MONASH University

DEPARTMENT OF MECHANICAL AND AEROSPACE ENGINEERING

DOCTORAL THESIS

Deep-Learning in Fruit Detection and Segmentation for Harvesting Robots

Author: Hanwen KANG

Supervisor: Dr Chao CHEN

This report will be submitted for the degree of Doctor of Philosophy of Engineering Science (Research) at
Monash University in 2020.
Department of Mechanical and Aerospace Engineering

LMGA

Laboratory of Motion Generation and Analysis

August, 2020

Copyright notice

Abstract

Robotic techniques show promising aspects in the future development of the agricultural industry, especially in harvesting tasks, which are labour intensive and time-consuming. Among many challenges, vision system is a key towards the fully-function of robotic harvesting. In this thesis, a machine vision system based on recent advancements in deep-learning for autonomous harvesting robots is developed. This thesis firstly reviews recent development in visual sensors, the state-of-the-art techniques in visual processing and vision processing algorithms based on the deep-learning. Chapter 3 develops an automatic labelling algorithm and a light-weight YOLO-based network for training and performing of fruit recognition. Chapter 4 proposes a multi-functional YOLO-based detector based on Chapter 3. The proposed network combines detection and semantic segmentation into a YOLO-based network architecture, which can perform fruit recognition and semantic segmentation on workspace simultaneously. Chapter 5 further improves the function of multi-functional network from previous chapters. The developed network detector can perform detection and instance segmentation on fruits and semantic segmentation on the workspace in orchard environments. To improve the computational efficiency of the network model, a light-weight designed backbone is also applied in the network. Experimental results showed that the developed multi-functional network achieved high accurate performance in both fruit detection, instance segmentation and semantic segmentation on workspace. Meanwhile, with light-weight design of network architecture, our proposed multi-functional network also achieved high computational efficiency. Combining with recent advancement in RGB-D cameras, the developed multi-functional detector can further improves the capability of harvesting robots in unstructured working environments by mapping semantic information from a RGB image to the 3D point clouds. Chapter 6 investigates a machine vision framework for processing 3D point clouds for robotic harvesting. The developed framework includes a workspace modelling algorithm and an object grasping pose estimation algorithm. By combining the aforementioned deep-learning based detector and 3D point clouds processing algorithm, a workflow control framework for harvesting robots is developed in this research. Experimental results of robotic harvesting showed that the developed control framework improved the success rate of harvesting compared to the method without computation of grasping pose.

Declaration

I, Hanwen Kang, declare that this thesis titled “Deep-Learning in Fruit Detection and Segmentation for Harvesting Robots” and work presented in it are my own. This thesis is an original work of my research and contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

Student’s Signature:

Print Name: Hanwen Kang

Date: 20th May 2020

Thesis Including Published Works Declaration

I, Hanwen Kang, hereby declare that this thesis contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

This thesis includes **four** original papers published in peer reviewed journals and **one** unpublished chapter. The core theme of the thesis is **vision perception and environmental modelling for robotic harvesting**. The ideas, development and writing up of all the papers in the thesis were the principal responsibility of myself, the student, working within the **Department of Mechanical and Aerospace Engineering** under the supervision of **Dr Chao CHEN**.

In the case of **Chapters 3, 4, 5, and 6** my contribution to the work involved the following: (continues in next page)

Thesis Chapter	Publication Title	Status	Nature and % of student contribution	Co-author name(s) Nature and % of contribution	Co-author(s) Monash student
3	Fast implementation of real-time fruit detection in apple orchards using deep learning	published	Conceptual, Program-ming, Validation, drafting, 80%	Chao CHEN: input to manuscript,20%	N
4	Fruit Detection and Segmentation for Apple Harvesting Using Visual Sensor in Orchards	published	Conceptual, Program-ming, Validation, drafting, 80%	Chao CHEN: input to manuscript,20%	N
5	Fruit detection, segmentation and 3D visualisation of environments in apple orchards	published	Conceptual, Program-ming, Validation, drafting, 80%	Chao CHEN: input to manuscript,20%	N
6	Visual Perception and Modeling for Autonomous Apple Harvesting	published	Conceptual, Program-ming, Validation, drafting, 75%	Hongyu Zhou : Review of Draft 5%. Chao CHEN: input to manuscript,20%	Y N

I **have not** renumbered sections of submitted or published papers in order to generate a consistent presentation within the thesis.

Student's Signature:

Date: 20th May 2020

The undersigned hereby certify that the above declaration correctly reflects the nature and extent of the students and co-authors contributions to this work. In instances where I am not the responsible author I have consulted with the responsible author to agree on the respective contributions of the authors.

Main Supervisor's Signature:

Date: 20th May 2020

Acknowledgements

First of all, I would like to thank my supervisors, Dr Chao Chen, Prof Ling Li, and Prof Jeremy Grummet, for their continuous and generous support on the research project. In particular, I would like to appreciate Dr Chao Chen, who working tirelessly with me over the tough challenges, providing financial support, seeking invaluable feedbacks from the industry and academic, and most importantly, for equipping me with the priceless and comprehensive skills for conducting researches. I would also thank my friends in Laboratory of Motion Generation and Analysis, who helped a lot in the researches. I would also ackonwledge Dr Lilian Khaw for helping me alot in English writing.

I further wish to express my greatest appreciation to my beloved parents, my father Mr. Kang Wencai and my mother Mrs. Yuan Hong. They had encouraged me to pursue a higher degree as I was in doubt, and have stretched everything they could to support me. I could not love them enough for shaping me into who I am and all the supports they have provided.

Lastly, I would like to express my gratitude to my beloved girlfriend, Mrs Chen Zhuo, for giving me the motivation, assurance and relief I need to battle against the challenges in my life and research career throughout the years, and also her patient wait in the days when we are separated.

List of Publications

Peer-Reviewed Journal Papers

H. Kang and C. Chen, "Fruit Detection and Segmentation for AppleHarvesting Using Visual Sensor in Orchards", Sensors, vol. 19, no. 20, p. 4599, 2019.

H. Kang and C. Chen, "Fast implementation of real-time fruit detection in apple orchards using deep learning", Computers and Electronics in Agriculture, vol. 168, p. 105108, 2020.

H. Kang and C. Chen, "Fruit detection, segmentation and 3D visualisation of environments in apple orchards", Computers and Electronics in Agriculture, vol. 171, p. 105302, 2020.

H. Kang, H. Zhou and C. Chen, "Visual Perception and Modeling for Autonomous Apple Harvesting," in IEEE Access, vol. 8, p. 62151-62163, 2020.

