Scene Graph Generation with Structured Aspect of Segmenting the Big Distributed Clusters

AR Khan\(^1\), Hamza Mukhtar\(^2\), Tanzila Saba\(^1\), Omer Riaz\(^3\), M. Usman Ghani Khan\(^2\) and Saeed Ali Bahaj\(^4\)

\(^1\)Artificial Intelligence & Data Analytics Lab CCIS Prince Sultan University Riyadh 11586, Saudi Arabia; drtasab@gmail.com, arkhan2030@gmail.com
\(^2\)National Center of Artificial Intelligence KICS UET, Lahore, 54000, Pakistan; hamza.mukhtar@kics.edu.pk, usman.ghani@kics.edu.pk
\(^3\)Department of Computer Science Islamic University, Bahawalpur Pakistan; omer.riaz@iub.edu.pk
\(^4\)MIS Department, College of Business Administration, Prince Sattam bin Abdulaziz University, Alkhairj, 11942, Saudi Arabia; s.bahaj@psau.edu.sa

ABSTRACT Accurate fruit counting is one of the important phenotypic traits for crucial fruit harvesting decision making. Existing approaches perform counting through detection or regression counting. Detection of fruit instances is very challenging because of the very small fruit size as compared to the whole size image of a tree, while regression-based counting gives impressive results but becomes inaccurate when the number of instances increases. Moreover, most approaches lack scalability and are applicable only on one or two fruit types. In this paper, we propose a fruit counting mechanism that combines loose segmentation and regression counting that works on six fruit types, such as Apple, Orange, Tomato, Peach, Pomegranate and Almond. Through relaxed segmentation, fruit clusters are segmented to extract the small image regions which contain the small cluster of fruits. Extracted regions are forwarded for the regression counting of fruits. Relaxed segmentation is achieved through a state-of-the-art deconvolutional network, while modified Inception Residual Networks (ResNet) based nonlinear regression module is proposed for fruit counting. For segmentation, 4,820 original images, including corresponding mask images, of all six fruit types are augmented to 32,412 images through different augmentation techniques, while 21,450 extracted patches are augmented to 89,120 images used for the regression module training. The proposed approach has superseded the counting accuracy of existing approaches of individual fruit types, but we have achieved an overall 94.71% accuracy.

INDEX TERMS Deep learning; segmentation; fruit counting; agricultural yield estimation; economic growth; agriculture; technological development

I. INTRODUCTION

Yield estimation is becoming increasingly important in digital agriculture, which assists farmers to streamline harvesting resources which boost the cost-cutting for harvesting, enabling them to market the yield in a better way to get higher profits. With prior estimation of yield, farmers can make substantiate decisions to arrange the labor and machinery for ripping the crop, early order of required packing stuff, manage logistics to transfer and prepare sizable storage and processing facilities [1]. With prior decisions, farmers can devise better marketing and sales strategy to get a higher price. On the other hand, manual fruit estimation in orchards is quite labor-intensive, giving inaccurate numbers and infeasible at a large scale.

Agri vision using image processing through Computer vision is a growing field that can also assist in yield estimation. Traditional image processing techniques are inefficient due to varying lighting conditions, color complexion, lack of robustness, occlusion and process hand-engineered features against each specific scenario [3], [4]. With Deep learning techniques, such as Convolutional Neural Networks (CNN), limits of image processing have been extended, which solves the complex Computer Vision problem, such as classification, detection and segmentation [5]. In addition, deep learning techniques are efficient enough to generalize across various fruit types and environments that are dynamic in lighting conditions.

Significant progress has been made to devise different approaches to formulate an efficient and accurate system for orchard fruit counting. Several object detection methods have been developed based on localization and classification of...
The extracted patches go to the counting module to get the count on individual patches, which is summed up to get a total count for a single image. Our experiment illustrates that the proposed loose segmentation-based counting model obtains better and more efficient counting output than detection-based regression methods.

In the next section, we have described the proposed approach for counting. Then, relating to the problem in previous systems, we have explained our estimation mechanism for fruit counting in the 3rd section. Then, in the 4th section, we have elaborated on the dataset, training setup, and evolution of output. Lastly, the 5th section summarizes our work and feeds the future course.

II. GUIDELINES FOR MANUSCRIPT PREPARATION

Object detection has been in the spotlight over the past years, and fabulous work has done [14], [15]. This problem is tried to be solved through unsupervised way through clustering of objects based on motion similarities [12] or structural similarity [13], but such unsupervised approaches have accuracy limitations, and to achieve higher accuracy supervised approaches are considered. Broadly, counting solutions are of three categories [16]: (1) clustering-based counting, (2) regressing based counting, and (3) object detection-based counting.

