New Zealand Shellfish Detection, Recognition and Counting: A Deep Learning Approach on Mobile Devices

Quan Nguyen, Minh Nguyen, Bowen Sun, and Huy Le

Auckland University of Technology, 55 Wellesley St E, Auckland 1010, New Zealand
minh.nguyen@aut.ac.nz, qyq0379@autuni.ac.nz

Abstract. New Zealand maintains excessive effort to organise the sustainable development of its marine resources, wildlife, and ecological environment. New Zealand has stringent rules to control fishing and to protect the continued growth of marine inhabitants. Fishing inspections, such as identifying and counting shellfish, are part of the daily routine of many New Zealand Fisheries officers. It is however considered labour-intensive and time-consuming work. This project, thus, develops a touch-less shellfish detection and counting web/mobile application on handheld devices using Mask R-CNN to assist New Zealand Fisheries officers in recognising and totalling shellfish automatically and accurately. New Zealand shellfish species are different from other places in the World. Thus, this study firstly investigates the best deep learning model to use for New Zealand shellfish recognition and detection. Selected shellfish dataset is collected from a local fish market in Auckland and trained by using the chosen artificial neural network. At last, a portable system is built to support Fisheries officers to count shellfish quickly and accurately. At this current stage, a web-based application has been successfully deployed at a local server (cvreact.aut.ac.nz) in which users can upload target objects to get results related to three major shellfish species including cockle, tuatua, and mussel. In the near future, this proposed model is scaled up to recognise more species to cover the popular shellfish species in New Zealand, thus benefiting the aquaculture as well.

Keywords: Object detection · Deep learning · Shellfish · Mask R-CNN · ResNet

1 Introduction and Backgrounds

New Zealand is well-known for its beautiful natural scenery and abundant natural resources. New Zealand has managed to organise the sustainable development of marine resources, wildlife and ecological environment. New Zealand, therefore, has legislated out for fishing different sea creatures such as finfish and shellfish in different regions and seasons. For instance, each fisherman can only gather 50 cockles per day in Auckland Coromandel area [1]. New Zealand Fisheries officers...
work hard to ensure the sustainability of New Zealand’s fishery. Checking and counting shellfish manually is one of the daily work of New Zealand Fisheries officers; however, it is very time-consuming (Fig. 1).

Recently, actual applications using deep neural networks (DNNs) technologies have been widely applied and achieved remarkable successes in both of the research and industrial fields. Object detection, classification, and recognition are some of the most frequently used techniques that efficiently extract specific objects from images or video stream by pre-trained models. Artificial intelligence (AI) applications have taken over the roles of Fisheries officers in the time-consuming. It is useful to have a portable system to support Fisheries officers to recognise and count shellfish quickly, thereby reducing the workload of Fisheries officers. This research aims to develop an application to recognise and count shellfish based on input images or videos from a handheld camera. Its approach also figures out whether shellfish species being caught meet New Zealand different fishing rules [1].

Visual Object counting (VOC) as a useful application of object recognition has been widely applied in many areas in the real-world to measure the number of target objects with the input from images or videos. For instance, using cell counting system to count cells in microscopic images provides a faster and affordable disease diagnose solutions [11]. Image processing techniques, such as neural network, Hough transform, clustering, and shape matching, are adopted to recognise and detect patterns or objects [9]. However, big concerns when using VOC are the frequent overlap between objects, occlusion, and complex background environment. The object counting process in image processing can be performed by object detection, regression, and segmentation techniques. All of these approaches require a machine learning process on labelled data to build a detection model, thereby predicting the number of target objects in an image [13]. Global regression-based VOC (GR-VOC) and density estimation based VOC (DE-VOC) are two significant techniques using supervised approaches [17]. DE-VOC method counts object instances by a density function from dense local features of the image [5]. Both these two techniques can perform efficiently with sufficient training images. Convolutional neural networks (CNN) and hardware-accelerated optimisation can improve the performance of regression-based counting approach significantly [11]. An innovative framework to count objects
without any preliminary training step was proposed and can count multiple object types [13]. To minimize the data preparation costs introduced by labelling training data of regression-based approaches, an unsupervised approach was introduced to count objects without object recognition [7].

