Precision Agric
DOI 10.1007/s11119-013-9323-8

# Immature peach detection in colour images acquired in natural illumination conditions using statistical classifiers and neural network

**Ferhat Kurtulmus · Won Suk Lee · Ali Vardar**



**Abstract** Detection of immature peach fruits would help growers to create yield maps which are very useful tools for adjusting management practices during the fruit maturing stages. Machine vision algorithms were developed to detect and count immature peach fruit in natural canopies using colour images. This study was the first effort to detect immature peach fruit in natural environment to the authors' knowledge. Captured images had various illumination conditions due to both direct sunlight and diffusive light conditions that make the fruit detection task more difficult. A training set and a validation set were used to develop and to test the algorithms. Different image scanning methods including finding potential fruit regions were developed and used to parse fruit objects in the natural canopy image. Circular Gabor texture analysis and 'eigenfruit' approach (inspired by the 'eigenface' face detection and recognition method) were used for feature extraction. Statistical classifiers, a neural network and a support vector machine classifier were built and used for detecting peach fruit. A blob analysis was performed to merge multiple detections for the same peach fruit. Performance of the classifiers and image scanning methods were introduced and evaluated. Using the proposed algorithms, 84.6, 77.9 and 71.2 % of the actual fruits were successfully detected using three different image scanning methods for the validation set.

**Keywords** Computer vision · Fruit detection · Immature peach · Yield mapping · Statistical classifiers

This study is a part of PhD thesis of the first author accepted on 23.11.2012 by the Graduate School of Natural and Applied Sciences of Uludag University.

F. Kurtulmus (✉) · A. Vardar
Department of Biosystems Engineering, Faculty of Agriculture, Uludag University, 16059 Bursa, Turkey
e-mail: ferhatk@uludag.edu.tr

W. S. Lee
Department of Agricultural and Biological Engineering, University of Florida, Rogers Hall, Museum Road, Gainesville, FL 32611, USA

Published online: 14 June 2013

## Introduction

Peach production in the World is about 10 Mt, and harvested area of peaches and nectarines is 271 567 ha in Europe according to the FAO (2010). Like any fruit grower, peach growers encounter unpredictable costs due to in-field spatial variability such as tree size, soil type, soil fertility and water content. Precision agriculture tools such as yield mapping try to find solutions for growers to understand reasons for in-field spatial variability. So, those tools are necessary for any producers who consider precision farming. In parallel with developing precision agriculture technologies, many commercial yield mapping systems have started being used. Yield mapping systems provide yield data at the mature stage of the fruit. By the time yield data is available, it is too late for growers to adjust site-specific management practices. Yield prediction maps which could be obtained at an earlier stage of fruit growth could be used to adjust management practices before harvesting. Early estimation of the yield is also needed for planning harvesting operations and deciding proper investments in advance. To develop a yield prediction mapping system, machine vision and image processing methods could help to detect immature fruit in the natural canopy.

For fruit locating or counting, computer vision and image processing techniques have been investigated in numerous studies. One of the earliest studies by Parrish and Goksel (1977) was to develop a system to detect mature apples for a harvesting robot. Pla et al. (1993) worked on detecting spherical objects using an artificial light source, and tested their detection method for citrus fruit at the initial stages of maturity. Jimenez et al. (2000) presented a survey about on-tree fruit detection studies up to year 2000. Their research outlined many of those studies using different image analysis methods and imaging systems such as black/white, coloured and spectral. Some research activities were also carried out for detecting apple fruit using thermal imaging techniques (e.g., Stajnko et al. 2004; Wachs et al. 2009). Multispectral and hyperspectral imaging methods were also utilized for detection of citrus fruit (e.g., Bulanon et al. 2010; Okamoto and Lee 2009). Most of these studies detected fruits at the mature stage or detected fruits using multispectral, hyperspectral and thermal imaging techniques which use high-priced equipment.

Due to different viewing angle combinations and occlusions in natural canopy images taken under natural conditions, many peach fruits cannot be seen separately from each other, and fruit objects do not usually have distinctive colour and uniform surface. In a peach canopy image, occlusion of leaves is a challenging factor for a fruit detection algorithm. In addition to colour similarity between leaves and some young fruits, leaves make immature peach detection difficult by partially or completely covering the fruits. Under natural illumination conditions, each captured image has its own specific illumination condition due to light angles, diffuse light and variable shadows on the fruits. These uneven circumstances of the natural canopy images require a detection algorithm to be capable of coping with images in widely different conditions.

The purpose of this study was to explore the feasibility of detecting immature peach fruit using just regular colour images. Detection of immature peach fruits would help growers to create yield prediction maps which would be very useful tools for adjusting management practices during the fruit maturing stages. Here, computer vision algorithms to detect and count immature peach fruits in natural images were developed and presented using regular colour images with artificial classifiers, three different image scanning methods, features based on principal component analysis and circular Gabor texture. Furthermore, it was intended to compare classifier performance for immature peach detection. As far as the authors know, the present study was the first effort to develop a computer vision algorithm to detect and count immature peach fruit in natural canopy images taken from a peach grove under natural illumination conditions.

## Materials and methods

### Image acquisition and pre-processing

The peach variety of this study was Elegant Lady (*Prunus persica*), and the peach grove was located in Bursa (29°17′6.17″E, 40°13′54.96″N), Turkey. A total of 96 peach tree canopy images were taken at an early stage of the immature peach fruit when fruit diameters were about 40–50 mm in June 2–9, 2011. Using a regular colour camera (Nikon Coolpix L22), peach canopy images were recorded randomly in the grove at various times in daytime. The captured images had various illumination conditions according to both direct sunlight and diffusive light conditions that could make fruit detection more difficult. The images were taken from both the sunny and shadow sides of the canopy. Using these images, training and validation image sets were constituted by randomly selecting the images for training and testing fruit detection algorithms.

The training set consisted of 32 peach canopy images, and a total of 64 images were used to create a validation image set. Image resolution was 800 $\times$ 600 pixels. To decrease the disadvantages of uneven illumination, an illumination enhancement method was followed as a part of the fruit detection algorithm. To apply illumination enhancement to an image, Hue–Saturation–Intensity (HSI) colour space conversion was first made. Colour layers were then separated to obtain the colour layer I component representing illumination (I) of the image. Natural logarithm function was applied to each pixel of the I component individually. After this stage, resulting pixel values were normalized to a suitable range (0–255) which was meaningful for representing an image. A histogram equalization procedure, using the 'adapthisteq' function of MATLAB (2012a), The MathWorks, Inc., Natick, Massachusetts, USA) based on contrast-limited adaptive histogram equalisation algorithm (Zuiderveld 1994) was then applied to the resulting image from the logarithm transform. Image reconstruction was performed by merging H, S and processed I layers of the image to obtain final enhanced image. This procedure was carried out for all images in the training and validation image sets before building the classifiers and fruit detection experiments as a pre-processing step.

### Image scanning methods with potential fruit regions and algorithm steps

In the present study, three different image scanning methods were performed for parsing fruit objects in the canopy images, and performance results of these methods were compared. Method 1 was to scan an entire image with a given step, method 2 was to extract potential fruit regions with a morphological opening operation of binary image, and method 3 was to extract potential fruit regions with a radial symmetry transform. These algorithm steps including three image scanning methods and classifiers are described in the following sections. A flow diagram, shown in Fig. 1, summarizes algorithm steps for all image scanning methods and classifiers used in this study.