Clustering, unsupervised learning, based approaches are the initial work on counting problems. Objects are clustered based on similar features, such as texture, appearance, color and motion [17], and objective is to maximize the likelihood, which grouped the individual object instances on low-level features. For example, a motion analysis-based mechanism is proposed for the moving objects where a parallel KLT tracker is used to observe the motion and appearance of feature points, and clustered are made based on observed features [18]. Unsupervised approaches use lower level features and perform inaccurately on counting when we see in contrast with state-of-the-art Deep learning approaches.

Regression-based counting approaches are very accurate and efficient because the counting mechanism is learnt explicitly rather than optimizing object localization. They learn the direct mapping from image features to count labels, and for this learning, a huge amount of annotated data is required. [14] proposed a method, called glance, which explicitly learns the counting by mapping labelled counts on the image. Regression-based approaches are inefficient and give low accuracy when object instances are in large numbers [15].

Counting through object detection, draw the bounding boxes on detected objects, and just count the bounding boxes. Ground-truth labels are given in bounding boxes around objects for training [19-22]. Perfect detection leads to perfect counting, but Chattopadhyay et al [15] manifested that detection method can perform poorly because the model needs to learn the object shape, size and localize it regardless of occluded real work conditions. Therefore, methods of
detection based on pixel-level ground truth are also proposed [2], [23].

Song et al. [24] suggested a counting method with two models: (1) bag-of-words model to discover the fruit instances in an image, and (2) aggregate model to sum up the count using a statistical approach on a bunch of given images. Maldonado et al [25] presented a method for green orange fruits counting based on correlation between visible fruits and whole fruits on trees. Feature extraction is performed by combining the techniques such as Gaussian blur thresholding, histogram, color conversion, spatial filtering and Sobel operator. Input image is converted into a bas-relief representation on which filtering is applied and forward to SVM which decides whether the object is fruit and counts the positive decisions. However, large adjustable parameters and manual feature extraction are prolonged and not robust in occluded conditions. Linker [35] suggested an estimation procedure based on light distribution. Dorj et al. [36] used color features to recognize fruit instances, conversion of RGB image to HSV, and different preprocessing techniques used for counting.

Rahmennonfar et al. [7] proposed an inception-ResNet based estimation approach that maps the labelled count on images and reduces detection and localization cost. Training is performed on synthetic data and tested on real tomato images. Chen et al [27] suggested a deep learning approach that directly maps total count to input images. Candidate regions are extracted through a convolutional network-based blob detector, another convolutional network is employed to estimate the count in each extracted region, and a regression model map estimated count to a final count. Qureshi et al [28] proposed two methods: (1) texture base segmentation based on K-Nearest neighbor classification and segmentation, and (2) segmentation-based method which uses a support vector machine for classification. Bargoti et al. [29] presented a segmentation-based approach that consists of a multilayer perceptron and convolutional neural network. Segmentation is generated using watershed segmentation and individual fruits are counted through conducted Hough transform.

Liu et al [30] presented a segmentation and 3D localization model for counting. Fully convolutional network is used for segmentation and localization using an incremental structure motion algorithm. Ponce at al [31] proposed the counting method based on mathematical morphology which segment the olives to extract feature representation. Häni et al. [32] proposed a semantic segmentation model based on U-Net architecture and CNN for classification. Belloccchio et al [33] presented a weakly-supervised framework for explicit counting without supervised labels, only label whether instances belonging to the fruit class is required. Proposed an objective function to keep track of the predictions at different spatial locations of image. Roy et al [37] presented a counting approach where a semi-supervised clustering based on coloring is performed for fruit identification, and spatial characteristics based on unsupervised clustering. Xiong et al [65] used YOLOv2 for fruit detection and linear regression for fruit counting.

Tu et al [34] presented a counting framework based on detection through multiple-scale faster-RCNN which detects the lower features effectively by incorporating feature maps for regions of interest. First, high and lower-level features are extracted through a multiple scale detector, then RGB and depth detectors are trained which are finally combined through late fusion methods.

