IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 18 June 2024, accepted 16 July 2024, date of publication 19 July 2024, date of current version 1 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3431244

RESEARCH ARTICLE

An In-Field Dynamic Vision-Based Analysis for Vineyard Yield Estimation

DAVID AHMEDT-ARISTIZABAL[1], DANIEL SMITH[2], MUHAMMAD RIZWAN KHOKHER[1], XUN LI[1], ADAM L. SMITH[3], LARS PETERSSON[1], VIVIEN ROLLAND[3], AND EVERARD J. EDWARDS[3]

[1]Imaging and Computer Vision Group, Data61, Commonwealth Scientific and Industrial Research Organisation (CSIRO), Canberra, ACT 2601, Australia
[2]Spatio-Temporal Analytics Team, Data61, Commonwealth Scientific and Industrial Research Organisation (CSIRO), Canberra, ACT 2601, Australia
[3]Department of Agriculture and Food, Commonwealth Scientific and Industrial Research Organisation (CSIRO), Canberra, ACT 2601, Australia

Corresponding author: David Ahmedt-Aristizabal (david.ahmedtaristizabal@csiro.au)

This work was supported in part by the Department of Agriculture, Water and the Environment through their Rural Research and Development for Profit Scheme and Wine Australia; and in part by the CSIRO Machine Learning and Artificial Intelligence Future Science Platform.

ABSTRACT Accurately predicting grape yield in vineyards is essential for strategic decision-making in the wine industry. Current methods are labour-intensive, costly, and lack spatial coverage, reducing accuracy and cost-efficiency. Efforts to automate and enhance yield estimation focus on scaling fruit weight assessments. Machine learning, particularly deep learning, shows promise in improving accuracy through automatic feature extraction and hierarchical representation. However, most of these methods have been developed for analyses at a particular time point and solutions able to consider temporal information captured across sequential frames are currently poorly developed. This paper addresses this gap by introducing a system for yield estimation, utilising publicly available data repositories, such as Embrapa WGISD, alongside an in-house dataset collected by a Blackmagic camera at the pre-harvest stage. We introduce a system that utilises bunch weight regression to estimate grape yield. Bunch weight estimates are obtained by summing samples randomly drawn from the grape bunch weight distribution through empirical calibration. Grapevine bunches are identified and segmented using Mask R-CNN with Swin Transformer, and a SiamFC-based tracking mechanism is employed to estimate the number of unique bunches per panel or row. The number of berries for each tracked bunch is determined using a density approach known as multitask point supervision. In our experiments, we demonstrate the effectiveness of the proposed system for yield estimation, achieving harvested weight errors of less than 5% in two of the three vineyard panels. Larger harvest weight errors (around 15%) were observed due to inaccuracies in tracking certain bunches caused by dense concentration of bunches in one panel. However, these errors should be contrasted with the current practice error of up to 30%, highlighting the potential of machine vision for hands-off yield estimation at scale.

INDEX TERMS Precision viticulture, bunch detection and segmentation with transformers, multi-bunch tracking and counting, density-based berry counting, weight regression, grapevine yield estimation.

I. INTRODUCTION

Yield estimation is an essential tool in horticulture, being used throughout the supply chain for everything from harvest, storage and processing logistics to product pricing and advertising. This is particularly critical for the wine industry due to the additional logistical complexities embedded in the winemaking process, as fruit cannot be stored prior to winemaking and fruit quality degrades if left on the vine. Further, even once fermentation is complete, maturation and storage require advance planning. Winemaking typically multiples the farm-gate value of the fruit three to four folds, demonstrating the critical nature of the planning around this process. Yield estimation in viticulture is heavily dependent

The associate editor coordinating the review of this manuscript and approving it for publication was Amin Zehtabian.

on the vineyard manager's experience, often being simply a visual assessment at various points in the growing season. Where formal yield estimation is undertaken, it is usually based on counting grape bunches in a very small subset of the vineyard, possibly combined with a small number of bunch weight measures. Consequently, yield estimation is labour-intensive and, therefore, expensive. This can result in significant error, potentially causing over- or under-supply of grapes to wineries and affecting the pricing of wine already in the market. Grape yield is strongly seasonal, often due to weather conditions during the growing season and bud formation in the prior season. This offers the potential to use modelling and forecasting of climate extremes to predict yield and potentially mitigate seasonal effects on grape yield through management changes [1].

Accurate yield predictions enable wineries to plan intake and make adjustments throughout the value chain accordingly. Even 20 years ago, the cost of best practice yield estimation to the industry was estimated as AU$16 per hectare [2], which is further amplified by high error rates. The same authors estimated, at the time, that a reduction in yield estimate error from 33% to 20% could save the industry around AU$85 million annually, and further reductions would result in even greater savings.

Grapevine yield can be divided into three main components: i) the average berry weight, ii) the average number of berries per bunch, and iii) the number of bunches within a given vineyard area. Berry weight will continue to change until harvest, while the maximum berry number is determined early in the season and the maximum bunch number is determined during bud formation and influenced by pruning [3]. These components can be assessed at different times, allowing for early yield estimates, but accuracy improves over time. However, traditional assessment methods are often manual and limited in scope.

The automation of yield prediction is a central challenge in smart agriculture. Some methods rely on classical image processing techniques, which involve developing segmentation, shape recognition, and feature extraction algorithms tailored to the task. There is a growing trend towards incorporating computer vision algorithms and machine learning to assist with these measurements. This approach utilises available data instead of subjective criteria and specialised algorithms. These algorithms have the potential to improve model accuracy by leveraging the wealth of gathered information, and their integration into data-driven algorithms holds promising prospects for enhancing overall analysis [4]. As a result, researchers are exploring precision viticulture and automated yield estimation methods [5], [6], [7], especially with the advent of modern deep learning approaches [8]. A set of typical viticulture practices has already benefited from the technical advances in computer vision and deep learning. This includes the detection of basic parts of the vine (shoots, canes, etc.), the detection of the structural elements of a vineyard, and supplementary detection sub-tasks that complement the basic practices [9]. Research efforts focused on yield estimation in viticulture are summarised in [10]. Specifically, the utilisation of deep learning is highlighted in [11] and [12]. In Section II, we provide an overview of current approaches for vision-based yield estimation in viticulture. This review section provides an overview of recent advancements in the field, focusing on work published from 2020 and onwards.

The literature highlights weaknesses, particularly the lack of focus on dynamic analysis considering temporal information across sequential frames, with most works concentrating on static analysis. This study contributes to this area by introducing algorithms for grapevine berry analysis, specifically targeting pre-harvest yield estimation from videos by estimating bunch mass alongside bunch counts. We employ a transformer-based instance segmentation approach to identify and segment grapevine bunches while employing a multi-object tracking-by-association method for real-time bunch tracking. Berry detection is achieved through a density estimation-based counting method. Additionally, a regression model is trained to represent the relationship between segmented grape bunches' features and their true weights. The proposed framework is validated through experiments conducted on an in-house dataset captured in an Australian vineyard using a Blackmagic camera attached to a *Kubota* RTV. Our approach enables fruit yield from video data without manual adjustments or additional data collection.

Detection, segmentation, tracking and weight estimation of fruits and vegetables are fundamental tasks not only for grapevine analysis but also for precision agriculture more broadly, facilitating yield estimation across various crops. The versatility of the proposed system lies in its ability to address common challenges in agricultural settings, including the uneven distribution of produce in the field, variations in illumination, occlusion caused by foliage or neighbouring fruits, and clutter from surrounding vegetation.

Our main contributions are summarised as follows:

1) Provided a summary of the most relevant recent research works concerning the adoption of deep learning techniques for bunch and berry detection and counting which are critical for yield estimation. Case studies reported in this paper are obtained from various journals, conference proceedings and open-access repositories published in English between 2020 and early 2024.
2) Designed and constructed a cost-effective image acquisition system using consumer-grade cameras, enabling researchers to capture sequential images for grapevine berries analysis.
3) Gathered a novel dataset comprising RGB videos captured during the pre-harvest stage. The videos were recorded in an undisturbed natural setting, without alterations to the background. Ground truth for object detection and segmentation in computer vision was

established through meticulous annotation, marking precise bounding boxes and masks around bunches and berries following established standards.

4) Employed a dynamic analysis that considers temporal information captured across sequential frames. This processing pipeline includes bunch detection, bunch segmentation, bunch tracking, berry detection, and bunch weight estimation using a regression model composed of segmented bunch features. The accuracy of the panel-wise counting results was assessed against the ground-truth counts, ultimately leading to a promising yield estimation.

II. RELATED WORKS: DEEP LEARNING FOR VITICULTURE YIELD ESTIMATION

Current vineyard yield estimation methods are limited by their labour-intensive and costly nature, resulting in inaccuracies and biases. However, automation through proximal sensing and mobile platforms offers non-invasive methods capable of covering larger vineyard areas, thereby enhancing accuracy. This advancement relies on the development of data-driven algorithms [13].

To date, automated yield estimation has focused on detecting grape bunches and determining both their count and the number of berries within each bunch. However, this task poses challenges for computer vision systems due to significant variations in fruit sizes, shapes, and colours, along with high occlusion rates and varying illumination conditions. Challenges also arise from neighbouring bunch separation, occlusions from leaves and shoots, and colour confusion between green grapes and foliage. Detecting individual berries within a bunch in a commercial vineyard setup is particularly challenging due to the lighting conditions and significant berry occlusion. Moreover, relying solely on berry count for yield estimation is prone to errors, given the variability in berry size influenced by grape variety, growing conditions, and yield fluctuations within the same vineyard. Despite these limitations, current practices rely on these measures as they represent the primary sole extractable information from visual (RGB image) data alone.

Previous studies have highlighted limitations in feature engineering and traditional machine learning methods, paving the way for advancements in deep learning architectures. Deep learning approaches have gained popularity in agricultural applications, due to their ability to address image-based perceptual challenges [14]. These methodologies have been successfully employed for tasks such as bunch and berry detection, segmentation, tracking, and counting, crucial for accurate yield estimates:

Yield estimation through bunch detection and counting involves capturing images using movable platforms and processing them to identify bunches. Object detection techniques identify bunches and extract features which are then used to regress bunch weights. Total yield is computed by aggregating estimated bunch weights [15].

Yield estimation through berry detection and counting requires a model to find individual berries in images, estimate their count, and determine total yield by summing the berry count across all images [16].

Deep learning approaches for bunch and berry analysis, summarised in Table 1 and Table 2, encompass various methods for vision-based detection and counting. These methods include object detection, instance segmentation, and semantic segmentation, applied to both bunches and berries. Additionally, density estimation methods are employed specifically for berry analysis. This section provides a comprehensive overview of deep learning-based approaches for bunch and berry detection and tracking.

A. GRAPEVINE BUNCH AND BERRY DETECTION AND COUNTING

Two primary schemes are commonly employed to build berry counting networks. The first focuses on explicit object localisation, where objects are detected before counting. This can involve identifying object centres, resulting in a density estimation heat map (a two-dimensional representation indicating areas with a high probability of containing objects). Alternatively, localisation may rely on bounding box detection or segmentation, with the latter being more widely adopted. The second scheme involves utilising a global or local direct regression model. Such models adopted for bunch detection [49], [64] are not discussed further in this section.

This section focuses on data-driven methods for grape bunch detection and segmentation, emphasising modern deep learning techniques [65]. While classical image processing and traditional machine learning have been used in viticulture applications, they are limited by the necessity for careful algorithm selection for feature extraction, shape detection, and classification, as well as the requirement for partial control of the environment with artificial backgrounds or lighting. Methods reliant on handcrafted feature engineering and rule-based algorithms [66], [67], [68] are excluded from our discussion, as they have been extensively reviewed in previous survey manuscripts [4], [5], [14]. These excluded techniques encompass segmentation using thresholds (colour-based segmentation), edge-based segmentation, contour analysis, texture and shape analysis, and traditional image processing methods like Otsu's threshold [13] or the identification of the Green-Red Vegetation Index [69].

For berry detection, we exclude thresholding and colour-based methods [18] that rely on specific colour thresholds to distinguish berries from the background. These techniques utilise colour information such as hue, saturation, and intensity for segmenting berries based on their distinct colour characteristics. We also exclude shape-based methods that focus on extracting and analysing berry shape characteristics, including contour analysis, circular Hough transform, or ellipse fitting. Although 3D information from reconstruction can enhance detection and segmentation when combined with 2D techniques, such as colour-based or

TABLE 1. Deep learning-based approaches for grapevine bunch detection.

Approach	Backbone	Frameworks	Author
Object Detection	Darknet-53, Uniformer+BiPANet, CSPNet, GhostNet, Swin Transformer, SAM-CSPDarkNet	YOLO series (*e.g.* YOLOv3, YOLOv4, YOLOv5, YOLOv7, YOLOv8, YOLOR, YOLO-GP)	[17]–[30]
	MobileNet-V1, Inception-V2, ResNet50	SSD	[31], [32]
	ResNet50	Faster-RCNN	[32], [33]
Semantic Segmentation	ResNet50+CBAM	PSPNet	[34], [35]
	ResNet50, ResNet50+CBAM	DeepLabV3+	[34], [36]–[38]
	ResNet50	FCN	[36]
	ResNet50, ResNet50+CBAM	U-Net	[34], [36]
	ResNet18, ResNet101, VGG19	CNN	[39], [40]
Instance Segmentation	Swin Transformer	Mask2Former	[28]
	ResNet50, ResNet101	HTC	[28]
	ResNet50, ResNet101, Swin Transformer	RCNN series (Mask-RCNN, Cascade Mask-RCNN)	[17], [28], [29], [41]–[44]
	ResNet101	YOLOACT	[44]

TABLE 2. Deep learning-based approaches for berry detection.