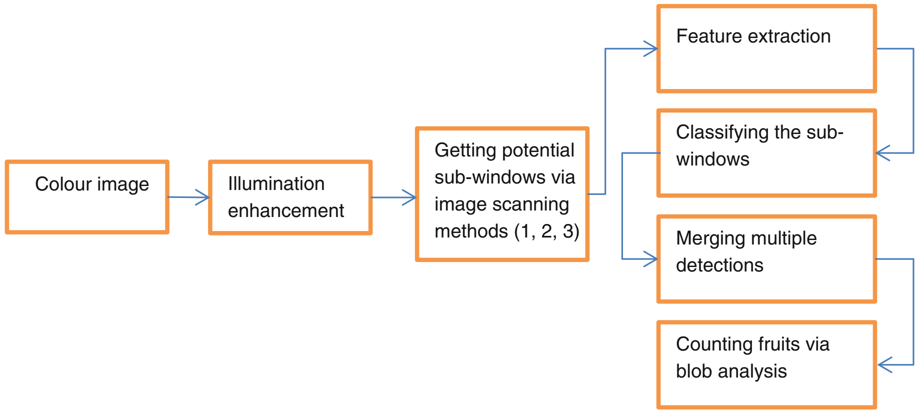


**Fig. 1** Flow diagram of the algorithms

*Method 1*

The first image scanning approach used in this research consisted of simply scanning an entire image using a shifting sub-window (square sub-region to take any local part of an entire image). In this method, a 20 pixel shifting step increment was used for both horizontal and vertical directions. This value which was determined from the training set was acceptable to not skip any distinctive fruit regions in the training set and to avoid single detection of touching fruits. In the scanning process, each sub-window scanned an image at three different scales as nested square sub-regions. Namely, each sub-window represented three nested square regions at three different scales. In the training set, fruit diameters were about 80–130 pixels, so 80 $\times$ 80, 100 $\times$ 100 and 130 $\times$ 130 pixels nested square regions were chosen as nested sub-window sizes.

By using method 1, all sub-windows of the entire peach canopy image were not classified by the classifiers whose details are described later. A background elimination method was also carried out to reduce false detections. For this purpose, a colour-based method was followed. Despite low colour contrast between the background and immature fruit objects in the peach canopy images, histogram distributions were investigated to find any distinctive features between fruits and background objects. Using the training set of peach canopy images, fruits and background objects were manually cropped, and then two mosaic images were created for fruit and background samples by sorting manually cropped samples in the image. Histogram curves of the mosaic images were observed in HSI and luminance–chrominance in blue-chrominance in red (YCbCr) colour spaces by splitting colour components. Although some immature peach fruits are relatively distinctive visually because of their different texture, histogram curves of the background and fruit samples were highly overlapped. However, little colour differences were found in Cr, Cb and H components, and colour thresholds for pixel discrimination were then obtained by observing histogram distributions of the three colour components to create a binary image. Using histograms of the mosaic images, thresholds for background pixels were found, and a binary mask image representing background pixels was created. While value 1 was used to represent background pixels, fruit pixels were assigned to value of 0.

The binary and corresponding colour images accounted for registered image data for the image scanning process. The object scanning process was carried out on different layers of data (corresponding to the same image scene) such as binary image and different colour components simultaneously. In the scanning process using method 1, a shifting sub-window process was performed over registered image data. A pixel ratio of the number of white pixels to the number of all pixels in the sub-window over the binary image was used to determine whether the sub-window was fruit or background, and thereby to eliminate that sub-window. A circular region of interest (ROI) was defined as a maximum circle inside the square sub-window, and was used for excluding background pixels at the corners of the sub-window due to the circular shape of the peach fruit. Namely, the pixels at the corners of the sub-window were not taken into consideration when calculating the pixel ratio. Sub-windows whose pixel ratio exceeded a threshold were eliminated. This threshold value was determined based on average of thresholds found by a trial and error method for each image in the training set. This value was found to be 0.5. Once sub-windows were eliminated, the algorithm skipped to the next sub-window. The sub-windows not eliminated by method 1 were processed in the next algorithm steps. Figure 2 shows an example peach canopy image, its binary image representing background pixels and a sub-window including a circular ROI by method 1.

*Method 2*

In this study, different methods were also investigated to classify only potential sub-regions of a given image to find fruit objects. Such a method was considered to reduce false detections of the fruit detection algorithm and to reduce computation time as well. Method 2 consisted of a set of image processing operations. In this method, to extract potential fruit locations from any peach canopy image, a colour histogram-based image binarization was performed first, which was also used in method 1. A created binary image output, shown in Fig. 3b, consisted of large connected and unconnected white pixel regions including background and fruit pixels. To narrow down the fruit searching area and thereby to extract potential fruit locations, a morphological opening operation was performed on the binary image using a structuring disk element. The radius of a flat, disk-shaped structuring element used in this study was determined to be 20 pixels from the training set. The result of the image opening process was another binary image containing circular-shaped objects. The morphological opening process would also help to avoid single detection of touching fruits by shrinking connected blobs. The steps of method 2 are shown in Fig. 3. A blob

a b

**Fig. 2** An example peach canopy image (**a**) and its binary form with a sub-window and circular ROI (**b**) analysis was carried out to locate blob centroids. The image scanning process was performed using square sub-windows over corresponding identified blob centroids. To increase the possibility of fruit detection, north, east, south, and west neighbours of each blob were also scanned as potential centres of the fruits. These neighbours were located at 20 pixel distance from identified centroid by the blob analysis. Each potential centre was also scanned using three different scales (80, 100 and 130 pixels square sub-windows). Thus, artificial classifiers, whose details were described later, performed 5 × 3 times for each identified potential fruit sub-region.

*Method 3*

Another approach used in the present work to find possible fruit centres was the radial symmetry transform (RST) by Loy and Zelinsky (2003). This method uses the gradient of an image to locate points of high radial symmetry. It computes the contribution for each pixel from their neighbouring pixels along the direction of the gradients (Xiang et al. 2011). It was considered that this method would help to extract points of interest for peach fruit detection because of the radial shapes of the fruits. For applying RST to any image, it is required to know possible radii of radial objects in advance. Therefore, 80, 100 and 130 pixels radii values was selected and processed according to fruit diameters found from the training set. To obtain gradient $g(p)$ at each pixel $p$, a Sobel mask convolution was applied to pre-converted grayscale peach canopy image. Then, the gradient vectors are normalized by:

$$g_{norm}(p) = g(p)\|g(p)\| \tag{1}$$

Then, the co-ordinates of positively and negatively affected pixels $p_{ve}(p)$ for each radius are given by:

$$p_{ve}(p) = p \pm round(n \times g_{norm}(p)) \tag{2}$$

The radial symmetry contribution $S_n$ at any radius $n$ is defined as the convolution of unsmoothed symmetry measure $F_n$ with a two-dimensional Gaussian filter $A_n$:

$$S_n = F_n \otimes A_n \tag{3}$$

The full transform $S$ is defined as the sum of the radial symmetry contributions over all the radii $N$ calculated:

$$S = \sum\nolimits_{n \in N} S_n \tag{4}$$

Once a symmetry map was computed, a threshold value was used to create a binary image from the symmetry map. The threshold values for each image in the training set were determined by a trial and error method. The average value of good thresholds for individual images in the training set was determined to be 9.1. As seen in Fig. 4b, bright points represent highly symmetrical regions. After binarization with thresholding, the resulting binary image, shown in Fig. 4c, yielded blobs representing potential fruits. After this stage, the remaining steps of method 3 were the same as for method 2 such as blob analysis and the scanning process using orthogonal neighbours with three scales. An example peach canopy image and the binary image after the RST are shown in Fig. 4.

These three image scanning methods are briefly summarized in Table 1.

