Real-time yield estimation based on deep learning

Rahnemoonfar, Maryam, Sheppard, Clay

Maryam Rahnemoonfar, Clay Sheppard, "Real-time yield estimation based on deep learning," Proc. SPIE 10218, Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping II, 1021809 (8 May 2017); doi: 10.1117/12.2263097

Event: SPIE Commercial + Scientific Sensing and Imaging, 2017, Anaheim, California, United States

Real-time Yield Estimation based on Deep Learning

Maryam Rahnemoonfar*, Clay Sheppard
Department of Computer Science, Texas A&M University-Corpus Christi, TX 78412

ABSTRACT

Crop yield estimation is an important task in product management and marketing. Accurate yield prediction helps farmers to make better decision on cultivation practices, plant disease prevention, and the size of harvest labor force. The current practice of yield estimation based on the manual counting of fruits is very time consuming and expensive process and it is not practical for big fields. Robotic systems including Unmanned Aerial Vehicles (UAV) and Unmanned Ground Vehicles (UGV), provide an efficient, cost-effective, flexible, and scalable solution for product management and yield prediction. Recently huge data has been gathered from agricultural field, however efficient analysis of those data is still a challenging task. Computer vision approaches currently face diffident challenges in automatic counting of fruits or flowers including occlusion caused by leaves, branches or other fruits, variance in natural illumination, and scale. In this paper a novel deep convolutional network algorithm was developed to facilitate the accurate yield prediction and automatic counting of fruits and vegetables on the images. Our method is robust to occlusion, shadow, uneven illumination and scale. Experimental results in comparison to the state-of-the art show the effectiveness of our algorithm.

Keywords: Deep learning, simulated learning, fruit counting, yield estimation

1. INTRODUCTION

Accurate yield prediction helps farmers to improve their crop quality. Moreover, it helps reducing the operation cost by making better decisions on intensity of crop harvesting and labor required. Generally, crop yield estimation is performed using past data, and workers manually count the fruit in chosen sampling locations in the field. This process is expensive both in terms of time and labor requirement and sometimes the chosen samples may not represent the population well [1]. Robots equipped with computer vision systems present an alternative solution. They don't need to be paid and can count fruit in hundreds of images per second. Computer vision based crop yield estimation methods can be divided roughly into two categories: 1) region or area based methods and 2) counting based methods. In the literature, there is ample amount of work dealing with region based methods [1-7]. Wang et al. [1] developed a stereo camera automatic crop yield estimation system for apple orchards. They captured images in the night time to reduce the effect of unpredictable natural illumination in the day time. Li et al. [2] developed an in-field cotton detection system based on region based semantic image segmentation. Lu et al. [3] developed region based color modeling for joint crop and maize tassel segmentation. In the literature very scarce attention has been paid on counting based yield estimation methods [8, 9]. Linker et al, [8] used color images to estimate the number of apples acquired in orchards under natural illumination. They could detect apples with more than 85% accuracy. The drawbacks are direct illumination and color saturation due to which a large number of false positives were observed. Park [9] developed a method to segment apple fruit from video using background modeling. There are various challenges faced by computer vision algorithms for counting fruits for yield estimation namely, illumination variance while capturing the image, occlusion by foliage and branches of the tree or crop, varied degree of overlap amongst fruits to be counted, counting the tomatoes which are under shadows, difference in scale while capturing the image. Due to the aforementioned challenges, it is very difficult to count the tomatoes or other fruits and perform yield estimation. The counting problem arises in several real world applications such as cell counting in microscopic images, forest flora and fauna counting in an aerial image, crowd monitoring in surveillance systems. Method proposed by Kim et. al [10] detects and tracks moving people with the help of a fixed single camera. Later, Lempitsky et. al [11] proposed a new supervised learning framework for visual object counting tasks that optimizes the loss based on the MESA-distance during the learning. Recently, Giuffrida et. al [12] proposed a learning-based approach for counting leaves in rosette (model) plants. They used a supervised regression model to relate image-based descriptors which are learned in an unsupervised fashion to leaf counts. Features extracted from Deep Convolutional Neural Networks [11] have been applied to numerous image understanding tasks such as object identification [5, 13-18] and semantic segmentation [19]. These features can also be regress to count. This method

Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping II,
edited by J. Alex Thomasson, Mac McKee, Robert J. Moorhead, Proc. of SPIE Vol. 10218,
1021809 · © 2017 SPIE · CCC code: 0277-786X/17/$18 · doi: 10.1117/12.2263097

estimates the count of tomatoes explicitly from the glance of the entire image. In this way it reduces the overhead of object detection and localization. The main advantage of this work is that thousands of real tomato images are not necessary for training. The network was trained using synthetic images and tested on real images and it works efficiently with 91% accuracy. The proposed methodology works efficiently even if there is illumination variance in the images and it can also count the tomatoes which are under shadows or occluded by foliage or overlap between tomatoes.

The following are the contributions of this work:

- A novel deep learning architecture for counting tomatoes was developed
- The algorithm works in real time
- The algorithm is scale invariant
- It can count accurately even if there is occlusion by foliage, branches or other tomatoes
- It can handle illumination variation and the effect of shadow

The rest of the paper is organized as follows. Proposed methodology is explained in section 2. Dataset description, evaluation measurements of the data used in the experiments, along with the experimental results are presented in section 3. Finally, the discussions and conclusions are drawn in section 4.

2. METHODOLOGY

2.1 Synthetic Image generation

Deep learning requires large datasets that are time consuming to collect and annotate. Due to this, we chose to instead use synthetic images. The synthetic images were generated as follows. A blank image is created followed by filling the entire image with green and brown color circles to simulate the background and the tomato plant, which are later blurred by Gaussian filter. To create the variable size tomatoes in the image, several circles of random size are drawn in random positions on the image. 24,000 images were generated for the training set, and 2,400 for the test set. Figure 1 shows some representative synthetic images that were generated to train the network.

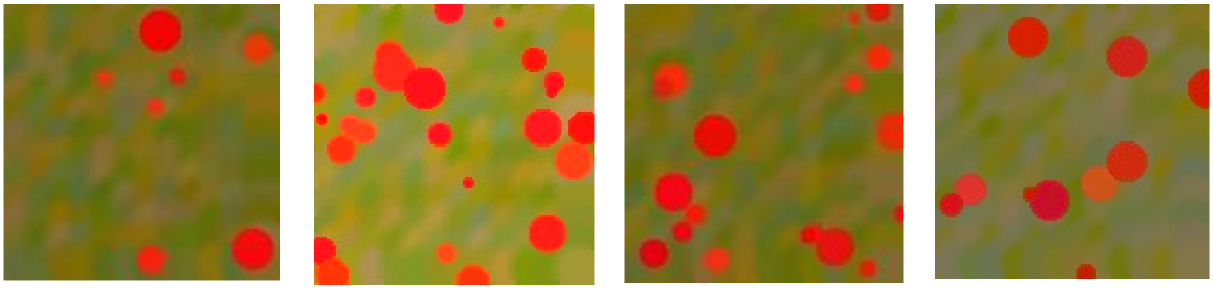


Figure 1. Representative samples of synthetic images

Synthetic tomato images were generated with some degree of overlap along with variation in size, scale and illumination in order to incorporate the possible complexities in real tomato images.

2.2 Description of the network architecture

The neural network design for this research is shown in Figure 2. The first layer of the network is 7X7 convolution layer followed by 3X3 max pooling layer with stride 2. This convolutional layer maps the 3 bands (RGB) in the input image to 64 feature maps using a 7X7 kernel function. Max pooling reduces the dimensions of the image by taking the maximum value in a window that is passed over the image. A stride of 2 means the window is moved 2 pixels at a time. This condenses information in the network. Reducing the dimensions of the image reduces computation time and allows the model to fit into GPU memory [20]. Similarly, in order to reduce the dimensionality of the feature maps a 1X1 convolution layer is used before another convolution layer of kernel size 5X5.