Approach	Backbone	Frameworks	Author
Object Detection	Darknet-53, CSDenseNet, E-ELAN	YOLO series (*e.g.* YOLOv3, YOLOv5, YOLOv7)	[45]–[48]
	RetinaNet	CNN	[49]
Semantic Segmentation	VGG16	SegNet	[50]–[52]
	UNet-PatchGAN	Pix2Pix	[53]
	MobileNetV2	DeepLabV3+	[54], [55]
	VGG16	FCN	[56]
Instance Segmentation	Swin Transformer-Tiny+ASFF	Two-stage (RPN + FCN)	[57]
	ResNet50, ResNet101, HRNet, Swin Transformer-Tiny	RCNN series (Mask-RCNN, Mask Scoring RCNN, Cascade Mask-RCNN)	[35], [57]–[61]
	ResNet101	SOLO	[60]
	ResNet101	YOLOACT	[60]
Density map estimation	VGG16	CDMENet	[62]
	VGG16	CSRNet	[63]

edge-based segmentation, none of these handcrafted methods are included in this review section [70], [71].

1) OBJECT DETECTION APPROACHES

Object detection is a challenging problem that requires the solution of two main tasks: recognition and localisation. Recent years have witnessed remarkable performance gains in object detection, attributable to advancements in deep convolutional neural networks. These detectors have been useful for the identification of key areas in each video such as spurs, grape bunches, and berries [9].

Object detectors can be categorised as *anchor-based*, *anchor-free* and *Transformer-based* [72]. The core idea of anchor-based models is to introduce a constant set of bounding boxes, referred to as *anchors*, which can be viewed as a set of pre-defined proposals for bounding box regression. Notable examples of anchor-based models include the RCNN [73], YOLO [74], and SSD [75] series of detectors. Anchor-free methods offer significant promise to cope with extreme variations in object scales and aspect ratios [76]. Such approaches, for example, can perform object bounding box regression based on anchor points instead of boxes (*i.e.,* the object detection is reformulated as a key-point localisation problem) (key-point-based and anchor-point-based methods). Such methods have not been explored in the domain of viticulture. Object detection algorithms based on the Transformer architecture can capture long-range dependencies over the object to extract useful global information. The use of Transformers has become a hot research direction within object detection, with prominent architectures based upon the DETR [77] and ViT [78] series.

The different approaches used for grape bunches and berry detection are outlined in Table 1 and Table 2. Among these, the YOLO series and SSD series stand out as prevalent methodologies, frequently employed for providing bounding boxes around the detected bunch and berry [19], [31], [39], [45], [46], [58]. While these object detection approaches are widely used, they have inherent limitations. The bounding box representation may not precisely outline the complex shapes of bunches or the contours of individual berries. This limitation results in a reduction in spatial precision, especially when objects are nearby or exhibit irregular shapes. In addition, the challenges become more pronounced when bunches or berries overlap, as object detectors struggle to separate and represent individual instances, leading to errors in counting and localisation.

2) SEMANTIC SEGMENTATION APPROACHES

Semantic segmentation assigns a class label to each pixel in an image, enabling pixel-wise identification of bunches and berries. By segmenting the image into, for example, bunch and non-bunch regions, this approach offers a fine-grained understanding of the spatial distribution of bunches. Traditional deep learning series such as U-Net [79] and Fully Convolutional Network (FCN) have been successfully applied to the semantic segmentation of vineyard images. FCNs were among the early methods for semantic segmentation, using convolutional layers to predict pixel-wise labels. The U-Net architecture consists of a contracting path, a bottleneck, and an expansive path. SegNet [80], another model adopted, uses an encoder-decoder architecture with pooling indices during the encoding phase and up-sampling during the decoding phase. Other popular approaches for vineyards analysis are the DeepLab [81] series which employs dilated convolutions and spatial pyramid pooling to capture multi-scale contextual information to bring improvements to the existing encoder-decoder architectures. Recently, the PSPNet [82] was adopted for bunch analysis using a pyramid pooling module to capture contextual information at different scales.

The main limitation of semantic segmentation models is their inability to separate closed or overlapping bunches. Small objects or objects with fine details may be challenging to segment accurately or merge with the background, resulting in inaccurate grape counting. Such approaches were adopted for bunch [34], [36] and berries segmentation [51], [54], [55], [56] as listed in Table 1 and Table 2.

3) INSTANCE SEGMENTATION APPROACHES

Unlike semantic segmentation, instance segmentation is tasked with distinguishing between individual instances of the same class, requiring the incorporation of mask prediction branches alongside bounding box and class prediction branches. Traditional two-stage detector approaches are commonly employed to tackle instance segmentation problems, involving two main stages: region proposal generation and instance mask prediction.

Prominent methods follow the two-stage approach including the RCNN series, Mask-RCNN [83] and cascade-RCNN [84], and hybrid task cascade (HTC) [85]. These methods are widespread adopted in both bunch [17], [28], [41], [42] and berry [35], [57], [58], [59] instance segmentation. Other methods such as Fully Convolutional Instance Segmentation (FCIS) [86] and PointRend [87] exemplify the effectiveness of the two-stage approach in accurately delineating object instances within an image. While two-stage methods may achieve higher accuracy, they can be computationally more demanding, prioritising precision. In contrast, one-stage methods, operating end-to-end, typically feature simpler architectures. Noteworthy examples of one-stage instance segmentation methods, chosen for their efficiency in berry detection, include the SOLO series and YOLO with a mask head (YOLOACT) [60].

It is worth recognising that variations in object detection, semantic segmentation, and instance segmentation models often stem from modifications to the primary components of the standard architecture. Key components subject to alteration include: i) The *backbone*, serving as the feature extractor, commonly involves feed-forward CNNs or residual networks, recently transformer-based backbones in the viticulture field. ii) The *neck*, additional layers positioned between the backbone and the head, designed to extract neighbouring feature maps from various stages of the backbone. Commonly, a neck comprises several bottom-up and top-down paths, facilitating the conveyance of enriched information to the head. Examples of the neck include the feature pyramid network (FPN) [88] and the path aggregation network (PANet) [89]. iii) The *head*, the network responsible for detection (classification and regression).

4) DENSITY MAP ESTIMATION APPROACHES

Techniques developed in the context of automatic crowd counting (people and vehicles) can be adapted to the scenario of berry detection due to their capability to solve counting problems in a highly congested scene. Detection-based methods discussed previously have been used for crowd counting, which may perform accurate detection in sparse scenes, however, with occlusion and extremely dense crowds, their performance is unsatisfactory. Regression techniques such as linear regression are used to learn a mapping function to the crowd counting, they, however, ignore spatial location information of the targets. Density map estimation methods, on the other hand, provide a continuous representation of the spatial distribution and density of berries within an image or a video frame [90]. Instead of using bounding boxes or segmentation masks, density maps assign a density value to each pixel, indicating the likelihood of a berry's presence in that location. To detect berries using density maps, a threshold can be applied to distinguish areas with a significant berry presence from background noise. By selecting an appropriate threshold, regions of interest can be identified where the berry density surpasses a certain threshold value. Density maps can facilitate accurate berry counting by summing up the density values across the entire map or within specific regions of interest.

Density map estimation uses weak labels (dot annotations on object centres), which are less tedious to annotate than bounding boxes or segmentation masks. Density estimation-based counting was first proposed in [91] which demonstrated higher counting accuracy with smaller amounts of data, particularly when individual object instances are challenging to detect or delineate. A traditional deep learning method known as Multi-column CNN (MCNN) [92] employs multiple columns (streams) of CNNs operating at different scales to capture objects of varying size, enabling the model to adapt to the density distribution in the image. However, the fixed configuration of multiple scales may not be optimal for all scenarios, and adapting to scale variations can still be challenging. Another popular approach is the convolutional sparse regression network (CSRNet) [93] designed to represent the relationship between image features and object density using sparse regression. Such a CSRNet model was adopted for berry counting [63]. Although density estimation approaches are specifically designed for counting objects, they do not provide instance-level information and the performance may vary on the complexity of scenes and the distribution of objects.

B. MULTI-OBJECT TRACKING

Visual trackers are used to maintain the unique identity of each grape bunch as it moves across the frames of a video. The raw detection of grape bunches across sequential video frames would greatly overestimate the true number of bunches, given each specific bunch would be detected multiple times. To mitigate this issue, a tracking algorithm is utilised to link the bounding box detections of a single grape bunch across video frames, forming a sequence referred to as a ''bunch track''. The number of tracks obtained using this algorithm is then used to estimate the grape bunch count. Given a detected grape bunch, by identifying 2D features belonging to it and matching, or triangulating, them across multiple frames, it is possible to find the same bunch instance in the following frames.

Following recent advancements in object detection, ''tracking-by-detection'' has emerged as the predominant approach for multiple object tracking. This approach consists of two distinct components: object detection and data association. The detection component focuses on identifying potential targets of interest within video frames, thereby guiding the tracking process. The data association component utilises both geometric and visual information to assign these detections to new or existing object trajectories, commonly referred to in the literature as ''tracklets''. An example of this task is the re-identification process [94]. A tracklet is defined as a set of linked regions across consecutive frames [95]. In numerous implementations, the data association computes the similarity between detections and existing tracklets, determines the optimal associations between detections and tracklets, and generates new tracklets as necessary.

Multi-object tracking approaches can be categorised based on their complexity into separate detection and embedding (SDE) methods and joint detection and embedding (JDE) algorithms. SDE methods involve distinct stages for detection and embedding extraction. This design facilitates adaptation to various detectors with fewer changes, as the two components can be fine-tuned independently; popular instances of SDE include DeepSORT [96] and ByteTrack [97]. On the other hand, JDE methods learn to detect objects and extract embeddings through a shared neural network. JDE methods such as Tracktor [98] and FairMOT [99] leverage multi-task learning to train the network effectively.

For bunch tracking, SDE methods are the predominant methods. They are illustrated in Table 3. However, these methods are not widely considered, as the majority of the works focus on purely static analysis (single images). Failure of the tracker to generate a predicted object location indicates one of two cases: the object has left the field of view, or the object has become untrackable due to occlusion by another object (leaf, post). Detected objects were tracked using Kernelized Correlation Filters to prevent double counting as well as a Hungarian algorithm to pair new object detections with existing trackers [32]. The feature matching of surf features and geometric verification using RANSAC was adopted by [29]. Other detection-based trackers such as DeepSORT designed to work in real-time using a deep association metric were validated for multi-object tracking of the grape bunch instances [29]. Regarding berry tracking, a time-lapse tracking over successive segmented berries in a control conditional was explored with tree combined independent methods: Baseline (matching of berry centre coordinates), Registration (estimating global bunch deformation before matching) and Matching Tree (processing the time steps in an optimal order) [100].

An alternative perspective in object tracking that leverages three-dimensional (3D) information, particularly through techniques like structure from motion (SfM), has gained increasing significance. A notable application of this approach is in the counting of grapes from videos, where an SfM method is employed to estimate the 3D positions along the camera path. This 3D information serves as a unique identifier, preventing the inadvertent counting of a single grape multiple times. Some studies [17], [29] have exploited a SfM software application, namely COLMAP [101]. In essence, COLMAP extracts sparse features from each frame and conducts a sequential all-versus-all search and matching of these features extracted from the video. These correspondences are then employed to triangulate the 3D points by minimising the 3D-to-2D re-projection error. However, it is noteworthy that, even with the sparse setting, the computational costs exhibit exponential growth with an increase in the number of frames. Existing SfM methods to reconstruct a 3D point cloud of a grape bunch have been exploited for berry counting [102].

Approaches that exploit stereo cameras or depth cameras for 3D reconstruction of detected objects and vineyard SLAM are not considered in this review [58].

C. GENERAL OBSERVATIONS REGARDING YIELD ESTIMATION

To estimate the weight of grapes with computer vision, the relationship between the computed features and the ground truth weight of grapes must first be identified. This relationship is usually established under ideal conditions, independent of whether the features being used are detected bunches, berry counts or classified grape pixels. A strong correlation between counted berries and grape weight was found in [103] confirming its potential value for yield prediction. Counting berries can be difficult, particularly closer to harvest, as bunches become more compact (berries touch each other). Furthermore, it is common for berries to not be visible as a result of bunch occlusion on the vine. [71] showed similar yield estimation performance before and after ripening.

These methods were evaluated in controlled conditions, *i.e.* laboratories, or in the field with artificial backgrounds,

TABLE 3. Representation of approaches for bunch tracking and counting.

Approach	Detector	Tracker/Feature matching	Author
SDE methods	YOLOv5	SURF + RANSAC	[29]
	YOLOv5	DeepSORT	[29]
	YOLOv5	SORT (Kalman filter + Hungarian algorithm)	[30]
	Faster-RCNN	KFC + Hungarian algorithm	[32]
3D-based methods	Mask-RCNN, YOLOv5	SfM: COLMAP	[17], [29]

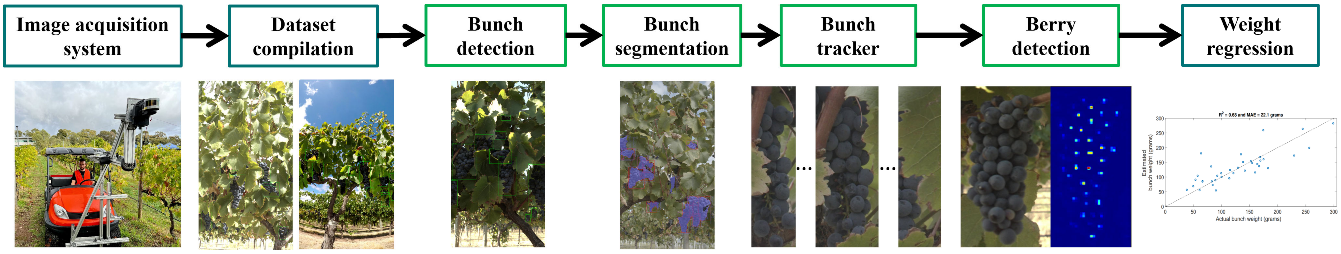


FIGURE 1. Overview of the bunch analysis workflow. 1. Design and construction of the hardware for the data collection. 2. Dataset processing and annotation per panel. 3. The algorithm identifies grapevine bunches throughout all frames of the video and represents them by enclosing them with bounding boxes. 4. The algorithm performs segmentation on each detected bunch within its corresponding bounding box, accurately delineating its precise boundaries and shape. 5. The algorithm tracks bunches per panel, providing the count of uniquely detected bunches. 6. The algorithm calculates the estimated quantity of berries for each of the tracked bunches that are visible in the panel. 7. The algorithm utilises a regression model to estimate the weight of each bunch by establishing a relationship between the extracted features and the actual weight.

which are not representative of the vineyard environment. A problem arises from deploying berry detection algorithms in the field. To obtain berry counts that are sufficiently representative for yield estimation, detectors will usually need to be employed with a relatively wide spatial coverage (*i.e.* multiple rows of vines) across a vineyard. This is mostly easily accomplished with the collection of video, however, the application of a berry detector to sequences of images will result in berries being multi-counted. Therefore, simplifying assumptions have been applied to avoid redundancy during counting. For instance, it is assumed that there is a uniform distribution of grape bunches on the vine. This can be a poor assumption, however, given bunches are typically distributed with a high degree of heterogeneity across a vine.

