Computers and Electronics in Agriculture 213 (2023) 108174

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Original papers

Grape yield estimation with a smartphone's colour and depth cameras using machine learning and computer vision techniques

Baden Parr, Mathew Legg *, Fakhrul Alam

Department of Mechanical and Electrical Engineering, Massey University, Auckland, New Zealand

ARTICLE INFO

Keywords:
Grapes
Yield estimation
Berry detection
YOLO
Depth camera
RGB-D

ABSTRACT

A smartphone with both colour and time of flight depth cameras is used for automated grape yield estimation of Chardonnay grapes. A new technique is developed to automatically identify grape berries in the smartphone's depth maps. This utilises the distortion peaks in the depth map caused by diffused scattering of the light within each grape berry. This technique is then extended to allow unsupervised training of a YOLOv7 model for the detection of grape berries in the smartphone's colour images. A correlation coefficient (R^2) of 0.946 was achieved when comparing the count of grape berries observed in RGB images to those accurately identified by YOLO. Additionally, an average precision score of 0.970 was attained. Two techniques are then presented to automatically estimate the size of the grape berries and generate 3D models of grape bunches using both colour and depth information.

1. Introduction

Accurate grape yield estimation is crucial for wine growers since it enables them to effectively plan, organise, and take necessary actions, such as pruning and thinning, to optimise the quality of the wine they produce. Traditionally, yield estimation has been conducted through manual techniques such as visual observation or by cutting and weighing samples, which can be subjective, destructive, and time-consuming. Moreover, manual methods can result in undersampling of the vineyard, leading to potential errors. As a result, researchers are exploring automated yield estimation methods, mainly utilising computer vision techniques (Barriguinha et al., 2021; Laurent et al., 2021; Moreno and Andújar, 2023).

Machine learning techniques have been used for detecting grape bunches in RGB (Red Green Blue) images. This has included convolutional neural networks (Santos et al., 2020) and different YOLO (You Only Look Once) models (Li et al., 2021; Zhao et al., 2022; Liu et al., 2023; Shen et al., 2023). However, for accurate yield volume estimations, it is desirable to count the number of berries within bunches and estimate the size of each berry.

Several studies have detected individual grape berries in RGB images using spectral reflectance peaks in the images obtained using artificial lighting of the grapes in controlled field or lab conditions using smartphones (Grossêtete et al., 2011; Grossetete et al., 2012; Aquino et al., 2018) and camera systems (Font et al., 2014; Mirbod et al., 2016). Machine learning has also been used to detect individual grape berries in RGB images. Coviello et al. (2020) used dilated convolutional neural networks to count grapes in smartphone images. Miao et al. (2021) used YOLOv3 to detect regions of interest around individual grapes. Additionally, a YOLO model for detecting individual grape berries can be downloaded from Ref. Roboflow Universe (2021). However, the training of these YOLO models will have been performed using manual labelling, which can be very time-consuming. Also, it would appear likely that this training would need to be repeated for different grape cultivar varieties.

For yield estimation, it is desirable to estimate the size of the individual berries within a bunch for accurate volume estimation. This is particularly the case for grape varieties that have a range of sizes within a bunch. Several studies have estimated the size of grapes and generated 3D models of grape bunches using Hough transforms to fit circles to grapes captured in camera and smartphone RGB images (Ang et al., 2018; Schmidtke, 2018; Liu et al., 2020b,a). These generally used backing boards to make the grapes more distinctive from the background and prevent circles from being detected in the background. Mirbod et al. (2016) used spectral reflectance peaks to first detect the location of each grape berry and then used circle detection in this region to estimate the size of grapes in images. Miao et al. (2021) also used a two-step process where a region of interest was identified around grape berries using a YOLOv3 model and then edge detection and ellipse fitting were used to estimate berry area in RGB images.

The size of the grape berries in an RGB image changes depending on the distance of the grapes from the camera due to perspective

* Corresponding author.
E-mail addresses: 1badenparr@gmail.com (B. Parr), M.Legg@massey.ac.nz (M. Legg), F.Alam@massey.ac.nz (F. Alam).

https://doi.org/10.1016/j.compag.2023.108174
Received 7 June 2023; Received in revised form 17 August 2023; Accepted 22 August 2023
Available online 6 September 2023

projection. One technique used to estimate the physical size of a grape berry from an RGB image is to place an object of known size next to the grapes. The size of an individual grape berry can then be obtained by comparing the size of the berry with the size of the reference object in the RGB image. Ang et al. (Ang et al., 2018; Schmidtke, 2018; National Wine and Grape Industry Centre, 2019) used a disk of known dimensions placed among the grapes or a checkerboard held next to the grapes to estimate the physical size of the grapes in a smartphone's or regular camera's images. Liu et al. (Liu et al., 2020b,a) also used a checkerboard image for this purpose. This allowed them to model the 3D structure of a grape bunch from 2D images. This process was extended by Xi et al. (Xin et al., 2020; Xin and Whitty, 2022) to include constraint-based reconstructed grammars to "grow" the full 3D grape bunch structure from a single view 2D image.

The distance that a camera is from a checkerboard can be measured using the camera's intrinsic calibration parameters, which can be obtained using camera calibration software. This technique may have been used in the above works that used checkerboards.

It is desirable, however, not to have to use a reference object for estimating the size of grape berries for yield estimation. The physical size of grape berries can be estimated from their sizes in an RGB image if one knows the distance of the camera from the grapes when the image was taken, and one knows the camera's calibration intrinsic parameters. Ivorra et al. (2015) were able to estimate the size of grapes from RGB images without the need for a calibration object. They achieved this by measuring the distance that the camera was from the grapes using a stereo-depth camera. They used this distance combined with the size of grapes in the stereo camera's raw RGB images to estimate the physical size of grape berries. However, these results were obtained in controlled lab environments where the lighting, background, and camera position were carefully regulated, and manual refinement was required.

Grape size estimation and 3D modelling of grape bunches have also been performed using high-resolution 3D scans of grapes. This has included the use of photogrammetry. However, this involves a high computational load and can take significant time to process (Rose et al., 2016). Stereo reconstruction has also been used to generate 3D models of grape bunches (Herrero-Huerta et al., 2015). There has also been work using commercial high-resolution 3D scanners to generate 3D models of grapes in lab environments (Schöler and Steinhage, 2015; Mack et al., 2018). However, these are expensive and do not seem suitable for practical use by farmers in the field.

There have been several studies that have used low-cost depth cameras to obtain 3D scans of grapes obtained using RGB-D (Red Green Blue - Depth) cameras for grape yield estimation. Marinello (Marinello et al., 2016) and Hacking (Hacking et al., 2020; Hacking, 2020) used the Microsoft Kinect V1 depth camera for yield estimation studies of grapes. This operates using infrared structured light, which did not work well in sunlight conditions due to the saturation of the projected infrared (IR) pattern. Kurtser et al. (Kurtser et al., 2020a,b) used an Intel RealSense D435 RGB-D camera for 3D scanning of grapes, which uses active stereo. These works did not measure individual berry information. This is likely due to the relatively low resolutions of the RGB-D cameras used.

Parr et al. (2022) compared the performance of several low-cost depth cameras for imaging grapes. It was shown that the ToF (Kinect V2 and Kinect Azure) and LiDAR (Intel RealSense L515) depth cameras produced distortions in the 3D scans of individual grapes in the form of peaks centred on each grape location due to diffused scattering within the grapes. It was suggested that these distortions could be exploited to make the detection of grapes in ToF depth scans easier.

Previous research has employed smartphones to investigate grape yield estimation (Tardaguila et al., 2021; Liu et al., 2020a,b; Grossêtete et al., 2011; Schmidtke, 2018; Aquino et al., 2018). An advantage of employing a smartphone in this context is that the majority of individuals already own one, thereby obviating the necessity for growers to invest in additional equipment. Many modern smartphones have built-in depth cameras in addition to RGB cameras. For example, the Samsung Galaxy Note 10+, Samsung Galaxy S20 Ultra, Huawei P30 Pro, etc. have built-in Time of Flight (ToF) cameras and the iPhone 12, 13 and 14 Pro and Pro Max models have built-in LiDARs. We are not aware of any previous works that have used the built-in depth cameras of a smartphone for grape yield estimation applications.

In this study, we utilise a Samsung Note 10+ smartphone to capture RGB images and ToF depth maps of Chardonnay grapes in field and lab environments. Grape detection is performed automatically by identifying distortion peaks in the ToF depth maps resulting from diffused light scattering within the grapes. We further train an unsupervised YOLOv7 model to detect the precise location of grape berries in RGB images, leveraging the initial grape identification from the depth maps. Additionally, we develop techniques to estimate the size of grape berries and generate 3D models of grape bunches.

This article has the following contributions.

- We introduce a novel technique for the automatic detection of grape berry locations in 3D based on the peaks observed in the ToF depth maps captured by the smartphone. Building upon this technique, we extend it to enable unsupervised training of a YOLOv7 model for grape berry identification. To the best of our knowledge, this is the first instance of unsupervised training of a YOLO model specifically for grape detection, and we are not aware of any previous work that has employed a similar approach.
- The physical size of grape berries can be estimated from their size in the smartphone's RGB images using the distances from the camera to the grape berries that are automatically measured by the smartphone's depth camera. This removes the need for placing a calibration object next to the grapes, as has been done in previous work related to estimating berry size from RGB images captured in the field.
- A novel iterative modelling technique is introduced for estimating the sizes of grape berries based on their detected 3D positions, eliminating the need to estimate berry sizes from the RGB images. This approach offers an alternative method that does not rely on analysing the RGB images to determine the berry sizes.

The remainder of the paper is organised as follows. Section 2 outlines the data collection methodology and processing used to generate RGB-D point clouds of grapes. The technique used to detect individual grapes from depth scans is described in Section 3. In Section 4, a technique used to train a YOLO model in an unsupervised manner is outlined. This model is then used to detect grape berries in the RGB images. The methods used to estimate the size of grape berries and perform 3D modelling of grape bunches are then presented in Section 5. Finally, the conclusion is presented in Section 6.

2. Methodology

2.1. Data collection

Field measurements were made of Chardonnay grapes at the Villa Maria Estate in Auckland, New Zealand. These were performed about two weeks before harvest (late February). A Samsung Note 10+ smartphone was used to perform measurements on the grapes. This smartphone contains an RGB camera and a Time of Flight (ToF) depth camera. For each depth image, it also generates a confidence map, which provides an indication of the accuracy and validity of each point in the depth map.

At the time, there was no app available to capture depth map images from this camera. Therefore, a custom Android application was developed for this purpose. For each capture event, the application automatically saved to file a 4032 × 3024 RGB image, a 640 × 480 depth map, and a time synchronised 640 × 480 confidence map.

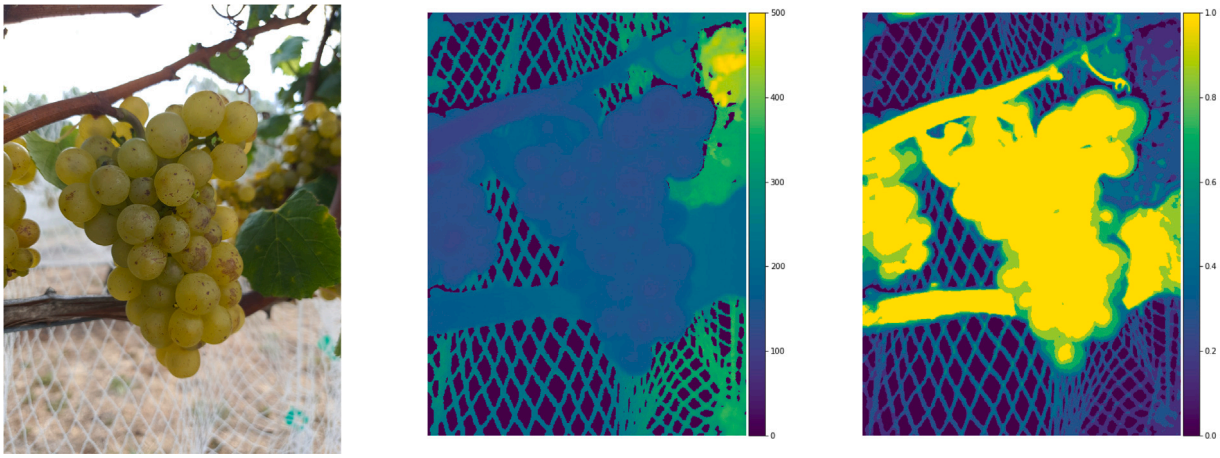


Fig. 1. Example images of the (a) RGB, (b) depth and (c) confidence maps captured by the Samsung Note 10+ of grapes in the field.

