Computers and Electronics in Agriculture 137 (2017) 88–101

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

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

Original papers

A computer vision system for early stage grape yield estimation based on shoot detection

Scarlett Liu [a,*], Steve Cossell [a], Julie Tang [a], Gregory Dunn [b], Mark Whitty [a]

[a] UNSW Sydney, Australia
[b] Department of Primary Industries, Agriculture NSW, Australia

ARTICLE INFO

Article history:
Received 5 December 2016
Received in revised form 12 March 2017
Accepted 21 March 2017
Available online 6 April 2017

Keywords:
Grape yield estimation
Shoot detection
Feature selection
Data classification
Vineyard mapping

ABSTRACT

Counting grapevine shoots early in the growing season is critical for adjusting management practices but is challenging to automate due to a range of environmental factors.

This paper proposes a completely automatic system for grapevine yield estimation, comprised of robust shoot detection and yield estimation based on shoot counts produced from videos. Experiments were conducted on four vine blocks across two cultivars and trellis systems over two seasons. A novel shoot detection framework is presented, including image processing, feature extraction, unsupervised feature selection and unsupervised learning as a final classification step. Then a procedure for converting shoot counts from videos to yield estimates is introduced.

The shoot detection framework accuracy was calculated to be 86.83% with an F1-score of 0.90 across the four experimental blocks. This was shown to be robust in a range of lighting conditions in a commercial vineyard. The absolute predicted yield estimation error of the system when applied to four blocks over two consecutive years ranged from 1.18% to 36.02% when the videos were filmed around E-L stage 9.

The developed system has an advantage over traditional PCD mapping techniques in that yield variation maps can be obtained earlier in the season, thereby allowing farmers to adjust their management practices for improved outputs. The unsupervised feature selection algorithm combined with unsupervised learning removed the requirement for any prior training or labeling, greatly enhancing the applicability of the overall framework and allows full automation of shoot mapping on a large scale in vineyards.

1. Introduction

Accurate yield estimation for wine grapes is essential for improving winery efficiency and wine marketing strategies. In addition to this, an accurate early season forecast allows growers to sensibly regulate vineyard yields to meet grape quality targets. Existing traditional yield estimation approaches can be used as early as bud burst, and generally require a bud, bunch, or berry counts (Martin and Dunn, 2003), requiring expensive and time consuming in-field measurements. These measures are often biased prone to error. An automated system has the potential to eliminate human error and reduce labor costs. In recent years, the application of image processing techniques to obtain data in vineyards and facilitate better forecasting has become a major research focus. These studies fit into two broad groups: 1. the detection of a single object of interest from a small group of images, or 2. extracting pertinent information from images on a large scale.

For the group 1, in-field detection techniques have been used to characterize disease (Meunkaewjinda et al., 2008), and vine components (Correa et al., 2011; Fernández et al., 2013; Klodt et al., 2015) from individual or stereo images. However, the techniques requires substantial processing time and images must have consistent illumination and composition. Furthermore, they require custom or stationary rigs to obtain stable imagery (Correa et al., 2011; Diago et al., 2012), which on a vineyard scale becomes a slow or expensive process. Requirements for a particular vine component to be present within an image for correct segmentation (Correa et al., 2011) requires further automation to increase feasibility, and classifications on a pixel level are also slow and not recommended for large-scale image processing. Thus, a robust in-field

* Corresponding author.
E-mail addresses: scarlett.liu@unsw.edu.au (S. Liu), scos506@gmail.com (S. Cossell), julie.tang@unsw.edu.au (J. Tang), gregory.dunn@dpi.nsw.gov.au (G. Dunn), m.whitty@unsw.edu.au (M. Whitty).
URL: http://www.robotics.unsw.edu.au/srv/ (S. Liu).

http://dx.doi.org/10.1016/j.compag.2017.03.013
0168-1699/

method is required. Liu and Whitty (2015) proposed an approach to detect bunches in natural environmental conditions by a Support Vector Machine. The work done by Liu et al. (2013) and Font et al. (2015) investigated relationships between grape bunch weight and single image descriptors generated by different image processing methods. Dorj et al. (2013) demonstrated a color segmentation approach to detect tangerine flowers. However, none of these studies explored relating detection results to final yield. The cost and robustness of these detection methods have not been tested for larger scale implementation.

Most recently, Payne et al. (2013), Wang et al. (2013) and Nuske et al. (2014) investigated yield forecasting through image processing. Their work is not limited to object detection but rather converts the detection results into a final yield after being combined with manual field sampling. This has provided a solution to estimate the final yield by image processing but only after the fruit

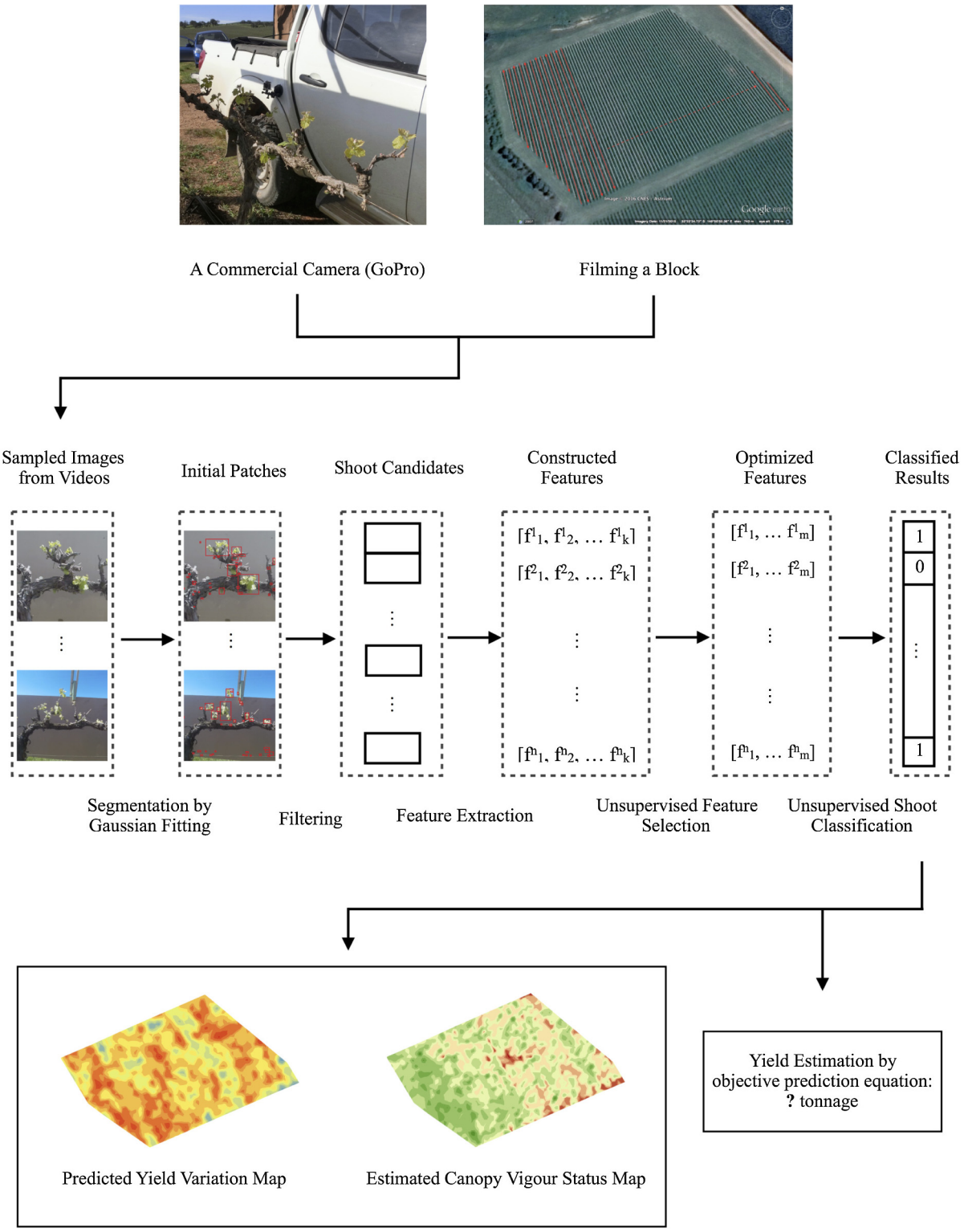


Fig. 1. Flowchart of the system for yield estimation presented in this paper based on computer vision, labeled with corresponding sections of this paper.