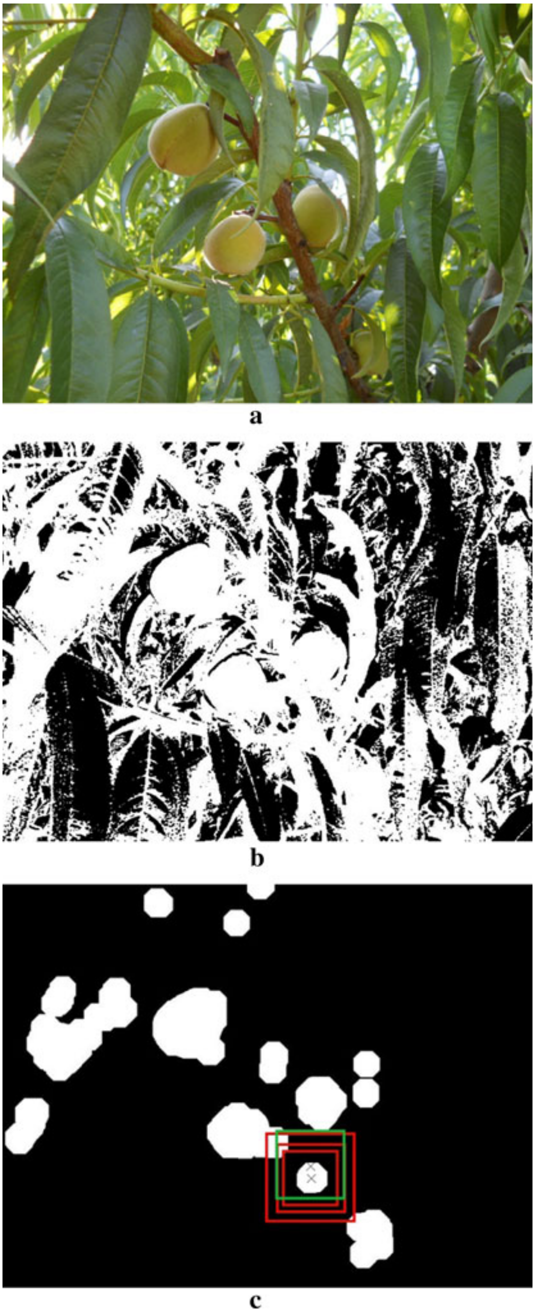


**Fig. 3** Example stages of method 2. **a** An example peach canopy image. **b** After binarization using histogram-based thresholding. **c** After morphological opening extracted blobs representing fruit candidates, a sub-window at three scales (*red squares*) and one of the 20-pixels-distance orthogonal neighbours (*green square*) over a candidate blob centre (Color figure online)

## Feature extraction

A PCA based 'eigenfruit' approach which was inspired by the 'eigenface' face detection and recognition method was introduced to detect green citrus fruits in natural canopies by

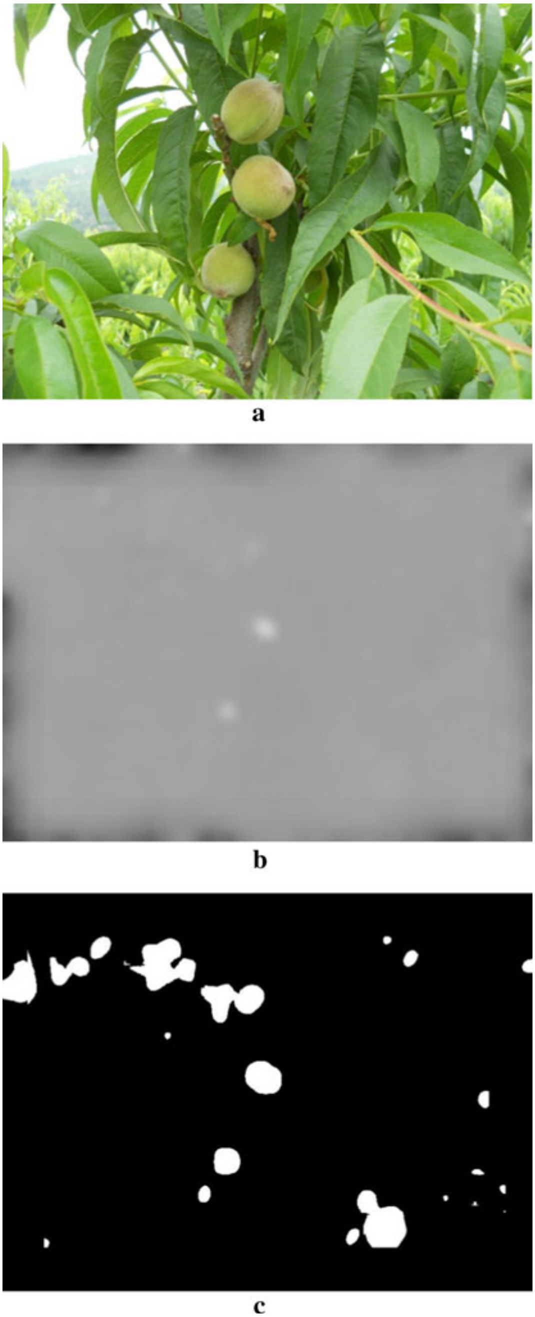


**Fig. 4** An example peach canopy image (**a**), symmetry map after the RST (**b**), and a binary image after thresholding (**c**)

Kurtulmus et al. (2011). A circular Gabor texture analysis was performed to extract texture features in the same study. Since the immature peach fruit detection task is also to identify circular objects in images, the same proven feature extraction methods were followed to detect immature peach fruits under natural illumination conditions with new image scanning methods and classifiers.

**Table 1** A summary of the image scanning methods used in this study

| Method 1 | Method 2 | Method 3 |
|---|---|---|
| Creates a binary image from colour information | Creates a binary image from colour information | Uses a radial symmetry transform of the grey level image |
| Uses a shifting sub-window to scan an entire image | Uses a morphological opening of the binary image to extract potential fruit regions | Uses symmetry map and thresholding to extract potential fruit regions |
| Uses a circular ROI and a pixel ratio to eliminate background regions | Extracts potential fruit regions based on blobs in the binary image and their neighbours | Extracts potential fruit regions based on blobs in the binary image and their neighbours |

The eigenface approach, as a face detection method, was proposed by Turk and Pentland (1991) and was first used as a feature extraction method for fruit detection in the previous study (Kurtulmus et al. 2011). It is derived from a principle component analysis. The eigenfruit approach provides robustness for partially occluded fruit detection. The feature extraction for the eigenfruit approach was made by calculating the distance from fruit space. More details about the eigenfruit feature extraction method can be found in Turk and Pentland (1991) and Kurtulmus et al. (2011). In this study, as two input features to the artificial classifiers, eigenfruit features of the intensity and saturation components were calculated separately in HSI colour space.

Texture information was also used as a feature in this study. Texture feature was extracted using circular Gabor texture analysis which was proposed by Zhang et al. (2002) by enhancing traditional 2-D Gabor filters. Feature extraction steps for this feature were conducted according to Zhang et al. (2002). Therefore, the following features were used in this study for peach fruit detection: eigenfruit features of intensity and saturation components and the circular Gabor texture feature.

## Data preparation

Training any classifier requires a training data set or feature vectors consisting of extracted features and class labels. To create a training data set, peach fruit images were manually centred and cropped from the training image set. In order to constitute a negative class corresponding to non-fruit objects, background frames including leaf, soil, twigs and sky pixels were also cropped from random regions of the training images. By calculating features of cropped samples and assigning class labels, a training data set of classifiers was prepared. Normalization was also applied to each feature to eliminate the drawback of significant differences between feature magnitudes. Each feature was normalized between −1 and 1 as described by:

$$X_N = \frac{2(X - X_{\min})}{X_{\max} - X_{\min}} \tag{5}$$

where $X_N$ is normalized value, $X$ is the value to be normalized, and $X_{min}$ and $X_{max}$ are minimum and maximum values of any feature, respectively.

## The classifiers

In this study, different types of classifiers were tested to find an effective classifier for immature peach fruit detection using extracted features. The following seven different classifiers were used including some parametric and non-parametric ones: discriminant analysis for classification, naive Bayes classifier, K-NN) classifier, classification trees classifier, regression trees classifier, support vector machine and neural networks. Using extracted features, the sub-windows from methods 1, 2 and 3 were classified by those classifiers.

*Discriminant analysis classifier*

Discriminant analysis is a parametric and supervised method that uses a Gaussian mixture model for data generation. Gupta et al. (2010) reported that the discriminant analysis is used in statistics, pattern recognition and machine learning applications to find a linear combination of features which characterizes or separates two or more classes of objects or events. As a classifier, discriminant analysis (DA) is widely used in classification problems. As reported by Guo et al. (2005) the traditional way of doing discriminant analysis is known as the linear discriminant analysis (LDA). In this study, a LDA was constructed as one of the seven classifiers.

