Mobile device machine vision estimation of mango crop load

Anand Koirala*[1], Kerry B. Walsh[1], Zhenglin Wang[1], Cheryl McCarthy[2]
1 Central Queensland University, Rockhampton, Australia
2 University of Southern Queensland, Toowoomba, Australia

Abstract

The application of machine vision in orchard was considered in context of mango crop load (fruit number and fruit size). An algorithm for automatic detection and counting of fruits in images of trees in orchard was developed. RGB images were acquired of two sides of mango trees ('dual view'). Fruit count per tree was obtained by harvest of trees, and by manual count of fruit in images. The R^2 and slope between dual-view and harvest count varied between 0.74 and 0.92, and 0.34 and 0.55, respectively, depending on canopy structure. The fruit counting model involved: (i) fruit-like object detection using HAAR cascade classifier using an AdaBoost technique; (ii) classification of detected region using a multilayer Convolutional Neural Network (CNN). The machine vision count achieved a precision = 0.94, recall= 0.89, and F1 score = 0.9 against a human count of fruit in images. For the estimation of fruit size individual fruits were imaged against a backing board (with a circular scale printed on a blue background), with an RMSE of 3.6 mm for lineal dimension measurement achieved.

Background

Knowledge of fruit size and counts can help to make decisions on agronomic treatments, resource management and market planning. Estimation of crop load and quantity is typically based on previous yield history, and manual fruit count and size measuring of a sample of orchard fruit. Manual in-field counting of fruits is time consuming and inconsistent. Similarly, manual measurement of fruit dimensions using callipers is slow. In practice, few farm managers make these measurements consistently, due to the labour requirement.

Machine vision can be applied to assessment of fruit number and size on tree, given consideration of:

1. variation in apparent size of fruit vary within canopy with distance from camera
    - variation in lighting conditions within the canopy
    - variation in colour of fruits and foliage
    - occlusion of fruit by other fruits, branches and foliage

Computer vision and machine learning has been applied to automatic fruit detection and sizing (Sethy *et al.*). High detection rates have been achieved through the combination of various sensor technologies and machine vision techniques. In particular, convolutional neural networks (CNN) have been successfully implemented in many image classification challenges in recent years.

However, the cost of the equipment, the long processing times, and complexity of use are shortcomings. A mobile device running a machine vision application could be used to capture images of a representative number of trees and fruits for estimation of fruit number and lineal dimensions, speeding the current human based sampling protocol. A mobile device based method is also suitable for small farms. A mobile device can also store images, record location and allow for connection to farm networked devices and upload/download data to servers or cloud. However, low processing load is required. The paper reports on the development of mobile applications for crop load estimation through sizing and counting of fruits in digital images.

Methods

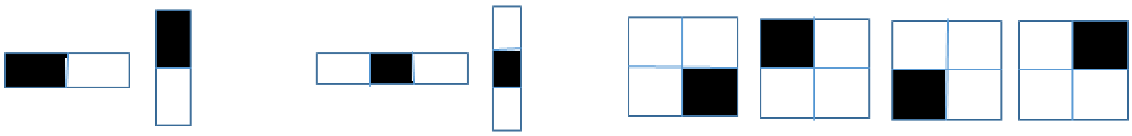
Equipment
Images were acquired of mango tree canopies using a Canon DSLR 750D camera (2448x2048 pixels).

Estimating tree crop load from dual view images
Images were captured of both sides of 18 trees in each of three mango orchards. Manual counts were made of fruit visible in images. The trees were strip harvested and fruit counted.

Training image set
20 RGB colour images of trees from a row were randomly selected for training. Fruits and backgrounds (leaves, trunks, branches, and sky) were annotated using the OpenCV annotation tool. This included fruits of various levels of occlusion. Annotated snips (n=2000) of each class (fruit and background) were cropped from the training image set. These snips were used to train both stages (cascade classifier and neural network).

Detection
The open- source computer vision and machine vision library OpenCV (OpenCV, 2017) was used for training and testing of the cascade classifier.

HAAR (Viola *et al.*, 2001) like features were extracted into a pool of features from all the training images and a boosting algorithm AdaBoost (Papageorgiou *et al.*, 1998) used to build a cascade classifier object detection model from the weak separate classifiers. In this experiment three basic HAAR like features (Figure 1) were used. Each feature is a value obtained by subtracting sum of pixels in black region from sum of pixels in white region.