Computer vision networks need to be trained to be robust to changes in vineyard conditions, whether that is variations in the vine canopy appearance, weather or lighting. Models that can adapt to a wide range of possible vineyard conditions are crucial for real-world applications [43]. Furthermore, developing models that can generalise well to various species of grapes without the need for extensive retraining could enhance the scalability and applicability of the technology.

III. PROPOSED METHOD

In this paper, we propose a system that employs bunch weight regression as a means to estimate grape yield. The system takes into consideration two crucial factors: the number of grape bunches and the berry count per bunch within the selected area being imaged. Our approach for pre-harvest yield estimation builds upon previous research that aimed to estimate bunch size using computer vision techniques. However, instead of relying on the traditional method of counting bunches and using historical average weights, we adopt a different principle. Our system directly estimates bunch weight through empirical calibration. The system workflow is illustrated in Figure 1.

The high-quality 2D images of the vineyard collected with the acquisition system are passed to a pre-processing phase that involves generating a new dataset for grape analysis by utilising both the acquired images and supportive publicly available datasets. This combined dataset enhances the performance of our analysis. The pipeline then progresses through several stages. Initially, grapevine bunches are detected across all video frames, resulting in bounding boxes. Subsequently, each bunch is segmented within its corresponding bounding box, enabling precise boundary determination and shape characterisation. Geometric features extracted from segmented bunches can be used for bunch weight estimation analysis (Section IV-A). To estimate the number of unique bunches per panel or row, we implement a tracking mechanism that monitors the movement of each identified bunch throughout video frames. This allows for accurate bunch counting (Section IV-B). Moreover, we estimate the number of berries for each tracked bunch through the utilisation of density maps. These maps provide valuable insights into the distribution and density of berries within each bunch (Section IV-C). To estimate bunch weight, we employ a regression model that establishes the relationship between the vision-derived features (*i.e.* area and berry count) and the actual weight of each bunch. Finally, we estimate the total grape weight in a panel or row by aggregating random samples drawn from a bunch

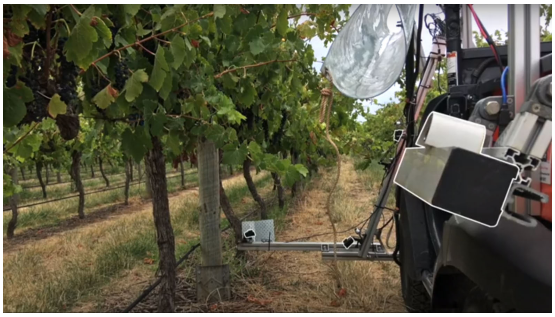


FIGURE 2. Acquisition system. Images were taken by commercially available cameras mounted to a moving vehicle.

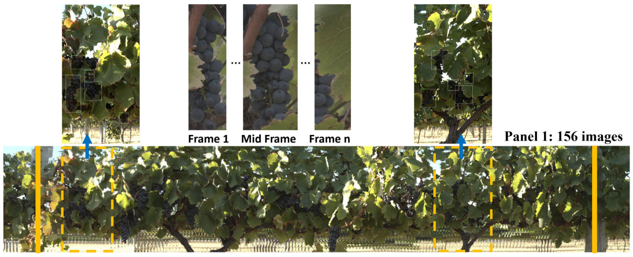


FIGURE 3. Representation of one panel of the vineyard recorded with the Blackmagic video camera (Panoramic image reconstructed from 156 sequential images).

weight distribution generated from our computer vision approach. This ensures a simple but comprehensive and reliable estimation of grape yield (Section IV-D). Section IV elaborates on the specific techniques used on the system workflow for dynamic analysis.

A. IMAGE ACQUISITION SYSTEM

A sensing platform was mounted on a *Kubota* RTV 500 and arranged utilising an aluminium frame to ensure precise and replicable alignment across all orientations and angles. The mechanical configuration is illustrated in Figure 2. All equipment and cabling were calibrated in the laboratory and placed in the field, as appropriate. A Blackmagic camera of 50 *fps* was mounted to the moving vehicle with an acquisition vehicle speed of approximately 4 *km/h*. The video length was typically 10 to 15 minutes. Figure 3 depicts a panoramic image reconstruction showcasing all of the images captured within a specific panel using the Blackmagic camera. It also shows a series of various perspectives (first, middle, and last views), emphasising the complexity of analysing the temporal dynamics of the views, as well as the occlusion of bunches and berries within a selected bunch.

B. DATASET COMPILATION

1) CSIRO PRE-HARVEST DATASET

To address the scarcity of annotated viticulture datasets, particularly video datasets, we generated a new dataset named the CSIRO Pre-harvest dataset. The proposed dataset consists of high-resolution images captured from a vineyard cultivating the Mataro grape variety during the 2021-2022 development and harvesting period. We selected 300 images captured by a Blackmagic camera with a resolution of 6,144 × 3.456 to train our models. These images were taken under various lighting conditions, including sunny and cloudy days. Bunch masks were labelled using polygon annotations, as shown in Figure 4, and point-based annotations were used for berry detection, as depicted in Figure 5.

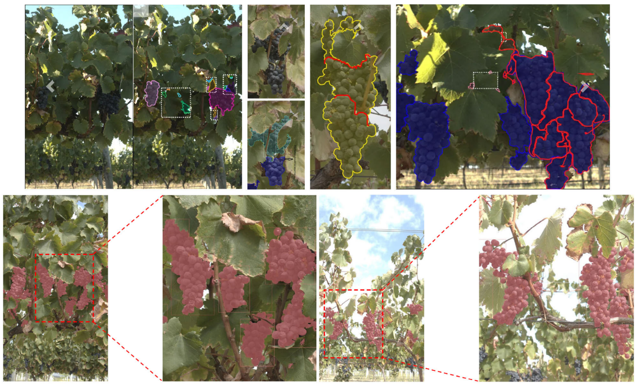


FIGURE 4. Top: Representation of the detailed annotation of all bunches. Bottom: Sample video frames labelled using polygons/masks (in red colour) to provide digital ground-truth.

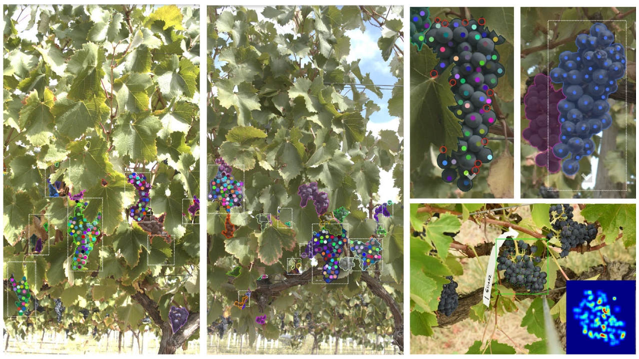


FIGURE 5. Sample video frames of the berry annotations (point-based annotations).

2) PUBLICLY AVAILABLE DATASETS

In constructing a comprehensive dataset for vineyard analysis, a combination of a public dataset, Embrapa WGISD [104], and our custom dataset was employed. The chosen dataset was selected based on its longevity and ability to support the development of different stages of our analysis pipeline, particularly bunch detection, segmentation, and berry detection. It's important to highlight that the decision to use a single publicly available dataset was influenced by the scarcity of alternatives providing video, a necessity for validating our dynamic analysis methodology. Researchers seeking datasets for vineyard analysis validation are encouraged to explore other available datasets listed below, considering their suitability for specific research objectives and methodologies.

- The Embrapa Wine Grape Instance Segmentation Dataset [104]. (Embrapa WGISD). The Embrapa

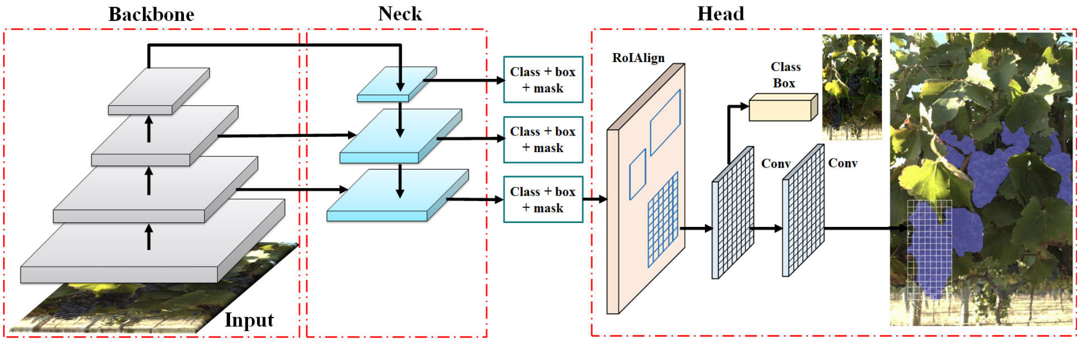


FIGURE 6. Grapevine bunch detection and segmentation model. The 'backbone' consists of a Swin Transformer network to extract deep and complex image features. The 'neck' is a feature pyramid network for multi-scale object detection, and the 'head' is comprised of a Mask-RCNN for bounding-box detection, class prediction, and segmentation mask generation for the grapevine bunches.

(WGISD) was developed to offer images and annotations for studying object detection and instance segmentation in the context of image-based monitoring and field robotics in viticulture. This dataset encompasses instances captured from five distinct grape varieties within vineyards (Chardonnay, Cabernet Franc, Cabernet Sauvignon, Sauvignon Blanc and Syrah). Extension of berry counting annotations using Huawei ModelArt were provided in [18].

- The Grape CS-ML database [10], released by Charles Sturt University, consists of five datasets showcasing 15 grape varieties at different stages of development, accompanied by size and Macbeth colour references.
- The CR1 and CR2 Datasets [63] (CR2) provides single berry annotations.
- The wGrapeUNIPD-DL dataset [105] (wGrapeUNIPD-DL) comprises 373 images of various grape varieties captured at different phenological stages across six Italian vineyard locations.
- GrapesNet [106], published in 2023, offers four datasets containing RGB and RGB-D images of grape bunches, facilitating tasks such as grape segmentation and weight prediction. However, such data are not sequential and thus do not allow proper dynamic analysis.

Other supportive datasets that are adopted for proper transfer learning of each component of the workflow are ImageNet [107] and COCO [108] datasets for detection and segmentation, and Motional Analysis and Re-identification Set (MARS) [109] dataset for Multi-object tracking.

IV. PIPELINE FOR YIELD ESTIMATION THROUGH DYNAMIC ANALYSIS OF GRAPEVINE BUNCHES

A. GRAPEVINE BUNCH DETECTION AND SEGMENTATION

In our prior research [3], we demonstrated the effectiveness of a detector utilising ResNeXt, a feature pyramid network (FPN), and RetinaNet deep learning models in efficiently identifying grapevine inflorescence across diverse lighting and background conditions. This approach allowed us to generate early yield potential estimates at budburst. The detection of small inflorescences in complex backgrounds with leaves, vines, other rows, clouds, and sun, proved to be difficult. Still, our detector was good enough to detect those inflorescences. In this work, we aim to extend and build upon this knowledge towards grapevine bunch detection, close to harvest. Our study introduces an object detection framework centred on instance segmentation techniques for bunch detection and segmentation as discussed in Section II-A3.

The proposed bunch detection and segmentation framework integrates a Swin Transformer [110] convolutional neural network for deep and complex feature extraction, an FPN [88] for capturing multi-resolution features, and a Mask-RCNN [83] to extract bounding-boxes, class labels, and segmentation masks. The architecture of this framework is visually represented in Figure 6 as three main networks: a backbone, a neck, and a head.

The backbone network is where an input image is fed to a Swin Transformer. This is a hierarchical Transformer whose representation is computed with Shifted windows. The shifted windowing scheme brings greater efficiency by limiting self-attention computation to non-overlapping local windows while also allowing for cross-window connection. Compared to our previously used ResNeXt backbone, the Swin Transformer provided better accuracy with similar model size, FLOPs, and latency. The image features extracted by the Swin Transformer are utilised by an FPN neck, which is an extension to Faster-RCNN and provides a robust way to deal with images of different scales while maintaining real-time performance. Finally, a Mask-RCNN head is used for detecting bounding-boxes, predicting their class labels, and generating their segmentation masks. Mask-RCNN is simple to train and outperforms many state-of-the-art segmentation models with a small overhead of predicting the segmentation mask on top of Faster-RCNN's bounding-box and class predictions. Region of interest align (RoIAlign) extracts small features for pixel-to-pixel alignment for each of the region proposals. This provides much better segmentation masks as output for each instance, in our case, grapevine bunches.