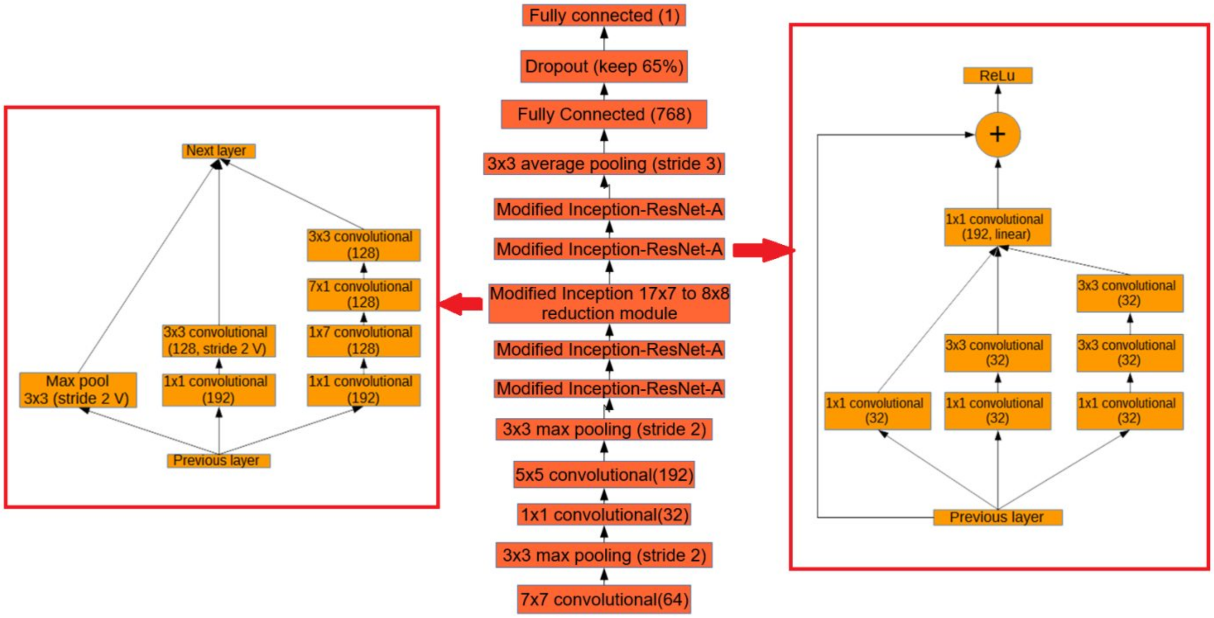


Figure 2.our proposed network for counting fruits

The size of the tomatoes in the images varies, so an architecture that can capture features at multiple scales is required. For this purpose, we modified the Inception-ResNet-A [18] layer. Two modified Inception-ResNet-A layers follow the normal convolutional layers. Inception-ResNet combines the ideas of Inception, which captures features at multiple sizes by concatenating the results of convolutional layers with different kernel sizes, and residual networks [15] which use skip connections to create a simple path for information to flow throughout a neural network. This architecture was used because of its high performance on several competitive image recognition challenges [18]. Residual networks converge faster because residual connections speed up training in deep networks [15]. The modified Inception-ResNet-A module consists of three parallel layers concatenated into one. The result of this concatenation is then added to the activations of the previous layer and passed through the Rectified Linear function. After the modified Inception-ResNet-A layers, a modified Inception reduction module is used to simultaneously reduce image size and expand the number of filters. Although deep neural nets with a large number of parameters are very powerful machine learning systems, overfitting is a serious problem in such networks. Large networks are also slow to use, making it difficult to deal with overfitting by combining the predictions of many different large neural nets at test time. Dropout is a technique for addressing this problem. The key idea is to randomly drop units (along with their connections) from the neural network during training [21]. 65% of connections were randomly kept while training the network. Finally, the last fully connected layer after the dropout layer gives the prediction for the number of tomatoes in the input image. Batch normalization was performed after every convolution to remove internal covariate shift [22]. The network was trained for 3 epochs on 24,000 synthetic images. To minimize the error an Adam optimizer is used [23] due to it requiring little tuning of hyperparameters. The learning rate for the Adam optimizer was set at a constant 1e-3. Mean squared error was used as the cost function. The network was evaluated using the exponential moving averages of weights. Weights were initialized using a Xavier initializer [24]. Xavier initialization keeps the scale of the gradients approximately the same throughout the network.