(a) Edge feature (b) Line feature (c) Radian feature

Figure 1. Basic HAAR like features

Multiscale detection from OpenCV was used to detect fruit of all possible sizes in the image. The minimum object size that this model can detect is the height and width parameters provided during the training. The smallest fruit size in the input images (2448x2048 pixels) was 28x28 pixels. A scale factor of value 1.1 was used, meaning the full image is reduced by 10 % in each step in multiscale detection from a single scale model. There can be many detection for the same object at different scales therefore the parameter 'minNeighbors' was set to 4, i.e. at least 4 detection windows must exist in order to consider the object as fruit. A Region of Interest (ROI) was generated for all objects being detected as fruit by the cascade classifier. These regions also contained many false positives such as leaves and branches. A validation process involving a convolutional neural network was used to remove the false positives.

Classification stage
The CNN model used in this paper follows LeNet (LeCun *et al.*, 1998) architecture with some modifications in the parameters. This model features a series of convolutional layers followed by max-pooling layers. Six layers were used (Fig. 2). The CNN model was implemented and trained in DL4J (Gibson *et al.*, 2016), an open source, distributed, deep learning library. The model was trained with 2000 snips each of fruits and background cropped from full canopy images. All three channels (RGB) were used as input to the neural network.

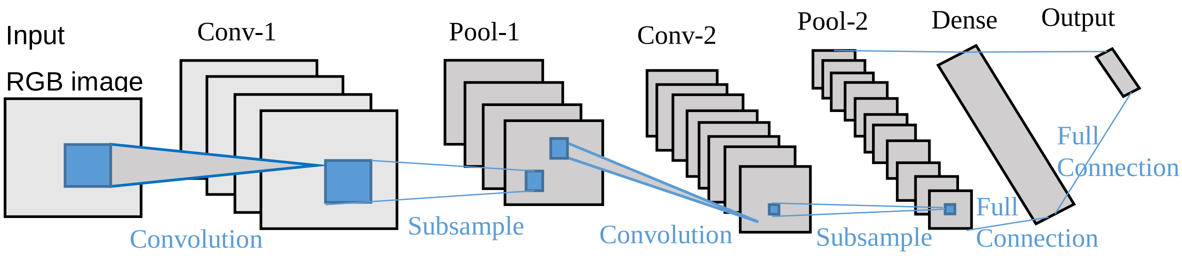


Figure 2. The LeNet architecture

The ROIs generated from the cascade classifier were resized to 28x28 pixels and fed as inputs to the CNN model. An activation function 'ReLU' was used for convolution and dense layers. The model passes the input through a series of convolution and subsampling (max-pool) layers and finally a probability score for each class (fruit/background) is generated using a logistic regression function 'softmax'. An optimum threshold value of 0.79 on the score was selected based on the precision-recall curve. Therefore any ROI with score > 0.79 was considered to be a fruit and included in the count.

Fruit sizing

RGB image captured from a mobile phone were converted to CIE L*a*b* colour space and the b* channel extracted. The colour of background was chosen blue and that of marker yellow because these two colours are opposing colours in b* channel. The fruit and scale was segmented from background using Otsu's (Otsu, 1979) thresholding algorithm. Morphological operations were performed on the resultant binary image to segment the fruit stalk, an unavoidable feature for in-field images. An upright rectangular bounding box was drawn around the fruit perimeter in the image and pixel dimensions for length and width of the fruit calculated. The ratio of diameter (pixel) of marker to the known physical diameter (4 cm) of circular marker was used as a scale to allow estimation of the lineal dimension of fruit from the images.

Results

Estimating tree crop load from dual view images

Though all the fruits on a tree are not visible in the dual view images, the correlation of dual view image count to harvest counts exceed 0.74 in all cases.

