

Final Project Report

Traditional Computer Vision and Deep-Learning for Egg Detection

Oulaiphone Ouankhamchanh

Supervisor: Seiji Hotta, Associate Professor

**Program: STEP @Tokyo University of
Agriculture and Technology 2019-2020**

31st July 2020

Traditional Computer Vision and Deep-Learning with Egg Detection Problem: A Review

Abstract: Object detection is one of an essential computer vision technique that enables us to understand more about images and video analysis by classifying and localizing objects in a given image. Recently, there has been numerous researches and studied related to Object Detection, especially in a deep-learning-based approach. In this pilot study, by reviewing some recent related works in literature, I try to implement an egg detection program using traditional object detection methods and implement cage-free egg detection using faster region-based CNN (faster R-CNN) deep-learning technique. The result shows that both egg detection receives 99% in terms of precision. However, cage-free egg detection with deep-learning has a great potential to learn semantic, high level, and deeper features for detecting egg in various settings and environments, leading to a significant improvement for in the robotics-based system.

1. Introduction

There are several computer vision techniques in visual recognition: image recognition, object detection, semantic segmentation, and instance segmentation[1]. Particularly, object detection includes bounding boxes for each object in a given image that has a class label associate with each bounding box, and a probability score. In recent years, object detection using a deep-learning technique for agriculture has been widely studied: detecting pest in crop field[2], cage-free floor egg detection[3], fast object detection in pastoral landscapes[4], etc. Due to its ability to learn semantic features, high-accuracy, end-to-end learning, and preprocessing power, solving some problems existing in traditional architectures[5], deep learning-based object detection has been a robust instrument to solve many real-world problems. In egg industries, many egg producers have faced various problems in manually collecting eggs, time-consuming, and laborious [3]. Developing an automatic egg detector could help them collecting eggs with less time, receiving high-quality eggs, and reduce the burden of workers.

In this study, I reviewed a related work paper in the section two and The experimental implementation of egg detection with the traditional computer vision technique and the Faster R-CNN deep-learning technique with the comparison between two models are included in section three. Lastly, section 4 is the conclusion.

2. Review of literature

The traditional object detection pipeline was divided into three primary steps: informative region selection, feature extraction, and classification[5]. In the **informative region selection** stage, the idea is to scan the whole image with multi-scale sliding windows to search for locations in an image that may consist of objects known as a region of interest (ROI)[5][1]. For the second step: **feature extraction** is to extract visual feature vector known as feature descriptors such as SIFT(Scale Invariant Feature Transform), Haar, HOG (Histogram of Gradients) to recognize different objects[5]. The last step: **classification**, a classifier needed to learn to assign labels for all target objects[1]; SVM (support vector machines), cascade learning, AdaBoost, DPM (Deformation Part Based) were used in this step.

Traditional egg detection with computer vision: the program was built on handcrafted features using several computer vision algorithms to receive 0.99 in precision on detecting and counting eggs from a conveyor in an egg factory. A technique includes: perform

a BGR-HSV conversion, split into channels, then apply morphological closing, take a distance transform, create a template, perform template matching, find the local maxima of the resulting image, and finally find contours[6]. A developer used Raspberry PI for hardware, Java as a programming language, OpenCV library, and PostgreSQL, Spring JDBC, Liquibase as a database. Even though the program received 0.99 in precision, there were several downsides in terms of speed. The requirement of skillful computer vision engineering to manually handcraft features made it difficult to learn semantic information in complex contexts.

To solve the problems that exist in traditional object detection techniques, object detection with a deep-learning-based technique was used and received a remarkable performance[7]. The deep-learning-based technique was inspired by neurons inside the human brain that allow computers to learn and understand various problems in the real world. There are two major categories in object detectors with deep learning: (1) two-stage detectors includes : R-CNN[8], SPP-net[9], Fast R-CNN[10] , Faster R-CNN[7] , R-FCN[11] , FPN[12] and Mask R-CNN[13]; (2) one-stage detectors includes: YOLO[14], SSD[15], YOLOv2 [16],[1], etc. Due to a deeper feature vector, two-stage detectors have appeared to be a state of the art of accuracy in many benchmark datasets such as COCO datasets[17].