has set. Also, Payne et al. (2013) and Wang et al. (2013)'s methods are sensitive to the light condition and cultivar since their detection methods rely on color segmentation and shape/texture filters. Nuske et al. (2014)'s work successfully utilizes supervised learning for assisting further berry detection. However it is not practical for processing different cultivars due to the requirement of manually labeled training data for variations in grape variety and canopy training system. Their vision system is comprised of high-resolution industrial cameras and professional simultaneously triggered flash and predominantly deployed at night to ensure the homogeneous illumination. However, the specialized nature of the equipment limits its widespread use.

Therefore, a low cost and robust vision system for yield forecasting at an earlier phenological stage than veraison, and indeed after veraison, is desirable. Here we investigated yield estimation at an early phenological stage prior to fruit set and demonstrated the ability to forecast yield using a low cost vision system along with a novel shoot detection framework. The novelty of this work lies in:

1. the ability to forecast yield and generate a map of potential yield variation at the early phenological stage of shoots,
2. no manual labeling required to build a classifier,
3. ability to deal with certain range of illumination changes and different scenarios with various noise based on unsupervised feature selection,
4. the ability to use low-cost off-the-shelf image collection equipment (e.g. a camera that can record video at a standard frequency),
5. identification of the optimum phenological stage for imaging shoots.

The proposed system for yield estimation at the shoot stage is shown in Fig. 1; which also presents an overview to this paper. The rest of this paper is organized as follows: Section 2 presents the data scope and experimental setup. Section 3 presents the framework for detecting shoots from video imagery and quantifies these results. Section 4 presents the yield forecasting algorithm based on the shoot classification results. Section 5 presents the corresponding yield variation map and discusses the correlation between it and the actual yield map. The conclusion and future work is presented in Section 6.

2. Data collection and data scope

All experiments were implemented on four blocks situated in Clare Valley, SA and Orange, NSW, Australia, consisting of two cultivars and two canopy training systems. Chardonnay and Shiraz were chosen since they are common white and red grape varieties in Australia. Data was collected from two different locations to exclude the effects of different terroirs. General information about these four blocks is listed in Table 1. The abbreviations listed below are used throughout this paper.

1. FT - Field Trip. FT1.2 means the second field trip in Season 1 while FT2.5 means the fifth field trip in Season 2.
2. TL - Temporal Locations: 20*2 panel (3 post) segments in each block. 20 two-panels segments (chosen by best-candidate sampling (Mitchell, 1991) counting walls as points with half weight and reassigning 15% of locations to be adjacent to existing points) were marked in each block. The locations of 20 TLs were not changed over the three year duration of the project. They are presented as pink short strips in Fig. 2.
3. SS - Sampling Segments: 60*0.6 m segments, destructively sampled on each field trip (approximately 6 times per season). Their locations were randomly generated by best candidate sampling (2D) and vary over three years. Yellow points in Fig. 2 are examples of a set of SSs at one time.
4. VID - Video of Rows: Video of rows at each field trip. A GoPro camera was installed on a vehicle that drives through a block and films the canopy following the method presented by Cossell et al. (2016). The field of view, resolution, and number of frames/second were set to *Medium, 1080p, 30fps* respectively.
5. V2015: associated with the 2015 vintage, ranging from approximately September to March.

Table 1
The information about experimental blocks.

Block name	40A	47A	B4	B12
Variety	Chardonnay	Shiraz	Shiraz	Chardonnay
Location	Clare, SA	Clare, SA	Orange, NSW	Orange, NSW
Area [Ha]	6.2	10.6	6.5	3.7
Altitude [m]	391–414	371–403	745–760	730–748
Pruning Type	Spur	Spur	Spur	Spur
Trellis System	Aussie sprawl	Aussie sprawl	VSP	VSP
Rows	84	134	78	83
V2015 Yield [t]	47.9	37.0	30.7	24.9
V2016 Yield [t]	67.1	59.32	69.005	61.742

(a) 40A, Clare Valley (b) 47A, Clare Valley (c) B4, Orange (d) B12, Orange

Fig. 2. Locations of TLs (pink line) and SSs (yellow points) on maps of the four experimental blocks. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Note that for yield estimation there are three essential aspects that need to be addressed: the accuracies of visible shoot detection, shoot counting, and yield estimation. Different datasets were used to verify the experimental results corresponding to these three aspects as shown in Table 2.

3. Shoot detection

The common challenges for shoot detection in vineyards are varying lighting conditions, undesired objects in the field of view (e.g. posts, cordon, grass, animals, wire, reflections), change of shoot position in the field of view, shadows and barren cordon. Fig. 3 illustrates the range of real objects encountered in our dataset. Instead of pre-processing images to obtain shoot candidates, some approaches divide images into sub-windows and process them all as object candidates (Chamelat et al., 2006); however, window search is exhaustive and time consuming (Liu and Whitty, 2015).

The common method for segmenting targets by image processing is to form a binary image using a threshold value. In this case, the threshold value cannot be fixed since the illumination condition changes as the vehicle moves. Using a dynamic threshold method such as that proposed by Otsu (1975) improves the results using a histogram of the image but the correct threshold cannot be guaranteed for shoot segmentation when the image contains complicated objects. The proposed shoot detection method segments potential shoot patches by Gaussian fitting based on color histograms for automatically locating the threshold value accurately. It then combines different scalar features into a feature vector to use as a descriptor of potential shoots.

Given that the data collection, here in the form of videos of shoots, is conducted during the daytime, illumination conditions and scenarios can vary significantly within a block. Supervised learning approaches are not preferable for this real world problem since labels are required each time the scenario or illumination condition changes and the time and cost of labeling is not practical. Considering the size of vineyards, we chose to develop a completely automatic way to address this problem. Unsupervised feature selection based on three correlation filters (Kendall et al., 1946; Higgins, 2003; He et al., 2005) are compared to enhance clustering performance and increase computational efficiency. Two unsupervised clustering approaches, K-means (MacQueen, 1967) and Agglomerative Hierarchical clustering (Hastie et al., 2005), are compared for better yield estimation.

3.1. Segmentation of initial patches

Each image is firstly transferred into the L*a*b color space because channels **a** and **b** provide more distinctive information than the RGB color space for segmenting shoots. It is then segmented based on the distribution of pixels in channels **a** and **b**. A Gaussian model is utilized to automatically allocate the threshold value of the histogram in channel **a** and **b** to segment the shoot candidates. Then some simple morphological operations are applied after this process to reduce the noise. Fig. 4 demonstrates some of the results of segmentation in this section.

