

Instance Segmentation of Edamame Pods

Kshitiz Dhakal, Puspa Kamal Poudel
School of Plant and Environmental Sciences
Virginia Tech, Blacksburg, VA, 24060
kshitiz@vt.edu

Abstract

Pod detection in edamame is an important trait to estimate the total yield. Mask R-CNN and YOLO are the popular and state of the art object detection methods. We compared the performances of Mask R-CNN and YOLO in detecting the pods in edamame and found that YOLO was better in detecting the pods in edamame plant images than the Mask R-CNN.

Introduction

Plant phenotyping is a method of extracting useful information from plant parts. Edamame (popularly known as vegetable soybean) is an important source of protein and has greater scope in the economy as the USA imports 75% of the edamame consumed in the country. It is labor intensive to physically measure the economically important yield traits of edamame such as pod (fruit part of the edamame plant) numbers and pod locations.

As compared to industrial automation, the automation in agriculture is difficult due to the uncertainty regarding plant structure and the field conditions. Proper detection and localization for yield components such as fruits are important components for monitoring, robotics and autonomous systems for agriculture [1]. Fruit counting and yield estimation are the most important benefits of detection not only to the farmers but also to the breeders and researchers [2,3,4,5]. Commonly available RGB cameras and the popular plus affordable computer vision tools can be used to remove the bottlenecks of plant phenotyping. There is a need for automatic pod detection to predict the yield of edamame.

The State-of-the-art computer vision systems based on deep convolutional neural networks can deal with various environments to robustly recognize the complex objects in outdoor environments. Recent research [6,7] have demonstrated that the Faster R-CNN (region-based convolutional neural network) architecture [8] can produce better results for fruit detection. These previous detection systems identify individual objects by bounding the boxes around the object. Such bounding boxes, if well fitted to the fruit's boundaries, could provide estimations of fruit shape and space occupancy for fruits. However, for edamame, rectangular boxes would not properly adjust to the pods as they are in clusters. The next steps can be instance segmentation, which can detect the several instances of the objects in an image. Instance segmentation can properly identify the pod pixels in the plant images.

Mask R-CNN [9], is a convolutional framework for instance segmentation that is simple to train and generalizes well [10], and YOLO [11], is a single-stage network that can detect objects without a previous region-proposal stage [12].

Instance segmentation (e.g., Mask R-CNN, and YOLOv5) can be used as a supervised machine learning problem to automatically count the number of pods located in the edamame plant images, but the first

thing that is needed for the machine learning problem is the large dataset of different varieties of plants with varying pod number. The dataset must have the mask that surrounds the objects(pods) in the images which can isolate pods from background pixels and from occluding objects. We need a neural network architecture able to simultaneously perform object detection and pixel classification.

Contribution

Our current work can contribute the following to the research (plant phenotyping and computer vision) community:

- A. A new methodology for image annotation that employs interactive image segmentation to generate object masks, identifying background and occluding foreground pixels:
- B. A new public dataset for pod detection and instance segmentation, comprising images, masks – this dataset is composed by images of hundred edamame varieties taken from the field:
- C. An evaluation of deep learning detection architectures for pod detection: Mask R-CNN, and YOLO; A pod counting methodology that can localize the detection results in space, avoid the multiple counting and accumulate evidence from different varieties to confirm detections.

Methods

Dataset Preparation

For this project we used two hundred plant images (6000 × 4000 pixels). Pods in each image were labeled manually using VGG Image Annotator (VIA), which generated a JavaScript Object Notation (JSON) file for all labeled images. The VIA [13] is a popular tool used by the computer vision community. It allows users to mark objects of interest using rectangles, circles, ellipses or polygons. The JSON annotation file was converted to COCO and then to YOLOv5PyTorch.txt format using Roboflow [14].



Fig 1: Sample of an annotated masked image using VIA

We divided the images into training (180 images), validation (10 images) and testing sets (10 images) for this project as shown in the table1. The total number of pods that were labelled or detected for each dataset are also listed in the table.