List of Figures

Contents

1 Introduction

1.1 Background

Nowadays, with the continuous increase in cost and availability of the labour resource [1], robotic techniques show huge potential in the future development of the agricultural industry. Autonomous equipments have been widely applied in the harvesting of crops such as corn, wheat, and rice in modified conditions. Compared to the autonomous harvesting in the structured workspace, robotic harvesting in unmodified orchards is much more challenging [162]. Firstly, crops are randomly located in the workspace, which requires a robotic vision to detect and localise the target and perform harvesting. Developing a machine vision system for harvesting tasks in orchard environments is challenging, as variances in crop appearances and environments can severely influence the performance of the system [73, 134, 183]. In the meanwhile, the visual-guided robotic system also has high requirements for real-time processing of sensory data to ensure harvesting efficiency and reduce the effect of environmental disturbance. Secondly, unmodified farm conditions always lead to a complex workspace, which further increases the difficulty of robotic harvesting 4, 171. Machine vision can help identify and model the obstacles within the workspace for better harvesting performance. The robotic system can rely on such information to plan the motion of manipulators and end-effectors correspondingly. Thus, a well-designed machine vision system is a crucial step towards the full function of robotic harvesting.

1.2 Problem Statement and Objectives

In this thesis, we study the case of robotic harvesting in apple orchards. The limitations and challenges of current harvesting robots in the field are listed below.

1. The performance of traditional machine-learning based detection algorithms are limited in terms of generalisation, robustness, and accuracy to be operated in orchard environments.
2. Multiple networks from the processed sensory data are needed for a robot to perform harvesting tasks. However, the current deep-learning networks are designed only for a specific task. Besides, stacking of different deep-learning algorithms will lead to computational inefficiency, difficult to maintain, and lower reliability. Therefore, a multi-functional network is required.

3. An algorithm to understand the working environment and target information is required. This model should be capable of extracting and modelling the workspace and targets from the sensory data, such as mapping the obstacles within workspace and estimating of grasping poses of each object.
4. A robotic operation-flow control framework to guide visual-based robotic harvesting by combining the three points mentioned above is required.

It is challenging for machine vision to accurately detect and localise the targets in varying illumination and complex background environments. Meanwhile, occlusion and overlapping between objects, changing objects' appearance, and view-angle can also severely influence the accuracy of machine vision. For robotic harvesting, real-time inference of machine vision is also another critical requirement for applied machine vision.

Advanced harvesting robots demand a machine vision that is robust and responsive to environmental variances, well-generalised and accurate for object appearances, and highly efficient in computation. This thesis aims to develop a machine vision that can meet the aforementioned requirements for harvesting robots by investigating the latest advancing techniques in vision sensors and processing algorithms. The following objectives are included in this research:

1. To create an accurate, robust, multi-functional, and computational efficient deep-learning vision detector for fruit detection and localisation in orchard environments.
2. To investigate a machine vision system which combines fruit detection and workspace modelling, to improve robots' understanding of working environments.
3. To construct a high-level control framework to guide robotic detachment of fruits by combining the aforementioned points.

1.3 Contributions

Our researches contribute to the development of a machine vision based harvesting control framework to accurately and robustly guide the robotic harvesting. The developed framework includes a deep-learning based multi-task neural network, an environment modelling algorithm, and a control framework. The proposed research yields four publications during the candidature, which are listed as follows:

Presented as Chapter 3:

H. Kang and C. Chen, "Fast implementation of real-time fruit detection in apple orchards using deep learning", Computers and Electronics in Agriculture, vol. 168, p. 105108, 2020.

Presented as Chapter 4:

H. Kang and C. Chen, "Fruit Detection and Segmentation for Apple Harvesting Using Visual Sensor in Orchards", Sensors, vol. 19, no. 20, p. 4599, 2019.

Presented as Chapter 5:

H. Kang and C. Chen, "Fruit detection, segmentation and 3D visualisation of environments in apple orchards", Computers and Electronics in Agriculture, vol. 171, p. 105302, 2020.

Presented as Chapter 6:

H. Kang, H. Zhou and C. Chen, "Visual Perception and Modeling for Autonomous Apple Harvesting," in IEEE Access, vol. 8, p. 62151-62163, 2020.

Chapter 3 introduces a fast labelling method for network training and developed a YOLO-based one-stage network for fruit detection. Chapters 4 and 5 further develop a real-time multi-functional deep-learning network Detection and Segmentation Network (DaSNet-v1 and DaSNet-v2) based on Chapter 3 for robotic harvesting. Chapter 6 investigates the workspace modelling in the vision processing, and includes it in high-level control strategies for vision-guided harvesting robots. The details of each chapter are introduced in the following section.

1.4 Thesis Structure

Chapter 2 presents the literature review on recent development in visual sensors, image processing algorithms, and harvesting robots. For the visual sensor, the commonly applied 2D imaging camera and 3D imaging/range sensors are reviewed. For the image processing algorithms, traditional machine-learning based methods and deep-learning based methods are critically reviewed, with particular attention paid to the deep-learning based methods. For the harvesting robots, a review of the recent development of a robotic system in visual-guided harvesting is included.

Chapter 3 presents a fast implementation framework of deep-learning based fruit detection algorithm to perform real-time fruit detection in apple orchards. The developed framework comprises an auto label generation module and a deep-learning based fruit detector 'LedNet'. The Label Generation al-

gorithm utilises the multi-scale pyramid and clustering classifier to assist the fast labelling of training data. LedNet adopts feature pyramid network and atrous spatial pyramid pooling to improve the detection performance of the model, while a light-weight backbone is utilised to increase computational efficiency.

Chapter 4 presents a multi-function network to perform the real-time detection and semantic segmentation of the apple and the branch in the orchards environment by using the visual sensor. The developed detection and segmentation Network utilizes the Atrous Spatial Pyramid Pooling and the Gate Feature Pyramid Network to enhance the feature extraction ability of the network. Meanwhile, to improve the real-time computation performance of the detection and segmentation network, a light-weight backbone network based on the Residual Network Architecture is developed.

Chapter 5 presents an improved deep neural network DaSNet-v2, which can perform detection and instance segmentation on fruits, and semantic segmentation on branches. In the previous work, a deep neural network DaSNet-v1, was developed to perform detection and segmentation on fruits and branches in orchard environments. However, semantic segmentation returns the mask for each class instead of each object. Segmentation on each fruit is important as it can provide abundant information on each object, especially for those overlapped fruits.