*Naive Bayes classifier*

This type of classifier is a parametric method based on Bayes' theorem. It is widely used for classification tasks. By assuming Gaussian distribution of the data, the naive Bayes classifier (NB) estimates a separate normal distribution for each class by computing the mean and standard deviation of the training data using maximum likelihood estimation in that class. In this study, prior probabilities of the classes were determined using training samples from each class (fruit and background). In the images of the training set, peach fruit pixels were about 6 % of the total pixels. Therefore, the prior probability of the fruit class and the prior probability of the background class were set to 0.06 and 0.94, respectively. For predicting any invisible sample, the classifier computes the posterior probability of that sample belonging to each class. Then a classification process is carried out on the validation samples according to the largest posterior probability.

*K-NN classifier*

Another classifier used in the present study is the non-parametric classifier K-NN, where K represents the number of neighbours. Simply, this classification method categorizes query points based on their distance to points in a training dataset. To perform a K-NN decision rule, all distances between the given test pattern and all training data are measured. For assigning any test pattern to a class, the minimum distance from the pattern is measured. For measuring distances between the test pattern and the patterns in the training set, Mahalanobis distance was computed. In the preliminary works of this study, 1 and larger K values were tried to find the optimum number of neighbours (K), and 1-NN classifier yielded slightly better result. To classify sub-windows, a 1-NN classifier was constructed and performed.

*Classification and regression tree classifiers*

Classification and regression trees (CART) are non-parametric statistical methods. These classifiers build a decision tree describing a response variable as a function of different explanatory variables. Tree structured classifiers are constructed by repeated splits of subsets of training data into two descendant subsets. Each split is defined by a simple rule based on a single explanatory variable (Deconinck et al. 2012; Questier et al. 2005).

To set up CART classifiers, firstly all objects are assigned to the root node. Then each explanatory variable is split at all its possible split points. The parent node is split into two child nodes by separating the objects with values lower and higher than the split point for the considered explanatory variable. A measure of impurity of a node is required to help decide on how to split a node, or which node to split. A variable and split point are chosen with the highest reduction of impurity. For a regression tree (RT) classifier, impurity for a node with $c$ objects is described by:

$$\text{impurity} = \sum_{i=1}^{c} (y_i - \bar{y})^2 \tag{6}$$

where $y_i$ is a response value, $\bar{y}$ is mean of the node.

For a classification tree (CT) classifier, criteria for choosing a split can be one of the common methods which are Gini's diversity index, the towing index and the information index. In the present study, Gini's diversity index method was used. The index of a node with $c$ objects and $k$ possible classes was calculated by:

$$\text{Gini} = 1 - \sum_{j=1}^{k} \left(\frac{c_j}{c}\right)^2 \tag{7}$$

where $c_j$ is the number of objects from class $j$ in the node.

Once the split is chosen, the parent node is split into the two child nodes according to the selected split point. These steps are repeated until the tree has maximum size.

*Support vector machine classifier*

The support vector machine (SVM), based on the statistical learning theory, is a maximal margin classifier. As a non-parametric approach, SVMs do not attempt to model probability distribution of the training vectors; however, they try to separate the different classes by directly searching for adequate boundaries between them instead (Keuchel et al. 2003). To create a classifier, SVM uses an optimum separating hyperplane in the feature space between the classes with a maximal margin (Kecman 2001). In this work, for classifying background and fruit sub-windows, a SVM which uses a linear kernel was constructed using the training set containing two classes.

*Neural network classifier*

Another non-parametric approach is the artificial neural network (ANN) which is one of the most common machine learning and classification methods that imitate biological neurons. In this work, the ANN method was also used as one of the classifiers to discriminate peach fruit sub-windows. To create an ANN classifier, numerous types of networks can be created according to a number of hidden layers, a number of neurons or nodes in the layers, training algorithm, transfer or activation functions, performance functions, etc. For classifying sub-windows using extracted features in this work, various network architectures were tried. To increase the generalization ability of the ANN classifier and thereby avoid over-fitting, a small number of neurons and a small number of hidden layers were tried at the beginning. As a result, it was found that a two-hidden-layer feed-forward network with 15 neurons per layer performed significantly better than any other options. The network training function used in this study was a gradient descent with momentum back-propagation.

The neural network toolbox in Matlab was used to construct and train the network used. The output layer of the network had two neurons corresponding to fruit and background sub-windows. In the input layer, three neurons were used corresponding to extracted features which were eigenfruit features of intensity and saturation components and circular Gabor texture feature. A hyperbolic tangent sigmoid transfer function, a Log-sigmoid transfer function and a linear transfer function were used for two hidden layers and an output layer, respectively. The performance function was a mean square error (MSE). The desired error (goal) was 0.001, and the maximum epochs or iterations parameter was set to 10 000. The training process of the network would continue until it reached the maximum epochs or the MSE goal. Using these parameters and the training dataset, the ANN was trained. After the training process, a network file was stored to use in fruit detection.

### Merging multiple detections

In the proposed algorithms, the classifiers yielded multiple detections for the same fruit for most of the scanning methods used, and these multiple detections required merging to find an actual fruit count in the image. For this purpose, each detection was marked as a blob representing its detection centre on a binary image in the scanning process. Solid circles represented detection centres (blobs). Blob analysis was performed for merging multiple detections for the same fruit and counting the number of fruits. Multiple touching detections for the same fruit created one larger blob which consisted of detection centres. The major axis of this larger blob was used to determine the final detection centre of the fruit. The midpoint of the major axis represented the final detection centre. After blob analysis, connected blobs were treated as a single blob and the total number of blobs provided the number of fruits in the image. An example step is described later.

## Results

### Uneven illumination conditions and enhancement

Images were captured under different outdoor illumination conditions. It would be a difficult task to classify all images with the same processing steps by accommodating all different illumination conditions due to numerous variables in the outdoor natural environment. To illustrate algorithm performance under different illumination conditions, images were captured from both sunny and shadow sides of the peach canopy. Figure 5 shows some of these example images.

The illumination enhancement method which was used in this study expanded darker regions of the images. A resulting image of the logarithm transform and histogram equalization is shown in Fig. 6. Darker areas in the original image had better illumination, and hidden details revealed after illumination enhancement are seen in the figure.

### Colour-based image binarization

In this study, methods 1 and 2 used colour information via histogram-based colour analysis to provide potential fruit locations. After histogram-based colour analysis, some

a b

**Fig. 5** Images having different illumination conditions: **a** an image from sunny side of the canopy and **b** an image from shadow side of the canopy

a b

**Fig. 6** Effect of illumination enhancement method used in this study: **a** before and **b** after logarithm transform and histogram equalization

differences between fruit and background samples were found in H, Cb and Cr colour components. Figure 7 shows histogram distributions of the mosaic images prepared using the training set. Although the histogram curves were not highly distinctive, threshold values could be used to create binary images which were used in methods 1 and 2. The threshold values in H, Cb and Cr components were 52, 120 and 130, respectively. Example resulting binary images are shown in Figs. 2b and 3b.

## Merging multiple detections

In the experiments, detection centres of the classifiers were marked on a binary image. Figure 8 shows an example process for merging multiple detections of the classifiers. The final circles shown in Fig. 8c were drawn, based on detection centres. Although this method did not provide precise locations of the fruits due to asymmetric shapes of the blobs, it could successfully yield the number of fruits. The present study focused on finding number of fruits, but a more accurate fruit locations.