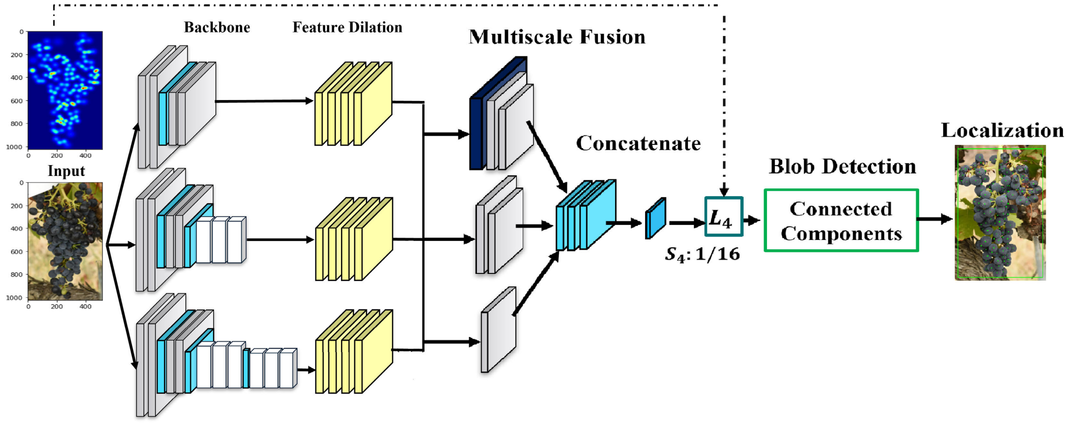


FIGURE 7. Berry detection based on multi-scale crowd counting and localisation by multitask point supervision approach [111] with task-specific customisation. Cropped, resized bunch bounding box and its ground truth density map at the left.

B. BUNCH TRACKING AND COUNTING

From existing SDE methods, as discussed in Section II-B, two multi-object tracking algorithms, DeepSORT [96] and K-Shortest Path Siamese Network (KSP-SiamFC) [3], were used to form tracks of individual grape bunches across consecutive video frames. A tracking algorithm was used to count the number of grape bunches visible on the vines reducing the effect of multi-counted bunches. The number of tracks formed from the DeepSORT or KSP-SiamFC algorithms was used to estimate the number of bunches on the vine. Furthermore, the bunch tracks were used to extract more robust bunch features, given multiple views of each bunch were combined for bunch weight estimation.

DeepSORT is a simple tracking-by-association method for real-time applications that exploit object detection results. DeepSORT utilises the motion and appearance of bounding box detections in order to track objects across video frames. In order to perform track assignment, a Kalman filter is used to predict the future position of tracks, whilst a deep embedding network is used to compare the appearance of tracks and newly detected objects. KSP-SiamFC uses the appearance of detected objects to form long-term tracks of short, discontinuous track fragments (*i.e.* the tracklets). The K-shortest path (KSP) algorithm, which is based on a Linear Programming formulation, is used to generate an optimal set of detected object tracks across a batch sequence of video frames. Furthermore, a deep Siamese network (SiamFC) is used to compare the appearance of bounding box detections between different frames of the KSP optimisation.

C. BERRY DETECTION AND COUNTING

In Section II, we introduced various methods to tackle the berry localisation and counting challenge. These methods include object detection, segmentation, instance segmentation and density estimation. For our approach, we leverage density-based crowd-counting techniques [112], which have been proven effective in handling counting tasks in densely crowded environments. Density approaches discussed in Section II-A4 employ crowd heads as point annotations and predict both location and number of people at the same time. Such methods are highly sensitive to the choice of the kernel and result in inconsistent performance with respect to varying crowd densities. In addition, this category of approach typically uses density feature maps to estimate the object count but fails to capture individual object information such as its location.

One promising work is the multitask point supervision (MPS) [111] which benefits from a multitask solution by learning multi-scale representations of encoded crowd images and uses point supervision to allow for both crowd numbers and locations to be accurately estimated. This approach demonstrates the effectiveness of both counting and localisation tasks on two popular datasets in the crowd-counting domain (ShanghaiTech A and B [92]).

We adopt the MPS model [111] with customised modification. Firstly, the original model used fused feature maps for counting and one single-scale feature map for localisation. We believe this creates some inconsistency in the training process. The proposed variant uses feature maps of varying scales combining the three losses of the different feature branches and a loss associated with a fused branch for training. In the meantime, the authors also set a case-specific weight for each loss, which needs manual configuration in different scenarios. It potentially increases the difficulty of training convergence and reduces the chance of achieving good estimation for both tasks simultaneously. Meanwhile, the main difference between the berry counting task and general crowd counting is that berries are of a similar scale within images as their distances to the camera do not vary largely. We, therefore, choose to use a fused feature map for both counting and localisation tasks and retain the loss for the fused branch in the training process. Such a framework is depicted in Figure 7. Similarly to the original paper, we use

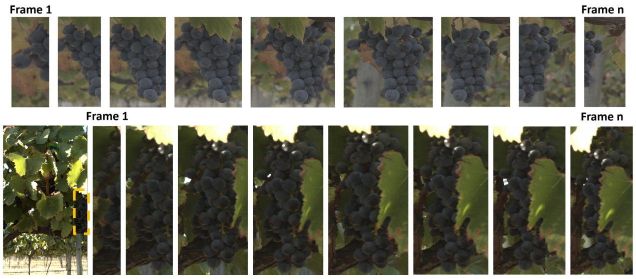


FIGURE 8. Representation of bunch sequences with and without occlusion and the impact on dynamic berry counting.

the Mean Squared Error (MSE) loss between the estimated map at a certain scale (consisting of head locations at this scale) and its ground truths.

The pipeline initiates with an input image featuring a crowd scene. This image undergoes feature extraction through three CNN encoders pre-trained on ImageNet [107] to derive meaningful features at multiple spatial scales. Subsequently, the extracted feature embedding is fed to dilated convolutional layers to extend the layers' receptive fields and capture higher-level features. Then the dilated embeddings are fed into the multi-scale fusion module, which generates the final crowd density map for both crowd counting and localisation. More specifically, three networks with different numbers of convolutional layers down-sample the dilated embedding at three spatial scales, respectively, and generate the same-sized outputs for concatenation. The concatenated embedding generates a comprehensive crowd density map that reflects the overall distribution. Beyond crowd density estimation, we also use the final embedding to perform individual localisation, utilising a connected components algorithm [113] to obtain the blobs, which represent the central point of each berry in the scene. Employing a multitask learning approach with point supervision, the network is trained to predict the coordinates of individuals as points on the density map. To effectively train the network, a defined loss function combines the density map estimation loss and the point localisation loss. This comprehensive loss function guides the network in mastering both tasks seamlessly.

It is noted that occlusion will exert a negative impact on the accuracy of grape counting, as it leads to non-linearities in the relationship between grape number and weighted heatmap. The recommended way is to use frames with a relatively low occlusion rate, as shown in Figure 8 where middle frames provide the best estimation of berry numbers.

D. BUNCH WEIGHT REGRESSION AND YIELD ESTIMATION

A regression model representing the relationship between the features of the segmented grape bunches and their true weights was trained. In this particular regression, two features of the segmented bunch were considered as independent variables; the pixel area and the number of berries.

The Swin Transformer Mask-RCNN model was used to detect and segment grape bunches that were tracked across consecutive video frames. To estimate the features of each grape bunch from its track sequence, different statistical measures were used. This includes the measures of the central tendency of a feature sequence (*i.e.,* mean or median) or measures that capture the features of a single element (*i.e.,* one bounding-box segmentation) in the bunch track. For instance, the maximum values of a bunch segmentation feature within the track sequence.

In previous studies of bunch weight estimation, other geometric features were shown to adversely affect prediction (*i.e.,* the perimeter of the segmented bunch) or were largely redundant when the pixel area feature was used (the major axis or minor axis of the segmented bunch).

Once the regression model was applied to grape bunches segmented from the video associated with the assessment area, a bunch weight distribution was generated. A threshold was employed to differentiate partial grape bunches (those partially occluded in the video) from full grape bunches based on the segmented area. Only the bunches classed as "full size" were used to fit the weight distribution to reduce sampling bias. If all of the segmented bunches were considered, including the occluded bunches, the weight distribution would be biased towards very small, low-weight grape bunches.

The number of random samples drawn from the bunch weight distribution was selected to be equal to the estimated grape bunch count *i.e.,* the number of bunch tracks estimated by the multi-object tracking algorithm. The random samples were then summed to generate a yield estimate over the assessment area.

V. EXPERIMENTS AND DISCUSSION

A. GRAPEVINE BUNCH DETECTION AND SEGMENTATION

1) EXPERIMENTAL SETUP

The experiments for the development of the grapevine bunch detection and segmentation model were performed using a single GPU of 16GB. During pre-processing, small annotations of bunches of less than 250 × 250 pixels were excluded (which usually contained 3 to 5 berries). This lets us focus on the bunches with better visibility. During inference, small bunches would typically become large enough for detection as they were captured from different perspectives across multiple frames of a video, as shown in Figure 8. The image dataset of 300 images captured by a Blackmagic camera with a resolution of 6,144 × 3.456 was split into 80% for training and 20% for testing.

We performed transfer learning that is initialising the network with pre-trained weights of the Swin Transformer on the COCO benchmark dataset. This allowed quick training and fine-tuning of the network on our dataset. Data augmentation included horizontal flips and images were resized to 1,536 × 864 to fit them into the GPU. The training phase comprised 50 epochs where one epoch means going through all training images once. A learning rate of 0.0001 and batch size of 1 were set, and the AdamW optimiser was used. During testing, an intersection-over-union (IOU) of 0.5 and a minimum bounding-box confidence of 0.3 were applied. IOU is defined as $(A \cap B)/(A \cup B)$, where A and B refer to ground truth and predicted bounding boxes, respectively.

To quantitatively evaluate the detection and segmentation results, precision and recall are calculated as $p = \mathrm{TP}/(\mathrm{TP} + \mathrm{FP})$ and $r = \mathrm{TP}/(\mathrm{TP} + \mathrm{FN})$, respectively. Here, TP, FP, and FN refer to true-positive, false-positive, and false-negative, respectively. Precision gives a percentage that shows how accurately a model predicts and recall gives a percentage that shows how many actual targets are detected out of the total targets. At each detection, a pair of precision and recall values is obtained to draw a curve called the recall-precision curve (RPC). From this curve, average precision (AP) is calculated, which provides a quantitative score that shows how good the detection model is. The AP is calculated by finding the area under the RPC by interpolating over every level of recall as,

$$\mathrm{AP} = \sum (r_{n+1} - r_n) p_{interp}(r_{n+1}), \tag{1}$$

$$p_{interp}(r_{n+1}) = \max_{\widetilde{r} \geq r_{n+1}} p(\widetilde{r}), \tag{2}$$

where $n = 0$ to all, r_n represents the nth recall value, $p_{interp}(r_{n+1})$ represents the interpolated precision at recall level r_{n+1}. Taking a mean of AP of different classes gives the mean average precision for detection (bounding-box mAP) and segmentation (segmentation mAP), calculated at $\mathrm{IOU} = 0.5$.

2) RESULTS

Firstly, the training process for grape bunch detection and segmentation is analysed. Three different types of losses: bounding-box loss, class loss, and segmentation mask loss, were calculated during the training process. These losses were minimised quickly and became stable after 6 to 7 epochs. The network was fine-tuned quickly for our dataset due to the transfer learning and initialising of the network using a pre-trained COCO model. We trained the network for 50 epochs.

Visual results of the grapevine bunch detection and segmentation are presented in Figure 9. The red-coloured bounding boxes and polygons represent the ground truth, and the blue-coloured bounding boxes with filled areas/masks represent the detected and segmented bunches. The percentages on the bounding boxes in blue colour represent the confidence of the model in predicting those detections. We have only one class: Bunch. From the visual results, we can see that our detection results match the ground truth quite well. Also, the segmentation results have boundaries that are very accurate compared to the ground-truth polygons. The detection and segmentation results for the isolated and non-overlapping bunches with fewer occlusions from leaves are shown in Figure 9(a) and for merged and overlapping bunches with occlusion from leaves are shown in Figure 9(b). Although the merged and overlapping bunches present greater challenges, our model can detect their boundaries with good accuracy.

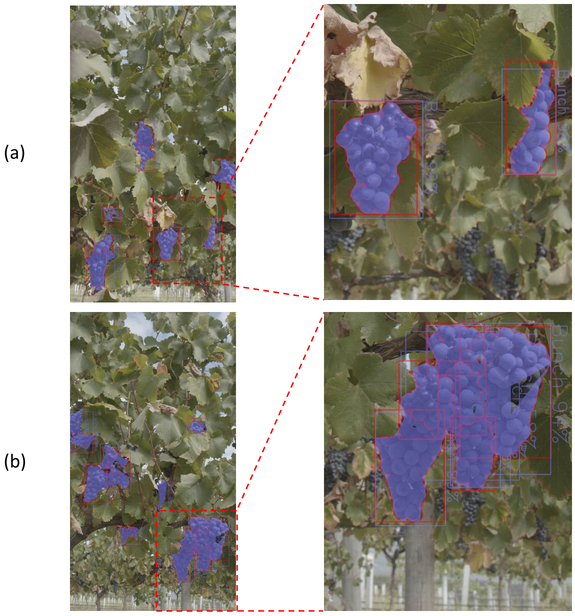


FIGURE 9. Grapevine bunch detection and segmentation results. Red-coloured bounding-boxes and polygons represent ground-truth. Blue coloured bounding-boxes and filled areas represent the detected bunches and their segmentation masks, respectively. The detection confidence score is given with each bounding-box in blue colour. (a) Isolated bunches with fewer occlusions. (b) Heavily merged and occluded bunches.

Finally, the test results of the bunch detection and segmentation model were evaluated quantitatively using mAP for bounding-box detections and segmentation masks of those detections. The model was trained for 50 epochs and the evaluation metrics: bounding-box mAP and segmentation mAP were obtained for the test subset after each epoch. The mAPs were raised quickly for the first 7 epochs and achieved more than 75% at epoch 10, as shown in Figure 10. The mAP@50 means that the mAP was calculated using $\mathrm{IOU} = 0.5$ during inference. The highest mAP for bounding-box predictions achieved was 77.3% and the highest mAP for segmentation mask predictions achieved was 77.1%, on the test subset.