The cage-free floor egg detection: three of convolution neural networks (CNNs): SSD, Faster R-CNN, R-FCN were used to evaluate performance to choose the optimal one as state of the art [3]. Among the three CNNs, Faster R-CNN detector is a robust detector with the highest recall ($98.4 \pm 0.4\%$), accuracy ($98.1 \pm 0.3\%$), precision ($99.7 \pm 0.2\%$) and processing speed ($201.5 \pm 2.3 \text{ ms. } \text{image}^{-1}$) [3]. However, the performance of Faster R-CNN for detecting brown eggs at the 1 lux light intensity was not as effective but could be improved by putting more light in dark areas and installing blowers to blow feathers on top of floor eggs[3]. For evaluating the performance of detectors: precision, recall, accuracy, RMSE, and processing speed were averaged, then compared to determine the optimal one and the intersection over union (IoU) for each bounding box was computed to make sure that egg had been correctly detected[3].

The state-of-the-art floor egg detector, Faster R-CNN, is an end-to-end approach that uses a unified neural network for object detection. It mainly consists of a region proposal network (RPN) to generate a fixed set of regions, as in Figure 1.

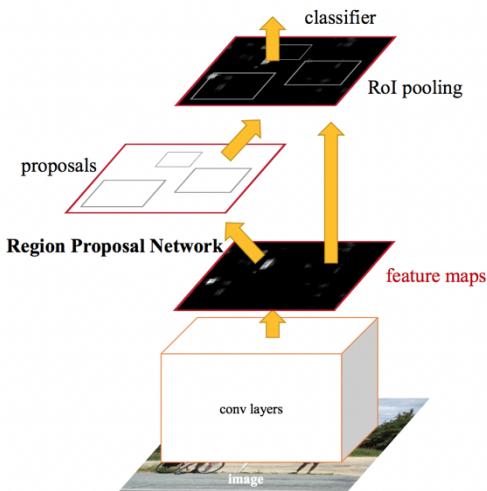


Figure 1: The faster R-CNN architecture Girshick et al. [7]

3. Experimental part

3.1 Objectives

To gain a deeper understanding of traditional computer vision and state of the art known as the Faster R-CNN deep-learning technique with object detection-based, this pilot study is to:

- Implement an egg detection model using both techniques
- Compare the performance between these two models.

3.2 Procedure

The experiments include two main parts: egg detection using traditional computer vision and egg detection using the deep-learning technique with the Faster R-CNN detector model.

Hardware: MacBook Pro 13, Processor: 2.5 GHz Dual-Core Intel Core i7

3.2.1. Traditional egg detection

Python version 3.6.10 as a programming language with OpenCV library was used, as well as an open-source web application: Jupyter Notebook and PyQt5 as a graphic user interface.

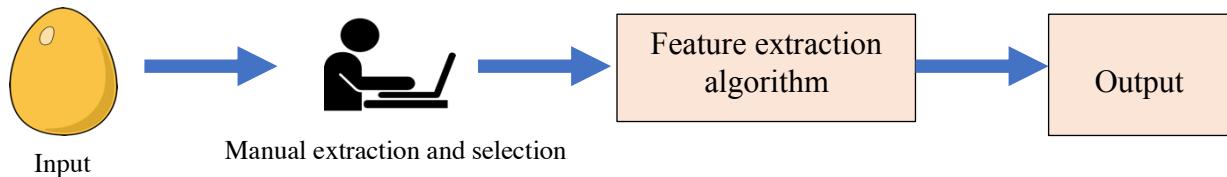


Figure 2: Workflow of traditional egg detection

3.2.2. The Faster R-CNN egg detection

Python version 3.6.10 was used as a programming language. Then downloaded other libraries and accessories to create a virtual environment includes Pillow, Matplotlib, OpenCV, TensorFlow API, Pandas, Lxml, LabelImg.

There are several steps to implementation. First, I prepared the datasets by taking 191 egg images with various camera angle setting, also download from Google. The Faster R-CNN model requires a huge amount of data for training to receive high performance and reduce the overfitting of data, so I used the data augmentation technique to increase the datasets from 191 images to 1066 images. Second step: split datasets into two folders: 1013 images for train dataset and 53 images for the test dataset. After that, labeled eggs and created .xml(XML) files using LabelImg tool to label eggs in images with bounding boxes. The LabelImg can save labeled egg as XML files in Pascal Visual Object Class format. Includes: file name, file path, image size, object identification, pixel coordinates () of bounding boxes. Third step: generating TFRecords that can be served as input data for training of the object detector. Fourth step: install CNN pre-trained object detector: The Faster R-CNN: "faster-rcnn-inception_v2_coco_2018_01_28" was downloaded from TensorFlow Detection model Zoo. Fifth step: created a label map, the label map is an id to a name. Here I used only one id name: egg after that creating a configuration for training the Faster R-CNN model. To see how well the model performs, I used TensorBoard, a TensorFlow's visualization toolkit, to see a report of the model. The Faster R-CNN model continued training until it reached 200,000 iterations, then it stopped training, and the developed detector was saved as an inference graph and output as .pb files. Finally, .pb files were used for testing the object detector.