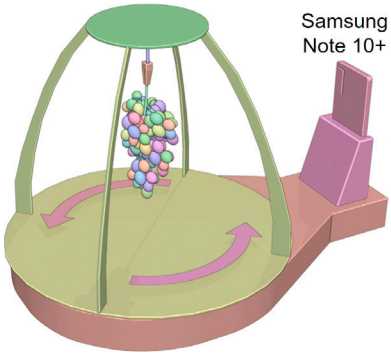


Fig. 2. Diagram of the experimental setup where a turntable was used to capture images of a grape bunch using the smartphone from a range of angles.

Additionally, a text file was also saved that contained the smartphone's GPS location and a reading from the smartphone's accelerometer taken at the time of capture.

Fig. 1 provides an example of the RGB, depth, and depth confidence maps captured using this app for a grape bunch. (This grape bunch data will be used in most examples presented in this work for consistency.) The camera was able to capture depth maps in direct sunlight. No direct effort was made to take captures at any predetermined distance from the graph cluster. The only restriction was that each grape cluster should ideally fill the camera's frame. In total, 400 sets of images were captured of unique grape clusters throughout the vineyard.

In order to build a YOLOv7 machine-learning model to identify grapes, a large number of scans of grapes were needed. To achieve this, 34 representative grape bunches were harvested from the vines and taken back to the lab. In turn, each grape bunch was suspended from a computer-controlled rotation table located 200 mm from the optical centre of the stationary Samsung Note 10+, see Fig. 2. This distance was chosen to ensure that all grape bunches would fit within the camera's frame while being as close as possible. This methodology imitates our typical use of the phone's cameras when capturing images of bunches located on the vine.

The grape bunches were rotated through 360° and the Samsung Note 10+ was used to capture an RGB image and depth and confidence maps at 10° degree increments. Angles were not included where the structure of the rotation platform obscured the grape cluster. This resulted in a total of 1062 images of 34 grape bunches taken at a range of angles. Refer to Fig. 3 for examples of scans captured using this technique. Additionally, 120 scans were taken from a range of angles in the lab of a potted grapevine, absent of grapes.

2.2. Camera calibration

The Note 10+ cameras produced an RGB image and depth and confidence maps. In order to generate 3D-coloured depth point clouds (RGB-D) from these, the calibration parameters of the smartphone's cameras needed to be known. The smartphone's API did have calibration parameters stored. However, slight errors were found when using these to align the colour and depth maps when calculating the RGB-D point cloud. Therefore, a series of calibration colour and depth images were taken from a range of angles of a checkerboard pattern that was glued onto a sheet of acrylic. These measurements were made at similar distances from which the grape measurements were made. The black ink used to print the checkerboard pattern absorbed the infrared light emitted by the ToF camera meaning it showed up as voids (black) on the depth map. This meant that the depth map images captured of the checkerboard could be used with camera calibration software.

The checkerboard images captured by the depth and colour cameras were separately calibrated using OpenCV v4.7.0. For this process, the RGB images were downsampled to be the same 640 × 480 resolution as the depth maps before calibration. This was done to reduce the processing burden and ease stereo registration. This 640 × 480 resolution will be used for RGB and depth images throughout the remainder of this work. Refer to Fig. 4 for examples of corresponding depth and colour images obtained during this calibration with a common "real-world" reference frame shown. The estimated intrinsic parameters for both the colour and depth images were used along with the detected checkerboard coordinates for stereo calibration to obtain the extrinsic parameters defining the transformation from the RGB camera's reference frame to that of the depth camera.

2.2.1. Projecting between depth map and RGB images

Consider a pixel in the depth map with 2D coordinates $\bar{p}_d$ in the X and Y axes directions, which has a depth value of Z. One can convert this into a 3D coordinate in the depth camera reference frame using

$$\bar{X}_d = \begin{bmatrix} Z\ (\bar{p}_d[1] - c_{d1})/f_{d1} \\ Z\ (\bar{p}_d[2] - c_{d2})/f_{d2} \\ Z \end{bmatrix}, \quad (1)$$

where $\bar{p}_d[1]$ and $\bar{p}_d[2]$ are respectively the pixel coordinates in the X and Y axes directions. Similarly, f_{d1} and f_{d2} are the depth camera's focal lengths in the X and Y axes directions, c_{d1} and c_{d1} are the coordinates of the central depth pixel in the depth map.

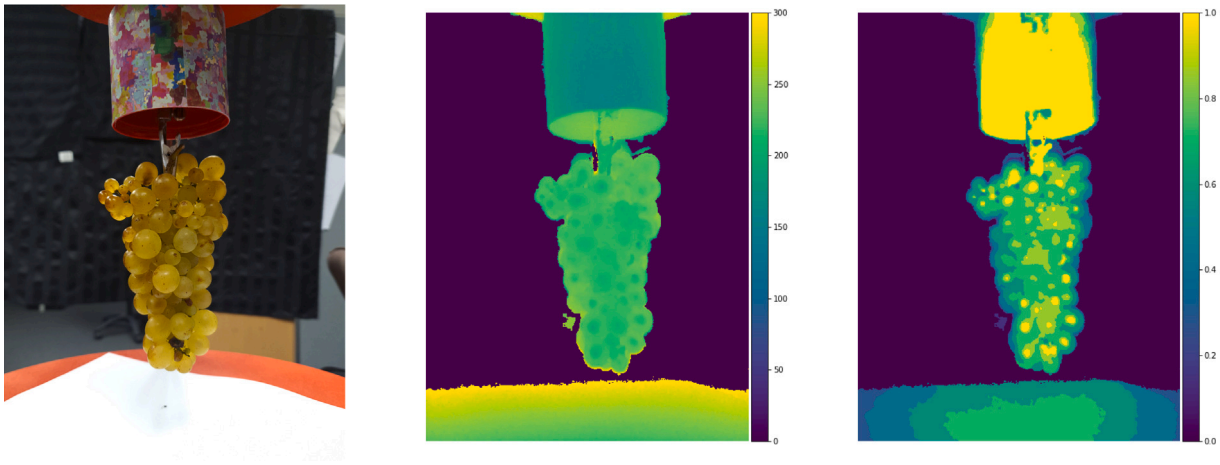


Fig. 3. Examples of (a) an RGB image and (b) depth and (c) confidence maps captured by the Note 10+ of a grape bunch in the lab on a turntable. These were used to train a YOLO model to detect individual grapes.

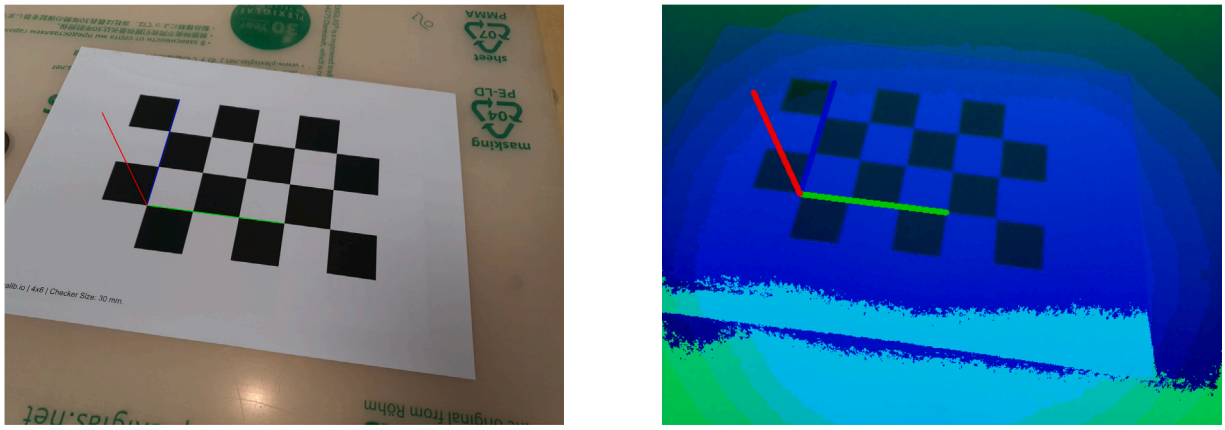


Fig. 4. Examples of the (a) RGB and (b) depth calibration images captured by the smartphone's cameras.

This 3D point $\bar{\boldsymbol{X}}_d$ can be moved from the depth camera's reference frame to the RGB camera's reference frame using the ridged body transformation

$$\bar{\boldsymbol{X}}_c = \boldsymbol{R}\ \bar{\boldsymbol{X}}_d + \bar{\boldsymbol{T}}, \tag{2}$$

where $\boldsymbol{R}$ is the stereo calibration rotation matrix and $\bar{\boldsymbol{T}}$ is the stereo translation vector.

This 3D point can be coloured by finding the colour of the corresponding pixel in the RGB image. The 3D point $\bar{\boldsymbol{X}}_c$ is first converted to normalised coordinates using

$$\bar{\boldsymbol{x}}_c = \begin{bmatrix} \bar{\boldsymbol{X}}_c[1]/Z \\ \bar{\boldsymbol{X}}_c[2]/Z \end{bmatrix}, \tag{3}$$

where $\bar{\boldsymbol{X}}_c[1]$ and $\bar{\boldsymbol{X}}_c[2]$ are respectively the X and Y axes components of $\bar{\boldsymbol{X}}_c$. This can be then converted into pixel coordinates on the RGB image using

$$\bar{\boldsymbol{p}} = \begin{bmatrix} f_{c1}\ \bar{\boldsymbol{x}}_c[1] + c_{c1} \\ f_{c2}\ \bar{\boldsymbol{x}}_c[2] + c_{c2} \end{bmatrix}, \tag{4}$$

where f_{c1} and f_{c2} are the RGB camera's focal lengths in the X and Y axes directions and c_{c1} and c_{c2} are the coordinates of the central pixel in the RGB image. No corrections were made for lens distortion or skew. The colour of this pixel can be used as the colour of the 3D point in either the RGB or depth camera's reference frames.

By repeating the above process for all pixels in the depth map, a coloured 3D point cloud can be generated. However, due to the perspective shift in some situations, multiple depth pixels will map to the same colour pixel. In this situation, only the point closest to the camera should be retained. Refer to Fig. 5 for an example of the RGB image and the corresponding 3D coloured point cloud obtained using this method.

3. Detection of berries in the ToF depth scans

Fig. 6 shows an example of a ToF camera scan of a single grape before and after it has been sprayed with an opaque coating (AESUB 3D Scanning Spray). This illustrates how the diffused scattering of light within the grapes causes a distortion of the shape of the grape in the 3D scan. This manifests as a distinctive peak centred at the location of each grape. Observing this effect led to the idea that these peaks could potentially be used to facilitate the automatic detection of individual grape berries in ToF depth images (Parr et al., 2022).

Fig. 7 shows a block diagram of the technique used to investigate this idea. Each depth map captured by the smartphone ToF camera was filtered to reduce noise using the corresponding confidence map. Depth pixels that had a confidence value of less than 50% were removed. This had the primary effect of removing distant points. In all cases, the camera presented high confidence for pixels representing the grapes'

(a) (b)

Fig. 5. Image (a) shows an example of the RGB photo captured by the smartphone's camera of a grape bunch in the field. Image (b) shows the corresponding colourised depth map.