3.2. Constructing features from shoot candidates

To represent each patch in feature space for clustering, a feature vector combining patch and texture properties (Gonzalez et al., 2004) is constructed for each candidate. In addition, closeness, compactness and extent (Liu and Whitty, 2015) are added into the feature vector too.

3.3. Unsupervised feature selection

N initial segmented patches are extracted from sampled fromes of all the videos collected across a block. Each patch that describes a shoot represents a point in attribute space. In this paper, unsupervised feature selection is conducted to narrow down the attribute space to recognize shoots without performance deterioration by a selection criteria without any prior information. Unsupervised feature selection can be categorized into filter and wrapper methods (Liu and Motoda, 2007). Since the wrapper method involves evaluation that depends on learning algorithms and is computationally expensive (Talavera, 2005) we only investigate the *correlation matrix filters*. The correlation matrix is used as a metric to measure the similarity of features in a pairwise manner. In this paper, the Normalized Mutual Information metric (*NMI*, also written as *MI*) (He et al., 2005) is applied to generate the notion of relationship between feature f and f':

$$\begin{aligned} H(f) &= \sum_{i=1}^{n} P(i) log_2 P(i) \\ MI(f,f') &= H(f) + H(f') - H(f,f') \\ NMI(f,f') &= \frac{MI(f,f')}{max(H(f),H(f'))} \end{aligned} \tag{1}$$

where n is the number of discrete categories and $P(i)$ is the probability of observing category i. $H(f)$ is the entropy of f and $H(f,f')$ is the joint entropy of f and f'. $NMI \in (0,1]$ means two features are identical while $NMI = 0$ indicates that two features are independent.

Once the correlation matrix is produced, the next step is setting the criteria for feature selection. A novel process of unsupervised

Table 2
Datasets tested in this paper in V2015 and V2016.

Topic	Datasets	Purpose	Sec
Shoot detection	80 videos of TLs collected during V2015 (20 TLs × 4 Blocks). 1281 images were sampled from this footage and 20,046 shoot patches were manual labeled from those 1281 images as ground truth	To test the accuracy of the proposed shoot detection framework	Section 3.5
Shoot counting	240 videos of TLs collected during V2015 and V2016 (20 TLs per Blocks × 12 times). Shoot numbers over each TL and the video capture date were manually counted in the field as ground truth	To test the accuracy of proposed shoot counting algorithm	Section 4.2.3
Automated yield estimation	Four experimental blocks were filmed 12 times (4 times in V2015 and 8 times in V2016). Manual yield forecasting was conducted for each block based on the manual counted shoot number for each block at each time. The actual yield for each block was reported by vineyard managers at the end of each season	To test the accuracy of proposed automated yield estimation algorithm. The predicted yields generated by the proposed algorithm were compared with the manual yield estimation and the actual yield	Section 4.4

Fig. 3. 12 sampled images from four different blocks at different times. Shoot detection is complicated by: varying lighting conditions, undesired objects in the field of view (e.g. posts, cordon, grass, animals, wire, reflections), change of shoot position in the field of view, shadows and barren cordon. The color version of this figure is included in the online version of the journal.

Fig. 4. Initial segmented patches (shoots), after filtering shown as overlaid red rectangle. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

feature selection is illustrated in Algorithm 1. In this algorithm, the only manually set parameter is the selection threshold τ, which was empirically chosen to be 0.2. This τ divides the 28 features generated in Section 3.2 into two classes. Features with correlation value $\in [0, \tau]$ are selected for classification and those features are are linearly concatenated into a vector $F(f_1, f_2, \ldots, f_D)$.

Algorithm 1. Unsupervised Feature Selection on Patch and Texture Properties.

```
procedure FEATURE SELECTION PROCEDURE[a]
  Num ← number of input features
  τ ← threshold
  for i = 1 : Num − 1 do
    for j = i + 1 : Num do
      NMI_{i,j} ← generate all pairwise combinations of
      correlation relationship (NMI)
      if NMI_{i,j} ∈ (τ, 1] then
        hFeature_i ← label as highly correlated feature
        hFeature_j ← label as highly correlated feature
      end if
    end for
  end for
  lFeatures ← input features excluding highly correlated
  features
end procedure
```

[a] Notation follows **Matlab** Syntax.

Unlike scenarios in other publications, in this practical case, the selected features cannot be fixed since the condition of sampled images varies with the time and location of data collection. Vital features revealing intrinsics of data are therefore diverse from block to block and this automated selection procedure adds substantial robustness.

3.4. Unsupervised shoot classification

In the absence of cluster labels, the next step in the framework is to apply unsupervised learning for clustering data into two groups. Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$ denote N points (N initial patches), $\mathbf{x}_i = F(f_1, f_2, \ldots, f_d)$ refers to the selected feature vector with d dimensions for each point, and $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2]$ denotes the class of objects, which is restricted to shoot or non-shoot in this paper.

The K-means (MacQueen, 1967) clustering algorithm k is designed to classify N points into two classes $\mathbf{C} = \{c_1, c_2\}$by minimizing an objective function which we chose to be the Within-Cluster Sum of error Squares (WCSS) as seen in Eq. (2) below.

$$J = \arg\min_{\mathbf{C}} \sum_{i=1}^{2} \sum_{\mathbf{x} \in c_i} \|\mathbf{x} - \mu_i\|^2 \tag{2}$$

where μ_i is the mean of c_i. $\|\mathbf{x} - \mu_i\|^2$ is a chosen distance measured between a data point $\mathbf{x}$ to its cluster center μ_i. The distance in this paper is chosen as:

$$d(\mathbf{x}, \mu) = \sum_{d=1}^{D} (\mathbf{x}_d - \mu_{i,d})^2 \tag{3}$$

where D is the dimension of selected features. Then the clustering results are obtained by:

$$\mathbf{Y} = k(\mathbf{X}, 2, d(\mathbf{x}, \mu), J) \tag{4}$$

In order to classify results by K-means, all points in the high-dimensional feature space are projected into the principal three dimensions using PCA (Principal Component Analysis Jolliffe, 2002). The 3D volumes of the corresponding convex hulls of the two clusters are then calculated. The group with smaller convex hull volume is classified as shoot and other group as non-shoot because greater consistency is expected from the shoots. Some classification results are demonstrated in Fig. 5, where green rectangles contain patches classified as shoots and red rectangles are classified as non-shoot.

3.5. Experimental results for shoot detection

As summarized in Section 2 and Table 2, 20,046 shoot candidates were processed automatically by the proposed unsupervised detection framework. For verifying the performance of the developed framework, all initial patches were manually labeled as shoot or non-shoot for ground truth. The detection results were calculated using evaluation metrics that are defined as:

- *True Positive* (TP) refers to a true shoot automatically classified by the developed algorithm and also labeled as a real shoot manually.
- *True Negative* (TN) refers to patches correctly detected by the algorithm as non-shoot.
- *False Positive* (FP) represents a patch falsely identified as a real shoot.
- *False Negative* (FN) refers a shoot not correctly classified by the algorithm but manually labeled as a real shoot.