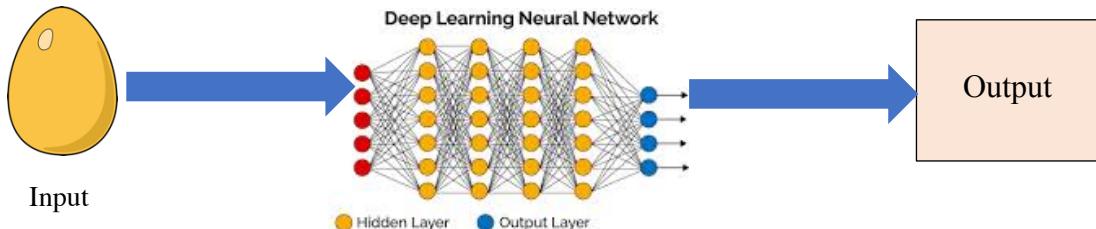


Figure 2.b Workflow of egg detection using deep-learning

3.3 Result

The implementation of egg detection shows that we can receive 99 % precision of the detection among 130 eggs by using the traditional computer vision approach, *as shown in Figure 3*. In the meantime, Figure 4 shows the sample when the Faster R-CNN model is implementing egg detection. The detection implemented in the Faster R-CNN model shows the same result with a traditional computer vision at 99 % precision at Global step = 5853. In the Faster R-CNN model, a loss function quantifies how well a given predictor is at classifying and localizing the input data points in a dataset, *as shown in Figure 5*. The lower loss means higher detection accuracy.

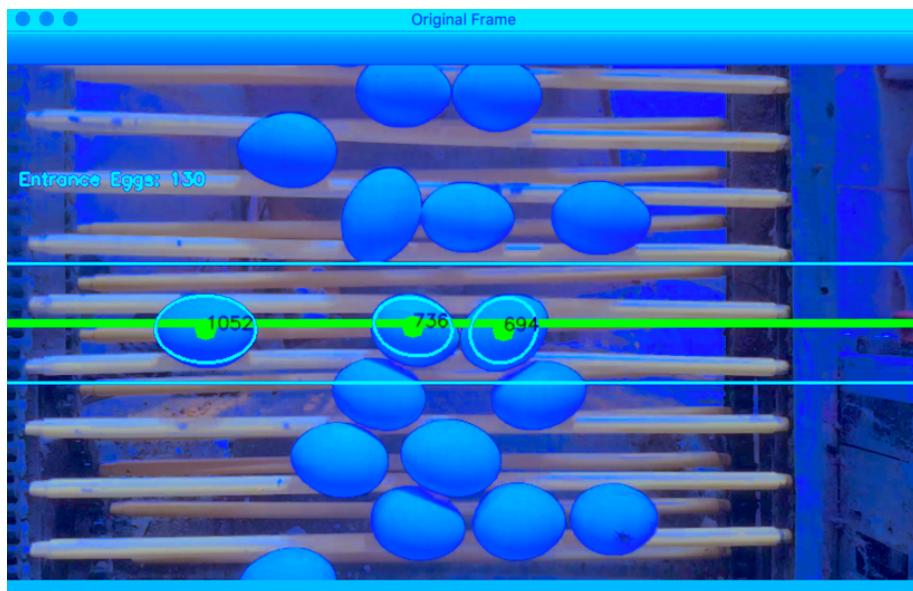


Figure 3: A part of a video file for detecting eggs and counting eggs using computer vision, inspired by https://github.com/ekapujiw2002/EggDetection/blob/master/20180910_144521.mp4

From the pilot study, we could see that the Faster R-CNN model is an end-to-end learning approach where the machine can learn to detect objects from given datasets without relying on manually feature extraction. The neural networks inside the model can discover the underlying patterns in images and figures the most useful features according to each specific class of object. Due to its ability to learn semantic features, it requires a large amount of data, many iterations, and a long time to train to reach a robust performance. In other words, the traditional model has a limited learning feature but can perform well with a few data as a summary of the comparison between the two models in Table 1.