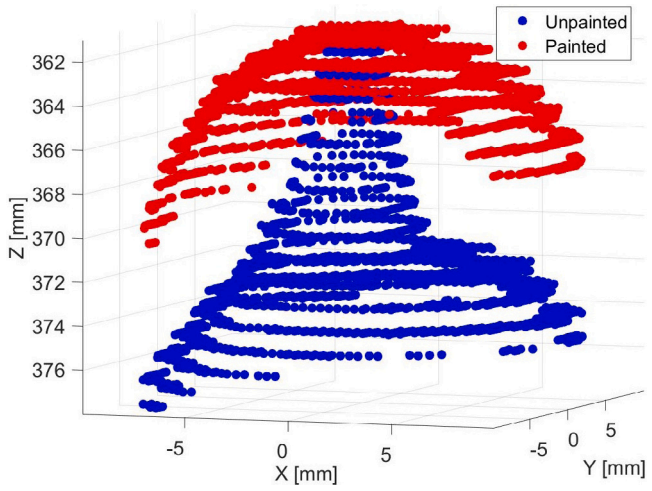


Fig. 6. Example plot showing a peak in the depth map due to a grape that has been converted to 3D point cloud before and after it had been sprayed by an opaque coating. This illustrates how diffused scattering within the grape berries causes distortion of the depth scan in the form of peaks.

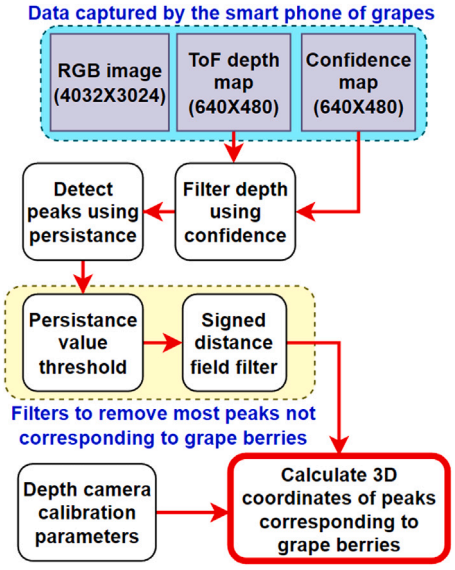


Fig. 7. Block diagram showing the technique used for calculating the 3D coordinates of peaks in the depth map corresponding to grapes.

surface. This 50% threshold was empirically determined from analysis of several images. Increasing this threshold caused the edges of grape clusters to erode slightly. Meanwhile, reducing the threshold caused background objects to be included and resulted in low persistence peaks to be detected due to the noise.

To identify potential grape locations, a peak detection algorithm was then used to identify peaks in the depth maps. A persistence homography technique (Huber, 2021, 2022) was utilised for this due to its speed and robustness to noise. The persistence homography technique generated a persistence value for each identified peak, representing how significant a local maxima peak is in comparison to other local peaks.

Fig. 8(a) shows an example depth image of a grape cluster in the field, with the peaks detected by the persistence algorithm overlaid as white crosses. The algorithm is capable of detecting peaks that correspond to individual grapes. However, it also identifies peaks that correspond to the edges of grapes, leaves, stems, and netting. Fig. 8(b) shows the Signed Distance Field (SDF) of this depth map, which was generated from a binary thresholded version of the depth map. This is utilised to remove peaks near edges by disregarding peaks that are closer than 7 pixels to an edge, as shown in Fig. 8(c) and (d). This threshold was chosen empirically to ensure only peaks close to the edge were removed and not those that may belong to small grapes. Future work will need to explore methods for scaling this threshold according to distance from the camera.

Manual analysis was performed for 50 of the scans of the grapes captured in the field. The total number of grapes visible in the images was manually counted. After which, peak locations were manually

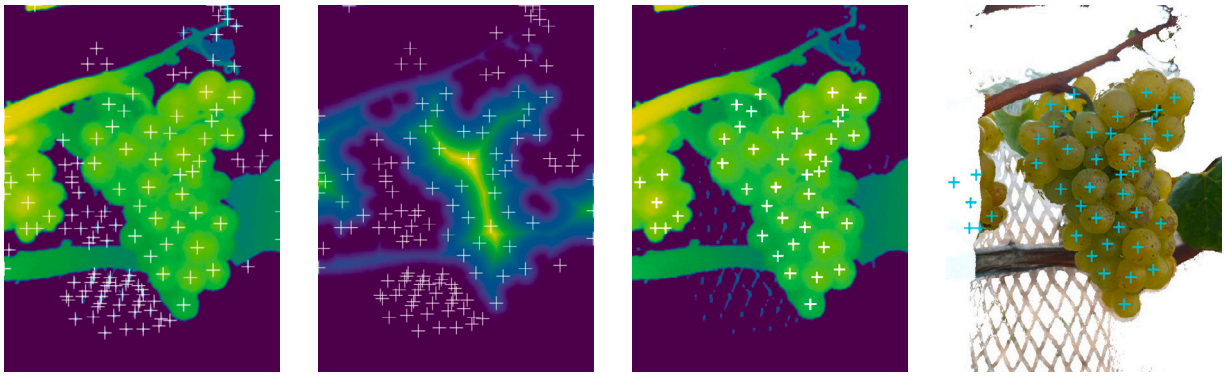


Fig. 8. These plots show the process of peak detection of a depth image captured in the field for the grape bunch shown in Fig. 1. Plot (a) shows the peaks (white crosses) detected using persistence. Many peaks have been found on the netting in the background. In Plot (b), these peaks are shown over the generated signed distance field. Plot (c) shows the resulting peaks after signed distance field filtering was used with the aim of removing peaks not corresponding to grapes. Plot (d) shows these filtered peaks overlaid on the colourised depth map.

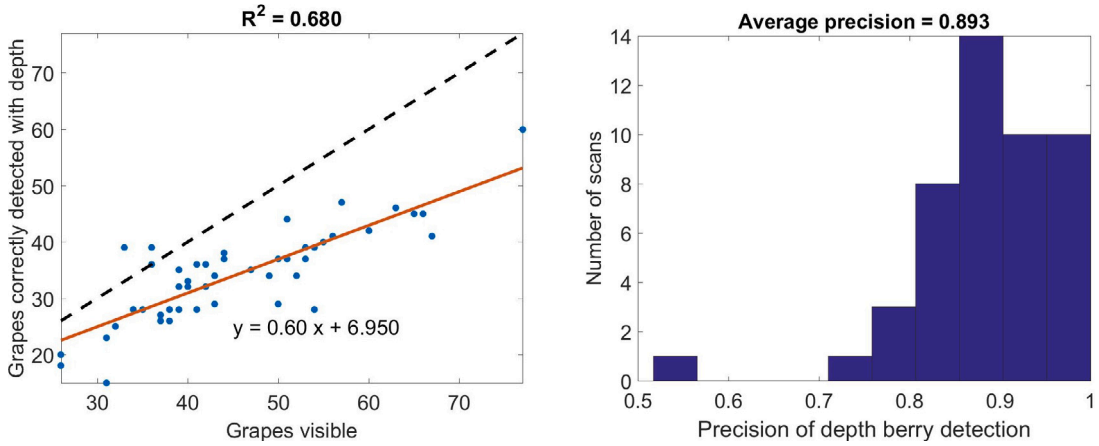


Fig. 9. Plot (a) shows the relationship between the number of grapes correctly detected using peak detection in the depth maps relative to the total number of grapes counted manually in the corresponding RGB images. The identity line is shown as a dotted line and the line of best fit is shown in orange. Plot (b) shows a histogram of the precision.

checked to see how many of the peaks corresponded to grapes and how many did not. Fig. 9(a) shows a plot of berries correctly detected (true positives) by the peak detection in depth maps relative to the total number of grapes visible in the corresponding RGB images. An R^2 value of 0.680 was obtained for the linear fit through this data. It can be seen that the technique underestimates the total number of grapes. Some grapes on the edge of the cluster were not detected, presumably because the centres of those grapes were occluded and therefore did not manifest as distinctive peaks in the depth map. In some cases, decreasing the SDF filtering threshold might result in an increase in the number of peaks being detected at the edges of the bunch. However, this will lead to an increase in the detection of peaks caused by other objects, such as leaves and netting, being erroneously identified as grapes.

The algorithm has been effective in eliminating most of the peaks that did not correspond to grapes. However, some incorrect peaks were detected, such as those corresponding to the peduncle between berries and on the rachis. Fig. 9(b) shows a histogram of the precision. The precision is calculated for each scan as the number of grapes correctly detected by the peak detection (true positives) divided by the total number of peaks identified as grapes (true positives plus false positives). An average precision of 0.893 was achieved.

The depth peak detection technique showed promise for automatically detecting grapes. However, it showed some limitations as described above. Work was therefore performed to investigate whether improved berry detection performance could be achieved by utilising the corresponding RGB images. This work is described in the following section.

4. Detection of individual berries in the RGB images

For this work, the popular YOLOv7 object detection model was chosen to facilitate the detection of grapes in smartphone's RGB images (Wang et al., 2022). This selection was based on its well-established performance for object detection in complex images, as well as its pre-trained weights and open-source code that simplifies training new classes (Wang, 2022). Training of a YOLO model requires images labelled in the form of bounding boxes around the object that the model is being trained to detect. This is traditionally done through supervised training; a process of manually selecting the bounds that encompass each instance of the object in question within an image. This process can be time-consuming, particularly for grape berry detection, which would require selecting individual grape berries in a large number of images, and would need to be repeated for distinct grape varieties (Ciarfuglia et al., 2023). Therefore, an automated technique was sought to perform unsupervised training utilising grapes detected through depth maps. The block diagram shown in Fig. 10 illustrates the technique used to investigate this idea.

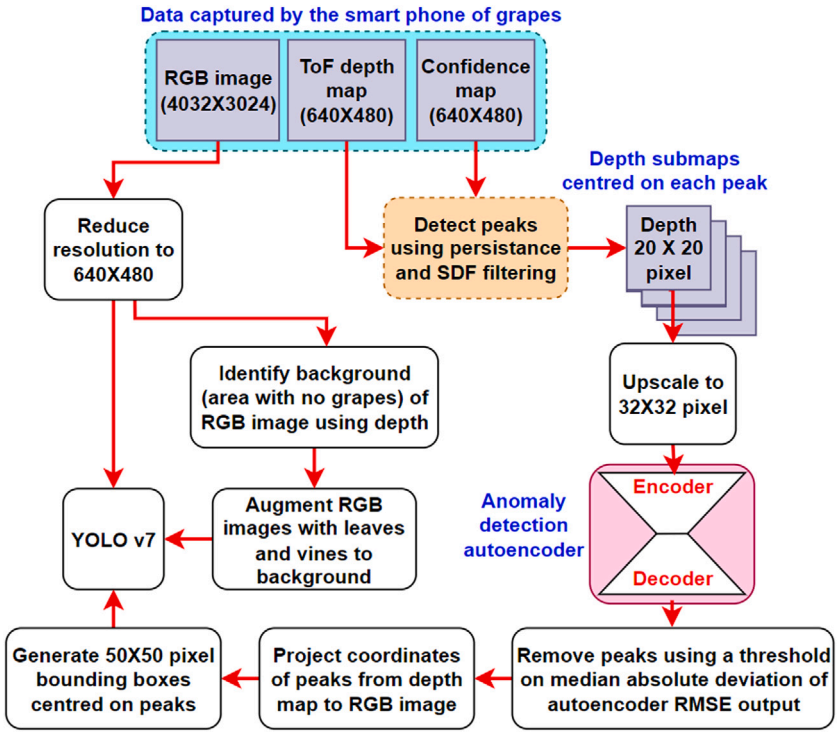


Fig. 10. Block diagram of the technique used to perform unsupervised training of a YOLOv7 model for detection of individual berries in the RGB images using the estimated 3D coordinates from the depth maps.

4.1. Dataset used for YOLO training

To reduce the potential of using false positives when automatically generating the bounding boxes, the scans of grapes captured in the lab were used for training, see Fig. 3. Due to the controlled environment, the grape clusters could more easily be isolated from the image. The RGB images were downsampled to have the same 640 × 480 pixel resolution as the depth maps.