Object detection has been a significant part of image processing and CV fields. It is a technique that localises and classifies an object in an input image by predicting the bounding box location that contains the object [4]. Geometry-based and appearance-based are two ways of object representation [14]. Object detection can be achieved using traditional machine vision approaches such as SVM (Support Vector Machines). The traditional method uses a sliding window to generate candidate regions on an input image, then extracts features and classifies the regions using a trained classifier [20]. Time-complexity is one of the main drawbacks of the traditional method. Compared to traditional methods, deep learning algorithms, such as CNN, are able to accept raw data as input and automatically learn features [10].

Object detection and counting techniques have been successfully implemented in agriculture fields. A model to detect and count plant seedling in the field using Faster R-CNN with Inception ResNet v2 was implemented [6]. Their model successfully achieved an F1 score of 0.969 at the IoU threshold of 0.5. Another model implemented a real-time corn kernel detection and counting to help farmers to estimate corn harvest and make marketing decisions [8]. The results indicated that the CNN model has higher accuracy (0.947) than HOG+SVM, since it can extract features automatically from an input. A fruit detection model was developed for strawberry harvesting check based on Mask R-CNN [18]. From their experiment results, the average precision rate is 95.78% out of 100 test images. A comparison of fruit detection and counting performance between Faster R-CNN with inception V2 and SSD with MobileNet on a fruit dataset was conducted [15]. Experiments have shown that Faster R-CNN has higher performance (94%) than SSD (90%). Furthermore, they pointed out that the detection performance of CNN architectures is relied on experimental dataset quality.

Apart from the agriculture fields, the use of deep learning methods for object detection and counting has been used in other areas in daily human life. A real-time vehicle counting system using SSD algorithm to monitor traffic was designed [2]. The SSD based network was replaced by ResNet-34, which is more accurate than the original based network (VGG-16). From the experiment results that the vehicle detection accuracy of the system can reach to 99.3% and the classification accuracy can achieve to 98.9%. A model to detect and count threatening objects from images using TensorFlow object detection API was built [12]. Their model was implemented by Faster R-CNN algorithm, and it is able to detect and classify two classes of threatening objects: knife and gun. The experiment results show relatively good accuracy and counting results.

This paper proposes the portable model based on Mask R-CNN, which recognises and counts target objects from an image or video to figure out a legal shellfish grabbing. The target objects are uploaded to a processing server, and
then its results are popped up on the officer’s handheld device. The notices or recommendations are also presented to make proper decisions.

2 Conceptualisation of Implementation Method

2.1 Overall System Design

A web application is easy to be used on a wide variety of operating systems of laptops and mobile devices. To facilitate the use of the detection system by New Zealand fisheries officers both in their offices and at beaches, this system is designed as a web application that allows users to upload images or videos to check the shellfish species and quantity. The entire recognition system with the shellfish detection model is hosted on a processing server. As shown in Fig. 2, the system is deployed on a processing server to detect and count the target shellfish objects from an uploaded image. The detection result that includes labelled shellfish and count numbers of each shellfish species is finally returned to the end-user.

![Fig. 2. Overall system design](image)

2.2 Dataset Preparation

Cockle, tuatua are major shellfish species that New Zealanders can gather from beaches both in South Island and North Island. Cockles are plump, round shells with fine ridges, and they are widespread in New Zealand harbours. Tuatua is endemic to New Zealand which can be found when low tide and has a more irregular shell shape. Therefore, this study selects cockle and tuatua as target shellfish species to conduct the shellfish detection and counting task. To extend the application of this detection model, mussel, another common and popular shellfish, is included in this research.

