Occlusion was found to be most detrimental to the overall accuracy.

The approach in Diago et al. (2012) uses a supervised classification approach to classify plant structure of a grapevine, with users selecting representative pixel sets for objects in the data. For each class the Mahalanobis distance is calculated between the reference pixels and the search region, the minimal distance to a class decides the resultant classification. Similarly, in, Dey, Mummert and Sukthankar (2012) they use a supervised learning approach to classify plant structures of a grapevine. A sequence of images is used to generate a dense 3D point cloud through state-of-the-art structure from motion algorithms. This circumvents some common limitations such as heavy occlusion and variable illumination, it also provides data registration. The data was manually labelled into three semantic classes (foliage, branch, fruit) and used to train an SVM. This facilitated the classification of points in the input images. Saliency is calculated at three different spatial scales at each point, calculating principal components of the spatial distribution; categorising the shape type. This results in a 12D feature vector for each point in the 3D point cloud. KD-Tree is used for fast lookup of neighbouring points. Their pipeline produced accurate results, showing AUROC (Area Under the Receiver Operating Characteristic curve) values of 0.98 and 0.96 for green and purple grapes respectively, the purple grape accuracy was said to be lower due to increased capture distance.

Colour and depth information is used to segment and discriminate between plants and the background, a triangular meshing of segmented regions in the 3D point cloud is then performed to approximate the leaf surface. In (Maldonado and Barbosa, 2016) global colour thresholds, histogram equalisation, spatial filtering, log transformations and Gaussian blur are used to generate bas-relief representations of their data. Multiple Support Vector Machines (SVM) were trained on different canopy and fruit sizes to increase the detection rate of oranges. The age and variety of the plants was noted to be the largest influence on detection accuracy. Detection of specular spherical reflection peaks in RGB images are proposed as a method for counting red grapes in vineyards in (Font, T. Pallejà et al., 2014). A threshold value obtained using Otsu (Otsu, 1979) was applied to the hue layer, morphological filtering removed small noisy objects and the image was smoothed to reduce the number of peaks on each spherical grape. A radial morphological filter requiring the centre point of each region must have greater intensity than surrounding pixels was used to detect the specular reflection peaks and consequently to count the number of grapes. The average error of this method was approximately 14% which when accounting for the heavy occlusion is highly accurate.

Aggregation of depth information to overcome the many aforementioned limitations researchers have faced when only using colour data is used in Nguyen et al. (2016). All points within distance and a colour range are filtered and Euclidean clustering used to segment red point clouds into multiple clusters; combining clusters of single apples and splitting large clusters of red apples far apart. The Circular Hough Transform (CHT) is then used to count the detected regions. The reported accuracy was 100% for fully visible apples and 80% for partially occluded ones. In (Si, G. Liu and J. Feng,

2015) they report a segmentation accuracy of 96% in their proposal of an apple (Fuji) localisation pipeline, they utilise colour based filtering to segment the apples and then use RRM (Random Ring Method) over CHT to estimate occluded surfaces. They justify this choice in their proposal by stating CHT is inaccurate when objects are heavily occluded, and imply that RRM is faster.

In, Kusumam et al. (2017) they also utilise euclidean clustering in their work, where a combination of view-point feature histogram and an SVM classifier is used to segment broccoli heads in real outdoor harvesting conditions with a time of flight camera. A real time 95.2% accuracy with good generalisation at 84.5% accuracy was achieved. In Baeten et al. (2008) their vision system needs to take into account what apples should be harvested first, they implemented an AFPM (Autonomous Fruit Picking Machine) that could harvest apples in an average time of 9 seconds. Depth information could have been a crucial feature in deciding the order apples should be picked in and even the path of the approach to them. They use a time of flight camera in L. Sun, Cai and Zhao (2015) to localise citrus fruits in real-time, achieving a detection speed of 50ms per fruit on average and an accuracy of 81.8% through image analysis techniques focusing on the spherical characteristics of the fruits.

In, Chaivivatrakul et al. (2014) they employ time of flight cameras for phenotyping corn plants. Their method consists of five main procedures, filtering and merging 3D point cloud data, segmentation of both the leaves and stems, extracting data for phenotyping from the segmented regions and holographic visualisation. One procedure to be especially noted is the stem segmentation, in which they slice the point cloud orthogonally to treat the 3D data as multiple 2D slices, then each slice is morphologically closed, and the largest contours are extracted. The largest overlapping contours of the slices are linked into a set of ellipses, these are then used to estimate the stem centre line by finding best fitting ellipses between them through least squares ellipse fitting. This was then used to model the stem in 3D space for phenotyping. They compare time of flight and stereo-vision systems for depth imaging of leaves in Kazmi et al. (2014), and conclude that time of flight cameras struggle from their low resolution and lack of robustness to lighting variations but have a high frame rate whereas stereo-vision cameras have a high resolution but struggle from a low frame rate and correspondence problems between points in both images.

Both Gongal et al. (2015) and Rajendra et al. (2009) mention that limited visibility, occluded fruits, obstructions and irregular shape features are just a few of the complex challenges that have to be overcome for accurate segmentation and localisation of fruit. This was observed in their experiments where they attempted to harvest strawberries in tabletop culture greenhouses. In this case, they used a three camera stereo-vision set-up, utilising the left and right cameras for depth calculation and the centre camera for position correction and detecting regions of interest through HSI colour filtering. In (Scarfe, 2012) an example of an autonomous robot for picking kiwifruit using stereo vision systems is provided. The robotic arm bends and pulls the peduncle of the fruit to minimise damage, an issue that (Baeten et al., 2008) also recognised in their results. In, (Scarfe, 2012) they concluded by validating the demand and plausibility of using artificial vision for autonomously picking fruit. Strawberries are soft fruits which provides challenges for rapid harvesting, (Hayashi et al., 2010) proposed automated harvesting of them at night where the temperature drop results in a harder pericarp. In Hayashi et al. (2010), the design of the mechanical arm mitigates the error of fruit localisation, by using a suction cup to allow for small errors. Chromaticity of the red RGB channel is calculated, and a global threshold is applied, then the colour space is converted to HSI and globally filtered in two stages to detect ripe and unripe sections of the regions of interest. Maturity is obtained for each region of interest using the proportion of ripe to unripe pixels. Regions containing at least 80% ripe pixels are labelled as target fruit for picking, sorted by distance. The location of the peduncle was assumed to be around 20 pixels above the calyx on each strawberry, so this was set as the search region for calculating where to cut, the closest connected region in this search area is labelled as the peduncle.

Limitations of this approach were complicated peduncle conditions such as discolouring and irregular shape. The accuracy of the system in (Hayashi et al., 2010) was low due to detection failure from matching the left and right images from the stereo-vision system. They note that an additional camera should mitigate this issue. In (Font, Tomàs Pallejà et al., 2014) they also discuss the difficulty of correlating information between sensors in stereo-vision systems and suggest a calibration procedure to minimise error. In this proposal, they use a single robotic arm and a low-cost stereo vision system to segment and harvest fruit. Their system comprises four steps: initial fruit detection, rough fruit approach, fine fruit approach and fruit pick-up. Equal Baseline Multiple Camera Set (EBMCS) is introduced in (Kaczmarek, 2017) for obtaining an accurate depth map of plants. This proposal compares the quality of the disparity maps from popular stereo matching algorithms using their five camera vision system. They note the error rate of distance estimation is reduced by 26.55% using five cameras and integrating multiple disparity maps. Calibrating their multi-camera system utilises the same algorithms standard stereo-visions use, since the cameras are treated independently of one another.

### 3.2.2 Convolutional Neural Network Based Detectors

The fruit detection research listed above from segmentation through to classification and yield estimation have typically used handcrafted detectors, carefully selected algorithms and techniques, that when combined can transform image data into fruit detections. With the advent of high-accuracy CNN based detectors, the trend of research has shifted to utilisation of learned solutions for the initial detection stages in many approaches. More recently, deep learning based methods have been applied with excellent results (Sa et al., 2016; Y. Chen, Won Suk Lee et al., 2019; Kirk, Cielniak and Mangan, 2020b; Zhou et al., 2021). This section explores a few of the applications of CNNs for soft-fruit detection.

Indeed, the recent DeepFruit (Sa et al., 2016) paper reports an F1 score of 0.838 for sweet peppers. More interestingly, the deep network can easily and relatively quickly, in four hours, be retrained for new fruits, achieving accurate fruit detection through transfer learning for Rock melon, Apple, Avocado, Mango, Strawberry and Orange. However, while most of the aforementioned approaches have been shown to be accurate in laboratory settings or on general image data-sets, their applicability to the challenges that a real world robot harvester will face, such as rapid and robust segmentation across changing weather, lighting, and seasonal conditions, remains largely unknown.

In Y. Chen, Won Suk Lee et al. (2019) they use CNNs to count the number of ground grown strawberry flowers and fruit in a field from drone imagery. To develop this system, the researchers used small UAVs to take near-ground RGB images for building orthoimages at 2 m and 3 m heights. After their generation, they split the original orthoimages into non overlapping frames for Faster R-CNN based detection, which was based on the ResNet-50 architecture and transfer learning from ImageNet. The best detection performance was for ripe fruit (most distinctive class), with an AP of 0.91. Immature fruit at a distance of 3m did not perform as well, which the authors attributed to the model confusing them with green leaves, having the worst AP of 0.61. The results showed that a CNN could effectively count flowers, with a detection accuracy of 84.1%, with an average occlusion of 13.5%.

Similarly, the authors in Zhou et al. (2021), also explored the area of using simple CNN based detectors with aerial imagery to classify strawberry maturity near-ground images. The YOLO (v3) model was applied to detect and classify three (in UAV images) and seven (in near-ground images) classes of strawberry maturity stages. For UAV image analysis, the highest AP was again for ripe fruit (visually distinct to other classes and background) to be 0.93 and the mAP was 0.88 in the test data set at a height of 2 m. The near-ground imaging method provided more detailed information about strawberry maturity stages, from flowers to ripe fruit. Again the ripe fruit class outperformed all others in terms of AP with 0.94, 0.06 higher than the mAP for all classes which was 0.89. The authors demonstrated a method for large scale data collection in strawberry fields.

Outside of yield estimation, detection systems provide a basis for detecting diseases that are observable in imagery. In, Kim et al. (2021) they look to detect a number of strawberry diseases from images by utilising a two-stage cascade disease CNN. It is common for detection networks to be pretrained on the ImageNet dataset due to its generality, however the authors explore the effect of training on a more representative dataset first. For transfer learning, this can substantially reduce the time it takes for weights to be learned in CNN architectures that relate to the horticultural domain due to more representative features already existing in the network feature extractor. The authors note the new transfer learning approach, based on the PlantCLEF plant dataset from the LifeCLEF 2017 challenge (Joly et al., 2017), results in a 3.2% mAP increase.

In, Nikolas Lamb and Mooi Choo Chuah (2018) they implement an SSD neural network for strawberry object detection. This work is the application of optimisation approaches that can be taken to increase the performance of CNNs without sacrificing accuracy. They achieve speed-up compared to the baseline system by compressing the input of the network to 360x640px and by the application of a colour mask to isolate regions of interest before inputting tiled images to the network. As well as fine-tuning the model, removing low performing filters and retraining to optimise the number of parameters in the model. This shows that systems can be optimised for use in real conditions where performance on energy restricted hardware.

The authors in Pérez-Borrero et al. (2020) present a method that is also motivated by the real-time requirements of this technology applied to strawberries. The authors modify the Mask R-CNN two-stage architecture, by reducing the number of layers in an effort to optimise the architecture originally developed and benchmarked on more complex datasets, containing a wider variety of objects and classes, to only detecting fruits (ripe and unripe). After comparing the proposed methodology with the Mask R-CNN network used in (Yu et al., 2019), under the same conditions, their proposal achieves an mAP of 43.85 compared to a baseline mAP of 45.36. However, the optimised network runs at 10fps instead of 2.5fps on the original high-resolution images in the data set, demonstrating the applicability of these methods on with limited hardware capability.

## 3.3 Fruit Tracking and Counting

Fruit detection enables numerous applications, as shown in Figure 1.1, however due to the systems being trained and tested to detect objects in disjointed frames, application of the systems to image sequences usually leads to unstable results for the same object instances. This section introduces approaches for tracking detections across frames, usually motivated to count or track instances for yield estimation or harvesting. Note, counting in this section refers to counting the number of objects across frames, not in single images as presenting in research in section 3.2.

Detect-to-track is applied to Mangos in X. Liu et al. (2019) where a Faster R-CNN model detects regions of interest and a combination of Kanade-Lucas-Tomasi (KLT) optical flow estimator, Kalman Filters, and the Hungarian Assignment algorithm are used to associate detections to tracks. Finally, an Structure From Motion (SfM) algorithm is applied to reduce the effect of double-counting, achieving an overall mean error of 27.8. Double counting occurs when a single identity contributes more than one count to the total, for example, applying only a detection model $N$ times to the same frame would result in increasing the count $C$ to a count of $NC$ without any additional steps.

Oranges and apples are counted per frame in S. W. Chen et al. (2017) utilising a neural network to produce detections and to regress the final image count, resulting in an overall $L2$ error (least square error) of 13.7 and 10.5 for oranges and apples. The authors in Xu Liu et al. (2018) extend this work from counting fruit in single images to sequences of images. The proposed pipeline integrates deep neural network detections with SfM algorithms to count fruit from a single camera, achieving an $L1$ error (least absolute deviations) of 203 for oranges and 322 for apples. The SfM reconstruction correction step improves count accuracy for oranges more significantly than that of apples, which is noted due to the orange data collection being performed under natural illumination featuring high occlusion and depth variation. Most orchards and farms feature similar uncontrolled conditions, suggesting the benefit of their algorithm in practical use cases.

Fruit counting from a mini-tractor of a Kiwi crop has shown good results, deployed commercially for over two years (Mekhalfi et al., 2020). The system consists of a vision sensor attached to a patrol vehicle to capture images of the environment. Offline, the images are then stitched via Speeded-Up Robust features (SURF) and counts of detected fruits are generated. Their proposed system counts Kiwis within average percentage errors of 5% to 15%. They validate the use of automated counting with commercial experience of reduced labour load and the development of crucial tools to infer yield estimations.

Gaussian Mixture Modelss (GMMs) are combined with CNNs in (Häni, P. Roy and Isler, 2020) to detect and count fruit. In this paper, they explore new fruit detection and counting methods and compare them for the task of yield estimation in apple orchards. They show different methods have different pitfalls, and compare each on multiple datasets. For fruit detection, their semi-supervised clustering technique, based on GMM, achieved the highest $F_1$. For fruit counting, the CNN approach on single image data sets performed better, but the R-CNN suffered from poor precision. The classical segmentation method combined with the CNN based counting approach achieved yield accuracies ranging from 95.56% to 97.83% compared to the GT. Utilisation of SfM and the semantic segmentation CNN, allows them to align the point-clouds to the pixels in the input image, creating a one to one correspondence. To prevent double counting of fruits on trees, they use the algorithm described in Dong, P. Roy and Isler (2018) to compute the intersection of fruit counts from both sides of the tree row. They then use the intersection area among connected clusters to compute the total fruit counts, taking into account the weighted parts.

In, W. Zhang et al. (2022) they utilise the YOLO detection CNN and a modified SORT tracking formulation to detect and track oranges in orchards. This work improves the accuracy of fruit detection and fruit tracking by taking into account small-scale characteristics of field orange data and the different occlusion statuses in the video sequence. It uses two sub-algorithms, OrangeYolo for fruit detection and OrangeSort for fruit tracking. Their proposed methodology achieves an AP value of 0.938 for detection of oranges within their field dataset. They extend SORT by estimating the tracking region (motion displacement estimation) for oranges with the assumption of camera motion and static objects. They use six video sequences from two fields containing 22 trees as a validation dataset and show the best performance is an Mean Absolute Error (MAE) of 0.081 and standard deviation of 0.08.

## 3.4 Fruit Phenotyping

Phenotyping is the detection of composite characteristics and traits of organisms. Estimation of fruit quality information and phenotypic traits is a crucial component in translating genomic knowledge into useful information for an efficient strawberry breeding programme (Mathey et al., 2013) and moreover successful robotic fruit harvesters (Xiong et al., 2019) and yield estimators. Extracting crops from images and providing further analysis enables many mandraulic processes (counting, harvesting etc.) to be automated. The estimated traits are grouped by region in terms of suitability for the respective industry and are currently reliant on the human eye to make assessments (Mathey et al., 2013). Recent work has expressed the importance of automating these processes for enhanced breeding efficiency using information only computer vision sensors can provide (Vázquez-Arellano et al., 2016). As stated in, (Goddard and Hayes, 2007) manual phenotypic trait estimation is more likely influenced by human bias and is not suitable for generation of commercial scale quantitative prediction models.

Vision systems aim to segment, classify and localise fruit instances in the environment and provide meaningful semantic information such as area, position, size and maturity (Kirk, Cielniak and Mangan, 2020c). The proposed methods depend on detected strawberry regions, and previous work indicates good detection accuracy using a variety of techniques. The authors in (G. Feng, Qixin and Masateru, 2008) use the HSI colour space to segment and calculate maturity of strawberries. Maturity is calculated based on the ratio of red pixels (ripe) to green in laboratory conditions. The principle axis of moment of inertia is then used for pose estimation and stem segmentation. In (An et al., 2016) the normalised green-red difference (Hunt et al., 2005), is used to distinguish the background and vegetation. Both texture and colour features can be integrated to increase segmentation and classification accuracy (Linker, Cohen and Naor, 2012). Smoothness and colour features are used to find fruit pixels. These sets of pixels are then expanded using k-d-trees for efficiency and contour arcs in close proximity are merged and compared against an ideal apple contour model. This was an effective way of segmenting the apples, an 85% and a

95% accuracy were obtained when capturing the images in natural lighting with some pre-processing to underexpose the images. More recently, in our previous work, deep networks have also been used as a method to detect the initial strawberry regions (Kirk, Cielniak and Mangan, 2020c) with good accuracy.

Approaches to compute industrial quality metrics describing strawberries have also been previously proposed. One such approach (Ishikawa et al., 2018) classifies the shape of strawberries into nine different groups using machine learning, and the results show that eight of the shape classifications can successfully be determined from image features. Strawberry orientation, major axis length and minor axis length computation methods based on image level features for stem picking point detection have also been proposed (Huang, Wane and Parsons, 2017). The best performing method in their approach simply intersects the lowest detected point with the centroid of the berry to determine the search region for the picking point detection method.

Previous work has used 3D information to estimate phenotypic traits of strawberries in laboratory conditions using point cloud meshes (J. He, Harrison and Li, 2017). The strawberry point clouds are constructed from a stereo imaging platform. The platform consists of a 360 deg revolving object and a high resolution RGB camera to match features between many RGB camera frames and calculate the 3D point information. Using this point information they calculate a mesh of the strawberry calyx (leaf) and exocarp (skin including achene regions) through Poisson Surface Reconstruction (Kazhdan, Bolitho and Hoppe, 2006) to estimate berry height, width, length, volume, calyx size and achene number. They show good agreement between GT data and predicted values. It is noted a further feasibility study is required to optimise this approach for application in current strawberry breeding programmes. They state an average processing time at least ten seconds per berry, as a rotating platform with manual strawberry placement would require even further time, however taking only compute time their method outperforms the same manual assessment by three times.

In, Jay et al. (2015) they use structure from motion for in-field phenotyping. They mention that stereo-vision systems do not always fulfil phenotyping-related constraints and that structure from motion cameras are simpler since they only require one camera and no prior calibration. Hyperspectral imaging is a non-destructive, chemical free method of capturing abundant spectral information of objects in the environment. The output of a hyperspectral system is a one dimensional spectrum containing physical and chemical information of each pixel in an image. This information can be captured in multiple different ways: point, line or area scan. Point scanning requires the longest amount of time, area scanning is unable to detect moving objects, and line scanning is the most commonly used data capture technique (ElMasry and D.-W. Sun, 2010). In, (Yang, Won Suk Lee and Gader, 2014) they note the successful application of this technique in agricultural research.

In, (ElMasry, Wang et al., 2007) they stress the importance of internal quality evaluation as well as external evaluation for strawberries; using non-destructive techniques to obtain this information reduces the level of manual labour required. In this proposal, they determine quality attributes of strawberries through hyperspectral imaging, taking advantage of chemical and physical variation. The spectral data is analysed using partial least square models for moisture, soluble solids and acidity prediction and texture analysis for maturity estimation is estimated based on a gray level co-occurrence matrix. The accuracy and performance shown in their results were high, at 89.61% maturity performance and correlation coefficients of 0.9, 0.8 and 0.87 for moisture content, total soluble solids and acidity levels respectively. The results obtained in this research showed that the reflectance curves of the strawberries were smooth over the spectra wavelengths, the chlorophyll absorption band at 680 nm was absent in ripe strawberries and the anthocyanin and sugar absorption bands were much higher than unripe strawberries. This consequently allows unripe strawberries to be easily segmented. In Tallada, Nagata and Kobayashi (2006) they also recognise the relationship between chlorophyll levels and maturity in their research, where they attempt to estimate firmness, which is an indicator of maturity of strawberries using NIR (Near Infra-red). The results they obtained through using stepwise multiple linear regression models was a correlation coefficient for prediction of 0.78. They mention high variability of firmness in unripe strawberries and a relatively stable uniform value for ripe strawberries showing reliable spectral difference between the crop states.

Non-destructively diagnosing the infection stages of anthracnose in strawberries using hyperspectral imaging was proposed in (Yeh et al., 2016) in which they achieve a classification accuracy of 82%, reinforcing hyperspectral system feasibility in the diagnosis of infections. They note this point especially because of the potential revenue stream gain due to the fact the early forms of this infection are invisible to the human eye. The hyperspectral camera range was 400-1000 nm at 4.6 nm spatial resolution. As with most hyperspectral camera systems, a reference and a dark image was used to calibrate the camera with regard to environmental and equipment interference. Classification results of fruit attributes similar to the ones mentioned in this section can be seen in a vast proportion of research, another example is available in (Rajkumar et al., 2012) where the use of visible and NIR spectra from a hyperspectral system are used to determine banana maturity and quality. In this paper, partial least squares analysis is used similarly to (ElMasry, Wang et al., 2007) for optimal spectra selection for independent variables; moisture content, firmness and total soluble solids. They note that a drawback of the capture device is the low capture speed, stating it is one of the reasons it has not fully been applied to on-line systems. GT data were obtained after image acquisition with respective industry equipment for each independent variable they measured, and they found that moisture content had a linear relationship with different maturity stages.

In, Piazzolla, Amodio and Colelli (2017) they use an NIR hyperspectral device for grape harvest time estimation using maturity heuristics. As with aforementioned research, estimation of: total soluble solids, acidity, phenols, and antioxidant activity were effective in relating intrinsic grape characteristics to maturity or ripeness with only 14 wavelengths. In, Yang, Won Suk Lee and Gader (2014) they explore the feasibility of hyperspectral systems for the classification of plant growth stages and the background. To address the vast quantity of information hyperspectral imaging provides they applied a binning technique, effectively reducing the spectral resolution and consequently the number of spectral bands to be analysed, increasing the analytical performance. Dimensionality reduction algorithms were used to select spectral bands to be analysed, once selected they are assessed through the use of supervised classifiers. The fluctuation of environmental properties such as reflectance are a limitation of this approach, reducing the quality of captured information. The combination of non-Gaussianity measures with K-Nearest neighbour classification achieved an overall accuracy of 98.7% for the four classes: mature fruit, intermediate fruit, young fruit and background which had accuracies of 97.5%, 100.0%, 98.9%, 98.7% respectively. This is another good example of the effectiveness of hyper-spectral imaging and attribute evaluation and plant evaluation. The research discussed in this section highlights some major advantages multi-spectral systems have over other techniques, they capture abundant information that can be used to derive object attributes in a chemical free, non-destructive way. However, they are slow and are expensive, so real-time use is not currently feasible in applications where being real time is critical.

# Chapter 4

# Soft Fruit Object Detection

In this chapter, coercive and free learning policies are introduced to shortcut learning more representative features. The motivation for this is to enable transfer of lab based object detection models on curated datasets to unseen outdoor data from multiple view points. Parts of this were published in *L*a*b*Fruits: A Rapid and Robust Outdoor Fruit Detection System Combining Bio-Inspired Features with One-Stage Deep Learning Networks* Kirk, Cielniak and Mangan (2020b). A neural architecture based on a single-stage detector, RetinaNet T. Lin et al., 2017 is introduced to detect soft-fruit. The training is formulated with early fusion of a more representative colour space (L*a*b*) for fruits to train faster and increase accuracy on unseen viewpoints. Training with these policies is shown to lead to minimal accuracy increase over regular colour spaces when applied to the representative training dataset, but when applied to unseen examples from multiple views that contain dramatic appearance changes (such as illumination and colour) that it leads to a much greater accuracy increase, further, generalising object detection networks for use within the horticultural industry.

The challenges and motivations for fruit detection systems were presented in Section 3.1. Automation of agricultural processes requires systems that can accurately detect and classify produce in real industrial environments that include variation in fruit appearance due to illumination, occlusion, seasons, weather conditions, etc. This chapter introduces our contributions and work towards the development of a robust detector for strawberries that can alleviate some of the issues, specifically the variation in appearance of crops from multiple-views and illumination constraints, and simplify the training process. This chapter addresses some of the main issues with object detectors in horticulture. Detection of fruit in outdoor conditions is difficult due to the variable illumination, which heavily impacts detector performance where representative datasets are not available. Data collection and annotation is extremely costly, methods are explored to improve the accuracy of detectors on datasets with illumination conditions not present in the training sets for CNNs. By utilising only the original datasets, the transformations of the input are shown to bolster the test-time performance of CNNs for object detection.

In this chapter, a visual processing approach inspired by colour-opponent theory in humans is combined with recent advancements in one-stage deep learning networks to accurately, rapidly and robustly detect ripe soft fruits (strawberries) in real industrial settings and using standard (*RGB*) camera input. The $F_1$ score is utilised, the harmonic mean of precision and recall, to show our system matches the state-of-the-art detection accuracy ($F_1$: 0.793 vs. 0.799) in controlled conditions; has greater generalisation and robustness to variation of spatial parameters (camera viewpoint) in the real-world data-set ($F_1$: 0.744); and at a fraction of the computational cost allowing classification at almost 30fps. It is proposed that the L*a*b*Fruits system addresses some of the most pressing limitations of current fruit detection systems and is well-suited to application in areas such as yield forecasting and harvesting. The resultant system was tested on an existent data-set captured in controlled conditions as well as our new real-world data-set captured on a real strawberry farm over two months. Beyond the target application in agriculture, this work also provides a proof-of-principle whereby increased performance is achieved through analysis of the domain data, capturing features at the input level rather than simply increasing model complexity.

A novel solution is presented that moves beyond the SOTA by displaying greater invariance to environmental changes in the agricultural domain. Our high throughput fruit detection system utilises a combination of recent advancements in deep learning that have been shown to remove variance within data sets (T.-Y. Lin, Dollár et al., n.d.; K. He, X. Zhang et al., 2016; T.-Y. Lin, Goyal et al., 2017). This approach incorporates an efficient feature extractor ResNet which fuses *RGB* and colour opponent data combined with a multiscale feature pyramid network to deal with scale invariance and RetinaNet for classification with the modified focal loss function reducing class imbalance. An evaluation of this system is presented on data sets collected from a real strawberry farm in natural conditions which compares its performance to the state-of-the-art network in Sa et al. (Sa et al., 2016). The specific contributions of this chapter are as follows:

1. Combining colour opponent features represented in *CIELab* space and *RGB* to provide greater multiple viewpoint invariance on networks trained on a singular view-point. This approach, referred to as early fusion, is then validated on viewpoints not present in the training data that show great variation in both spatial properties such as shape and illumination changes affecting colour.

2. Development of an accurate, high resolution and high throughput fruit detection system based on efficient network topology that can be trained on a low number of images in only one hour using state-of-the-art approaches such as Feature Pyramid Networks (T.-Y. Lin, Dollár et al., n.d.), Residual Neural Networks (K. He, X. Zhang et al., 2016) and RetinaNet (T.-Y. Lin, Goyal et al., 2017).

3. Study of the proposed system in Section 4.6. Showing the effect individual components of the system have on overall accuracy, such as reduction of data set size and different permutations of model input.

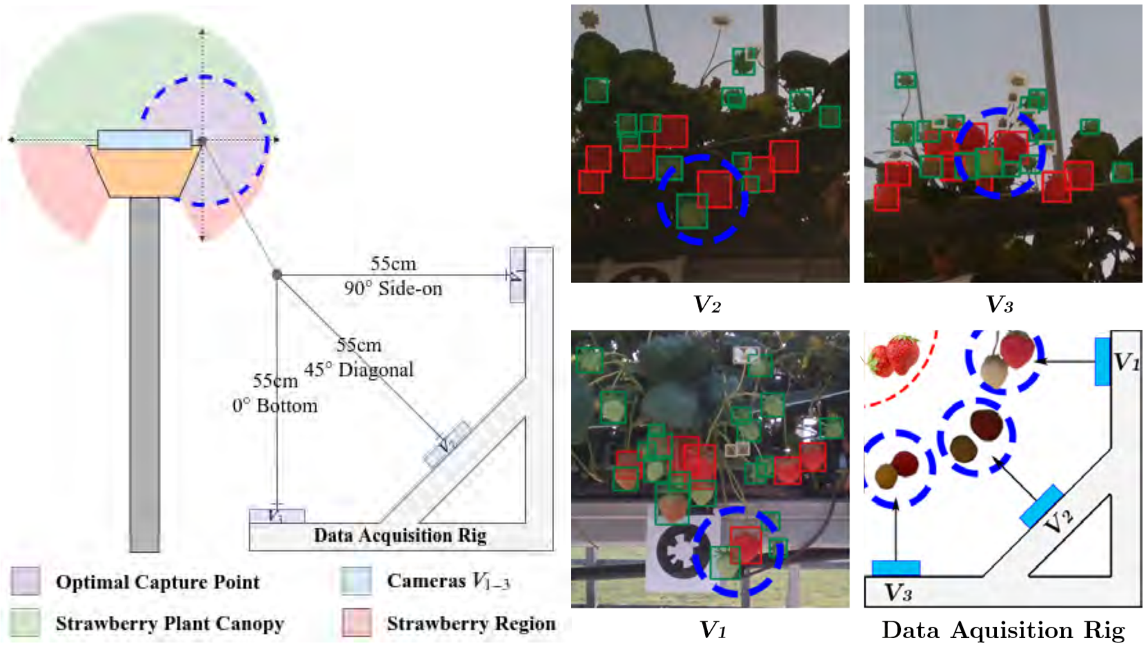
4. Publication of an open access longitudinal strawberry data set captured in real agricultural environments from multiple views over a period of two months, each providing weather data, camera parameters, RGB, stereo infrared images and registered point clouds (available here).