The *accuracy*, *recall*, *precision* and *F1 Score* were calculated from these metrics, defined as follows:

- *Accuracy* (ACC) refers to the percentage of true and false shoots automatically classified correctly in the total population

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{5}$$

- *Recall* (True Positive Rate, TPR) annotates the percentage of true shoots automatically classified out of all manually labeled real shoots

$$TPR = \frac{TP}{TP + FN} \tag{6}$$

(a) Fig.3b (b) Fig.3f (c) Fig.3h (d) Fig.3k

Fig. 5. Shoot detection results of Fig. 4. Green rectangles contain patches classified as shoots and red rectangles are classified as non-shoot. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

- *Precision* (Positive Predictive Value, PPV) annotates the percentage of true shoots automatically classified against all shoots automatically classified as positive, regardless of their manual label

$$PPV = \frac{TP}{TP + FP} \tag{7}$$

- *F1 Score* refers to a measure of good binary classification by considering *precision* and *recall* together. Statistically it is a weighted average of *precision* and *recall* and it reaches its best value at 1 and worst at 0

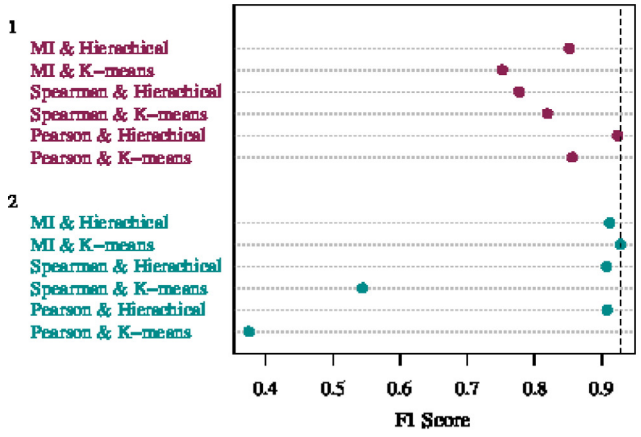
$$F1 = 2 \cdot \frac{PPV \cdot TPR}{PPV + TPR} \tag{8}$$

The results are presented in Table 3. Selecting independent features by Normalized Mutual Information and clustering data by K-means based on a block level can deal with different scene conditions at a certain level; achieving 86.83% *accuracy*, 91.52% *recall*, 89.18% *precision*, and 0.90 *F1-score* on average.

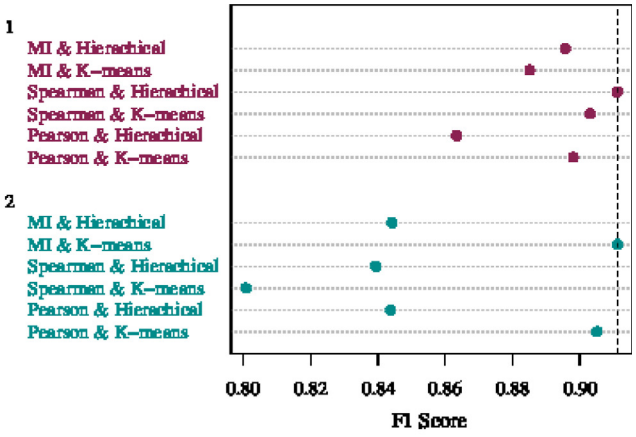
To evaluate the performance of the proposed method, another 11 combinations of feature selection and clustering approaches were tested on the same datasets for comparison. For the unsupervised feature selection step, Pearson's correlation and Spearman's rank correlation (Higgins, 2003) were tested in addition to NMI. Agglomerative hierarchical clustering (Hastie et al., 2005) was compared against K-means in the unsupervised clustering step.

Table 3
The average performance of detection results based on block level (MI & K-means).

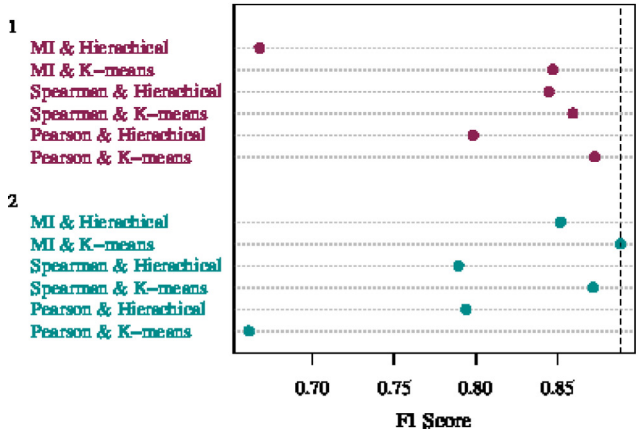
Block	Patches	Block level (MI & K-means)			
		ACC	TPR	PPV	F1
47A	3084	0.8977	0.9703	0.8891	0.9279
40A	4977	0.8923	0.9380	0.8857	0.9111
B4	7665	0.8770	0.9463	0.8368	0.8882
B12	4320	0.8060	0.8061	0.9554	0.8744
Average	5012	0.8683	0.9152	0.8918	0.9004



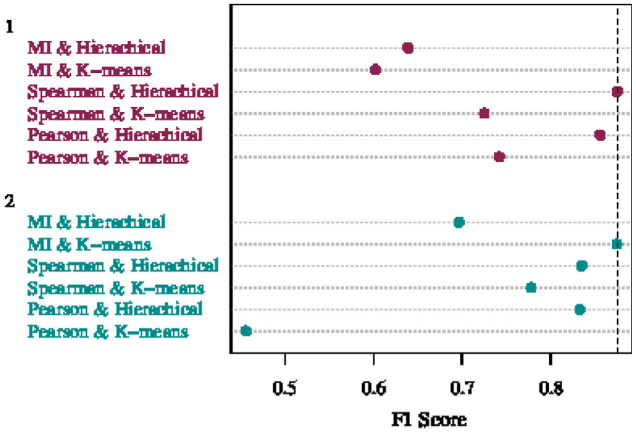
(a) Block 47A



(b) Block 40A



(c) Block B4



(d) Block B12

Fig. 6. The combination results of different feature selection and clustering methods. Group 1 indicates highly correlated features while group 2 in each sub-figure refers to features without redundancy. It can be seen that MI & K-means in group 2 has the best performance overall.

Feature selection with redundancy (Group 1) as opposed to without redundancy (Group 2) was also evaluated and Fig. 6 illustrates all 12 combinations of results. The four plots in Fig. 6 show that selecting features without redundancy (Group 2) by Normalised Mutual Information and clustering by K-means (MI & K-means) have the best and comparatively most robust performance in terms of *F1-score* for the four blocks.

4. Yield estimation based on shoot counting

Based on the information extracted from videos, a new objective forecasting equation at shoot stage for block yield is proposed. Given a robust method to detect shoots within an image, as proposed in Section 3, the number of shoots in an entire block can be estimated by filming every row of the block and providing selected frames from each row's video to the above shoot detection framework.

4.1. Extrapolating real shoot counts from images

The density of shoots, particularly when their leaves start to overlap, means that many patches contain more than one shoot and this is more noticeable in Chardonnay as opposed to Shiraz blocks. Therefore, an approach for extrapolating the real shoot counts is necessary. The clarity of video captured by the GoPro is also not sufficient to do further image processing. Statistical information detected from images, along with the size of each patch and number of patches, provides a good reference for the current block status.

In order to exclude outliers from the data, quartiles are used as they are resistant to the presence of outliers, which is ideal when dealing with different datasets. The 25th($Q1$), 50th($Q2$) and 75th ($Q3$) percentiles calculated for each dataset are used as vital quartile features to define the edge between 'normal' and 'extreme' size of classified patches.

On a block level, we have two metrics for each patch classified as containing shoots: one is the number of patches within each video frame (denoted as $X = [X_1, \ldots, X_N]$) while the other is the size of each patch (pixel counts for each individual patch, denoted as $Y = [y_1, \ldots, y_M]$), where N is the number of frames in a block and M is the number of patches in a block.

