Automated Agronomy: Evaluation of Fruits Ripeness Using Machine Learning Approach

Grzegorz Chmaj, Saugat Sharma, and Henry Selvaraj

Department of Electrical and Computer Engineering, University of Nevada, Las Vegas, USA
{grzegorz.chmaj, henry.selvaraj}@unlv.edu, sharms8@unlv.nevada.edu

Abstract. Fruit orchards require a lot of tasks, monitoring the current state of fruits and anticipated time of harvest. Precise estimation of harvest time is a key knowledge for the supply chain and timely delivery to grocery stores. The current process relies on human agronomists, who visit orchards often and do the visual evaluation of number of fruits, their ripeness and expected time of harvest. In this paper, we propose a preliminary work for automation of the fruits’ evaluation process, using machine learning algorithms to evaluate pictures of the trees. For our current system we have used pictures – one picture per tree, to constitute the base for evaluation of each tree using multiple pictures and video captured by drones. In this paper, Convolutional Neural Network (CNN) is used for fruit image classification based on ripeness stage of the fruits. Fruits classification was based on appropriate surface color and shape features and CNN was used to extract these features for classification. The model showed 96.43% accuracy. The ultimate goal of this work is to fully automate the process of orchard trees evaluation, including estimation of number of fruits on the tree (including non-visible ones), their ripeness and time of harvest to address the commercial market delivery planning requirements.

Keywords: Machine learning · Agronomy · Drones · UAV

1 Introduction

The delivery of grocery products to the stores is a sophisticated and time sensitive process. Perishable products such as fruits and vegetables are not only important elements of grocery store operation, but also time sensitive as they need to be harvested right in time and then delivered to stores in a timely manner. Orchard trees fructify multiple times a year and not synchronously in locations with most sunny and warm days over the year. Therefore, fruits are harvested at different times from different trees and visual/tactile evaluation is needed periodically, to determine the ripeness of fruits and the number of fruits that will be ripe and ready for harvest on a specified, future date. The knowledge of the quantity of fruits possible to harvest at specified date is a key information for grocery stores and their supply planning process. The evaluation process is currently done by agronomists who visit orchards in person. The industry demand is to introduce automation of this process by using drones to deliver pictures
and videos of orchard trees that would then be delivered to computational system, which would do the estimation of number of fruits on the tree, ripeness and timing evaluation automatically.

In this paper, we present a partial element for such an orchard evaluation system. We focus on algorithms for evaluating pictures of trees. In future papers we will describe the remaining components of the system: drone (UAV) operation, UAV-to-computation multimedia delivery, UAV path planning and more advanced features of machine learning based algorithm: estimating number of fruits on the tree based on partial images/videos, categorized evaluation of ripeness, estimating the total number of deliverable fruits from an orchard for any given day in the future. Due to CoViD-19, current access to the testing orchard is limited, thus training of algorithms for the results in this paper is limited to the number of pictures we already have. Once orchards reopen, presented algorithms will be retrained and results will be included in future papers. The target system is presented in Fig. 1.

![Fig. 1. General structure of the target system](image)

## 2 Literature Review

The research work that is essential for the proposed system falls into the following categories: a) agricultural publications related to state of fruits and ways of their ripeness evaluation b) image processing for object recognition and features evaluation and c) machine learning for feature extraction and recognition.