This chapter is organised as follows: an introduction to the data collection for benchmarking our methods in Section 4.1, multiple-viewpoint data acquisition in Section 4.2, an overview of vegetation indexes in Section 4.3, an introduction into the colour opponent process in Section 4.4, and a description of the proposed methodology in Section 4.5. Section 4.6 presents the experiments used to validate our hypothesis of removing the effect luminance has on object detection through approximated human vision mechanisms, and finally this chapter concludes with a short summary and discussion of future work in Section 4.7.

## 4.1 Data Requirements in Horticulture

For any deep learning architecture one of the most important factors is the quality and quantity of annotations in the data set, a deep network cannot learn features not present in the GT labels. A large data set was captured in real agricultural conditions containing all of the constraints discussed above (Variable weather, illumination etc.) and a low number of annotated images for use in this project. However, capturing all of the temporal differences and all of the attributes that influence the constraints mentioned above in an agricultural and other non-trivial environments in a single data set would be extremely difficult. It is unrealistic for current SOTA systems to learn the relationship of the extrinsic environment parameters (illumination, orientation of acquisition platform, weather etc.) through including a much greater number of examples in the training data. This method scales training time and data collection time linearly with the number of variable parameters and decreases the efficiency of applying the system on other object detection problems. Scale and orientation of objects and the perceptual differences over time are issues the current systems cannot fully accommodate, shown later in Table 4.3. To truly challenge these constraints, alternate methods to boost performance and generalisation must be sought, even when using small amounts of data. Figure 4.1b shows three images captured at the same location from different viewpoints, demonstrating some of the challenges that an automated perception system will face, such as the occlusion introduced, changed shape, perceived colour and texture. A fusion of $RGB$ features and colour opponent features is proposed to mitigate part of the illumination constraint and consequently increase the generalisation to viewpoints not previously observed.

Colour is one of the most relevant cues in detecting ripe soft fruit such as strawberries and shown to be directly related to their intrinsic attributes such as sugar level (Meulebroeck, Thienpont and Ottevaere, 2016). Yet, the visual appearance of fruit changes due to (a) different shape and texture between levels of maturity (b) variation of natural conditions such as weather, illumination, seasonal condition and growing cycles or (c) changes of camera viewpoint, shown in Figure 4.1a and Figure 4.1b. Many approaches have shown promising results for classification, segmentation, and



(**a**) Camera Configuration

(**b**) Camera Output and Variation

Figure 4.1: Camera configuration for viewpoints $V_1$, $V_2$ and $V_3$, where, $V_{1-3}$ are camera identifiers in (a). View point introduced illumination variance for $V_1$, $V_2$ and $V_3$. Blue circles show the effect viewpoint has on class appearance in (b).

localisation of crops (Kaczmarek, 2017; Maldonado and Barbosa, 2016; Dey, Mummert and Sukthankar, 2012; Song et al., 2014; W S Qureshi et al., 2014). However, as noted in (Sa et al., 2016), the problem of creating a fast and reliable fruit detection system still persists due to challenges described above.

Colour drives many modern approaches in object detection and classification, from classical engineered approaches such as Euclidean clustering of RGB-D point clouds (Nguyen et al., 2016) to sensitive colour specific units shown in two popular Deep Network architectures, VGG19 and AlexNet. Earlier layers of their respective architectures are shown to be sensitive to colour and not to class (Engilberge, Collins and Süsstrunk, 2017). The latter paper analyses the importance of colour sensitive features and concludes by validating the use of pre-trained models as feature extractors, since earlier network layers are shown to be sensitive to colour and not to specific classes; generalised detection power is fuelled by colour sensitive feature units in earlier layers. The purpose of data collection is to validate our approaches to improve our deep network architecture with an early fusion of perceptually uniform colour features to provide greater invariance to the luminance and viewpoint problem object detection architectures experience in production. A data set containing image triplets of Strawberries is presented, captured at three distinct angles, shown in Figure 4.1b. To constrain our training stages, our deep network is trained on a single viewpoint $V_1$ and validated on unrepresentative viewpoints (although different viewpoints may contain the same objects), $V_{2-3}$ which both show great change in luminance and class structure. Strawberries are visually distinguishable with their vibrant red colours when ripe, less so when unripe due to similarities with the background class, and colour is shown to be directly related to intrinsic attributes of the fruit, (Meulebroeck, Thienpont and Ottevaere, 2016) validating the motivation of using approximated colour opponent mechanisms in our system introduced in Section 4.4.

Variable weather, illumination, seasonal conditions, growth cycles, human picker interaction, speed and multiple views are all constraints that detrimentally impact the performance and robustness of approaches mentioned in this section. Automated horticultural systems must be able to (a) detect the produce of interest (b) infer aspects of the produce appearance (e.g., size, ripeness, heath) and (c) parse guidance (such as position) information to other systems. Research to date has largely focused on proof-of-principle studies investigating the best combination of sensing hardware and software processing (for a review, see (Nguyen et al., 2016)). Standard CCD and CMOS sensors have been used (Maldonado and Barbosa, 2016; Diago et al., 2012) (An et al., 2016; Dey, Mummert and Sukthankar, 2012) to detect pineapples (W S Qureshi et al., 2014), peppers (Song et al., 2014), pomegranates (A. Roy et al., 2011), apples (Linker, Cohen and Naor, 2012) and strawberries (G. Feng, Qixin and Masateru, 2008). More recently, 3D imaging using time of flight (L. Sun, Cai and Zhao, 2015; Chaivivatrakul et al., 2014; Kusumam et al., 2017), structured light (Nguyen et al., 2016) or stereo-camera methods (Rajendra et al., 2009; Hayashi et al., 2010; Scarfe, 2012; Font, Tomàs Pallejà et al., 2014; Kaczmarek, 2017) have been used, enabling more precise analysis of pose and shape information from objects in a scene.

The current SOTA in machine learning for object detection is represented by deep learning methods which have been also applied to agriculture with excellent results, such as the DeepFruit network (Sa et al., 2016). The network achieves very good performance, but its robustness to natural variations is unknown since data acquisition in the presented work relies on heavily controlled lighting conditions (i.e., visible and near infrared LEDs in combination with a canopy) and the use of multi-modal sensing (*RGB* and Near Infra-red (*NIR*)). Our dataset looks to reduce these issues by providing a benchmark to test methodologies within horticulture specifically, while remaining simple to capture.

## 4.2 Multiple Viewpoint Data Acquisition

This section presents a longitudinal data set recorded in a real working agricultural environment containing RGB, stereo infrared images and point clouds as well as providing camera parameters, localisation and metadata describing capture conditions such as humidity and temperature. This data set was created in order to capture the variance present in natural outdoor conditions. The performance of any machine learning method, including deep neural networks, depends heavily on the quality and quantity of training data sets. Large data sets at the time of publication do not exist for real-world agricultural settings, and small data-sets will struggle to encompass all variations of parameters such as illumination. Therefore, the development in this field must look at alternate methods to boost performance and generalisation even when using small amounts of data. Sa et al. (2016) note that variation in outdoor agricultural environments affects vision systems greatly and many of the introduced factors such as sunlight and weather are detrimental to the performance of machine vision systems. Current computer vision systems are either developed in controlled indoor conditions that avoid real-world constraints or use external equipment to minimise illumination variance in their data sets (see adaptation of data set from (McCool et al., 2016) by (Sa et al., 2016)).

In total 6189 images were captured over 2 months, August and September 2018, and 150 were manually annotated. Table 4.1 shows the number of images across each view that was used in the model training and testing stages. All the strawberries were labelled into two classes, Ripe and Unripe. The production site where the data was captured was at the University of Lincoln research farm at Riseholme campus. Two poly tunnels with table-top strawberry rows were constructed, one row was tagged with visual markers (Lightbody, Krajník and Hanheide, 2017) to indicate the points along the row where data should be collected, and the subsequent data collection process occurred singularly on this tagged row three times a day three times a week to capture various light intensities, weather conditions and plant growth stages. The species of strawberry were *Amesti*, captured at the flowering and fruiting stages of the plant.

Table 4.1: Distribution of images across training and testing sets for $V_1$, $V_2$ and $V_3$.

| **Viewpoint** | **Training** | **Testing** | **Total** |
|---|---|---|---|
| $V_1$ | 120 (80%) | 10 (6.6%) | 130 |
| $V_2$ | 0 (0%) | 10 (6.6%) | 10 |
| $V_3$ | 0 (0%) | 10 (6.6%) | 10 |
| Total | 120 (80%) | 30 (20%) | 150 |

The images were captured at 1920 × 1080 px resolution and the network was trained without resizing them. The data acquisition rig is visualised in Figure 4.1b and shown in Figure 4.2. Three cameras were mounted 45 degrees apart to capture as much spatial information from the strawberry crops as possible. The top, middle and bottom cameras will each be referred to as $V_1$, $V_2$ and $V_3$ respectively from here on. Capturing at these three distinct points ensured the information captured by each camera would have a good spread of dissimilar semantic information about each class. For example, $V_1$ and $V_3$ would contain visually very different information for each class, whereas $V_2$ would share more instance information about each class with viewpoints $V_1$ and $V_3$; As shown in Figure 4.1b. This enabled us to compare the impact that viewpoint variance had on model performance. Each class could be trained on a training set that contained information from each viewpoint $V_{1-3}$, however the experiments were configured so that only $V_1$ would be used in the model training stage. This was done to simulate the real-world effect of illumination, shape and texture changes introduced by unpredictable viewpoint variations caused by indeterministic environmental effects such as weather and human interaction.

During data collection the acquisition rig was mounted on a modular robotic platform

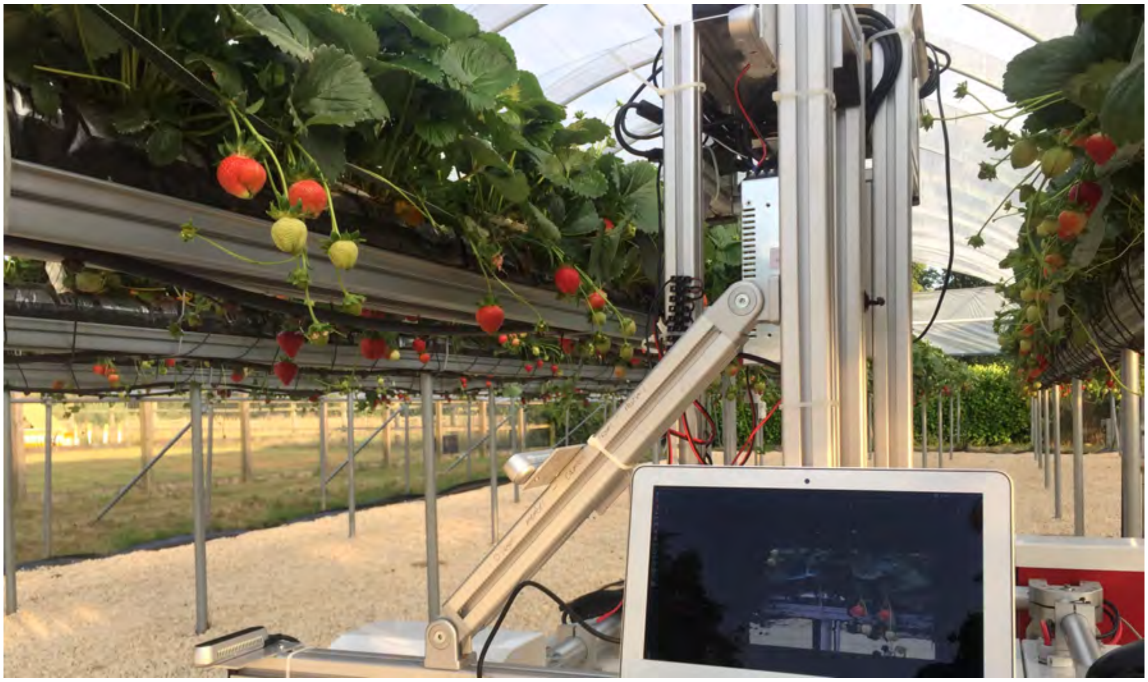


Figure 4.2: The image acquisition rig inside the strawberry polytunnels.

Thorvald (Grimstad and P. From, 2017) and moved incrementally to each visual marker to ensure consistency between data collection cycles.

The data set presented should be considered a complex data set in the sense it contains classes with heavy occlusion and highly varied illumination. The images were captured over a period of 24 days, with an intraday variance of 11 hours. It contains two classes: Ripe Strawberry and Unripe Strawberry with uneven distribution as shown in Table 4.2.

Table 4.2: Distribution of labelled classes across training and testing sets for $V_1$, $V_2$ and $V_3$.

| **Bounding Boxes** | **Ripe** | **Unripe** | **Total** |
|---|---|---|---|
| Training | 673 | 2680 | 3353 |
| Testing | 217 | 649 | 886 |
| Total | 890 | 3329 | 4219 |

An example of the split used for training is shown in Figure 4.4. The difficulty of the data set is reflected in the quantitative assessment later in this chapter. The data set has been made publicly available, in order to support key advances in this research area. Strawberry Flower and Bad/Diseased Strawberries are also included, but not used in the evaluation in this chapter.

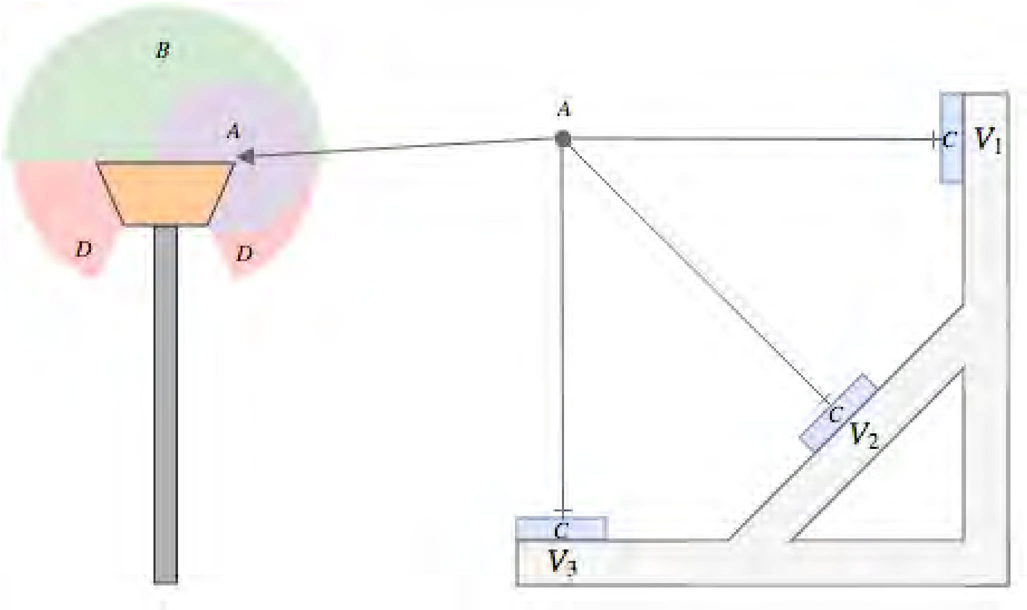


Figure 4.3: Data acquisition setup where $A$ is the optimal capture point, $B$ is the plant canopy, $E$ is the camera equipment and $D$ is the fruit and calyx region.

## 4.3 Vegetation Indexes

Segmentation of soft fruits has seen many different approaches in research, many of which are mentioned below. However, inspiration for the segmentation process in this proposal is interestingly inspired from research by Mathibela, Posner and Newman (2013). In which they use the opponent colour model for the detection of roadworks in images, this information is then used to increase the accuracy of autonomous vehicle navigation through assessment of how valid prior maps are to the current situation. The opponent colour model is based on a theory proposed by MacLeod et al. (2006) on how colour is perceived by humans, he proposed that cone photo-receptors in the eye are linked together in three independent colour-pairs: black and white, green and red and finally blue and yellow. When one member of a colour pair is stimulated, the other inversely stimulated. This theory explains why some colour pairs such as greenish reds and yellowish blues are never observed. The opponent colour model is represented by some widely used colour spaces; L*a*b* colour space consists of three channels similarly to the opponent colour model, L*, a* and b* which resemble the black and white, green and red and blue and yellow colour pairs of the opponent colour model respectively. The L*a*b* colour space, referred

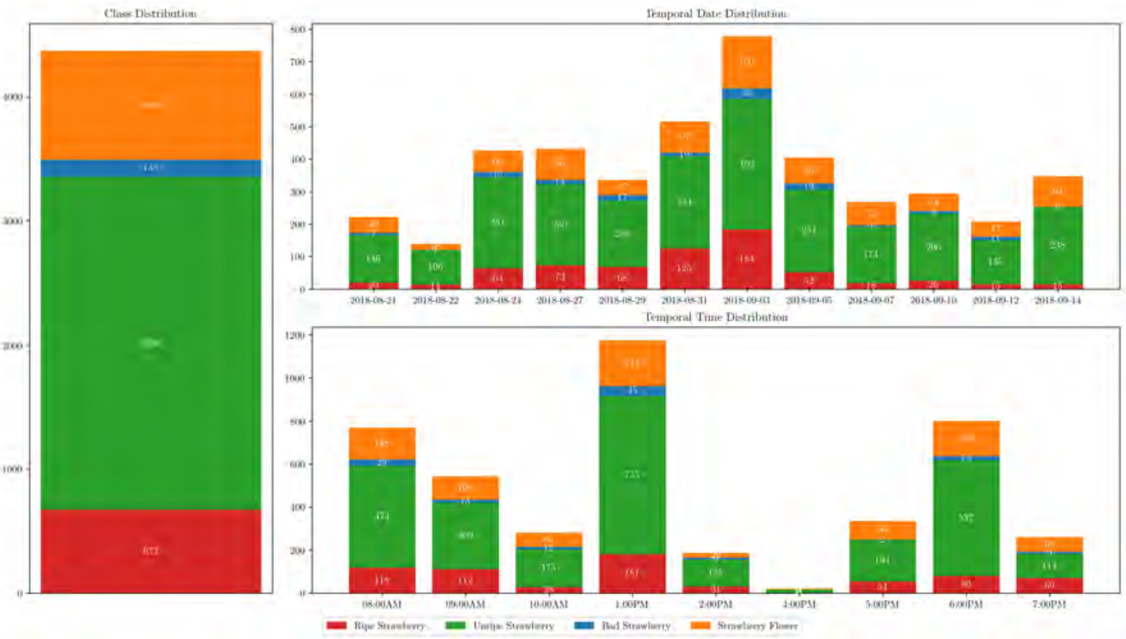


Figure 4.4: Class and time description of the Strawberry training dataset split.

to as Lab or CIE Lab, has been used in some crop segmentation approaches before (Achanta et al., 2012)(X. Bai et al., 2014). (X. D. Bai et al., 2013) use the colour space due to the uniform distribution of colours within it, W S Qureshi et al. (2014) use it to segment pineapples and for separating super pixel boundaries similarly to how it is also used in Achanta et al. (2012) for generating super-pixels.

The green and red opponency results in natural segmentation of red crops from backgrounds comprised of green leaves and branches. Since the objects generally surrounding strawberries and tomatoes are green vegetation, this would suggest that reasonably good prior segmentation results can be obtained. Since simple colour based transformations such as Excess Green Vegetation Index (ExG) and Excess Green minus Excess Red Vegetation Index (ExGExR) only perform well for green crop segmentation (X. D. Bai et al., 2013), this provides a fast and simple alternative for crop segmentation where maturity deviates from this assumption. Figure 4.5 shows some preliminary green-red colour pair segmentation results in the Lab colour space, compared to Excess Red Vegetation Index (ExR) with an arbitrarily chosen threshold of 0.6. As you can see in this comparison, the green and red opponent channel appears to have consistently lowered error results. The third, first and second images are captured in different lighting conditions, the third being the most illuminated and the second having the lowest illumination, while it can be seen that lighting is major detrimental to the ExR algorithm, the Lab results appear to be stable showing an invariance to illumination conditions. This is due to colour representation in RGB varying greatly based on the illumination in the scene, whereas in the Lab space colour is represented more uniformly mainly due to illumination being represented solely by its L channel.

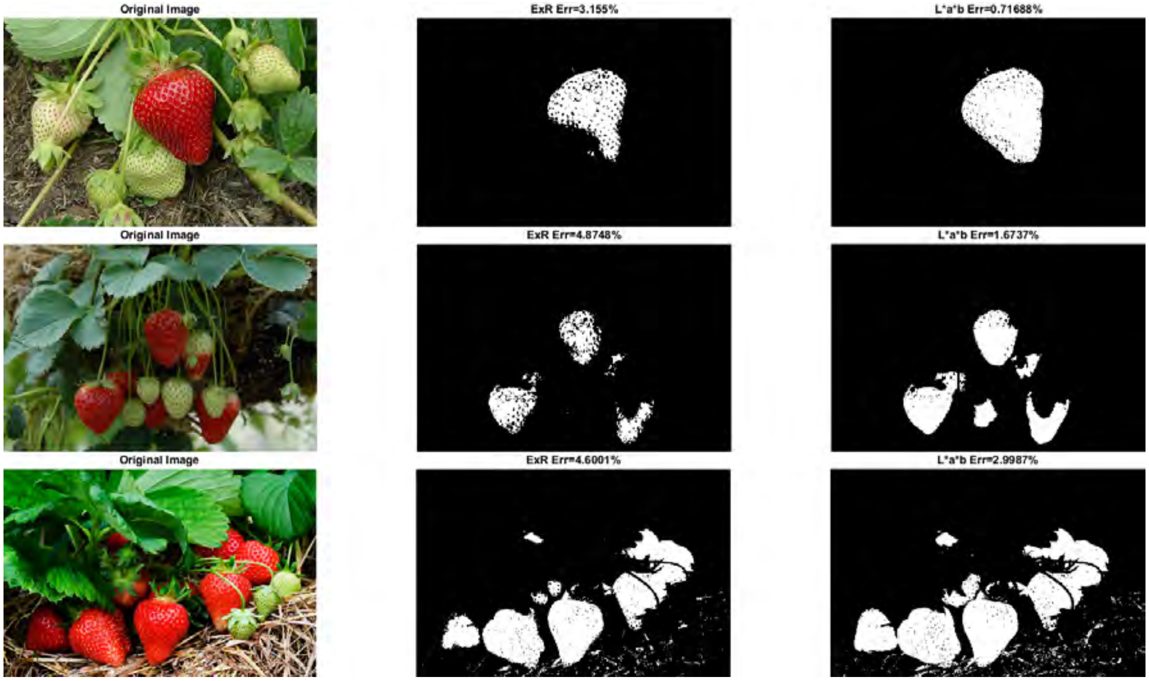


Figure 4.5: Segmentation quality assessment (OTSU threshold) using ExR and CIE L*a*b

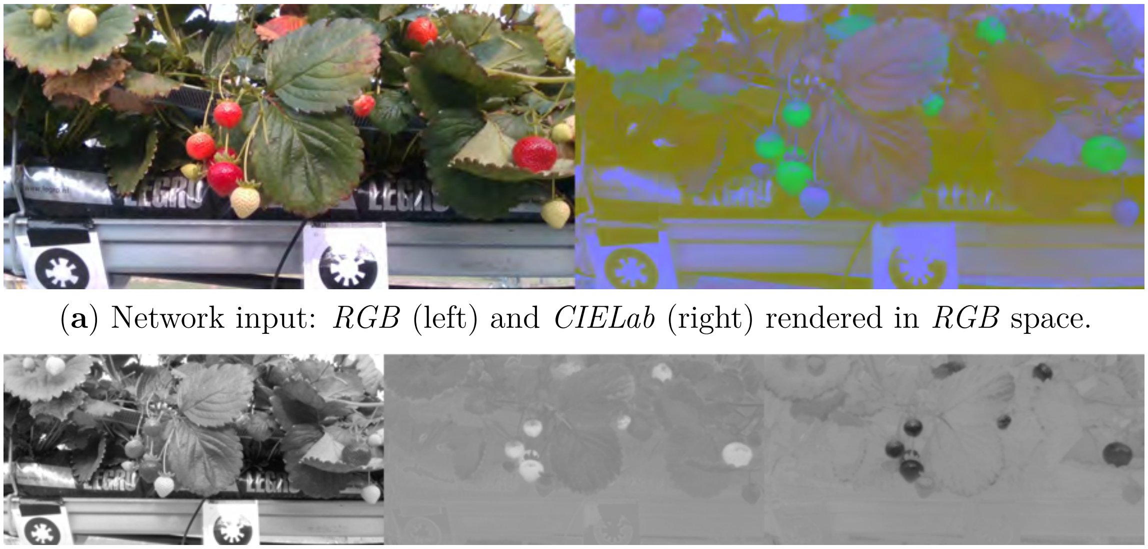
## 4.4 Colour Opponent Process

The approach presented in this chapter is based on colour opponent process theory, network input features are modelled as an approximation of logarithmic function responses of photo-receptive materials in the human eye. Colour opponent process theory explains how the human vision system perceives colour information (MacLeod et al., 2006). It explains colour vision as the combination of energy differences between opponent energy pairs. Red versus green, blue versus yellow and finally white versus black (Mathibela, Posner and Newman, 2013). The first two opponent pairs model the perceived colour, and the later opponent pair determines the perceived luminance of an observed object. Simply, the opponent process is a translation between rod/cone responses to the combination of colours perceived by humans.

Paul Newman introduced colour opponency in, (Mathibela, Posner and Newman, 2013) where they used colour opponency to extract information from a scene in real-time for applications within SLAM. They wanted a fast way to estimate roadwork activity in an image so that his global model could be updated. So they used the idea of colour opponency and the fact that the design of roadworks are almost modelled based on the colour opponent theory since they are usually only a single colour of each colour opponent pair i.e., yellows, greens, reds and blues. He found that using colour opponency, they could successfully predict roadwork activities in scenes with great accuracy. Similar attributes are shared for soft-fruits, a visual distinction from background vegetation.

The motivation behind this is that luminance is contained entirely within a single opponent pair such that the three channels represent perceptually uniform colour, helping reduce one of the biggest constraints visions systems face; the impact of variable illumination in the environment. In computer vision, the *CIELab* colour space approximates perceptually uniform human vision, which means any change in the *CIELab* space should induce a similar change in the colour perceived. *CIELab* has three channels, each representing one of the colour spaces mentioned above, $L$ represents white versus black, $a$ represents red versus green and $b$ represents blue versus yellow. Figure 4.6 visualises the *RGB* data and *CIELab* data in the *RGB* colour space for strawberries.

Many visual processes have been presented to explain the human vision system, more accurately how colour information is perceived. One of the leading theories is Opponent-process theory. It explains colour vision as the combination of energy differences between opponent energy pairs. Red versus green, blue versus yellow and finally white versus black. The first two opponent pairs model the perceived colour, and the later opponent pair determines the perceived luminance of an observed object. The opponent process is a translation between rod/cone responses to the combination of colours perceived. Conceptually, this theory can help understand



(**a**) Network input: *RGB* (left) and *CIELab* (right) rendered in *RGB* space.

(**b**) CIE Lab channels: Lightness (left), green-red opponent channel (mid) and blue-yellow opponent channel (right).

Figure 4.6: Network Input: Visualisation of *RGB* (top left) and *CIELab* (top right) used in model training. It is evident in the opponent feature channels (bottom row) of *CIELab* that this colour space is appropriate for fruit detection due to the maximal and minimal response of fruit pixels.

conditions such as colour blindness, and phenomena such as negative after image, and describe why colours that stand in opposing pairs such as reddish greens and blueish yellows cannot be perceived since the sensory processes of this model and colour pairs are antagonistic.

The opponent colour model has been applied in research numerous times (Achanta et al., 2012; Mathibela, Posner and Newman, 2013; X. D. Bai et al., 2013; X. Bai et al., 2014; Teimouri et al., 2014), usually to tackle variable luminance. *CIELab* can naturally segment regions containing perceptually opposing colour channels. In Mathibela, Posner and Newman (2013) they state that objects have been designed/exist to be easily perceived by the human visual cortex. Things are described as easily perceived when the colour features maximally activate single components in each opponent pair. In an example, ripe and unripe strawberries both activate different ends of their respective red/green opponent pair. *CIELab* is used in (Mathibela, Posner and Newman, 2013) to detect the presence of roadworks without ever explicitly modelling any of the objects; traffic signs are usually bright oranges (yellow), blues and reds, corresponding to maximal activation of one component in each colour pair. *CIELab* extracts the visual saliency of colour features in objects and is used in research to model *RGB* more uniformly. As in (X. D. Bai et al., 2013) where it is used to generate colour models invariant of lighting and illumination changes.

In the follow-up paper, (X. Bai et al., 2014) they improve on the performance by clustering the input through Particle Swarm Optimisation into vegetation/non-vegetation classes before then generating the model for only vegetation regions. The *CIELab* colour space was used in a novel super pixel algorithm presented in (Achanta et al., 2012), where the colour space was used to achieve similar performance to SOTA super-pixel algorithms through a simpler algorithm deemed Simple Linear Iterative Clustering. Finally and most notably, the *CIELab* space is used in (Mathibela, Posner and Newman, 2013) to detect the presence of roadworks without ever explicitly modelling any of the objects associated to it. This is achieved by exploiting the visual salience of roadworks and their design, which maximally activates opponent colour pairs in the human perception system to generate and classify signatures of the *a* and *b* *CIELab* components.

The intuition behind these papers is that by using the *CIELab* colour space they can naturally segment regions in images that show perceptually meaningful opposing colour channels and some papers state that artificial objects as well as naturally occurring ones have been designed/exist to be easily perceived by the human visual cortex since the colours maximally activate one side of each opponent pair (Mathibela, Posner and Newman, 2013). In example, traffic signs generally fit this description since they are generally bright oranges (yellow), blues and reds. In, (X. D. Bai et al., 2013) they use *CIELab* to generate colour models invariant of lighting and illumination changes. The constraints of their approach are that they build the model at discrete levels in the *L* channel meaning the computational complexity is high and due to the simple morphological operations the approach would not work well in complex multi-class environments.