3. EXPERIMENTAL RESULTS

The network was implemented using TensorFlow [25] running on an NVidia 980Ti GPU. For training, 24,000 images were used. For testing, a different set of 2400 synthetic images and 100 randomly selected real tomato images were used. Validation on 2400 synthetic images gives a mean squared error for the count of about 1.16. Figure 3 shows the mean square error for training where abscissa represents number of steps and ordinate represents mean square error. The network was trained with three different dropout values (50, 65, and 80) to find the least value for mean square error and

65% was chosen as the dropout keep probability for the network. Figure 3 shows the mean square error for dropout value 65. Looking at the graph in Figure 3, it is clear that the network converges quickly; that is why the network was trained for only three epochs.

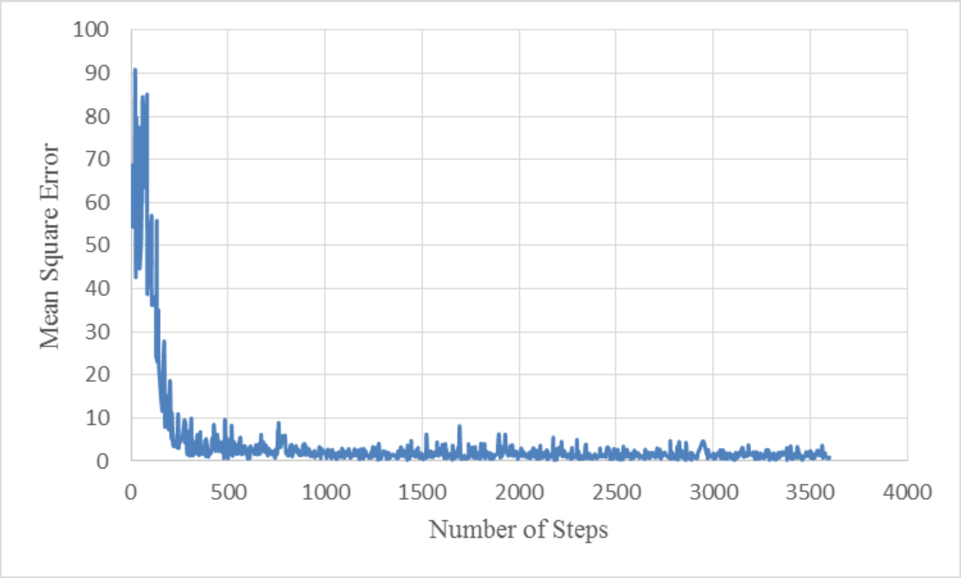


Figure 3. Mean square error for training at dropout value 65

3.1 Experimental results with real data

Despite not having real images for training, the network performs well for real images. The algorithm was tested over 100 randomly chosen images. Table 1 shows eight representative images for display along with their predicted and actual count. In Table 1 column R contains real images, column P contains predicted count and column GT contains actual count (ground truth). The network was trained to count only ripe and half-ripe tomatoes. Unripe (green) tomatoes were not considered. The algorithm can handle variation in illumination. As can be seen in Table 1, the randomly chosen images are at different illumination condition. The accuracy was calculated as follows:

$$percentage\ accuracy = \left[1 - \frac{\left| predicted\ count - actual\ count \right|}{\left| actual\ count \right|} \right] \times 100$$

The average accuracy for a hundred images is found to be 91%

The performance of the proposed method was compared with the conventional area based or region based method. The area method is based on calculating the total pixel coverage of the tomatoes. Later the total pixel coverage is divided by the average pixel coverage of one tomato to get the count. There are several area based methods available in the literature, the only difference is how the area is calculated. These methods are highly affected by the occlusion and scale of the image. The average accuracy over one hundred images is 66.16%. Table 2 shows the average time required to count the tomatoes in one test image using the proposed method, area based method and the time required by a human.