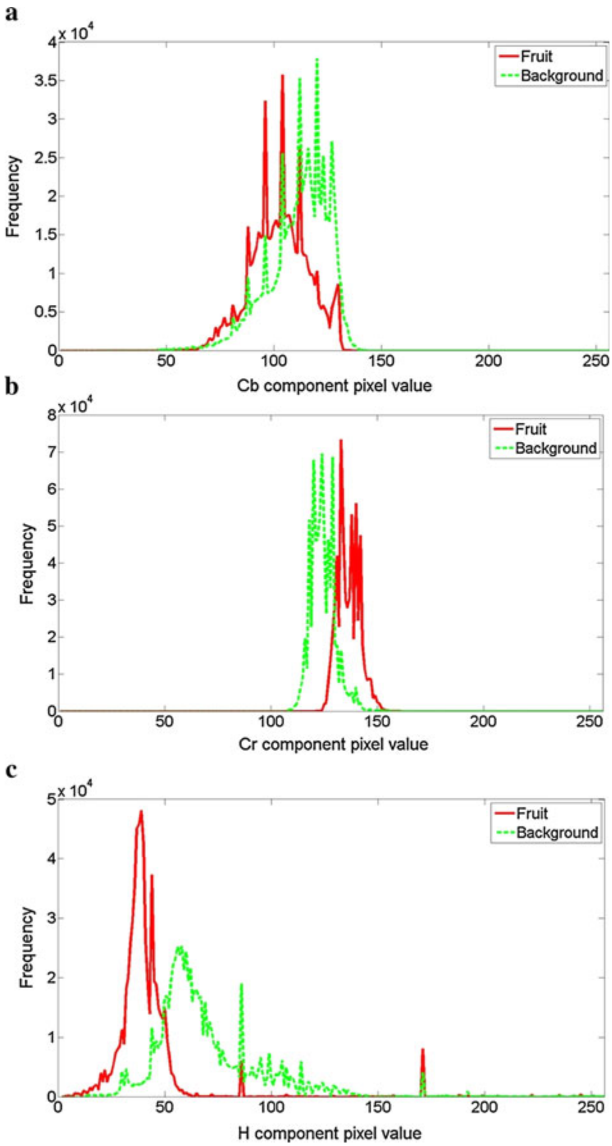


**Fig. 7** Histogram plots of Cb, Cr and H components for fruit and background classes

## Performances of the algorithms

Using the proposed image scanning methods and artificial classifiers, experiments were carried out for the training and validation image sets. A complete example procedure including different algorithm steps is shown in Fig. 9 for method 2 using the ANN classifier. The proposed algorithms provided the number of peach fruits in the canopy images. Performance of the algorithms were measured according to the number of correct detections, false positives, and missing fruit. Some objects were not clearly identified as either fruit or background due to much occlusion or visual complexity of the tree canopy, and

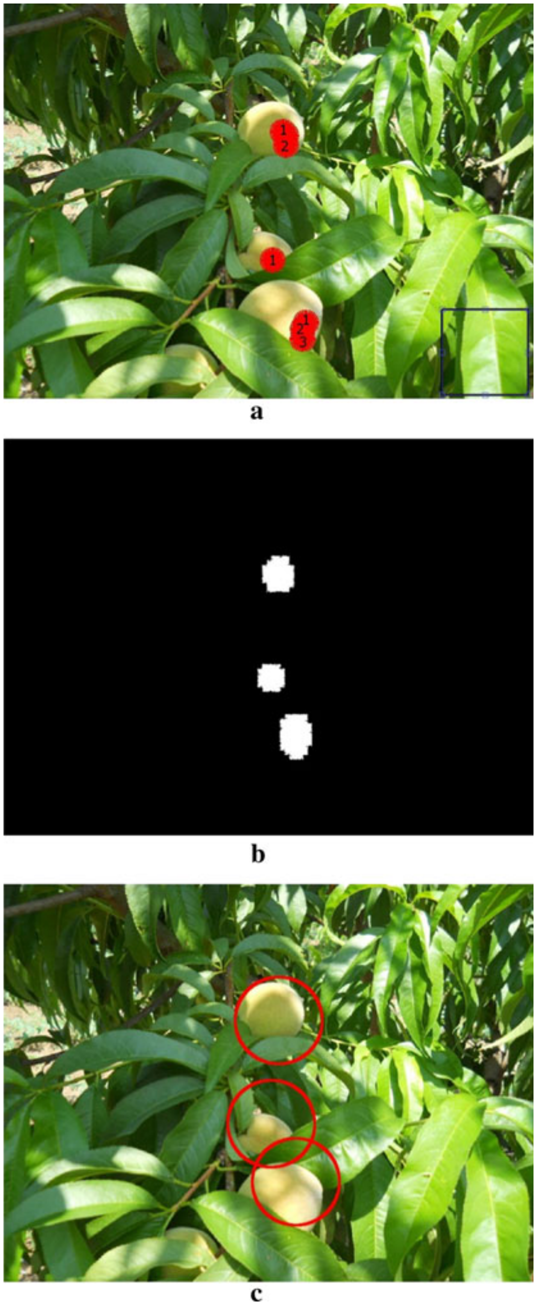


**Fig. 8** Steps of merging multiple detections: **a** multiple detections, **b** binary image representing detection centres and **c** resulting image

they were considered as background. In the present study, processing time was not considered, since the research focus was exploring the feasibility of the detection performance. Nevertheless, processing times of proposed image scanning methods were measured to make a comparison between different proposed methods. Figure 10 shows correctly

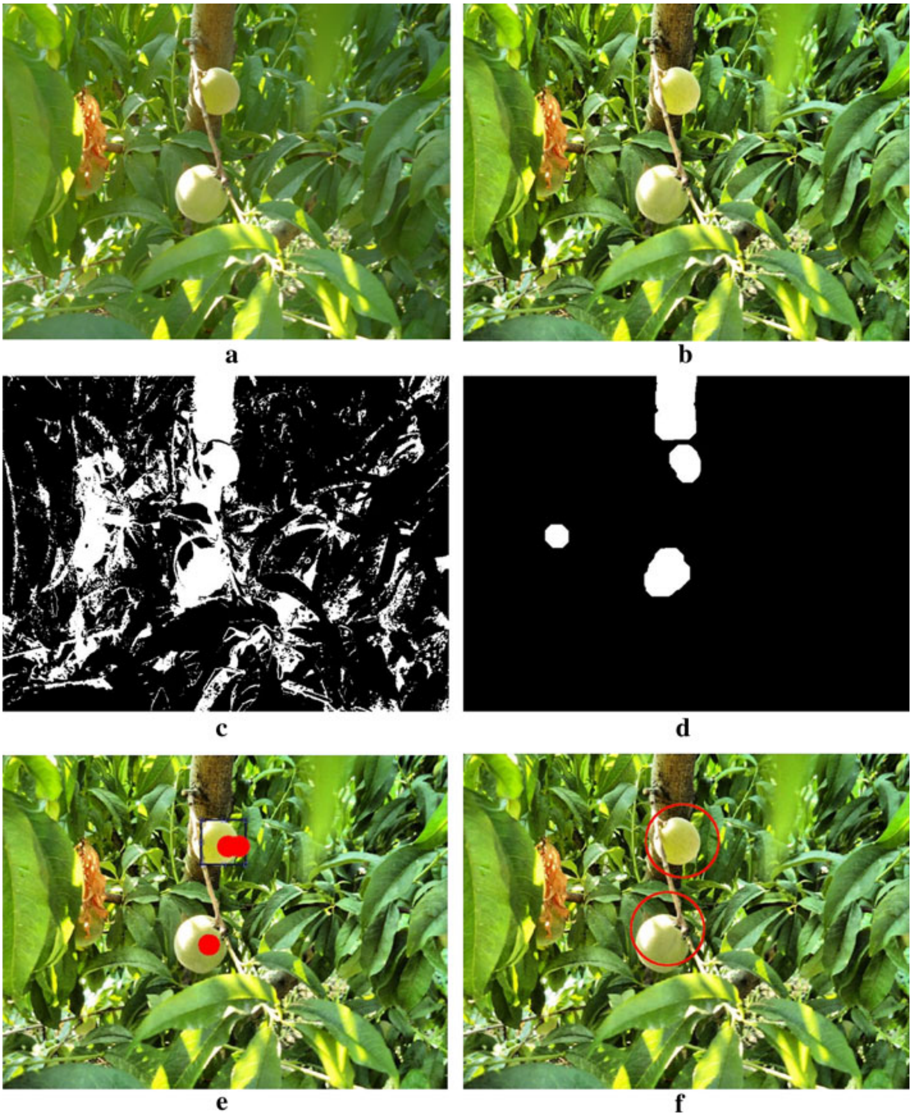


**Fig. 9** An example complete procedure including algorithm steps for method 2 using the ANN classifier. **a** An example peach canopy image. **b** Illumination enhancement. **c** Histogram based binarization. **d** Fruit canditates after the morphological opening. **e** Detection results by the ANN classifier. **f** Final detections after the blob analysis.

identified fruit rates of the validation set for method 1, 2 and 3; more details can be found in the following section.