III. Proposed Approach

This section illustrates the proposed loose binary semantic segmentation-based yield estimation approach where binary segmentation extracts small patches from an image containing a fruit cluster. The high-level design of our approach pursues a traditional computer vision workflow where the counting module follows the segmentation module. Two-step computational process for yield estimation is outlined in Fig. 1. The proposed segmentation module generates the loosely segmented fruit cluster regions from RGB images on the first step. Then, responding fruit cluster regions are extracted from the input RGB image based on segmented regions. Each extracted region is forwarded as input to the counting module to obtain the individual fruit count. At the end, individual counts are summed up to get the overall prediction count against a given input image, both segmentation and counting modules are built on deep learning architectures.
For each module, task-oriented convolutional architectures are introduced, which are trained without prior knowledge about the fruit type to build a generalized yield estimation approach that can be trained only from the data. Although both modules are trained separately, they are not independent entirely since binary masks produced by the segmentation module will be used to extract the sub patches containing fruit instances from original images. These extracted sub-images are used for the training of counting modules. In two subsections below, both modules are described along with rationale behind design.

A. Segmentation

Regression counting on the whole image at once is computationally expensive when hundreds of trees are in an orchard. It requires many labelled samples that are pretty tedious to get and become extremely time-consuming when there are hundreds of fruit instances in a single image. As earlier established in [14], [15], regression-based counting achieves great results when the number of instances in images are small; however, accuracy gets compromised as the number of instances per image increases. Moreover, fruits grow in clusters, and processing the whole image is costly. So instead of processing the image as whole, counting over the non-overlapping patches containing clusters of fruit, is required. Therefore, disjoint patches of segmented fruit clusters are generated to provide the thousands of small patches for the training of the counting module.

From the design point of view, output of the segmentation module is kept loose because instead of segmenting the individual fruit instances we want to segment the clusters so that corresponding patches can be extracted. Moreover, due to a large number of fruit instances in the image, annotation of the exact ground truth is highly tedious, and becomes even more expensive as the Deep learning paradigm requires thousands of such annotated images. Since the background is almost uniform, learning for regression with loose and exact segmentation also becomes similar, and eventually, chances of involving the distinctive features from background are very low. Zhou et al [41] testifies the claim by visualizing the network that reveals saliency in the foreground. Fruit instances are very small compared to the whole image and also partially occluded; therefore, soft segmentation of fruit clusters is a suitable thing to do.

Due to the fewer number of training parameters, we have used the SegNet architecture [42], [43] for generating the loose segmentation of fruit clusters instead of deconvolutional networks with fully connected (FC) layers [63] having many more training parameters. As for cardinality of categories and nature of the domain is concerned, dealing with the loose segmentation is less complex as compared to multi-class semantic segmentation because variations in pixel intensities are restricted in single image. Additionally, main purpose is not to obtain an overall highly accurate segmentation mask, rather aim is not to miss any fruit cluster region in the image for the training of the counting module eventually. Fig. 2 illustrates the used segmentation network. The front-side convolutional substructure of the segmentation network is based on VGG architecture [26] where five 2×2 max-pooling operations are followed by convolutional and nonlinearity layers, which helps compress the feature map to 32 times before backend deconvolutional operation.

Class imbalance is very high in the fruit counting domain since the ratio of fruit cluster to background incurs a big difference. To address the class imbalance problem, weighted categorical class-entropy is involved as a loss function that allows adjusting the weights depending on the misclassification.

B. Counting

From Fig. 1, it can be seen that RGB images used for counting modules are extracted after segmenting out the fruit clusters by the segmentation module. Extraction of multiple patches from the original big image satisfies the need of a huge training dataset so that the model can learn input-output mapping. We have used the deep learning inspired counting approach to get the generalizable and robust counting solution. The combination of convolutional and pooling layers, CNN, is the deep learning approach that replicates the operational mechanism of the human vision system [44]. Input to the CNN is an image that goes through different convolutional and pooling layers and produces the representative feature map as output. Journey of the feature map between input to output layer goes from many hidden layers which consist of a stack of convolutional and pooling layers. Training of CNN goes through two stages: (1) feedforward and (2) backward propagation. During feedforward stage, loss is calculated based on the predicted output from the produced feature maps and labelled outputs. In backpropagation, gradient of loss is calculated with respect to each weight parameter, and parameters are updated for next feedforward calculations based on gradient. Two
staged processes go through many iterations and terminate when loss stops to decrease further.

Typically, CNN learns a feature map with two spatial and one channel dimensions simultaneously, increasing parameters. On the other hand, inception models ease this process and learn feature representation with fewer parameters because they work on spatial and cross-channel correlations. Although different inception models had been introduced with slight variation [45], [47], but, Inception-ResNet [48] outperformed the ImageNet dataset [49]. Influenced by this performance, we used the modified Inception-ResNet-A with proposed CNN network. Usually, fruits are extremely crowded and vary in size due to natural variation in size and image capturing position incurs the size variability; therefore, high-level semantic feature plays a crucial role compared to receptive fields. Reason to this, indulgence of modified Inception-ResNet-A enlarges the receptive field [50]. The proposed network architecture is shown in Fig. 3.