From the data processing for shoot counting by processing images, there are two scenarios that may arise for each frame:

- there are a large number of small patches, with several patches belonging to one shoot, as shown in Fig. 7a,
- there are few big patches, and each patch may contain a few shoots, as shown in Fig. 7b.

Thus, a logical way to merge or divide those patches is vital for accurately predicting the total shoot number in a block. For correcting the count originally generated after clustering, a new algorithm (summarized in Algorithm 2[3]) is proposed here for estimating the real shoot number based on the quartiles of the full block data:

- C is the corrected shoot count per frame by Algorithm 2.
- $Q1_X$ and $Q3_X$ are the 25th and 75th percentile of the dataset X of total number of shoot patches in a single frame for N frames within a block.
- $Q1_Y$ is the 25th percentile of the dataset Y of M patch sizes within a block.
- $Q2_{Y_i}$ is the 50th percentile of the dataset Y_i of m patch sizes within frame i.

Algorithm 2. Real Shoot Count per Frame.

for each Frame i **do**
  if $Q2_{Y_i} < Q1_Y$ **then**
    if $m > Q3_X$ **then**
      $index \leftarrow Y_i < Q1_Y$;
      $C = round(\sum Y_i(index)/Q3_Y) + \sum \neg index$
    else
      $C \leftarrow m$
    end if
  else if $Q2_{Y_i} > Q3_Y$ **then**
    if $m < Q1_X$ **then**
      $index \leftarrow Y_i > Q3_Y$;
      $C = round(\sum Y_i(index)/Q3_Y) + \sum \neg index$
    else
      $C \leftarrow m$
    end if
  else
    $C \leftarrow m$
  end if
end for

(a) Multiple patches may belong to one shoot.

(b) For more developed vines there is too much overlap between shoots and multiple shoots may exist in a single patch.

Fig. 7. Two scenarios showing the extreme relationships between patches and shoots.

4.2. Counting shoots over a length

Once the count per frame has been determined, this needs to be transformed into an actual count by first localizing each frame and then integration over a given length to give the density of shoots per meter.

4.2.1. Video frame localisation

Given the assumption of constant speed knowing the coordinates of the start and end frame, the location of any frame in the middle of this video can be calculated by linear interpolation. For a non-constant velocity, the method of estimating velocity by Cossell et al. (2016) can be substituted without loss of generality. For non-straight rows and those varying in altitude, this can be readily adapted if the shape of the row is known. In our four experimental blocks, the vines are planted along straight rows and the vehicle that was used for filming drove at constant speed, around 10 km/h.

For a row with a certain length, a video with n frames is recorded from the beginning to the end of this row. Assuming the starting frame's location is $f_1 = [x_1, y_1]$ while the end one is $f_n = [x_n, y_n]$, as shown in Fig. 8, the location of frame i can be located as:

$$f_i = [x_i, y_i] = \left[x_1 + \frac{x_n - x_1}{n-1} \cdot (i-1), y_1 + \frac{x_n - y_1}{n-1} \cdot (i-1)\right] \quad (9)$$

where $[x, y]$ are the real world coordinates of the related frame.

4.2.2. Integrating image counts to obtain density per meter

The vehicle was driven along the center of each two rows with the space between rows estimated at $2.9 - 3.1$ meters. The distance between the camera and center of the canopy is

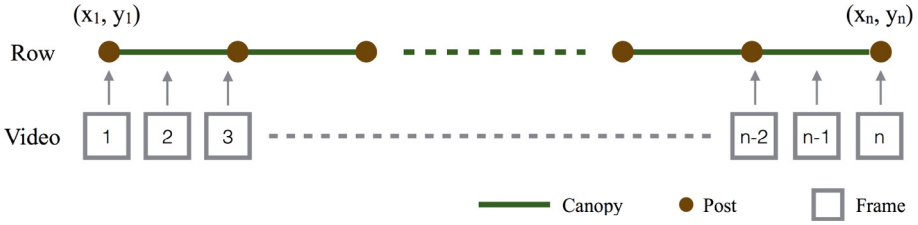


Fig. 8. Linear interpolation for about frame registration, assuming straight rows, constant vehicle velocity and relatively flat rows.

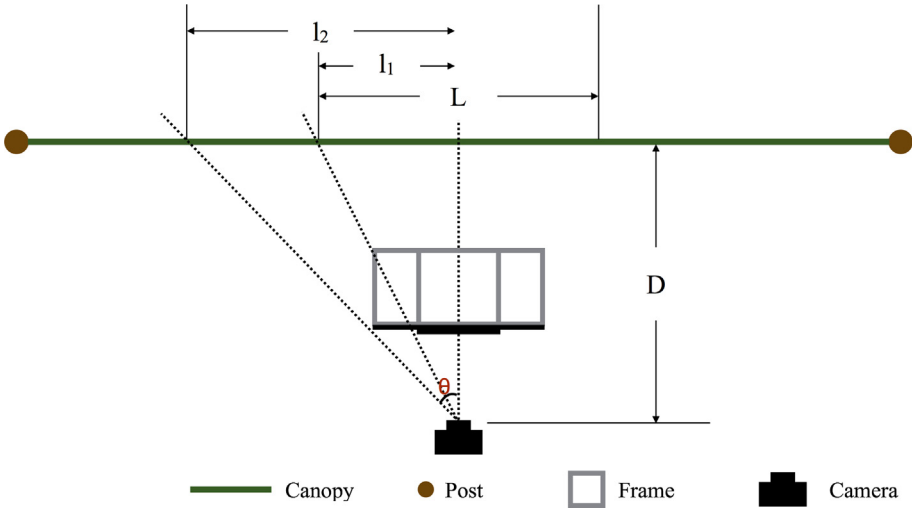


Fig. 9. The geometrical model of camera and cordon.

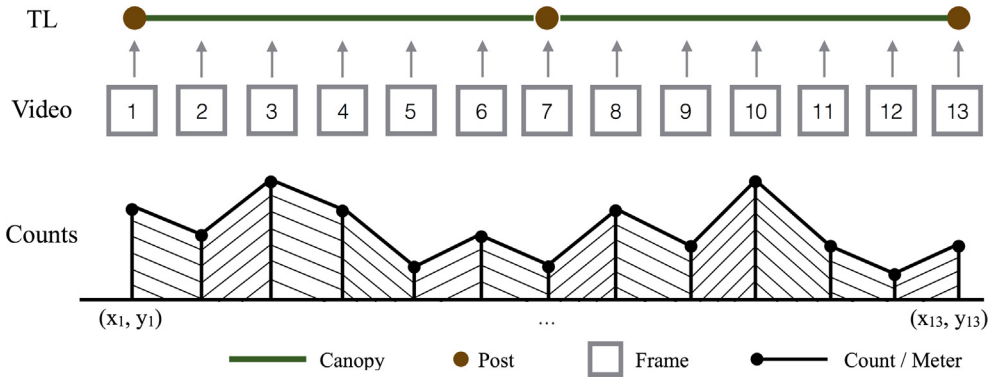


Fig. 10. Counting shoots over a TL by sampling frames in a related video.

Table 4
Shoot count error per TL, compared with E-L stage ranges across the block at the time of filming.