*Success rates of the classifiers using method 1*

Classification results of the classifiers are presented in Table 2 for method 1. For the validation set, the DA classifier, the ANN classifier and the SVM classifier yielded the highest detection performance. Using method 1, about 85 % of the actual number of the

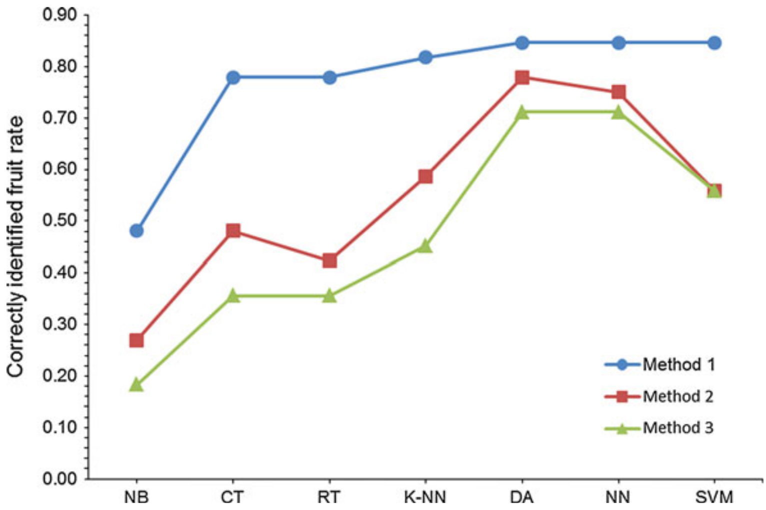


**Fig. 10** Correctly identified fruit rates for the validation set

fruits in the validation set were successfully detected. The false detections of these classifiers were about 27.8, 22.8 and 26.7 %, respectively (the percentage of false detections is provided with regard to the total number of correct detection plus the false positives). The missed fruits rate by those three classifiers was about 15 %. Success rates for the K-NN classifier accounted for the second highest detection performance which were about 82, 22 and 18 % for true positives, false detections and missed, respectively. The worst success rates for method 1 were for the naïve Bayes classifier. The CART classifiers had almost the same detection rates and could detect about 78 % of the fruits with about 26 % false detections and 22 % missed. While false positive rates were close to each other for the classifiers, the K-NN classifier and the ANN classifier provided the least false positive rates with about 23 %. Using method 1, the processing time including decision of the classifiers varied in the range between 72.8 and 112 s for individual images in the validation set.

Performance of the classifiers was also evaluated according to different illumination conditions. For the validation set, Table 3 shows detection performance of method 1 with respect to the images taken from the shadow and sunny sides of the canopy. Except for the naive Bayes classifier, all classifiers yielded slightly better performance for the shadow side of the canopy. The highest success rate (91 %) was achieved by the SVM classifier for the images taken in shadow. False detection rates varied between 22 and 30 % for method 1.

*Success rates of method 2*

Success rates of the classifiers by method 2 are shown in Table 4. Method 2 used a set of image processing operations to narrow down the fruit detection area in the image and thereby reduce false detections. As seen in Table 4, false positives of the classifiers were reduced to between 3 and 10 % for the validation set compared to method 1. However, the accuracies for correctly identified fruits were lower than for method 1. The DA classifier accounted for the best detection performance with a correctly identified rate of 78 %. The Naïve Bayes classifier yielded dramatically poor detection performance. Successful detection rates of the CART classifiers were 48 and 42 %, respectively. The second-best

**Table 2** Results of the classifiers using method 1 with respect to detection performance

| Method 1 | Training set (fruit count: 58) | | | Validation set (fruit count: 104) | | |
|---|---|---|---|---|---|---|
| | Correctly identified Fruit count (%) | False positives[a] Fruit count (%) | Missed Fruit count (%) | Correctly identified Fruit count (%) | False positives[a] Fruit count (%) | Missed Fruit count (%) |
| Discriminant analysis | 56 (0.97) | 14 (0.20) | 2 (0.03) | 88 (0.85) | 34 (0.28) | 16 (0.15) |
| K-nearest neighbour | 51 (0.88) | 6 (0.11) | 7 (0.12) | 85 (0.82) | 24 (0.22) | 19 (0.18) |
| Classification tree | 51 (0.88) | 6 (0.11) | 7 (0.12) | 81 (0.78) | 29 (0.26) | 23 (0.22) |
| Naïve Bayes | 40 (0.69) | 1 (0.02) | 18 (0.31) | 50 (0.48) | 17 (0.25) | 54 (0.52) |
| Regression tree | 51 (0.88) | 6 (0.11) | 7 (0.12) | 81 (0.78) | 28 (0.26) | 23 (0.22) |
| Neural networks | 52 (0.90) | 10 (0.16) | 6 (0.10) | 88 (0.85) | 26 (0.23) | 16 (0.15) |
| Support vector machine | 53 (0.91) | 7 (0.12) | 5 (0.09) | 88 (0.85) | 32 (0.27) | 16 (0.15) |

[a] The percentage of false detections is provided with regard to the total number of correct detection plus the false positives: false positive (%) = false positives/(correct detections + false positives) × 100

**Table 3** Results of the classifiers using method 1 with respect to the images taken from shadow side and sunny side of the canopy

| Method 1 (validation set) | Shadow side (fruit count: 45) | | | Sunny side (fruit count: 59) | | |
|---|---|---|---|---|---|---|
| | Correctly identified Fruit count (%) | False positives[a] Fruit count (%) | Missed Fruit count (%) | Correctly identified Fruit count (%) | False positives[a] Fruit count (%) | Missed Fruit count (%) |
| Discriminant analysis | 39 (0.87) | 17 (0.30) | 6 (0.13) | 49 (0.83) | 17 (0.26) | 10 (0.17) |
| K-nearest neighbour | 38 (0.84) | 12 (0.24) | 7 (0.16) | 47 (0.80) | 12 (0.20) | 12 (0.20) |
| Classification tree | 38 (0.84) | 11 (0.22) | 7 (0.16) | 43 (0.73) | 18 (0.30) | 16 (0.27) |
| Naïve Bayes | 17 (0.38) | 4 (0.19) | 28 (0.62) | 33 (0.56) | 13 (0.28) | 26 (0.44) |
| Regression tree | 38 (0.84) | 11 (0.22) | 7 (0.16) | 43 (0.73) | 17 (0.28) | 16 (0.27) |
| Neural networks | 39 (0.87) | 11 (0.22) | 6 (0.13) | 49 (0.83) | 15 (0.23) | 10 (0.17) |
| Support vector machine | 41 (0.91) | 15 (0.27) | 4 (0.09) | 47 (0.80) | 17 (0.27) | 12 (0.20) |

[a] The percentage of false detections is provided with regard to the total number of correct detection plus the false positives: false positive (%) = false positives/(correct detections + false positives) × 100

detection performance was observed with the ANN classifier for method 2. The processing time for method 2 including decision times of the classifiers varied in the range between 5.3 and 21 s for images in the validation set.

