Improving Vision-Based Detection of Fruits in a Camouflaged Environment with Deep Neural Networks

Jinky G. Marcelo, Joel P. Ilao, and Macario O. Cordel II

1 Introduction

Detecting objects from a similarly colored environment has been an open problem in computer vision. Camouflage or color similarity is one of the major issues where the object gets occluded or merged when they are of similar color with the environment resulting in a difficult object detection [1]. One of the applications in agriculture where camouflage problem exists is at detecting fruits or vegetables in a camouflaged environment for yield estimation, e.g. the foreground contains green fruits or vegetables and the background contains green foliage.

Harvest time is critical particularly for vegetables and fruits; thus, there is a need to automatically detect and localize fruits in images for yield estimation for proper planning in labor, market and transport arrangement [2, 3]. Recent advances (e.g. [2, 3]) in computer vision has led to obtaining fruit detection and counting from images; however, this area still faces challenges because of illumination changes, scale variation, occlusion and situations of camouflage. Camouflage situations refer to the blending of the fruits to its foliage, stems and other objects of the environment which in turn makes the fruit detection a difficult task.

Initial efforts on fruit detection and counting using deep neural networks were presented in [2–6, 9]. Keresztes et al. [4] achieved an R^2 correlation of 0.96 for 45

J. G. Marcelo (✉) · J. P. Ilao · M. O. Cordel II
De La Salle University, Manila, Philippines
e-mail: jinkymarcelo@cmu.edu.ph

J. P. Ilao
e-mail: joel.ilao@dlsu.edu.ph

M. O. Cordel II
e-mail: macario.cordel@dlsu.edu.ph

J. G. Marcelo
Central Mindanao University, Maramag, Bukidnon, Philippines

J. Billingsley and P. Brett (eds.), *Mechatronics and Machine Vision in Practice 4*,
https://doi.org/10.1007/978-3-030-43703-9_5

samples of grapes and 0.85 for 150 samples of apples between the manual and automatic counting. Chen et al. [2] proposed fruit counting based on a combination of two convolutional neural networks which achieved an accuracy of 0.96 on 71 images with 7,200 oranges and 0.91 on 21 images with 1,749 apples. Rahnemoonfar and Sheppard [3] attained a 0.91 accuracy on 100 real images for automatic yield estimation using synthetic training data. Fourie et al. [5] implemented a fruit detection and localizer algorithm on 21 apple images with 442 objects resulting in 0.98 accuracy. Stein et al. [6] used a pre-trained detector and Lidar to efficiently detect, track, count and localize every piece of fruit with an error of 0.014 in a total of 522 trees with 71,609 mangoes. Sa et al. [9] developed a real-time fruit detector that can perform up to a 0.83 F1 score with a field farm dataset comprising of at most 170 samples. All the mentioned prior works have promising results; however, these systems were trained to detect objects (e.g. oranges and tomatoes) with high color difference from the leaves. This allows them to train their network with few samples. By increasing the number of samples for the training dataset, deep learning models can achieve better performance and generalization.

Though general object detection frameworks have brought remarkable breakthroughs in detecting different types of objects [7, 8], the current object detection algorithms fail in specific application scenarios like fruit detection due to occlusion and color similarity between the objects and the environment as illustrated in Fig. 1, last column. As shown in the last column of Fig. 1, the model pre-trained on MS COCO dataset [10] fails to detect any bell pepper or chili pepper. On the contrary, as illustrated in Fig. 1, third column, our work performs object detection and counting on fruits with heavy occlusion and high color similarity with its environment by