Error (%)/E-L stage	Season 1	Season 2			
Block	FT1.2	FT2.3	FT2.4	FT2.5	FT2.6
40A	−36.10 E-L4-9		**−28.58** **EL9-12**	−29.54 E-L12-15	
47A	**−6.10** **E-L7-11**	19.37 E-L6-7		**−2.46** **E-L9-12**	
B4	**−10.31** **E-L9-10**		**−13.71** **E-L7-9**		−67.54 E-L13-15+
B12	−30.47 E-L11-12		**−9.74** **E-L8-10**		−21.64 E-L13-15+

approximately 0.4 meters. The size of each frame is 1080×1920 pixels and the horizontal field of view of the GoPro is 94.4°. The frame was cropped into 1080×1080 based on the center for convenience of image processing. The geometrical model of this case is illustrated in Fig. 9.

Based on this information, the actual length of cordon covered by a frame can be calculated by Eq. (10), where θ is half the field of view of the camera (here a GoPro) and D is the distance between the camera and cordon.

$$\begin{aligned} l_1 &= 1080/1920 \cdot l_2 \\ l_2 &= \tan\theta \cdot D \\ L &= 2 \cdot l_1 = 2 \cdot 1080/1920 \cdot \tan\theta \cdot D \end{aligned} \tag{10}$$

The corrected number of shoots per meter is calculated as C_i/L, where i is the frame number. Given a video of a vine row, and knowing the positions of the end posts, each sampled frame is located by the method described in Section 4.2.1.

Since C_i varies dramatically from frame to frame, we calculate the total shoot number along a row by integration. For example, to calculate the total number of shoots in a TL we sum up all filled areas under the lines shown in Fig. 10. The overall calculation procedure is described in Algorithm 3.[1] This per-row count is then summed across the block to estimate the total number of shoots in the block.

Algorithm 3. Counting Shoots over a Length.

$C_i \leftarrow$ Image Processing sampled frames
$f_i \leftarrow$ Sampled frames registration
for $i = 2 : n$ **do**
  $Area_{i-1} = (C_i + C_{i-1}) \div L \times (f_i - f_{i-1}) \div 2$
end for
$Counts = \sum_{i=1}^{n-1} Area_i$

4.2.3. Experimental results on TLs

As described in Section 2, 12 sets of videos in TLs were recorded over two seasons at multiple phenological stages. We will use the modified Eichhorn-Lorenz (E-L) system (Dry and Coombe, 2009) in this paper, hereafter denoting each phenological stage as an E-L stage. The footage was processed for shoot counting by image processing as described in Sections 4.1 and 4.2, and the total number of shoots calculated for each of the 20 TLs in each block. Manual shoot counts were collected from the TLs in the field at the corresponding stages to validate the accuracy of the shoot detection framework.

The calculated results are listed in Table 4, along with the corresponding E-L stages. It is clear that the early (prior to E-L stage 9) and later (beyond E-L stage 12) results for shoot counting are less reliable. This was due to shoots being too small to reliably detect (in the case of E-L stage 4) or too vigorous to differentiate visible (in the case of beyond E-L stage 12). For the remaining 6 datasets, shown in boldface in Table 4, the average accuracy of shoot detection was found to be 88.18% against real ground truth data, validating the overall shoot counting framework and laying the groundwork for an early stage yield estimation.

In short, the method of shoot detection presented in Section 3, in combination with the algorithm and method for extrapolating these to real shoot counts as presented in Sections 4.1 and 4.2 are able to count the number of shoots to within 12% of the real value, using only a GoPro camera, backing board and vehicle. This holds true for blocks from E-L stages 7–11, with a shoot density of less than 40 shoots per meter. This marks the first time shoots have been counted automatically in vineyards and lays the foundation for early stage yield estimation as well as mapping of the spatial variation across a block at such an early stage.

4.3. Converting shoot counts to yield

Once the number of shoots can be estimated within a block, an early estimate of potential yield can be calculated. Eq. (11) shows the proposed relationship between shoot counts and estimated yield. It is based on using the number of shoots as an early indicator of the number of bunches, the main variable determining yield.

$$PY = NS \times PRV \times R_{BS} \times BW \times (1 - PR) \times H_e \times (1 - SP) \tag{11}$$

where:

- PY is the total predicted block yield (mass of fruit without rachis, assuming machine harvested so as to leave the rachis on the vine).
- NS is the number of shoots detected from videos at the block level.
- PRV is the proportion of recorded VID for the whole block (should videos of rows be missing or incomplete).
- R_{BS} is the ratio of bunches to shoots from historical data.
- BW is the average bunch weight at harvest in previous seasons.
- PR is the proportion of rachis weight to bunch weight.
- H_e is a harvester efficiency factor.
- SP is the percentage of any destructively sampled fruit before harvest.

The ratio of bunches to shoots (R_{BS}) varies among cultivars, the location of vineyards, the timing of water shoots and the weather and thus may vary from season to season. In this work, R_{BS} was assumed to be one, as historical records from the vineyard manager showed the range to be 0.98 to 1.3. Practically, it can either be measured manually by counting the number of shoots in particular locations two weeks after bud burst and bunches in the same

[1] Notation follows **Matlab** Syntax.

locations closer to harvest, or using automated methods such as the proposed solution in Section 3 and previous work by Liu and Whitty (2015).

Harvester efficiency was a factor introduced to account for the ratio of fruit on the vine that actually ends up on the winery weighbridge; values recommended by Martin and Dunn (2003) were used in this paper.

4.4. Yield estimation in experimental blocks

In total, we collected videos in two successive growth seasons: V2015 and V2016. In the first season, we only collected footage once for each block at the shoot stage. During this time, we found was that it was difficult to define the optimal time to capture video of shoots. Hence in the second season, we collected multiple datasets for each block before the inflorescence stage, as shown in Table 4. Four additional video datasets for which corresponding manual counts were not available were also processed, giving a total of 16 datasets.

To compare the results of our proposed prediction method with traditional approaches and gauge its accuracy, we counted shoots by hand at 20 marked locations (TLs) across a block as mentioned in Section 2 for ground truth. To estimate the total number of shoots in a block manually, we extrapolated the sampled counts up to the length of all rows in a block, minus the known length of non-bearing vine. Although this method of manual shoot counting is rarely undertaken and somewhat naïve, it follows the pattern of traditional later stage yield estimation.

After getting the shoot number for a whole block (either manually or by computer vision), we applied Eq. (11) to calculate the final yield. Table 5 provides a comparison of the predicted yield against the actual weighbridge values.

To compare the yield estimation results accomplished by the proposed method and a naïve manual sampling approach, Table 6 demonstrates the error between actual final yield at the weighbridge and predicted yield by both methods. It is obvious that yields estimated by the proposed method are more accurate in general.

5. Spatial yield variation map generation

In parallel to a system for yield estimation, we introduce a method for GPS-free variation mapping which is economical for farmers and easy to apply. It uses the outputs from image processing to generate a predicted yield variation (or yield potential) map as well as an estimated canopy vigor map.

The discrete values of data from image processing on each frame in each block are kriged to generate a continuous surface across the entire block. In this paper, the default settings in ArcGIS (v10.3) were applied and a subset of 25% of the frames for each block in each E-L stage is used in order to match the spatial resolution of the provided yield monitor data, for easier visualization.

Based on the shoot counts processed from video frames, combined with Eq. (11), a predicted yield map can be produced across the entire block. The top row in Fig. 11 illustrates the predicted yield variation maps for four experimental blocks. It is important to note that this prediction is being carried out five months prior to harvest, and makes the assumption that the spatial variation of yield within a block will tend towards the final spatial variation at harvest. Finally, it assumes that no bunch thinning operations will take place and is thus more suitable for low to medium yielding regions.