Fig. 4 illustrates the architecture of the modified inception-ResNet-A model. Last layer, having 1 x 1 convolutional, calculates 126 feature maps instead of 256 as in original Inception-ResNet [48]. The Inception layer consists of three concatenated layers, and the result is added to activation of the previous layer which passes from a rectified linear unit. After the inception layer, 3 x 3 convolutional is again applied, followed by 2 x 2, stride 2, max-pooling, which increases the accuracy when used before a fully connected layer [52]. Size of the fully connected layer is 626. Deep learning models are prone to overfitting, which can be mitigated through dropout technique [53] where we have randomly dropped the 40% connections. Instead of regression output, we have applied SoftMax with 11 outputs because the number of fruit instances in extracted regions are less than 12 which makes SoftMax suitable.

IV. Experiment

This section demonstrates the effectiveness of proposed methodology. First, we explain the used datasets for both modules. Next, training methodology for both modules along with training setup and implementation details. Lastly, evaluation and comparison with other proposed approaches are given.

A. Dataset

The dataset consists of 6 different fruits including Apple, Almond, Orange, Peach, Tomato, and Pomegranate. Sample image of each fruit type is shown in Fig 5. 4820
original images are augmented to 32,412 images, and used 80% for the training of segmentation module, while remaining are used for validation. Although images for segmentation are gathered from different datasets including Google Images, and have different sizes, they are resized to 900 x 650 pixels. Fifty images of each fruit type are used to test the system's accuracy. Annotation of dataset especially for segmentation is very tedious to obtain for huge dataset, with this reason, different augmentation techniques are used to enlarge the dataset. Data augmentation is essential to teach the network the desired invariance and robust properties, when only few training samples are available.

Figure 5: Samples for all fruit types are shown in following order: (1) Pomegranate, (2) Tomato, (3) Orange, (4) Apple, (5) Peach, and (6) Almond.

We have applied the transformation with generalization ability [54]. Commonly used transformations, such as left-right flipping, elastic deformations [55], and rotation, are applied. These transformations are also applied with the same parameters on corresponding mask images. Breakdown of the images after augmentation against each fruit type that is used for segmentation training is given in Tab. 1.

TABLE 1: TYPE WISE BREAKDOWN OF THE DATASET USED.

| Fruit type  | Original Images | After Augmentati on | Training images | Testing images |
|-------------|----------------|---------------------|-----------------|----------------|
| Apple       | 1120           | 8174                | 6529            | 1635           |
| Orange      | 950            | 5541                | 4432            | 1109           |
| Tomato      | 680            | 4902                | 3922            | 980            |
| Pomegranate | 560            | 3829                | 3064            | 765            |
| Almond      | 620            | 4783                | 3826            | 957            |
| Peach       | 890            | 5183                | 4150            | 1036           |

For the training of the counting module, 21,450 sub-images are extracted from the original images, and each patch contains 0 to 11 fruit instances. Maximum value of the assigned label was 11. We also used augmentation techniques, such as lift-right flipping, color changes, and rotation, to enlarge, but preserve the assigned label simultaneously. After augmentation, we have 89,120 sub-images divided into training, validation and test sets. 88,820 sub-images are used for training, while 17,764 are used for validation which becomes almost 20% of the training set. Finally, 300 sub-images, having 50 images of each fruit type are used to test the counting accuracy.

B. Training Setup

For segmentation, the network was trained for 2,000 epochs, with batch size of 16, over the augmented dataset. To minimize the error, SGD-momentum was used with learning rate 0.02, momentum 0.8, and weight decay 0.0002. Xavier initializer was used to initialize the parameters [56]. For the training of the counting module, Adam optimizer was involved in having learning rate and weight decay equal to 0.0001. Then, network was trained for 150,000 epochs with batch size 32. Both networks were implemented using Keras on a machine having 16 GB RAM, and Nvidia 1080Ti GPU.

C. Analysis & Comparisons

The proposed approach has been evaluated qualitatively, and counting results are also compared with results reported by different articles which worked on fruit counting specific to the focused fruit types. Loss and accuracy graphs of both segmentation and counting modules are also given.