Chapter 6 presents a robotic vision system to perform fruit recognition, modelling, and environment modelling for autonomous apple harvesting. The fruit recognition applies a deep-learning model Dasnet to perform detection and segmentation on fruits, and segmentation on branches. Fruit modelling localises the center and computes the grasp pose of each fruit based on Hough Transform. Environment modelling adopts Octrees to represent the occupied space within the working environment of the robot. The robot control computes the path and guides the manipulator to pick the fruits based on the computed 3D model of the crop.

Chapter 7 summarises the outcomes of the research and contributions from this work. Key limitations on the practical implementation of current methods and technologies and suggestions for future research are made.

2 Literature Review

Compared with traditional autonomous harvesting, visual-guided robotic harvesting requires precise information of target to guide the manipulator to detach the fruits or vegetables. A typical robotic harvesting system consists of a centre control, a vision system, a manipulator, an end-effector, and a mobile platform. The accuracy and robustness of the vision system are essential for operating the robotic harvesting system in orchard environments. Compared to the vision systems which are widely applied in controlled industrial environments, the operation of the vision system in orchard environments is more challenging. Various uncontrolled environmental factors, such as changing illumination, appearance of fruits (size, colour, shape, and texture), complex and loosely structured workspace, and occlusion between objects, can severely influence the performance of the robotic harvesting system. This chapter reviews the related methods and techniques which have been developed in visual-guided harvesting robots. The review includes three sections, which are vision sensors in Section 2.1, vision processing algorithms, and recent development in harvesting robots in Section 2.4. The review of the vision processing algorithm includes two parts: traditional machine-learning based methods in Section 2.2 and deep-learning based methods in Section 2.3.

2.1 Vision Sensors for Harvesting Robots

2.1.1 2D Imaging Sensors

(a) Colour image taken by RGB camera (b) NIR image taken by infrared camera

Figure 1: RGB and NIR images are applied in fruit detection in greenhouse environments [134].

Visual perception of target objects can be conducted by different types of visual sensors, which can be classified into two classes: 2D imaging sensors and 3D imaging sensors [181]. 2D imaging sensors

include RGB colour camera, infrared imaging sensors, spectral imaging sensors [90]. RGB camera senses the colour information of the objects and environments, while infrared imaging camera captures the temperature distributions of the plant canopy (as shown in Figure 1). Some researchers fuse the sensory data of RGB colour camera and infrared imaging camera to improve the performance of the vision system in different conditions [116, 158]. 2D imaging sensors have advantages such as low cost and easy access. However, they cannot provide 3D spatial information of objects within the workspace. Some vision servo methods apply a monocular camera to reconstruct the spatial information by using structure from motion, which requires cooperation between manipulator motion and vision system [108, 138]. Such methods can reconstruct the 3D location of objects while they would increase the complexity of the computation and system design. With the recent development of the 3D visual sensors, combining 2D imaging sensors and 3D measurement sensors can provide both colour and 3D spatial information of objects and environments [98].

2.1.2 3D Visual Sensors

3D visual sensors can obtain spatial information of target objects and workspace. The commonly applied 3D visual sensors in robotic harvesting include stereo-camera, RGB-D camera, and Light Detection and Ranging (LiDAR). The following sections will briefly introduce the principles, advantages, and limitations of these visual sensors.

a. Stereo Camera

The stereo camera applies two or multiple RGB cameras to reconstruct the colour and depth information by matching through the triangulation principle [38]. Based on different approaches to perform matching, the Stereo camera can be divided into dense-matching based Stereo camera [37, 55] and sparse-matching based Stereo camera [163]. A dense-mapping based stereo-camera reconstructs the 3D information of the environment by matching every pixel between the images from the left and right cameras, as shown in Figure 2. A sparse-matching based stereo camera applies key-point detection and image feature descriptors to reconstruct the spatial location of these extracted key points [118]. Similar to the 2D imaging sensors, the stereo camera is low cost and easy access. However, the stereo camera also has many limitations, such as low accuracy, low computation efficiency, and low robustness to the environmental variances [159]. Also, the application of the stereo camera is limited in scenarios that lack texture features.

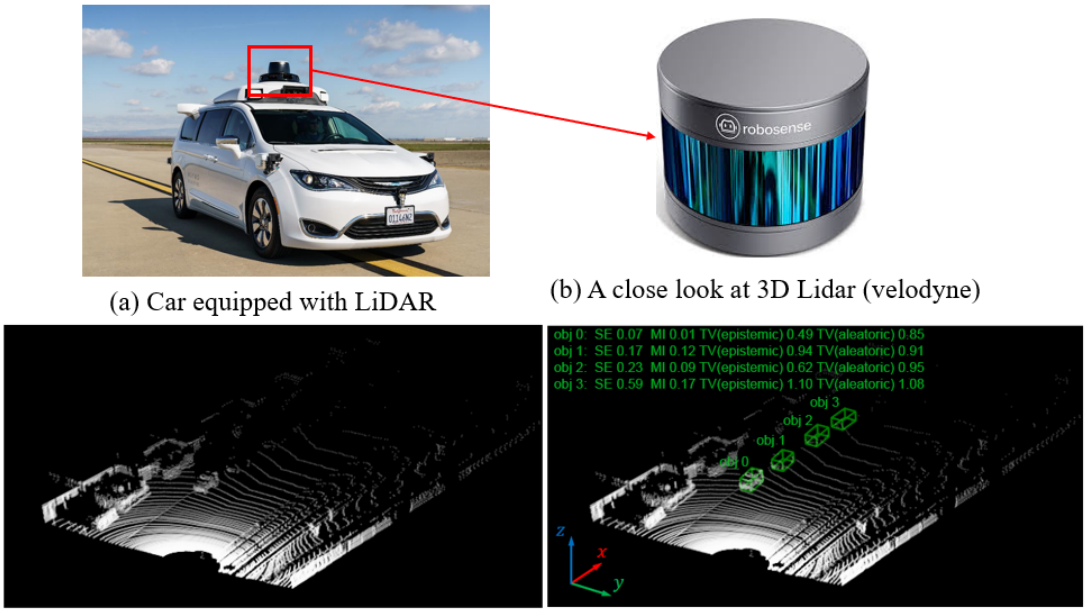
(a) Stereo-camera relies on triangle to estimate depth of pixels

(b) Stereo-camera by ZED

(c) Reconstructed depth maps from the stereo-camera

Figure 2: Woking principle of the stereo-camera, sub-image (c) shows the example of reconstructed depth map of scenes [137].

b. LiDAR



(a) Car equipped with LiDAR

(b) A close look at 3D Lidar (velodyne)