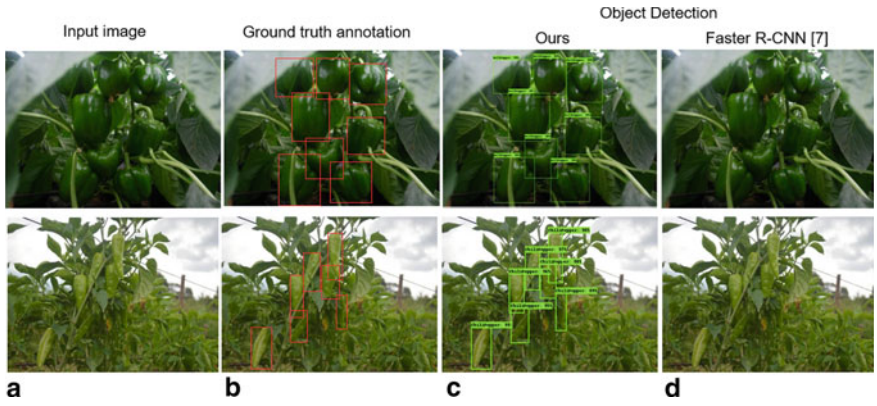

Fig. 1 **a** Input image: input images with high occlusion and color similarity of the objects with the environment (bell pepper or chili pepper). **b** Ground truth: each image contains groundtruth rectangular bounding box around each object which identifies the xmin, ymin, xmax, ymax for the object. **c** Ours: by re-purposing the region-based convolutional network to bell pepper and chili pepper images, our proposed system performs well in detecting heavily-occluded and camouflaged objects in a similarly-colored foreground and background. **d** Faster R-CNN: the result predicted by a generic object detection system

increasing the number of images and re-purposing a region-based convolutional neural network. The contributions of this paper are:

(1) we built two datasets (Bell Pepper and Chili Pepper) for sweet pepper detection. The sample images were taken from the sweet pepper field, capturing the natural settings of the object. The datasets are composed of 7700 images with 29,915 chili peppers and 3312 images with 14,548 bell peppers;
(2) we exhaustively tested various region-based convolutional neural networks (Faster R-CNN Inception, Faster R-CNN Resnet50, Faster RCNN Resnet101, R-FCN Resnet101) for a very challenging task of detecting heavily occluded objects and highly-similar color of objects with its environment. To our knowledge, this is also the first attempt that a region-based convolutional neural network is used in recognizing and localizing camouflaged objects for agricultural applications.

2 Proposed System

Figure 2 shows the overall block diagram of the proposed system. The network takes an image and outputs a set of objects with rectangular bounding boxes and the probabilities associated with it.

2.1 Image Datasets

Sweet pepper was chosen as the sample dataset since it is considered as a high-value crop in the Philippines. Two varieties of peppers (green bell pepper and green chili pepper) were used, that were high in occlusion and have high color similarity with the environment. The samples for the green bell pepper dataset were collected during the

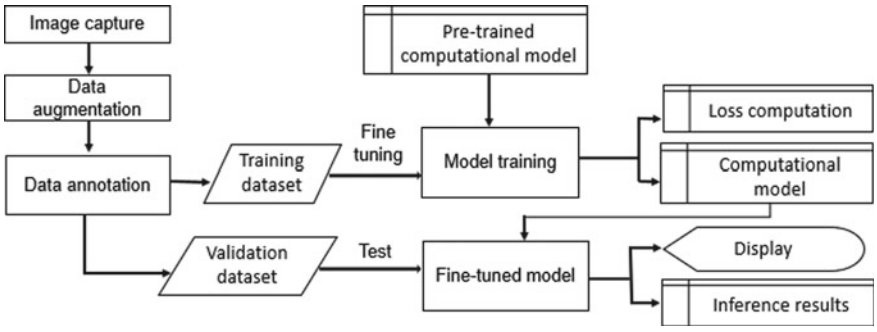

Fig. 2 Block diagram of our proposed system. The proposed system leverages the strength of region-based convolutional neural network for fruit detection in a camouflaged environment

day with no artificial lighting in a greenhouse in Impasugong, Bukidnon, Philippines. A total of 552 images were taken for green bell pepper dataset. The samples for the green chili pepper dataset were collected under an uncontrolled variability in illumination since the climatic condition of the farm location was relatively cold and humid in a cultivated farm in Lantapan, Bukidnon, Philippines. A total of 2200 images were taken for green chili pepper dataset. The images of both datasets were acquired of size 3008 $\times$ 2000 in jpeg format using a Nikon D3200 24.2 MP digital SLR camera.