C.1 Segmentation Module

Here, evaluation of segmentation against three metrics are given. Performance of proposed segmentation module is assessed against generating loose binary segmentation, and precision, recall, and accuracy are calculated. Values for precision (~87) and recall (~84) are seemed low because the ground-truth masks are loosely annotated; however, loosely marked contours involve almost all the fruit patches in the image. We have visually examined the test segmentation result and find almost no fruit containing region undetected by the segmentation network. Nevertheless, higher segmentation accuracy is achieved. The precision, recall and accuracy score are shown in Tab. 2 where TP is the number of true positives (correct segmentation), TN is the true negative, FP is the number of false positives (false segmentation), and FN is the number of false negatives (miss).

TABLE 2: EVALUATION MATRIX OF BINARY SEGMENTATION.

| Metric             | %   |
|--------------------|-----|
| Precision = TP / (TP+FP) | 86.69 |
| Recall = TP / (TP + FN) | 84.48 |
| Accuracy = (TP + TN) / All | 95.52 |

The segmentation module is trained for 2,000 epochs and the final training and validation accuracies are 95.5% and 87.8% respectively. From the gap of training and validation accuracy, it can be concluded that the model is slightly overfitting the training data which is a curse associated with deep learning models. The accuracy graph in Fig. 6 shows the training and validation accuracies corresponding to epochs.
Segmentation loss for training is started decreasing from 9.2 and lowered to 0.11, while validation loss is reduced from 9.35 to 0.25 after 2,000 epochs. The graph in Fig 7 shows the loss journey throughout training.

**C.2 Counting Module**

Counting module is training for 150,000 epochs with approximately 89,000 images divided into training and validation sets with 80% and 20% ratio respectively. Training and validation losses (Fig. 8) started reducing from 8.6 approximately, but training loss went to 0.07 and validation loss ended up at 0.12 at the final epoch. Validation loss went lower to training loss at some epoch but remained high most of the training time.

From Fig. 9, it could be seen that counting modules also faced overfitting as there is a difference between the training and validation accuracy where validation accuracy remained lower than training accuracy. At the end of the last epoch, training and validation accuracy ended at 97.2% and 93.9%, respectively.

**TABLE 3: ACCURACY AGAINST EACH CATEGORY**

| Fruit type | Testing Images | Actual Count | Predicted Counted | Accuracy |
|------------|----------------|--------------|-------------------|----------|
| Apple      | 50             | 492          | 473               | 96.2     |
| Orange     | 50             | 553          | 515               | 93.1     |
| Peach      | 50             | 454          | 426               | 93.8     |
| Tomato     | 50             | 604          | 557               | 92.3     |
| Almond     | 50             | 671          | 602               | 89.6     |
| Pomegranate| 50             | 380          | 344               | 90.5     |
| Overall    | 300            | 3154         | 2917              | 92.5     |

In Tab. 4, we have compared achieved result in state of art on same datasets. Häni et al [58] used the same apple dataset had achieved 94% counting accuracy, while we achieved 96.2%. Dorj et al [36] had reached 93% accuracy on orange counting, while we have gained 93.1%. We have achieved 92.3% on Tomato counting, while Rahmoomoofar et al [7] achieved 91.03%. Overall, we have performed 94.71% counting accuracy on all six fruit types.

**TABLE 4: RESULTS COMPARISON IN STATE OF ART APPROACHES**

| Method       | Apple (%) | Tomato (%) | Orange (%) | Almond (%) | Pomegranate (%) | Peach (%) |
|--------------|-----------|------------|------------|------------|-----------------|-----------|
| Roy et al [57]| 91.3      | -          | -          | -          | -               | -         |
| Häni et al [58]| 94        | -          | -          | -          | -               | -         |
| Rahmoomoofar et al [7] | 91.0 | -          | -          | -          | -               | -         |
| Malik et al [59] | -       | 91.3       | -          | -          | -               | -         |
| Dorj et al [36] | -        | -          | 93         | -          | -               | -         |
| Wang et al [62] | -        | -          | 85.6       | -          | -               | -         |
| Li et al [61] | -         | -          | 84.6       | -          | -               | -         |

**V. CONCLUSION**
Best to our knowledge, this was the first attempt to involve multiple fruit types to estimate fruit yield simultaneously. Almost all the known fruits have some common characteristics, such as circular shape, skin texture, and background which makes it a suitable fit to count the lack of big dataset for single fruit type. Through shared features, we made a single pipeline for fruit counting. Moreover, relax segmentation mitigates the unnecessary process of the image regions where fruit instances are not present. It’s very difficult to obtain the exact mask of the image, so loose segmentation allows to extract the cluster regions for further processing to count the instances. Use of SegNet makes the segmentation generation faster due to having a smaller number of parameters. As established in literature, regression method shows state-of-the-art result, and the involvement of inception-ResNet-A incurs not only higher accuracy but also lower the computation cost.