The Deep Fruits system (Sa et al., 2016) attempts to solve the luminance problem by fusing multiple spectra, the visual *RGB* and infrared images. The aim in this chapter is to solve a similar problem by modelling the luminance through antagonistic colour pairs instead. The benefit of our approach is that it only requires *RGB* data from a standard camera and a non-linear transform described in Equation 4.1 and Equation 4.3 to convert between the two colour spaces. First to convert to the *CIE XYZ* space, described in Equation 4.1 where $Y$ is modelled as luminance, $Z$ is quasi-equal to blue stimulation, and $X$ is a linear combination of cone response curves. In Equation 4.1 the values used for $D$ are calculated with regard to the *D65* illuminant (Schanda, 2007).

$$\begin{gathered}\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = D * \begin{bmatrix} R \\ G \\ B \end{bmatrix} \\ D = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix}\end{gathered} \tag{4.1}$$

Once *RGB* values have been transformed to *CIE XYZ* colour space, a non-linear transformation described in Equation 4.2 and Equation 4.3 is applied to directly convert to *CIELab* space. In Equation 4.2 the values used for $X_n$, $Y_n$ and $Z_n$ are $X_n = 95.047$, $Y_n = 100.000$, $Z_n = 108.883$ and are calculated under the *D65* illuminant (Schanda, 2007). Note that this conversion from *RGB* to *CIELab* is device dependant and must be converted to an absolute colour space such as *CIE XYZ* or *sRGB*.

$$\begin{aligned} L &= 116 f\left(\frac{Y}{Y_n}\right) - 16 \\ a &= 500\left(f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right) \\ b &= 200\left(f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right) \end{aligned} \tag{4.2}$$

Where $f(x)$ adds the non-linearity and $\delta$ is equal to $\frac{6}{29}$.

$$f(x) = \begin{cases} \sqrt[3]{x}, & \text{if } t > \delta^3 \\ \frac{x}{3\delta^2} + \frac{4}{29}, & \text{otherwise} \end{cases} \tag{4.3}$$

## 4.5 Fruit Detection System

The fruit detection system presented in this chapter initially retrieves an image from either the data set (training) or the camera (testing). Afterwards, the *RGB* data is captured and the transformation described in Section 4.4 is applied to convert into *CIELab* space. The two images (RGB and L*A*B*) are then stacked depth-wise $C = 6$ to form a tensor of size $W \times H \times C$ where $C$ is the number of channels, $W$ and $H$ are the width and height respectively. At this point, the fused image tensor is input into the network, where a convolutional layer with stride 2 increases the number of channels to 64 (chosen number of filters) via 2D convolutions of kernels of size $7 \times 7$. The ResNet-18 feature extractor then generates four feature maps from four blocks of a $3 \times 3$ convolution and ReLU activation function, repeated twice at increasing number of input channels $D$. The latter three feature maps are then used in the feature pyramid network to generate five multiscale feature maps, this process is visualised in Figure 4.7. For each scale created, a classification and regression subnet are applied. Respectively, the subnets output tensors of size $K \cdot A$ and $4A$ where $K$ are the classes and $A$ are the predefined region proposals (anchors). In summary, the classification subnet outputs class predictions for each anchor and the regression subnet outputs $4A$ regressed bounding boxes at each spatial location.

### 4.5.1 Early Feature Fusion

In our approach, the selected colour spaces at the model input level are aggregated and our first layer of the model architecture is defined accordingly. The first layer is defined as having $D$ channels, where $D = 3$ for either *RGB* or *CIELab* and $D = 6$ for the early fusion model composed of both *RGB* and *CIELab* channels. A late fusion approach was also considered, which would combine predictions from two networks

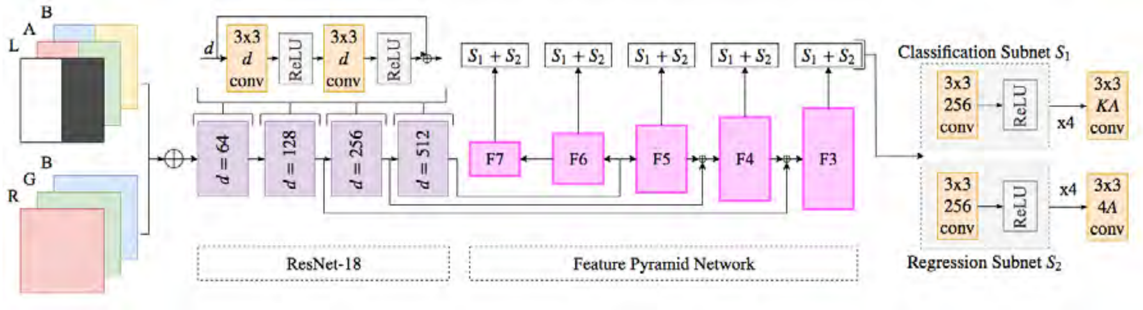


Figure 4.7: Model Architecture: RetinaNet implementation showing early fusion of *RGB* and *CIELab* features, where $Fn$ are convolutional layers with resolution $2^n$ of the input and 256 channels. $F6$ and $F7$ are obtained by a $3 \times 3$ convolutional layer with stride 2 of $F5$ and $3 \times 3$ convolution with stride 2 and intermediate ReLU activation layer of $F6$ respectively.

trained on *RGB* and *CIELab* inputs respectively, but as found in (Sa et al., 2016) the performance gain is small and incurs linear increase of computational cost per network. In this case, it was discovered that doubling the resources necessary for this small performance increase was an inefficient method of combining the information contained within each input transform domain, as was the finding in (Sa et al., 2016). Our early feature fusion method could accurately detect both classes of strawberry in our image datasets for observed viewpoints as shown in Figure 4.8 and unseen viewpoints.

### 4.5.2 Feature Extraction

Using the colour opponent process as model input attempts to maximise luminance invariance when training deep networks. Where variable luminance is a problem intrinsic to the data and contained classes, deep networks face other challenges. Some of the challenges are mitigated in this implementation through state-of-the-art approaches, such as ResNet-18 to help with the vanishing gradient constraint of deeper networks and Feature Pyramid Networks to mitigate issues with scale disparity between class samples in the training set. The combination of these approaches results in an architecture that can better learn features with high variation in luminance, spatial resolution (size of classes in input images) and intra-class balance (the ratio of class observations to other class observations, i.e., number of ripe strawberries objects in the training set to number of unripe strawberries).

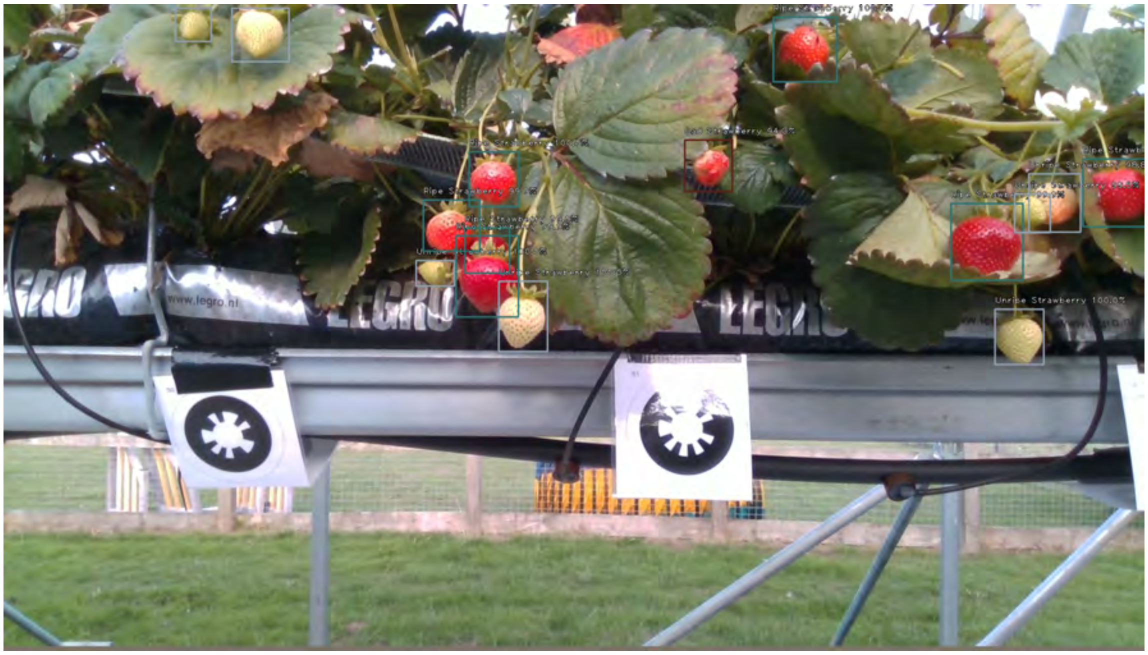


Figure 4.8: L*a*b*Fruits model output on a representative viewpoint (similar viewpoints exist in the training sample).

Due to the nature of object detectors for soft-fruit, the deep architectures demonstrated in Section 2.6 designed to accommodate a large number of classes usually have much larger feature extraction CNNs. Motivated by speed there is a trade-off, sacrificing a deeper and larger feature extractor and instead implementing a smaller network based on the ResNet-18 (K. He, X. Zhang et al., 2016) variant, shown in, Figure 4.7 that demonstrates comparable state-of-the-art performance as shown in Table 4.4.

### 4.5.3 Feature Pyramid Networks

Detecting objects at multiple sizes and scales is a difficult problem in machine learning and has seen many different approaches in the computer vision domain. As mentioned above it is unfeasible to construct a data set where the objective classes are well represented over all possible scales, illumination, shapes, colours and many other attributes. Such a data set would be need to be larger, meaning increased network training times and require infeasible levels of maintenance and annotation. One of the most popular recent advances in deep learning is Feature Pyramid Networks (T.-Y. Lin, Dollár et al., n.d.). An image pyramid consists of multiple feature maps at different scales and are generally the output of sequential convolutional layers (i.e., an input image down sampled by a factor of 2, $n$ the number of times will create a feature pyramid where each layer is a different scale of the original down sampled image). Until recently, this approach was mainly avoided due to the computational complexity and memory overhead they add to an architecture. To overcome the overhead, approaches have included using a single feature map from the feature pyramid, which looses the semantic information of the lower/higher layers or pyramidal feature hierarchies computed by sequential convolutional layers in a deep network (T.-Y. Lin, Dollár et al., n.d.). However, in these approaches there is a disparity between how semantically strong each layer is and therefore the effectiveness of this approach.

In, (T.-Y. Lin, Dollár et al., n.d.) they exploit the inherent multiscale, pyramidal hierarchy of deep convolutional networks to compute the Feature Pyramids at a much lower memory and computational cost while maintaining greater semantic information across each layer in the pyramid. The key contribution is the combination of lateral and top-down connections in the pyramid construction. Since lower level feature maps are not semantically strong the model will find it harder to learn from this information, generally deeper layers contain semantically strong information and are useful for classification/regression tasks. This approach uses top down connections, so the model can learn as effectively or up to as well as the deepest layer containing the greatest semantic information. This process is described in much greater detail, and clarification can be found for the terms in (T.-Y. Lin, Dollár et al., n.d.). Using this Feature Pyramid Network in our approach helps maximise scale/size invariance while maintaining similar performance to using a single layer for feature extraction as mentioned above. In the original paper they increased the accuracy by 8.0% on the MS COCO data set (T.-Y. Lin, Maire et al., 2014b) using this approach, for small objects generally missed, they increased the accuracy by 12.9%.

### 4.5.4 Architecture

As discussed above, the inclusion of the Feature Pyramid Network on top of the feature extractor used, named ResNet (K. He, X. Zhang et al., 2016), helps increase model performance over multiple scales. RetinaNet is a one-stage dense detector first presented in (T.-Y. Lin, Goyal et al., 2017), the motivation behind this architecture development came from the fact that one-stage detector performances were consistently trailing behind that of two-stage detectors such as Faster R-CNN (Ren, K. He, Ross B Girshick et al., 2015a). The benefit of using one stage detectors is the speed, however until RetinaNet the speed generally cost model accuracy. The model accuracy loss was attributed to class imbalance during model training. To which they mitigated with the novel loss function they introduced, named Focal Loss. This loss function reshapes standard cross entropy loss in a way that down weights well classified examples. With this new approach, RetinaNet outperformed all two-stage detectors and matched the speed of one stage-detectors at the time of publication. Our network architecture is based on RetinaNet to reduce the impact of class imbalance on network performance, this was key, especially due to the data imbalance between ripe and unripe strawberries in the data set.

In our approach, an 18 layer ResNet architecture with the discussed Feature Pyramid Network is used on top, calculating feature maps at three scales from the ResNet-18 feature extractors basic blocks. For each scale, the probability that objects are present is computed for each class $K$ and at anchors $A$, and then it regresses anchor boxes $A$ to nearby bounding boxes present in the GT. To achieve this, two very similar subnets are used, a classification subnet and a regression subnet respectively. Composed of four $3 \times 3$ convolutional layers, each with a ReLU activation layer attached. For the classification subnet there is a final convolutional layer $K \cdot A$ of filters and for the regression subnet a $3 \times 3$ convolutional layer with $4A$ outputs. Both subnets have a final sigmoid activation layer attached to output binary predictions for $K \cdot A$ classifications and $4A$ regressed boxes per spatial location, respectively. These subnets are described in greater detail in (T.-Y. Lin, Goyal et al., 2017), where our implementation is based. Finally, $A$ the number (9 in our approach, and common of SOTA object detectors) of boxes are generated at each location and focal loss for the regression and classification subnets are calculated (using $\alpha = 0.25$, $\gamma = 2.00$ in our approach, similar to the original paper). This constitutes the final loss as the sum of both classification and regression focal loss. The model architecture used in our experiments is defined in Figure 4.7.

## 4.6 Experiments and Discussion

The following section presents our findings on reducing illumination and viewpoint variance on a challenging, real-world data set. It presents (a) benchmark results for models trained on *RGB* data in Section 4.4, (b) model results using the *CIELab* colour space, (c) model performance with early fusion of both colour spaces, (d) an evaluation of viewpoint (spatial) invariance between the three trained models describing a level of generalisation between unobserved views that alter the spatial appearance (shape, texture and colour of the class), and finally (e) a comparison to the Deep Fruits system (Sa et al., 2016) which similarly attempted to maximise illumination invariance but through multi-spectra fusion. Although it is taken further to test our proposed solution on unseen views, Deep Fruits was found to be the closest baseline.

$F_1$ scores and the mean average precision metric used in the ImageNet challenge (Deng et al., n.d.) are used in this chapter to evaluate the experiments. The equation to compute the $F_1$ score using precision and recall is presented in Equation 2.10. An object is considered correctly detected in our results when the predicted bounding boxes have an intersection over union (IoU) of at least 0.5 (50%) with the GT annotation. However, results are also provided using a value of 0.4 (40%) to enable more accurate comparison to the DeepFruits experiments. The justification provided for using the smaller intersection over union threshold in (Sa et al., 2016) is that objects in the data set are smaller than in the ImageNet challenge, therefore require less overlap. Values of 0.5 are used for non-maximum suppression.

### 4.6.1 RGB and Early Fusion Comparison

In order to determine the effect of perceptually uniform colour spaces on viewpoint invariance, three different experiments were conducted. The original motivation of this approach is that error due to variation in luminance of each class described in Section 4.1 would be minimised. To test this, the *CIELab* colour space was used in order to capture the colour feature components present in the image more uniformly and thus fortify the features learnt in the network. This method utilises an early fusion method, introduced in (Sa et al., 2016). A late fusion method that combined two separate models was also proposed, but it was determined that doubling the number of parameters present in the network, computation time and GPU utilisation was an insufficient method to deal with luminance for reaching greater viewpoint performance.

Table 4.3: $F_1$, Average Precision (AP) and Average Recall (AR) scores of *RGB*, *CIELab* and Early Fusion at 50% intersection over union (IoU) for each detected class. Bold indicates the best result in each row.

| Class | View | Score | RGB | CIE Lab | Early Fusion |
|---|---|---|---|---|---|
| Both Classes | $V_1$ | $F_1$ | 0.744 | 0.710 | **0.747** |
| Both Classes | $V_1$ | AP | 0.722 | 0.695 | **0.748** |
| Both Classes | $V_1$ | AR | 0.870 | 0.844 | **0.909** |
| Both Classes | $V_{2-3}$ | $F_1$ | 0.680 | 0.622 | **0.704** |
| Both Classes | $V_{2-3}$ | AP | 0.659 | 0.586 | **0.694** |
| Both Classes | $V_{2-3}$ | AR | 0.812 | 0.761 | **0.851** |
| Ripe Strawberry | $V_1$ | $F_1$ | 0.683 | 0.625 | **0.697** |
| Ripe Strawberry | $V_1$ | AP | 0.616 | 0.571 | **0.678** |
| Ripe Strawberry | $V_1$ | AR | 0.807 | 0.767 | **0.892** |
| Ripe Strawberry | $V_{2-3}$ | $F_1$ | 0.697 | 0.662 | **0.729** |
| Ripe Strawberry | $V_{2-3}$ | AP | 0.659 | 0.621 | **0.719** |
| Ripe Strawberry | $V_{2-3}$ | AR | 0.806 | 0.777 | **0.877** |
| Unripe Strawberry | $V_1$ | $F_1$ | **0.805** | 0.795 | 0.797 |
| Unripe Strawberry | $V_1$ | AP | **0.828** | 0.819 | 0.818 |
| Unripe Strawberry | $V_1$ | AR | **0.933** | 0.922 | 0.927 |
| Unripe Strawberry | $V_{2-3}$ | $F_1$ | 0.663 | 0.582 | **0.679** |
| Unripe Strawberry | $V_{2-3}$ | AP | 0.658 | 0.552 | **0.668** |
| Unripe Strawberry | $V_{2-3}$ | AR | 0.819 | 0.745 | **0.825** |

As shown in Table 4.3 in terms of $F_1$ the early fusion approach outperforms both *RGB* and *CIELab* by 2.4% and 8.2% respectively on $V_{2-3}$. On the unseen viewpoints the $F_1$ score is lower as was expected since no images from either of these orientations were included in the training data set. The early fusion $F_1$ score for $V_{2-3}$ is 4.3% less than the result on the singular view $V_1$, the small difference in scores compared to the 6.4% and 8.8% drop for *RGB* and *CIELab* show that this approach can better generalise to unseen views of the classes.

It can be seen in both Table 4.3 and Figure 4.9 that *RGB* and *CIELab* are both consistently outperformed by the early fusion method, while the fusion of these two features shows a great improvement over a singular approach alone. It is also

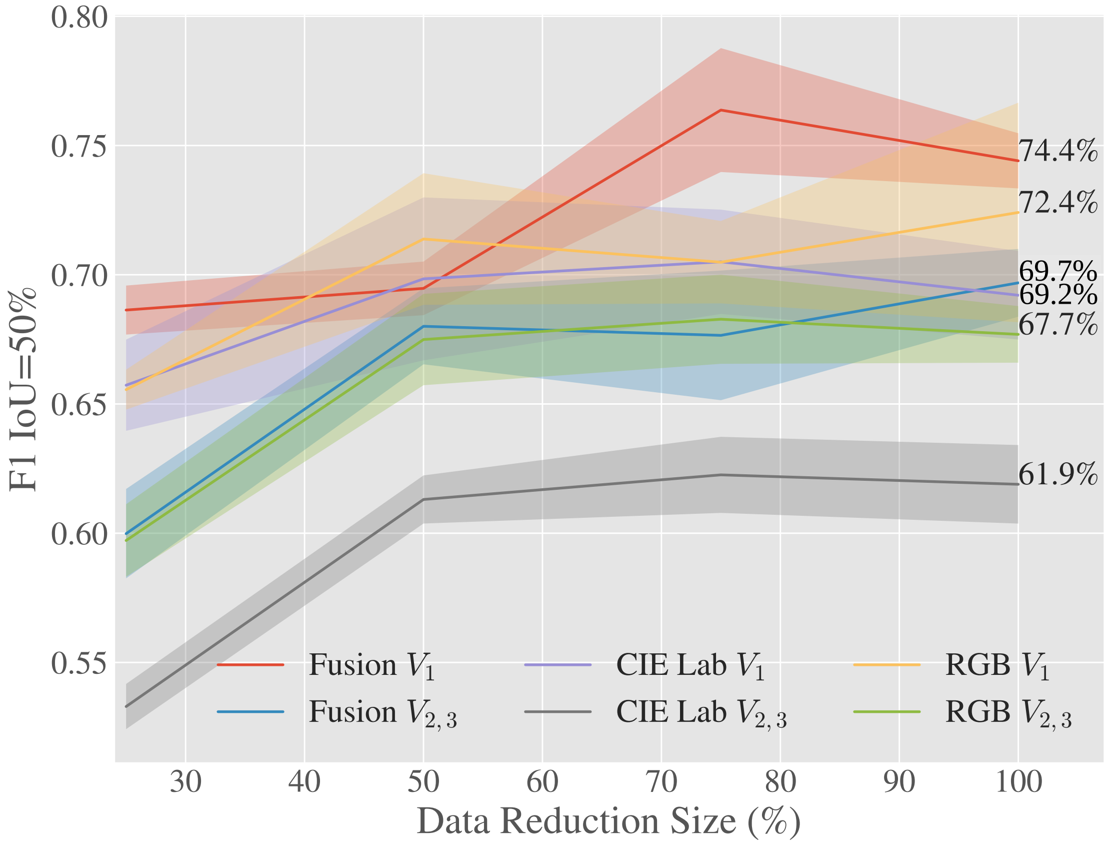


Figure 4.9: $F_1$ score for 50% intersection over union on $V_1$ and $V_2 + V_3$ testing data sets.

evident from this table that the lesser opponent class "Unripe Strawberry" has higher performance in *RGB* space, however is still beaten by early fusion when on unseen views. The early fusion approach demonstrates greater invariance to luminance and achieves excellent results on previously unseen views, described in Figure 4.1b.

Figures 4.9 and 4.10 show how the early fusion approach responds to reduced data sizes. The original data size of 120 images described in Table 4.1 is a small data set and in the experiments shown in these figures it was reduced by a factor of 25%, 50% and 75%. It is observed from the gradient of the line that the methods evaluated may perform better when more data is available. It can be seen that for a heavily reduced data set, the methods have much lower performance than they do with full data access. Data augmentation (only flipping was used) could be further used to boost the performance of early fusion by altering the luminance values within the fused data to uniformly alter colour features in the original *RGB* data. Figure 4.12 shows a false positive analysis of the early fusion network on the DeepFruits data set, you can clearly see in this graph most of the model inaccuracy comes from misclassification (BGR on Figure 4.12), data augmentation is one approach that could help minimise this issue. Figure 4.13 provides the output of our early fusion approach compared to $RGB$ alone, highlighting the cases where it surpasses baseline performance. The results shown are from a network trained on $V_1$ data and evaluated on the most dissimilar view $V_3$ to stress the detector performance across high variation of shape and luminance.

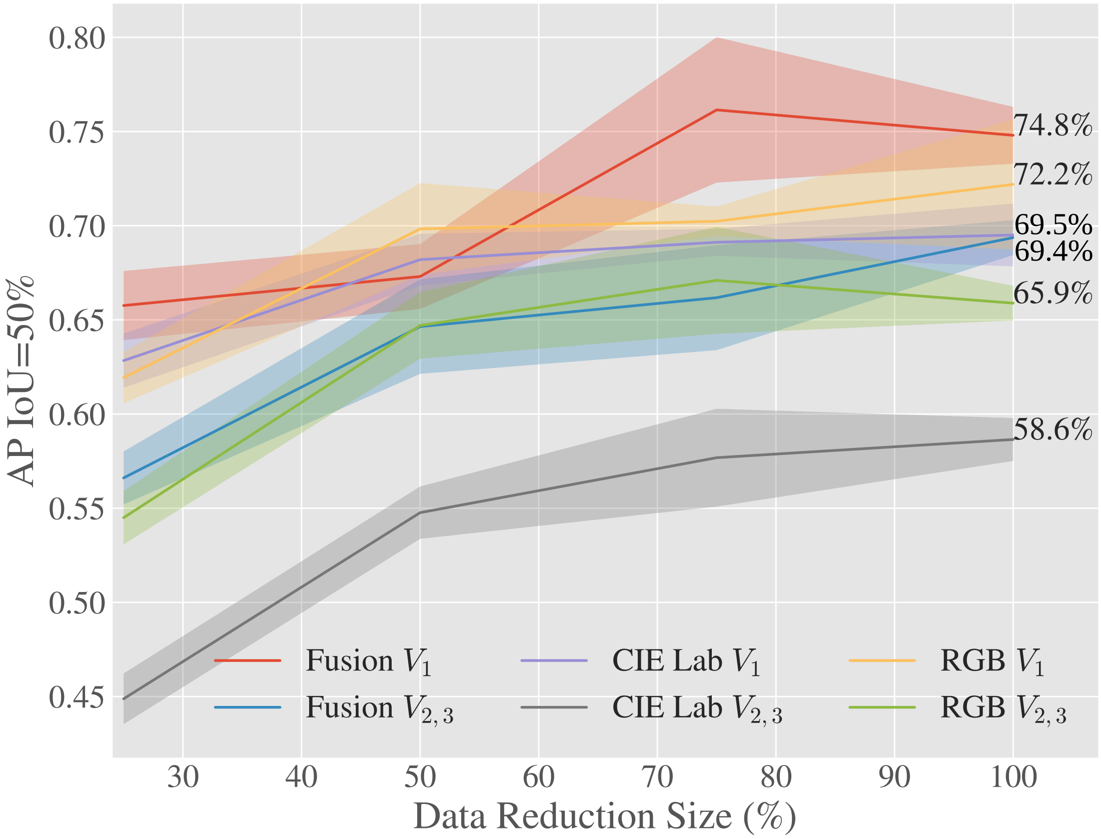


Figure 4.10: Average Precision for 50% intersection over union on $V_1$ and $V_2 + V_3$ testing data sets.

### 4.6.2 State-of-the-Art Comparison

In Table 4.4 the current state-of-the-art for deep learning based strawberry detectors is summarised. The columns in the upper table denote the number of images available in each data set, the availability of the data set, the camera viewpoint, whether the data set contains more than one imaging modality and finally if the data was captured

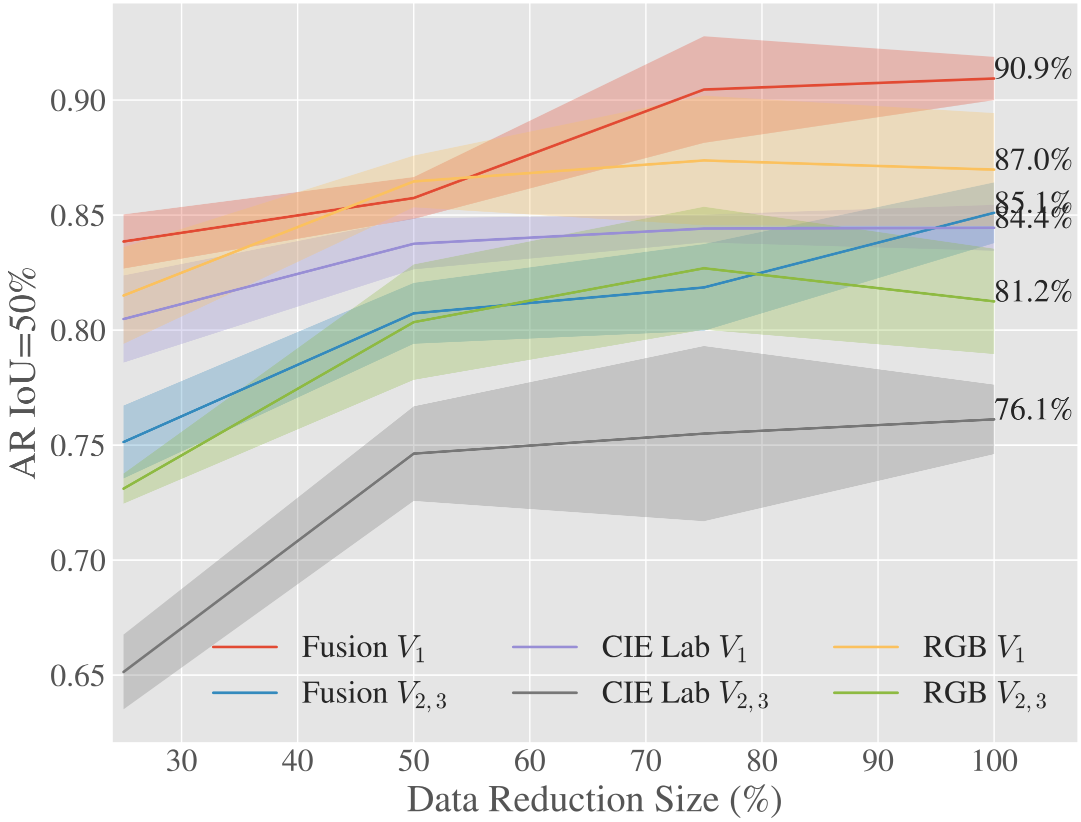


Figure 4.11: Average Recall or 50% intersection over union on $V_1$ and $V_2 + V_3$ testing data sets.

in controlled (e.g., under controlled lighting) or natural agricultural conditions. The columns in the lower table denote the network architectures used, the average precision (used in the ImageNet challenge (Deng et al., n.d.)), the $F_1$score at an intersection over union of 0.5 and finally the inference speed (the time taken to generate predictions from a given input) at a specific image resolution.

Methods were sought that were trained and tested with data-sets of comparable number of images and characteristics (viewpoint, multiple modalities, environmental conditions) allowing for fair and meaningful comparisons. Table 4.4 shows that the closest approach with data that was freely available was Sa et al. (2016). While this section does not compare with the remaining methods due to the inaccessibility of data-sets, Table 4.4 defines the performance scores and inference speed obtainable of different architectures on specific data sets. It is evident from the results that increased data set sizes and simpler viewpoints correlate to greatly improved accuracies, and it

Table 4.4: Summary of SOTA approaches to Strawberry Detection in Deep Learning.

| Method | # Images | Availability | Viewpoint | Multi Spectra | Controlled | Natural |
|---|---|---|---|---|---|---|
| Yu et al. (2019) | 1900 | ✗ | Side on (Close) | ✗ | ✓ | ✗ |
| Y. Chen, Won Suk Lee et al. (2019) | 12526 | ✗ | Aerial | ✗ | ✗ | ✓ |
| N. Lamb and M. C. Chuah (2018) | 4550 | ✗ | Ground | ✗ | ✗ | ✓ |
| Ge et al. (2019) | - | ✗ | Side on | ✗ | ✗ | ✓ |
| Sa et al. (2016) | **122** | ✓ | Side on | ✓ | ✓ | ✗ |
| L*a*b*Fruits (Ours) | **150** | ✓ | Multiple | ✓ | ✗ | ✓ |

| Method | Network | AP (IoU 0.5) | $F_1$(IoU 0.5) | Inference Speed (s) |
|---|---|---|---|---|
| Yu et al. (2019) | Mask R-CNN - ResNet-50 | - | - | 0.13 @ 640 × 480 px |
| Y. Chen, Won Suk Lee et al. (2019) | Faster R-CNN - ResNet50 | 0.77 | - | 0.11 @ 480 × 380 px |
| N. Lamb and M. C. Chuah (2018) | Single Shot Detector (SSD) | 0.84 | - | 0.61 @ 360 × 640 px |
| Ge et al. (2019) | Mask R-CNN - ResNet-101 | 0.81 | 0.90 | 0.62 @ 640 × 480 px |
| Sa et al. (2016) | Faster RCNN - VGG-16 | - | 0.79 | 0.39 @ 1296 × 964 px |
| L*a*b*Fruits (Ours) | RetinaNet, ResNet-18 | 0.75 | 0.75 | 0.07 @ 1920 × 1080 px |

is difficult to compare different methodologies due to the variable complexity levels in the respective data-sets and their availability. To address this issue, a baseline data-set is provided gathered in a real agricultural setting, including multi viewpoints and modalities for future benchmarking studies.