4.1.1. Bounding box generation using depth map data

To automatically generate bounding boxes in the RGB images used for YOLO training, the corresponding depth maps were employed. Firstly, the depth maps were filtered to isolate the grape clusters by removing points that were more than 300 mm away from the camera. This was chosen as the grapes were suspended 200 mm from the camera, and thus anything captured beyond 300 mm did not belong to the grape bunch. Next, the same technique explained in Section 3 was employed to detect peaks in the depth maps that corresponded to grapes. The confidence map with a threshold was applied to filter the depth map, following which the persistence algorithm was utilised to detect peaks. Finally, the signed distance field was used to eliminate peaks that were too close to the edges of the grape bunch.

4.1.2. Autoencoder based outlier rejection

As discussed in Section 3, the peak detection technique described would occasionally detect peaks that did not correspond to grapes. Inspection of these peaks showed they often related to the peduncle visible between berries or on the rachis where the clusters were hung. In each case, the erroneous peaks had significantly different profiles than the true positives, which themselves had relatively uniform shapes. See Fig. 12 for examples of both cases. These false positives could influence the YOLO training and it was felt that a machine-learning technique could be used to detect these anomalies and filter out peaks that may not correspond to the centres of grape berries.

It was decided that an autoencoder would be used to identify peaks in the depth map that may not correspond to grapes. This decision was based on the idea that the autoencoder would be able to learn information about the shape of different grape peaks, such as scaling factors and symmetries, making it effective for identifying outliers (Zhu et al., 2016).

To ensure efficient training and minimise overfitting of potential outliers in the training set, the autoencoder's latent space was intentionally reduced in size. This reduction also aimed to prevent excessive complexity without generating artefacts in the reconstructed images.

The autoencoder was implemented using TensorFlow in Python. Fig. 11 shows a block diagram of the autoencoder used in this work. The model consists of two parts: the encoder and the decoder. The encoder maps the input image to a lower-dimensional representation, while the decoder maps the encoded representation back to the original image.

To feed the autoencoder, a 20 × 20 pixel sub-map was taken from the depth map centred on the location of a detected peak, see Fig. 12. This size was empirically chosen to be large enough to capture the majority of a peak's surface but not so large that it gets conflated by the surface of neighbouring grapes. In total, 33,844 of these sub-maps were generated and used to train the autoencoder. For convenience, each sub-map was then upsampled to a 32 × 32 resolution to make it a power of two suitable for use with the autoencoder.

The encoder takes the input image of size 32 × 32 × 1 and applies three convolutional layers with 8, 4, and 2 filters respectively, each using a 4 × 4 kernel, stride of 2, and a ReLU activation function. This was designed to reduce the image size by half in subsequent layers, creating an effective encoding funnel for dimensionality reduction without relying on large dense layers.

The decoder takes the encoded representation as input and reconstructs the original image. The decoder starts with a fully connected layer with 32 units, followed by a reshape layer that transforms the output into a 4 × 4 × 2 tensor. Then, three transposed convolutional layers with 4, 8, and 1 filters, respectively, each using a 4 × 4 kernel, a stride of 2, and the ReLU activation function, are applied to the tensor. The last transposed convolutional layer has a sigmoid activation function, which maps the output to values between 0 and 1, representing

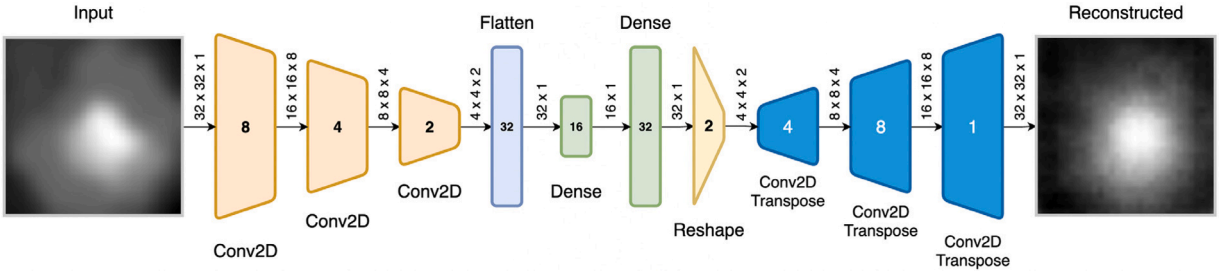


Fig. 11. Block diagram of the autoencoder convolutional neural network.

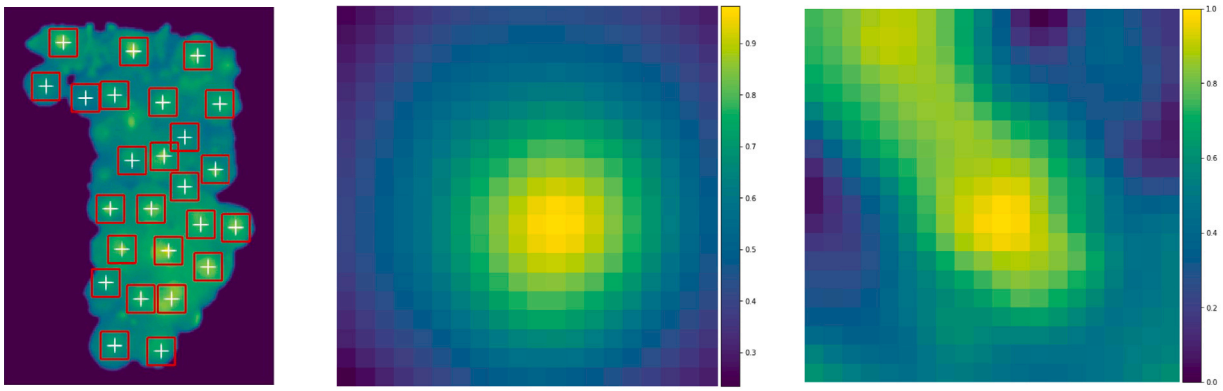


Fig. 12. Plot (a) shows the 20 × 20 pixel sub-maps shown as red boxes surrounding the detected peaks in the depth map that are used as the inputs of the autoencoder. Plot (b) shows the average sub-map of the training set. Plot (c) shows an example of an erroneous sub-map relating to the peduncle visible in a cluster.

the pixel intensities of the reconstructed image. The model takes the encoded representation as input and produces the reconstructed image as output. This model is trained using mean squared error as the loss function between the original image and the reconstructed image.

Through empirical evaluation, the above architecture demonstrated optimal performance given the defined constraints and objectives. Decreasing the number of filters in each of the convolutional and transpose convolutional layers caused noticeable blocky artefacts in the reconstructions. Similarly, reductions in the latent space size (e.g., from 16 × 1 to 8 × 1) resulted in reconstructed images that exhibited similarity regardless of the input shape. These observations informed the decision to strike a balance between reducing complexity and preserving image fidelity. Future work will involve exploring alternative architectures to identify optimal designs.

After training, the MSE value generated by the autoencoder can be used with a threshold to classify if a peak corresponds to a grape or some other object (an anomaly). This threshold was determined by assessing the distribution of MSE scores of every sub-map in the dataset and filtering using the Median Absolute Deviation (MAD). The median of all scores was computed, and then the distance to this median was computed for all sub-maps. The threshold was set to two times the median of these distances, see Fig. 13. This allowed the autoencoder to be used as a strong filter to remove potential outlier peaks that might not correspond to the centre of the grapes.

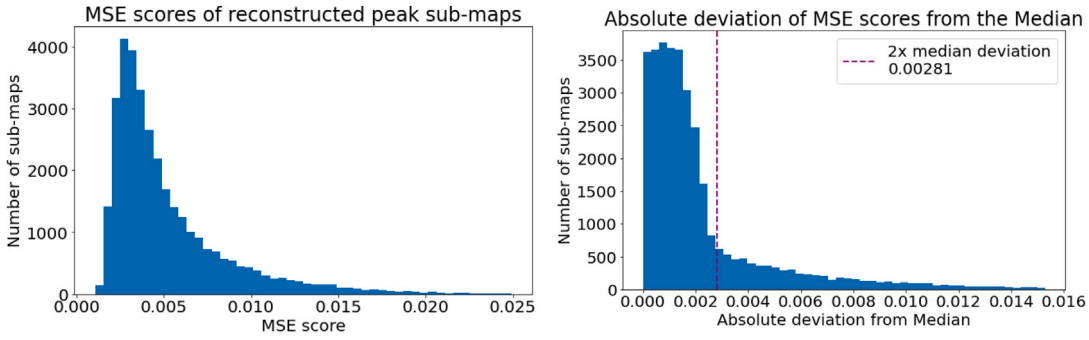
The peaks remaining after the above filtering had been performed were then used to automatically generate bounding boxes in the RGB images for YOLO training. The coordinates of the peaks in the depth map were converted to coordinates in the corresponding RGB images using the stereo calibration parameters. Bounding box coordinates in the RGB image were then calculated using a 50 × 50 pixel square centred on the calculated peak location. This size was chosen to ensure that the grape was completely encompassed. Additionally, a second class label and bounding box were generated for the entire grape cluster based on the overall bounds of the detected grapes and an additional margin of 40 pixels.

4.1.3. Background augmentation

A limitation of the lab-collected data set was that it did not include any images of leave or stems in them. This would have resulted in the YOLOv7 model not being applicable to the field trails. To address this, for each of the original turntable RGB images, two additional background-augmented images were added to the training data set. Each augmented RGB image was generated by taking an original turntable RGB image, isolating the grape bunch from the backgrounds using the depth map information, and overlaying the extracted grape bunch image over an RGB image captured of a grapevine randomly selected from a set of 120 images. The labels for the original source image were directly applied to these augmented images as the grape cluster itself remained unchanged. Examples of the two resulting images for one particular source image are shown in Fig. 14.

4.2. YOLO training

The dataset used to train and test the YOLOv7 model consisted of 3186 images labelled with grapes and grape clusters. The dataset was split into a training set (60%), a validation set (20%) and a test set (20%). The training process followed the method described in the official repository (Wang, 2022). The default configuration parameters were used, and the training process was initiated with pre-trained weights provided in the official repository as “yolov7.pt”. To keep memory requirements low, a batch size of 8 was used for training. The training process was run for a total of 20 epochs, and although more epochs were explored, no significant improvement was observed. The training process was completed in 0.615 h using an Nvidia RTX 3090. Refer to Fig. 15 for plots of the training results.



(a) (b)

Fig. 13. Plot (a) shows the distribution of mean square error (MSE) scores of all peaks that make up the training set for the autoencoder. Plot (b) shows the autoencoder's distribution of absolute deviations from the median along with the threshold used when applying these scores as a filter.

(a) (b)

Fig. 14. Images of grapes captured on the turntable in the lab with images of leaves and vines added to the background.