In the future, we plan to involve more fruit types and build a counting mechanism for video which will eventually converted into mobile application.

Acknowledgement
This research is supported by Artificial Intelligence & Data Analytics Lab (AIDA) CCIS Prince Sultan University, Riyadh, Saudi Arabia. The authors also would like to acknowledge the support of Prince Sultan University for paying the Article Processing Charges (APC) of this publication.

Conflicts of Interest
The authors declare that they have no conflicts of interest to report regarding the present study and all authors contributed equally scientifically.

REFERENCES
[1] P. J. Cullen, V. P. Valdravidis, B. K. Tiwari, S. Patil, P. Bourke et al., “Ozone processing for food preservation: an overview on fruit juice treatments,” Ozone: Science & Engineering, vol. 32, no. 3, pp. 166-179, 2010, doi. 10.1080/01919511003785361.
[2] L. Long, E. Shehhamer and T. Darrell, “Fully convolutional networks for semantic segmentation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3431-3440.
[3] Z. S. Pothen and S. Nuske, “Texture-based fruit detection via images using the smooth patterns on the fruit,” in IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 2016, pp. 5171-5176.
[4] A. Gongal, A. Silwal, S. Amatya, M. Karkee, Q. Zhang et al., “Apple crop-load estimation with over-the-row machine vision system,” Computers and Electronics in Agriculture, vol. 120, pp. 26-35, 2016, doi. 10.1016/j.compag.2015.10.022.
[5] N. O’Mahony, S. Campbell, A. Carvalho, S. Harapannahalli, G. V. Hernandez et al., “Deep learning vs. traditional computer vision,” in Science and Information Conference, Cham, Switzerland, 2019, pp. 128-144.
[6] S. Bargoti and J. Underwood, “Deep fruit detection in orchards,” in 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 2017, pp. 3626-3633.
[7] M. Rahnemoonfar and C. Sheppard, “Deep count: fruit counting based on deep simulated learning,” Sensors, vol. 17, no. 4, pp. 905, 2017, doi. 10.3390/s17040905.
[8] V. Lempitsky and A. Zisserman, “Learning to count objects in images,” in Advances in neural information processing systems, 2010, pp. 1324-1332.
[9] S. W. Chen, S. S. Shivakumara, S. Dcunha, J. Das, E. Okon et al., “Counting apples and oranges with deep learning: A data-driven approach,” IEEE Robotics and Automation Letters, vol. 2, no. 2, pp. 781-787, 2017, doi. 10.1109/LRA.2017.2651944.
[10] F. Lateef and Y. Ruichek, “Survey on semantic segmentation using deep learning techniques,” Neurocomputing, vol. 338, pp. 321-348, 2019, doi. 10.1016/j.neucom.2019.02.003.
[11] M. Thoma, “A survey of semantic segmentation,” arXiv preprint arXiv:1603.07285, 2016.
[12] V. Rabaud and S. Belongie, “Counting Crowded Moving Objects,” in 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06), New York, NY, USA, 2006, pp. 705-711.
[13] N. A. Abu and S. Todorovic, “Extracting Texels in 2.1D Natural Textures,” in 2007 IEEE 11th International Conference on Computer Vision, Rio de Janeiro, Brazil, 2007, pp. 1-8.
[14] V. Lempitsky and A. Zisserman, “Learning to count objects in images,” in Advances in neural information processing systems, 2010, pp. 1324-1332.
[15] P. Chattopadhyay, R. Vedantam, R. R. Selvaraju, D. Batra and D. Parikh, “Counting everyday objects in everyday scenes,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1135-1144.
[16] C. C. Loy, K. Chen, S. Gong and T. Xiang, “Crowd counting and profiling: Methodology and evaluation,” in Modeling, simulation and visual analysis of crowds, vol. 11, New York, NY, USA: Springer, pp. 347-382, 2013.
[17] P. Tu, T. Sebastian, G. Doretto, N. KrahnaStoever, J. Rittsch et al., “Unified crowd segmentation,” in European conference on computer vision, Heidelberg, Germany, 2008, pp. 691-704.
[18] V. Rabaud and S. Belongie, “Counting Crowded Moving Objects,” in 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06), New York, NY, USA, 2006, pp. 705-711.
[19] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 779-788.
[20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed et al., “Ssd: Single shot multibox detector,” in European conference on computer vision, Cham, Switzerland, 2016, pp. 21-37.
[21] 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.
[22] M. Z. Khan, S. Harous, S. U. Hassan, M. U. Ghani Khan, R. Iqbal et al., “Deep unified model for face recognition based on convolution neural network and edge computing,” IEEE Access, vol. 7, pp. 72622-72633, 2019, doi. 10.1109/ACCESS.2019.2918275.
[23] S. Jégou, M. Drozdzka, D. Vazquez, A. Romero and Y. Bengio, “The one hundred layers tiramisu: Fully convolutional dense nets for semantic segmentation,” in Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2017, pp. 11-19.
[24] Y. Song, C. A. Glasbe, G. W. Horgan, G. Polde and J. A. Dieleman et al., “Automatic fruit recognition and counting from multiple images,” Biosystems Engineering, vol. 118, pp. 203-215, 2014, doi. 10.1016/jbiosystems.2013.12.008.
[25] W. Maldonado and J. C. Barbosa, “Automatic green fruit counting in orange trees using digital images,” Computers and Electronics in Agriculture, vol. 127, pp. 572-581, 2016, doi: 10.1016/j.compag.2016.07.023.