The shellfish used in this study were collected randomly. The appearance of the same shellfish species varies among different regions. Therefore, the cockles used in this study are gathered initially from both South Island and North Island New Zealand, as shown in Fig. 3. Due to the resource constraints, the tuatua and mussel are originally gathered from North Island New Zealand.
In this research study, 700 images that include 200 images of each shellfish species and 100 images of combined shellfish species are selected for the further detection experiment. The sample images of cockle, tuatua, and mussel are shown in Fig. 4. Figure 5 describes the data pre-processing process that includes data
annotation, data format conversion. Data augmentation is an essential technique for expanding dataset in deep learning tasks.

**Fig. 5. Data pre-processing process**

### 2.3 Detection Model Design

Since the purpose of this shellfish detection system is to accurately recognise and count shellfish for New Zealand fishery control, this paper focuses more on detection accuracy rather than detection speed. Faster R-CNN and Mask R-CNN have better performance than YOLO and SSD [19]. Furthermore, the results of general object detection experiments on the MS COCO dataset shows that Mask R-CNN has higher AP than other deep learning object detection methods. Due to the advantage in accuracy, Faster R-CNN and Mask R-CNN are considered for this purpose.

Bearing in mind the model comparison and results of previous researches above, this paper adopts Mask R-CNN to train the shellfish detection model. Figure 6 demonstrates the structure design of the Mask R-CNN shellfish detection model that is used in this paper. This paper uses ResNet101 and FPN as the backbone network to extract feature maps from an input image.

**Fig. 6. Model structure of Mask R-CNN shellfish detector**

### 2.4 Web Application Deployment Design

Once the detection model is trained, the final model and a user interface will be deployed as a web application. This application is written in Python. The web user interface is developed using Streamlit, which is an app framework designed for data analysis and machine learning. Moreover, the application deployment uses Docker which is a platform to help automatic software delivery using containers. Docker builds a lightweight image which can be deployed on various
cloud platforms such as AWS and Azure [3]. Figure 7 demonstrates the workflow of the system deployment. First, the system settings and package requirements are configured within a Docker file. Then the container image that contains the system libraries and application code is pushed and deployed on a cloud server. Finally, the cloud server will host the application and provide a website for users.

The system provides the users with simple GUI application, to keep the page straightforward so that users with different knowledge backgrounds can easily use the application. Users can upload and preview an image by clicking ‘browse files’ on the main page. Once the image is uploaded, users can use this application to detect and view the predicted results by clicking ‘Detect and Count’ buttons on the web page.

3 Experiment

3.1 Data Pre-processing

Since this research aims to conduct the object detection task, the target objects on each image need to be labelled for training purpose. This project uses Labelme, which is an easy-to-use image annotation tool to label the shellfish objects on each image. As shown in Fig. 8, all target objects in the image are selected and labelled within a polygon that covers on each target object. Labelme generates a JSON file for each image to store the image information and annotations.

This project trains the detection model based on COCO pre-trained weight and uses COCO evaluation metrics. The shellfish data should be converted to COCO format dataset. A Python script randomly allocates 80% of images in the shellfish dataset to train dataset and the other 20% to the validation dataset. The detection model is trained with the training dataset and evaluated with the validation dataset. The size of training dataset for deep learning is critical for the performance of a model. The more data are involved, the more robust the model is, so data augmentation techniques that automatically enlarge the dataset with annotations are very crucial for dealing with small dataset in a CNN model training process. Data augmentation includes colour operations and
geometric operations. Flipping, cropping, and rotation are most used augmentation techniques in object detection tasks [16]. Data augmentation is able to increase detection accuracy over +3.2 mAP on COCO dataset [21] (Figs. 9 and 10).

3.2 Model Implementation

Due to the limitation of hardware, the detection model training process is performed on Google Colaboratory. Additionally, this research uses TensorBoard to visualise the summary output of the model training process, such as AP and loss. Transfer learning is also adopted to train the detector. It is faster than training a model from scratch and can help the model achieve better performance in training on a small dataset.
Fig. 10. Data distribution of instances among categories

In the experiment, the Mask R-CNN model is trained with a maximum of 10000 iterations. The values of other parameters are set to 0.9 momenta, 0.0001 weight decay, and batch size 2. The based model is trained with a learning rate of 0.02 on 8 GPUs. Since this research uses 1 GPU to train detection model, the learning rate should be 0.0025 in theory. As shown in Table 1, the inference results of Mask R-CNN ResNet101+FPN model with learning rate 0.001 has the highest bounding box prediction AP (86.826%) and segmentation AP (87.191%). The AP in the table is averaged over three categories on 10 IoU thresholds from 0.5 to 0.95. Additionally, the model with learning rate 0.001 has an acceptable inference speed which is around 0.1s per image. Therefore, the learning rate is set to 0.001 to train the detection model.