The bunch detection and segmentation results that were presented were promising both quantitatively and visually. Consequently, the results of this detection and segmentation approach were used for bunch counting, berry counting, and weight regression tasks proceeding in the analysis pipeline.

B. BUNCH COUNTING

1) EXPERIMENTAL SETUP

The dataset utilised for training the tracking algorithms was comprised of a collection of 38 grape bunch tracks. These grape bunches, belonging to the Mataro species, were distinctly marked on the vine, allowing for their unique identification within the video. The trajectory of each grape

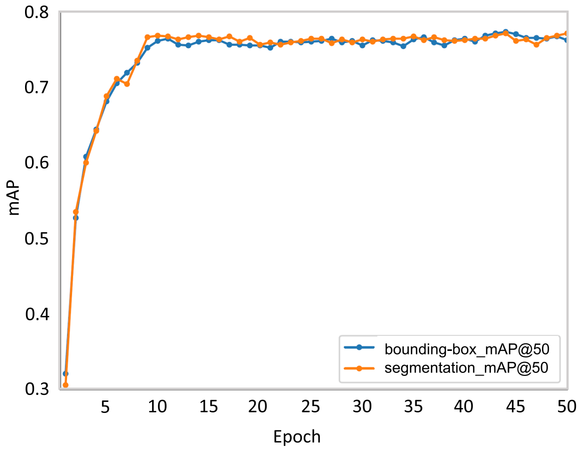


FIGURE 10. Mean average precision (mAP) versus the number of epochs for the grapevine bunch bounding-box detection and segmentation on the test dataset.

bunch was generated by manually annotating the bounding box of the bunch across frames where it was at least partially observable. To validate the bunch counting algorithm, three vineyard panels featuring the Mataro species were selected. The number of grape bunches in each panel was manually counted by two independent observers using video footage. The final bunch count for each panel was computed as the average of the counts provided by the two observers.

The estimation of grape bunch counts in each panel was accomplished by integrating a grape bunch detector with a tracking-by-detection-based algorithm. Three different combinations of object detectors and deep network-based tracking algorithms were considered: i) the YOLOv4 detector with the DeepSORT tracker, ii) the YOLOv4 detector with the K-Shortest Path Siamese Network (KSP-SiamFC)tracker, and iii) the Swin Transformer Mask-RCNN detector with the KSP-SiamFC tracker. These combinations were selected based on the literature reviewed in Section II-B. The deep embedding networks of the DeepSORT and the KSP-SiamFC trackers were trained using pairs of bounding boxes associated with a bunch track. The trackers were fine-tuned with 1000 instances of bounding box pairs randomly selected from the training set of 38 bunch tracks.

2) RESULTS

Table 4 shows that a combination of the Swin Transformer-based Mask-RCNN (Swin-T Mask-RCNN) detector and KSP-SiamFC tracker achieved the most accurate bunch count estimates. It achieved the lowest estimation error of 11.3% when averaged across the three panels and the most accurate bunch counts for panels 1 and 3 in particular. The Yolov4 and KSP-SiamFC combination was the next most accurate approach for bunch counting, with an average panel estimation error of 23.6%. The baseline combination of Yolov4 and DeepSORT offered the lowest bunch count accuracy with an average panel estimation error of 28.3%, despite producing a perfect bunch count estimate for panel 2.

TABLE 4. A comparison of grape bunch counts estimated from the video of three vineyard panels using three different tracking-by-detection approaches. Each approach considered a different combination of object detection and tracking methods. The observed count refers to the number of grape bunches manually counted from the video of each panel.

Vineyard Panel	Observed Counts	Estimated Counts		
Panel 1	88	44	49	**61**
Panel 2	72	**72**	81	75
Panel 3	107	144	126	**118**
	Detector	YOLOv4		Swin-T Mask-RCNN
	Tracker	DeepSORT	KSP-SiamFC	

Tracking grape bunches was found to be challenging as a result of how bunches were distributed on the vine. Bunches were often densely concentrated on the vine resulting in the partial or full occlusion of some bunches. Furthermore, the appearance of bunches was often quite similar in terms of texture, shape and colour, and hence, when several bunches were close to one another, tracking algorithms were highly susceptible to the misassignment of bounding boxes with other nearby bunch tracks.

C. BERRY DETECTION

1) EXPERIMENTAL SETUP

Two experiments were conducted—one utilising our CSIRO pre-harvest dataset and another employing a combined dataset that included both public (Embrapa) and our customised data. The goal was to develop a tailored berry counting model for our specific application.

The total number of observable berries in the scene is obtained by summing the predicted multi-scale fusion map. Counting performance is evaluated using the mean absolute error (MAE) of the predicted count in comparison to the ground truths. To assess localisation results, average precision is employed, measured as the area under the precision-recall curve. A detected object is considered a true positive if it overlaps with the ground truth object location within a specified threshold.

Berry counting is conducted after each bunch has been detected in the image frame. Since the camera depth remains consistent, and no morphological traits are utilised at this step, the bunch bounding boxes are cropped and resized to a predefined size (1,024 × 512) before entering the berry counting stage. Simultaneously, ground truth berry annotations in terms of (x, y) coordinates in the original image frame are transferred into local coordinates in the cropped and resized individual image, with both density map and point map generated for training. Note that these locations must be normalised to be used at different scales.

The implementation details are outlined as follows: PyTorch served as the primary deep-learning platform for berry detection. In the first experiment, we trained our model using only our customised dataset with a split of 3:1:1 for training, validation, and testing. A total of 110 bunch bounding boxes were provided as input. The batch size was set to 512, and the Adam optimiser was used with a

TABLE 5. Berry counting performance.

Experiment	↓ MAE	↑ AP
Exp 1	6.23	**0.78**
Exp 2	**4.4**	0.64

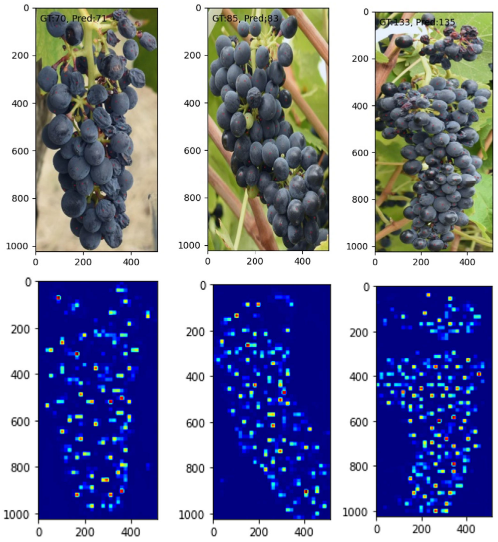


FIGURE 11. Predicted berry counts and their heat map from testing images.

momentum of 0.937 and an initial learning rate of 0.0001. The model converged quickly within 20 epochs. In the second experiment, we combined our CSIRO pre-harvest dataset with the Embrapa dataset. Since multiple bounding boxes are provided for each raw image in the Embrapa dataset, a total of 4,404 bunch bounding box images were used as input for berry counting (training, validation, and testing).

2) RESULTS

The results of the two described experiments are presented in Table 5. As evident from the table, counting performance improves with an increased amount of training data. However, the accuracy of localisation is impacted by the greater variation in bunch images introduced by the public dataset. Given our application's objective of delivering precise berry counts and the necessity for the model to generalise across diverse conditions (full bunches, partially occluded bunches, different viewing angles, etc.), we opt to utilise the model trained in Experiment 2 for our application. Qualitative results are illustrated in Figure 11.

D. WEIGHT REGRESSION

1) EXPERIMENTAL SETUP

The dataset used for training and testing the bunch weight regression models was a set of 38 annotated tracks of visually tagged Mataro bunches. The Mask-RCNN model was used to segment the bounding boxes of each grape bunch track. The individual weights of these Mataro bunches were recorded during harvest.

A three-fold cross-validation method was used to train and test the bunch weight regression models. Four different regression methods were used; two linear models (the standard linear regression and Huber regression), and two non-linear models (Support Vector Regression (SVR) and Random Forest Regression).

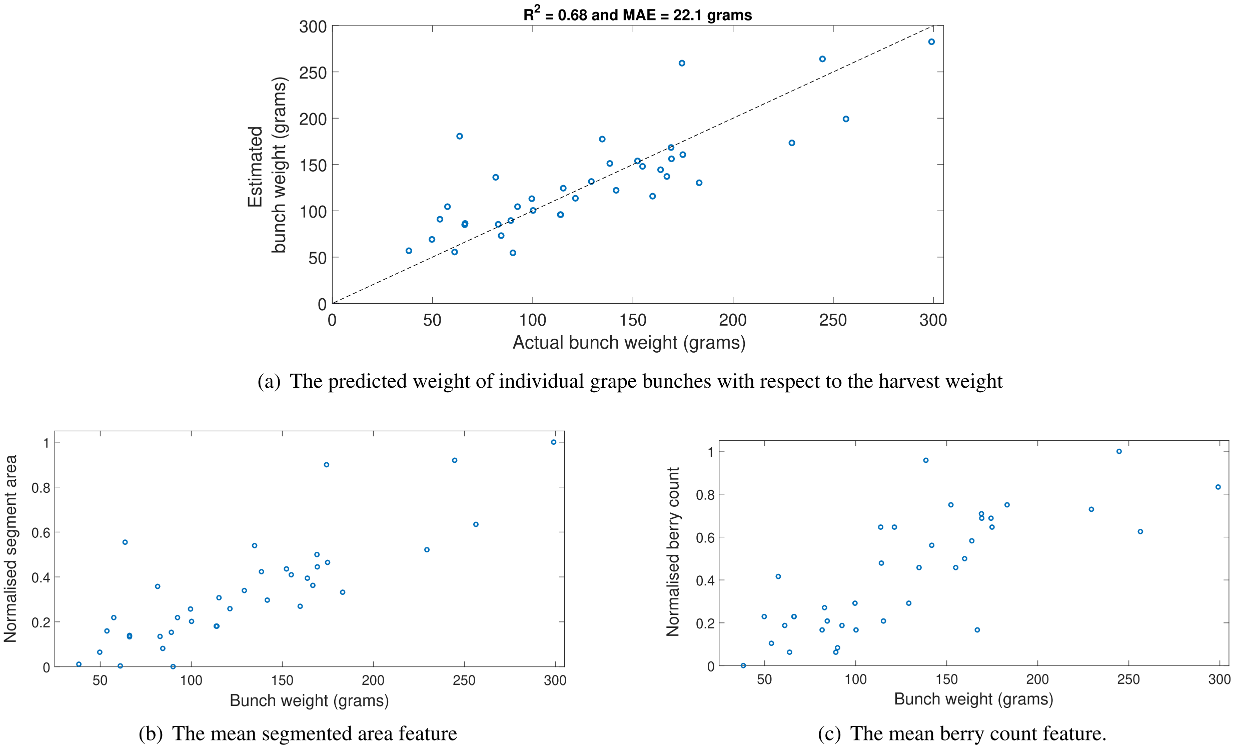
Three different attribute sets were used to represent each grape bunch; the segmented pixel area, the berry count of the segmented bunch or a combination of the pixel area and berry count. The bunch features used by the regression model were then estimated by computing the mean, the median or the maximum value (*max*) of each feature in the segmented sequence track independently. The fourth estimation approach was based on the maximum value of a joint set of the pixel area and berry count features (max-jnt). This measure was computed by independently scaling the two features between 0 and 1; the sum of the scaled features was computed for each sequence element. The original features associated with the sequence element with the maximum summed value were then used to estimate the bunch features. The R-squared (R^2) and Mean Absolute Error (MAE) metrics were used to evaluate the performance of the bunch weight regression models.

2) RESULTS AND DISCUSSION

Table 6 shows the MAE and R^2 performance of four different regression models averaged across different feature sets and statistical measures for feature estimation. The linear models achieved superior estimation performance to the non-linear models with an average MAE and average R^2 improvement of 16.9% and 72.9%, respectively. The Huber model achieved the highest bunch weight performance with an average MAE of 27.0 grams and average R^2 of 0.44, which is a 4.2% and 12.8% improvement over the standard linear regression model. The Huber model's advantage was associated with it using a loss function that was less influenced by sample outliers (such as the outlier features of grape bunches shown in Figure 12) compared to the least-squares loss of a standard linear regression.

Table 7 shows the bunch weight estimation performance of the Huber regression models with respect to the different feature sets and feature estimation methods. In general, regression models comprised of bunch area and berry count features achieved superior performance to the regression models composed of individual features. The joint feature model offered an MAE and R^2 improvement over the bunch area-based regression model of 6.6% and 28.6%, respectively. The joint feature model achieved an even larger MAE and R^2 advantage over the berry count regression model of 18.3% and 171.4%, respectively.

Table 7 shows the four statistics used to estimate the independent variables of the regression, the statistics of central tendency (mean and median) were shown to offer superior weight estimation performance to the maximum



(a) The predicted weight of individual grape bunches with respect to the harvest weight

(b) The mean segmented area feature

(c) The mean berry count feature.

FIGURE 12. The bunch weight estimation performance of a Huber regression model based on the berry count and bunch area features. The computed features are shown with respect to their true bunch weights.

TABLE 6. A comparison of the test performance of four different regression models averaged over four feature estimation methods and three sets of geometric features. The performance of Linear regression, Huber regression (Linear regression with outlier detection), Random Forest regression (Forest) and Support Vector Regression (SVR) were evaluated with respect to the Mean Absolute Error (MAE) and R-squared (R^2) criteria.