(c) 3D point clouds from the 3D LiDAR

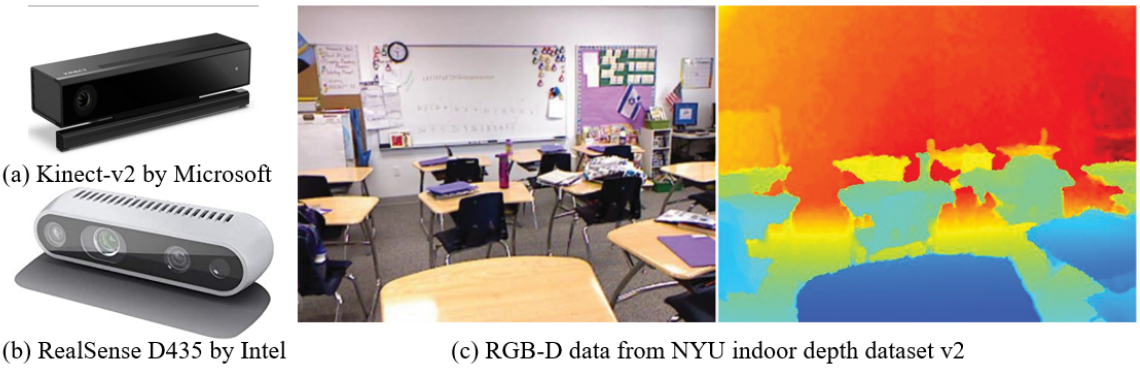
(d) Object detection on 3D point clouds

Figure 3: 3D LiDAR sensors are widely applied in autonomous driving, (c) and (d) are from the [31]

LiDAR is a range sensor that can obtain spatial information of environments. The sensory data of the LiDAR is the 2D or 3D point clouds based on different types of sensors. LiDAR has advantages in refresh rate, sensory accuracy, and robustness. For example, Velodyne HDL-64E LiDAR (as shown in Figure 3) can sense the environments up to 120 meters in a real-time refresh rate. However, LiDAR cannot provide colour information of the objects and environments, which limits the information that

can be used by the recognition algorithm. Meanwhile, the cost of the 3D range LiDAR is much higher. 3D LiDAR, co-operated with colour imaging sensors or other distance measurement, is widely applied in studies and applications in autonomous driving [166, 189]. The 2D LiDAR is widely applied in Simultaneously Localisation and Mapping (SLAM) project in indoor environments [56].

c. RGB-D Camera



(a) Kinect-v2 by Microsoft

(b) RealSense D435 by Intel

(c) RGB-D data from NYU indoor depth dataset v2

Figure 4: RGB-D data by using Kinect-v2 in indoor scene, which are from NYU dataset [143].

The RGB-D camera can obtain both colour and depth information of the workspace. Similar to the binocular camera, most of the RGB-D cameras utilises the design of stereo-camera, while one of the cameras is replaced with a distance measurement sensor. Based on the different type of applied distance measurement sensor, RGB-D cameras can be divided into light structural based solutions (such as Microsoft Kinect-v1) and Time of Flight (ToF) solution (such as Microsoft Kinect-v2 and Intel RealSense series, as shown in Figure 4). Both solutions require the registration between depth images and colour images, to find the correct depth value of each pixel within the colour images [76]. Compared to the LiDAR, the working range distance of the RGB-D camera is relatively smaller, which can go up to 10 meters. However, an RGB-D camera can provide rich information, including both colour and spatial data of objects and workspace. Compared to the stereo-camera, RGB-D camera has significant advantages in terms of measurement accuracy, robustness, and computational efficiency.

2.2 Traditional Machine-Learning Methods in Vision Processing

Traditional machine-learning based recognition methods apply feature descriptor to extract and encode the colour, shape, and texture features of the objects [115]. These feature descriptors are also

called as hand-crafted features. Based on extracted feature descriptors, machine-learning based classifier is applied to learn the distribution of object' features in order to perform classification, recognition, or segmentation on sensory data [26]. The following section will introduce the related works developed in feature descriptors and traditional machine-learning based classifier in separately.

2.2.1 Feature Descriptors

Feature descriptor is used to encode the distribution of objects' appearance in sensory data. According to different processing data, feature descriptors can be divided into image feature descriptors and point clouds feature descriptors.

a. Image Feature Descriptors

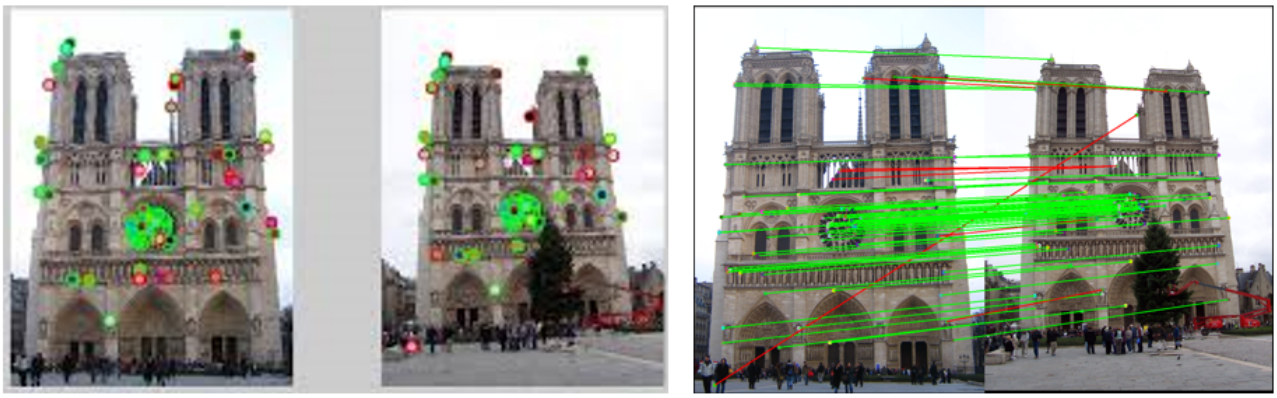
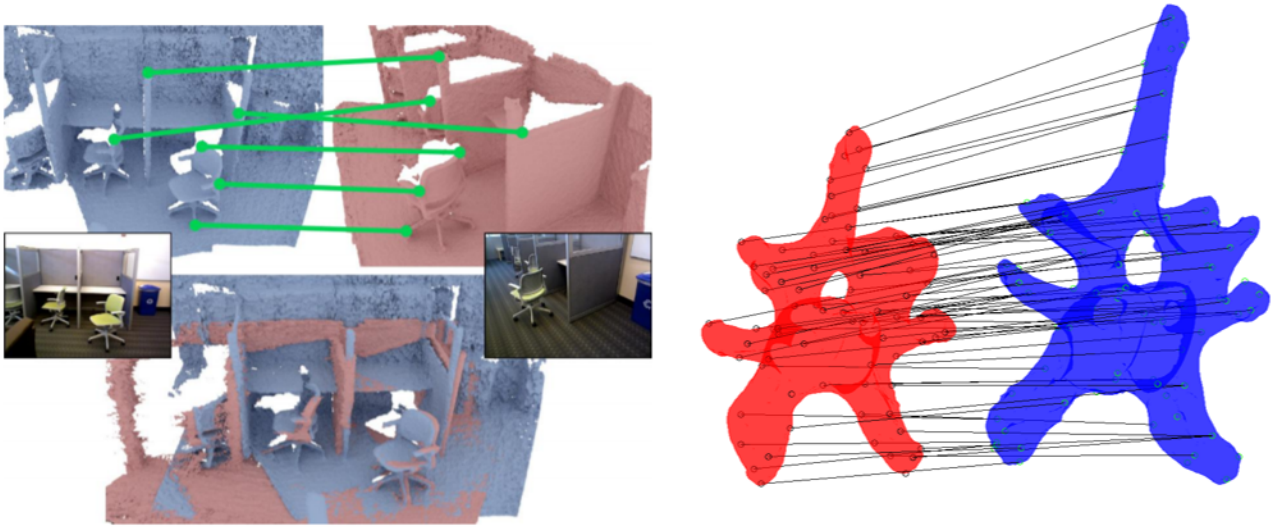