Table 1. Mask R-CNN ResNet101+FPN with different learning rates

| Learning rates | bbox AP(%) | segm AP(%) | Train time (s/iter) | Inference time |
|----------------|------------|------------|---------------------|---------------|
| 0.01           | 84.480     | 85.860     | 0.6579              | 0.1516        |
| 0.005          | 84.974     | 86.965     | 0.3783              | 0.1043        |
| 0.0025         | 85.151     | 86.200     | 0.3782              | 0.1043        |
| 0.001          | 86.826     | 87.191     | 0.3783              | 0.1038        |

The experiment evaluates the model on the validation dataset using COCO evaluation metrics during the training process. The evaluation performs every 500 iterations. Additionally, the experiment metrics like loss and average
precision are visualised and monitored within TensorBoard during the training process. Finally, an inference is performed on the validation dataset using the final trained model. To evaluate the performance of the final detection model, models of Mask R-CNN with ResNet50+FPN, Mask R-CNN with ResNeXt101+FPN, and Faster R-CNN with ResNet101+FPN have experimented with the same settings as a comparison.

4 Analysis and Discussion

Since the main propose of this research is to recognize and locate the target shellfish objects. This research mainly analyses and discusses the results of the bounding box prediction accuracy. As Shown in the table, the average precision (AP) is averaged over 3 categories at 10 IoU thresholds which are 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95. The AP of the bounding box prediction is 87.414%, while the AP of segmentation prediction is 87.032%. Furthermore, AP50, which is similar to PASCAL VOC evaluation metrics, is calculated at 0.5 IoU threshold. The AP at 0.5 IoU threshold is 99.592% for both bounding box and segmentation prediction. The prediction results of AP50 show a good performance of the final prediction model. The detection results for each shellfish category that the model has better detection capabilities for mussel and tuatua, and the AP of bounding box prediction can reach 91.084% and 90.576% respectively.

Because the samples of cockle are collected from both South Island and North Island of New Zealand, their shapes and colors are different. Therefore, the average precision is a bit lower than that of mussel and tuatua, which is 80.584% for bounding box prediction and 81.142 for segmentation prediction. The training speed of the detection model is 0.3776 s for each iteration. The inference time is 0.1006 s for each image on one device. The testing results on random validation images show good detection performance, and the classification score for each object can reach over 95%.

Figure 11a shows the accuracy of classification during the training process. Accuracy indicates the percentage of the correctly predicted results in the total predictions. After around 500 steps, the accuracy of object classification remains at between 98% to 99%.

During the training process, a COCO model evaluation inferences on the validation dataset every 500 iterations over 1050 images. Figure 11b demonstrates the total loss of final Mask R-CNN model on training and validation dataset respectively. The total loss is obtained by the loss of classification, bounding box, and mask. As shown in Fig. 11b, the total loss stops dramatically decreasing and shows a smooth delay after 500 iterations. The total loss of both training and validation shows a slightly downward trend as expected.
4.1 Model Comparison

Compared to the network with 101 layers, due to the smaller depth of ResNet50, the model training should be faster than the network with more layers. ResNet50 model is faster than other models both on train time and inference time. The training time of ResNet50 model can reach 0.289 s per iteration and the inference time is 0.073 s per image. In theory, a model with ResNeXt backbone should have better performance than ResNet. According to [19], ResNeXt is powerful and Mask R-CNN with ResNeXt101+FPN has slightly better prediction performance than ResNet101+FPN on MS COCO Test dataset. In this experiment, ResNeXt101 does not perform well on predicting the bounding box, the AP of IoU threshold from 0.5 to 0.95 is 87.417%, which is similar to ResNet101 (87.414%). This may be caused by the fact that shellfish dataset and training iterations are small.