Similar to method 1, classifiers performed better for the shadow side. The DA and the ANN classifiers accounted for the best performances for both the shadow and the sunny side (Table 5). While those classifiers successfully detected 84 % of the fruits in shadow,

**Table 4** Results of the classifiers using method 2 with respect to detection performance

| Method 2 | Training set (fruit count: 58) | | | Validation set (fruit count: 104) | | |
|---|---|---|---|---|---|---|
| | Correctly identified Fruit count (%) | False positives[a] Fruit count (%) | Missed Fruit count (%) | Correctly identifie Fruit count (%) | False positives[a] Fruit count (%) | Missed Fruit count (%) |
| Discriminant analysis | 54 (0.93) | 1 (0.02) | 4 (0.07) | 81 (0.78) | 6 (0.07) | 23 (0.22) |
| K-nearest neighbour | 42 (0.72) | 1 (0.02) | 16 (0.28) | 61 (0.59) | 5 (0.08) | 43 (0.41) |
| Classification tree | 40 (0.69) | 1 (0.02) | 18 (0.31) | 50 (0.48) | 4 (0.07) | 54 (0.52) |
| Naïve Bayes | 33 (0.57) | 0 (0.00) | 25 (0.43) | 28 (0.27) | 1 (0.03) | 76 (0.73) |
| Regression tree | 40 (0.69) | 1 (0.02) | 18 (0.31) | 44 (0.42) | 5 (0.10) | 60 (0.58) |
| Neural networks | 52 (0.90) | 1 (0.02) | 6 (0.10) | 78 (0.75) | 8 (0.09) | 26 (0.25) |
| Support vector machine | 48 (0.83) | 1 (0.02) | 10 (0.17) | 58 (0.56) | 5 (0.08) | 46 (0.44) |

[a] The percentage of false detections is provided with regard to the total number of correct detection plus the false positives: false positive (%) = false positives/(correct detections + false positives) $\times$ 100

their success rates for the sunny side were about 73 and 68 %, respectively. The SVM and the K-NN classifiers yielded the second-best success rates for the validation set.

*Success rates of method 3*

In method 3, the radial symmetry transform was applied to extract potential fruit locations by assuming ideal circular shapes of the fruits without any occlusion. The best success rates (71 %) were obtained by the DA and the ANN classifiers for the validation set (Table 6). The false positives of those classifiers were about 33 and 41 %, respectively. The SVM classifier could detect about 56 % of the fruits in the validation set. Quite poor success rates were obtained for the rest of the classifiers for method 3. By using this method, processing time varied in the range between 11 and 31.3 s for images in the validation set.

Table 7 shows success results for method 3 with respect to the shadow and sunny sides. Considering performance of method 3 with respect to shadow and sunny sides, maximum performance was for the DA and the ANN classifiers for both sides. While success rates of those classifiers were 80 and 82 %, false positive rates were high. The SVM classifier performed at the success rate of 62 % for the shadow side with a false positive rate of 39 %.

## Discussion

In these immature peach fruit detection experiments, algorithms could not detect all fruits in the image sets. Unwanted false detections also occurred. One major reason for mis-classifications was colour similarity between fruits and background objects. In the natural canopy images, colours of some leaves were very similar to fruit colour. However, this was not the only colour-based obstacle to the proposed fruit detection algorithms as some soil and twig objects also had similar colours. Therefore, colour-based mis-classifications

**Table 5** Results of the classifiers using method 2 with respect to the shadow side and the sunny side of the canopy

| Method 2 (validation set) | Shadow side (fruit count: 45) | | | Sunny side (fruit count: 59) | | |
|---|---|---|---|---|---|---|
| | Correctly identified Fruit count (%) | False positives[a] Fruit count (%) | Missed Fruit count (%) | Correctly identified Fruit count (%) | False positives[a] Fruit count (%) | Missed Fruit count (%) |
| Discriminant analysis | 38 (0.84) | 2 (0.05) | 7 (0.16) | 43 (0.73) | 4 (0.09) | 16 (0.27) |
| K-nearest neighbour | 29 (0.64) | 2 (0.07) | 16 (0.36) | 32 (0.54) | 3 (0.09) | 27 (0.46) |
| Classification tree | 23 (0.51) | 1 (0.04) | 22 (0.49) | 27 (0.46) | 3 (0.10) | 32 (0.54) |
| Naïve Bayes | 11 (0.24) | 0 (0.00) | 34 (0.76) | 17 (0.29) | 1 (0.06) | 42 (0.71) |
| Regression tree | 22 (0.49) | 2 (0.08) | 23 (0.51) | 22 (0.37) | 3 (0.12) | 37 (0.63) |
| Neural networks | 38 (0.84) | 3 (0.07) | 7 (0.16) | 40 (0.68) | 5 (0.11) | 19 (0.32) |
| Support vector machine | 30 (0.67) | 2 (0.06) | 15 (0.33) | 28 (0.48) | 3 (0.10) | 31 (0.53) |

[a] The percentage of false detections is provided with regard to the total number of correct detection plus the false positives: false positive (%) = false positives/(correct detections + false positives) × 100

**Table 6** Results of the classifiers using method 3 with respect to detection performance

| Method 3 | Training set (fruit count: 58) | | | Validation set (fruit count: 104) | | |
|---|---|---|---|---|---|---|
| | Correctly identified Fruit count (%) | False positives[a] Fruit count (%) | Missed Fruit count (%) | Correctly identified Fruit count (%) | False positives[a] Fruit count(%) | Missed Fruit count (%) |
| Discriminant analysis | 45 (0.78) | 7 (0.14) | 13 (0.22) | 74 (0.71) | 36 (0.33) | 30 (0.29) |
| K-nearest neighbour | 34 (0.59) | 5 (0.13) | 24 (0.41) | 47 (0.45) | 14 (0.23) | 57 (0.55) |
| Classification tree | 33 (0.57) | 5 (0.13) | 25 (0.43) | 37 (0.36) | 16 (0.30) | 67 (0.64) |
| Naïve Bayes | 27 (0.47) | 1 (0.04) | 31 (0.53) | 19 (0.18) | 4 (0.17) | 85 (0.82) |
| Regression tree | 33 (0.57) | 6 (0.15) | 25 (0.43) | 37 (0.36) | 14 (0.28) | 67 (0.64) |
| Neural networks | 45 (0.78) | 9 (0.17) | 13 (0.22) | 74 (0.71) | 52 (0.41) | 30 (0.29) |
| Support vector machine | 35 (0.60) | 7 (0.17) | 23 (0.40) | 58 (0.56) | 26 (0.31) | 46 (0.44) |

[a] The percentage of false detections is provided with regard to the total number of correct detection plus the false positives: false positive (%) = false positives/(correct detections + false positives) × 100

occurred when using the image scanning methods which used colour information to extract possible fruit locations or to eliminate background regions.

Uneven illumination was another factor in reducing success rates of the classifiers. Especially, partially illuminated fruits caused disruption of shape and texture information despite illumination enhancement. However, the results showed that the algorithms (for all three methods) could perform better for the images captured from the shadow side of the canopy where illumination was diffuse and more homogeneous.