Figure 5: Key-point matching by using local feature descriptor and RANSAC algorithm [66].

Image feature descriptors encode the appearance of objects in the 2D images, which can be used in colour channel of the stereo-camera or RGB-D camera. Based on different features encoded by feature descriptors, image feature descriptors can be classified into gradient-based methods, texture-based methods, frequency-based methods, and moment and probability-based methods [88]. Most of image feature descriptors applied in objects recognition belong to local feature descriptor [89], which record features of objects in a given local region, such as Histogram of Gradient (HoG) [141], Local Binary Pattern (LBP) [2], and Binary Robust Independent Elementary Features (BRIEF) [20]. These image feature descriptors can be used to perform matching or registration by sliding windows or Key-point detection [81]. Key-point detection is another important step in the processing of image feature descriptor [109]. Key-point detection extracts the salient pixels within the images and then use feature descriptors to encode the neighbour region of these extracted key points. Many image feature

descriptors include Key-point detection during the process as it can improve computational efficiency in image recognition, matching (as shwon in Figure 5), and registration [96]. The representative works of this method are Scale Invariant Feature Transform (SIFT) [104], Speeded Up Robust Features (SURF) [15], and Oriented FAST (Features from Accelerated Segment Test) and Rotated BRIEF (ORB) [130]. These image feature descriptors are widely applied in image matching /registration [146] and SLAM [112].

b. Point Clouds Feature Descriptors



(a) Keypoint detection in 3D point clouds and matching (b) 3D shape matching by using FPFH and RANSAC

Figure 6: 3D shape matching by using Key-point detector and local feature descriptor [179].

Point clouds feature descriptors extract the appearance of objects in the 2D or 3D point sets, which can be applied in LiDAR or RGB-D camera cases. Point clouds feature descriptors can be divided into local feature descriptors and global feature descriptors [47]. Local feature descriptor encodes local geometric features of objects, which can be used to perform recognition, segmentation, and registration [33]. The commonly used local point cloud feature descriptor includes spin images [69], shape context [16], Point Feature Histogram (PFH) [133], and Fast Point Feature Histogram (FPFH) [132] and so on. Global feature descriptor encodes geometric information of whole point cloud, which is applied in shape retrieval and classification [153, 167]. The commonly used local point cloud feature descriptor includes Global Radius-based Surface Descriptor (Global RSD) [106], Viewpoint Feature Histogram (VFH) [111], 3D-SIFT [139], and so on. Similar to the image feature descriptors, key-point detection is also an important step in the descriptor extraction in point cloud cases [135].

Considering point clouds are in sparse and unordered form, key-point extraction in point clouds case is even challenging compared to the key-point detection in image space. Key-point detection in point clouds can follow different principles to detect salient points, such as surface curvature or other surface variances [45, 48]. By combining key-point detection and feature descriptors, point clouds feature descriptors can be used to perform different visual tasks on point clouds data, such as 3D shape registration shown in Figure 6.

2.2.2 Traditional Machine-Learning Based Classifier

Feature descriptor encodes objects' information into vectors, which modelling the objects' appearance in feature space. Machine-learning based classifier can be trained to learn the distribution of feature descriptors and perform classification and detection based on extracted feature vectors [156]. Based on different learning and training strategies, machine-learning based classifier can be divided into supervised learning methods, unsupervised learning methods, and semi-supervised learning methods [3]. Supervised learning methods require ground-truth on each training data [190]. The typical supervised machine-learning based classifier includes decision tree learning, K-Nearest Neighbour (KNN) [29], Neural Network (NN) [43], Ensemble Learning [27], and Support Vector Machine (SVM) [149]. Unsupervised learning method does not require ground-truth of training data. It can classify data into different groups based on the internal distribution of the data [39]. The most representative method of unsupervised learning is clustering, such as k-mean clustering and Gaussian Mixture Models (GMMs) [172]. Semi-supervised learning methods are designed to use a small number of labelled training data to conduct feature learning on a large number of un-labelled training data based on given assumptions [190]. In the vision tasks, supervised learning methods are more commonly used than the other two methods since studies showed that it could provide a more accurate and robust classifier to perform visual tasks [140].

A typical traditional machine-learning based visual recognition algorithm includes three steps: feature descriptor extraction, classifier training, and classifier prediction [49]. In the step of feature extraction, one or more feature descriptors can be used to describe the appearance of target objects in the sensory data from different aspects, such as colour, texture, and shape. These feature vectors can be concatenated together to form an ensemble feature vector to describe the appearance of the objects [61]. The applied machine-learning algorithm will train the classifier based on the extracted feature vectors from the training data. At the prediction step, the sliding window is used to extract

feature vectors of each sub-region within the images, and the classifier is used to predict the class of each feature vectors from the previous step [83]. To improve the computational efficiency of the algorithm, selective searching of Region of Interest (RoI) is used to replace the sliding window [28, 157]. That is, only the sub-regions of the RoI are predicted by classifier, which largely increases the running speed of the algorithm.