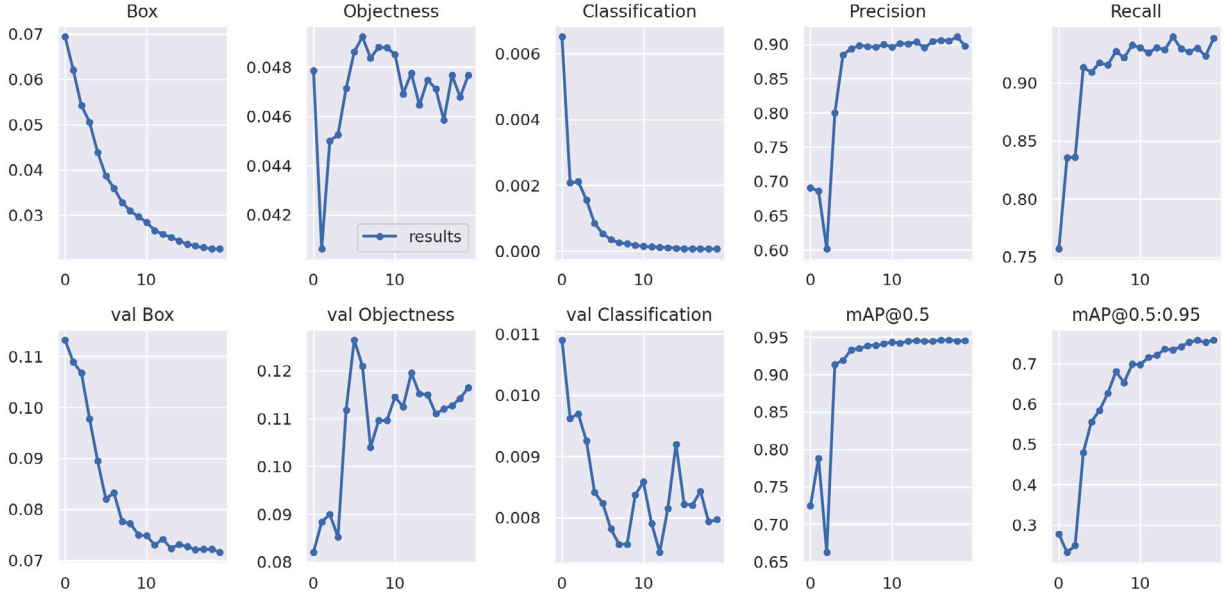
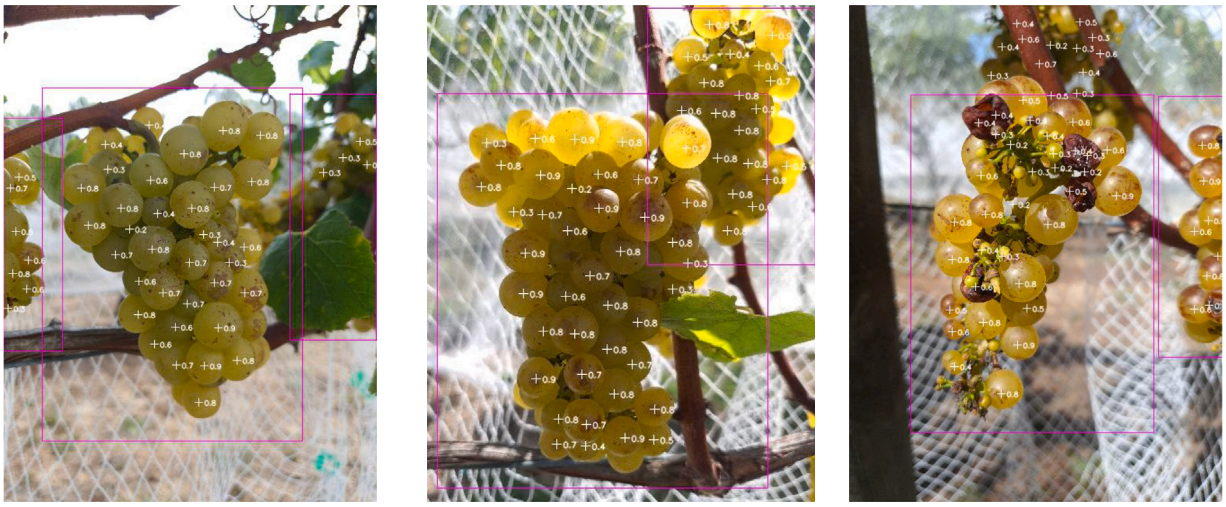


Fig. 15. Results of YOLOv7 training over 20 epochs.



(a) (b) (c)

Fig. 16. Example photos from the field of grapes with YOLO detection of individual grapes berries overlaid as white crosses and confidence values. Also shown as magenta boxes is the YOLO detection of grape bunches. Plot (c) presents one of the more challenging images in the dataset where multiple withered grapes are visible and the trunk has been incorrectly labelled.

4.3. YOLO results

The trained YOLO model was utilised to detect grapes in the RGB images captured in the field. Fig. 16 presents examples of the detected grapes using the trained model. Most visible grape berries are accurately identified, although detection accuracy diminishes for out-of-focus grape bunches in the background. Additionally, some grape berries at the edge of the bunch remain undetected. Incorrect detection of netting, vines, or leaves as berries in the background also occur. Another issue arises when withered grapes are mistakenly identified as multiple grapes, as seen in Fig. 16(c). This can be attributed to the absence of withered grapes during the YOLO model training process.

To evaluate the performance of YOLO for detecting individual berries, 50 field trial RGB images were randomly selected for manual analysis. (Note these RGB images corresponded to the same depth maps used for manual analysis of the depth peak detection technique shown in Fig. 9.) These had a grape bunch centred in the image. Other grape bunches in the background were ignored in the analysis since generally either only a part of these secondary grape bunches could be seen or they were out of focus in the RGB images. Manual counting was then performed for the central grape bunch of the number of berries correctly and incorrectly detected by the YOLO model. These were then compared with the total number of grapes able to be manually counted in the grape bunch.

Fig. 17(a) compares the number of berries correctly detected using the YOLO model (true positives) to the total number of berries visible for each of the main grape bunches. There is a systematic underestimation of the number of berries counted using YOLO. Observations suggest that this is mainly due to missed grapes around the outside of the grape bunch, many of which have only a fraction of a berry visible. The fit through the data has a R^2 value of 0.946 and shows an increasing deviation from the one-to-one line as the number of berries in the cluster increase.

Fig. 17(b) shows a histogram of the precision. The precision for each scan is calculated from the number of berries counted by YOLO (true positives) divided by the sum of the total number of berries detected by YOLO (true positives + false positives). An average precision of 0.970 was achieved. The number of false positives within the bounded box selected by YOLO as the main grape bunch was 2.9% of the total number of visible grapes in the main bunch with only 12% of the images having more than 3 false positives.

4.4. Location of YOLO detections in 3D

The process of projecting the detected grape locations in an RGB image into 3D space is achieved by reversing the mapping process discussed in Section 2.2.1 to identify the closest corresponding depth pixel coordinate. However, due to differences in perspective and the way peaks align with the direction of measurement, these projected locations do not necessarily correspond to peaks in the point cloud. As seen in Fig. 6, the peaks are the closes to the true surface of the grape. Therefore, using points from other areas on the surface will lead to significant errors in depth and subsequent estimated grape location.

To address this, a gradient descent technique was used to move the detected grape locations to the peaks in the depth map before projection. (Note that "gradient descent" is used rather than "gradient ascent" since the peaks were towards the camera and hence had lower depth values.) Specifically, the depth map scan of the grape bunch was filtered by removing pixels with corresponding confidence map values less than 50% and smoothed using a 5×5 pixel sliding average kernel. This reflects the confidence thresholding used earlier. An iterative gradient descent technique was then employed to move the grape locations to the top of the peaks. The 5×5 pixel filter was selected empirically to provide suitable noise reduction ensuring the descent will not get stuck in small local minima but also retains definition so that individual peaks can be found. The effectiveness of this technique in improving the accuracy of grape location detection is illustrated in Fig. 7.

The gradient descent algorithm adjusts the depth pixel location iteratively using the following formula:

$$\bar{p}_{i+1} = \bar{p}_i - \alpha \nabla \boldsymbol{J}(\bar{p}_i) \tag{5}$$

where $\bar{p}_i$ is the pixel coordinate at the i_{th} iteration, $\alpha = 0.5$ is the traversal rate, and $\nabla \boldsymbol{J}(\bar{p}_i)$ is the gradient of the smoothed depth map $\boldsymbol{J}$ evaluated at coordinate $\bar{p}_i$. This traversal rate was chosen empirically due to its stability and rate of convergence. Values significantly greater than this caused instabilities and values smaller caused convergence to take longer.

The gradient of the smoothed depth map with respect to the X and Y axes is computed as follows:

$$\frac{\partial \boldsymbol{J}}{\partial x} = \frac{\boldsymbol{J}(y, x+1) - \boldsymbol{J}(y, x-1)}{2} \tag{6}$$

$$\frac{\partial \boldsymbol{J}}{\partial y} = \frac{\boldsymbol{J}(y+1, x) - \boldsymbol{J}(y-1, x)}{2}, \tag{7}$$

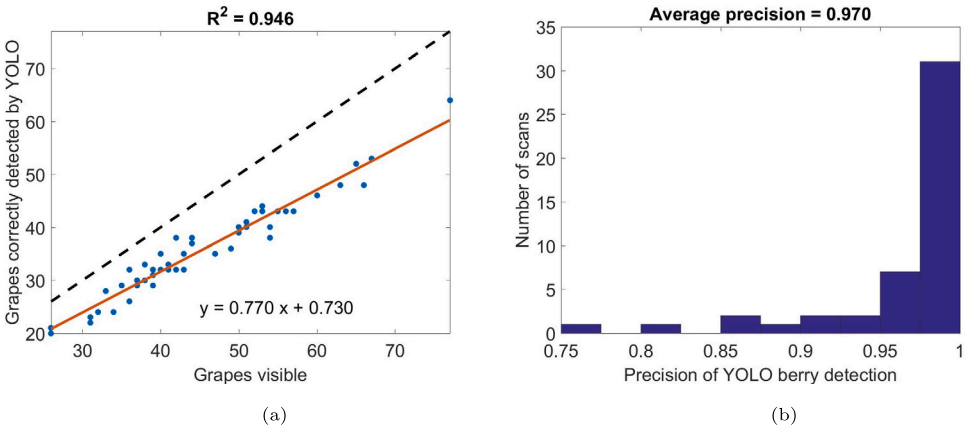


Fig. 17. Plot (a) shows the number of grapes correctly detected by YOLO versus the number manually identified in the photos for 50 grape bunches. The identity line is shown as a dotted line and the line of best fit is shown in orange. Plot (b) shows a histogram of the precision.

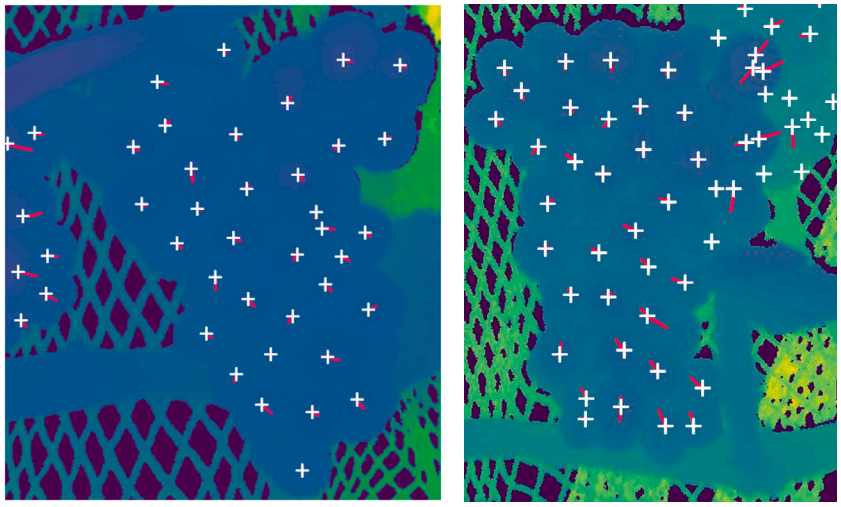


Fig. 18. Plots showing cropped versions of the depth maps corresponding to the grape bunches shown in Fig. 16(a) and (b). The red lines show the path taken using the gradient descent technique to move from the berry locations obtained by YOLO to the peaks in the depth map, which are shown as white crosses.

where $\boldsymbol{J}(y,x)$ is the value of the smoothed depth map at pixel location (y,x). The pixel location is updated using the gradient descent formula until the algorithm terminates:

$$\bar{\boldsymbol{p}}_{i+1} = \bar{\boldsymbol{p}}_i - \alpha \begin{bmatrix} \frac{\partial \boldsymbol{J}}{\partial y}(y,x) \\ \frac{\partial \boldsymbol{J}}{\partial x}(y,x) \end{bmatrix}. \tag{8}$$

This was repeated for 50 iterations in order to move the berry location to the top of the nearest peak, see Fig. 18. In all tested cases, this number of iterations was suitable to reach convergence. In cases where the traversal distance exceeded 15 pixels, the original coordinate was kept to prevent convergence on peaks too distant. This threshold was empirically determined to give the best results across the dataset.