### 4.6.3 System Evaluation on Other Fruit

The Sa et al. (2016) paper noted the crucial component of autonomous fruit harvesters to be an accurate vision system, to which they attributed illuminance variation, occlusion and colour similarity between crop and background class to be the three major constraints limiting current approaches. They proposed a system based on the two stage detector Faster R-CNN that utilised early and late fusion of *RGB* and near Infrared imagery. Early fusion was a singular model with four input channels ($RGB + IR$) and late fusion trained two separate models (*RGB*, *IR*) and combined the detected objects in the final stage. Sa et al. (Sa et al., 2016) found late fusion to be the best approach achieving a $F_1$score of 0.838 for Sweet Peppers, however they also noted this approach requires double the number of network parameters, computational cost, power, GPU utilisation, training time and inference time. Ultimately concluding the small decrease in accuracy of the early fusion approach from 0.838 to 0.799 as a worthy trade-off.

Our approach is compared using a one stage detector, RetinaNet (with a smaller backbone) to theirs, in the following section. Experiments directly compare the performance of three of our networks trained on RGB, *CIELab* and Early Fusion inputs to their late and early fusion approach. The effectiveness is compared of our perceptually uniform features *CIELab* to that of IR to remove luminance variance within the data set, described in Table 4.5. The evaluation metrics used are in correspondence with the original paper (IoU at 0.4) and only classification scores greater than 0.9 are considered. The images contained in the sweet pepper data set were not as high resolution as with our Strawberry data set described in Table 4.1 but instead were 1296 × 964 px which were sampled to be divisible by 32 at 1280 × 736 pixels.

Table 4.5: Distribution of training and testing images used in DeepFruit models

| **Class** | **Train** | **Testing** | **Total** |
|---|---|---|---|
| Sweet Pepper (Capsicum) | 100 (82%) | 22 (18%) | 122 |

In this experiment, lower performance values are expected due to the fact a one stage detector is used over the two stage detector used in the original paper, as well as working with a data set less colour centric than ours. It is less colour centric due to the single class sweet pepper sharing very similar colour features with the background class. However, it is shown that the effectiveness of our approach at achieving what the addition of IR tried to achieve in DeepFruits, fortifying the prior viewpoint experiment results and luminance removal even when classes share much of their colour features that *CIELab* is based upon. Within our results presented in Table 4.6 there is a larger disparity between our early fusion approach and standard *RGB* showing an improvement despite the colour properties of the class and heavy occlusion.

Table 4.6: $F_1$scores of *RGB*, *CIELab* and Early Fusion at AP40 and 50 on the DeepFruit data set (0.799 at AP40). Bold indicates the best result in each row.

| **IoU** | **Metric** | **RGB** | **CIE Lab** | **Early Fusion** |
|---|---|---|---|---|
| 40% | $F_1$ | 0.789 | 0.763 | **0.793** |
| 40% | AP | 0.759 | 0.758 | **0.821** |
| 50% | $F_1$ | **0.789** | 0.738 | 0.772 |
| 50% | AP | 0.759 | 0.705 | **0.787** |

Moreover, the early fusion approach attempted in the Sa et al. (2016) paper failed at surpassing the $F_1$score of standard *RGB* and their approach using late fusion (which did outperform the *RGB* baseline) was dependent on simultaneous collection of IR data as well as training two separate networks, ultimately only showing a 2.2% increase.

Similar results are observed in Table 4.6. Our early fusion approach closely follows the *RGB* $F_1$scores and matches the performance obtained by Sa et al. (2016) (0.799). Our average precision scores outperform the standard *RGB* results by 2.8% and 6.2% for AP50 and AP40 respectively, suggesting the network more accurately classifies than it detects. Although statistically similar results to DeepFruits are shown, our approach is considerably faster (6.6×) at 0.06 s per image compared to 0.393 s and only needs *RGB* data instead of the *RGB* + Near Infrared data their approach requires. Interestingly, the early fusion approach maintains similar precision increases over the experiments as in Table 4.3 and Figure 4.10. In Figure 4.14 the early, fusion approach is compared to standard *RGB* on networks trained from the data provided by Sa et al. (2016) originally captured in (McCool et al., 2016). The results provided are representative of the results presented in Table 4.6.

Across experimental conditions, a consistent improvement in viewpoint invariance is observed using early fusion of *RGB* and CIE Lab. Utilising the RetinaNet architecture as a base allowed us to remove class imbalance through the Focal Loss function and improve detection for objects at multiple scales through the implemented Feature Pyramid Network. The method achieves near real time performance as seen in Table 4.7, where speeds are presented. Similar to what is stated as near real-time in relevant literature (W. Liu et al., 2015). Our early fusion approach adds to the architecture by providing results less sensitive to colour specificity of trained classes, and can be seen as a more generalised approach to solving this problem than introducing multiple spectra as in (Sa et al., 2016).

Table 4.7: Performance of the Early Fusion Network on an Nvidia GTX 1080 Ti, 11 GB (single forward pass).

| **Resolution** | **Model Inference Time** | **Frames Per Second** |
|---|---|---|
| 1920 × 1080 | 0.073 s | 13.71 |
| 1280 × 736 | 0.038 s | 26.33 |

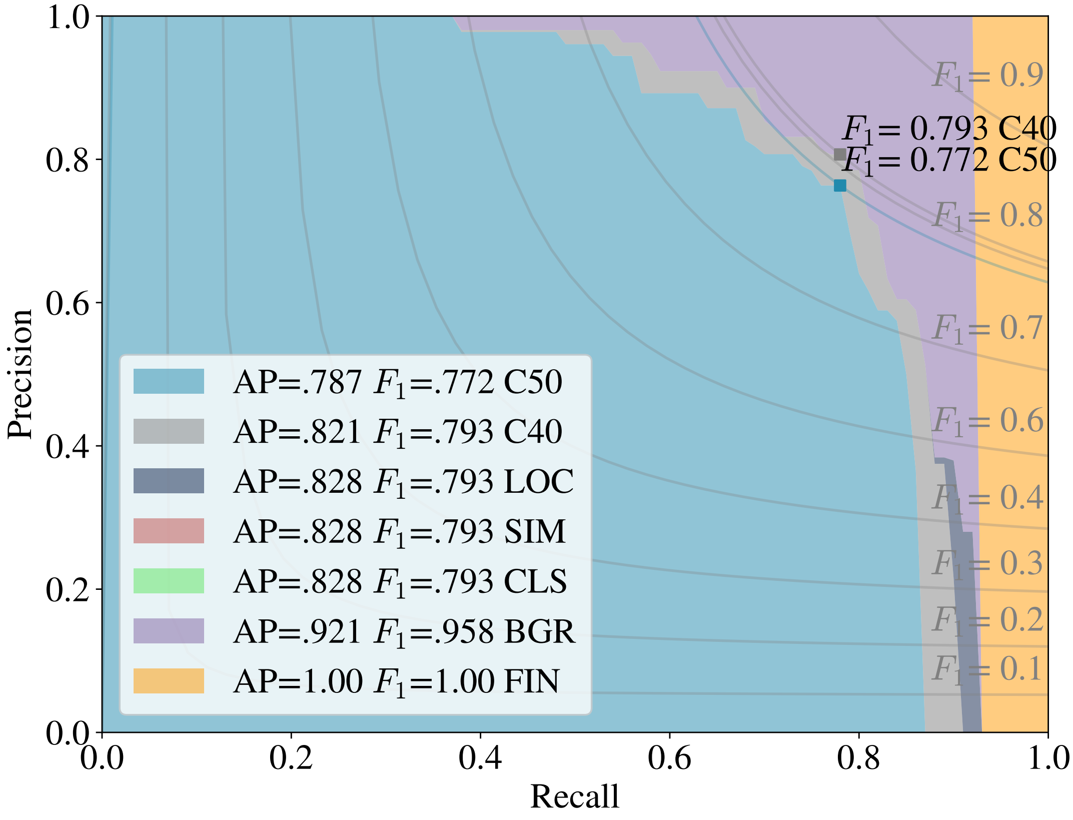


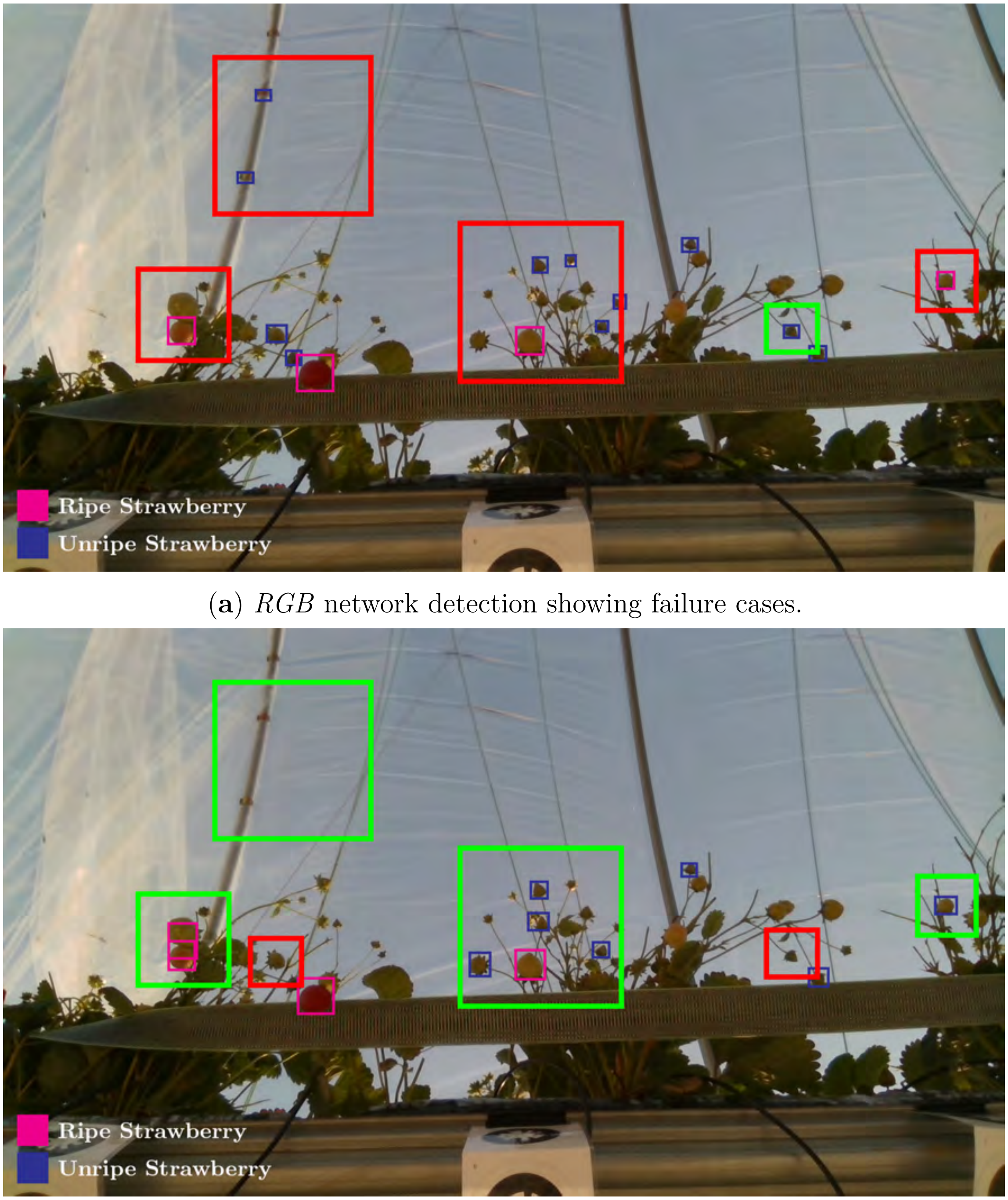
Figure 4.12: Precision-recall (PR) analysis: PR curves of the trained Sa et al. (2016) early fusion network. C50, C40 and LOC correspond to PR curves for IoU values of 0.5, 0.4 and 0.1, LOC when all localisation errors are removed. SIM and CLS when errors from similar categories and all classification labels are removed. BGR is the PR curve when all other class/background false positives are removed, and finally FIN shows PR containing no errors.

## 4.7 Conclusions

This chapter, presents an example of improving network performance on unseen data through a structured approach and analysis of the network input. A fusion of features was selected instead of modifying network architecture and depth to increase generalisation to non-representative images. The results observed indicate that using bio-inspired features can avoid increased model complexity for increases in accuracy and generalisation capabilities. For colour centric data classes, it is concluded that this approach shows great promise in increasing the robustness of trained deep networks in real world conditions. The addition of *CIELab* helps increase viewpoint invariance by training on more specific colour features across a wider luminosity range within each class. With the introduction of multiple viewpoints or unknown viewpoints the environmental factors contributing to the appearance of objects in a scene change and *CIELab* provides a more normalised representation of each class when they are colour centric (maximally activate a single component in colour opponent pairs).

A 2.4% and 8.2% increase is achieved with our early fusion approach on unseen viewpoints $V_{2-3}$ over the standard *RGB* and *CIELab* modalities alone. In comparison, the standard *RGB* and *CIELab* drop by 6.4% and 8.8% respectively for $\mathrm{F}_1$scores between viewpoints $V_1$ and $V_{2-3}$. Similarly, when applied to the DeepFruits data set, an AP score increase of 2.8% (IoU = 0.5) and 6.2% (IoU = 0.4) is gained over *RGB* alone. Our $\mathrm{F}_1$scores match those presented in the original paper, suggesting the added *CIELab* opponent features assist in classification of the detected objects more so than aiding the initial detection, since our obtained AP scores are consistently higher than *RGB* in all cases (2.8% and 6.2%). Our approach also gains a performance increase of 6.6 times that of the DeepFruits early fusion method utilising IR and only considering a single class. This improvement is likely to increase the applicability of the method to robotic fruit monitoring and harvesting systems that have limited computational and power resources.

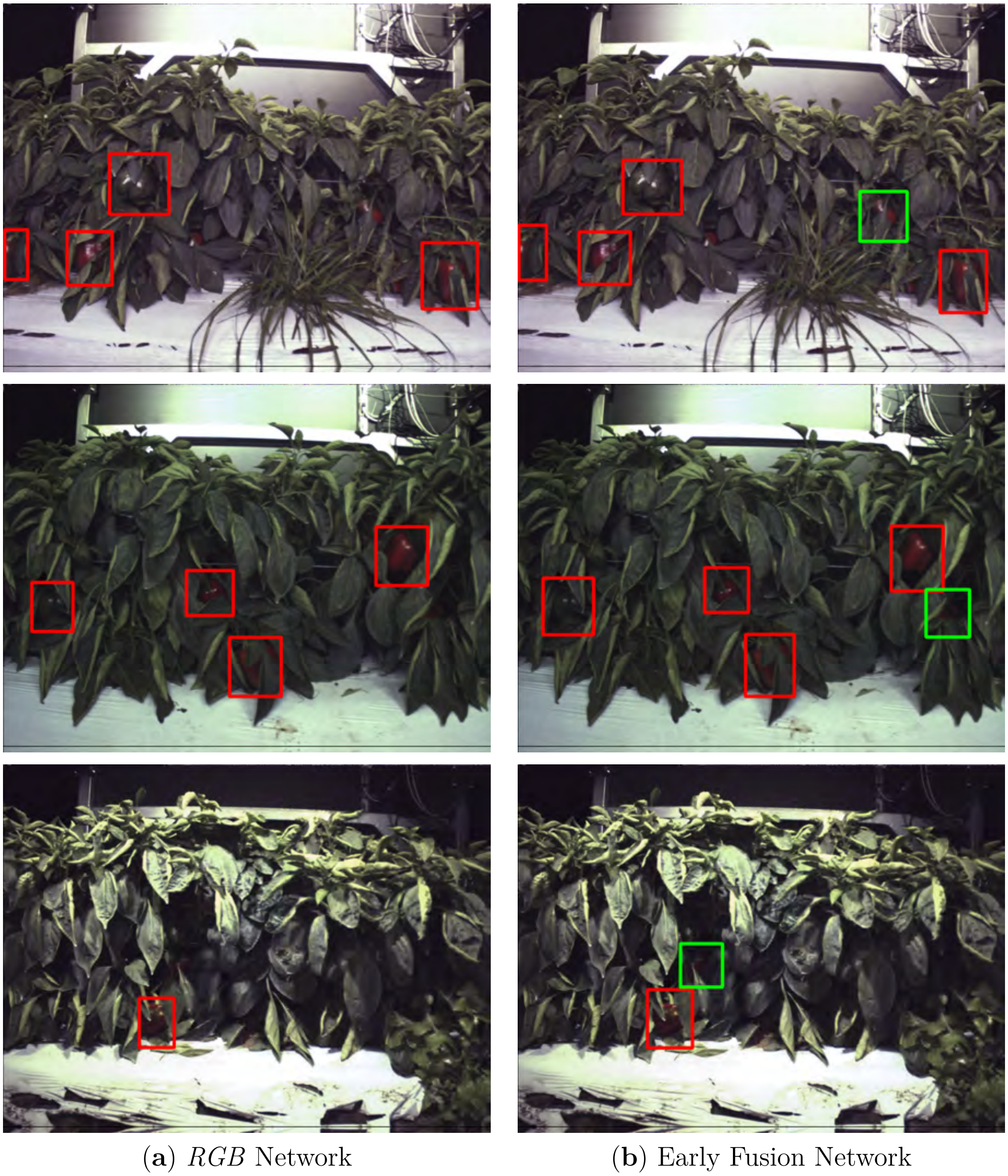
Leveraging *CIELab* colour opponent features with *RGB* helped mitigate some luminance variation in the testing sets. As can be seen in Figures 4.9 and 4.10 the early fusion approach appears to improve with larger amounts of data. Investigation into the benefits provided by this approach as the data set size increases would provide insight to the limitations and optimal accuracy increase through our proposed methods. As well as calibrating the cameras to improve the colour accuracy over multiple sensors. Visualisation of features and filters learned in the network would also provide intuition as to what the network is learning, which would be useful in seeing the difference between learnt *RGB* filters and colour opponent filters. To validate the removal of luminance further, this analysis could compare network activation for synthetically created Strawberries at variable luminosity, where uniform activation over variable parameters would indicate the removal of the detrimental effects of the parameter on overall accuracy. A single sensor was utilised in this experiment, to further our results calibrated colour from multiple sensors would show more representative results. Finally, analysis into accuracy increase with fewer classes or binned classes would show whether error is introduced through learning multiple classes, due also to the fact this chapter compares to Sa et al. (2016) which noted multi-class detection as further work than the scope of the paper.



(**a**) *RGB* network detection showing failure cases.

(**b**) Early Fusion network detection showing improved results.

Figure 4.13: Performance on difficult input: Early Fusion and *RGB* models evaluated on $V_3$, the view with the highest spatial variation. The early fusion approach maintains detection accuracy over huge illumination and shape alterations (introduced from viewpoint). Improved results are shown in green and detrimental results shown in red.


(**a**) *RGB* Network

(**b**) Early Fusion Network

Figure 4.14: DeepFruits evaluation: Early Fusion (right) and *RGB* (left) models evaluated on the DeepFruits Capsicum data. It can be seen that the early fusion approach more frequently detects objects the *RGB* network misses (highlighted in green).

# Chapter 5

# Tracking Soft-Fruit

Novel extensions of detect-to-track based object tracking frameworks are introduced to count soft-fruit in images (detect) and across image-sequences (track) in this chapter. Parts of which were published in *Robust Counting of Soft Fruit Through Occlusions with Re-identification* Kirk, Mangan and Cielniak (2021). Our framework is based on a re-identification and motion based tracker (DeepSort Wojke, Bewley and Paulus, 2017), the de-facto state-of-the-art tracker on the MOTA challenge at the time of publication, to count and track strawberry instances across frames addressing the baseline inaccuracy with the standard approach on small homogeneous clustered objects. Our main contributions are (1) a novel first re-identification and label probability based tracking framework, generalising the approach for multiple classes, applied on mobile robots for the purpose of counting fruits (2) extension of a popular re-identification tracking formalisation to embed contextual, shape and class information into association cost (3) four sequences of hand labelled Strawberry data for tracking in complex environments shared for bench-marking with the community, (4) validation of the counting accuracy for the purpose of yield estimation and (5) a Bayesian semi-supervised approach for generating re-identification datasets through weak trackers.

Fruit counting and tracking is a crucial component of fruit harvesting and yield forecasting applications within horticulture. A novel multi-object, multi-class fruit tracking system is introduced to count fruit from image sequences. First a residual neural network (RNN) is trained comprised of a feature extractor stem and two heads for re-identification and maturity classification. Then the network is applied to detected fruits in image sequences and utilises the output of both network heads to maintain track consistency and reduce intra-class false positives between maturity stages. The counting-by-tracking system is evaluated by comparing with a popular detect-to-track architecture and against manually labelled tracks (counts). Our proposed system achieves a mean average percentage error (MAPE) of 3% ($L1$ loss=7/233) improving on the baseline multi-object tracking approach which obtained an MAPE of 21% ($L1$ loss=41/233), validating the applicability of this approach for use in horticulture.

## 5.1 Counting Fruit with Detection Based Trackers

Fruit counting is a critical process in effective management of a fruit crop. It informs decisions on harvesting, labour management and yield estimates. Labour constitutes 65% of the total fruit harvesting cost and yield estimates typically have high uncertainty, motivating the need for accurate counting systems. With the advent of mobile agricultural robots and the success of convolutional neural network (CNN) based detectors, traditionally laborious tasks such as flower counting (a strong indicator of future yield) can now be automated.

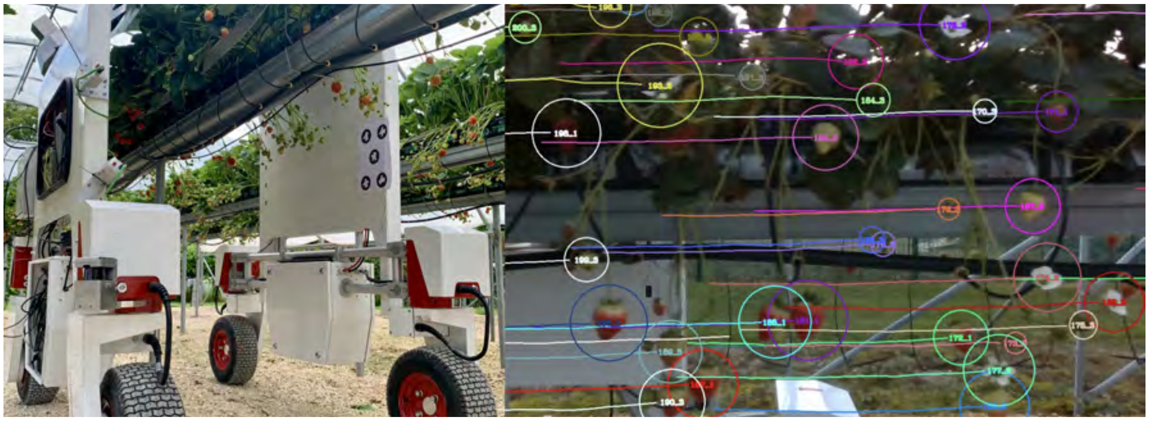


Figure 5.1: Mobile robot tracking platform in the *Katrina-1* Strawberry row (left). Fruit counting-by-tracking (Ripe, Unripe and Flower) visualisation (right). The circles show the tracked fruit identities over time and tracks generated from our proposed method. Strawberry maturity classes are omitted for visualisation purposes, individual instances are in the format *TrackID_ClassID* where 1, 2, 3 are ripe, flower and unripe respectively.

Insight into the capabilities of a vision-based tracking system on a mobile robot is yet to be evaluated for fruit counting. Multi-object trackers in the detect-to-track paradigm have shown promise applied to people tracking, and some of these approaches have been successfully applied to fruit counting (Bellocchio et al., 2019; Santos et al., 2020; Mekhalfi et al., 2020).

Tracking fruit, for the purpose of counting, faces many challenges due to the nature and complexity of farm environments. A tracking algorithm must be able to disambiguate near identical instances of fruit, handle changing appearances, manage varying factors such as illumination or altering-viewpoint, and re-identify after disappearances due to other issues such as occlusion. Examples are depicted in Figure 5.1. Recently proposed solutions (Kirk, Cielniak and Mangan, 2020b; S. W. Chen et al., 2017) leverage deep learning to accurately detect fruit in varying conditions, with newer models subsequently adding detect-to-track based approaches as the counting method (X. Liu et al., 2019; Xu Liu et al., 2018). These approaches generally deal with a single class for each fruit: the utilisation of a fruit maturity stage and a mobile robot to enable more effective tracking is investigated.

A comparison of SOTA trackers (Leal-Taixé, Anton Milan, Schindler et al., 2017) in the MOT challenge attribute the recent rise in tracker performance to the inclusion of stronger affinity and appearance models in tracking architectures. Enabling tracks to be maintained over more complex sequences. This insight is explored in this chapter, extending an appearance model-based tracking framework for improved counting of objects in horticultural environments. A video of the tracking system and the code to run, train and reproduce the experiments can be found at the fruit_tracking repository.

## 5.2 Fruit Tracking System

This section describes our solution to counting rows of fruit in table-top farm environments autonomously from a mobile robotic platform with a mounted colour camera. We are interested in obtaining the total fruit count (flower, unripe, ripe) per row for the purpose of informing farmers decisions on yield and labour. To count fruit, our solution aims to associate bounding boxes between cameras frames, where

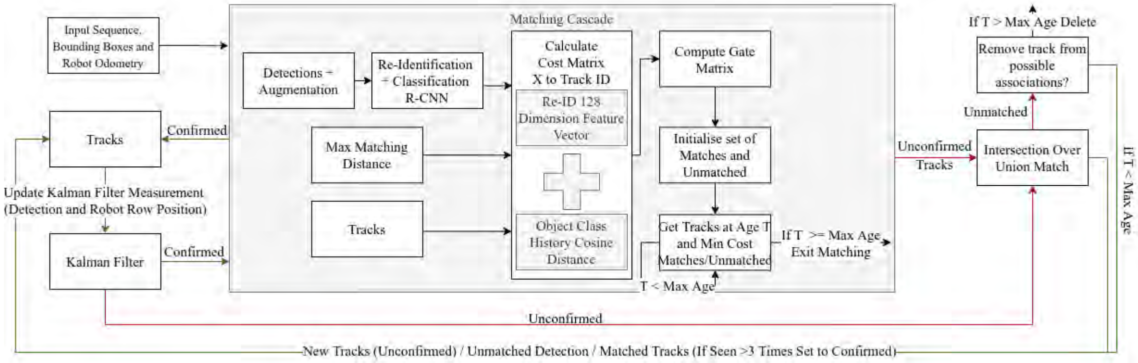

Input Sequence, Bounding Boxes and Robot Odometry
Matching Cascade
Detections + Augmentation
Re-Identification + Classification R-CNN
Calculate Cost Matrix X to Track ID
Re-ID 128 Dimension Feature Vector
Object Class History Cosine Distance
Max Matching Distance
Tracks
Compute Gate Matrix
Initialise set of Matches and Unmatched
Get Tracks at Age T and Min Cost Matches/Unmatched
If T >= Max Age Exit Matching
T < Max Age
Tracks
Confirmed
Update Kalman Filter Measurement (Detection and Robot Row Position)
Kalman Filter
Confirmed
Unconfirmed
New Tracks (Unconfirmed) / Unmatched Detection / Matched Tracks (If Seen >3 Times Set to Confirmed)
Unconfirmed Tracks
Intersection Over Union Match
Unmatched
Remove track from possible associations?
If T > Max Age Delete
If T < Max Age

Figure 5.2: The proposed fruit counting pipeline. The pipeline consists of three main stages, starting with the input of detections in the format of bounding boxes and the corresponding images, generation of re-identification feature vectors and class descriptors and then finally a matching cascade prioritising newer tracks with IoU matching.

the total number of unique associated detection IDs provides the total count. The basis of our approach is inspired by DeepSort (Wojke, Bewley and Paulus, 2017) a tracking framework from the Multi-Object Tracking (MOT) challenge (A. Milan et al., 2016), that has outperformed much more complex architectures on the MOT benchmark. A set of novel components is introduced to bolster the tracking accuracy, (1) the CNN architecture is updated (2) a novel classification branch is added to deal with multi-class data and force more distinct embeddings in the appearance feature space (3) robot odometry data is integrated (position along row) into the Kalman filter state space and (4) the input data is augmented with contextual and aspect preserving qualities. The proposed pipeline is shown in Figure 5.2. The various changes are detailed in the following section, at its core the problem formulation as in (Xu Liu et al., 2018) is for a set of images $\mathbf{I} = (I_k)_{k=1}^{n}$ containing $n$ consecutive frames collected from a moving robotic platform and $c \in \mathbb{N}$ fruit in which the objective is to find the mapping between the true fruit count $c$ and estimated count $\hat{c}$.

### 5.2.1 Tracking

Our formulation for track handling and Kalman filtering is similar to SORT (Bewley et al., 2016) and other MOT benchmarks. No ego-motion information is assumed to be available, but odometry information relating to robot position along a fruit row is provided. Note that the camera is not calibrated. In the method, only standard

Kalman filters are utilised with a constant velocity motion model. The Kalman filter state space is defined as the 10 dimensional vector $(u, v, \gamma, h, r, \dot{u}, \dot{v}, \dot{\gamma}, \dot{h}, \dot{r})$ where $(u, v)$ are the bounding box co-ordinates, $(\gamma, h)$ are the aspect ratio and height and finally $(r)$ is the row position of the robot. Row position $r$ is a unit length from the start of robot operation at the start of the row. Co-ordinates $(u, v, \gamma, h, r)$ are the respective object state and the following are respective velocities in image coordinates.