In addition, map of canopy vigor can be generated based on the total green pixel count of those shoots (Cossell et al., 2016). The

Table 5
Comparison of the estimated yield at shoot stage by the system presented in this paper (CV) against extrapolating manual counts (Manual) and the weighbridge weights (Actual) for fruit harvested in the each block. FT refers the field trips as defined in Section 2.

Yield (t)		Season 1	Season 2			
Block		FT1.2	FT2.3	FT2.4	FT2.5	FT2.6
40A	CV	45.04	39.96	43.04	60.86	
	Manual	45.35		45.38		
	Actual	47.92		67.1		
47A	CV	13.40	57.41		67.38	
	Manual	48.60		46.77		
	Actual	36.98		59.3		
B4	CV	49.46		63.06	61.09	82.91
	Manual	40.82		38.90		
	Actual	30.70		69.0		
B12	CV	25.20	46.89	46.46	41.42	75.42
	Manual	29.43		23.35		
	Actual	24.91		64.7		

Table 6
Comparison of the error in yield estimation between the proposed method and a naïve manual approach.

Error (%)		Season 1	Season 2			
Block		FT1.2	FT2.3	FT2.4	FT2.5	FT2.6
40A	CV	−6.01	−40.45	−35.86	−9.30	
	Manual	−5.36		-32.37		
47A	CV	−63.77	−3.22		13.59	
	Manual	31.42		-21.16		
B4	CV	61.12		−8.61	−11.47	18.97
	Manual	32.96		-43.62		
B12	CV	1.18	−27.57	−28.24	−36.02	16.49
	Manual	18.15		-63.93		

(a) 40A at FT2.4 (2 Oct 2015) (b) 47A at FT2.3 (24 Sep 2015) (c) B4 at FT2.4 (2 Oct 2015) (d) B12 at FT2.4 (2 Oct 2015)

(e) 40A at FT2.4 (2 Oct 2015) (f) 47A at FT2.3 (24 Sep 2015) (g) B4 at FT2.4 (2 Oct 2015) (h) B12 at FT2.4 (2 Oct 2015)

Fig. 11. Predicted yield variation maps (top row, red indicates higher predicted yield than blue) and the corresponding estimated canopy variation maps (bottom row, green refers to higher canopy vigor than red). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

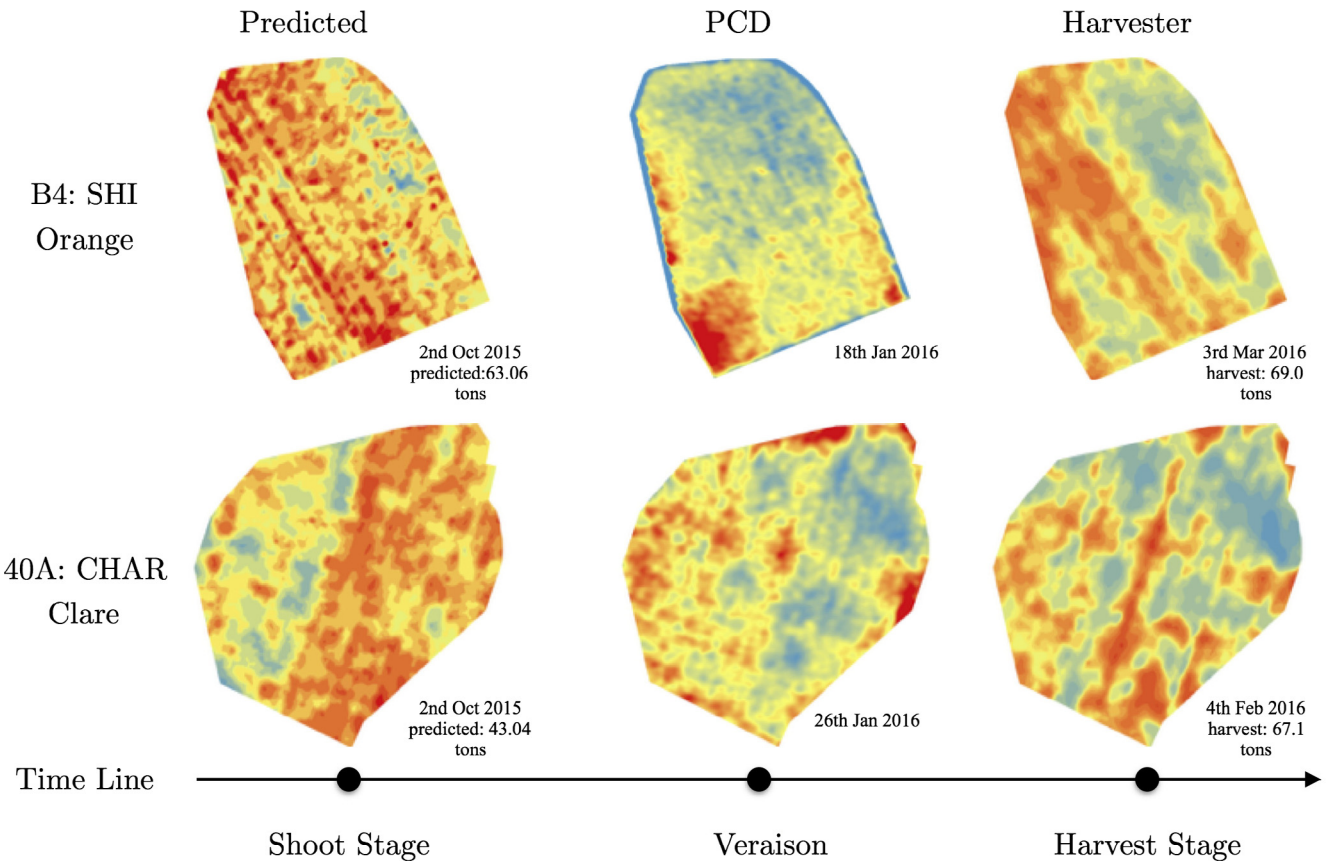


Fig. 12. The comparison of timing to obtain yield variation map. Maps in the first column are generated by proposed method; maps in the second column are PCD maps collected from aerial imagery; maps in the third column are generated by a commercial on-harvester monitoring device.

bottom row in Fig. 11 presents the estimated current developing status of canopy in the field for the four experimental blocks.

The ability to detect shoots in order to predict yield as well as map shoot density is of notable positive value for the wine industry. The estimated canopy vigor maps do not necessary match with the corresponding predicted yield variation maps. Whereas the number of shoots (once formed) will remain relatively constant within a season, the vigor changes rapidly, hence slightly different patterns are visible in the pairs of maps. The shoot number is generated based on classified shoot patches and related statistical analysis, as explained in Section 4. The canopy vigor is the sum of the areas of classified shoots. Overall, the estimated canopy vigor maps produced are still able to provide a good reference for vineyard managers about which sections are growing more rapidly; which in turn can aid irrigation and management decisions.

Plant Cell Density (PCD) maps can give information about canopy vigor, as discussed before, and may be related to the final yield variation map but that is not guaranteed as seen in block B4 in Fig. 12. Whereas the predicted yield variation maps from shoot counts will vary depending on the relative development of the shoots, through conditions such as aspect, altitude (see Table 1) and irrigation, from Fig. 12 it can be observed that the predicted yield variation maps for both blocks start to form a similar pattern to the actual yield map, even though imagery was collected four to five months prior to harvest. PCD maps on the other hand are more commonly collected at veraison, when the canopy is fully developed, and when taking management actions to regularise yield and improve profit is more difficult. Having a predicted yield variation map early in the season can provide a clear benefit for vineyard managers to act, as they have at least a map of the yield potential based on the shoot counts.