The method of ripeness estimation based on video sourced from aerial vehicle was presented in [1] – authors used color features and genetic algorithms to determine four classes of apple ripeness (unripe, half-ripe, ripe, overripe). The algorithm detects apples and determines their ripeness – artificial neural networks are used in conjunction with genetic algorithms. Training was done using static images, authors assumed that in their future work such images would be delivered from aerial vehicle. In [2], the vision based robotic system for ripeness evaluation for sweet pepper is described. It includes detection, ripeness plus quantity evaluation and tracking. Fruits are tracked to avoid duplicate counting of each identified fruit. Vision system delivers 30 frames per second image input, coming from slowly moving robotics platform. Another method for apple orchards was presented in [3]. Instead of vision input, the ethylene sensors are used to determine the apple ripeness. The efficiency of the system was determined using simulated environment, including distribution of ethylene in the orchard environment and the impact of the wind generated by UAV motors, as the ethylene sensor would be embedded on an unmanned aerial vehicle. The results show that this method of
measurement is sensitive to wind, and that a UAV can sense ethylene at a maximum of 10% detection probability and at less than 20 feet from the ground. Authors of [4] use the smartphone-based spectrometer for ripeness testing. The proposed solution includes light source, spectrometer, filters, microcontroller, wireless circuits and software application running on the smartphone directly. The method uses the spectral data analysis acquired as fluorescence from chlorophyll on fruit skin exposed to ultraviolet light. Next method, using another type of data input, has been presented in [5]. Ripeness of pears is determined using acoustical vibrations – based on time-course changes in the elasticity index and texture index. The vibrational method is non-destructive and uses a laser Doppler vibrometer. Results show the correlation of mechanical attributes (hardness, crunchiness, and thickness) with elasticity index and small or no correlation of chemical attributes (sweetness, juiciness, and acidity).

In [6] convolution neural network (CNN) is used for the strawberry image ripeness classification. In [6] the input image is converted into matrix using the 3 * 3 convolutional layer, then batch normalization is performed. After that, it has used 2 * 2 Max-pooling where final data is flattened and fed into the Dense layer to generate the 3-classification output. In [6] training is performed with a total of 240 images and testing is done on 60 images, with an overall accuracy of 91.6%. Similarly, in [7], it shows the CNN implementation for the CIFAR-10 Image classification performed in multithread GPU. The CIFAR-10 has RGB images similar to images we used. The architecture used in [7] has used 3 convolutional layers, where the depth of layers is 16,16 and 32 respectively. The final images were applied to a fully connected layer and finally categorized into 10 different classes. In [8], the grapefruit ripeness classification accuracy is compared between CNN and SVM methods. Color feature and shape of the grapes [8] were chosen as features for classification where CNN model showed 79.49% accuracy, whereas SVM showed 68% classification accuracy, concluding that CNN is better for image classification than SVM. Similarly, in [9], CNN is used to classify the ripeness of the pineapple fruit. Color is used as features for classification [9]. In this research, image data are collected by taking pictures through 3 webcams tricked from Arduino Controller and labeled as unripe, partially ripe and fully ripe. This research has used already trained MobileNet CNN architecture and has predicted 98.38% and 90.77% of training and test accuracy [9].

3 Methods

For the classification of ripeness of Orange fruit, we divide fruits into three categories: 1. Green Unripe 2. Partial ripening (yellow color stage) 3. Ripe. The labeling of the fruits is done providing unique index to each of the stage. Index “0” is labeled as green unripe fruits, index “1” as partial ripe fruits and index “2” as ripe fruits (Fig. 2). Indexing is done randomly—the index number to any fruits category does not matter and any number can be assumed to any class but it must start from “0” as the label file is processed in python “List” and python “List” index starts with “0”. The images of the oranges are all stored in a folder and label is stored in a separate script file – each orange is given a tag either “0”, “1” or “2” depending upon there categories they belong to. This constitutes input data set.
The dataset used in experimentation can be characterized as follows (Table 1):

| Category of fruits | Number of images | Label |
|--------------------|------------------|-------|
| Green unripe       | 44, ids: 1–45    | 0     |
| Partial unripe     | 40, ids: 46–86   | 1     |
| Ripe               | 50, ids: 87–137  | 2     |

Convolutional Neural Networks (CNN) architecture was used to train and classify the Orange fruit stage. Before feeding into the CNN, all the images of different forms are first reshaped into size of 224 * 224 pixels and is stored in the list along with its label values. The array of labeled data (0 to 2) is converted into one-hot vector. The whole set of images is split into training and test sets, where 20% of images is training set. Our images consist of RGB components whose pixel values ranges from 0–255. The picture pixel values are kept as matrix form. For simplicity, these pixel values are normalized or rescaled between range 0–1 by a rescaling factor of 1/255. Thus, this will make mathematical computation easier with small values between 0–1 than the higher values between 0–255.