Regression Model	↓ MAE	↑ R^2
Linear	28.2	0.39
Huber	**27.0**	**0.44**
Forest	32.8	0.25
SVR	33.6	0.23

TABLE 7. The test performance of Huber regression models that estimate the weight of individual grape bunches based on three-fold cross-validation with respect to four different feature estimation methods and three different sets of geometric features. The Huber regression model using area and berry count features, estimated from the mean features of a segmented bunch sequence, achieved the lowest Mean Absolute Error (MAE) of 22.1 grams and highest R-squared of 0.68.

	Feature estimation							
Geometric	max-jnt		max		mean		median	
Features	MAE	R^2	MAE	R^2	MAE	R^2	MAE	R^2
area + berry count	27.3	0.55	26.2	0.57	**22.1**	**0.68**	22.9	0.67
area	28.5	0.38	27.0	0.43	24.1	0.58	25.4	0.53
berry count	28.7	0.24	29.6	0.24	30.6	0.22	31.7	0.21

statistic when developing the single and joint feature regression models. Given bunch segmentation and berry detection methods introduced error into feature estimation, computing the mean or median of a sequence of bunch features smoothed out some of the estimation error relative to models that were computed from a single feature of the sequence track. Finally, the mean computed features achieved the highest weight estimation performance with a minor MAE and R^2 improvement of 3.4% and 4.8% over the median computed features.

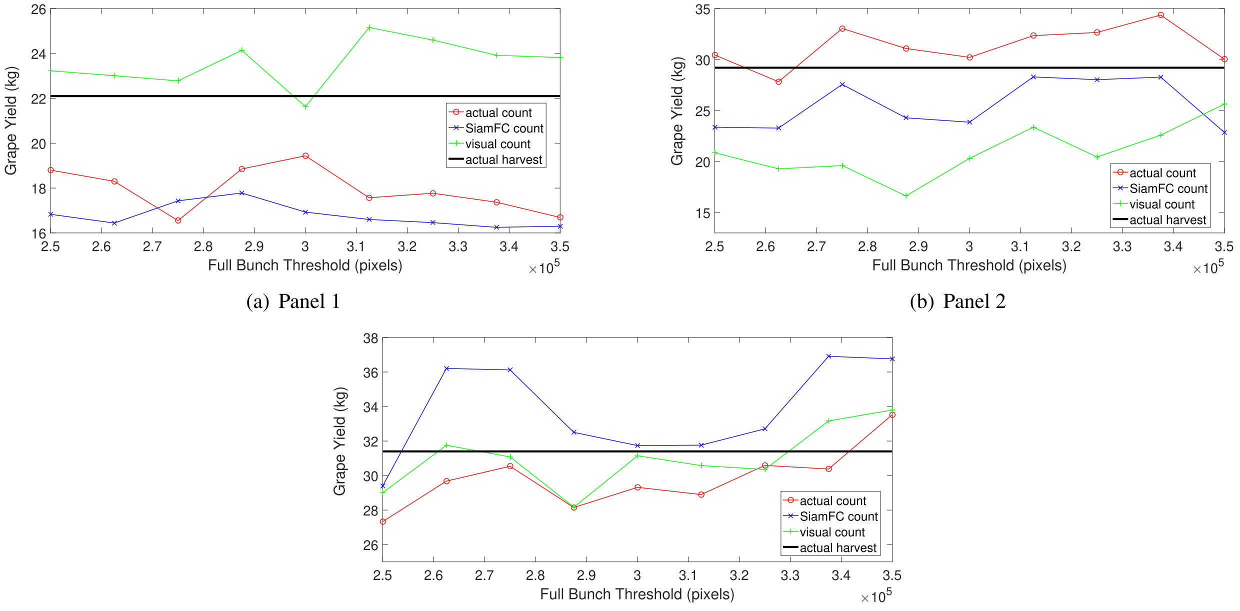
The Huber regression model using the mean area and mean berry count features as independent variables achieved the highest weight estimation performance with an MAE of 22.1 grams and R^2 of 0.68. Figure 12 shows the estimated and true weight of the bunches for this particular regression model based on three-fold cross-validation.

E. YIELD ESTIMATION

1) EXPERIMENTAL SETUP

The video from the three vineyard panels that were used to validate the bunch counting model was also used to evaluate yield estimation with our proposed system. The videos were recorded on the day of harvest; the total count and total weight of Mataro bunches harvested in each of the three panels were recorded.

Our initial experiments involved the training and validation of computer vision modules developed for separate analysis tasks. In this section, these separate modules were combined to form a system to produce grape yield estimates. The Swin transformer Mask-RCNN model was used to detect and segment grape bunches from individual video frames of the



(a) Panel 1

(b) Panel 2

(c) Panel 3

FIGURE 13. The proposed computer vision system was utilised to estimate the harvested yield of three different vineyard panels. For each panel, yield estimates were obtained by sampling from a distribution of grape bunch weights computed through computer vision, and adjusted by the estimated number of grape bunches on the vine. These bunch weight distributions were computed using the Huber weight regression model, incorporating mean berry count and mean area features, with a range of full bunch thresholds (*i.e.* number of pixels required to be a fully segmented bunch). The accuracy of yield estimates was assessed using three different approaches to estimate the number of grape bunches: (i) the recorded number of harvested grape bunches (actual count), (ii) the number of grape bunches manually counted from video (visual count), and (iii) an automated bunch count estimated with the SiamFC tracking model (SiamFC count). These three yield estimates were then compared to the true yield values recorded for each panel.

vine canopy. The SiamFC algorithm was then used to track the segmented grape bunches across the video to count the number of bunches.

The bunch weight regression model with the highest performance in Section V-D2 (using mean pixel area and mean berry count features) was employed to generate a bunch weight distribution for each panel. Segmented grape bunches were classified as either full or partial using an area based threshold. Only "full" grape bunches were considered to be sufficiently observable within the video to be used to estimate the bunch weight. A range of thresholds between 2.5×10^5 and 3.5×10^5 pixels were considered to classify the segmented bunches within panel videos. A bunch weight distribution was then generated by fitting the "full" grape bunch weights to a beta distribution.

Yield estimates were then obtained by computing the sum of samples randomly drawn from a grape bunch weight distribution. Yield estimates of each panel were compared by using three different approaches to draw random samples from the bunch weight distribution. The number of samples was equivalent to the number of bunches (i) harvested from the panel (actual count) (ii) visually counted from the panel video (visual count) or (iii) automatically counted from the panel video based on the number of tracks identified with the SiamFC tracking algorithm. In each case, the sampling process was repeated 300 times to compute a mean estimate.

2) RESULTS AND DISCUSSION

Figure 13 shows the estimated grape yields for the three harvested vineyard panels. The Huber regression model using mean area and mean berry count features was used to generate grape bunch weight distributions across various full bunch thresholds. The yield estimates were then compared using three different approaches to determine the number of grape bunches in each panel (as outlined in Section V-E1): the actual count, the visual count and the automated SiamFC tracking count. Given vines had only been imaged from one side, the number of bunches visible within the video was significantly lower than the actual number present on the vine. To enhance the accuracy of yield estimates, we made the assumption that grape bunches were uniformly distributed on the vine. This assumption implied that the number of bunches on opposite sides of the panel vines was equivalent. Consequently, for the visual and SiamFC tracking approaches, bunch counts were doubled to estimate the yield.

Figure 13(a) shows the yield estimates based on the actual bunch count consistently underestimated the true yield of panel 1 with a minimum estimation error of 10.4% and mean estimation error of 18.9%. These results suggest the computer vision-derived bunch weight distribution was biased towards lower-weight bunches. In contrast, the yield estimates produced by the visual bunch count had a minimum error of 3.1%, whilst the automated SiamFC count had a far larger minimum error of 19.1%. Interestingly, in this particular case, the smaller yield estimation errors produced by the visual bunch count could be attributed to its significant under-estimation of the true bunch count of panel 1, as is shown in Table 4.

Figure 13(b)) shows the yield estimation errors derived from the actual bunch count of panel 2 were smaller than panel 1 with a minimum error of 2.7% and mean error of 8.3%. This suggests the computer vision-derived bunch weight distribution of panel 2 was more representative than the corresponding weight distribution of panel 1. In contrast, the yield derived from the visual bunch count was shown to be under-estimated consistently with a minimum error of 12.3%.

Table 4 shows the SiamFC tracking algorithm slightly overestimated the observed grape bunch count in panel 2. Consequently, it is the inflated bunch count that explains why the SiamFC approach offered superior yield estimates to the visual approach with a minimum estimation error of 3.0%.

Figure 13(c) shows the yield estimates of panel 3 were the most accurate of the three panels. The yield estimates derived from the actual bunch count achieved a minimum estimation error of 2.5% and mean estimation error of 6.0% indicating the computer vision-derived bunch weight distributions of panel 3 were the most representative of all three panels. Furthermore, the yield estimates produced by observed and SiamFC derived bunch counts were accurate with a minimum estimation error of 1.0% and 1.3%, respectively.

Performance Comparison With Previous Work: Most of the computer vision work in viticulture does not propose a complete system for grape yield estimation, instead focusing on one particular aspect (*i.e.* bunch segmentation) of a potential solution. There are a couple of examples of completely automated solutions that we will discuss here. [114] proposed a remote sensing-based approach to grape yield estimation using NDVI data products with yield errors ranging between 5.9% and 14.8%. In contrast to all other approaches, the results in [114] were reported on the training data as opposed to independent test data, and hence, were likely to be of a higher accuracy.

Nuske et.al [16] proposed an automated grape yield solution based on an unsupervised approach to berry detection producing yield errors of between 6.48% and 11.47%. In contrast to our more general approach, their methodology was tailored for use with a specialised imaging system that acquired images at a fixed distance to the vine canopy with artificial lighting at night.

Most similar to our work, [115] proposed an end-to-end deep learning model that produced grape yield errors of between 15% and 27.1% in a commercial vineyard. Although it was impossible to directly compare the performance of different automated methods, due to the significant differences between their evaluation trials, the reported results suggest our complete computer vision-based system was highly competitive with panel yield errors ranging between 1% and 19.3%.

VI. CONCLUSION

In grapevine production and breeding, accurate yield estimation and forecasting are crucial for managing logistics throughout the value chain. There is also a perceived trade-off between fruit quality and quantity, which is often reflected in grower contracts through limits on maximum yield. Early yield forecasting is, therefore, crucial as it allows targeted berry thinning, ensuring a high-quality outcome.

Our study contributes significantly to vineyard yield estimation by (1) Summarising recent research on deep learning techniques for bunch and berry detection and counting, (2) Designing a cost-effective image acquisition system, (3) Curating a novel dataset of RGB videos, and (4) Implementing a dynamic analysis pipeline for accurate yield estimation. Our pipeline integrates computer vision tasks for precise grape bunch and berry detection and counting from videos. A bunch weight regression model, using features extracted from segmented bunch tracks, computes a bunch weight distribution for the imaged area. Yield estimations are generated by randomly sampling this distribution based on the detected bunch count.

The yield estimates obtained with the proposed computer vision system were reasonable in two of the three vineyard panels with harvested weight errors of less than 5%. There was still some estimation inconsistency, however, as demonstrated by the large harvest weight errors (of more than 15%) for both the visually observed and tracking-based estimates of one panel. However, these errors should be contrasted with current practice errors of up to 30%.

Whilst the experiments indicate that there are errors associated with all tasks in the computer vision pipeline, the bunch tracking models were considered to be the major source of estimation error. Bunch tracking was used to produce an automated count of the number of bunches on the vine. Yield estimates produced by sampling computer vision-derived bunch weight distributions with true harvested bunch counts were found to be more robust and consistent than the yield estimates obtained by sampling distributions with tracking-based bunch counts. The dense concentration of bunches on the vine, bunch occlusion within images and the similarity of bunch appearance made tracking challenging and led to inaccurate bunch counts.

One potential future direction is to address the bunch tracking problem by employing a 3D multi-object tracking approach where depth information can be used to assist with bunch occlusion and appearance issues. Whilst the computational complexity associated with constructing three-dimensional structures from sequences of 2D images is traditionally high, the field is evolving. New computationally efficient methods are being developed to construct 3D structures in real time.

When the geometric features of bunches are used to estimate the yield, the accuracy will be affected by the distance at which the bunches are imaged. Consequently, the ultimate goal is to develop methods to detect and size individual berries that potentially reduce the effect of imaging distance to produce more accurate yield estimates.

ACKNOWLEDGMENT

The authors would like to acknowledge Qiyu Liao and Changming Sun for their assistance in data generation and validation. They also acknowledge Wine Australia which invests in and manages research, development, and extension on behalf of Australia's grape growers and winemakers and the Australian Government.

REFERENCES

[1] O. Chatrabgoun, R. Karimi, A. Daneshkhah, S. Abolfathi, H. Nouri, and M. Esmaeilbeigi, "Copula-based probabilistic assessment of intensity and duration of cold episodes: A case study of malayer vineyard region," *Agricult. Forest Meteorol.*, vol. 295, Dec. 2020, Art. no. 108150.

[2] P. R. Clingeleffer et al., "Final report to grape and wine research and development corporation," Grape Wine Res. Develop. Corp. (GWRCD), Wayville, SA, Australia, 2001.

[3] M. R. Khokher, Q. Liao, A. L. Smith, C. Sun, D. Mackenzie, M. R. Thomas, D. Wang, and E. J. Edwards, "Early yield estimation in viticulture based on grapevine inflorescence detection and counting in videos," *IEEE Access*, vol. 11, pp. 37790–37808, 2023.

[4] A. Barriguinha, M. de Castro Neto, and A. Gil, "Vineyard yield estimation, prediction, and forecasting: A systematic literature review," *Agronomy*, vol. 11, no. 9, p. 1789, Sep. 2021.

[5] C. Laurent, B. Oger, J. A. Taylor, T. Scholasch, A. Metay, and B. Tisseyre, "A review of the issues, methods and perspectives for yield estimation, prediction and forecasting in viticulture," *Eur. J. Agronomy*, vol. 130, Oct. 2021, Art. no. 126339.

[6] M. V. Ferro and P. Catania, "Technologies and innovative methods for precision viticulture: A comprehensive review," *Horticulturae*, vol. 9, no. 3, p. 399, Mar. 2023.