To accurately solve assignment between newly arrived detections and existing tracks, our approach is formulated into an assignment problem solvable by the Hungarian algorithm. Multiple models are used to represent motion, appearance and class description. Motion is incorporated in the model as squared Mahalanobis distance between predicted Kalman states dictated by $d^{(1)}$ in Equation 5.1, where the $i$-th track distribution projection is denoted in measurement space as $(\mathbf{y}_i, \mathbf{S}_i)$ and the $j$-th new observation $\mathbf{d}_j$. Using this metric with a 10-dimensional measurement state space allows us to easily filter highly improbable associations with the 95% quartile of the Chi-square distribution $t^{(1)} = 11.07$ with 10 degrees of freedom, metrics are admissible if they are within this threshold by the indicator from Equation 5.2 where $x = 1$.

$$d^{(1)}(i, j) = (\mathbf{d}_j - \mathbf{y_i})^T \mathbf{S}_i^{-1} (\mathbf{d}_j - \mathbf{y_i}) \tag{5.1}$$

$$z_{i,j}^{(x)} = \mathbf{1}[d^{(x)}(i, j) \leq t^{(x)}] \tag{5.2}$$

Mahalanobis' distance for multi-object data association might fail in situations while tracking on a mobile robot, such as when small angular/Cartesian movements of the camera are introduced. This can result in large shift in position in image space. Due to this, Two more models are incorporated for improving the assignment problem. From the baseline implementation in, (Wojke, Bewley and Paulus, 2017) the appearance descriptor shown in Equation 5.3 is used and extended in Equation 5.4 where the smallest cosine distance between $i$-th tracks and $j$-th bounding boxes is measured. For the appearance descriptor, $r_j$ a gallery $\mathbf{R}_k = \{\mathbf{r}_k^{(i)}\}_k^{O_k}$ is kept of the last $O_k = 100$ observations for each track $k$. The model is extended by adding a new class description metric that considers the probability of different maturity stages. For a given observation with constant motion, occlusion can alter the intra-class appearance (i.e., covering the red part of a Strawberry may make it appear as unripe instead of ripe), to ensure consistent re-identification a class description is computed of all previous class observations.

$$d^{(2)}(i,j) = min\{1 - \mathbf{r}_j^T \mathbf{r}_k^i \mid \mathbf{r}_k^i \in \mathbf{R}_i\} \tag{5.3}$$

The class description metric $w_j$ is computed from the map of label probabilities of the classification head projected onto the unit hyper-sphere of previous class observations where $||\mathbf{w}_j|| = 1$. Similarly to computing the appearance metric, a gallery $\mathbf{W}_k = \{\mathbf{w}_k^{(i)}\}_k^{O_k}$ is utilised of previous observations of each track $k$. Trivially inconsistent tracks before weighting with similar appearance but differing classes, the metric is $\mathbf{w}_j + \mathbf{r}_j \leq \mathbf{r}_j + 1$. Equation 5.4 denotes the cosine distance between label probabilities and previous label probability observations.

$$d^{(3)}(i,j) = min\{1 - \mathbf{w}_j^T \mathbf{w}_k^i \mid \mathbf{w}_k^i \in \mathbf{W}_i\} \tag{5.4}$$

Combining the metrics ensures the association is more robust, taking into account motion, appearance and class description. As with squared Mahalanobis distance, a binary variable is introduced to discount improbable associations for both appearance $t^{(2)}$ and class description $t^{(3)}$ in Equation 5.2. The association problem cost matrix $m$ is the combination of all the metrics with a weight parameter $\lambda$ where $x = 3$ and the association is admissible if it is within the gating region in Equation 5.6.

$$m_{i,j} = \lambda d^{(1)}(x-1) + \sum_{p=2}^{x} (1-\lambda) d^{(p)}(i,j) \tag{5.5}$$

$$z_{i,j} = \prod_{p=1}^{3} b^{(p)} \tag{5.6}$$

Suitable thresholds $t^{(x)}$ for each gating function $z_{i,j}^{(x)}$ can be found. For the motion metric chi-square distribution threshold will remove unlikely associations, for the appearance metric values of cosine similarity of similar bounding boxes from training data can be computed to find a suitable threshold and finally for the class description metric values $t^{(3)} \leq 1$ is a suitable value where closer to zero forces more consistent label probability history. The weight variable $\lambda$ can be optimised depending on the domain, small values will prioritise appearance and class description over motion and higher values close to 1 will use mostly motion information to calculate the association and gating cost. The same matching cascade is used as in (Wojke, Bewley and Paulus, 2017) extended with the new cost matrix $m$ defined in Equation 5.5.

### 5.2.2 Re-Identification and Class Description Network

To discriminate between different identities, a residual neural network is trained to generate feature vectors $\mathbf{r}_j$ and $\mathbf{w}_j$ that minimise the cosine similarity between $j$-th bounding boxes of the same instance. The feature extraction stem architecture is described in Figure 5.3. To train the network, two classification heads are added. The re-identification classification head attempts to map the identities vector $\mathbf{r_j}$ to the ground truth instance ID. The objective loss is Cross Entropy Loss (named Cosine Soft Max in the Wojke, Bewley and Paulus (2017) paper). The label classification head maps $\mathbf{r_j}$ to label probabilities $\mathbf{w_j}$, Cross Entropy Loss is used to minimise the most probable class to the ground truth maturity stage (flower, unripe, ripe). The training procedure attempts to minimise both losses. For a batch size of 128 one forward pass of the network takes 412$\mu$s per bounding box on a modern GPU (Nvidia GeForce GTX 3090) making the method suitable for online tracking, with a max capability of $>$ 2400 bounding boxes per second.

To deal with multi-class tracking as well as multi-object, the label classification head is added in order to coerce more separable features depending on the fruit maturity stage. Figure 5.4 visualises the Principal Component Analysis (PCA) of learned feature vectors from a trained network with the baseline (bottom) and with our network (top) this extra label classification step. The 128 dimension feature vectors are visualised as two principal components and the colour is the x, y components

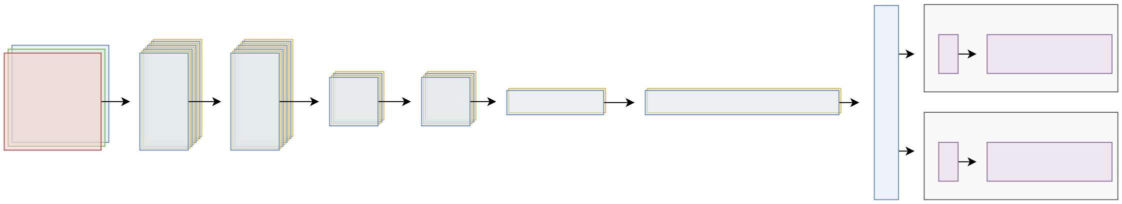
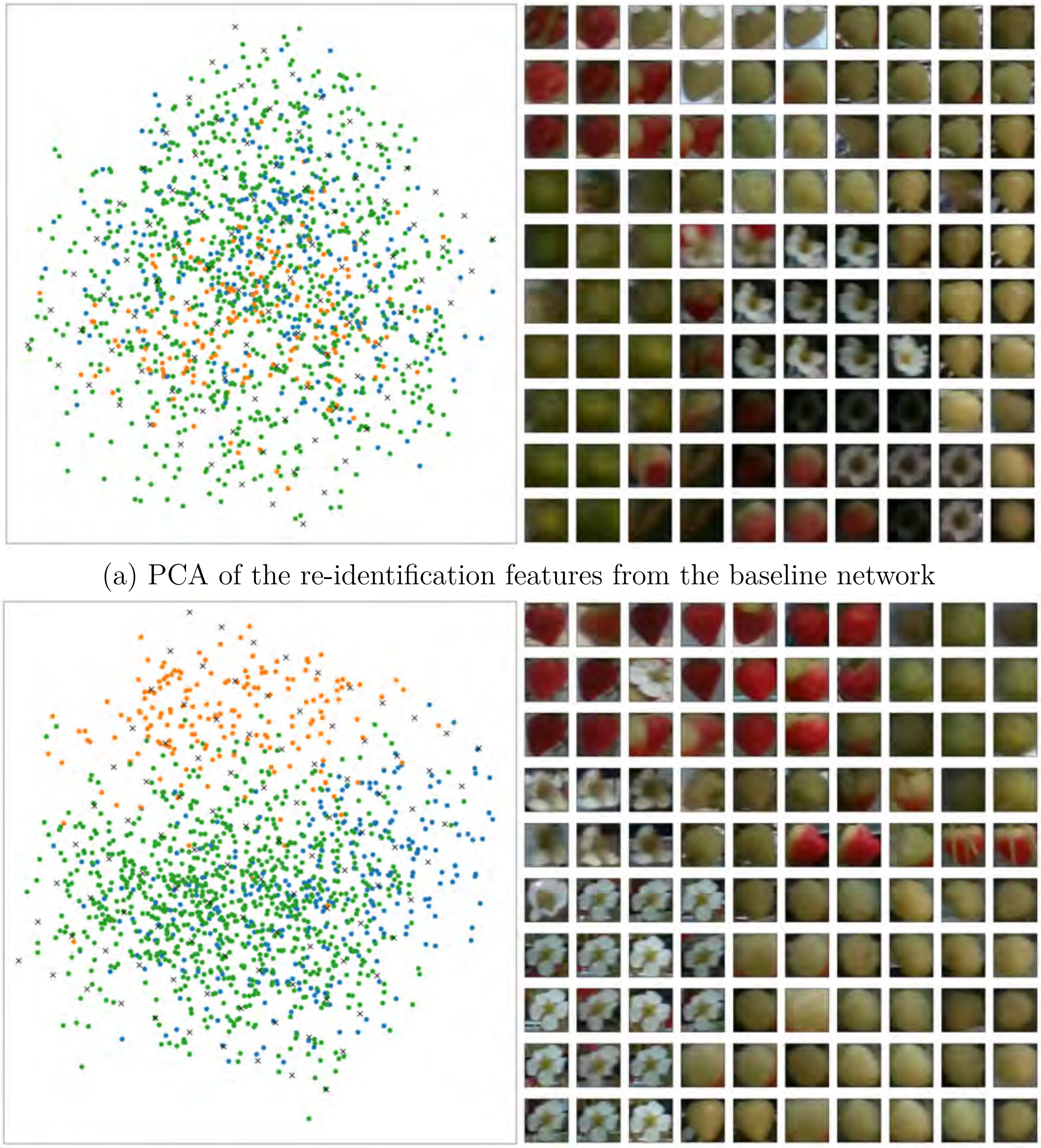


Figure 5.3: Residual Neural Network Re-ID feature extractor. Input is a 64x64 patch of a bounding box detection and output is a 128 length feature vector projected onto the unit hyper sphere (for use with cosine similarity metric $d^{(2)}$). Each convolutional block (1, 2) is a 2D convolution followed by 2D batch normalisation and ReLU activation function. Each residual block (4, 5, 6, 7, 8, 9) is a basic ResNet block (K. He, X. Zhang et al., 2015). There are two heads to coerce more separable feature embeddings while training, a re-identification head and maturity classifier head. Total trainable parameters are 825,152.

with the label class. It can be seen adding the extra network head creates much more separable features. To visualise the component space a grid plot of 100 points around the extrema regions of principal components is made, and the PCA inverse transform of $u$, $v$ components is taken to get a representative feature vector. From the representative features a k-d tree of the original feature vectors is built and take the smallest cosine distance to the representative feature vectors to visualise an example bounding box (shown on the right) from the low dimensional u, v components. Much tighter and more logical groupings are observed and the weight of different labels is more uniform, creating a more stable cosine distance between classes.

### 5.2.3 Tracking Sequences

Image sequences of commercial strawberry plants were collected at the University of Lincoln research farm at Riseholme, UK from a RGBD camera (Intel RealSense D435i) mounted on the agricultural robot Thorvald (Grimstad and P. J. From, 2017) (see Figure 5.1). The robot was deployed in two 8x24m poly-tunnels containing 5 table-top rows separated by a distance of 1.5m. The central row of each tunnel was used to capture the *Driscoll Amesti* and *Driscoll Katrina* data sequences. Sequences were collected one day apart late in the season (September) with the camera height aligned to the strawberry soil bags. The data was acquired at 7Hz with the robot traversing the rows at 0.1m/s (*Amesti-1, Katrina-1, Katrina-2*) and 0.2m/s (*Amesti-2*) at a



(a) PCA of the re-identification features from the baseline network

(b) PCA of the re-identification features from our improved network

Figure 5.4: PCA (left) and inverse mapping of 10 by 10 grid points (right, denoted by black crosses). Best viewed online. Analysis performed on the re-identification network feature vector of (a) baseline network and (b) improved network with classification head. The PCA (left) shows us the classification network (b) has better separation between classes (green=flower, blue=unripe, red=ripe) when the inverse of the PCA function is applied to the X and Y positions of each grid point.

resolution of 1920 × 1080. Data was annotated manually with expert knowledge into 3 distinct classes: Ripe (> 85% red coverage), Flower (white petals with no calyx shown) and Unripe (small to large immature green calyx visible). Each of the sequences captured was stopped at 500 frames. These images, shown in Table 5.1, were captured using the RaymondKirk/topic_store and IntelRealSense/realsense-ros packages hosted on GitHub.

Table 5.1: Tracking Image Sequence Summaries

| **Variety** | **Row** | **Capture Hz** | **Robot m/s** | **Instances** | **Count** |
|---|---|---|---|---|---|
| Amesti | 1 | 6.99 | 0.1 | 12219 | 233 |
| Amesti | 2 | 7.38 | 0.2 | 4895 | 172 |
| Katrina | 1 | 7.14 | 0.1 | 15850 | 299 |
| Katrina | 2 | 6.94 | 0.1 | 14507 | 326 |

Four sequences were collected to validate our approach, containing two sides of table-top poly tunnel grown Strawberries of Amesti and Katrina variety. The sequences are referred to in the format of *Variety-Side* where 1 and 2 are left and right sides of each row respectively. Amesti1-2 and Katrina 1-2 contain 12219, 4895, 15850 and 14507 bounding box annotations with 233, 172, 299 and 326 tracklets (count) respectively. To evaluate our proposed method, a *training* and *testing* split of a 75% to 25% ratio is defined, to ensure no bias in the evaluation of the system and promote generalisation when tracking. The training split consists of the images from *Katerina-1*, *Katerina-2* and *Amesti-2* sequences, whereas the testing split consists of *Amesti-1* image sequences. The training set was then split again by 75% and 25% to serve as training and validation data for optimising the re-identification model. The splits were chosen this way to ensure no bias in the final testing set used for evaluation. Some experiments contain data augmentation; two different types are applied which are deemed square and padding, these transformations given in Equation 5.7 and 5.8 of the raw bounding boxes are to preserve aspect ratio of textual features and embed surrounding environment context respectively. The data is shared to support bench-marking in the community, since no other openly available sequences were found.

$$s(x,y,w,h) = \begin{cases} (x, (y+\frac{h}{2}) - \frac{w}{2}, w, w) & w > h \\ ((x+\frac{w}{2}) - \frac{h}{2}, y, h, h) & w \leq h \end{cases} \tag{5.7}$$

$$pad(x,y,w,h,p) = (x-p, y-p, w+2p, h+2p) \tag{5.8}$$

### 5.2.4 Evaluation Metrics

Let $f : \mathbf{I}^{(j)} \rightarrow c \in \mathbb{N}$ for a sequence, $\{\mathbf{I}^{(j)}\}_{j=1}^{500}$ our proposed methods are evaluated as the Least Absolute Deviation ($L1$ loss) of $c^{(j)}$ and $\hat{c}^{(j)}$.

$$\sum_{j=1}^{500} |f(\mathbf{I}^{(j)}) - \hat{c}^{(j)}| \tag{5.9}$$

For specific classes, only $c$, $\hat{c}$ fruit are considered as belonging to the specific class and others are ignored. This loss is used in model selection and to validate the proposed system. Specifically, Equation 5.9 concerns the testing set *Amesti-1* for evaluation.

## 5.3 Results and Discussion

This section evaluates our counting-by-tracking pipeline. By first detailing the training regime for each experiment. The baseline system (Wojke, Bewley and Paulus, 2017) is then compared to seven other experiments with the following modifications: (1) addition of a label probability cosine cost $d^{(3)}$ and label classification head to the re-identification network (2) detection augmentations *square* in Equation 5.7 and *pad* in Equation 5.8 and (3) a combined improvements network consisting of the all modifications that resulted in an improved score shown in Figure 5.5. All evaluation in this section is applied to the hand labelled *Amesti-1* data sequence which was hidden during network training against the predicted count data per-frame and total per-sequence.

To train the models an input batch of 128 bounding boxes per iteration for a total of 6400 iterations was run, reducing the base LR of 0.1 by a factor of 10 at 80%

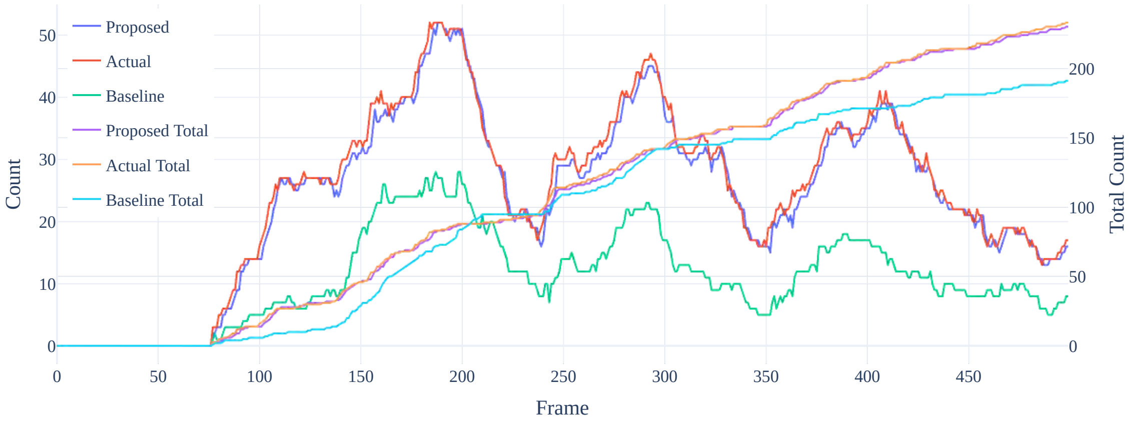


Figure 5.5: Counting-by-tracking performance of the baseline approach and our proposed combined method, detailed in Table 5.2. Total counts over time are given, and also frame by frame counting results for all classes combined (class agnostic tracking). It can be seen our proposed system (dark blue, purple) performs much better and achieves almost perfect frame-by-frame counting. Whereas the baseline system (light blue, green) has a much lower overall count accuracy.

and 90% of total iterations. To ensure consistent results, the data augmentation was applied to the data directly before training and output the re-identification accuracy and label classification accuracy (relevant only for the Sub-Net experiment) every 400 iterations. The re-identification accuracy during training for the baseline model and Sub-Net model were 97% and for all other experiments the accuracy was >99% when rounded to two significant figures are shown in Figure 5.6.

Table 5.2 describes the results of the experiments, presented per-class and class agnostic since intrinsically the baseline tracking system is class agnostic. The results show a strong improvement over the baseline experiment across all experiments. Bold values of $L1$ loss indicate the best experiment for each evaluation metric.

## 5.4 Conclusions

A framework has been presented for accurately tracking and counting fruit in a complex scene from bounding boxes, extending on current tracking architecture. Our system utilises re-identification features and label probability vectors with cosine similarity as well as robot odometry formatted as row position and data augmentation methods to maintain track consistency through occlusions and to maintain tracks in

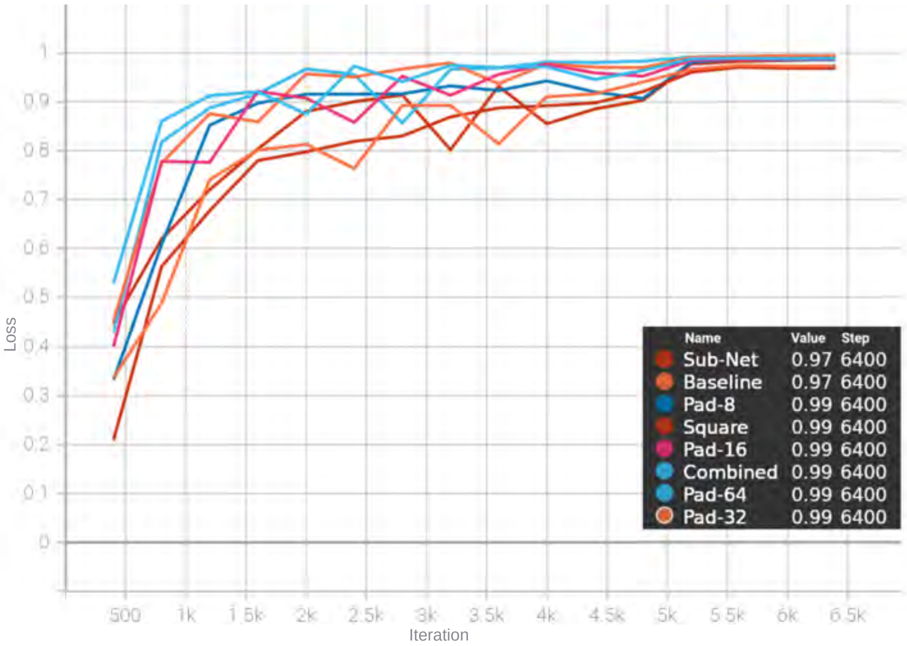


Figure 5.6: Re-Identification accuracy of the trained networks.

Table 5.2: $L1$ Loss of Berry Count on Amesti-1

| | Class | | | |
|---|---|---|---|---|
| **Experiment** | **Agnostic** | **Flower** | **Ripe** | **Unripe** |
| Baseline | 41 (192/233) | **1 (23/24)** | 19 (69/50) | 59 (100/159) |
| | Maintaining Aspect Ratio | | | |
| Square | 57 (176/233) | 2 (26/24) | 3 (53/50) | 62 (97/159) |
| | Classification Network Cost Matrix | | | |
| Sub-Net | 31 (202/233) | **1 (25/24)** | 3 (53/50) | 35 (124/159) |
| | Embedding Context in Detection | | | |
| Pad-8 | 33 (266/233) | 14 (38/24) | 29 (79/50) | 10 (149/159) |
| Pad-16 | 86 (319/233) | 13 (37/24) | 36 (86/50) | 37 (196/159) |
| Pad-32 | 25 (258/233) | 3 (27/24) | 12 (62/50) | 10 (169/159) |
| Pad-64 | 8 (241/233) | 3 (27/24) | 4 (54/50) | **1 (160/159)** |
| | Combined Improvements | | | |
| Combined | **7 (240/233)** | 2 (26/24) | **2 (52/50)** | 3 (162/159) |

densely clustered detection regions. The visual results of our method can be seen in Figure 5.7.

The results demonstrate that our system is capable of reliably tracking and counting multiple classes in clusters from multiple view points. An off the shelf cheap computer vision camera (Intel Realsense) and modern GPUs were used so that our system can

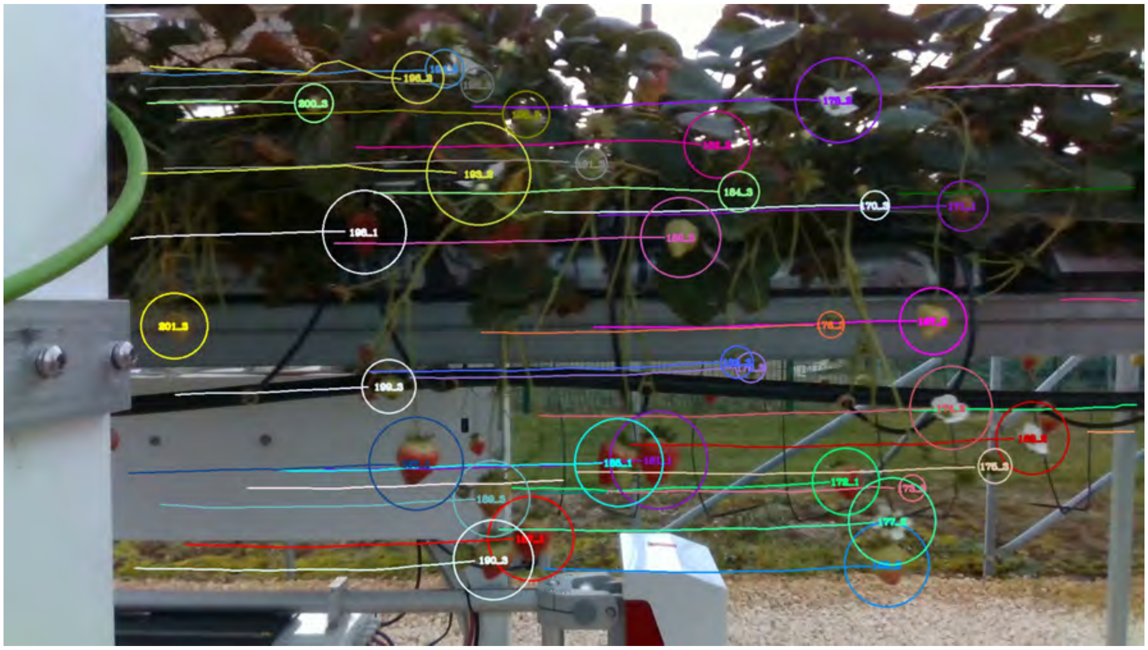


Figure 5.7: Tracking results on the Amesti testing data

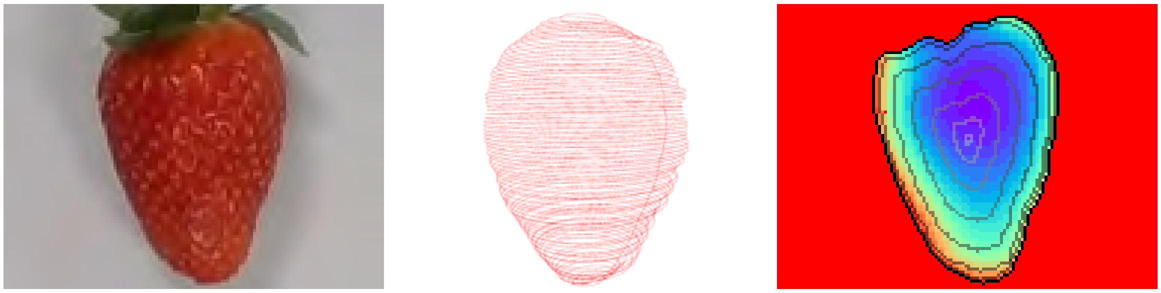
be applied easily. Our results indicate an improvement of $L1$ loss from 41, 1, 19 and 59 (all classes, flower, ripe, unripe) to 7, 2, 2, 3 error in counting 500 frames of the *Amesti-1* sequence not used in training. On a modern GPU is are able to process, >2400 bounding box detections per-second in a single forward pass of the network, enabling online tracking applications.

# Chapter 6

# Soft Fruit Trait Extraction

This chapter introduces approaches to maximise the value of detection and tracking within horticulture to extract phenotypic traits from object detections and tracks. Parts of which were published in *Non-destructive Soft Fruit Mass and Volume Estimation for Phenotyping in Horticulture* Kirk, Cielniak and Mangan (2021a) Destructive phenotyping (measurement of properties from a genotype) is an expensive rarity within the industry due to the time and margin constraints of fruit growers during the season, however the data provided generates critical insights for crop management and breeding policies. In this work, phenotyping relates to the estimation of fruit traits from image data. Work is presented to transform image data with bounding box or segmentation based detections to volume, size, and weight estimations in real-time. Enabling the collection and analysis of millions of samples quickly. This chapter presents (1) three novel approaches to estimate phenotypic traits width, height, cross-section length, volume, and mass from only image segmentations and depth information of strawberries, (2) a thorough evaluation of the proposed methods in lab conditions against GT data, and, (3) application and validation of the proposed methods in-field from a robotic platform.

Manual assessment of soft-fruits is both laborious and prone to human error. Methods are presented to compute phenotypic traits width, height, cross-section length, volume and mass using computer vision cameras from a robotic platform, shown in Figure 6.1. Estimation of phenotypic traits from a camera system on a mobile robot is a non-destructive and non-invasive approach to gathering qualitative fruit data which is critical for breeding programmes, in-field quality assessment, maturity estimation and yield forecasting. Our presented methods can process 324-1770 berries per second on consumer grade hardware and achieve low error rates of 3.00 $\text{cm}^3$ (13% error compared to the median strawberry volume of 28.5$\text{cm}^3$) and 2.34g for volume and mass estimates. Our methods require object masks from 2D images, a typical output of segmentation architectures such as Mask R-CNN, and optionally depth data for computing scale. There are many agricultural applications that would benefit from robotic monitoring of soft fruit, examples include harvesting and yield forecasting. The feasibility of using vision based modalities for precise, cheap, and real time computation of phenotypic traits: mass and volume of strawberries from planar RGB slices and optionally point data is investigated. Soft fruit detection from RGB and RGB-D data is becoming increasingly prevalent in the horticulture industry. Concerns of market demand, farm efficiency and a growing population are pushing the industry to find new ways to increase yield per hectare while consuming fewer resources such as fuel, electricity, chemical treatments and labour. It is critical for vision based fruit detection methods to estimate traits such as size, mass and volume for quality assessment, maturity estimation and yield forecasting. Our best method achieves a marginal error of 3.00 $\text{cm}^3$ for volume estimation. The planar RGB slices can be computed manually or by using common object detection methods such as Mask R-CNN.



(**a**) Actual = 35.00 $\text{cm}^3$ (**b**) 34.11 $\text{cm}^3$ (**c**) 34.53 $\text{cm}^3$

Figure 6.1: Volume predictions of Strawberry in image (Figure 6.1a) via methods deemed disc summation (Figure 6.1b) and surface area integration (Figure 6.1c) described in Section Section 6.1

Image-based fruit recognition is an area fast gaining interest in the horticultural industry. The environmental challenges posed by the fast-growing population and climate concerns are spurring new innovative approaches to fruit detection, harvesting and yield estimation using computer vision, e.g (Kirk, Cielniak and Mangan, 2020c; Xiong et al., 2019; Y. Chen, W. S. Lee et al., 2019). Phenotypic information about the fruit is important for all of these approaches and is crucial for effective breeding programmes. For harvesting, it allows to automatically grade and harvest specific type of berries, for yield more specific estimates such as detection of waste strawberries or estimating a total yield volume can be computed and for quality more accurate assessments can be made.