2.2 Data Augmentation and Annotation

To improve the performance and generalization of deep neural networks, data augmentation was applied to the existing datasets. Data augmentation was implemented by applying horizontal flipping, rotations, shears, cropping and translation to existing datasets. These techniques ensure that the model can work under multiple angles and different orientations. The augmented images were added as additional samples to the training and test sets. After data augmentation, there was a total of 3312 images from 552 naturally captured images of bell pepper and 7700 images from 2200 naturally captured images of chili pepper.

The regions of interest for ground truth annotations were drawn and extracted using a labeler software [11]. As illustrated in the second column of Fig. 1, a rectangular box was drawn around each object in each image and these generated bounding boxes were exported into xml files in Pascal Voc format which stores the coordinates of the bounding box of regions of interest.

2.3 Implementation Details

The deep learning models were implemented using Tensorflow [14] by leveraging on transfer learning of deep vision systems. To fully explore the capability of CNNs in detecting and localizing fruit objects, four pre-trained models from MS COCO dataset [10] were fine-tuned and evaluated for bell pepper and chili pepper datasets. These pre-trained models include Faster R-CNN Inception-v2 [7, 12], Faster R-CNN Resnet50 [7, 13], Faster R-CNN Resnet101 [7, 13] and R-FCN Resnet101 [8, 13]. The pre-trained model was downloaded from Tensorflow Object Detection API [14], which is an open-source framework built on top of Tensorflow and trained on the Microsoft COCO [10] dataset.

Table 1 shows a summary of the training and testing datasets. We fine-tuned the modified object detector networks [7, 8] using our respective datasets for green chili and green bell pepper, with momentum equal to 0.9 and an initial learning rate of 0.0003. The learning rate decreases by a factor of 3×10^{-5} every 9×10^5 iterations. The learning rate was further reduced to 3×10^{-6} at 1.2×10^6 iterations. All

Table 1 Summary of training and testing datasets. The annotated bell pepper dataset and chili pepper dataset were randomly split into two datasets: 70% for training and 30% for validation

Dataset	Bell peppers images objects		Chili peppers images objects	
Training (70%)	2318	10,141	5390	20,361
Validation (30%)	994	4407	2310	9554
Total images and objects	3312	14,548	7700	29,915

the models were trained with a momentum optimizer. We trained our system using NVIDIA GeForce RTX2070. A network was trained separately for each dataset with a batch size of 1. The network was trained for 6000 epochs and 20,000 epochs for bell pepper and chili pepper dataset, respectively.

2.4 Evaluation Metrics

Object classification per category was evaluated using average precision (*AP*). As shown in Eq. (1), the *AP* score is defined as the mean precision at the set of 11 equally spaced recall values.

$$AP = \frac{1}{11} \times (AP_r(0) + AP_r(0.1) + \cdots + AP_r(1.0)) \tag{1}$$

In order to evaluate the model on the task of object localization, we determined how well the model predicted the location of the object. As shown in Eq. (2), the localization task was evaluated based on the thresholds of Intersection over Union (IoU). A threshold of 0.5 was set which means that if the IoU exceeds the threshold, then the detection is marked as correct detection. The model with the highest average precision at 0.5 IoU was selected as the model for detecting and localizing camouflaged fruits for inference to validation data.

$$IoU = \frac{Area\ of\ Overlap}{Area\ of\ Union} \tag{2}$$

3 Results

Table 2 and 3 present the result of the detection performance of four different models and their corresponding training time to bell pepper dataset and chili pepper dataset, respectively. Out of the four fine-tuned models, Faster R-CNN Resnet101 exhibited the best performance for the bell pepper dataset, yielding an AP of 0.966. For the

Table 2 Comparison of the results obtained by our proposed system employing different region-based CNNs on the bell pepper dataset. Faster R-CNN Resnet101 (Bold text) outperforms other models with AP@0.5 = 0.966

Method	Basenet	Training time	Average precision
R-FCN	Resnet101	34 min	0.963
Faster R-CNN	**Resnet101**	**33 min**	**0.966**
Faster R-CNN	Resnet50	23 min	0.964
Faster R-CNN	Inception	21 min	0.960