Table 1 Real tomato images with predicted and actual count

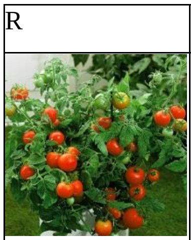
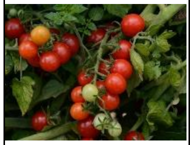
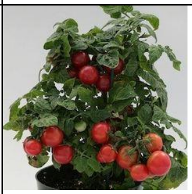
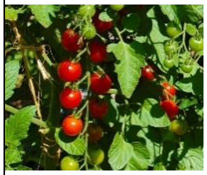
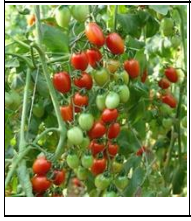
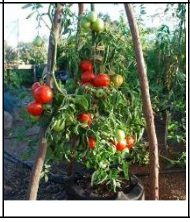
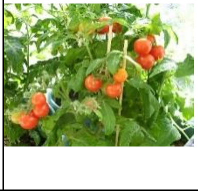
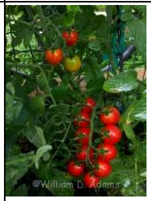
R	P	GT	R	P	GT	R	P	GT	R	P	GT
	22	25		21	23		15	14		12	12
	22	22		13	12		14	14		14	13

Table 2. Average time required to count the tomatoes

Method	**Average Time required for one test image (seconds)**
Proposed method	0.006
Area based method	0.05
Manual counting	6.5

With the help of Table 2, it is clear that the proposed method is faster than that of the area based and manual counting method. Table 3 shows the average accuracy and standard error over one hundred images using the proposed method and area based counting.

Table 3 the average accuracy and standard error over one hundred images

Method	**Average accuracy (%)**	**Standard Error**
Proposed method	91.03	2.5
Area based counting	66.16	7.9

With the help of Table 3, it can be inferred that the proposed method is significantly better than the area based method. The standard error of the area based method is very large as compared to the proposed method. The reason is the area based method is not scale invariant. Moreover, the main problem with area based methods is, whenever there is occlusion by other tomatoes, foliage or branches, the total pixel coverage of the tomatoes will be less than the actual coverage and will lead to a false count of the tomatoes. Our method [26] also shows a great improvement in comparison to shallow network.

4. CONCLUSION

We proposed a convolutional neural network based system for counting tomatoes. We based our architecture on Inception-ResNet to achieve high accuracy and to lower the computation cost. It is very difficult to get a sufficient number of real images to be used as training data for deep learning, so we generated synthetic tomato images for our experiment. We observed 91% accuracy for one hundred randomly chosen real images. Our algorithm is robust under poor conditions. It can count accurately even if tomatoes are under shadow, occluded by foliage, branches or if there is some degree of overlap amongst tomatoes. Although our algorithm was trained to count tomatoes, it can be applied to other fruits. In the future, we wish to perform localization along with the counting. Furthermore, unmanned aerial vehicle (UAV) and unmanned ground vehicle (UGV) can be utilized to monitor large-scale fields. Then we can use the obtained ground-based data for some agricultural applications, such as yield estimation and planting density.

REFERENCES

[1] Q. Wang, S. Nuske, M. Bergerman, and S. Singh, "Automated crop yield estimation for apple orchards," in *Experimental Robotics*, 2013, pp. 745-758.

[2] Y. Li, Z. Cao, H. Lu, Y. Xiao, Y. Zhu, and A. B. Cremers, "In-field cotton detection via region-based semantic image segmentation," *Computers and Electronics in Agriculture*, vol. 127, pp. 475-486, 2016.