In the majority of situations, this technique works well. However, in some edge cases, problems can show up. Some clear examples of these are demonstrated in Fig. 18(b). In some cases, the gradient descent process will cause multiple berry predictions to converge to the same peak within the depth map. This appears to happen most prominently on grapes that are occluded by a nearby grape causing the gradient to be stronger towards the peak of the occluding grape. In other cases, predictions of grapes behind the primary cluster (see the top right) ascend into the primary cluster. This is more common on grapes identified to the right and behind the primary cluster due to the parallax shift between the colour and depth sensors. This convergence behaviour also may help in some situations where the YOLO model predicts multiple grapes where only one exists. In this case, these predictions will converge to the same peak in the depth map. How these limitations can be solved or exploited will be the focus of future work.

Fig. 19 compares the scans captured using the depth peak detection technique described in Section 3 and YOLO. We can see in the RGB image that there are slight differences between where the two methods have identified the location of the berries to be. However, for the 3D plot, gradient descent has been used to move the YOLO berry locations to the depth peaks. This results in similar berry locations being obtained using both methods for the 3D point cloud.

5. Modelling of grape bunches

For grape yield estimation, it is desirable not only to count the number of grapes but also to be able to estimate the size of individual grapes so that grape volume can be estimated. This is particularly the case for grape varieties that typically have a wide range of berry diameters. The grapes used in this trial had a “hen and chicken” (Millerandage) effect where some grapes were smaller than others. Knowing the size distribution of grapes is a useful metric for effective vineyard management (Miras-Ávalos et al., 2019; Mirbod et al., 2016). Additionally, it is desirable to know the 3D structure of the grapes

(a) (b)

Fig. 19. Cropped versions of the corresponding photo (a) and colourised depth map (b) of a grape bunch. Overlaid are the detected berry locations obtained using depth peak detection (red crosses) and the YOLO model (blue crosses). For the depth map, the YOLO berry locations were moved to local peaks using gradient descent.

to allow better estimation of the total grape bunch volume and allow merging of scans of a grape bunch from multiple angles. Initial work was therefore conducted to estimate the size of the grapes detected and also construct a 3D model of the visible grapes.

5.1. Estimation of berry size from RGB images

The size of individual grape berries was detected from the RGB images using Hough transform circle detection. Initial trials using this technique over the entire RGB image gave poor results and were sensitive to hyper-parameter tuning; a limitation observed in existing works (Ang et al., 2018; Schmidtke, 2018; National Wine and Grape Industry Centre, 2019). Therefore, a two-step process was adopted that exploits the available understanding of where berries are located. For each berry location detected by the YOLO model, a 480 × 480 pixel sub-image was extracted from the original high resolution 4032 × 3024 colour image, see Fig. 20. This is similar to the technique that was used by Miao et al. (2021).

This sub-image was converted to grayscale and edge detection was performed using a Sobel kernel. This kernel was then used to find the gradient at each pixel in the X and Y axes and the magnitude of these two obtained. Circles were then detected using a Hough transform. In cases where multiple distinct circles were detected, the circle closest to the berry location detected by the YOLO model was used. The process was repeated for all detected berry locations, see Fig. 21.

The radius of the detected circles in pixels was able to be converted to a physical radius estimate using the knowledge of the distance of the camera from the grapes given by the depth camera and the camera calibration parameters. Similarly, the 3D location of each grape was also able to be estimated by projecting the YOLO detected locations onto the depth map using the process discussed in Section 4.4.

This information allowed a 3D model of the visible portion of the grape bunch to be generated. Spheres corresponding to the grapes were generated using their estimated size and 3D locations. This was done under the assumption that the peak found in the depth map corresponds to the closest point on the grape's surface to the camera. Furthermore, each grape can be modelled as a sphere where the point representing the peak is one of a pair of antipodal points, which, together with the camera origin and centre of the sphere, form a collinear set.

The 3D coordinate of the i_{th} sphere ($i = 1, \ldots, N$) is calculated using

$$\bar{\boldsymbol{C}}_i = \bar{\boldsymbol{X}}_i + \bar{\boldsymbol{d}}_i \, r_i \tag{9}$$

where $\bar{\boldsymbol{X}}_i$ is the 3D position of the detected peak, r_i is the radius of the sphere identified using circle detection, and $\bar{\boldsymbol{d}}_i$ is the normalised direction vector from the origin to the detected peak given by

$$\bar{\boldsymbol{d}}_i = \frac{\bar{\boldsymbol{X}}_i}{\|\bar{\boldsymbol{X}}_i\|}. \tag{10}$$

Refer to Fig. 22 for an example of a 3D model obtained using this technique overlain over the coloured 3D scan of the grapes generated from the depth and colour camera data.

This circle-fitting technique gives an approximation of the sizes of the grapes using the RGB images. However, errors can also be caused by the circles fitting to other features in the image such as the edge of another grape or colour changes on the surface of the grape. Also, many of the grapes appear as ellipses in the image rather than circles, which can lead to size estimation errors. Manual inspection of the fitted circles over the RGB images indicated that the circle fitting predominately resulted in some degree of underestimation compared to the true grape size. Refer to Fig. 20(c) and (e) for examples of this. Future work should explore using more advanced techniques such as the Holistically nested Edge Detection (HED) and ellipse fitting technique described in the work by Miao et al. (2021).

These issues raised the question of if it is possible to estimate the size of the grapes without measuring their size from the image. The following section investigates this in more detail.

5.2. Estimating berry size using depth

A technique was developed that estimates the size of the grapes and generates a 3D model using the identified locations of the grapes and the depth scan data rather than measuring the grape size from colour images. This approach works under the assumption that grape clusters are tightly packed and that they can be approximated as overlapping spheres. We also assume that the amount of overlap is proportional to their size.

Modelling of the 3D shape of the part of the grape bunch visible to the cameras was performed by creating a sphere for each grape using the method discussed in the previous section. To estimate the size of the

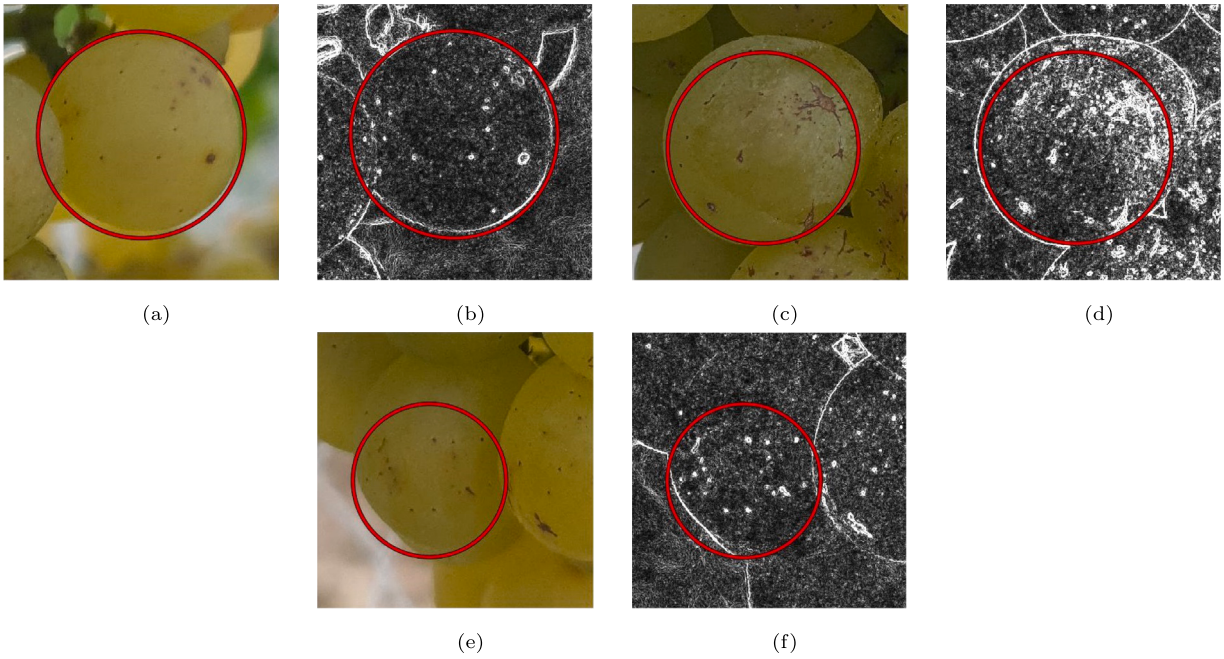


Fig. 20. Photos (a), (c) and (e) show example RGB images that have been automatically cropped to be centred on a berry location identified by YOLO. Plots (b), (d) and (f) show the corresponding Sobel magnitude versions that emphasise edges. Overlaid are the detected circles obtained using a Hough transform on the Sobel filtered images. It can be seen that underestimation in the sizing of the grapes occurred due to factors such as the elliptical shape of the grapes and occlusion by neighbouring grapes.

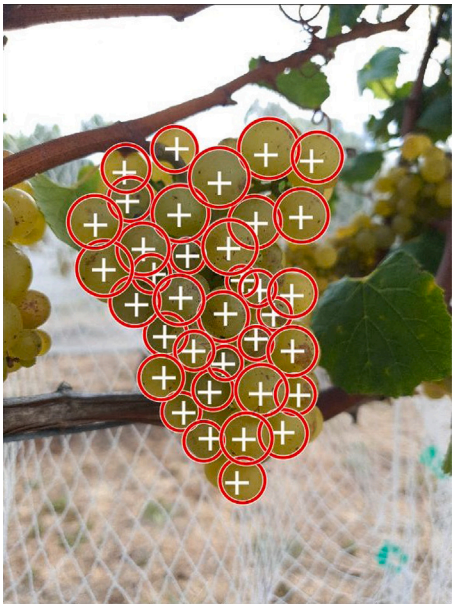


Fig. 21. Sizing of grapes using circle detection.

grapes, the size of each sphere was iteratively adjusted with the aim of optimising the overlap between neighbouring spheres and limiting the maximum size to be within a limit realistic for grapes.

We want to optimise the maximum overlap distance between the sphere being optimised and the neighbouring spheres while keeping the maximum radius $r_{\max}$ of each sphere under a limit. For this work, this maximum radius was chosen to be 10 mm to ensure enough range to capture the largest berries we could expect in a bunch of Chardonnay grapes.

The maximum overlap of the i_{th} sphere with its k_{th} neighbour is determined by

$$\gamma_i = \max\{(r_i - r_k) - f(i,k)\} : \text{for } k = 1, \ldots, N, \tag{11}$$

where $f(i,k)$ is a function that returns the distance between the centres of the i_{th} and k_{th} spheres, and N is the total number of spheres. For each iteration, the algorithm calculates a change in radius Δr_i, for the i_{th} sphere based on the maximum overlap with neighbouring spheres. If the maximum overlap, γ_i is less than 50% of the sphere's current radius and the sphere's radius is less than $r_{\max}$, then the radius is increased by a fixed amount of $\Delta r = 0.2$ mm. However, if the maximum overlap is larger than 50% of the sphere's current radius or the radius is over $r_{\max}$, then the radius is decreased by 10% of the current overlap. This can be expressed as

$$\Delta r_i = \begin{Bmatrix} 0.2, & \text{for } \gamma_i \leq 0.5 r_i \\ & \text{and } r_i \leq r_{\max} \\ -0.1\gamma_i, & \text{for } \gamma_i > 0.5 r_i \\ & \text{or } r_i > r_{\max} \end{Bmatrix}. \tag{12}$$

These thresholds and step sizes were chosen from empirical testing to help the simulation converge swiftly while also being stable. The 50% overlap threshold attempts to capture the squishing behaviour observed in the tightly grouped chardonnay bunches at the particular stage of development that images were captured. Different thresholds can be used to achieve different results and more work will need to be done to explore its impact on the simulations accuracy for different cultivars or stages of growth.