The methods for estimating phenotypic traits are presented using models derived in laboratory conditions and validated in in-field conditions. These traits are estimated from images based on the intuition that most fruits and berries such as kiwi, strawberries and grapes are ellipsoidal in nature and symmetrical around their major-axis, meaning the methods presented are applicable to most of the soft-fruit family. Geometrically, the major axis is the longer axis of an ellipse passing through its foci or centre of gravity in the case of our planar segment; the minor axis is the shorter axis directly perpendicular to the major. Our methodologies and findings in this chapter are restricted to strawberries, they are one of the more difficult crops in the soft-fruit family that have an ellipsoidal shape to phenotype due to high variation in shape from different viewpoints and maturity levels.

Traditionally, fruit phenotyping requires a human agent to manually derive fruit quality attributes, which commercially is not viable due to an already increasing labour demand and the subjectivity between agents in different lighting and environmental conditions. Robotic monitoring platforms are a promising solution to automating these processes and removing human error. Our methods are applied to data captured in-field from a mobile robotic platform, and analyse the suitability for use as an online phenotyping tool over multiple maturity stages. Current large scale phenotyping techniques are generally restricted to in-lab conditions (J. He, Harrison and Li, 2017; Pound et al., 2016), restricting the collection of large quantities of statistical data necessary for commercial application and in-field use where the relationship between plants and fruit is more accurately modelled.

Recent data driven approaches have shown automation of the phenotyping process using computer vision to be a powerful tool for many of the challenges faced in horticulture (Pound et al., 2016). Automating the phenotyping process enables applications that can obtain large amounts of data (high-bandwidth phenotyping) and high-fidelity from real environments. Robotic monitoring platforms allow us to do apply these processes over large areas in a shorter amount of time, concurrently to other tasks such as harvesting and fruit counting. Our methods can be applied using cheap off the shelf components that are widely available, since the minimum requirements are only colour imagery obtained from consumer cameras. These methods with further work can be used as a cost-effective phenotyping technique for use at commercial scale. The contributions of our research are detailed below:

1. Three novel approaches to estimate phenotypic traits width, height, cross-section length, volume and mass from only image segmentations and optionally depth information of strawberries are presented.

2. A thorough evaluation of the proposed methods in lab conditions against GT data.

3. Application and validation of the proposed methods in-field from a robotic platform.

This chapter is organised as follows: the proposed methodologies for quality trait estimation are presented in Section 6.1. Section 6.2 then follows, detailing the experiments performed for validation and prediction of the phenotypic traits. A suitability study is presented in this section in the form of an analysis of in-field application, which is used to determine the applicability of our methods from mobile robotic platforms. It also quantifiably details the evaluation of the proposed techniques. Section 6.3 then summarises the work and discusses future improvements.

## 6.1 Fruit Trait Extraction System

This section introduces the methods used to extract phenotypic information from 2D binary segmentations. A segmentation is a binary mask detailing all of the pixels that belong to an object in an image. These segmentations are obtained from GT data, where each strawberry pixel was labelled. From these segmentations it is trivial to compute the width, height and cross-section length of the strawberries by computing the minor and major axis of each segment. The cross-section length is equivalent to the minor axis length for most soft-fruits, so this value is used as standard, when depth information is available the cross-section length is computed as two times the difference in min and max of the depth values contained in each segmentation instead.

The motivation of calculating phenotypic traits this way is that the computational resources required to process these segmentations are very low and are a typical output of modern object detectors in this field, meaning this approach is easily integrated with existing work with negligible overhead. The computation statistics are later presented in Table 6.2. Each of the volumetric estimator methods will use the minor, major and cross length estimates of the segment, so the assumption is made that for most soft-fruits the surface bounded by each segmentation is symmetric, as the hidden surface is estimated to be the same volume as the visible surface.

### 6.1.1 Volume Estimation

This section presents three methods to extract the volume of a segment, the three evaluated methods are ellipsoidal, surface area integration and disc summation. The ellipsoidal method for brevity, trivially computes the volume as $\frac{4}{3}\pi m_i m_a d$ where $m_i$ is the minor axis, $m_a$ is the major and $d$ is the cross-section lengths. These measurements are computed from both the segmentation data and optionally measurements extracted from the depth map.

When depth information is available, the scale of the estimates can also be computed. The presented methods approximate the volume in pixels (px$^3$). To calculate the volume in centimetres (cm$^3$) the segmentation contour $c$ can simply be deprojected by the camera intrinsic parameters focal length $f_x, f_y$, principal point $p_x, p_y$ and an estimated distance $z_{max}$ from the camera obtained from the max value bounded by the segment. For the disc method, the $z_{max}$ value is equal to the local max at each row rather than the entire segment. The deprojection step is shown in Equation 6.1

and is applied prior to volume estimation. Method evaluation is given throughout in median absolute error in cubic centimetres (cm$^3$) and grams (g).

$$c'_x = \frac{z_{max}}{f_x}(c_x - p_x) \qquad c'_y = \frac{z_{max}}{f_y}(c_y - p_y) \tag{6.1}$$

### Surface Area Integration

The surface area integration method uses the relationship between surface area and volume. For an ellipsoid, the volume is the integral of the surface area with respect to the radius. In our case the radius is known to be the cross-section length, however soft-fruits are not perfectly spherical or ellipsoidal and have deformities around the contour, strawberries in particular have a more teardrop profile. To account for this, it is necessary to instead calculate the surface area of the actual contour of each segmentation rather than the bounding ellipse to compute a more accurate volume estimate. The centre of mass is also not guaranteed to be half of the dimensions of each segmentation either, as with a perfect ellipse, so it is also necessary to consider the scaled contour around our segmentation centre of mass when scaling the contour with respect to the cross-section length. For a contour, $c$ it can be scaled around its centre of mass with respect to the cross-section length $r$ by applying the function $f(c,r)$ as shown in Equation 6.2.

$$f(c,r) = \frac{r}{m_i}\left(\frac{\sum_{k=1}^{n}(c_{xk}, c_{yk})}{n} - c\right) + c \tag{6.2}$$

To calculate the surface area of each scaled contour, $s$ the shoelace algorithm $a(s)$ can be used for finding the area of a simple polygon, with no intersection or holes, expressed as Cartesian coordinates of a segmentation as shown in Equation 6.3. To use this method, first, it is necessary to order the points in the scaled contour counter-clock wise. The surface area of the segmentation could also be expressed as the sum of all the binary pixels, however the shoelace method is more generalisable when also computing scaled volumetric estimates using depth information.

$$a(s) = \frac{1}{2} \left| \sum_{i=1}^{n-1} s_{xi} s_{yi+1} + s_{xn} s_{y1} - \sum_{i=1}^{n-1} s_{xi+1} s_{yi} - s_{x1} s_{yn} \right| \tag{6.3}$$

To compute the volume estimate $V$ of the segment, the integral of the scaled contours surface areas with respect to the estimated cross-section length can be computed. The integral for computing the volume $V$ of an irregular segmentation is shown in Equation 6.4 by taking the product of $\mathrm{d}x$, the height of each slice and the contour $c$, which is scaled by each slice radius $r$ in function $f(c, r)$ Equation 6.2 and calculating its surface area $a(f(c, r))$. The integral range $[0, r]$ is used, and the result is multiplied by 2 to only consider positive scaling of the initial contour values.

$$V = 2 \int_0^r 2a(f(c, r))\mathrm{d}x \tag{6.4}$$

### Disc Summation

The disc summation method estimates the volume of the segmentation by treating each row of the contour $c$ of size $d_y$ as a cylinder. Where each cylinder height is $d_y$, the unit distance between each row, and radius is half the row width. When depth information is available, the row can also be treated as a cylinder with an elliptical cross-section, since the cross-section length can be different to the minor axis length for each row. The volume of each cylindrical row can now be computed as $\pi r^2 d_y$. This method should be more robust than the integration step in cases when the orientation estimate error is large or when the volume of the object's hidden surface is very different to the volume of the visible surface. Since each row is treated independently, a more complete surface not dependent on axial symmetry can be reconstructed, whereas with integration the entire contour is used with a singular estimate of the cross length. The method for computing the volume $V$ from a contour $c$ is shown in Equation 6.5.

$$v_i = \sum_{j=1}^{n} c_{ij} \qquad v = \left( \pi \frac{v_i^2}{4} d_y \right)_{i=1}^{n} \qquad V = \sum_{k=1}^{n} v_k \tag{6.5}$$

### 6.1.2 Mass Estimation

To calculate the mass of an object $m$ from a volume estimate, $V$ one can take the product of volume and density. The average density $p_{avg}$ of the all GT samples collected was 0.858 g/cm$^3$, therefore the estimated mass is $m = p_{avg}V$. Our density measure only considers mature Beltran berries, however in-field this measurement is dependent on many varying factors such as water content, environmental conditions, growth stage and variety.

## 6.2 Results and Discussion

The following section introduces the dataset that was collected, an evaluation of the methodologies presented above, the phenotyping metrics, and validation of our methods applied to real in-field data.

### 6.2.1 Data Collection

In order to evaluate our methods, mass and volumetric data of soft fruit was required. It was chosen to evaluate strawberries as they are readily available and have one of the most challenging shapes in the soft fruit family compared to blackberries, blueberries etc. their surface is not as ellipsoidal and has a more teardrop profile. In total 20 samples were collected of class 1 ripe strawberries shown in Figure 6.2b. Ideally, more samples would be included, given class-1 strawberries are very similar in appearance and shape, the results should be significant against a population of class-1 strawberries, further data should be collected to validate for other classes of strawberries. The strawberries that were collected were of the Beltran (Fragaria Hybrid) variety purchased from local supermarkets a couple of hours before the data collection. Each berry was stored in $3°C$ until measurements were taken to ensure the best quality and to minimise potential deformations.

To capture the data necessary a 2 cm$^3$ precision volumetric beaker was used, a 5g confidence scale accurate to 1g, a 0.01 mm accurate digital caliper and an Intel Realsense D415 computer vision camera to capture RGB images and depth information,

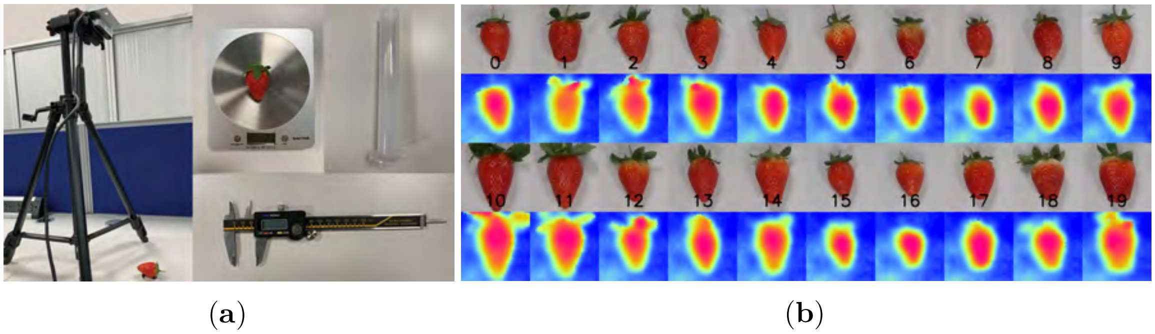


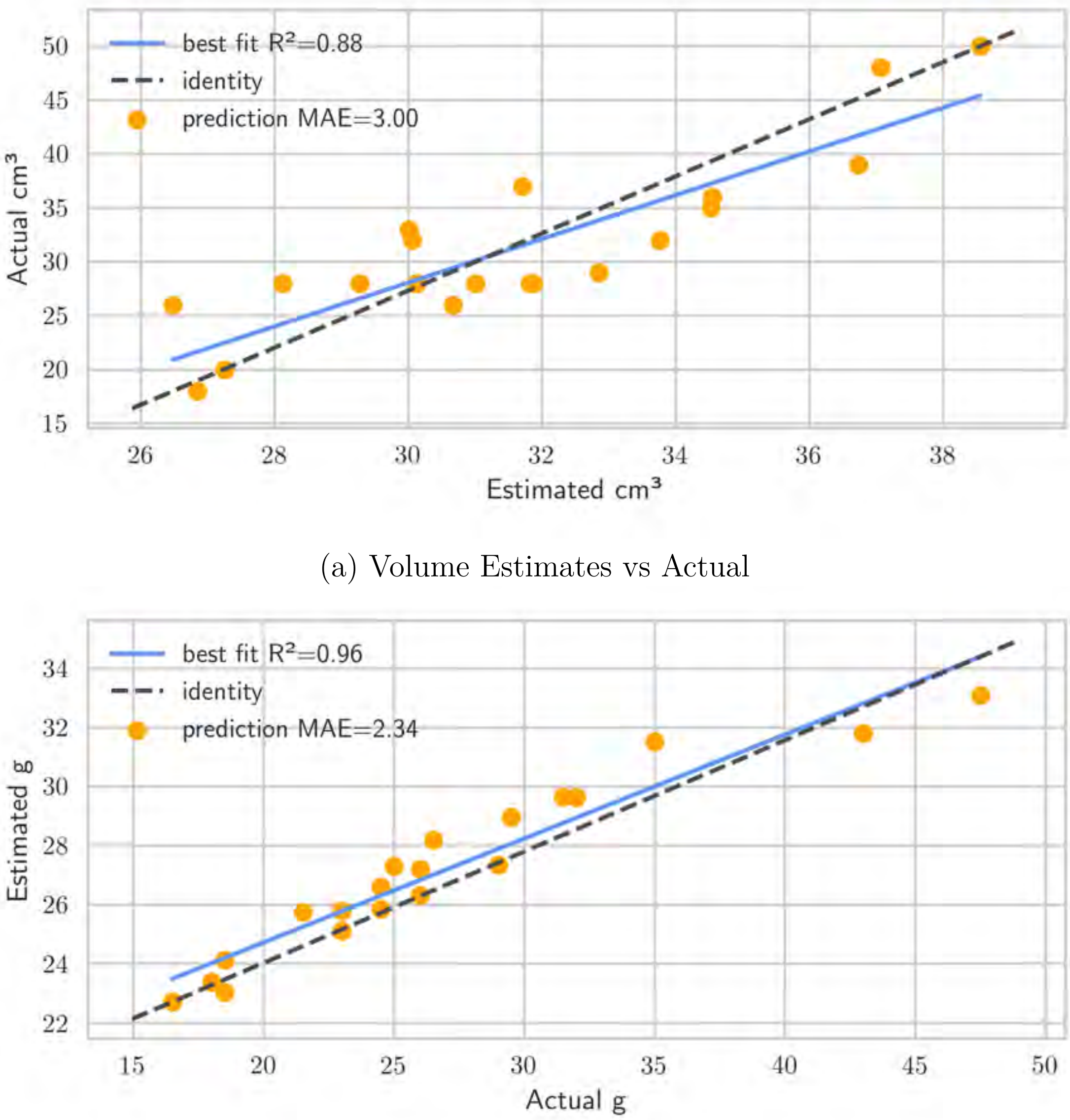
Figure 6.2: Data collection equipment Figure 6.2a and colour/depth images captured Figure 6.2b

pictured in Figure 6.2a. Each strawberry was measured in three dimensions manually through its minor, major and cross-sections which are the widest, tallest and deepest lengths of the berry respectively. Then it was weighed and placed in the volumetric beaker containing $20°C$ water and a control rod of a known volume was used to fully submerge the berry to get more accurate readings. Finally, the berry was placed at a set distance away from the downwards facing camera, flat on a table to simulate the conditions met in-field (hanging from the stem, captured side on) and the colour and depth information was captured and logged.

### 6.2.2 Phenotypic Trait Predictions

Our methods were evaluated by comparing our predicted volume and mass estimates against our GT data and take the median absolute error $V_{err} = median(|\hat{V}_1 - V_1|, ..., |\hat{V}_n - V_n|)$ and $m_{err} = median(|\hat{m}_1 - m_1|, ..., |\hat{m}_n - m_n|)$, where a value of 0 would indicate a perfect estimate. Median absolute error is given in cubic centimetres (cm$^3$) and grams (g), for example an error of 3.375cm$^3$ for a volume estimate is equivalent to the volume of a cube with side lengths of 1.5cm. Additionally, to measure agreement and correlation between the values that were used and the coefficient of determination (R$^2$). Figure 6.3 and Figure 6.3 show the estimated values against the actual GT.

Table 6.1 presents the statistical analysis of the methods. In this case *GT* denotes that the method used measurements for minor, major and cross lengths from GT



(a) Volume Estimates vs Actual

(b) Mass Estimates vs Actual

Figure 6.3: Surface area integration estimates agreement against GT data

data instead of segmentation or depth data to illustrate where possible error in reconstruction of the volume that could have occurred.

Table 6.1: Median Absolute Error of volume and mass estimation methods, bold indicates the best method

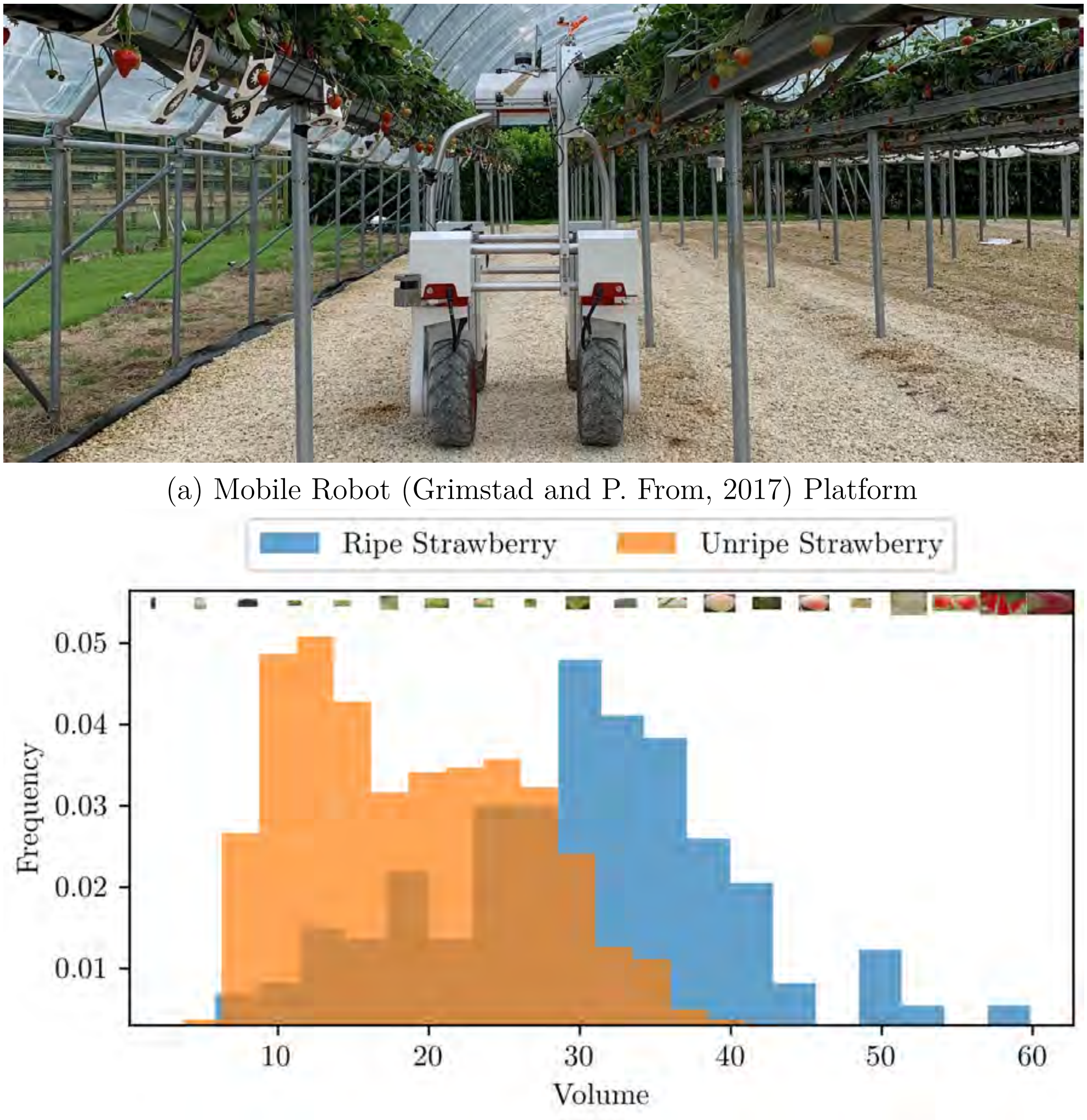
| **Trait** | **Ellipsoid** |  | **Integration** |  | **Disc Summation** |  |
|---|---|---|---|---|---|---|
|  | ***GT*** | ***Depth*** | ***GT*** | ***Depth*** | ***GT*** | ***Depth*** |
| **Volume** | 3.28 $\text{cm}^3$ | 3.94 $\text{cm}^3$ | 3.11 $\text{cm}^3$ | **3.00** $\mathbf{cm}^3$ | 3.75 $\text{cm}^3$ | 3.20 $\text{cm}^3$ |
| **Mass** | 2.62 g | 4.34 g | 2.46 g | **2.34 g** | 2.99 g | 2.39 g |

### 6.2.3 In-Field Experiments

The application of these phenotyping methods will be best used on a robotic platform for online trait estimation while performing tasks such as: harvesting, yield estimation, quality assessment, automated weighing and disease detection. To illustrate the effectiveness of the approach, image data was collected at a local strawberry farm from a mobile agricultural robot Thorvald (Grimstad and P. From, 2017), pictured in Figure 6.4. The RGBD camera was placed 30-50 cm from the Strawberry row and 80 images were captured containing 1250 examples of ripe and unripe berries of varying sizes and shapes. The strawberry varieties captured was Amesti.

Figure 6.5 shows volume and mass predictions made on the data collected in-field. It is evident from this initial tests that this method can be used successfully in-field to estimate these attributes online. Table 6.2 shows the performance speed of each of our proposed methods, any value greater than 30 estimates per second is appropriate for online application. The camera used is an off the shelf sensor that can be easily added to already existing platforms. The average strawberry size in the 1250 in-field data samples was 50.21 $\text{px}^2$. The ellipsoid, disc summation and integration methods could respectively process 1770, 1623 and 324 strawberries per second at this scale.

Figure 6.4 shows the volume estimation frequency distribution for both unripe and ripe berries grouped into 20 inter-class bins. Ripe strawberries are expected to be larger in general than unripe, and this separation is confirmed in the volume predictions for both berry types.



(a) Mobile Robot (Grimstad and P. From, 2017) Platform

(b) Volume Distribution Of Strawberries In-Field

Figure 6.4: In-field volume distribution (b) over 2 classes and 1250 samples ran from a mobile robotic platform (a).

Table 6.2: Performance of phenotyping methods in calculations per second on an Intel Core i7-8700 CPU. 50.21 px$^2$ was the average in-field berry size.

| Method | Object Sizes px$^2$ | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | ***16*** | ***32*** | ***50*** | ***64*** | ***128*** | ***256*** | ***512*** | ***1024*** |
| **Ellipsoid** | 3341 | 2407 | 1770 | 1601 | 682 | 232 | 62 | 13 |
| **Disc Summation** | 2862 | 2278 | 1623 | 1427 | 675 | 231 | 64 | 13 |
| **Integration** | 424 | 362 | 324 | 269 | 172 | 90 | 36 | 11 |

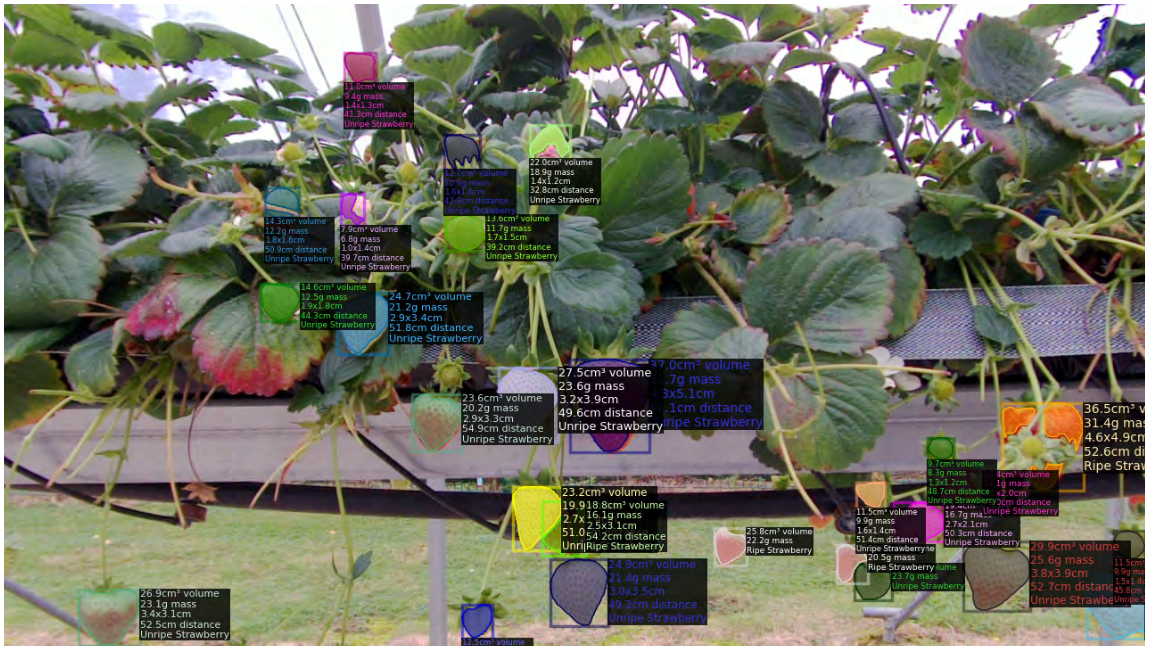

11.0cm$^3$ volume 9.4g mass 1.4x1.3cm 41.3cm distance Unripe Strawberry
12.7cm$^3$ volume 10.9g mass 1.6x1.6cm 42.9cm distance Unripe Strawberry
22.0cm$^3$ volume 18.9g mass 1.4x1.2cm 32.8cm distance Unripe Strawberry
14.3cm$^3$ volume 12.2g mass 1.8x1.6cm 50.9cm distance Unripe Strawberry
7.9cm$^3$ volume 6.8g mass 1.0x1.4cm 39.7cm distance Unripe Strawberry
13.6cm$^3$ volume 11.7g mass 1.7x1.5cm 39.2cm distance Unripe Strawberry
14.6cm$^3$ volume 12.5g mass 1.9x1.8cm 44.3cm distance Unripe Strawberry
24.7cm$^3$ volume 21.2g mass 2.9x3.4cm 51.8cm distance Unripe Strawberry
27.5cm$^3$ volume 23.6g mass 3.2x3.9cm 49.6cm distance Unripe Strawberry
23.6cm$^3$ volume 20.2g mass 2.9x3.3cm 54.9cm distance Unripe Strawberry
36.5cm$^3$ 31.4g 4.6x4.9cm 52.6cm Ripe
9.7cm$^3$ volume 8.3g mass 1.3x1.2cm 48.7cm distance Unripe Strawberry
23.2cm$^3$ volume
18.8cm$^3$ volume 16.1g mass 2.5x3.1cm 54.2cm distance Ripe Strawberry
25.8cm$^3$ volume 22.2g mass Ripe Strawberry
11.5cm$^3$ volume 9.9g mass 1.6x1.4cm 51.4cm distance Unripe Strawberry
16.7g mass 2.7x2.1cm 50.3cm distance Unripe Strawberry
20.5g mass Ripe Strawberry
23.7g mass Unripe Strawberry
29.9cm$^3$ volume 25.6g mass 3.8x3.9cm 52.7cm distance Unripe Strawberry
26.9cm$^3$ volume 23.1g mass 3.4x3.1cm 52.5cm distance Unripe Strawberry
24.9cm$^3$ volume 21.4g mass 3.0x3.5cm 49.2cm distance Unripe Strawberry
12.5cm$^3$ volume

Figure 6.5: In-field volume estimation results

### 6.2.4 Object Reconstruction

From the mass estimation methods, an intermediate step requires the surface of each strawberry to be reconstructed in some way to calculate the resultant volume for mass prediction. The reconstructed meshes for each strawberry used in the evaluation above are shown in Figure 6.6.

## 6.3 Conclusions

It is clear from the statistical results presented in Table 6.1 that both mass and volume can be estimated accurately from only two-dimensional data (segmentations) and optionally depth data for converting pixels to cm$^3$ and mass measures. Both

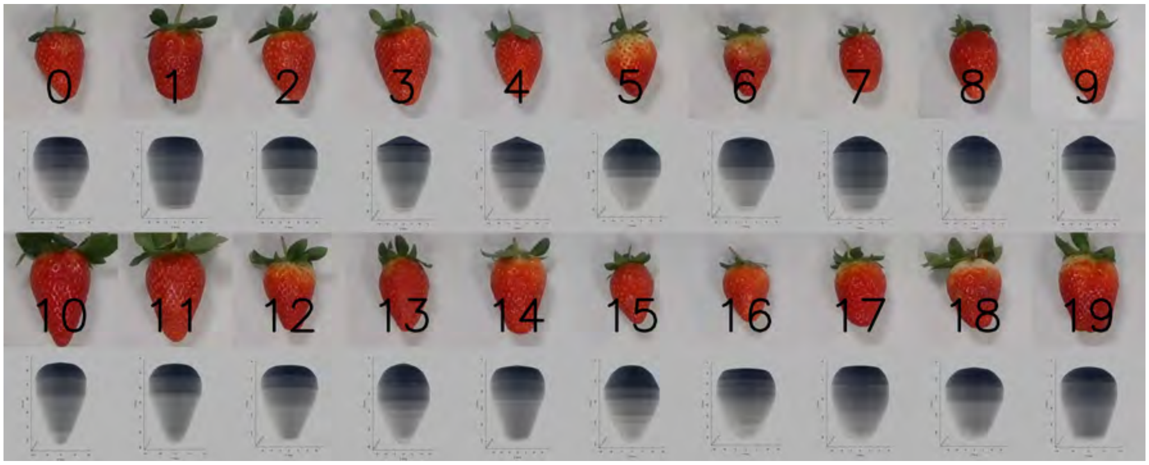


Figure 6.6: Mesh Reconstruction Summary all strawberries from 2D segment

methods have very low errors relative to the expected mass and volume measures. The median absolute error for volume is only 3.00 $cm^3$ (13% mean absolute percentage error) for the best method surface area integration, which is only 1.00 $cm^3$ above the maximum precision of the volumetric measurements. The results for mass estimation are also very similar, having only 2.34 g of error for the same method, which is well below the 5 g confidence interval of the 1 g accurate scales used.