[7] C. Poblete-Echeverría and J. Tardaguila, "Digital technologies: Smart applications in viticulture," in *Encyclopedia of Smart Agriculture Technologies*. Cham, Switzerland: Springer, 2023, pp. 1–13.

[8] E. Akleman, "Deep learning," *Computer*, vol. 53, no. 9, pp. 1–17, Sep. 2020.

[9] E. Vrochidou and G. A. Papakostas, "Leveraging computer vision for precision viticulture," *Comput. Vis. Mach. Learn. Agricult.*, vol. 3, pp. 177–213, Aug. 2023.

[10] K. P. Seng, L.-M. Ang, L. M. Schmidtke, and S. Y. Rogiers, "Computer vision and machine learning for viticulture technology," *IEEE Access*, vol. 6, pp. 67494–67510, 2018.

[11] L. Mohimont, F. Alin, M. Rondeau, N. Gaveau, and L. A. Steffenel, "Computer vision and deep learning for precision viticulture," *Agronomy*, vol. 12, no. 10, p. 2463, Oct. 2022.

[12] C. Charisis and D. Argyropoulos, "Deep learning-based instance segmentation architectures in agriculture: A review of the scopes and challenges," *Smart Agricult. Technol.*, vol. 8, Aug. 2024, Art. no. 100448.

[13] S. F. Di Gennaro, P. Toscano, P. Cinat, A. Berton, and A. Matese, "A low-cost and unsupervised image recognition methodology for yield estimation in a vineyard," *Frontiers Plant Sci.*, vol. 10, p. 559, May 2019.

[14] A. Kamilaris and F. X. Prenafeta-Boldu, "Deep learning in agriculture: A survey," *Comput. Electron. Agricult.*, vol. 147, pp. 70–90, Apr. 2018.

[15] D. Font, M. Tresanchez, D. Martínez, J. Moreno, E. Clotet, and J. Palacín, "Vineyard yield estimation based on the analysis of high resolution images obtained with artificial illumination at night," *Sensors*, vol. 15, no. 4, pp. 8284–8301, Apr. 2015.

[16] S. Nuske, K. Wilshusen, S. Achar, L. Yoder, S. Narasimhan, and S. Singh, "Automated visual yield estimation in vineyards," *J. Field Robot.*, vol. 31, no. 5, pp. 837–860, Sep. 2014.

[17] T. T. Santos, L. L. de Souza, A. A. dos Santos, and S. Avila, "Grape detection, segmentation, and tracking using deep neural networks and three-dimensional association," *Comput. Electron. Agricult.*, vol. 170, Mar. 2020, Art. no. 105247.

[18] G. Deng, T. Geng, C. He, X. Wang, B. He, and L. Duan, "TSGYE: Two-stage grape yield estimation," in *Proc. Int. Conf. Neural Inf. Process.*, 2020, pp. 580–588.

[19] H. Li, C. Li, G. Li, and L. Chen, "A real-time table grape detection method based on improved YOLOv4-tiny network in complex background," *Biosystems Eng.*, vol. 212, pp. 347–359, Dec. 2021.

[20] M. Sozzi, S. Cantalamessa, A. Cogato, A. Kayad, and F. Marinello, "Grape yield spatial variability assessment using YOLOv4 object detection algorithm," in *Precision Agriculture*. Wageningen, The Netherlands: Wageningen Academic Publishers, 2021, pp. 193–198.

[21] M. Sozzi, S. Cantalamessa, A. Cogato, A. Kayad, and F. Marinello, "Automatic bunch detection in white grape varieties using YOLOv3, YOLOv4, and YOLOv5 deep learning algorithms," *Agronomy*, vol. 12, no. 2, p. 319, Jan. 2022.

[22] S. Su, R. Chen, X. Fang, Y. Zhu, T. Zhang, and Z. Xu, "A novel lightweight grape detection method," *Agriculture*, vol. 12, no. 9, p. 1364, Sep. 2022.

[23] S. Lu, X. Liu, Z. He, X. Zhang, W. Liu, and M. Karkee, "Swin-transformer-YOLOv5 for real-time wine grape bunch detection," *Remote Sens.*, vol. 14, no. 22, p. 5853, Nov. 2022.

[24] C. Zhang, H. Ding, Q. Shi, and Y. Wang, "Grape cluster real-time detection in complex natural scenes based on YOLOv5s deep learning network," *Agriculture*, vol. 12, no. 8, p. 1242, Aug. 2022.

[25] R. Zhao, Y. Zhu, and Y. Li, "An end-to-end lightweight model for grape and picking point simultaneous detection," *Biosystems Eng.*, vol. 223, pp. 174–188, Nov. 2022.

[26] I. Pinheiro, G. Moreira, D. Queirós da Silva, S. Magalhães, A. Valente, P. Moura Oliveira, M. Cunha, and F. Santos, "Deep learning YOLO-based solution for grape bunch detection and assessment of biophysical lesions," *Agronomy*, vol. 13, no. 4, p. 1120, Apr. 2023.

[27] C. Guo, S. Zheng, G. Cheng, Y. Zhang, and J. Ding, "An improved YOLO v4 used for grape detection in unstructured environment," *Frontiers Plant Sci.*, vol. 14, pp. 1–23, Jul. 2023.

[28] A. Blekos, K. Chatzis, M. Kotaidou, T. Chatzis, V. Solachidis, D. Konstantinidis, and K. Dimitropoulos, "A grape dataset for instance segmentation and maturity estimation," *Agronomy*, vol. 13, no. 8, p. 1995, Jul. 2023.

[29] T. A. Ciarfuglia, I. M. Motoi, L. Saraceni, M. Fawakherji, A. Sanfeliu, and D. Nardi, "Weakly and semi-supervised detection, segmentation and tracking of table grapes with limited and noisy data," *Comput. Electron. Agricult.*, vol. 205, Feb. 2023, Art. no. 107624.

[30] L. Shen, J. Su, R. He, L. Song, R. Huang, Y. Fang, Y. Song, and B. Su, "Real-time tracking and counting of grape clusters in the field based on channel pruning with YOLOv5s," *Comput. Electron. Agricult.*, vol. 206, Mar. 2023, Art. no. 107662.

[31] A. S. Aguiar, S. A. Magalhães, F. N. dos Santos, L. Castro, T. Pinho, J. Valente, R. Martins, and J. Boaventura-Cunha, "Grape bunch detection at different growth stages using deep learning quantized models," *Agronomy*, vol. 11, no. 9, p. 1890, Sep. 2021.

[32] J. Jaramillo, J. Vanden Heuvel, and K. H. Petersen, "Low-cost, computer vision-based, prebloom cluster count prediction in vineyards," *Frontiers Agronomy*, vol. 3, p. 8, Apr. 2021.

[33] M. Woodson and J. Zhang, "Evaluating self-supervised transfer performance in grape detection," in *Proc. Sci. Inf. Conf.*, 2023, pp. 1043–1057.

[34] S. Chen, Y. Song, J. Su, Y. Fang, L. Shen, Z. Mi, and B. Su, "Segmentation of field grape bunches via an improved pyramid scene parsing network," *Int. J. Agricult. Biol. Eng.*, vol. 14, no. 6, pp. 185–194, 2021.

[35] L. Shen, S. Chen, Z. Mi, J. Su, R. Huang, Y. Song, Y. Fang, and B. Su, "Identifying veraison process of colored wine grapes in field conditions combining deep learning and image analysis," *Comput. Electron. Agricult.*, vol. 200, Sep. 2022, Art. no. 107268.

[36] Y. Peng, A. Wang, J. Liu, and M. Faheem, "A comparative study of semantic segmentation models for identification of grape with different varieties," *Agriculture*, vol. 11, no. 10, p. 997, Oct. 2021.

[37] Y. Peng, S. Zhao, and J. Liu, "Segmentation of overlapping grape clusters based on the depth region growing method," *Electronics*, vol. 10, no. 22, p. 2813, Nov. 2021.

[38] R. P. Devanna, G. Reina, and A. Milella, "Automated detection and counting of grape bunches using a farmer robot," in *Multimodal Sensing and Artificial Intelligence: Technologies and Applications*, vol. 12621. Bellingham, WA, USA: SPIE, 2023, pp. 266–271.

[39] H. Cecotti, A. Rivera, M. Farhadloo, and M. A. Pedroza, "Grape detection with convolutional neural networks," *Expert Syst. Appl.*, vol. 159, Nov. 2020, Art. no. 113588.

[40] R. Marani, A. Milella, A. Petitti, and G. Reina, "Deep neural networks for grape bunch segmentation in natural images from a consumer-grade camera," *Precis. Agricult.*, vol. 22, no. 2, pp. 387–413, Apr. 2021.

[41] L. Ghiani, A. Sassu, F. Palumbo, L. Mercenaro, and F. Gambella, "In-field automatic detection of grape bunches under a totally uncontrolled environment," *Sensors*, vol. 21, no. 11, p. 3908, Jun. 2021.

[42] W. Yin, H. Wen, Z. Ning, J. Ye, Z. Dong, and L. Luo, "Fruit detection and pose estimation for grape Cluster–Harvesting robot using binocular imagery based on deep neural networks," *Frontiers Robot. AI*, vol. 8, Jun. 2021, Art. no. 626989.

[43] A. Chiatti, R. Bertoglio, N. Catalano, M. Gatti, and M. Matteucci, "Surgical fine-tuning for grape bunch segmentation under visual domain shifts," in *Proc. Eur. Conf. Mobile Robots*, Sep. 2023, pp. 1–7.

[44] G. Coll-Ribes, I. J. Torres-Rodríguez, A. Grau, E. Guerra, and A. Sanfeliu, "Accurate detection and depth estimation of table grapes and peduncles for robot harvesting, combining monocular depth estimation and CNN methods," *Comput. Electron. Agricult.*, vol. 215, Dec. 2023, Art. no. 108362.

[45] Y. Miao, L. Huang, and S. Zhang, "A two-step phenotypic parameter measurement strategy for overlapped grapes under different light conditions," *Sensors*, vol. 21, no. 13, p. 4532, Jul. 2021.

[46] X. Wei, F. Xie, K. Wang, J. Song, and Y. Bai, "A study on shine-muscat grape detection at maturity based on deep learning," *Sci. Rep.*, vol. 13, no. 1, p. 4587, Mar. 2023.

[47] E. Badeka, E. Karapatzak, A. Karampatea, E. Bouloumpasi, I. Kalathas, C. Lytridis, E. Tziolas, V. N. Tsakalidou, and V. G. Kaburlasos, "A deep learning approach for precision viticulture, assessing grape maturity via YOLOv7," *Sensors*, vol. 23, no. 19, p. 8126, Sep. 2023.

[48] Y. S. Woo, P. Buayai, H. Nishizaki, K. Makino, L. M. Kamarudin, and X. Mao, "End-to-end lightweight berry number prediction for supporting table grape cultivation," *Comput. Electron. Agricult.*, vol. 213, Oct. 2023, Art. no. 108203.

[49] F. Khoroshevsky, S. Khoroshevsky, and A. Bar-Hillel, "Parts-per-object count in agricultural images: Solving phenotyping problems via a single deep neural network," *Remote Sens.*, vol. 13, no. 13, p. 2496, Jun. 2021.

[50] J. Grimm, K. Herzog, F. Rist, A. Kicherer, R. Töpfer, and V. Steinhage, "An adaptable approach to automated visual detection of plant organs with applications in grapevine breeding," *Biosystems Eng.*, vol. 183, pp. 170–183, Jul. 2019.

[51] F. Palacios, P. Melo-Pinto, M. P. Diago, and J. Tardaguila, "Deep learning and computer vision for assessing the number of actual berries in commercial vineyards," *Biosystems Eng.*, vol. 218, pp. 175–188, Jun. 2022.

[52] F. Palacios, M. P. Diago, P. Melo-Pinto, and J. Tardaguila, "Early yield prediction in different grapevine varieties using computer vision and machine learning," *Precis. Agricult.*, vol. 24, no. 2, pp. 407–435, Apr. 2023.

[53] J. Kierdorf, I. Weber, A. Kicherer, L. Zabawa, L. Drees, and R. Roscher, "Behind the leaves: Estimation of occluded grapevine berries with conditional generative adversarial networks," *Frontiers Artif. Intell.*, vol. 5, Mar. 2022, Art. no. 830026.

[54] L. Zabawa, A. Kicherer, L. Klingbeil, A. Milioto, R. Topfer, H. Kuhlmann, and R. Roscher, "Detection of single grapevine berries in images using fully convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 1–29.

[55] L. Zabawa, A. Kicherer, L. Klingbeil, R. Töpfer, H. Kuhlmann, and R. Roscher, "Counting of grapevine berries in images via semantic segmentation using convolutional neural networks," *ISPRS J. Photogramm. Remote Sens.*, vol. 164, pp. 73–83, Jun. 2020.

[56] J. Grimm, K. Herzog, F. Rist, A. Kicherer, R. Töpfer, and V. Steinhage, "An adaptive approach for automated grapevine phenotyping using VGG-based convolutional neural networks," 2018, *arXiv:1811.09561*.

[57] W. Du and P. Liu, "Instance segmentation and berry counting of table grape before thinning based on AS-SwinT," *Plant Phenomics*, vol. 5, p. 0085, Jan. 2023.

[58] A. K. Nellithimaru and G. A. Kantor, "ROLS : Robust object-level SLAM for grape counting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 2648–2656.

[59] P. Buayai, K. R. Saikaew, and X. Mao, "End-to-end automatic berry counting for table grape thinning," *IEEE Access*, vol. 9, pp. 4829–4842, 2021.

[60] Y. Chen, X. Li, M. Jia, J. Li, T. Hu, and J. Luo, "Instance segmentation and number counting of grape berry images based on deep learning," *Appl. Sci.*, vol. 13, no. 11, p. 6751, Jun. 2023.