Table 1: Number of images and pods used in the training, validation and testing dataset

Features	Training	Validation	Testing	Total
Number of Images Used	180	10	10	200
Number of Pods Labeled (Masked)	5280	1123	1187	7590

Automatic detection of pods

In this study, for automatic estimation of the number of pods per plant, two different approaches with two deep neural networks were evaluated and their efficiency were analyzed. Our first approach employed the state-of-the-art instance segmentation network Mask-RCNN for segmentation of each instance of pod in the edamame images and estimate their numbers simultaneously. In our second approach we used YOLOv5, which is a single-stage network that can detect objects without a previous region-proposal stage. In YOLO, the image is split into a fixed grid of $S \times S$ cells. We compare YOLO and Mask R-CNN results on pod detection, and we evaluate Mask R-CNN results on pod instance segmentation.

CNN architecture

Mask R-CNN [9] is a derived version of Faster R-CNN able to perform instance segmentation, jointly optimizing region proposal, bounding box regression and semantic pixel segmentation. However, unlike object detection in which rectangular bounding boxes annotations are sufficient for training, instance segmentation needs image pixels to be properly attributed to an instance or to the background in the training dataset for supervised learning. In the experiment section, we describe a methodology for pod instance segmentation based on Mask R-CNN. Mask R-CNN [9] is essentially the combination of a Faster R-CNN object detector [8] and a fully convolutional network (FCN) [15] for semantic segmentation, providing a complete, end-to-end, instance segmentation solution. The Mask R-CNN employs as feature extractor a feature pyramid network (FPN) [16], an architecture able to create semantic feature maps for objects at multiple scales, built over a ResNet [9].

Another approach to object detection is to predict the locations and the objects' class in a single step, in order to avoid a previous region proposal procedure. Huang et al. [12] refer to this approach as single shot detector meta-architecture, and the YOLO networks proposed by Redmon et al. [9]; Redmon & Farhadi [17] are prominent members of this family. In the YOLO networks, the image is split into a fixed grid of $S \times S$ cells. A cell is responsible for performing a detection if an object center is over it. Each cell is associated with B boxes, composed by 5 values representing the object center (cx, cy), the object width and height and a confidence score that represents the model confidence that the box contains an object and the accuracy of the box boundaries regarding the object. The box also includes C conditional class probabilities, one to each class of objects. The training step tries to minimize a loss function defined over such a tensor, performing detection and classification in a single step. The YOLOv5 and other YOLO networks have a few differences, mainly regarding their feature extraction convolution part.

Training

Table.1 shows the splitting between training, validation and test sets. In this Section, we will show that although the differences in the numbers of images and pods, the results are very similar for all edamame varieties. For instance, segmentation, a set of 200 images presenting masks is available for training. We have split it into a 180 images training set (5280 pods) and a validation set composed of 10 images (1123 pods). We also had 10 images for testing which had 1187 pods.

For instance, segmentation, we employed Keras/TensorFlow-based implementation for Mask R-CNN by Matterport [18]. We used customized Matterport's implementation of Mask R-CNN for patch-based processing for our final image analysis approach. Pixel level instance masks and point masks were used for training Mask R-CNN. The network was initialized using the weights previously computed for the COCO Dataset [16]. No layer was frozen during training, so all weights could be updated by the training on the edamame dataset. The input images have pixel sizes of $6000 \times 4000 \times 3$ tensors. We also used YOLOv5[19], which is a family of object detection architectures and models pretrained on the COCO dataset. The

implementation of YOLOv5 was based on PyTorch; initialized using pre-trained weights from ImageNet [20]. All the networks were trained and tested on GPU (Nvidia GeForce GTX 1660 Ti) with a machine having Intel® core i5-9400 CPU, and 60 GHz processor. For Mask R-CNN, training was performed in approximately (80 epochs, around 8 minutes and 52 seconds per epoch). YOLO training was performed in approximately (100 epochs, around 6 minutes and 33 seconds per epoch).