3.1 Creating Model

The model type used is “Sequential” built in Keras. Keras is high level neural network API [10]. It was originally written in Python and it allows very fast experimentation [10]. Keras runs on top of TensorFlow, CNTK, or Theano [11]. TensorFlow is deep neural network library [11]. It is used for numerical computational and machine intelligence [11]. We have used Keras on top of TensorFlow for our research. “Sequential Model” allows us to build the model layer by layer [11]. There are fully connected layers, max pool layers and activation layers built with Keras in Sequential model [11]. Thus, these layers make the architecture of the CNN. At first our image is converted into pixel values in matrix format. The input shape of our image is 224 * 224 * 3—Height,
Width and number of channels. The convolutional layer attracts the feature of our image by using filter. Filter moves over the input image and produces the feature map. We have used 3 * 3 size convolutional layer. Then, the activation layer is used. There are three types of most commonly used activation function: 1) sigmoid function 2) hyperbolic tangent function 3) Rectified Linear Unit (ReLU) [11]. We have used ReLu layer for our research.

Fig. 3. Flowchart showing the CNN model.
ReLu layer converts all the negative pixel values in our feature map to zero keeping the positive values untouched [11]. After the ReLu layer, pooling operation is done by pooling layer. Pooling layer is used to reduce the dimension of the feature map [11]. We have used Max-Pooling layer for our research. Max-Pooling layer computes the maximum of a local feature map [12]. We have used $2 \times 2$ as size of the Max-Pooling layer. Hence, it reduces the dimension of feature map. The Final layer is the Fully Connected layer, which performs the classification process. Before feeding into the fully connected layer, the final output from the Max-Pooling layer is flattened. Flatten layers converts the Pooled Feature Map into a single column and feeds into a fully connected layer/Dense layer [11]. The Dense layer converts every input to every output by a weight [11].

In our research, the final output from the Fully Connected layer is categorized into 3-classes: 1) green unripe 2) partially ripe 3) ripe. For final classification after the Fully Connected layer, we have used “SoftMax” activation function. “SoftMax” activation function squashes the final output values into probabilities by normalizing the prediction [13]. We have used “Adam” as optimizer and “categorical cross-entropy” as a loss function. Optimizer controls the learning rate [13]. The total parameters are 2,797,795.

Figure 3 shows the CNN Model where the input (RGB) image is applied to Convolutional, ReLu and Max-Pooling layer in series—these steps are executed 3 times in series - and finally it is flattened and passed to Dense layer whose output generates the final output classifying the image.

### 4 Results

The model was run for 3-epoches. An epoch is the number of times a complete data set is passed for training in the learning process of the learning machine. Thus, we have sent the complete data set 3 times for learning process. Accuracy is obtained by evaluating the number of correctly categorized images in our test data. Our research achieved an accuracy of 96.43% (Table 2).

| Epoch | Train loss (%) | Train accuracy (%) | Validation loss (%) | Validation accuracy (%) |
|-------|----------------|--------------------|---------------------|------------------------|
| 1     | 88.10          | 47.22              | 52.932              | 75                     |
| 2     | 57.49          | 75                 | 41.88               | 64.29                  |
| 3     | 52.13          | 69.44              | 28.93               | 96.43                  |

Graph in Fig. 4. shows the result obtained after completing the training in each epoch. The training accuracy is represented by blue line and validation accuracy by green line. The X-axis represents the number of epochs (total of 3 in our research) and y-axis represents the accuracy (0 to 1 or 0% to 100% accuracy). It shows the plot for accuracy in each epoch for training and validation. The figure shows the training
accuracy being 0.4 in the first epoch, 0.75 in the second epoch and 0.69 in our final epoch. Similarly, validation accuracy was 0.75, 0.64 and 0.96 in first, second and third epoch respectively.