2.2.3 Traditional Vision Algorithms in Harvesting Robots

Traditional machine-learning based algorithms are widely applied in vision system for harvesting robots [73]. Based on different features and classifiers applied in methods, traditional vision algorithms can be divided into single-feature based methods and multiple-features based methods.

a. Single-Feature Based Fruit Recognition

Colour is one of the most commonly used features to distinguish fruits from the background. Arefi et al. [6] combined colour information from RGB, HSI, and YIQ spaces to extract ripe tomatoes from greenhouse environments, an accuracy of 96.36% on detection and localisation of fruits was reported. Yin et al. [174] used a k-mean clustering algorithm to perform recognition of ripe tomatoes in L*a*b colour space. Huang and He [58] applied a fuzzy entropy-based classifier and combined colour information of L*a*b, HSI, and LCD space to perform detection of Fuji apple on the tree canopy. Si et al. [142] applied a decision tree classifier on RGB colour space to detect and localise the apples by using a stereo-camera. Zhao et al. [182] applied an adaptive threshold algorithm to perform tomato recognition based on fused colour features in L*a*b, and YIQ colour spaces, an accuracy of 93% on detection was reported. Zhao et al. [183] developed a tomato recognition algorithm, which use Adaboost classifier [50] on colour features to perform detection. The author reported that 96% on accuracy of detection was achieved by using their method. Zhou et al. [187] applied a decision tree to perform detection on Gala apples based on colour information. Except for the colour, shape, and texture are the other commonly used features in fruit recognition. Bracamontes et al. [113] applied hough transform to detect blueberries. Xie et al. [170] applied a modified hough transform on the spherical shaped fruits such as apple, citrus, and tomato. Due to the various factors such as illumination, objects' appearances, view-angle of the camera, the accuracy and robustness of single-feature based is limited [7]. In the following section, traditional vision algorithms apply multiple-features are introduced.

b. Multiple-Features Based Fruit Recognition

Combining or fusion of multiple features of the target object can improve the accuracy and robustness of the fruit recognition algorithm in different conditions. Zhao et al. [180] developed a fruit recognition algorithm which applies colour and texture features to perform detection, localisation, and guides the robotic harvesting in orchard environments. Patel et al. [117] combined features of intensity, colour, image edge, and orientation to perform detection on fruits. The author reported that the proposed multiple-features based recognition algorithm achieve 90% on the accuracy of detection. Rakun et al. [123] developed a texture, colour, and 3D shape properties based detection algorithm to estimate the yield of apples in orchard environments. Luo et al. [105] combined features from RGB, HSI, L*a*b, and YCbCr colour spaces and textures to encode the appearance of grape, then an Adaboost classifier is used to perform classification on extracted features. Wang et al. [161] combined colour, texture, and shape features and an Adaboost classifier to perform recognition on citrus fruits. McCool et al. [107] developed a colour, shape, and texture features fused fruit recognition algorithm to perform pixel-level segmentation on sweet peppers in the indoor environments, and a Laplacian of Gaussian (LoG) multi-scale blob detector [97] is used to detect peppers from the segmentation results. Kang and Chen [72] developed a fruit recognition algorithm which applies the hierarchy multi-scale feature extraction of colour and shape features, then a K-mean and CNN are used to perform segmentation and classification on extracted RoIs. Although multi-features based fruit recognition improves the performance of the algorithm, such a system still requires large improvements on both computational efficiency, accuracy, and robustness [181].

2.3 Deep-Learning in Vision Processing

Deep-learning is a kind of supervised machine-learning method, which shows superior performance compared to the traditional algorithms. The accuracy and generalisation of the traditional machine-learning based algorithm is limited by applied hand-crafted feature descriptors [75], which have limited space to store the features of different objects [136]. Deep-learning includes deep neural network and deep Convolution Neural Network (CNN). Compared to the deep neural network, CNN applies convolution operation and back-propagation training to extract the features of the objects, largely increases the accuracy and generalisation of the algorithm [79, 160, 184]. Based on different tasks, deep-learning architecture can be grouped into a classification network, semantic segmentation net-

work, object detection network, and instance segmentation network [44]. Recent advance of CNN allows people to process unordered and sparse data, such as point clouds of the 3D range sensors [63]. This section reviews related works of deep-learning from the aforementioned aspects, a review of deep-learning methods in agricultural harvesting applications is also included.

2.3.1 Deep-Learning in Image Classification

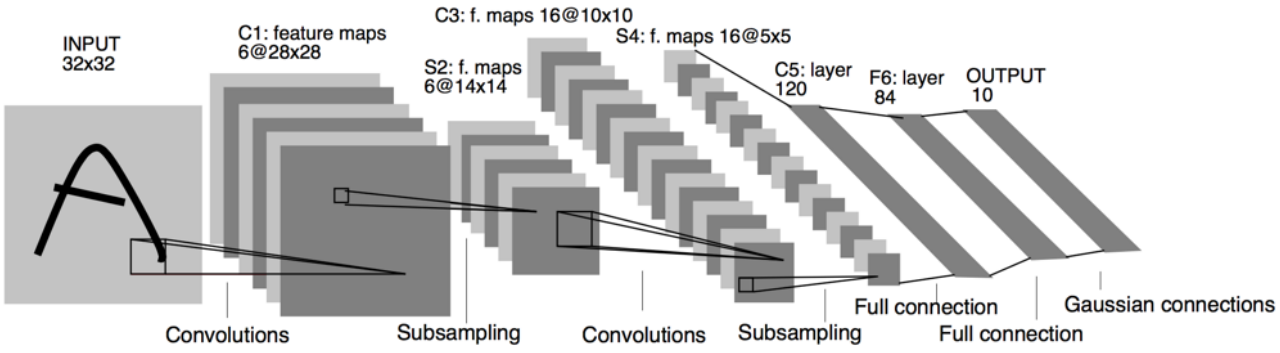
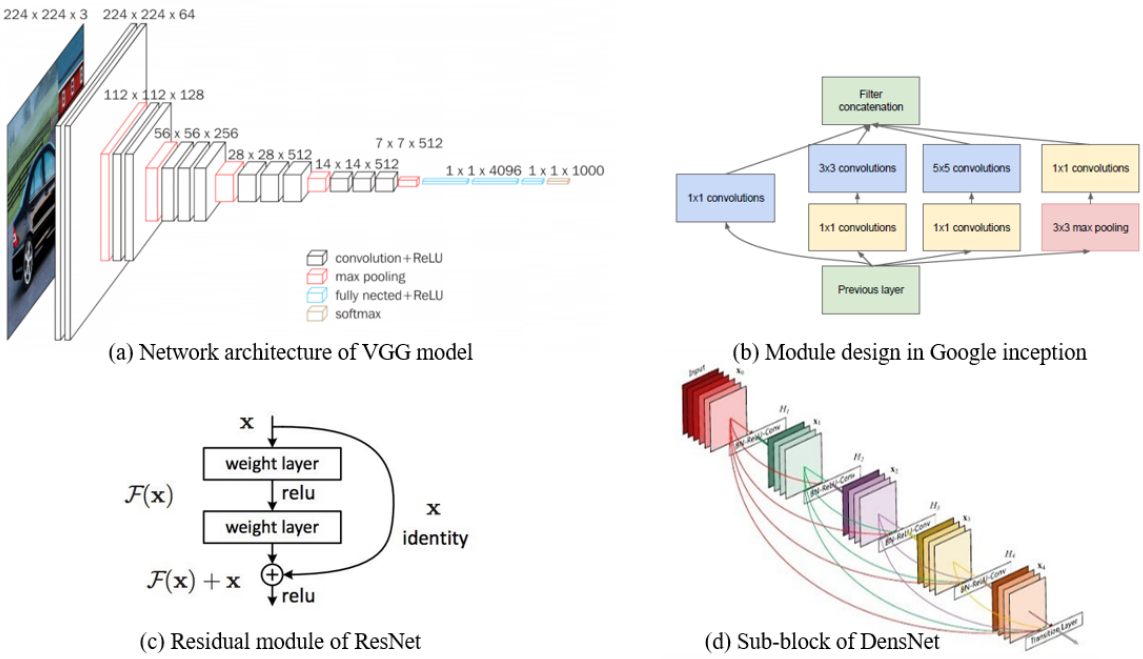