This chapter presented a non-invasive, non-destructive, inexpensive method for volume and mass estimation in-field designed for use on a robotic platform. The evaluation of the methods has shown they are accurate in lab conditions and also work successfully in outdoor scenarios mounted on a Thorvald robot. In this chapter it is shown that the in-field estimates are in the range of expected values and that the methods can process between 324-1770 strawberries per second on consumer hardware. Future work will include gathering images and GT data in-field for fully evaluating the overall accuracy of the methods. The density value used in the mass calculation is also specific to only one variety of strawberry and may not generalise well to other varieties. Further work could improve on this by first classifying the variety, otherwise this measure will always need changing on a per-application basis. The current approach works under the assumption that the surface that is hidden is symmetric to the surface that is seen, future work would try to reconstruct the berry from partial view to get a more complete accuracy measurement from different orientations or to first correctly orient the berry segmentation to correct for the viewpoint. Finally, our data collection equipment was characterised by low precision, with subgram/submillimeter accurate equipment, better equipment could better evaluate the proposed methods.

# Chapter 7

# Conclusions and Future Work

In this thesis, methods are presented for automation in horticulture, focusing on non-destructive methods to detect, track and extract traits such as volume and mass from fruit. A combination of these approaches results in a system that covers practical application requirements for generating data points for horticultural processes such as yield estimation, disease prediction, cultivation management and enabling of robotic applications such as harvesting, data acquisition and autonomous precision farming. Driven by industry challenges, three novel solutions are presented to detect fruit in variable conditions of illumination and viewpoint in Chapter 4, a tracking component to re-identify fruit in clusters across image sequences in Chapter 5 and trait extraction methods that are applied on top of the tracking or detection outputs to non-destructively estimate crop parameters such as size and volume in Chapter 6. The motivation of this thesis is twofold: firstly, to demonstrate the capabilities of computer vision techniques applied in horticulture; secondly, the potential for computer vision techniques to facilitate more efficient and accurate crop assessment. Our proposed approaches have been shown to have several advantages over traditional manual approaches: they are non-destructive, fast, achieve state-of-the-art performance, are scalable and are relatively cheap. Specifically, this thesis is formed through the following contributions:

- **Chapter 2** *and* **Chapter 3** - A comprehensive index of computer vision, robotics, and autonomy approaches within agriculture and a review of current literature in this domain.
- **Chapter 4** - Coercive and free learning policies to shortcut learning more

representative features in *L*a*b*Fruits: A Rapid and Robust Outdoor Fruit Detection System Combining Bio-Inspired Features with One-Stage Deep Learning Networks* Kirk, Cielniak and Mangan (2020b). For combating the transfer of lab based object detection models on curated datasets to unseen outdoor data from multiple view points.

- **Chapter 5** - Novel extensions of detect-to-track based object tracking frameworks in *Robust Counting of Soft Fruit Through Occlusions with Re-identification* Kirk, Mangan and Cielniak (2021) to count soft-fruit in images (detect) and across image-sequences (track).
- **Chapter 6** - Introduction of approaches, for *Non-destructive Soft Fruit Mass and Volume Estimation for Phenotyping in Horticulture* Kirk, Cielniak and Mangan (2021a) to maximise the value of detection and tracking within horticulture to extract phenotypic traits from object detections and tracks in Chapter 6.

Our approach demonstrates detection, counting, and analysis of fruit through image-sequences, which makes the solution flexible enough to work on any type of fruit from video feeds, as opposed to static images containing information restricted to one point in time that does not exploit any spatial relationship. It is worth noting that the application and processes followed in this thesis can be applied in different domains, where perhaps problems are bound by similar constraints. An example is provided of how computer science practitioners can hone their systems to fulfil challenges posed by industry and societal needs to build better and more efficient relationships between research systems and practical deployments. The extracted information about the fruit can be used to recommend useful suggestions to a grower, for example reducing the number of required fruit-pickers, estimating harvest yield, reducing collection efforts, or optimising the harvest period for higher market cost. Ultimately, this enables farmers to make better use of their resources while optimising horticultural processes ahead of time, meeting environmental and management targets. Figure 7.1 shows the progression of published literature since the inception of this thesis, it shows a positive increase in interest within this domain, motivating the impact of published work and future works in this domain and adoption on farm.

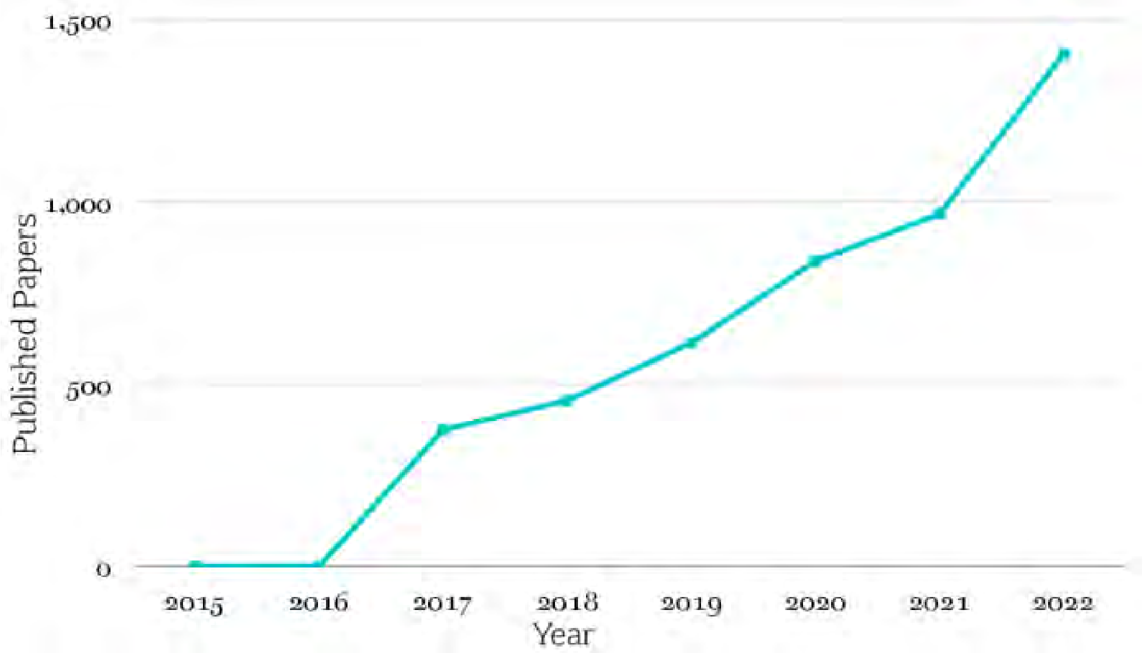


Figure 7.1: Research publications in Information and Computing Sciences for *soft fruit computer vision deep learning* keywords between 2016 and 2021 (mean citations 9) showing the progression of research in the domain since the inception of this thesis and peer-reviewed research papers.

## 7.1 Summary

In, Chapter 4 a 2.4% and 8.2% increase is achieved with our early fusion approach for detection on unseen viewpoints $V_{2-3}$ over the standard *RGB* and *CIELab* modalities alone. In comparison, the standard *RGB* and *CIELab* drop by 6.4% and 8.8% respectively for $F_1$scores between viewpoints $V_1$ and $V_{2-3}$. Similarly, when applied to the DeepFruits data set, an AP score increase of 2.8% (IoU = 0.5) and 6.2% (IoU = 0.4) is gained over *RGB* alone. Our $F_1$scores match those presented in the original paper, suggesting the added *CIELab* opponent features assist in classification of the detected objects more so than aiding the initial detection, since our obtained AP scores are consistently higher than *RGB* in all cases (2.8% and 6.2%). Our approach also gains a performance increase of 6.6 times that of the DeepFruits early fusion method utilising IR and only considering a single class. This improvement is likely to increase the applicability of the method to robotic fruit monitoring and harvesting systems that have limited computational and power resources. The limitations of the methods in this section is the use of a single sensor in the data set, colour calibration and multiple sensors in the dataset would allow better colour space representation and would demonstrate the generalisability between variable environmental conditions and hardware considerations.

Chapter 5 presents a framework for accurately tracking and counting fruit in a complex scene from bounding boxes, extending on current tracking architecture. Our system utilises re-identification features and label probability vectors with cosine similarity as well as robot odometry formatted as row position and data augmentation methods to maintain track consistency through occlusions and to maintain tracks in densely clustered detection regions. The results demonstrate that our system is capable of reliably tracking and counting multiple classes in clusters from multiple view points. An off the shelf cheap computer vision camera and modern GPU is used so that our system can be applied easily. Our results indicate an improvement of $L1$ loss from 41, 1, 19 and 59 (all classes, flower, ripe, unripe) to 7, 2, 2, 3 error in counting 500 frames of the *Amesti-1* sequence not used in training. On a modern GPU this is able to process >2400 bounding box detections per-second in a single forward pass of the network, enabling online tracking applications. The limitation of our tracking formalisation is the contextual switch between classes in clusters, surrounding objects can impact the network ability to classify an object into a specific class. Altering the network architecture to predict maturity as a regressed value rather than a softmax classification would allow a fuzzier association step and would hopefully increase the accuracy of the method in complex cases. In this chapter an evaluation of the situations where camera motion and object motion are not constant in the image is not made and instead a general assumption that the motion is of a constant velocity is made which decreases the number of application areas.

In Chapter 6 a non-invasive/destructive, inexpensive method for volume and mass estimation in-field designed for use on a robotic platform is presented. The evaluation of the methods has shown they are accurate in lab conditions and also operate well in outdoor scenarios mounted on a robotic platform. It is shown that the in-field estimates are in the range of expected values and that the methods can process between 324-1770 strawberries per second on consumer hardware. Collection of an outdoor dataset similar to the one presented in this chapter would be very expensive, however would allow better evaluation of the applicability of the methods to outdoor use on a mobile robot.

## 7.2 Future Work

Leveraging *CIELab* colour opponent features with *RGB* helped mitigate some luminance variation in the validation sets. Investigation into the benefits provided by this approach as the data set size increases would provide insight to the limitations and optimal accuracy increase through our proposed methods. As well as calibrating the cameras to improve the colour accuracy over multiple sensors. Visualisation of features and filters learned in the network would also provide intuition as to what the network is learning, which would be useful in seeing the difference between learnt *RGB* filters and colour opponent filters. To validate the removal of luminance further, this analysis could compare network activation for synthetically created Strawberries at variable luminosity, where uniform activation over variable parameters would indicate the removal of the detrimental effects of the parameter on overall accuracy. A single sensor is utilised in this experiment, to further our results calibrated colour from multiple sensors would show more representative results. Finally, analysis into accuracy increase with fewer classes or binned classes would show whether error is introduced through learning multiple classes.

For our tracking contributions, future work will include further evaluation of the proposed tracking system and further ablation studies into tracking by using additional cues, utilising different feature extraction networks, class embeddings and extending to full 3D space with approaches such as depth data integration or recent structure from motion approaches. Our counting system is applied to soft-fruit, however the generic nature of the proposed solution makes it applicable to a wide range of object counting applications beyond the soft-fruit scenario where a mobile robot can operate. Phenotypic trait extraction is a large field, and our methods only aim to extract useful features from objects already detected by our previous approaches. Future work will include estimating more parameters for each object and gathering images and GT data in-field that the methods can be fully evaluated regarding the overall accuracy of the methods. The density value used in the mass calculation is also specific to only one variety of strawberry and may not generalise well to other varieties. Further work could improve on this by first classifying the variety, otherwise this measure will always need changing on a per-application basis. The current approach works under the assumption that the surface that is hidden is symmetric to the surface that is seen, future work would try to reconstruct the berry from partial view to get a more complete accuracy measurement from different orientations or to first correctly orient the berry segment to correct for the viewpoint. Finally, our data collection equipment was characterised by low precision. With subgram/submillimeter accurate equipment, a better evaluation of the proposed methods could be given. Further future work would aim to combine all of the systems in a single open source repository in the form of a challenge to motivate future advances in the field of deep learning systems for horticulture.

## 7.3 Other Notable Contributions

In this section, non-scientific contributions to the industry, research domain and community are presented. The work presented in this thesis required contributions which have realised impact with their respective domains.

### 7.3.1 Software

In completion of the requirements in this thesis, numerous software packages to aid with developing or deploying novel computer vision solutions for horticulture have been developed. The capabilities of each piece of software are summarised below to enable future research programmes in this space. The software packages relate to the deployment of deep learning models and the collection of data in horticultural/robotic environments.

**Modular Robotic Object Detection Framework** (*2020*) A modular object detection, segmentation, and tracking (extension) package for the robotic operating system, to allow robots to utilise state-of-the-art detectors in the field. Allows the deployment of custom models easily and in a standard format (Kirk, 2020a).

**Modular Robotic Object Tracking Framework** (*2021*) An extension of the Rasberry Perception package (above), enables modular implementations of trackers in the field. Contains an implementation and extension of the Bayes People Tracker (Bellotto and Hu, 2010), to support tracking soft-fruits on a robotic platform from object detectors and segments (Kirk, 2021a) which utilises a custom implementation of the C++ framework for Bayesian Filter Tracking (UKF, EKF, Particles) to support re-identification features (Kirk, Bellotto et al., 2021).

**Topic Store** (*2020*) Robotic operating system (ROS) or standalone python package to enable large-scale and easy data collection in-field. Data collected via this package is fully searchable and queryable allowing the creation of complex datasets easily. In example, querying for images based on GPS location, weather condition or with specific sensors (Kirk, 2020b).

**Topic Compression** (*2020*) A compression/decompression library to allow the collection of high bandwidth data sources in the field, such as multiple cameras of high-resolution depth/hyper-spectral images (Kirk, 2021c). Implements common compression methods such as JPEG/PNG and has implementations of state-of-the-art 16-bit compression methods such as RVL (Wilson, 2017).

**Long Term Depth Capture** (*2021*) A simple wrapper library for capturing long image sequences, primarily developed to allow soft-fruit growers and researchers to capture time-lapse data of an entire berry season (Kirk, 2021b).

**Multi-Camera Data Collection** (*2018*) A fast C++ data capture library primarily for capturing depth and RGB data, supports capturing from multiple devices and aligning depth maps for multi-camera deployments (Kirk, 2018).

### 7.3.2 Dissemination Activities

During the period of this thesis, the following denotes dissemination activities attended and awards achieved contributing to the requirements of this thesis:

**BBC Countryfile Autumn Diaries** (*2019*) I presented multiple challenges where algorithms present in this thesis were pitted against presenters in a wide variety of horticultural tasks *BBC One - Countryfile Autumn Diaries, 2019, Episode 2* 2022.

**IAgreE Best Talk** (*2019*) Institution of Agricultural Engineers awarded the best talk award for an oral presentation of our paper *"L*a*b*Fruits: A Rapid and Robust Outdoor Fruit Detection System Combining Bio-Inspired Features with One-Stage Deep Learning Networks"* Kirk, Cielniak and Mangan, 2020b.

**New Scientist** (*2020*) New Scientist Live present thought-provoking talks from the world's best science speakers, I presented our work in this thesis pushing the need for application in agriculture titled *"Future of food and agriculture"* *New Scientist Live - Future of Food and Agriculture* 2022.

**Berry Garden Growers Annual Conference** (*2020*) I presented the advancements possible deploying technology developed during this thesis to over 40% of all berry growers members in the UK.

**CERES Award** (*2021*) Grant award of £250,000 to further develop solutions extended from the **tracking** paper to develop yield forecasting solutions for the industry.

**ICVS Best Paper** (*2021*) I was acknowledged and presented with the best paper award at the ICVS conference for our work *"Robust Counting of Soft Fruit through Occlusions with Re-Identification"* Kirk, Mangan and Cielniak, 2021.

**BBC Countryfile** (*2021*) I deployed the system presented in Kirk, Mangan and Cielniak, 2021 and Chapter 5 to the task of fruit counting for yield estimation against two farmworkers with 20 years of experience *Countryfile - Harvest Special* 2022.

**Innovate UK Grant** (*2022*) Grant award of £400,000 to deploy AI systems presented in this thesis within the industry.

**ICVS Chair** (*2022*) Invitation to chair the ICVS 2022 conference in Crete *ICVS 2022 - Crete* 2022.

**FruitCast** (*2022*) First spin-out of the University of Lincoln. Driving innovation and research adoption in soft-fruit analytics for horticulture.

# References

Achanta, Radhakrishna et al. (2012). 'SLIC superpixels compared to state-of-the-art superpixel methods'. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.11, pp. 2274–2281. ISSN: 01628828. DOI: 10.1109/TPAMI.2012.120 (cit. on pp. 88, 91, 92).

Agarap, Abien Fred (2018). 'Deep Learning using Rectified Linear Units (ReLU)'. In: *CoRR* abs/1803.08375. arXiv: 1803.08375. URL: http://arxiv.org/abs/1803.08375 (cit. on p. 25).

An, Nan et al. (2016). 'Plant high-throughput phenotyping using photogrammetry and imaging techniques to measure leaf length and rosette area'. In: *Computers and Electronics in Agriculture* 127, pp. 376–394. ISSN: 01681699. DOI: 10.1016/j.compag.2016.04.002. URL: http://dx.doi.org/10.1016/j.compag.2016.04.002 (cit. on pp. 63, 73, 83).

Anagnostis, Athanasios et al. (2022). 'Machine Learning Technology and Its Current Implementation in Agriculture'. In: *Information and Communication Technologies for Agriculture—Theme II: Data*. Springer, pp. 41–73 (cit. on p. 3).

Baeten, Johan et al. (2008). 'Autonomous fruit picking machine: A robotic apple harvester'. In: *Springer Tracts in Advanced Robotics* 42, pp. 531–539. ISSN: 16107438. DOI: 10.1007/978-3-540-75404-6{_}51 (cit. on pp. 66, 67).

Bai, X. D. et al. (2013). 'Crop segmentation from images by morphology modeling in the CIE L*a*b*color space'. In: *Computers and Electronics in Agriculture* 99, pp. 21–34. ISSN: 01681699. DOI: 10.1016/j.compag.2013.08.022 (cit. on pp. 88, 91, 92).

Bai, Xiaodong et al. (2014). 'Vegetation segmentation robust to illumination variations based on clustering and morphology modelling'. In: *Biosystems Engineering* 125, pp. 80–97. ISSN: 15375110. DOI: 10.1016/j.biosystemseng.2014.06.015. URL: http://dx.doi.org/10.1016/j.biosystemseng.2014.06.015 (cit. on pp. 88, 91, 92).

*BBC One - Countryfile Autumn Diaries, 2019, Episode 2* (2022). en-GB. URL: https://www.bbc.co.uk/programmes/m0009vj0 (visited on 28th Mar. 2022) (cit. on p. 148).

Bellocchio, E. et al. (2019). 'Weakly Supervised Fruit Counting for Yield Estimation Using Spatial Consistency'. In: *IEEE Robotics and Automation Letters* 4.3, pp. 2348–2355. DOI: 10.1109/LRA.2019.2903260 (cit. on p. 114).

Bellotto, Nicola and Huosheng Hu (May 2010). 'Computationally Efficient Solutions for Tracking People with a Mobile Robot: An Experimental Evaluation of bayesian Filters'. In: *Autonomous Robots* 28. DOI: 10.1007/s10514-009-9167-2 (cit. on p. 147).

Bewley, Alex et al. (2016). 'Simple Online and Realtime Tracking'. In: *CoRR* abs/1602.00763. arXiv: 1602.00763. URL: http://arxiv.org/abs/1602.00763 (cit. on pp. 51, 55, 115).

Bochinski, Erik, Volker Eiselein and Thomas Sikora (Aug. 2017a). 'High-Speed Tracking-by-Detection Without Using Image Information'. In: *International Workshop on Traffic and Street Surveillance for Safety and Security at IEEE AVSS 2017*. Lecce, Italy. URL: http://elvera.nue.tu-berlin.de/files/1517Bochinski2017.pdf (cit. on p. 51).

– (Aug. 2017b). 'High-Speed Tracking-by-Detection Without Using Image Information'. In: *International Workshop on Traffic and Street Surveillance for Safety and Security at IEEE AVSS 2017*. Lecce, Italy. URL: http://elvera.nue.tu-berlin.de/files/1517Bochinski2017.pdf (cit. on p. 53).

Bochinski, Erik, Tobias Senst and Thomas Sikora (Nov. 2018). 'Extending IOU Based Multi-Object Tracking by Visual Information'. In: *IEEE International Conference on Advanced Video and Signals-based Surveillance*. Auckland, New Zealand, pp. 441–446. URL: http://elvera.nue.tu-berlin.de/files/1547Bochinski2018.pdf (cit. on pp. 52, 53).

Bolya, Daniel et al. (2019). 'YOLACT: Real-time Instance Segmentation'. In: *CoRR* abs/1904.02689. arXiv: 1904.02689. URL: http://arxiv.org/abs/1904.02689 (cit. on p. 50).

Chaivivatrakul, Supawadee et al. (2014). 'Automatic morphological trait characterization for corn plants via 3D holographic reconstruction'. In: *Computers and Electronics in Agriculture* 109, pp. 109–123. ISSN: 01681699. DOI: 10.1016/j.compag.2014.09.005. URL: http://dx.doi.org/10.1016/j.compag.2014.09.005 (cit. on pp. 60, 66, 83).

Chen, S. W. et al. (2017). 'Counting Apples and Oranges With Deep Learning: A Data-Driven Approach'. In: *IEEE Robotics and Automation Letters* 2.2, pp. 781–788. DOI: 10.1109/LRA.2017.2651944 (cit. on pp. 71, 114).

Chen, Yang, W. S. Lee et al. (July 2019). 'Strawberry Yield Prediction Based on a Deep Neural Network Using High-Resolution Aerial Orthoimages'. In: *Remote Sensing* 11. DOI: 10.3390/rs11131584 (cit. on p. 128).

Chen, Yang, Won Suk Lee et al. (2019). 'Strawberry Yield Prediction Based on a Deep Neural Network Using High-Resolution Aerial Orthoimages'. In: *Remote Sensing* 11.13. ISSN: 2072-4292. DOI: 10.3390/rs11131584. URL: https://www.mdpi.com/2072-4292/11/13/1584 (cit. on pp. 68, 69, 104).

*Countryfile - Harvest Special* (2022). en-GB. URL: https://www.bbc.co.uk/iplayer/episode/m0010kl6/countryfile-harvest-special (visited on 28th Mar. 2022) (cit. on p. 148).

Cybenko, G. (Dec. 1989). 'Approximation by superpositions of a sigmoidal function'. In: *Mathematics of Control, Signals, and Systems (MCSS)* 2.4, pp. 303–314. ISSN: 0932-4194. DOI: 10.1007/BF02551274. URL: http://dx.doi.org/10.1007/BF02551274 (cit. on p. 19).

Deng, Jia et al. (2009). 'ImageNet: A large-scale hierarchical image database'. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848 (cit. on p. 33).

– (n.d.). *ImageNet: A Large-Scale Hierarchical Image Database.* Tech. rep. URL: http://www.image-net.org. (cit. on pp. 99, 103).

Dey, Debadeepta, Lily Mummert and Rahul Sukthankar (2012). 'Classification of plant structures from uncalibrated image sequences'. In: *Proceedings of IEEE Workshop on Applications of Computer Vision*, pp. 329–336. ISSN: 21583978. DOI: 10.1109/WACV.2012.6163017 (cit. on pp. 62, 64, 82, 83).

Diago, Maria Paz et al. (2012). 'Grapevine yield and leaf area estimation using supervised classification methodology on RGB images taken under field conditions'. In: *Sensors (Switzerland)* 12.12, pp. 16988–17006. ISSN: 14248220. DOI: 10.3390/s121216988 (cit. on pp. 64, 83).

Dong, Wenbo, Pravakar Roy and Volkan Isler (2018). 'Semantic Mapping for Orchard Environments by Merging Two-Sides Reconstructions of Tree Rows'. In: *CoRR* abs/1809.00075. arXiv: 1809.00075. URL: http://arxiv.org/abs/1809.00075 (cit. on p. 72).

ElMasry, Gamal and Da-Wen Sun (2010). 'Principles of hyperspectral imaging technology'. In: *Hyperspectral imaging for food quality analysis and control*, pp. 3–43 (cit. on p. 75).

ElMasry, Gamal, Ning Wang et al. (2007). 'Hyperspectral imaging for nondestructive determination of some quality attributes for strawberry'. In: *Journal of Food Engineering* 81.1, pp. 98–107. ISSN: 02608774. DOI: 10.1016/j.jfoodeng.2006.10.016 (cit. on pp. 75, 76).

Engilberge, M, E Collins and S Süsstrunk (Sept. 2017). 'Color representation in deep neural networks'. In: *2017 IEEE International Conference on Image Processing (ICIP)*, pp. 2786–2790. DOI: 10.1109/ICIP.2017.8296790 (cit. on p. 82).

Everingham, M. et al. (n.d.). *The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results.* http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.ht (cit. on p. 45).

Everingham, Mark et al. (June 2010). 'The Pascal Visual Object Classes (VOC) Challenge'. en. In: *International Journal of Computer Vision* 88.2, pp. 303–338.

ISSN: 1573-1405. DOI: 10.1007/s11263-009-0275-4. URL: https://doi.org/10.1007/s11263-009-0275-4 (visited on 23rd Apr. 2022) (cit. on p. 33).

Feng, Guo, Cao Qixin and Nagata Masateru (2008). 'Fruit Detachment and Classification Method for Strawberry Harvesting Robot'. In: *International Journal of Advanced Robotic Systems* 5.1, pp. 41–48. ISSN: 17298814. DOI: 10.5772/5662 (cit. on pp. 60, 63, 73, 83).

Font, Davinia, T. Pallejà et al. (2014). 'Counting red grapes in vineyards by detecting specular spherical reflection peaks in RGB images obtained at night with artificial illumination'. In: *Computers and Electronics in Agriculture* 108, pp. 105–111. ISSN: 01681699. DOI: 10.1016/j.compag.2014.07.006. URL: http://dx.doi.org/10.1016/j.compag.2014.07.006 (cit. on pp. 60, 65).

Font, Davinia, Tomàs Pallejà et al. (2014). 'A proposal for automatic fruit harvesting by combining a low cost stereovision camera and a robotic arm'. In: *Sensors (Switzerland)* 14.7, pp. 11557–11579. ISSN: 14248220. DOI: 10.3390/s140711557 (cit. on pp. 60, 67, 83).

Food, F A O - and Agriculture Organization of the United Nations (2009). *Global Agriculture Towards 2050*. DOI: 10.1007/978-94-6091-609-0{_}1. URL: http://www.fao.org/fileadmin/templates/wsfs/docs/Issues%5C%5Fpapers/HLEF2050%5C%5FGlobal%5C%5FAgriculture.pdf (cit. on p. 59).

Forssén, Per Erik (2007). 'Maximally stable colour regions for recognition and matching'. In: *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. ISSN: 10636919. DOI: 10.1109/CVPR.2007.383120 (cit. on p. 63).

Fu, Cheng-Yang, Mykhailo Shvets and Alexander C. Berg (2019). 'RetinaMask: Learning to predict masks improves state-of-the-art single-shot detection for free'. In: *CoRR* abs/1901.03353. arXiv: 1901.03353. URL: http://arxiv.org/abs/1901.03353 (cit. on p. 49).

Ge, Y. et al. (2019). 'Fruit Localization and Environment Perception for Strawberry Harvesting Robots'. In: *IEEE Access* 7, pp. 147642–147652. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2019.2946369 (cit. on p. 104).

Girshick, Ross B. (2015). 'Fast R-CNN'. In: *CoRR* abs/1504.08083. arXiv: 1504.08083. URL: http://arxiv.org/abs/1504.08083 (cit. on pp. 37, 40).

Girshick, Ross B. et al. (2013). 'Rich feature hierarchies for accurate object detection and semantic segmentation'. In: *CoRR* abs/1311.2524. arXiv: 1311.2524. URL: http://arxiv.org/abs/1311.2524 (cit. on pp. 35, 51).

Goddard, M.E. and B.J. Hayes (2007). 'Genomic selection'. In: *Journal of Animal Breeding and Genetics* 124.6, pp. 323–330. DOI: 10.1111/j.1439-0388.2007.00702.x (cit. on p. 73).

Gongal, A. et al. (2015). 'Sensors and systems for fruit detection and localization: A review'. In: *Computers and Electronics in Agriculture* 116, pp. 8–19. ISSN: 01681699. DOI: 10.1016/j.compag.2015.05.021. URL: http://dx.doi.org/10.1016/j.compag.2015.05.021 (cit. on pp. 7, 67).

Grimstad, Lars and Pål From (Sept. 2017). 'The Thorvald II Agricultural Robotic System'. In: *Robotics* 6.4, p. 24. ISSN: 2218-6581. DOI: 10.3390/robotics6040024. URL: http://www.mdpi.com/2218-6581/6/4/24 (cit. on pp. 86, 136, 137).

Grimstad, Lars and Pål Johan From (2017). 'The Thorvald II Agricultural Robotic System'. In: *Robotics* 6.4. ISSN: 2218-6581. DOI: 10.3390/robotics6040024. URL: https://www.mdpi.com/2218-6581/6/4/24 (cit. on p. 119).

Häni, Nicolai, Pravakar Roy and Volkan Isler (2018). 'A Comparative Study of Fruit Detection and Counting Methods for Yield Mapping in Apple Orchards'. In: *CoRR* abs/1810.09499. arXiv: 1810.09499. URL: http://arxiv.org/abs/1810.09499 (cit. on p. 9).

– (2020). 'A comparative study of fruit detection and counting methods for yield mapping in apple orchards'. In: *Journal of Field Robotics* 37.2, pp. 263–282. DOI: https://doi.org/10.1002/rob.21902. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/rob.21902. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21902 (cit. on p. 72).

Hayashi, Shigehiko et al. (2010). 'Evaluation of a strawberry-harvesting robot in a field test'. In: *Biosystems Engineering* 105.2, pp. 160–171. ISSN: 15375110. DOI: 10.1016/j.biosystemseng.2009.09.011. URL: http://dx.doi.org/10.1016/j.biosystemseng.2009.09.011 (cit. on pp. 60, 67, 83).

He, Joe, Richard Harrison and Bo Li (Dec. 2017). 'A novel 3D imaging system for strawberry phenotyping'. In: *Plant Methods* 13. DOI: 10.1186/s13007-017-0243-x (cit. on pp. 74, 128).

He, Kaiming, Georgia Gkioxari et al. (2017). 'Mask R-CNN'. In: *CoRR* abs/1703.06870. arXiv: 1703.06870. URL: http://arxiv.org/abs/1703.06870 (cit. on p. 40).

He, Kaiming, Xiangyu Zhang et al. (2015). 'Deep Residual Learning for Image Recognition'. In: *CoRR* abs/1512.03385. arXiv: 1512.03385. URL: http://arxiv.org/abs/1512.03385 (cit. on p. 119).