Both the training time (0.8224 s/iter) and inference time (0.1916 s/img) of ResNeXt101 are doubled compared to those of ResNet101 (train time: 0.3776 s/iter, inference time: 0.1006 s/img). When comparing with the results, ResNet101 is more suitable for shellfish detection than ResNeXt101. Because the prediction performance of ResNet101 model is similar to ResNeXt101, but the inference time, which is important for user experience, is twice as fast.

As shown in the detection results of the final Mask R-CNN model and Faster R-CNN model, the final Mask R-CNN model performs better than Faster R-CNN model. The AP of Faster R-CNN model is 86.273% which is lower than the final Mask R-CNN model (87.414%). The AP of Faster R-CNN model at 0.5 IoU threshold, which is 99.537%, is similar to Mask R-CNN model (99.592%). The results also point out that Faster R-CNN is slower both on train and inference speed than Mask R-CNN. The training speed and inference speed of Faster R-CNN are 0.6198 s per iteration and 0.1553 s per image respectively.

4.2 Results Demonstration

The evaluation results that are computed from the validation dataset is not enough to determine the model performance. Because the dataset used in the
experiments is small, overfitting problem may be introduced by training. The final model should be tested using images that are not in the experimental dataset. Figure 12 shows the test result on a new image from the web interface. As can be seen, every tuatua is detected and counted from the test image.

Figure 13 demonstrates the detection results of an image that are downloaded from the Internet using the final model. The model can detect almost all shellfish objects even on the under layers of other objects. But the model classifies 2 cockles which are partly covered by other objects as tuatua. The object overlap makes the detection more difficult. Making the shellfish objects spread instead of overlapping with each other will increase the detection results. Additionally, the detection will be improved by increasing the dataset size and including tuatua and cockle from multiple locations both of the North Island and South Island of New Zealand.

![Detect and Count]

**Detected Shellfish:**

*tuatua: 24*

**Result**

**Fig. 12.** Shellfish detection and counting results
5 Conclusion and Future Work

Recently, deep learning technology is developed strongly; it has been becoming beneficial for object detection techniques in order to reduce repetitive tasks. This study designs and develops a web/mobile application to recognize and count New Zealand shellfish from images or videos. This is a tool potentially built for helping New Zealand Fisheries officers to control shellfish gathering. At present, there are no studies that develop models to detect and count New Zealand shellfish species. This study can reduce not only the workload of New Zealand Fisheries officers but also fill the gap in the literature review of shellfish detection field. This research demonstrates a procedure of object detection implementation, which includes data pre-processing, model training, and evaluation. From the evaluation results of COCO evaluate metrics, the final Mask R-CNN detection model can recognize and count three major shellfish species, for example, cockle, tuatua, mussel. In terms of bounding box prediction, the average precision is 87.4% when the IoU threshold is calculated from 0.5 to 0.95. The AP at 0.5 IoU threshold can reach 99.6%. The proposed model has proven its potential application in fishery, and other fields to reduce the workload and improve the efficiency.
References

1. Ministry for Primary Industries: Auckland and Kermadec Fishing Rules (2019). https://www.fisheries.govt.nz/travel-and-recreation/fishing/fishing-rules/auckland-kermadec-fishing-rules/

2. Chen, L., Zhang, Z., Peng, L.: Fast single shot multibox detector and its application on vehicle counting system. IET Intell. Transp. Syst. 12(10), 1406–1413 (2018)

3. Cochrane, K., Chelladhurai, J.S., Khare, N.K.: Docker Cookbook: Over 100 Practical and Insightful Recipes to Build Distributed Applications with Docker. Packt Publishing Ltd., Birmingham (2018)