[3] H. Lu, Z. Cao, Y. Xiao, Y. Li, and Y. Zhu, "Region-based colour modelling for joint crop and maize tassel segmentation," *Biosystems Engineering*, vol. 147, pp. 139-150, 2016.

[4] G. Schillaci, A. Pennisi, F. Franco, and D. Longo, "Detecting tomato crops in greenhouses using a vision based method," in *Proceedings of International Conference Ragusa SHWA2010, Ragusa Ibla, Italy*, 2012, p. 252258.

[5] L. Wang, S. Liu, W. Lu, B. Gu, R. Zhu, and H. Zhu, "Laser detection method for cotton orientation in robotic cotton picking," *Transactions of the Chinese Society of Agricultural Engineering*, vol. 30, pp. 42-48, 2014.

[6] M. Teixidó Cairol, D. Font Calafell, T. Pallejà Cabrè, M. Tresánchez Ribes, M. Nogués Aymamí, and J. Palacín Roca, "Definition of linear color models in the RGB vector color space to detect red peaches in orchard images taken under natural illumination," *Sensors, 2012, vol. 12, núm. 6, p. 7701-7718,* 2012.

[7] W. Jie-ding, F. Shu-min, W. Mu-lan, and Y. Jian-ning, "Research on the Segmentation Strategy of the Cotton Images on the Natural Condition Based upon the HSV Color-Space Model [J]," *Cotton Science*, vol. 1, p. 010, 2008.

[8] R. Linker, O. Cohen, and A. Naor, "Determination of the number of green apples in RGB images recorded in orchards," *Computers and Electronics in Agriculture*, vol. 81, pp. 45-57, 2012.

[9] A. L. Tabb, D. L. Peterson, and J. Park, "Segmentation of apple fruit from video via background modeling," in *2006 ASAE Annual Meeting*, 2006, p. 1.

[10] J.-W. Kim, K.-S. Choi, B.-D. Choi, and S.-J. Ko, "Real-time vision-based people counting system for the security door," in *International Technical Conference on Circuits/Systems Computers and Communications*, 2002, pp. 1416-1419.

[11] V. Lempitsky and A. Zisserman, "Learning to count objects in images," in *Advances in Neural Information Processing Systems*, 2010, pp. 1324-1332.

[12] M. V. Giuffrida, M. Minervini, and S. A. Tsaftaris, "Learning to count leaves in rosette plants," 2016.

[13] J. Han, D. Zhang, X. Hu, L. Guo, J. Ren, and F. Wu, "Background prior-based salient object detection via deep reconstruction residual," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 25, pp. 1309-1321, 2015.

[14] V. Nair and G. E. Hinton, "3D object recognition with deep belief nets," in *Advances in Neural Information Processing Systems*, 2009, pp. 1339-1347.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *arXiv preprint arXiv:1512.03385*, 2015.

[16] Y. Zhu, R. Urtasun, R. Salakhutdinov, and S. Fidler, "segdeepm: Exploiting segmentation and context in deep neural networks for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4703-4711.

[17] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91-99.

[18] C. Szegedy, S. Ioffe, and V. Vanhoucke, "Inception-v4, inception-resnet and the impact of residual connections on learning," *arXiv preprint arXiv:1602.07261*, 2016.

[19] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, *et al.*, "DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition," in *ICML*, 2014, pp. 647-655.

[20] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," *arXiv preprint arXiv:1603.07285*, 2016.

[21] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research,* vol. 15, pp. 1929-1958, 2014.

[22] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167,* 2015.

[23] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980,* 2014.

[24] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Aistats*, 2010, pp. 249-256.

[25] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, *et al.*, "TensorFlow: Large-scale machine learning on heterogeneous systems, 2015," *Software available from tensorflow. org,* vol. 1, 2015.

[26] M. Rahnemoonfar and C. Sheppard, "Deep Count: Fruit Counting Based on Deep Simulated Learning," *Sensors*, vol. 17, 2017.