Figure 7: Network architecture of LeNet, which includes three convolution layers (C1, C3 and C5), two pooling layers (S2 and S4), and one fully-connected layer (F6) [84].

The image classification task is to predict the class of the objects within the given images. Different from the traditional machine-learning based methods which apply handcrafted features to encode the appearance of objects, deep-learning based methods apply back-propagation training to automatically extract features based on network training [35]. Therefore, deep-learning methods are also known as adaptive feature learning [148] or automatic feature engineering [46]. In the image classification task, multi-layers perceptron flatten the images to 1-dimensional vectors and uses fully-connected layer to extract and process the information within the images [17, 114, 131]. Later, the convolution Neural Network (CNN), which applies convolution operation to preserve the spatial distribution of the features in images, largely improves the accuracy of the model. LeNet [84, 85] is the first CNN architecture developed for digit number classification. A typical network includes three different operations: convolution (convolution layer), pooling (pooling layer), and matrix multiplication (fully-connected layer). LeNet includes three convolution layers and one fully-connected layer, to extract features from images and perform classification based on extracted features.

The early works of CNN showed limited performance in general classification tasks. To improve the accuracy and generalisation of the network model, researchers add more layers into the network architecture, such as AlexNet [82], ZF-Net [177], Visual Geometry Group Network (VGG-Net) [144],



(a) Network architecture of VGG model

(b) Module design in Google inception

(c) Residual module of ResNet

(d) Sub-block of DensNet

Figure 8: Network architecture design in VGG [144], GoogleNet [151], ResNet [54], and DensNet [57].

and GoogLeNet [150–152]. However, with the increasing number of the layer in the network model, back-propagation training becomes unstable and inefficient due to the gradient vanishing and exploding [71]. Introducing of Relu activation function [42] and batch-norm layer [64] largely optimise training in deep network architecture, providing efficient solutions to train very deep network. In the year 2015, Residual Network (Resnet) [54] (as shown in Figure 8) applies local shortcuts connection within the residual block, which solve the degradation problem in training of the very deep network. Resnet-152 (152 layers architecture) achieved 96.43% in top-5 of ImageNet Large Scale Visual Recognition Competition [25] in year 2015 (ILSVRC-2015). After introduction of Resnet architecture, GoogLeNet-inception-v4 [150], Renext [169], and DenseNet [57] were further developed, which further improves the accuracy of model in the classification task. With continuous improvements made, CNN models have achieved state-of-the-art performance in the image classification task. Based on the CNN architecture, deep-learning methods are widely used in other computer vision tasks, including semantic segmentation and object detection [65], which are introduced in the following sections.

Figure 9: Semantic segmentation is to predict the class of every pixels within image, the image shown above is from the Visual Object Classes Challenge 2012 (VOC2012) [30]

2.3.2 Deep-Learning in Semantic Segmentation

The image classification task is to predict the class of objects within the image, while semantic segmentation task is to predict the class of every pixel within the image (as shown in Figure 9). Traditional semantic segmentation algorithms apply sliding windows and handcrafted features to perform classification on each pixel, which is computation in-efficiency and time-consuming [10, 62, 165]. Deep-learning based semantic segmentation algorithm applies CNN model to extract features and restore classification information of each pixel by using upsampling operations. From the visualisation of feature maps from different levels of CNN [177], feature maps from shallow levels contain more spatial information of objects, while feature maps from deeper levels contain more semantic information of objects.

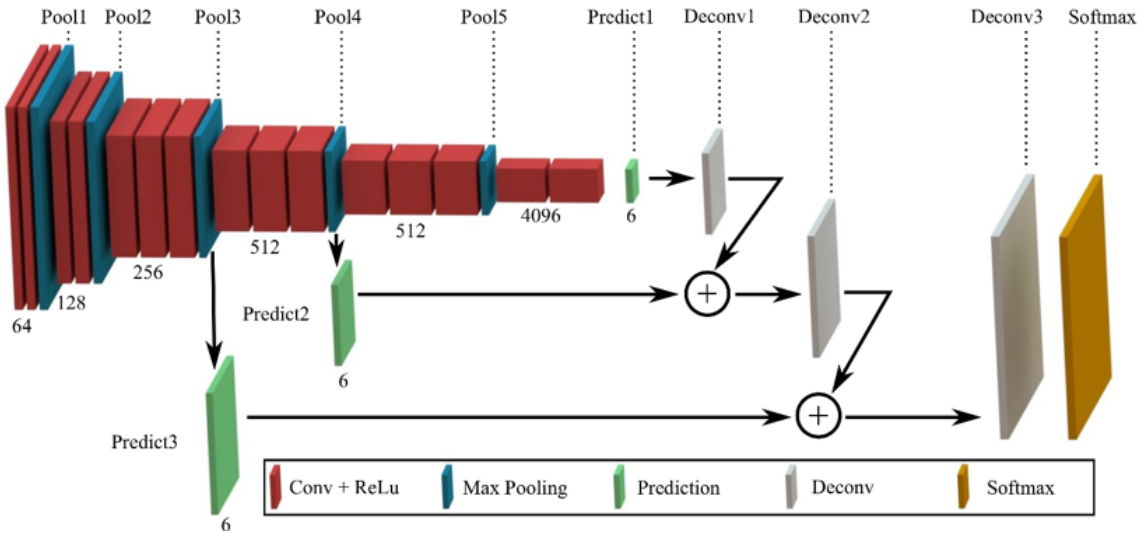


Figure 10: Architecture of FCN models [80], which combines multiple-levels feature maps to perform segmentation.