Experimental Results

We expected to increase the accuracy of Mask-RCNN 0.99 and increase the accuracy YOLO of edamame pods detection as compared to 0.90 [4]. The results obtained from our experiment are presented in Table 2.

Method	mAP	Precision	Recall	F1-Score	Accuracy
MRCNN	0.43	0.86	0.51	0.64	0.47
YOLOv5	0.66	1.00	0.70	0.82	0.85

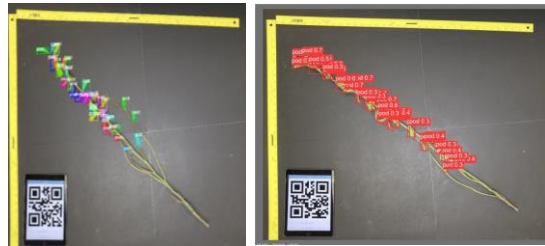


Fig 2: A Detection result from Mask-RCNN and B. Detection result from YOLO

The mean average prediction was 0.43 for Mask R-CNN and almost 0.66 for YOLO. In the object detection for Mask R-CNN, we used 180 images containing 6403 masked pods as shown in Fig.1. Mask R-CNN results have been evaluated using mAP (mean Average Precision), and efficiency.

Discussions

We analyzed and compared the performance of different models on the 115 varieties of edamame plant images and made the image datasets publicly available. Our experiments will be providing useful information for edamame breeding research and early yield prediction. The validation set was employed to select the best models for further evaluation on the test set. Table 2 presents the evaluation of predictions produced by Mask R-CNN and YOLOv5 for instance segmentation, considering the masked test set (1123 pods in the ground truth). The table shows the images and pods that were detected by both methods.

Figure 2 shows examples of instance segmentation results produced by the Mask R-CNN and YOLOv5. It illustrates the network capability to learn shape and size variability and discriminate occluding foreground as iPad's barcode rulers. Edamame pod segmentation is challenging, even to the human annotators: occlusions and the absence of 3-D input or on-site annotation make the dataset error-prone regarding the correct segmentation of large agglomerations of pods.

Table 2 presents the results for object detection, considering the validation set of 1123 pods in 10 images. As mentioned previously, the models were trained using the masked training set, composed of 200 images (5280 pods). The YOLOv5 network presented superior results as compared to the Mask R-CNN. The mAP for YOLOv5 seems to be higher than the Mask R-CNN and the YOLOv5 is more time efficient than the Mask

R_CNN. The detection accuracy and other quantification matrices were more for YOLOv5 as compared to the Mask R-CNN, making the YOLOv5 faster and more accurate object detection method for our dataset. The number of pods identified in both the cases were not the same as the total number of pods present in the testing dataset. Out of 1187 pods present in the testing dataset, 535 pods were detected by the Mask R-CNN, while YOLOv5 accurately identified 1009 pods out of 1187 testing dataset.

Agronomic constraints could be explored: how the occluded pods (pods that are underneath the branches or underneath another pod) be counted and detected. In other words, the operational and agronomical context should be explored to define the scales of interest.

Conclusion

This work presents a methodology for pod detection, tracking and counting in RGB images. We have reached mAP scores, which were more than the other similar research (for instance detection in wine grapes). The same methodology could be used successfully for other crops such as apples, peaches and berries. Further research could consider more integration between the photogrammetry and perception modules, looking for more sophisticated scene understanding.

Contributions

KD and PKP both prepared the input files for the project that included image collection and annotation. KD performed the Mask R-CNN, while PKP performed YOLO. We also tried U-Net for the dataset, but the accuracy was very low (12%), so we did not put the results in this report. KD plan to do it along with U-Net and publish the results in some machine learning related plant science journal.