**Table 7** Results of the classifiers using method 3 with respect to the shadow side and the sunny side of the canopy

| Method 3 (validation set) | Shadow side (fruit count: 45) | | | Sunny side (fruit count: 59) | | |
|---|---|---|---|---|---|---|
| | Correctly identified Fruit count (%) | False positives[a] Fruit count (%) | Missed Fruit count (%) | Correctly identified Fruit count (%) | False positives[a] Fruit count (%) | Missed Fruit count (%) |
| Discriminant analysis | 36 (0.80) | 21 (0.37) | 9 (0.20) | 38 (0.64) | 15 (0.28) | 21 (0.36) |
| K-nearest neighbour | 19 (0.42) | 4 (0.17) | 26 (0.58) | 28 (0.48) | 10 (0.26) | 31 (0.53) |
| Classification tree | 15 (0.33) | 5 (0.25) | 30 (0.67) | 22 (0.37) | 11 (0.33) | 37 (0.63) |
| Naïve Bayes | 4 (0.09) | 0 (0.00) | 41 (0.91) | 15 (0.25) | 4 (0.21) | 44 (0.75) |
| Regression tree | 14 (0.31) | 5 (0.26) | 31 (0.69) | 23 (0.39) | 9 (0.28) | 36 (0.61) |
| Neural networks | 37 (0.82) | 39 (0.51) | 8 (0.18) | 37 (0.63) | 13 (0.26) | 22 (0.37) |
| Support vector machine | 28 (0.62) | 18 (0.39) | 17 (0.38) | 30 (0.51) | 8 (0.21) | 29 (0.49) |

[a] The percentage of false detections is provided with regard to the total number of correct detection plus the false positives: false positive (%) = false positives/(correct detections + false positives) × 100

Occlusion was also a challenge in successfully detecting immature peach fruits in the natural canopies. In the experiments, some peach fruits were highly-overlapped by leaves and other fruits, and could not be successfully detected by the algorithms. Occlusion also created a situation similar to partially illuminated fruits. Some fruit surfaces in the images were divided into multiple regions by the leaves. In particular, methods 2 and 3 relied on potential detection centres which were affected by these negative conditions more than with method 1. Divided fruit surfaces created eccentric potential centres, and so corresponding fruits could not be combined thoroughly by the sub-window with any classifiers. The shape information used by method 3 was also affected by some divided surfaces from both illumination-based and occlusion-based disruptions. Nevertheless, the proposed algorithms were able to detect some of those occluded fruits.

Method 2 provided less false detection rates at the expense of some true positive rate (correctly identified) in comparison to method 1. Classifiers such as the DA and the ANN yielded false detection rates of 7 and 9 % for the validation set by using method 2, respectively. On the other hand, method 3 did not produce low false detection rates due to the fact that some leaves and leaf clusters were of similar shapes to peach fruits.

Considering overall performance of the classifiers, parametricity was not a significant factor in detection performance with respect to classifier type. Eventually, the ANN classifier which is non-parametric and the DA classifier which is parametric provided successful detections in the experiments. While the SVM and the K-NN classifiers which are non-parametric yielded good success rates when using method 1, their detection accuracies were poor when using methods 2 and 3. In any classification task, performance of the classifier mostly depends on its ability to reveal hidden relationships between the features. The features used in this study represented different information such as shape and texture. The naive Bayes classifier yielded the lowest detection performance for all the methods used in the experiments by using these features. It is known that this kind of classifier is usually less accurate than other supervised methods due to its inadequate ability to deal with complex interactions between features, and it does not guarantee classification performance. In this sense, the ANN and the DA classifiers could perform better with extracted features. The performance of the classifiers showed their capabilities of discovering interactions between features.

## Conclusions

Using colour images captured from a natural canopy, computer vision algorithms were developed to detect and count immature peach fruits under natural illumination conditions for estimating immature peach yield to help growers create yield prediction maps. Colour, shape, and texture information were used by extracting the features from peach canopy images and constituting different image scanning methods. Classifiers with extracted features were utilized to classify potential sub-windows for detection of peach fruits. Multiple detections of the classifiers for same fruits were merged via blob analysis. Using three different image scanning methods, experiments were carried out and the results were evaluated.

The detection of immature peaches under outdoor conditions is subject to specific obstacles such as uneven illumination and occlusion of the canopy objects. Nevertheless, the proposed algorithms could successfully detect about 85 % of the immature peach fruits in images of the validation set. The ANN and the DA classifiers exhibited the best detection accuracies for extracted features. Despite lower detection accuracies, the image scanning approaches relying on potential fruit locations reduced false positive rates and promised immature peach detection using just colour images.

This study was the first effort to detect immature peach fruits in natural environments to the authors' knowledge. The feasibility of detecting immature peach fruits for yield prediction using just regular colour images was explored. Developing an early peach yield prediction system with cameras installed in a mobile platform and continuous image acquisition is the final aim for this study.

## References


Bulanon, D. M., Burks, T. F., & Alchanatis, V. (2010). A multispectral imaging analysis for enhancing citrus fruit detection. *Environment Control in Biology, 48*(2), 82–91.

Deconinck, E., Sacréa, P. Y., Coomans, D., & De Beer, J. (2012). Classification trees based on infrared spectroscopic data to discriminate between genuine and counterfeit medicines. *Journal of Pharmaceutical and Biomedical Analysis, 57*, 68–75.

FAO 2010. Food and agriculture organization of the United Nations http://faostat.fao.org/site/567/DesktopDefault.aspx?PageID=567#ancor. Accessed 12 Dec 2012.

Guo, Y., Hastie, T., & Tibshirani, R. (2005). Regularized discriminant analysis and its application in microarrays. *Biostatistics, 1*(1), 1–18.

Gupta, V., Singh, G., Mittal, M., & Pahuja, S. K. (2010). Fourier transform of untransformable signals using pattern recognition technique. In *Proceedings of the second international conference on advances in computing, control, and telecommunication technologies (ACT'10)*. IEEE Computer Society, Washington, DC, USA, pp. 6–9.

Jimenez, A. R., Ceres, R., & Pons, J. L. (2000). A survey of computer vision methods for locating fruit on trees. *Transactions of the ASAE, 43*, 1911–1920.

Kecman, V. (2001). *Learning and soft computing*. Cambridge: MIT Press.

Keuchel, J., Naumann, S., Heiler, M., & Siegmund, A. (2003). Automatic land cover analysis for Tenerife by supervised classification using remotely sensed data. *Remote Sensing of Environment, 86*, 530–541.

Kurtulmuş, F., Lee, W. S., & Vardar, A. (2011). Green citrus detection using 'eigenfruit', color and circular Gabor texture features under natural outdoor conditions. *Computers and Electronics in Agriculture, 78*(2), 140–149.

Loy, G., & Zelinsky, A. (2003). Fast radial symmetry for detecting points of interest. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 25*, 8.

Okamoto, H., & Lee, W. S. (2009). Green citrus detection using hyperspectral imaging. *Computers and Electronics in Agriculture, 66*, 201–208.

Parrish, E. A, Jr, & Goksel, A. K. (1977). Pictorial pattern recognition applied to harvesting. *Transactions of the ASAE, 20*, 822–827.

Pla, F., Juste, F., & Ferri, F. (1993). Feature extraction of spherical objects in image analysis: an application to robotic citrus harvesting. *Computers and Electronics in Agriculture, 8*, 57–72.

Questier, F., Put, R., Coomans, D., Walczak, B., & Heyden, Y. V. (2005). The use of CART and multivariate regression trees for supervised and unsupervised feature selection. *Chemometrics and Intelligent Laboratory Systems, 76*, 45–54.

Stajnko, D., Lakota, M., & Hoevar, M. (2004). Estimation of number and diameter of apple fruits in an orchard during the growing season by thermal imaging. *Computers and Electronics in Agriculture, 42*, 31–42.

Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience, 3*, 71–86.

Wachs, J., Stern, H. I., Burks, T., & Alchanatis, V. (2009). Apple detection in natural tree canopies from multimodal images. In *Proceedings of the joint international agricultural conference*, JIAC, Wageningen Academic |Publishers, The Netherlands, pp. 293–302.

Xiang, D., Tian, J., Deng, K., Zhang, X., Yang, F., & Wan, X. (2011). Retinal vessel extraction by combining radial symmetry transform and iterated graph cuts. In *Proceedings of IEEE Engineering in Medicine and Biology Society 2011, Boston*, Massachusetts, USA, pp. 3950–3953.

Zhang, J., Tan, T., & Ma, L. (2002). Invariant texture segmentation via circular Gabor filters. In *Proceedings of the 16th international conference on pattern recognition*. IEEE Computer Society, Quebec City, Quebec, Canada, 2002, (2), pp. 901–904.

Zuiderveld, K. (1994). Contrast limited adaptive histogram equalization. *Graphics Gems IV*, 474–485.