6. Conclusion

In this paper, a completely automatic system for grapevine yield estimation by visual shoot detection has been presented. It combines a robust shoot detection framework with an algorithm and method for converting the image processing results to actual shoot counts using only a GoPro camera, backing board and vehicle.

The shoot detection framework is fully automated, through the use of image pre-processing, feature extraction, unsupervised feature selection and unsupervised clustering as a final classification step. This eliminates manual labeling or training while increasing robustness to varying real-world conditions.

Experiments were conducted on Shiraz and Chardonnay blocks in commercial vineyards across two seasons and at two locations to reduce bias from a single terroir. In total 1,281 images which contain 20,046 shoot patches were tested and the shoot detection framework achieved an *accuracy* of 86.83%, *recall* of 91.52%, *precision* of 89.18% and an *F1-score* of 0.90 on average. This demonstrated robustness to both read and white grape varieties and various illumination conditions to a good level given the large scale of the data set.

In terms of the error of visual shoot counting versus the real counts, the developed algorithm and method in Section 4.2 achieved an average error of −11.82% for Shiraz and Chardonnay. The counting results were found to be most promising when videos were captured around E-L stage 9. At this stage, the absolute predicted yield estimation error of the system ranged from 1.18% to 36.02%, remarkable given these predictions were made five months prior to harvest.

The introduced mapping methods for generating a predicted yield variation map based on the shoot detection and counting algorithms has clear advantages over PCD and harvest yield monitor maps, as demonstrated in Fig. 12. By using only natural light and consumer grade GoPro cameras for data collection, widespread adoption of the technology is facilitated, increasing the potential for industry benefit.

To the best of the authors' knowledge, this is the earliest stage at which complete maps of the variation of yield potential within a block have been generated. By providing these critical maps, farmers are better able to understand crop variation, and thereby adjust their management practices to their advantage.

This approach relies heavily on an accurate estimate of the bunch to shoot ratio, which can be derived from bud fruitfulness. If this data can be collected on a sufficiently wide scale, it will inform the selection of the bunch to shoot ratio and improve the accuracy of the yield estimates, however the time and effort required to collect and analyze these samples is significant and prone to selection bias. Future work will focus on testing this approach on the coming years' datasets, extending the analysis to different trellis types and cultivars.

Acknowledgement

Many thanks to Jarretts of Orange and Treasury Wine Estates for provision of facilities critical for the conducting of this research and Wine Australia for financial support through project DPI 1401, "Improved Yield Prediction for the Australian Wine Industry." The authors would also like to thank Paul Petrie and Angus Davidson for assistance with data collection, and Lily Shi for data labeling.

References

Chamelat, R., Rosso, E., Choksuriwong, A., Rosenberger, C., Laurent, H., Bro, P., 2006. Grape detection by image processing. In: IECON 2006-32nd Annual Conference on IEEE Industrial Electronics. IEEE, pp. 3697–3702.

Correa, C., Valero, C., Barreiro, P., Diago, M.P., Tardáguila, J., 2011. A comparison of fuzzy clustering algorithms applied to feature extraction on vineyard. In: Proceedings of the XIV Conference of the Spanish Association for Artificial Intelligence.

Cossell, S., Whitty, M., Liu, S., Tang, J., 2016. Spatial map generation from low cost ground vehicle mounted monocular camera. IFAC-PapersOnLine 49 (16), 231–236.

Diago, M.-P., Correa, C., Millán, B., Barreiro, P., Valero, C., Tardaguila, J., 2012. Grapevine yield and leaf area estimation using supervised classification methodology on rgb images taken under field conditions. Sensors 12 (12), 16988–17006.

Dorj, U.-O., Lee, M., Lee, K.-k., 2013. A computer vision algorithm for tangerineyield estimation. Int. J. Bio-Sci. Bio-Technol. 5 (5), 101–110.

Dry, P., Coombe, B.G., 2009. Viticulture: Volume 1-Resources, Royal New Zealand Foundation of the Blind.

Fernández, R., Montes, H., Salinas, C., Sarria, J., Armada, M., 2013. Combination of rgb and multispectral imagery for discrimination of cabernet sauvignon grapevine elements. Sensors 13 (6), 7838–7859.

Font, D., Tresanchez, M., Martínez, D., Moreno, J., Clotet, E., Palacín, J., 2015. Vineyard yield estimation based on the analysis of high resolution images obtained with artificial illumination at night. Sensors 15 (4), 8284–8301.

Gonzalez, R.C., Woods, R.E., Eddins, S.L., 2004. Digital image processing using MATLAB, Pearson Education India.

Hastie, T., Tibshirani, R., Friedman, J., Franklin, J., 2005. The elements of statistical learning: data mining, inference and prediction. Math. Intell. 27 (2), 83–85.

He, X., Cai, D., Niyogi, P., 2005. Laplacian score for feature selection. In: Advances in Neural Information Processing Systems, pp. 507–514.

Higgins, J.J., 2003. Introduction to modern nonparametric statistics.

Jolliffe, I., 2002. Principal Component Analysis. Wiley Online Library.

Kendall, M.G., et al., 1946. The Advanced Theory of Statistics, second ed.

Klodt, M., Herzog, K., Töpfer, R., Cremers, D., 2015. Field phenotyping of grapevine growth using dense stereo reconstruction. BMC Bioinform. 16 (1), 143.

Liu, H., Motoda, H., 2007. Computational Methods of Feature Selection. CRC Press.

Liu, S., Whitty, M., 2015. Automatic grape bunch detection in vineyards with a svm classifier. J. Appl. Logic.

Liu, S., Marden, S., Whitty, M., 2013. Towards automated yield estimation in viticulture. In: Proceedings of the Australasian Conference on Robotics and Automation, Sydney, Australia, vol. 24.

MacQueen, J., 1967. Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, Oakland, CA, USA, pp. 281–297.

Martin, R.D.S., Dunn, G., 2003. How to forecast wine grape deliveries. Technique report, Department of Primary Industries.

Meunkaewjinda, A., Kumsawat, P., Attakitmongcol, K., Srikaew, A., 2008. Grape leaf disease detection from color imagery using hybrid intelligent system. 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2008 (ECTI-CON 2008), vol. 1. IEEE, pp. 513–516.

Mitchell, D.P., 1991. Spectrally optimal sampling for distribution ray tracing. ACM SIGGRAPH Comput. Graph. 25 (4), 157–164.

Nuske, S., Wilshusen, K., Achar, S., Yoder, L., Narasimhan, S., Singh, S., 2014. Automated visual yield estimation in vineyards. J. Field Robot. 31 (5), 837–860.

Otsu, N., 1975. A threshold selection method from gray-level histograms. Automatica 11 (285–296), 23–27.

Payne, A.B., Walsh, K.B., Subedi, P., Jarvis, D., 2013. Estimation of mango crop yield using image analysis–segmentation method. Comput. Electron. Agric. 91, 57–64.

Talavera, L., 2005. An evaluation of filter and wrapper methods for feature selection in categorical clustering. In: Advances in Intelligent Data Analysis VI. Springer, pp. 440–451.

Wang, Q., Nuske, S., Bergerman, M., Singh, S., 2013. Automated crop yield estimation for apple orchards. In: Experimental Robotics. Springer, pp. 745–758.