4. Galeone, P.: Hands-on neural networks with TensorFlow 2.0: understand TensorFlow, from static graph to eager execution, and design neural networks. Packt (2019). http://ezproxy.aut.ac.nz/login?url=search.ebscohost.com/login.aspx?direct=true&db=cat05020a&AN=aut.b27409545&site=eds-live

5. Huang, X., Zou, Y., Wang, Y.: Example-based visual object counting for complex background with a local low-rank constraint. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1672–1676. IEEE (2017)

6. Jiang, Y., Li, C., Paterson, A.H., Robertson, J.S.: DeepSeedling: deep convolutional network and Kalman filter for plant seedling detection and counting in the field. Plant Methods 15(1), 141 (2019). https://doi.org/10.1186/s13007-019-0528-3

7. Katsuki, T., Morimura, T., Idé, T.: Unsupervised object counting without object recognition. In: 2016 23rd International Conference on Pattern Recognition (ICPR), pp. 3627–3632. IEEE (2016)

8. Khaki, S., Pham, H., Han, Y., Kuhl, A., Kent, W., Wang, L.: Convolutional neural networks for image-based corn kernel detection and counting. Sensors 20(9), 2721 (2020)

9. Konam, S., Narni, N.R.: Statistical analysis of image processing techniques for object counting. In: 2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 2464–2469. IEEE (2014)

10. Manzoor, S., Joo, S.H., Kuc, T.Y.: Comparison of object recognition approaches using traditional machine vision and modern deep learning techniques for mobile robot. In: 2019 19th International Conference on Control, Automation and Systems (ICCAS), Control, Automation and Systems (ICCAS), pp. 1316–1321 (2019). http://ezproxy.aut.ac.nz/login?url=search.ebscohost.com/login.aspx?direct=true&db=edseee&AN=edseee.8971680&site=eds-live

11. Marsden, M., McGuinness, K., Little, S., Keogh, C.E., O’Connor, N.E.: People, penguins and petri dishes: adapting object counting models to new visual domains and object types without forgetting. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 8070–8079 (2018)

12. Sai, B.K., Sasikala, T.: Object detection and count of objects in image using tensor flow object detection API. In: 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT), pp. 542–546. IEEE (2019)

13. Setti, F., Coniglio, D., Tobanelli, M., Cristani, M.: Count on me: learning to count on a single image. IEEE Trans. Circuits Syst. Video Technol. 28(8), 1798–1806 (2018)

14. Treiber, M.A.: An Introduction to Object Recognition: Selected Algorithms for a Wide Variety of Applications. Springer, London (2010). https://doi.org/10.1007/978-1-84996-235-3
15. Vasconez, J., Delpiano, J., Vougioukas, S., Cheein, F.A.: Comparison of convolutional neural networks in fruit detection and counting: a comprehensive evaluation. Comput. Electron. Agric. 173, 105348 (2020)

16. Wang, K., Fang, B., Qian, J., Yang, S., Zhou, X., Zhou, J.: Perspective transformation data augmentation for object detection. IEEE Access 8, 4935–4943 (2019)

17. Wang, Y., Zou, Y., Chen, J., Huang, X., Cai, C.: Example-based visual object counting with a sparsity constraint. In: 2016 IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6. IEEE (2016)

18. Yu, Y., Zhang, K., Yang, L., Zhang, D.: Fruit detection for strawberry harvesting robot in non-structural environment based on mask-RCNN. Comput. Electron. Agric. 163, 104846 (2019)

19. Zhao, Z.Q., Zheng, P., Xu, S.T., Wu, X.: Object detection with deep learning: a review. IEEE Trans. Neural Netw. Learn. Syst. 30(11), 3212–3232 (2019)

20. Zhiqiang, W., Jun, L.: A review of object detection based on convolutional neural network. In: 2017 36th Chinese Control Conference (CCC), pp. 11104–11109. IEEE (2017)

21. Zoph, B., Cubuk, E.D., Ghiasi, G., Lin, T.Y., Shlens, J., Le, Q.V.: Learning data augmentation strategies for object detection. arXiv preprint arXiv:1906.11172 (2019)