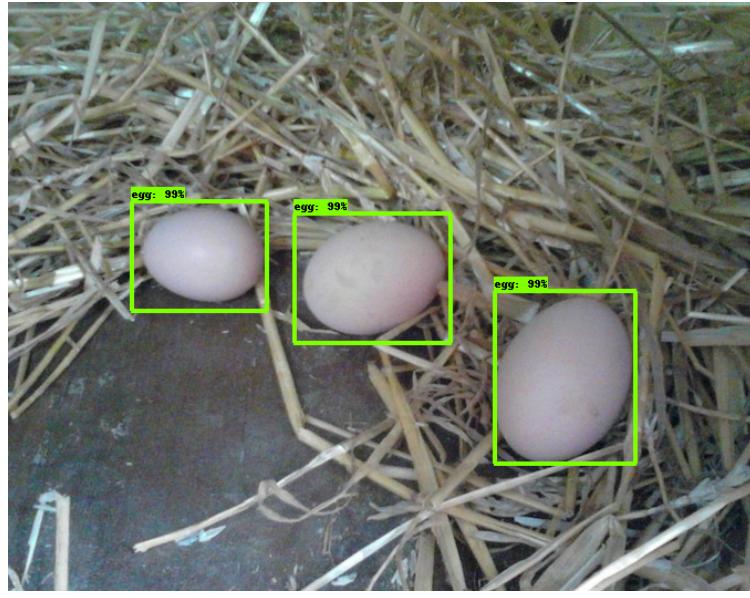


Figure 4: Detected eggs using the Faster R-CNN model.
The green bounding box with label and probability number indicates eggs in an image.

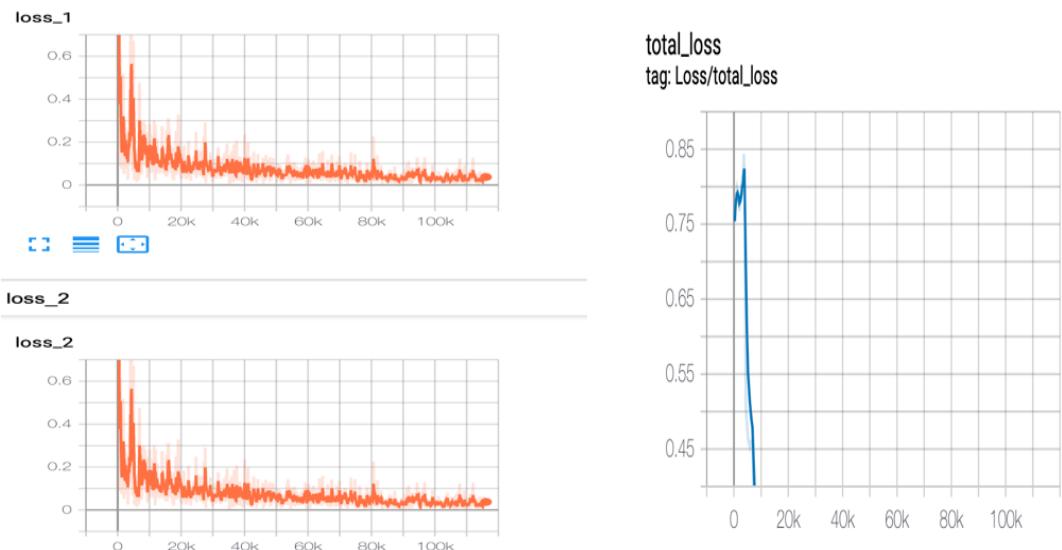


Figure 5. The loss at 200,000 iterations of the Faster R-CNN during training

4. Conclusion

Traditional egg detection program used a fundamental computer vision technique. It is suitable for simplified work that does not require a huge amount of data. However, it requires a Computer Vision engineer to extract features manually.

For the Faster R-CNN deep-learning-based approach, is end-to-end learning that requires massive data for training and learning. The more suitable data we have, the more accuracy for the Faster R-CNN model to learn. Moreover, the Faster R-CNN egg detection can learn deeper features with various settings and environments, putting a significant step toward an effective automatic detection system for further development.

There is a drawback from the deep-learning model, such as time-consuming during training and require a robust computer to train the model.

Traditional and deep-learning techniques have significant potential on their own, depending on a task we choose to perform. However, there could be a possible way to combine the strength of these two techniques and create a useful model that can help to solve a problem in the real world. This idea could be practice for further study.