Table 1. Correlation between dual view image count and in-field harvest count for several orchards

		Dual view image count vs Harvest Count	
farm	Number of trees	R^2	Slope
A	18	0.79	0.34
B	18	0.74	0.35
C	18	0.77	0.41
D	18	0.84	0.55
E	18	0.92	0.46

Fruit detection and count

In-field tree canopy images contain a varying number of fruits as well as much background (leaves, branches, sky, etc.).The cascade structure of the classifier was successful in discarding most of negative samples in a few early stages, based on the evaluation of a small set of features (Fig. 3). A precision = 0.94, recall =0.89, and F1 score = 0.9 assessed against human image count was achieved with the combination of the HAAR classifier and the CNN.

Figure 3. Fruit detection in the images

Fruit sizing
Imaging fruit against the background and scale (Fig. 4), a R^2 = 0.95 and RMSE = 3.6 mm was achieved (n = 40 fruit).

Figure 4. In-field imaging (left) and the result of fruit sizing mobile app (right)

Discussion
Estimating tree crop load from dual view images
The slope of the correlation between the dual view and the actual tree count varied with orchard, with denser canopies having more occluded, non-visible fruit. This was also true between trees within an orchard, with more variation accounting for a decreased R^2 between dual view and the actual tree count (data not shown). Future work will consider measures of foliage density as indices of the proportion of occluded fruit, to be used in fruit count correct per tree.

Fruit detection and count

In-field tree canopy images contain a varying number of fruits as well as much background (leaves, branches, sky, etc.).The cascade structure of the classifier was successful in discarding most of negative samples in a few early stages, based on the evaluation of a small set of features. The CNN classifier stage was needed to further reduce false positives, but it was applied to only a subset of the original image. Thus the time for detection was greatly reduced. For an image having an average of 103 detections, the algorithm running on a laptop (CPU 2.4 GHz, RAM 8 GB, 64 bit Windows OS) took an average of 2.2 seconds for detection and 1.6 seconds for classification of all objects detected. A couple of seconds delay in processing the image is all right for the sampling approach as the user is expected to spend some time while moving through the orchard to acquire the next sample.

The fruits occluded by other fruit in clusters were sometimes not detected by the HAAR cascade classifier and if the CNN model was not trained with sufficient samples of images having adequate variation in lighting conditions, model performance was degraded. Model results plateaued for training set sizes > 1500 in this case, but the image number for training will depend on the variation expected in the validation sets. The use of scene specific knowledge and constraints should be considered throughout the model training process. The proposed model for fruit counting should generalize well to other tree fruit crops through transfer learning, after some fine tuning of CNN model parameters.

Conclusion and future work

Use of mobile device applications for yield estimation have potential as a low-cost and easily adoptable solution for farmers/growers, feeding into decision support tool for timely resource planning and orchard management.

Further work is required to (i) develop a canopy fruit occlusion index, for correction of dual view to total fruit count; (ii) collect images from further orchards, varying in canopy shape and fruit and foliage colour, (iii) documentation of the effect of test set size on CNN model performance and (iv) addition of a feature to allow cropping of the image on the screen of the mobile device, allowing for fruit load estimation of only one tree canopy at a time. For the fruit sizing mobile device application, addition of a distance measuring capability (e.g. time of flight laser range meter) would remove the need for use of a scale bar.

Acknowledgments

We acknowledge Groves and Simpsons farm for orchard access and support through HIA project MT14018. Anand Koirala acknowledges support of a RUN scholarship provided by CQ University.

References

Gibson A, Patterson J, Nicholson C 2016. Deeplearning4j: Open-source, Distributed Deep Learning for the JVM [Online]. Available: https://deeplearning4j.org [Accessed 16 June 2017].

Lecun Y, Bottou L, Bengio Y, Haffner P 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE 86: 2278–2324.

OPENCV 2017. OpenCV library [Online]. Forge. Available: http://opencv.org/ [Accessed 16 June 2017].

Otsu N 1979. A threshold selection method from gray-level histograms. IEEE transactions on systems, man, and cybernetics 9: 62–66.

Papageorgiou CP, Oren M, Poggio T 1998. A general framework for object detection. Computer vision, 1998. Sixth international conference on. IEEE 555–562.

Sethy PK, Panda S, Behera SK, Rath AK. On tree detection, counting & post-harvest grading of fruits based on image processing and machine learning approach - A review.

Viola P, Jones M 2001. Rapid object detection using a boosted cascade of simple features. Computer Vision and Pattern Recognition. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on, 2001. IEEE, I-I.