References

- [1] Duckett, T., Pearson, S., Blackmore, S., & Grieve, B. (2018). *Agricultural robotics: The future of robotic agriculture*. CoRR, abs/1806.06762. URL: <http://arxiv.org/abs/1806.06762>. arXiv:1806.06762.
- [2] Kicherer, A., Herzog, K., Bendel, N., Klück, H.-C., Backhaus, A., Wieland, M., Rose, J. C., Klingbeil, L., Labe, T., Hohl, C., Petry, W., Kuhlmann, H., Seiffert, U., & Töpper, R. (2017). *Phenoliner: A new field phenotyping platform for grapevine research*. *Sensors*, 17. doi:10.3390/s17071625.
- [3] Rose, J., Kicherer, A., Wieland, M., Klingbeil, L., Töpper, R., & Kuhlmann, H. (2016). *Towards Automated Large-Scale 3D Phenotyping of Vineyards under Field Conditions*. *Sensors*, 16, 2136. doi:10.3390/s16122136.
- [4] Rahim, U. F., Utsumi, T., & Mineno, H. (2021). *Comparison of Grape Flower Counting Using Patch-Based Instance Segmentation and Density-Based Estimation with Convolutional Neural Networks* (No. 6539). EasyChair.
- [5] Santos, T. T., de Souza, L. L., dos Santos, A. A., & Avila, S. (2020). *Grape detection, segmentation, and tracking using deep neural networks and three-dimensional association*. *Computers and Electronics in Agriculture*, 170, 105247.
- [6] Bargoti, S., & Underwood, J. (2017a). *Deep fruit detection in orchards*. In *2017 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 3626–3633). IEEE. doi:10.1109/ICRA.2017.7989417. arXiv: arXiv:1610.03677v2.
- [7] Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., & McCool, C. (2016). *DeepFruits: A Fruit Detection System Using Deep Neural Networks*. *Sensors*, 16, 1222. doi:10.3390/s16081222.
- [8] Ren, S., He, K., Girshick, R. and Sun, J., “Faster r-cnn: Towards real-time object detection with region proposal networks,” *Adv. Neural Inf. Process. Syst.*, 91–99 (2015).
- [9] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). *You only look once: Unified, real-time object detection*. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779–788).
- [10] Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., & Pietikainen, M. (2018). *Deep Learning for Generic Object Detection: A Survey*, URL: <http://arxiv.org/abs/1809.02165>. arXiv:1809.02165.
- [11] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). *You only look once: Unified, real-time object detection*. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779–788).
- [12] Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., & Murphy, K. (2017). *Speed/accuracy trade-offs for modern convolutional object detectors*. In *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [13] Dutta, A., Gupta, A., & Zissermann, A. (2016). *VGG image annotator (VIA)*. <http://www.robots.ox.ac.uk/~vgg/software/via/>. Version: 2.0.6, Accessed: April 23, 2019.
- [14] Alexandrova, S., Tatlock, Z., & Cakmak, M. (2015, May). *RoboFlow: A flow-based visual programming language for mobile manipulation tasks*. In *2015 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 5537-5544). IEEE.

- [15] Shelhamer, E., Long, J., & Darrell, T. (2017). *Fully Convolutional Networks for Semantic Segmentation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39, 640–651. doi:10.1109/TPAMI.2016.2572683.
- [16] Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). *Feature pyramid networks for object detection*. In *the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
- [17] Redmon, J., & Farhadi, A. (2017). *Yolo9000: better, faster, stronger*. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7263–7271).
- [18] Matterport, Inc (2018). *Mask R-CNN for Object Detection and Segmentation*.
https://github.com/matterport/Mask_RCNN. Commit: 4f440de, Accessed: December 8, 2021.
- [19] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu and C. Xu, "GhostNet: More Features from Cheap Operations", *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1577-1586, 2020.
- [20] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). *Imagenet: A large-scale hierarchical image database*. In *2009 IEEE Conference on Computer Vision and Pattern Recognition* (pp. 248–255). Ieee.