– (2016). 'Deep Residual Learning for Image Recognition'. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. ISBN: 978-1-4673-8851-1. DOI: 10.1109/CVPR.2016.90. URL: http://image-net.org/challenges/LSVRC/2015/%20http://ieeexplore.ieee.org/document/7780459/ (cit. on pp. 79, 80, 96, 97).

Hertwich, Edgar G., Ester van der Voet and Arnold Tukker (2010). *Assessing the Environmental Impacts of Consumption and Production. Priority Products and Materials*, p. 112. ISBN: 9789280730845 (cit. on p. 59).

Huang, Zhuoling, Sam Wane and Simon Parsons (2017). 'Towards Automated Strawberry Harvesting: Identifying the Picking Point'. In: *TAROS* (cit. on p. 74).

Hunt, E. Raymond et al. (2005). 'Evaluation of digital photography from model aircraft for remote sensing of crop biomass and nitrogen status'. In: *Precision Agriculture* 6.4, pp. 359–378. ISSN: 13852256. DOI: 10.1007/s11119-005-2324-5 (cit. on pp. 63, 73).

*ICVS 2022 - Crete* (2022). URL: https://www.icvs.info/index.php/2-uncategorised/80-icvs-2022-crete (visited on 28th Mar. 2022) (cit. on p. 148).

Ishikawa, T. et al. (May 2018). 'CLASSIFICATION OF STRAWBERRY FRUIT SHAPE BY MACHINE LEARNING'. In: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences* XLII-2, pp. 463–470. DOI: 10.5194/isprs-archives-XLII-2-463-2018 (cit. on p. 74).

Jay, Sylvain et al. (2015). 'In-field crop row phenotyping from 3D modeling performed using Structure from Motion'. In: *Computers and Electronics in Agriculture* 110, pp. 70–77. ISSN: 01681699. DOI: 10.1016/j.compag.2014.09.021. URL: http://dx.doi.org/10.1016/j.compag.2014.09.021 (cit. on p. 74).

Joly, Alexis et al. (2017). 'LifeCLEF 2017 Lab Overview: Multimedia Species Identification Challenges'. In: *Experimental IR Meets Multilinguality, Multimodality, and Interaction*. Ed. by Gareth J.F. Jones et al. Cham: Springer International Publishing, pp. 255–274. ISBN: 978-3-319-65813-1 (cit. on p. 70).

Kaczmarek, Adam L. (2017). 'Stereo vision with Equal Baseline Multiple Camera Set (EBMCS) for obtaining depth maps of plants'. In: *Computers and Electronics in Agriculture* 135, pp. 23–37. ISSN: 01681699. DOI: 10.1016/j.compag.2016.11.022. URL: http://dx.doi.org/10.1016/j.compag.2016.11.022 (cit. on pp. 59, 60, 62, 68, 82, 83).

Kazhdan, Michael, Matthew Bolitho and Hugues Hoppe (2006). 'Poisson Surface Reconstruction'. In: *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*. SGP '06. Cagliari, Sardinia, Italy: Eurographics Association, pp. 61–70. ISBN: 3905673363 (cit. on p. 74).

Kazmi, Wajahat et al. (2014). 'Indoor and outdoor depth imaging of leaves with time-of-flight and stereo vision sensors: Analysis and comparison'. In: *ISPRS Journal of Photogrammetry and Remote Sensing* 88, pp. 128–146. ISSN: 09242716. DOI: 10.1016/j.isprsjprs.2013.11.012. URL: http://dx.doi.org/10.1016/j.isprsjprs.2013.11.012 (cit. on p. 66).

Kim, Byoungjun et al. (2021). 'Improved Vision-Based Detection of Strawberry Diseases Using a Deep Neural Network'. In: *Frontiers in Plant Science* 11. ISSN: 1664-462X. DOI: 10.3389/fpls.2020.559172. URL: https://www.frontiersin.org/article/10.3389/fpls.2020.559172 (cit. on p. 69).

Kirk, Raymond (Jan. 2018). *StrawberryData*. Version 1.0.0. URL: https://github.com/RaymondKirk/StrawberryData (cit. on p. 147).

– (Jan. 2020a). *rasberry$_p$erception*. Version 1.0.0. URL: https://github.com/RaymondKirk/rasberry_perception (cit. on p. 147).

– (Jan. 2020b). *topic$_s$tore*. Version 1.0.0. URL: https://github.com/RaymondKirk/topic_store (cit. on p. 147).

– (Jan. 2021a). *rasberry$_t$racking*. Version 1.0.0. URL: https://github.com/RaymondKirk/rasberry_tracking (cit. on p. 147).

– (Jan. 2021b). *realsense$_s$ave$_e$xample*. Version 1.0.0. URL: https://github.com/RaymondKirk/realsense_save_example (cit. on p. 147).

– (Jan. 2021c). *topic$_c$ompression*. Version 1.0.0. URL: https://github.com/RaymondKirk/topic_compression (cit. on p. 147).

Kirk, Raymond, Nicola Bellotto et al. (Jan. 2021). *bayestracking*. Version 1.0.0. URL: https://github.com/RaymondKirk/bayestracking (cit. on p. 147).

Kirk, Raymond, Grzegorz Cielniak and Michael Mangan (2020a). 'Feasibility Study of In-Field Phenotypic Trait Extraction for Robotic Soft-Fruit Operations'. In: *UKRAS* (cit. on p. 14).

– (2020b). 'L*a*b*Fruits: A Rapid and Robust Outdoor Fruit Detection System Combining Bio-Inspired Features with One-Stage Deep Learning Networks'. In: *Sensors* 20.1. ISSN: 1424-8220. DOI: 10.3390/s20010275. URL: https://www.mdpi.com/1424-8220/20/1/275 (cit. on pp. 13, 14, 68, 78, 114, 142, 148).

– (2020c). 'L*a*b*Fruits: A Rapid and Robust Outdoor Fruit Detection System Combining Bio-Inspired Features with One-Stage Deep Learning Networks'. In: *Sensors* 20.1. ISSN: 1424-8220. DOI: 10.3390/s20010275. URL: https://www.mdpi.com/1424-8220/20/1/275 (cit. on pp. 73, 74, 128).

– (2021a). 'Non-destructive Soft Fruit Mass and Volume Estimation for Phenotyping in Horticulture'. In: *ICVS - International Conference on Computer Vision Systems*. Springer International Publishing, pp. 223–233 (cit. on pp. 14, 15, 126, 142).

– (2021b). 'Robust Counting of Soft Fruit Through Occlusions with Re-identification'. In: *ICVS - International Conference on Computer Vision Systems*. Computer Vision Systems. Springer International Publishing, pp. 211–222 (cit. on p. 15).

Kirk, Raymond, Michael Mangan and Grzegorz Cielniak (2021). 'Robust Counting of Soft Fruit Through Occlusions with Re-identification'. In: *International Conference on Computer Vision Systems*. Springer, pp. 211–222 (cit. on pp. 13, 112, 142, 148).

Konstantinovic, M et al. (2007). 'Detection of root biomass using ultra wideband radar–an approach to potato nest positioning'. In: *Agricultural Engineering International: CIGR Journal* (cit. on p. 1).

Kusumam, Keerthy et al. (2017). '3D-vision based detection, localization, and sizing of broccoli heads in the field'. In: *Journal of Field Robotics* December 2016, pp. 1–14. DOI: 10.1002/rob.21726 (cit. on pp. 60, 66, 83).

Lamb, N. and M. C. Chuah (Dec. 2018). 'A Strawberry Detection System Using Convolutional Neural Networks'. In: *2018 IEEE International Conference on Big Data (Big Data)*, pp. 2515–2520. DOI: 10.1109/BigData.2018.8622466 (cit. on p. 104).

Lamb, Nikolas and Mooi Choo Chuah (2018). 'A Strawberry Detection System Using Convolutional Neural Networks'. In: *2018 IEEE International Conference on Big Data (Big Data)*, pp. 2515–2520. DOI: 10.1109/BigData.2018.8622466 (cit. on p. 70).

Leal-Taixé, Laura, Anton Milan, Ian D. Reid et al. (2015). 'MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking'. In: *CoRR* abs/1504.01942. arXiv: 1504.01942. URL: http://arxiv.org/abs/1504.01942 (cit. on p. 35).

Leal-Taixé, Laura, Anton Milan, Konrad Schindler et al. (2017). 'Tracking the Trackers: An Analysis of the State of the Art in Multiple Object Tracking'. In: *CoRR* abs/1704.02781. arXiv: 1704.02781. URL: http://arxiv.org/abs/1704.02781 (cit. on p. 114).

Lightbody, Peter, Tomáš Krajník and Marc Hanheide (2017). *An Efficient Visual Fiducial Localisation System*. ISBN: 9781450344869. URL: http://dx.doi.org/10. (cit. on p. 85).

Lin, Tsung-Yi, Piotr Dollár et al. (n.d.). *Feature Pyramid Networks for Object Detection*. Tech. rep. URL: https://arxiv.org/pdf/1612.03144.pdf (cit. on pp. 47, 48, 79, 80, 96, 97).

Lin, Tsung-Yi, Priya Goyal et al. (2017). 'Focal loss for dense object detection'. In: *arXiv preprint arXiv:1708.02002*, pp. 2980–2988. ISSN: 15505499. DOI: 10.1109/ICCV.2017.324. URL: https://github.com/facebookresearch/Detectron. (cit. on pp. 47, 79, 80, 98).

Lin, Tsung-Yi, Michael Maire et al. (2014a). 'Microsoft {COCO:} Common Objects in Context'. In: *CoRR* abs/1405.0 (cit. on pp. 32, 34).

– (2014b). 'Microsoft {COCO:} Common Objects in Context'. In: *CoRR* abs/1405.0 (cit. on p. 97).

Lin, Tsung-Yi et al. (2017). 'Focal Loss for Dense Object Detection'. In: *CoRR* abs/1708.02002. arXiv: 1708.02002. URL: http://arxiv.org/abs/1708.02002 (cit. on pp. 13, 29, 51, 78).

Linker, Raphael, Oded Cohen and Amos Naor (2012). 'Determination of the number of green apples in RGB images recorded in orchards'. In: *Computers and Electronics in Agriculture* 81, pp. 45–57. ISSN: 01681699. DOI: 10.1016/j.compag.2011.11.007.

URL: http://dx.doi.org/10.1016/j.compag.2011.11.007 (cit. on pp. 60, 63, 64, 73, 83).

Liu, Scarlett and Mark Whitty (2015). 'Automatic grape bunch detection in vineyards with an SVM classifier'. In: *Journal of Applied Logic* 13.4, pp. 643–653. ISSN: 15708683. DOI: 10.1016/j.jal.2015.06.001. URL: http://dx.doi.org/10.1016/j.jal.2015.06.001 (cit. on pp. 59, 60).

Liu, Wei et al. (2015). 'SSD: Single Shot MultiBox Detector'. In: *CoRR* abs/1512.02325. arXiv: 1512.02325. URL: http://arxiv.org/abs/1512.02325 (cit. on pp. 45, 49, 106).

Liu, X. et al. (2019). 'Monocular Camera Based Fruit Counting and Mapping With Semantic Data Association'. In: *IEEE Robotics and Automation Letters* 4.3, pp. 2296–2303. DOI: 10.1109/LRA.2019.2901987 (cit. on pp. 71, 114).

Liu, Xu et al. (2018). 'Robust Fruit Counting: Combining Deep Learning, Tracking, and Structure from Motion'. In: *CoRR* abs/1804.00307. arXiv: 1804.00307. URL: http://arxiv.org/abs/1804.00307 (cit. on pp. 71, 114, 115).

MacLeod, Robert B. et al. (2006). *Outlines of a Theory of the Light Sense.* Vol. 80. 1. Berlin: Springer, p. 163. DOI: 10.2307/1420566 (cit. on pp. 87, 89).

Maldonado, Walter and José Carlos Barbosa (2016). 'Automatic green fruit counting in orange trees using digital images'. In: *Computers and Electronics in Agriculture* 127, pp. 572–581. ISSN: 01681699. DOI: 10.1016/j.compag.2016.07.023. URL: http://dx.doi.org/10.1016/j.compag.2016.07.023 (cit. on pp. 60, 62, 65, 82, 83).

Mathey, Megan et al. (Oct. 2013). 'Large-Scale Standardized Phenotyping of Strawberry in RosBREED'. In: *Journal- American Pomological Society* 67, pp. 205–216 (cit. on p. 73).

Mathibela, Bonolo, Ingmar Posner and Paul Newman (2013). 'A roadwork scene signature based on the opponent colour model'. In: *IEEE International Conference on Intelligent Robots and Systems*, pp. 4394–4400. ISSN: 21530858. DOI: 10.1109/IROS.2013.6696987 (cit. on pp. 87, 89–92).

McCool, Christopher et al. (May 2016). 'Visual detection of occluded crop: For automated harvesting'. In: *Proceedings - IEEE International Conference on Robotics and Automation.* Vol. 2016-June. IEEE, pp. 2506–2512. ISBN: 9781467380263. DOI: 10.1109/ICRA.2016.7487405. URL: http://ieeexplore.ieee.org/document/7487405/ (cit. on pp. 84, 106).

Mekhalfi, Mohamed Lamine et al. (2020). 'Vision System for Automatic On-Tree Kiwifruit Counting and Yield Estimation'. In: *Sensors* 20.15. ISSN: 1424-8220. DOI: 10.3390/s20154214. URL: https://www.mdpi.com/1424-8220/20/15/4214 (cit. on pp. 71, 114).

Meulebroeck, Thienpont and Ottevaere (Dec. 2016). 'Photonics enhanced sensors for food monitoring: part 1'. In: *IEEE Instrumentation Measurement Magazine* 19.6, pp. 35–45. ISSN: 1094-6969. DOI: 10.1109/MIM.2016.7777651 (cit. on pp. 81, 83).

Milan, A. et al. (Mar. 2016). 'MOT16: A Benchmark for Multi-Object Tracking'. In: *arXiv:1603.00831 [cs]*. arXiv: 1603.00831. URL: http://arxiv.org/abs/1603.00831 (cit. on p. 115).

Miller, George A. (Nov. 1995). 'WordNet: A Lexical Database for English'. In: *Commun. ACM* 38.11, pp. 39–41. ISSN: 0001-0782. DOI: 10.1145/219717.219748. URL: https://doi.org/10.1145/219717.219748 (cit. on p. 33).

Minsky, Marvin and Seymour Papert (1969). *Perceptrons: An Introduction to Computational Geometry*. Cambridge, MA, USA: MIT Press (cit. on p. 20).

Mohamed, Eslam et al. (July 2021). 'INSTA-YOLO: Real-Time Instance Segmentation'. In: *arXiv:2102.06777 [cs]*. arXiv: 2102.06777. URL: http://arxiv.org/abs/2102.06777 (visited on 2nd May 2022) (cit. on p. 49).

*New Scientist Live - Future of Food and Agriculture* (2022). en-GB. URL: https://agri-epicentre.com/event/new-scientist-live-future-of-food-and-agriculture/ (visited on 28th Mar. 2022) (cit. on p. 148).

Nguyen, Tien Thanh et al. (2016). 'Detection of red and bicoloured apples on tree with an RGB-D camera'. In: *Biosystems Engineering* 146, pp. 33–44. ISSN: 15375110. DOI: 10.1016/j.biosystemseng.2016.01.007. URL: http://dx.doi.org/10.1016/j.biosystemseng.2016.01.007 (cit. on pp. 61, 65, 82, 83).

Ning, Guanghan et al. (2016). 'Spatially Supervised Recurrent Convolutional Neural Networks for Visual Object Tracking'. In: *CoRR* abs/1607.05781. arXiv: 1607.05781. URL: http://arxiv.org/abs/1607.05781 (cit. on p. 51).

Otsu, Nobuyuki (1979). 'A Threshold Selection Method from Gray-Level Histograms'. In: *IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS* 9.1, pp. 62–66 (cit. on p. 65).

Payne, A. B. et al. (2013). 'Estimation of mango crop yield using image analysis - Segmentation method'. In: *Computers and Electronics in Agriculture* 91, pp. 57–64. ISSN: 01681699. DOI: 10.1016/j.compag.2012.11.009. URL: http://dx.doi.org/10.1016/j.compag.2012.11.009 (cit. on p. 64).

Pérez-Borrero, Isaac et al. (2020). 'A fast and accurate deep learning method for strawberry instance segmentation'. In: *Computers and Electronics in Agriculture* 178, p. 105736. ISSN: 0168-1699. DOI: https://doi.org/10.1016/j.compag.2020.105736. URL: https://www.sciencedirect.com/science/article/pii/S0168169920300624 (cit. on p. 70).

Piazzolla, Francesca, Maria Luisa Amodio and Giancarlo Colelli (2017). 'Spectra evolution over on-vine holding of Italia table grapes: prediction of maturity and discrimination for harvest times using a Vis-NIR hyperspectral device'. In: *Journal* *of Agricultural Engineering* 48.2, p. 109. ISSN: 2239-6268. DOI: 10.4081/jae.2017.639. URL: http://www.agroengineering.org/index.php/jae/article/view/639 (cit. on p. 76).

Pound, Michael et al. (May 2016). 'Deep Machine Learning provides state- of-the-art performance in image-based plant phenotyping'. In: *GigaScience* 6. DOI: 10.1101/053033 (cit. on pp. 128, 129).

Qureshi, W S et al. (2014). 'Dense segmentation of textured fruits in video sequences'. In: *International Conference on Computer Vision Theory and Applications (VISAPP), 2014* 2, pp. 441–447 (cit. on pp. 60, 62–64, 82, 83, 88).

Qureshi, W. S. et al. (2017). 'Machine vision for counting fruit on mango tree canopies'. In: *Precision Agriculture* 18.2, pp. 224–244. ISSN: 15731618. DOI: 10.1007/s11119-016-9458-5 (cit. on p. 64).

Rahnemoonfar, Maryam and Clay Sheppard (2017). 'Deep Count: Fruit Counting Based on Deep Simulated Learning'. In: *Sensors* 17.4. ISSN: 1424-8220. DOI: 10.3390/s17040905. URL: https://www.mdpi.com/1424-8220/17/4/905 (cit. on p. 9).

Rajendra, Peter et al. (2009). 'Machine Vision Algorithm for Robots to Harvest Strawberries in Tabletop Culture Greenhouses'. In: *Engineering in Agriculture, Environment and Food* 2.1, pp. 24–30. ISSN: 18818366. DOI: 10.1016/S1881-8366(09)80023-2. URL: http://www.aptech.kais.kyoto-u.ac.jp/e/summary/date/sum5%5C%5F5.pdf (cit. on pp. 60, 67, 83).

Rajkumar, P. et al. (2012). 'Studies on banana fruit quality and maturity stages using hyperspectral imaging'. In: *Journal of Food Engineering* 108.1, pp. 194–200. ISSN: 02608774. DOI: 10.1016/j.jfoodeng.2011.05.002. URL: http://linkinghub.elsevier.com/retrieve/pii/S026087741100238X (cit. on p. 76).

Redmon, Joseph, Santosh Divvala et al. (May 2016). 'You Only Look Once: Unified, Real-Time Object Detection'. In: *arXiv:1506.02640 [cs]*. arXiv: 1506.02640. URL: http://arxiv.org/abs/1506.02640 (visited on 1st May 2022) (cit. on p. 42).

Redmon, Joseph and Ali Farhadi (2016). 'YOLO9000: Better, Faster, Stronger'. In: *CoRR* abs/1612.08242. arXiv: 1612.08242. URL: http://arxiv.org/abs/1612.08242 (cit. on p. 49).

– (2018). 'YOLOv3: An Incremental Improvement'. In: *CoRR* abs/1804.02767. arXiv: 1804.02767. URL: http://arxiv.org/abs/1804.02767 (cit. on pp. 44, 51).

Ren, Shaoqing, Kaiming He, Ross B Girshick et al. (2015a). 'Faster {R-CNN:} Towards Real-Time Object Detection with Region Proposal Networks'. In: *CoRR* abs/1506.0 (cit. on p. 98).

– (2015b). 'Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks'. In: *CoRR* abs/1506.01497. arXiv: 1506.01497. URL: http://arxiv.org/abs/1506.01497 (cit. on p. 38).

Roy, Ankush et al. (2011). 'Statistical video tracking of pomegranate fruits'. In: *Proceedings - 2011 3rd National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics, NCVPRIPG 2011*, pp. 227–230. DOI: 10.1109/NCVPRIPG.2011.67 (cit. on pp. 60, 62, 83).

Sa, Inkyu et al. (2016). 'DeepFruits: A Fruit Detection System Using Deep Neural Networks'. In: *Sensors* 16.8, p. 1222. ISSN: 1424-8220. DOI: 10.3390/s16081222. URL: http://www.mdpi.com/1424-8220/16/8/1222 (cit. on pp. 31, 59, 62, 68, 80, 82–84, 92, 95, 99, 100, 103–107, 109).

Santos, Thiago T. et al. (2020). 'Grape detection, segmentation, and tracking using deep neural networks and three-dimensional association'. In: *Computers and Electronics in Agriculture* 170, p. 105247. ISSN: 0168-1699. DOI: https://doi.org/10.1016/j.compag.2020.105247. URL: https://www.sciencedirect.com/science/article/pii/S0168169919315765 (cit. on p. 114).

Scarfe, Alistair John (2012). 'Development of an Autonomous Kiwifruit Harvester'. In: pp. 380–384 (cit. on pp. 60, 67, 83).

Schanda, János (2007). 'CIE Colorimetry'. In: *Colorimetry.* Wiley-Blackwell. Chap. 3, pp. 25–78. ISBN: 9780470175637. DOI: 10.1002/9780470175637.ch3. URL: https://onlinelibrary.wiley.com/doi/abs/10.1002/9780470175637.ch3 (cit. on p. 93).

Si, Yongsheng, Gang Liu and Juan Feng (2015). 'Location of apples in trees using stereoscopic vision'. In: *Computers and Electronics in Agriculture* 112, pp. 68–74. ISSN: 01681699. DOI: 10.1016/j.compag.2015.01.010. URL: http://dx.doi.org/10.1016/j.compag.2015.01.010 (cit. on pp. 59, 65).

Simonyan, Karen and Andrew Zisserman (Apr. 2015). 'Very Deep Convolutional Networks for Large-Scale Image Recognition'. In: *arXiv:1409.1556 [cs]*. arXiv: 1409.1556. URL: http://arxiv.org/abs/1409.1556 (visited on 24th Apr. 2022) (cit. on p. 37).

Snell, Jake, Kevin Swersky and Richard S. Zemel (June 2017). 'Prototypical Networks for Few-shot Learning'. In: *arXiv:1703.05175 [cs, stat]*. arXiv: 1703.05175. URL: http://arxiv.org/abs/1703.05175 (visited on 3rd May 2022) (cit. on p. 50).

Song, Y. et al. (2014). 'Automatic fruit recognition and counting from multiple images'. In: *Biosystems Engineering* 118.1, pp. 203–215. ISSN: 15375110. DOI: 10.1016/j.biosystemseng.2013.12.008. URL: http://dx.doi.org/10.1016/j.biosystemseng.2013.12.008 (cit. on pp. 60, 62, 63, 82, 83).

Stajnko, D., M. Lakota and M. Hoevar (2004). 'Estimation of number and diameter of apple fruits in an orchard during the growing season by thermal imaging'. In: *Computers and Electronics in Agriculture* 42.1, pp. 31–42. ISSN: 01681699. DOI: 10.1016/S0168-1699(03)00086-3 (cit. on p. 64).

Sun, Li, Jian-Rong Cai and Jie-Wen Zhao (2015). 'A Vision System Based on TOF 3d Imaging Technology Applied to Robotic Citrus Harvesting'. In: *Intelligent*

*Automation & Soft Computing* 21.3, pp. 345–354. ISSN: 1079-8587. DOI: 10.1080/10798587.2015.1015767. URL: http://www.tandfonline.com/doi/full/10.1080/10798587.2015.1015767 (cit. on pp. 60, 66, 83).

Tallada, Jasper G., Masateru Nagata and Taichi Kobayashi (2006). 'Non-destructive estimation of firmness of strawberries (Fragaria x ananassa Duch.) using NIR hyperspectral imaging'. In: *Environmental and Control Biology* 44.4, pp. 245–255. ISSN: 1880554X (cit. on p. 75).

Teimouri, Nima et al. (2014). 'A novel artificial neural networks assisted segmentation algorithm for discriminating almond nut and shell from background and shadow'. In: *Computers and Electronics in Agriculture* 105, pp. 34–43. ISSN: 01681699. DOI: 10.1016/j.compag.2014.04.008. URL: http://dx.doi.org/10.1016/j.compag.2014.04.008 (cit. on p. 91).

Teixidó, Mercè et al. (2012). 'Definition of linear color models in the RGB vector color space to detect red peaches in orchard images taken under natural illumination'. In: *Sensors (Switzerland)* 12.6, pp. 7701–7718. ISSN: 14248220. DOI: 10.3390/s120607701 (cit. on p. 59).

Tian, Hongkun et al. (2020). 'Computer vision technology in agricultural automation —A review'. In: *Information Processing in Agriculture* 7.1, pp. 1–19. ISSN: 2214-3173. DOI: https://doi.org/10.1016/j.inpa.2019.09.006. URL: https://www.sciencedirect.com/science/article/pii/S2214317319301751 (cit. on pp. 1, 5).

Tripathi, Mukesh Kumar and Dhananjay D. Maktedar (2020). 'A role of computer vision in fruits and vegetables among various horticulture products of agriculture fields: A survey'. In: *Information Processing in Agriculture* 7.2, pp. 183–203. ISSN: 2214-3173. DOI: https://doi.org/10.1016/j.inpa.2019.07.003. URL: https://www.sciencedirect.com/science/article/pii/S2214317318303834 (cit. on p. 10).

*TuberScan* (2022). en. URL: https://b-hiveinnovations.co.uk/projects/tuberscan (visited on 6th Jan. 2022) (cit. on p. 1).

Uijlings, Jasper RR et al. (2013). 'Selective search for object recognition'. In: *International journal of computer vision* 104.2, pp. 154–171 (cit. on p. 38).

Vázquez-Arellano, Manuel et al. (2016). '3-D Imaging Systems for Agricultural Applications—A Review'. In: *Sensors* 16.5. ISSN: 1424-8220. DOI: 10.3390/s16050618 (cit. on p. 73).

Wagner, Nikolaus et al. (2021). 'Efficient and Robust Orientation Estimation of Strawberries for Fruit Picking Applications'. In: (cit. on p. 15).

Williams, Henry A.M. et al. (2019). 'Robotic kiwifruit harvesting using machine vision, convolutional neural networks, and robotic arms'. In: *Biosystems Engineering* 181, pp. 140–156. ISSN: 1537-5110. DOI: https://doi.org/10.1016/j.biosystemseng.

2019.03.007. URL: https://www.sciencedirect.com/science/article/pii/S153751101830638X (cit. on p. 9).

Wilson, Andy (Oct. 2017). 'Fast Lossless Depth Image Compression'. In: *Proceedings of the 2017 ACM International Conference on Interactive Surfaces and Spaces (ISS '17).* ACM. URL: https://www.microsoft.com/en-us/research/publication/fast-lossless-depth-image-compression/ (cit. on p. 147).

Wojke, Nicolai, Alex Bewley and Dietrich Paulus (2017). 'Simple Online and Realtime Tracking with a Deep Association Metric'. In: *CoRR* abs/1703.07402. arXiv: 1703.07402. URL: http://arxiv.org/abs/1703.07402 (cit. on pp. 13, 51, 56, 112, 115, 116, 118, 122).

Xiong, Ya et al. (2019). 'Development and field evaluation of a strawberry harvesting robot with a cable-driven gripper'. In: *Computers and Electronics in Agriculture* 157, pp. 392–402. ISSN: 0168-1699. DOI: https://doi.org/10.1016/j.compag.2019.01.009 (cit. on pp. 73, 128).

Yang, Ce, Won Suk Lee and Paul Gader (2014). 'Hyperspectral band selection for detecting different blueberry fruit maturity stages'. In: *Computers and Electronics in Agriculture* 109, pp. 23–31. ISSN: 01681699. DOI: 10.1016/j.compag.2014.08.009. URL: http://dx.doi.org/10.1016/j.compag.2014.08.009 (cit. on pp. 75, 76).

Yeh, Yu Hui et al. (2016). 'Strawberry foliar anthracnose assessment by hyperspectral imaging'. In: *Computers and Electronics in Agriculture* 122, pp. 1–9. ISSN: 01681699. DOI: 10.1016/j.compag.2016.01.012. URL: http://dx.doi.org/10.1016/j.compag.2016.01.012 (cit. on p. 76).

Yu, Yang et al. (2019). 'Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN'. In: *Computers and Electronics in Agriculture* 163, p. 104846. ISSN: 0168-1699. DOI: https://doi.org/10.1016/j.compag.2019.06.001. URL: http://www.sciencedirect.com/science/article/pii/S0168169919301103 (cit. on pp. 70, 104).

Zhang, Baohua et al. (2015). 'Computer vision recognition of stem and calyx in apples using near-infrared linear-array structured light and 3D reconstruction'. In: *Biosystems Engineering* 139, pp. 25–34. ISSN: 15375110. DOI: 10.1016/j.biosystemseng.2015.07.011. URL: http://dx.doi.org/10.1016/j.biosystemseng.2015.07.011 (cit. on p. 62).

Zhang, Hui et al. (2020). 'Mask SSD: An Effective Single-Stage Approach to Object Instance Segmentation'. In: *IEEE Transactions on Image Processing* 29, pp. 2078–2093. DOI: 10.1109/TIP.2019.2947806 (cit. on p. 49).

Zhang, Wenli et al. (Feb. 2022). 'Deep-learning-based in-field citrus fruit detection and tracking'. In: *Horticulture Research* 9. uhac003. ISSN: 2052-7276. DOI: 10.1093/hr/uhac003. eprint: https://academic.oup.com/hr/article-pdf/doi/10.1093/hr/uhac003/43708799/uhac003.pdf. URL: https://doi.org/10.1093/hr/uhac003 (cit. on p. 72).

Zhou, Xue et al. (2021). ‘Strawberry Maturity Classification from UAV and Near-Ground Imaging Using Deep Learning’. In: *Smart Agricultural Technology* 1, p. 100001. ISSN: 2772-3755. DOI: https://doi.org/10.1016/j.atech.2021.100001. URL: https://www.sciencedirect.com/science/article/pii/S2772375521000010 (cit. on pp. 68, 69).

Zitnick, C. Lawrence and Piotr Dollár (2014). ‘Edge Boxes: Locating Object Proposals from Edges’. In: *Computer Vision – ECCV 2014*. Ed. by David Fleet et al. Cham: Springer International Publishing, pp. 391–405. ISBN: 978-3-319-10602-1 (cit. on p. 38).