Table 3 Comparison of the results obtained by our proposed system employing different region-based CNNs on the chili pepper dataset. Faster R-CNN Resnet101 (Bold text) outperforms other models with AP@0.5 = 0.922

Method	Basenet	Training time	Average precision
R-FCN	Resnet101	1 h 42 min	0.904
Faster R-CNN	**Resnet101**	**1 h 35 min**	**0.922**
Faster R-CNN	Resnet50	1 h 5 min	0.917
Faster R-CNN	Inception	47 min	0.903

chili pepper dataset, Faster R-CNN Resnet101 also outperformed other models with an average precision of 0.922.

The features of Faster R-CNN Resnet101, a very deep network, were sufficient in the transfer learning for the detection of bell peppers and chili peppers. It is evident that the region proposal network contributed to higher accuracy and efficiency. From this result, it can be concluded that the model performs well in predicting the occurrence and position of the fruits in an image amidst high levels of occlusion and even those highly-similar in color between the fruits and the background.

After finishing the training, the model trained with the highest average precision was selected as the best model and exported to a single file for inference. The inference system's performance was measured using the validation datasets for bell peppers and chili peppers. Figure 3 presents the sample result of inference of our proposed system

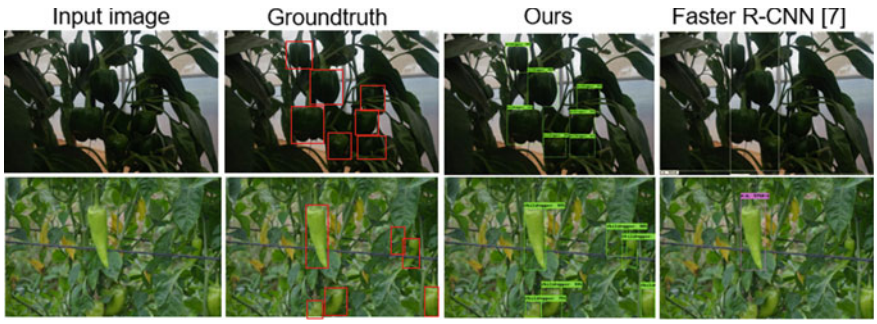

Fig. 3 Inference system on the highly-similarly colored environment. Our proposed system (3rd column) substantially performed better than the generic region-based object detector (4th column) on detecting bell peppers and chili peppers in a camouflaged environment

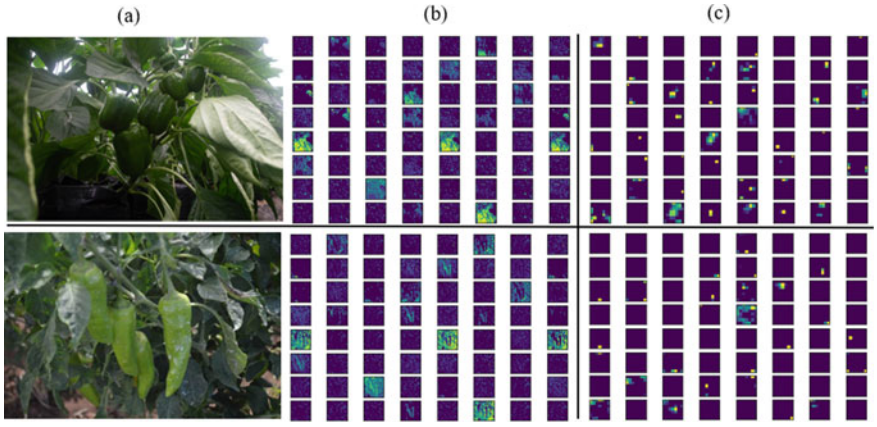

Fig. 4 Visualization of feature maps to bell pepper and chili pepper input. **a** Input images (bell pepper and chili pepper); **b** visualization image of feature maps of the first convolutional layer; **c** visualization image of feature maps of the last convolutional layer

and a pre-trained Faster R-CNN object detection system on images with high color similarity and heavy occlusion. Despite the high degree of color similarity between the fruits and the foliage, our proposed method can detect the fruits efficiently and correctly. Also, the system correctly recognized and localized the fruits even those fruits which are almost hidden due to heavy occlusion.