Table 1: Compare special features between the Faster R-CNN egg detection model and the Traditional egg detection:

Faster R-CNN	Traditional Computer Vision
<ul style="list-style-type: none">- End-to-end learning.- Being able to learn semantic and high-level features.- Need a large amount of data to achieve high performance.- Training time consuming.- Need a powerful computer.	<ul style="list-style-type: none">- Requirement of skillful computer vision engineering.- Built on handcrafted features.- Unable to learn deeper features.- Use to solve a simplified and specific task.

References:

- [1] X. Wu, D. Sahoo, and S. C. H. Hoi, "Recent advances in deep learning for object detection," *Neurocomputing*, vol. 396, pp. 39–64, 2020, doi: 10.1016/j.neucom.2020.01.085.
- [2] W. Li, P. Chen, B. Wang, and C. Xie, "Automatic Localization and Count of Agricultural Crop Pests Based on an Improved Deep Learning Pipeline," *Sci. Rep.*, vol. 9, no. 1, pp. 1–11, 2019, doi: 10.1038/s41598-019-43171-0.
- [3] G. Li, Y. Xu, Y. Zhao, Q. Du, and Y. Huang, "Evaluating convolutional neural networks for cage-free floor egg detection," *Sensors (Switzerland)*, vol. 20, no. 2, pp. 1–17, 2020, doi: 10.3390/s20020332.
- [4] E. J. Sadgrove, G. Falzon, D. Miron, and D. Lamb, "Fast object detection in pastoral landscapes using a Colour Feature Extreme Learning Machine," *Comput. Electron. Agric.*, vol. 139, pp. 204–212, 2017, doi: 10.1016/j.compag.2017.05.017.
- [5] Z. Q. Zhao, P. Zheng, S. T. Xu, and X. Wu, "Object Detection with Deep Learning: A Review," *IEEE Trans. Neural Networks Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, 2019, doi: 10.1109/TNNLS.2018.2876865.
- [6] Ivan Ursul, "How we wrote chicken egg counter on a Raspberry PI | ivanursul." <https://ivanursul.com/counting-eggs-in-opencv> (accessed 29th July, 2020).
- [7] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017, doi: 10.1109/TPAMI.2016.2577031.
- [8] R. (RCNN)Girshick, J. Donahue, T. Darrell, J. Malik, U. C. Berkeley, and J. Malik, "1043.0690," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, p. 5000, 2014, doi: 10.1109/CVPR.2014.81.
- [9] K. He, X. Zhang, S. Ren, and J. Sun, "SPPNet 【PAMI】 .pdf," vol. 37, no. 9, pp. 1904–1916, 2015.
- [10] R. Girshick, "Fast R-CNN," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2015 International Conference on Computer Vision, ICCV 2015, pp. 1440–1448, 2015, doi: 10.1109/ICCV.2015.169.
- [11] Y. Li, K. He, J. Sun, and others, "R-fcn: Object detection via region-based fully convolutional networks," *Adv. Neural Inf. Process. Syst.*, no. Nips, pp. 379–387, 2016, [Online]. Available: <http://papers.nips.cc/paper/6465-r-fcn-object-detection-via-region-based-fully-convolutional-networks.pdf>.
- [12] X. Li, T. Lai, S. Wang, Q. Chen, C. Yang, and R. Chen, "Weighted feature pyramid networks for object detection," *Proc. - 2019 IEEE Intl Conf Parallel Distrib. Process. with Appl. Big Data Cloud Comput. Sustain. Comput. Commun. Soc. Comput. Networking, ISPA/BDCloud/SustainCom/SocialCom 2019*, pp. 1500–1504, 2019, doi: 10.1109/ISPA-BDCloud-SustainCom-SocialCom48970.2019.00217.
- [13] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, 2020, doi: 10.1109/TPAMI.2018.2844175.
- [14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-December, pp. 779–788, 2016, doi: 10.1109/CVPR.2016.91.
- [15] W. Liu *et al.*, "SSD: Single shot multibox detector," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9905

- LNCS, pp. 21–37, 2016, doi: 10.1007/978-3-319-46448-0_2.
- [16] J. Redmon and A. Farhadi, “YOLO9000: Better, faster, stronger,” *Proc. - 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, vol. 2017-January, pp. 6517–6525, 2017, doi: 10.1109/CVPR.2017.690.
- [17] M. Tan, R. Pang, and Q. V. Le, “EfficientDet: Scalable and Efficient Object Detection,” 2019, [Online]. Available: <http://arxiv.org/abs/1911.09070>.