Figure 5 shows the plot for train loss and validation loss. Train loss is represented by blue line and validation loss by green line. The figure shows the loss for both train and validation decreasing in every epoch. The training loss decreased from 0.88 to 0.57 and finally to 0.52 in first, second and third epoch respectively. Similarly, validation loss also decreased from 0.52 to 0.41 and finally to 0.28 in first, second and third epoch respectively. Hence, it is proved that loss is minimized on each successive iteration (epoch).

Fig. 4. Graph showing training accuracy vs validation accuracy
5 Conclusion

This preliminary work shows the basic platform for automatic classification of fruit images for agricultural purposes. Research based on available images shows high efficiency of used methods. Future goals are to provide automatic delivery of images, estimating number of ripe fruits on the tree and predicting number of ripe fruits for anticipated date. For this, we can categorize the set of fruit images into ripeness categories and depending upon the prediction done by CNN: if the ripeness predicted fall in one category, then we can say what is the approximate time it will take to mature into next categories. In addition, the exact labeling of the fruit images manually is a must to determine in which stage of ripeness the fruit falls. Thus, we can apply appropriate object detection algorithm in order to detect and count the number of fruits in a tree and apply algorithm to extract those detected images and predict its stages of ripeness and estimate its ripeness time (if is not ripened) with the algorithm developed in this research paper.

Work prepared with the cooperation with U.S. entity ATACT (Academy Of Threat Assessments, Countermeasures and Technology).

References

1. Sabzi, S., et al.: An automatic non-destructive method for the classification of the ripeness stage of red delicious apples in orchards using aerial video. Agronomy 9(2), 84 (2019). https://doi.org/10.3390/agronomy9020084
2. Halstead, M., McCool, C., Denman, S., Perez, T., Fookes, C.: Fruit quantity and ripeness estimation using a robotic vision system. IEEE Robot. Autom. Lett. 3(4), 2995–3002 (2018). https://doi.org/10.1109/lra.2018.2849514

3. Valente, J., Almeida, R., Kooistra, L.: A comprehensive study of the potential application of flying ethylene-sensitive sensors for ripeness detection in apple orchards. Sensors 19(2), 372 (2019). https://doi.org/10.3390/s19020372

4. Das, A.J., Wahi, A., Kothari, I., Raskar, R.: Ultra-portable, wireless smartphone spectrometer for rapid, non-destructive testing of fruit ripeness. Nat. Sci. Rep. 6. Article number 32504. https://doi.org/10.1038/srep32504

5. Taniwakia, M., Hanadab, T., Tohroc, M., Sakurai, N.: Non-destructive determination of the optimum eating ripeness of pears and their texture measurements using acoustical vibration techniques. J. Postharvest Biol. Technol. 51, 305–310 (2009). https://doi.org/10.1016/j.postharvbio.2008.08.004

6. Thakur, R., Suryawanshi, G., Patel, H., Sangoi, J.: An innovative approach for fruit ripeness classification. In: International Conference on Intelligent Computing and Control Systems (ICICCS). IEEE

7. Han, S.-H., Lee, K.-Y.: Implementation of image classification CNN using multi thread GPU. In: International SoC Design Conference (ISOCC). IEEE

8. Kangune, K., Kulkarni, V., Kosamkar, P.: Grapes ripeness estimation using convolutional neural network and support vector machine. In: Global Conference for Advancement in Technology (GCAT). IEEE (2019)

9. Chaikaew, A., Thanavanich, T., Duangtang, P., Sriwanna, K., Jaikhang, W.: Convolutional neural network for pineapple ripeness classification machine. In: International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON). IEEE (2019)

10. Arnold, T.B.: kerasR: R interface to the Keras deep learning library. J. Open Source Softw. 51, 305–310 (2017). https://doi.org/10.1016/j.postharvbio.2008.08.004

11. Sewak, M.: Practical convolutional Neural Network. Packt Publishing Ltd., Birmingham (2018)

12. Atienza, R.: Advanced Deep Learning with Keras. Packt Publishing Ltd., Birmingham (2018)

13. Priyanka, C.H., Rachel, V.S., Harshith, B., Moulisha, R.: Color recognition algorithm using a neural network model in determining the ripeness of a banana. J. Eng. Sci. 9(2), 305–313 (2009)