The simulation is run for a fixed number N_{iter} of iterations to ensure convergence. Changing the radius causes the position of the sphere to change, see Eq. (9). Therefore, two passes over the spheres are conducted for each iteration. The first calculates Δr_i for each sphere, and the second applies this change and updates the centre of the sphere per Eq. (9). The change in radius is applied to calculate the updated

(a) (b)

Fig. 22. Example of the 3D modelling of the grape bunch overlaid onto the colourised depth map shown in Fig. 5. Plot (a) and (b) respectively show opaque and semi-transparent versions of the modelled spheres with diameters obtained using the circle fitting technique.

(a) (b)

Fig. 23. Plots (a) and (b) respectively show opaque and semi-transparent versions of a 3D modelling of grapes generated by growing spheres at the location of grapes obtained from the peaks in the depth map. These are overlain over a 3D depth map scan of the grape bunch.

centre as follows

$$\bar{\boldsymbol{C}}_i[j+1] = \bar{\boldsymbol{d}}_i(\Delta r_i + r_i[j]) + \bar{\boldsymbol{X}}_i \tag{13}$$

where j ($j = 1, \ldots, N_{\text{iter}} - 1$) is the current iteration.

In this way, the position of the sphere is constrained by the relationship between its size and the amount of overlap with neighbouring spheres. The sphere will grow or shrink as necessary to avoid excessive overlap with neighbouring spheres, but it cannot exceed the maximum radius specified. The size of the spheres can then be used to estimate the size of the grape berries.

Refer to Fig. 23 for an example of a model of a grape bunch using this technique. This method shows similar results compared to those obtained using the circle fitting technique shown in Fig. 22. However, the resulting 3D scan does appear to be more accurate than the circle size approach when compared to the underlying colour image.

5.3. Comparison of grape sizes obtained using the RGB circle detection and depth techniques

Fig. 24 presents a comparison of the grape sizes obtained using Hough transform circle fitting technique with those obtained using the depth technique for the grape bunch presented in Figs. 22 and 23. It can be seen that the radii obtained using the RGB method was systematically lower than that obtained using the depth method. This is in line with expectations since the circle fitting tended to fit to one end of the ellipsoid shape of the grapes causing an systematic underestimation of the grape sizes, as illustrated in Fig. 20 (c - f). Additionally, the distribution of these underestimations changes throughout the grape bunch depending on the shape or occlusion of individual berries. This explains some of the outliers present in the data and by extension the low correlation. In two cases, the simulated grape sizes have reached

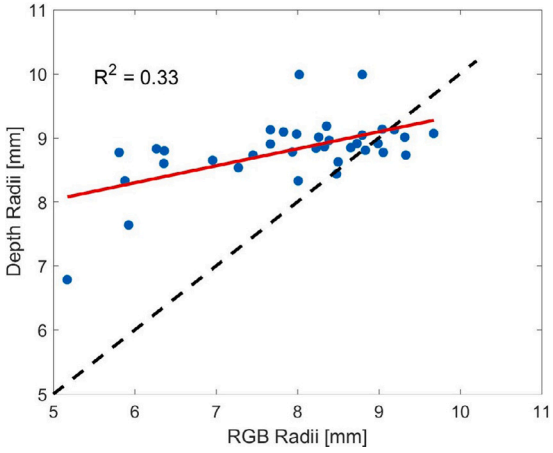


Fig. 24. Comparison of the sizes of grapes obtained from circle detection in RGB images compared to those obtained using the peaks in the depth maps. The identity line is shown as a dotted line and the line of best fit is shown in red.

the maximum allowed by the simulation, 10 mm. This indicates that those grapes were floating and did not have nearby grapes to constrain their size. In such a case, it may be more correct to use the RGB size estimations or a combination of the two. More work is needed to evaluate the performance of both of these methods against ground truth data and explore opportunities to combine both techniques for a robust solution.

6. Conclusion

An Android app was developed for a Samsung Note 10+ smartphone to capture RGB images and depth and confidence maps simultaneously from its colour and ToF depth cameras. Stereo calibration of these two cameras was then performed using a checkerboard pattern. This allowed projection from the depth map to the corresponding RGB image along with mapping from the RGB image back to the depth map. Coloured 3D point clouds were able to be generated from the RGB and depth data. The colour in these point clouds was not utilised in this work but there is the potential for this to be used for improved results in future work.

The smartphone was used in field trials to perform scans of Chardonnay grapes in situ. Additionally, measurements were taken in the lab with samples of grape bunches from the field. A turntable was used to capture scans of each of these grape bunches at a range of angles.

A technique was developed to automatically identify grape berries in the depth maps using peak detection. This exploited the distortions in the ToF depth camera images due to diffused scattering within the berries. A persistence algorithm was used to detect peaks in the depth map. A signed distance field filter was used to remove peaks at the edges of objects and those corresponding to netting or leaves. This technique successfully detected most of the visible grapes, though some were missing particularly at the edges. An R^2 value of 0.68 was obtained for a linear fit between the number of grapes visible in the RGB photos and those correctly detected using the depth peak fitting technique. An average precision of 0.893 was achieved.

Automatically identifying grape berries from peaks in the depth maps shows promise and further improvements could be made in future work. For example, the autoencoder that was developed for the YOLO training could be used to help improve the rejection of peaks that do not correspond to grapes. Including registered colour information in addition to depth could also help with improving the accuracy of this peak detection technique. Convolutional Neural Networks (CNN) may also provide an effective means of classifying which peaks are grapes.

A YOLOv7 model was trained to detect grape berries in RGB images captured by the smartphone. The dataset was constructed from lab-captured RGB and depth images. A technique was devised to facilitate unsupervised training by leveraging the peaks detected in the corresponding depth maps. An autoencoder was implemented to eliminate non-berry peaks, including those associated with visible rachis or peduncles. To enhance the dataset's adaptability to outdoor environments, training images were augmented with diverse foliage backgrounds through depth-based masking of grape clusters.

An R^2 value of 0.946 was achieved between a fitting of the number of berries correctly detected by YOLO and those manually counted in the RGB images and an average precision of 0.970 was achieved. The fit shows an underestimation in the number of berries detected by YOLO compared to those counted manually in the images. However, the strong relationship suggests that linear compensation would be an effective method of correction. The grapes that were missed by YOLO were mainly those around the edges of the grape bunch. This may be due to the fact that only a fraction of many of the berries on the edges of the bunch are clearly visible due to occlusions by other berries. However, it could also partly be related to the way the YOLO dataset was constructed and the low sensitivity of the peak detection process to occluded grapes on the edge of clusters. This may have meant that the YOLO model did not have sufficient training for grapes at the edge of the bunch. In future work, the manual selection of bounding boxes around berries missed by the peak detection could help improve the performance of the YOLO model in these cases. Additional training to remove YOLO detection of withered-up grapes could also be performed.

The YOLO model also struggled with grape bunches in the background where the RGB image was out of focus. In future work, this could be addressed using depth information by identifying the grape bunch of interest and filtering YOLO-detected points that would be out of focus in the RGB image. The grape bunch of interest could be identified based on its 3D position in the scan. Alternatively, one could manually click on the grape bunch in an image when capturing the scan using the app. One could also remove some of the false positives in the YOLO results using spatial filtering such as calculating the mean distance from the location of each detected berry to that of its K-nearest neighbours. More support could also be given to YOLO by producing augmented training data where grape bunches are blurred. Additionally, YOLO occasionally produced false positive results by incorrectly identifying items in the background, such as netting or leaves, as grape berries. More varied background augmentations will help add robustness to these cases.

YOLO models are traditionally trained manually by labelling images by hand. However, this can be very time-consuming. For individual grape berry detection, this training would need to be repeated for different grape varieties. This is perhaps why only two works were found where YOLO has been used to detect individual grape berries (Miao et al., 2021; Roboflow Universe, 2021). The automated approach introduced in this research, designed for unsupervised training of a YOLO model to detect grape berries, has the capacity to accelerate the training of YOLO models for a variety of grape types. The results presented here showed good accuracy. However, more work is needed to compare the accuracy obtained using this technique with that obtained using the traditional manual labelling method. Additionally, future work should investigate if adding manual labelling, particularly around the edges of grape bunches, could help improve the accuracy of the automated technique described in this work.

The berry locations detected by YOLO were able to be projected onto the depth map using the depth and RGB camera stereo calibration parameters. However, these predicted berry locations were generally slightly misaligned relative to the peaks in the depth map. A gradient descent technique was therefore developed that moved the projected YOLO berry locations to the top of nearby peaks. A potential issue with this approach is that it can result in two or more points detected by YOLO converging to the same 3D peak location. This can be seen demonstrated in Fig. 18(b) where berries detected in the background have ended up on a peak in the main bunch. Future work could

investigate alternative methods of combining the presented YOLO and peak-based detection methods to provide a more robust approach to berry detection and filtering false positives.

Estimation of the size of grape berries in the grape bunches was performed with a two-step process. Firstly, circles were detected in the RGB images at the grape locations obtained using the YOLO model. Next, the physical size of each grape could be estimated from its size in the RGB image, the distance of the grape from the camera calculated from the depth map, and the RGB camera's intrinsic parameters. This eliminates the need for placing a reference object next to the grapes as has been used in previous works.

The generated size estimates were utilised to construct a 3D model of the grape bunch. By projecting the YOLO-detected berry locations from the RGB image onto the corresponding depth map, appropriately sized spheres were positioned at their respective 3D coordinates. Although this technique showed potential, it often underestimated berry sizes in our observations. In future work, one could investigate fitting ellipsoids to the data rather than spheres, as demonstrated by Miao et al. (2021). However, employing ellipses introduces additional hyperparameters that significantly increase the transform space and may be influenced by image noise.

A sphere-growing optimisation technique was therefore developed to estimate the size of the berries in a grape bunch without having to measure their sizes in the RGB images. This approach works under the assumption that grape clusters are tightly packed and that this can be approximated as overlapping spheres. Spheres were placed as before with the sizes iteratively adjusting to optimise the overlap among the entire cluster. This approach is sensitive to cases where grapes do not in reality touch other grapes or if some grapes are missed by the YOLO model. Future work could look at combining size estimates from RGB circle detection with this simulated approach as a method for constraining size expectations for each berry.

These grape size estimation results obtained using both the RGB and depths techniques showed promise. However, these results are qualitative. More work is needed in the future to compare these results with ground truth measurements of the physical sizes of the grape berries using callipers or scanning techniques such as laser scanners, photogrammetry etc.

Further work is also needed to build an understanding of the grapes within the cluster not visible to the camera. Past research has approximated these with a simple scaling factor. Our 3D models may also be accurate enough to extend to a complete phenotype estimate of the hidden structure, a process typically used with high-resolution 3D scans (Schöler and Steinhage, 2015). Additionally, scans from multiple angles may be able to be combined to increase the proportion of the grape bunch able to be included in the modelling.

This work was performed using a Samsung Note 10+ smartphone. However, the techniques should extend to any system that has a combined RGB camera and ToF or LiDAR depth cameras. This includes a range of modern smartphones from Apple and Android and low-cost depth camera systems that are currently commercially available such as the Microsoft Azure Kinect DK. Additionally, the YOLO model is suitable for berry detection with standalone RGB cameras without a smartphone for depth sensors.

The data used in this work was for Chardonnay grapes, which are green in colour. Initial lab-based trials were also performed on red table grapes, though the results are not presented here. Similar peaks were observed in the ToF depth maps captured of these red grapes compared to those presented in this work. This leads to some confidence that the technique would be extendable to other grape varieties. However, additional experiments with different grape varieties would be beneficial.

The field trials were performed on the grapes approximately two weeks before harvest. It is beneficial for growers to perform yield estimation measurements at this stage of growth so that they can estimate the volume of grapes that will be harvested, etc. However, it is also desirable to be able to perform yield estimations at different stages of grape maturity. It is likely that the diffused scattering within the grapes that is causing the peak distortion may change with grape maturity. Therefore, it would be desirable in future work to perform further trials of grapes at a range of maturity levels.