[26] K Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.

[27] S. W. Chen, S. S. Shivakumar, S. Dcunha, J. Das, E. Okon et al., “Counting apples and oranges with deep learning: A data-driven approach,” IEEE Robotics and Automation Letters, vol. 2, no. 2, pp. 781-788, 2017, doi: 10.1109/LRA.2017.2651944.

[28] W. S. Qureshi, A. Payne, K. B Walsh, R. Linker, O. Cohen et al., “Machine vision for counting fruit on mango tree canopies,” Precision Agriculture, vol. 18, no. 2, pp. 224-244, 2017, 10.1007/s11119-016-9458-5.

[29] S. Bargoti and J. P. Underwood, “Image segmentation for fruit detection and yield estimation in apple orchards,” Journal of Field Robotics, vol. 34, no. 6, pp. 1039-1060, 2017, doi: 10.1002/rob.21699.

[30] X. Liu, S. W. Chen, S. Aditya, N. Sivakumar, S. Dcunha et al., “Robust fruit counting: Combining deep learning, tracking, and structure from motion,” in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 2018, pp. 1045-1052.

[31] J. M. Ponce, A. Aquino, B. Millan and J. M. Andújar, “Automatic counting and individual size and mass estimation of Olive-fruits through computer vision techniques,” IEEE Access, vol. 7, pp. 59451-59465, 2019, doi: 10.1109/ACCESS.2019.2915169.

[32] N. Hăni, P. Roy and V. Isler, “A comparative study of fruit detection and counting methods for yield mapping in apple orchards,” Journal of Field Robotics, vol. 37, no. 2, pp. 263-282, 2020, doi: 10.1002/rob.21902.

[33] E. Belloccchio, T. A. Ciarfuglia, G. Costante and P. Valigi, "Weakly Supervised Fruit Counting for Yield Estimation Using Spatial Consistency," in IEEE Robotics and Automation Letters, vol. 4, no. 3, pp. 2348-2355, 2019, doi: 10.1109/LRA.2019.2903260.

[34] S. Tu, J. Pang, H. Liu, N. Zhuang, Y. Chen et al., “Passion fruit detection and counting based on multiple scale faster R-CNN using RGB-D images,” Precision Agriculture, vol. 21, pp. 1072-1091, 2020, doi: 10.1007/s11119-020-09709-3.

[35] R. Linker, “A procedure for estimating the number of green mature apples in night-time orchard images using light distribution and its application to yield estimation,” Precision Agriculture, vol. 18, no. 1, pp. 59-75, 2017, doi: 10.1007/s11119-016-9467-4.

[36] U. O. Doj, M. Lee and S. S. Yun, “An yield estimation in citrus orchards via fruit detection and counting using image processing,” Computers and Electronics in Agriculture, vol. 140, pp. 109-122, 2017, doi: 10.1016/j.compag.2017.05.019.

[37] P. Roy, A. Kislay, P. A. Plonski, J. Luby and V. Isler, “Vision-based preharvest yield mapping for apple orchards.,” Computers and Electronics in Agriculture, vol. 164, pp. 104897, 2019, doi: 10.1016/j.compag.2019.104897.

[38] D. Onoro-Rubio and R. J. López-Sastre, “Towards perspective-free object counting with deep learning,” in European Conference on Computer Vision, Cham, Switzerland, 2016, pp. 615-629.

[39] A. Bearman, O. Russakovsky, V. Ferrari and L. Fei-Fei, “What’s the point: Semantic segmentation with point supervision,” in European conference on computer vision, Cham, Switzerland, 2016, pp. 549-565.