[61] P. Upadhyaya, M. Karkee, S. Kshetri, and A. Paudel, "Automated lag-phase detection in wine grapes using a mobile vision system," *Smart Agricult. Technol.*, vol. 7, Mar. 2024, Art. no. 100381.

[62] Y. Li, Y. Tang, Y. Liu, and D. Zheng, "Semi-supervised counting of grape berries in the field based on density mutual exclusion," *Plant Phenomics*, vol. 5, p. 0115, Jan. 2023.

[63] L. Coviello, M. Cristoforetti, G. Jurman, and C. Furlanello, "GBCNet: In-field grape berries counting for yield estimation by dilated CNNs," *Appl. Sci.*, vol. 10, no. 14, p. 4870, Jul. 2020.

[64] F. Khoroshevsky, S. Khoroshevsky, O. Markovich, O. Granitz, and A. Bar-Hillel, "Phenotyping problems of parts-per-object count," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 261–278.

[65] J. Bömer, L. Zabawa, P. Sieren, A. Kicherer, L. Klingbeil, U. Rascher, O. M'uller, H. Kuhlmann, and R. Roscher, "Automatic differentiation of damaged and unharmed grapes using RGB images and convolutional neural networks," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 347–359.

[66] S. Liu and M. Whitty, "Automatic grape bunch detection in vineyards with an SVM classifier," *J. Appl. Log.*, vol. 13, no. 4, pp. 643–653, Dec. 2015.

[67] R. Pérez-Zavala, M. Torres-Torriti, F. A. Cheein, and G. Troni, "A pattern recognition strategy for visual grape bunch detection in vineyards," *Comput. Electron. Agricult.*, vol. 151, pp. 136–149, Aug. 2018.

[68] F. Palacios, M. P. Diago, and J. Tardaguila, "A non-invasive method based on computer vision for grapevine cluster compactness assessment using a mobile sensing platform under field conditions," *Sensors*, vol. 19, no. 17, p. 3799, Sep. 2019.

[69] A. Milella, R. Marani, A. Petitti, and G. Reina, "In-field high throughput grapevine phenotyping with a consumer-grade depth camera," *Comput. Electron. Agricult.*, vol. 156, pp. 293–306, Jan. 2019.

[70] S. Liu, M. Whitty, and S. Cossell, "A lightweight method for grape berry counting based on automated 3D bunch reconstruction from a single image," in *Proc. ICRA, Int. Conf. Robot. Autom. Workshop Robot. Agricult.*, 2015, p. 4.

[71] S. Liu, X. Zeng, and M. Whitty, "A vision-based robust grape berry counting algorithm for fast calibration-free bunch weight estimation in the field," *Comput. Electron. Agricult.*, vol. 173, Jun. 2020, Art. no. 105360.

[72] W. Chen, Y. Li, Z. Tian, and F. Zhang, "2D and 3D object detection algorithms from images: A survey," *Array*, vol. 19, Sep. 2023, Art. no. 100305.

[73] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[74] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[75] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. 14th Eur. Conf.*, Oct. 2016, pp. 21–37.

[76] W. Ke, T. Zhang, Z. Huang, Q. Ye, J. Liu, and D. Huang, "Multiple anchor learning for visual object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10203–10212.

[77] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 213–229.

[78] A. Dosovitskiy, B. L., K. A., W. D., Z. X., T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Represent. Learn.*, 2021, pp. 213–229.

[79] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention*. Cham, Switzerland: Springer, 2015, pp. 234–241.

[80] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder–decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[81] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and Y. Al, "Semantic image segmentation with deep convolutional nets and fully connected crf," in *Proc. Int. Conf. Represent. Learn.*, 2015, pp. 1–26.

[82] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, “Pyramid scene parsing network,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 6230–6239.

[83] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask R-CNN,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[84] Z. Cai and N. Vasconcelos, “Cascade R-CNN: Delving into high quality object detection,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6154–6162.

[85] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang, C. C. Loy, and D. Lin, “Hybrid task cascade for instance segmentation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4969–4978.

[86] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, “Fully convolutional instance-aware semantic segmentation,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4438–4446.

[87] A. Kirillov, Y. Wu, K. He, and R. Girshick, “PointRend: Image segmentation as rendering,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9796–9805.

[88] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 936–944.

[89] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, “Path aggregation network for instance segmentation,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8759–8768.

[90] G. Gao, J. Gao, Q. Liu, Q. Wang, and Y. Wang, “CNN-based density estimation and crowd counting: A survey,” 2020, *arXiv:2003.12783*.

[91] V. Lempitsky and A. Zisserman, “Learning to count objects in images,” in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 23, 2010, pp. 1–15.

[92] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, “Single-image crowd counting via multi-column convolutional neural network,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 589–597.

[93] Y. Li, X. Zhang, and D. Chen, “CSRNet: Dilated convolutional neural networks for understanding the highly congested scenes,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1091–1100.

[94] L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, “Scalable person re-identification: A benchmark,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1116–1124.

[95] J. Peng, T. Wang, W. Lin, J. Wang, J. See, S. Wen, and E. Ding, “TPM: Multiple object tracking with tracklet-plane matching,” *Pattern Recognit.*, vol. 107, Nov. 2020, Art. no. 107480.

[96] N. Wojke, A. Bewley, and D. Paulus, “Simple online and realtime tracking with a deep association metric,” in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3645–3649.

[97] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, “Bytetrack: Multi-object tracking by associating every detection box,” in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 1–27.

[98] P. Bergmann, T. Meinhardt, and L. Leal-Taixé, “Tracking without bells and whistles,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 941–951.

[99] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, “FairMOT: On the fairness of detection and re-identification in multiple object tracking,” *Int. J. Comput. Vis.*, vol. 129, no. 11, pp. 3069–3087, Nov. 2021.

[100] B. Daviet, C. Fournier, L. Cabrera-Bosquet, T. Simonneau, M. Cafier, and C. Romieu, “Ripening dynamics revisited: An automated method to track the development of asynchronous berries on time-lapse images,” *Plant Methods*, vol. 19, no. 1, p. 146, Dec. 2023.

[101] J. L. Schönberger and J.-M. Frahm, “Structure-from-motion revisited,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4104–4113.

[102] Y. S. Woo, Z. Li, S. Tamura, P. Buayai, H. Nishizaki, K. Makino, L. M. Kamarudin, and X. Mao, “3D grape bunch model reconstruction from 2D images,” *Comput. Electron. Agricult.*, vol. 215, Dec. 2023, Art. no. 108328.

[103] M. P. Diago, J. Tardaguila, N. Aleixos, B. Millan, J. M. Prats-Montalban, S. Cubero, and J. Blasco, “Assessment of cluster yield components by image analysis,” *J. Sci. Food Agricult.*, vol. 95, no. 6, pp. 1274–1282, Apr. 2015.

[104] T. Santos, L. De Souza, A. Dos Santos, and A. Sandra, “Embrapa wine grape instance segmentation dataset–embrapa wgisd,” *Zenodo*, 2019.

[105] M. Sozzi, S. Cantalamessa, A. Cogato, A. Kayad, and F. Marinello, “WGrapeUNIPD-DL: An open dataset for white grape bunch detection,” *Data Brief*, vol. 43, Aug. 2022, Art. no. 108466.

[106] D. K. Barbole and P. M. Jadhav, “GrapesNet: Indian RGB & RGB-D vineyard image datasets for deep learning applications,” *Data Brief*, vol. 48, Jun. 2023, Art. no. 109100.

[107] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[108] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, and P. Dollar, CL Zitnick, “Microsoft COCO: Common objects in context,” in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[109] *MARS: A Video Benchmark for Large-Scale Person Re-Identification*. Cham, Switzerland: Springer, 2016.

[110] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, “Swin transformer: Hierarchical vision transformer using shifted windows,” in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9992–10002.

[111] M. Zand, H. Damirchi, A. Farley, M. Molahasani, M. Greenspan, and A. Etemad, “Multiscale crowd counting and localization by multitask point supervision,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 1820–1824.

[112] Q. Wang, J. Gao, W. Lin, and X. Li, “NWPU-crowd: A large-scale benchmark for crowd counting and localization,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 6, pp. 2141–2149, Jun. 2021.

[113] K. Wu, E. Otoo, and A. Shoshani, “Optimizing connected component labeling algorithms,” in *Medical Imaging 2005: Image Processing*, vol. 5747. Bellingham, WA, USA: SPIE, 2005, pp. 1965–1976.

[114] L. Sun, F. Gao, M. Anderson, W. Kustas, M. Alsina, L. Sanchez, B. Sams, L. McKee, W. Dulaney, W. White, J. Alfieri, J. Prueger, F. Melton, and K. Post, “Daily mapping of 30 m LAI and NDVI for grape yield prediction in California vineyards,” *Remote Sens.*, vol. 9, no. 4, p. 317, Mar. 2017.

[115] A. G. Olenskyj, B. S. Sams, Z. Fei, V. Singh, P. V. Raja, G. M. Bornhorst, and J. M. Earles, “End-to-end deep learning for directly estimating grape yield from ground-based imagery,” *Comput. Electron. Agricult.*, vol. 198, Jul. 2022, Art. no. 107081.

DAVID AHMEDT-ARISTIZABAL received the B.Sc. degree in mechatronics engineering in 2008, and the Ph.D. degree in computer vision from the Queensland University of Technology, Australia, in 2019. He is currently a Senior Research Scientist leading the Efficient Computer Vision Team, Data61, Commonwealth Scientific and Industrial Research Organisation (CSIRO), Australia. He specializes in translating research in machine learning and computer vision into practical applications for industry, including agriculture, environmental surveys, health, defense, and surveillance. He has been contributing to the production of scientific knowledge via self-investigation of original concepts and through supervision of students in the area of privacy-preserving computer vision, activity recognition, and 3D scene understanding.

DANIEL SMITH is currently a Senior Research Scientist with Data61, Commonwealth Scientific and Industrial Research Organisation (CSIRO), working on research and commercial projects that translate machine learning research into practical systems. His specialty is the research and development of sensor-based systems that have been developed for a range of sectors, including sports science, agriculture, aquaculture, manufacturing, and water management. His research interests include spatio-temporal forecasting, temporal representation learning, and neuro-symbolic architectures.

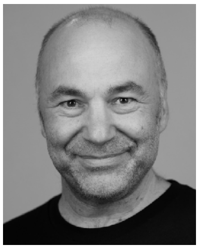
MUHAMMAD RIZWAN KHOKHER received the Ph.D. degree from the School of Electrical, Computer, and Telecommunications Engineering, University of Wollongong, Australia, in 2018. He is currently a Research Scientist with Data61, Commonwealth Scientific and Industrial Research Organisation (CSIRO), Australia, carrying out research in image and video processing, computer vision, machine and deep learning, and artificial intelligence. He is working on various commercial research and applied projects on topics of object detection in images/videos, semantic and instance segmentation, image and video classification, medical image analysis, action and behavior recognition, single and multiple object tracking, 3D image analysis, natural language processing, generative artificial intelligence, and data science.

XUN LI received the Ph.D. degree in geospatial engineering from the University of New South Wales (UNSW), Sydney, Australia. She is currently a Senior Research Engineer with the Imaging and Computer Vision Research Group, Data61, Commonwealth Scientific and Industrial Research Organisation (CSIRO), where she is dedicated to crafting AI-driven computer vision tools for addressing real-world challenges. She is also an experienced and engaged Researcher in the field of computer vision, specializing in the application of artificial intelligence to enhance visual analytics across various sectors. She has spearheaded multiple interdisciplinary initiatives over a decade, focusing on areas such as human behavior analysis, digital agriculture, and human-environment interaction. Her journey has also included significant contributions to plant phenotyping at CSIRO and research in video analysis for human behavior recognition during her postdoctoral tenure at UNSW.

ADAM L. SMITH was born in Guyra, Australia, in 1990. He received the bachelor's degree (Hons.) in mechatronics engineering from The University of Newcastle, Australia, in 2017. From 2018 to 2022, he was a Research Technician with Commonwealth Scientific and Industrial Research Organisation (CSIRO), Adelaide, involved in digital viticulture projects, with the core work as sensor deployment and a focus on computer vision. He is currently working on avionics with Gilmore Space. His other research and development experience includes "Hone," a start-up located within the HMRI facility in Newcastle, building lab equipment, and Elite Robotics (another Newcastle start-up) where he designed computer vision modules for robot navigation.

LARS PETERSSON is currently a Senior Principal Research Scientist leading the Imaging and Computer Vision Group, Data61, Commonwealth Scientific and Industrial Research Organisation (CSIRO), Australia. He is also leading one of the activities under CSIRO's Machine Learning and Artificial Intelligence Future Science Platform effort where data science problems from the smallest of microscopy scales to the largest of astronomical scales are addressed. Before joining Data61/CSIRO, he was a Principal Researcher and the Research Leader with NICTA's Computer Vision Research Group, where he was leading projects, including smart cars, automap, and distributed large-scale vision.

VIVIEN ROLLAND is currently a Senior Research Scientist and the Leader of the Crops Digital Twin Team, Commonwealth Scientific and Industrial Research Organisation (CSIRO), Agriculture and Food, where he is using machine learning and computer vision to accelerate the rate of innovation in the Agrifood sector–one example being the development of new digital tools for crop breeders.



EVERARD J. EDWARDS received the B.Sc. degree (Hons.) in plant sciences from The University of Sheffield, in 1992, and the Ph.D. degree in post-harvest physiology of potatoes from Nottingham Trent University, in 1997. He started his career in the U.K. He was a Postdoctoral Fellow with the University of York and The Australian National University, Australia, examining climate change effects on plants and plant ecology. Since 2006, he has been applying his background in whole plant physiology to perennial horticulture at Commonwealth Scientific and Industrial Research Organisation (CSIRO). Much of this has been in optimizing winegrape management, including collaborative projects with institutions, such as The University of Adelaide, Charles Sturt University, SARDI, NSW DPI, and DEDJTR Victoria.