CRediT authorship contribution statement

Baden Parr: Conceptualization, Methodology, Software, Formal analysis, Investigation, Data curation, Writing – original draft, Writing – review & editing. **Mathew Legg:** Conceptualization, Methodology, Software, Formal analysis, Writing – original draft, Writing – review & editing, Funding acquisition, Project administration, Supervision. **Fakhrul Alam:** Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Baden Paerr reports financial support was provided by Bragato Research Institute (Rod Bonfiglioli PhD Scholarship).

Data availability

Data will be made available on request.

Acknowledgement

The researchers would like to acknowledge Bragato Research Institute (a subsidiary of New Zealand Winegrowers) as this research was supported in part by the Rod Bonfiglioli Scholarship.

References

Ang, L.-M., Seng, K., Oczkowski, A., Deloire, A., Schmidtke, L., 2018. Development of a smartphone app for berry quality assessment. In: Vigne Et Vin Publications. pp. 79–85, 7th Symposium of the OENOVITI International Network, 25 April.

Aquino, A., Barrio, I., Diago, M.-P., Millan, B., Tardaguila, J., 2018. vitisBerry: An Android-smartphone application to early evaluate the number of grapevine berries by means of image analysis. Comput. Electron. Agric. 148, 19–28. http://dx.doi.org/10.1016/j.compag.2018.02.021.

Barriguinha, A., de Castro Neto, M., Gil, A., 2021. Vineyard yield estimation, prediction, and forecasting: A systematic literature review. Agronomy 11 (9), 1789. http://dx.doi.org/10.3390/agronomy11091789.

Ciarfuglia, T.A., Motoi, I.M., Saraceni, L., Fawakherji, M., Sanfeliu, A., Nardi, D., 2023. Weakly and semi-supervised detection, segmentation and tracking of table grapes with limited and noisy data. Comput. Electron. Agric. 205, 107624. http://dx.doi.org/10.1016/j.compag.2023.107624.

Coviello, L., Cristoforetti, M., Jurman, G., Furlanello, C., 2020. GBCNet: In-field grape berries counting for yield estimation by dilated CNNs. Appl. Sci. 10 (14), 4870. http://dx.doi.org/10.3390/app10144870.

Font, D., Pallejà, T., Tresanchez, M., Teixidó, M., Martinez, D., Moreno, J., Palacín, J., 2014. Counting red grapes in vineyards by detecting specular spherical reflection peaks in RGB images obtained at night with artificial illumination. Comput. Electron. Agric. 108, 105–111. http://dx.doi.org/10.1016/j.compag.2014.07.006.

Grossêtete, M., Berthoumieu, Y., Da Costa, J.-P., Germain, C., Lavialle, O., Grenier, G., 2011. A new approach on early estimation of vineyard yield: Site specific counting of berries by using a smartphone. In: European Conference on Precision Agriculture. pp. 8–pages.

Grossetete, M., Berthoumieu, Y., Da Costa, J.-P., Germain, C., Lavialle, O., Grenier, G., et al., 2012. Early estimation of vineyard yield: Site specific counting of berries by using a smartphone. In: International Conference of Agricultural Engineering—CIGR-AgEng.

Hacking, C.J., 2020. 2-D and 3-D proximal remote sensing for yield estimation in a Shiraz vineyard (Ph.D. thesis). Stellenbosch: Stellenbosch University.

Hacking, C., Poona, N., Poblete-Echeverría, C., 2020. Vineyard yield estimation using 2-D proximal sensing: A multitemporal approach. OENO One 54 (4), 793–812. http://dx.doi.org/10.20870/oeno-one.2020.54.4.3361.

Herrero-Huerta, M., González-Aguilera, D., Rodriguez-Gonzalvez, P., Hernández-López, D., 2015. Vineyard yield estimation by automatic 3D bunch modelling in field conditions. Comput. Electron. Agric. 110, 17–26. http://dx.doi.org/10.1016/j.compag.2014.10.003.

Huber, S., 2021. Persistent homology in data science. In: Haber, P., Lampoltshammer, T., Mayr, M., Plankensteiner, K. (Eds.), Data Science – Analytics and Applications. Springer Fachmedien Wiesbaden, Wiesbaden, pp. 81–88.

Huber, S., 2022. Topological peak detection in two-dimensional data. Available online: https://www.sthu.org/code/codesnippets/imagepers.html, (Last visited on 3 September 2023).

Ivorra, E., Sánchez, A., Camarasa, J., Diago, M.P., Tardáguila, J., 2015. Assessment of grape cluster yield components based on 3D descriptors using stereo vision. Food control 50, 273–282. http://dx.doi.org/10.1016/j.foodcont.2014.09.004.

Kurtser, P., Ringdahl, O., Rotstein, N., Andreasson, H., 2020a. PointNet and geometric reasoning for detection of grape vines from single frame RGB-D data in outdoor conditions. In: 3rd Northern Lights Deep Learning Workshop, Tromsö, Norway, Vol. 1. NLDL, pp. 1–6. http://dx.doi.org/10.7557/18.5155.

Kurtser, P., Ringdahl, O., Rotstein, N., Berenstein, R., Edan, Y., 2020b. In-field grape cluster size assessment for vine yield estimation using a mobile robot and a consumer level RGB-D camera. IEEE Robot. Autom. Lett. 5 (2), 2031–2038. http://dx.doi.org/10.1109/LRA.2020.2970654.

Laurent, C., Oger, B., Taylor, J.A., Scholasch, T., Metay, A., Tisseyre, B., 2021. A review of the issues, methods and perspectives for yield estimation, prediction and forecasting in viticulture. Eur. J. Agron. 130, 126339. http://dx.doi.org/10.1016/j.eja.2021.126339.

Li, H., Li, C., Li, G., Chen, L., 2021. A real-time table grape detection method based on improved YOLOv4-tiny network in complex background. Biosyst. Eng. 212, 347–359. http://dx.doi.org/10.1016/j.biosystemseng.2021.11.011.

Liu, B., Luo, L., Wang, J., Lu, Q., Wei, H., Zhang, Y., Zhu, W., 2023. An improved lightweight network based on deep learning for grape recognition in unstructured environments. Inform. Process. Agric. http://dx.doi.org/10.1016/j.inpa.2023.02.003.

Liu, S., Zeng, X., Whitty, M., 2020a. 3DBunch: A novel iOS-smartphone application to evaluate the number of grape berries per bunch using image analysis techniques. IEEE Access 8, 114663–114674. http://dx.doi.org/10.1109/ACCESS.2020.3003415.

Liu, S., Zeng, X., Whitty, M., 2020b. A vision-based robust grape berry counting algorithm for fast calibration-free bunch weight estimation in the field. Comput. Electron. Agric. 173, 105360. http://dx.doi.org/10.1016/j.compag.2020.105360.

Mack, J., Schindler, F., Rist, F., Herzog, K., Töpfer, R., Steinhage, V., 2018. Semantic labeling and reconstruction of grape bunches from 3D range data using a new RGB-D feature descriptor. Comput. Electron. Agric. 155, 96–102. http://dx.doi.org/10.1016/j.compag.2018.10.011.

Marinello, F., Pezzuolo, A., Cillis, D., Sartori, L., et al., 2016. Kinect 3D reconstruction for quantification of grape bunches volume and mass. Eng. Rural Dev. 15, 876–881.

Miao, Y., Huang, L., Zhang, S., 2021. A two-step phenotypic parameter measurement strategy for overlapped grapes under different light conditions. Sensors 21 (13), 4532. http://dx.doi.org/10.3390/s21134532.

Miras-Ávalos, J.M., Buesa, I., Yeves, A., Pérez, D., Risco, D., Castel, J.R., Intrigliolo, D.S., et al., 2019. Unravelling the effects of berry size on 'Tempranillo' grapes under different field practices. Ciência e Técnica Vitivinícola 34 (1), 1–14. http://dx.doi.org/10.1051/ctv/20193401001.

Mirbod, O., Yoder, L., Nuske, S., 2016. Automated measurement of berry size in images. IFAC-PapersOnLine 49 (16), 79–84. http://dx.doi.org/10.1016/j.ifacol.2016.10.015.

Moreno, H., Andújar, D., 2023. Proximal sensing for geometric characterization of vines: A review of the latest advances. Comput. Electron. Agric. 210, 107901. http://dx.doi.org/10.1016/j.compag.2023.107901.

National Wine and Grape Industry Centre, 2019. WineOz SmartGrape. https://play.google.com/store/apps/details?id=com.nwgic.grapeyield. (Last visited on 5 Sept 2022).

Parr, B., Legg, M., Alam, F., 2022. Analysis of depth cameras for proximal sensing of grapes. Sensors 22 (11), http://dx.doi.org/10.3390/s22114179.

Roboflow Universe, 2021. Berry_yoloV5 Dataset. Roboflow Universe, Roboflow, https://universe.roboflow.com/new-workspace-hzmvk/berry_yolov5-slwnw. (Last visited on 20 March 2023).

Rose, J.C., Kicherer, A., Wieland, M., Klingbeil, L., Töpfer, R., Kuhlmann, H., 2016. Towards automated large-scale 3D phenotyping of vineyards under field conditions. Sensors 16 (12), http://dx.doi.org/10.3390/s16122136.

Santos, T.T., de Souza, L.L., dos Santos, A.A., Avila, S., 2020. Grape detection, segmentation, and tracking using deep neural networks and three-dimensional association. Comput. Electron. Agric. 170, 105247. http://dx.doi.org/10.1016/j.compag.2020.105247.

Schmidtke, L., 2018. Developing a phone-based imaging tool to inform on fruit volume and potential optimal harvest time. Tech. Rep. Project No. CSU 1501, Charles Sturt University, National Wine and Grape Industry Centre, Charles Sturt University, Wagga Wagga, New South Wales, Australia.

Schöler, F., Steinhage, V., 2015. Automated 3D reconstruction of grape cluster architecture from sensor data for efficient phenotyping. Comput. Electron. Agric. 114, 163–177. http://dx.doi.org/10.1016/j.compag.2015.04.001.

Shen, L., Su, J., He, R., Song, L., Huang, R., Fang, Y., Song, Y., Su, B., 2023. Real-time tracking and counting of grape clusters in the field based on channel pruning with YOLOv5s. Comput. Electron. Agric. 206, 107662. http://dx.doi.org/10.1016/j.compag.2023.107662.

Tardaguila, J., Stoll, M., Gutiérrez, S., Proffitt, T., Diago, M.P., 2021. Smart applications and digital technologies in viticulture: A review. Smart Agric. Technol. 1, 100005. http://dx.doi.org/10.1016/j.atech.2021.100005.

Wang, C.-Y., 2022. Official YOLOv7. GitHub repository, GitHub, https://github.com/WongKinYiu/yolov7. (Last visited on 18 May 2023).

Wang, C.-Y., Bochkovskiy, A., Liao, H.-Y.M., 2022. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv preprint arXiv:2207.02696.

Xin, B., Liu, S., Whitty, M., 2020. Three-dimensional reconstruction of *Vitis vinifera* (L.) cvs Pinot Noir and Merlot grape bunch frameworks using a restricted reconstruction grammar based on the stochastic L-system. Aust. J. Grape Wine Res. 26 (3), 207–219. http://dx.doi.org/10.1111/ajgw.12444.

Xin, B., Whitty, M., 2022. A 3D grape bunch reconstruction pipeline based on constraint-based optimisation and restricted reconstruction grammar. Comput. Electron. Agric. 196, 106840. http://dx.doi.org/10.1016/j.compag.2022.106840.

Zhao, R., Zhu, Y., Li, Y., 2022. An end-to-end lightweight model for grape and picking point simultaneous detection. Biosyst. Eng. 223, 174–188. http://dx.doi.org/10.1016/j.biosystemseng.2022.08.013.

Zhu, Z., Wang, X., Bai, S., Yao, C., Bai, X., 2016. Deep learning representation using autoencoder for 3D shape retrieval. Neurocomputing 204, 41–50. http://dx.doi.org/10.1016/j.neucom.2015.08.127.