[40] I. H. Laradji, N. Rostamzadeh, P. O. Pinheiro, D. Vazquez and M. Schmidt M, “Where are the blobs: Counting by localization with point supervision,” in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 547-562.

[41] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva and A. Torralba, “Learning deep features for discriminative localization,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2921-2929.

[42] S. Aich and I. Stavness, “Leaf counting with deep convolutional and deconvolutional networks,” in Proceedings of the IEEE International Conference on Computer Vision Workshops, 2017, pp. 2080-2089.

[43] V. Badrinarayanan, A. Kendall and R. Cipolla, "SegNet: a deep convolutional encoder-decoder architecture for image segmentation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 12, pp. 2481-2495, 2017, doi: 10.1109/TPAMI.2016.2644615.

[44] F. Silvio and L. A. Alexandre, "From the human visual system to the computational models of visual attention: A survey," Artificial Intelligence Review, vol. 39, no. 1, pp.1-47, 2013, doi: 10.1007/s11626-012-0842-4.

[45] C. Szegedy, W. Liu, Y. Jia Y, P. Sermanet, S. Reed et al., “Going deeper with convolutions,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1-9.

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

[47] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the inception architecture for computer vision," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2818-2826.

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

[49] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh et al., “Imagenet large scale visual recognition challenge,” International journal of computer vision, vol. 115, no. 3, pp. 211-252, 2015, doi: 10.1007/s11263-015-0816-y.

[50] X. Chen, R. Guo, W. Luo and C. Fu, “Visual crowd counting with improved Inception-ResNet-A module,” in 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO), Kuala Lumpur, Malaysia, 2018, pp. 112-119.

[51] K. He, X. Zhang, S. Ren and J. Sun, “Deep residual learning for image recognition,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770-778.

[52] M. Lin, Q. Chen and S. Yan, “Network in network,” arXiv preprint arXiv:1312.4400, 2013.

[53] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” The journal of machine learning research, vol. 15, no. 1, pp. 1929-1958, 2014.

[54] R. Wu, S. Yan, Y. Shan, Q. Deng and G. Sun, “Deep image: Scaling up image recognition,” arXiv preprint arXiv:1501.02876, vol. 7, no. 8, 2015.

[55] P. Y. Simard, D. Steinkraus and J. C. Platt, “Best practices for convolutional neural networks applied to visual document analysis,” Icdar, vol. 3, no. 2003, 2003.

[56] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in Proceedings of the thirteenth international conference on artificial intelligence and statistics, 2010, pp. 249-256.

[57] P. Roy and V. Isle, “Vision-based apple counting and yield estimation,” in International Symposium on Experimental Robotics 2016, pp. 478-487.

[58] N. Hani, P. Roy and V. Isle, “Apple Counting using Convolutional Neural Networks,” 2018 IEEE/RSJ International
Conference on Intelligent Robots and Systems (IROS), Madrid, Spain 2018, pp. 2559-2565

[59] Z. Malik, S. Ziauddin, A. R. Shahid and A. Safi, “Detection and counting of on-tree citrus fruit for crop yield estimation,” IJACSA International Journal of Advanced Computer Science and Application, vol. 7, no. 5, 2016.

[60] M. Z. Khan, S. Hassan, H. U. Draz HU, M. U. Khan and R. Iqbal, “Detection and Identification of vehicles from high-resolution aerial images using deep learning approaches with the tuned parameters,” In Unmanned Aerial Vehicles in Smart Cities, 1st ed, Cham, Switzerland: Springer, pp. 65-83, 2020.

[61] H. Li, W. S. Lee and K. Wang, “Immature green citrus fruit detection and counting based on fast normalized cross correlation (FNCC) using natural outdoor colour images,” Precision agriculture. Vol. 17, no. 6, pp. 678-697, 2016.

[62] C. Wang, W. S. Lee, X. Zou, D. Choi, H. Gan et al., “Detection and counting of immature green citrus fruit based on the local binary patterns (lbp) feature using illumination-normalized images,” Precision agriculture. Vol. 19, no. 6, pp. 1062-1083, 2018.

[63] H. Noh, S. Hong and B. Han, “Learning deconvolution network for semantic segmentation,” in Proceedings of the IEEE international conference on computer vision, 2015, pp. 1520-1528.

[64] V. Nair and G. Hinton,”Rectified linear units improve restricted boltzmann machines,” in ICML, 2010, pp. 807-814.

[65] J. Xiong, Z. Liu, S. Chen, B. Liu, Z. Zheng et al., “Visual detection of green mangoes by an unmanned aerial vehicle in orchards based on a deep learning method,” Biosystems Engineering, vol. 194, pp. 261-272, 2020.