Figure 4 shows the visualization of feature maps after applying the filters at the first and last convolutional layer in the Resnet101 model for bell pepper and chili pepper input, respectively. It can be observed that the result of applying filters in the first convolutional layer retains most of the input image features. This means that there are many activations on the edges and textures within the image. But as the network goes deeper into the model, the feature maps become more sparse and visually less interpretable. This implies that the filters abstract the features from the image into more general concepts and convert it to the required output classification domain.

4 Conclusion and Future Work

We presented a system that automatically detects and localizes fruits from images captured from the natural settings of the fruits. By increasing the number of images and leveraging on the four pre-trained networks, the evaluation results show that the fine-tuned model on Faster R-CNN Resnet101 performed the best among all the models in detecting heavily-occluded and camouflaged fruits. It yielded an average precision of 0.92 for chili pepper and 0.96 for bell pepper. The inference shows that the fine-tuned model on Faster R-CNN detected very well to heavy-occluded and

similarly-colored foreground and background bell pepper and chili pepper images. This indicates that the trained fruit detection and counting model can be integrated into applications for precision agriculture such as automated fruit harvesting, yield estimation, and plant phenotyping.

One direction of future work is to integrate the trained fruit detector to an unmanned ground vehicle. Moreover, it can also be extended to detect other parts of a plant such as leaves, flowers, and stems which may be used for plant phenotyping and plant pathology. The proposed system can still be improved by extending its functions to more camouflaged images in agriculture and other domains.

Acknowledgements We would like to thank the farm owners and farmers for helping us in data collection. The first author acknowledges the Commission on Higher Education, in collaboration with De La Salle University and Central Mindanao University for funding the scholarship grant.

References

1. Ratthi, K., Iyshwarya. V. S., Yogameena, N. B, Menaka, K. (2017) Foreground segmentation using motion vector for camouflaged surveillance scenario. In *International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET)* (pp. 172–176).
2. Chen, S. W., Shivakumar, S. S., Dcunha, S., Das, J., Okon, E., Qu, C., et al. (2017). Counting apples and oranges with deep learning: A data-driven approach. *IEEE Robotics and Automation Letters* 2(2).
3. Rahnemoonfar, M., & Sheppard, C. (2017). Deep count: Fruit counting based on deep simulated learning. *Sensors, 17*(4), 905.
4. Keresztes, B., Abdelghafour, F., & Randriamanga, D. (2018) Real-time fruit detection using deep neural networks. In *Proceedings of the 14th International Conference on Precision Agriculture.*
5. Fourie, J., Hsiao, J., & Werner, A. (2017). Crop yield estimation using deep learning. In *7th Asian-Australasian Conference on Precision Agriculture.*
6. Stein, M., Bargoti, S., & Underwood, J. (2016). *Image based mango fruit detection.* Sensors: Localisation and Yield Estimation Using Multiple View Geometry.
7. Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems* (pp. 91–99).
8. Dai, J., Li, Y., He, K., & Sun, J. (2016). R-FCN: Object detection via region-based fully convolutional networks. In *Advances in Neural Information Processing Systems* (pp. 379–387).
9. Sa, I., Ge, Z., Dayoub, F., Upcroft, B., Perez, T., & McCool, C. (2016). Deepfruits: A fruit detection system using deep neural networks. *Sensors, 16*(8), 1222.
10. Lin T. et al. (2014) Microsoft COCO: Common objects in context. In D. Fleet, T. Pajdla, B. Schiele & T. Tuytelaars (Eds.), *Computer Vision ECCV 2014. Lecture Notes in Computer Science*, *vol*: 8693. Cham: Springer.
11. Tzutalin (2015). LabelImg. Git code. https://github.com/tzutalin/labelImg.
12. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 2818–2826).

13. He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 770–778).
14. Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., … & Ghemawat, S. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.