To restore classification information on each pixel, a semantic segmentation network requires to combines semantic information from deep level and spatial information from the shallow level of the backbone. Due to the pooling operation, the size of feature maps in a deep level is smaller than the feature maps in a shallow level. Hence upsampling (interpolation or transpose convolution operation) is used to match the size of features between different levels. The semantic segmentation network is required to restore spatial information of objects from the feature maps of the higher level, to perform classification on each pixel. Therefore, upsampled operations, such as interpolation and deconvolution operation, are applied in the semantic segmentation network to restore spatial information of feature maps in the higher level [34]. Fully Convolution Network (FCN) [103] is the first work of deep-learning based semantic segmentation, which use feature maps from C5 (32-times size reduced), C4 (16 times size reduced), and C3 (8 times size reduced) in a Feature Pyramid Network (FPN) of a VGG backbone to perform classification on each pixel (as shown in Figure 10). Semantic segmentation network are widely applied in many computer vision tasks, such as U-net [129] and V-net [110] for medical image analysis, Segnet [9] and Deeplab [21, 22] for segmentation in general scenes.

2.3.3 Deep-Learning in Object Detection

Object detection is another essential task in computer vision, which is required to search the object of interest, localise the corresponding Object Bounding Box (OBB), and predict the class of detected objects within images. Compared to the semantic segmentation networks, object detection networks need to return how many objects of interest are in the image and where these objects are. Based on network architecture design, deep-learning based object detection can be divided into two groups: two-stage detection and one-stage detection [185]. Moreover, object detection can only return a bounding box of objects within the image while losing the details of the shape. Therefore, instance segmentation network is further developed, which can perform segmentation on each detected object.

a. Two-stage detection

Two-stage detection extends the traditional machine-learning based detection methods by applying CNN in the stage of object classification and OBB regression. The first work of two-stage detection is Region Convolution Neural Network (RCNN) [41] (as shown in Figure 11), which includes two steps in detection: RoI proposing and RoI classification. In the RoI searching step, the selective searching method [157] is used to predict the bounding box of objects. In the RoI classification step, each

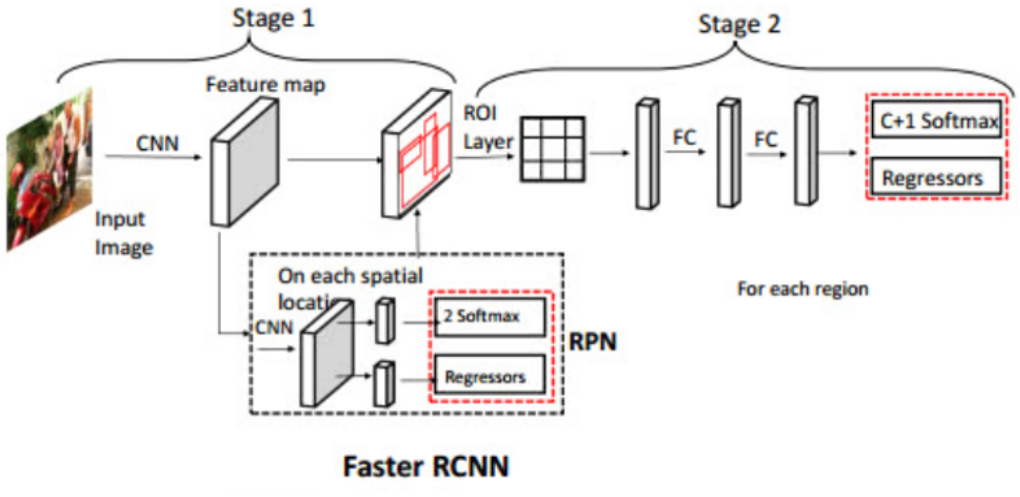


Figure 11: Workflow of two-stage detection network Faster-RCNN [23], which includes RoI proposing and classification during the detection.

proposed RoI is processed by CNN to perform classification and OBB regression. Since each RoI requires one forward inference of network, hence RCNN is limited by the slow running speed [53]. Fast-RCNN [40] improves computational efficiency by direct proposing ROI from the extracted feature maps from the backbone network. Moreover, Faster-RCNN [127] introduces the Region Propose Network (RPN) to generate RoI for detection, improving the RoI searching speed and accuracy compared to the selective searching methods. Although two-stage detection significantly improves the accuracy, robustness, and computational efficiency, it still cannot achieve real-time detection, which is an important requirement in robotic vision system [186].

b. One-stage detection

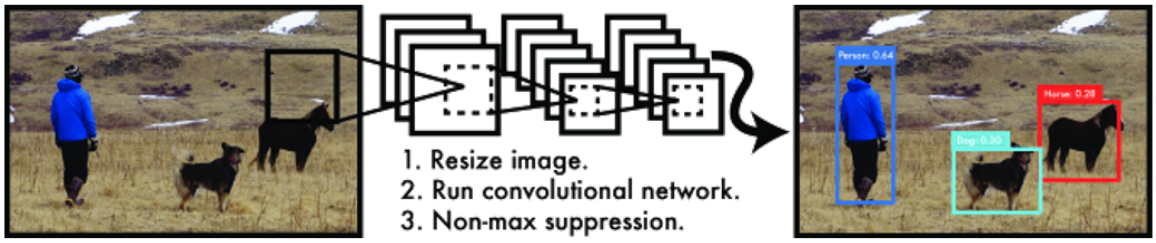


Figure 12: Workflow of one-stage detection network such as YOLO [124], which does not requires RoI proposing during the detection.

Two-stage detection methods divide the object detection into RoI proposing and RoI classification, which cannot achieve real-time processing of input sensory data. One-stage detection combines RoI

proposing and RoI classification into one step, which significantly speeds up the running of algorithms. Compared to the two-stage detection methods, which requires RPN to search RoIs and extract correspond RoI region from feature maps to perform classification, one-stage detection methods predict objects on each grid of the feature maps. The architecture of one-stage detection networks is similar to architecture of the FCN. The difference is that each grid of feature maps in a one-stage detection network contains information of bounding boxes and class of objects. You Only Look Once (YOLO) [124–126] (as shown in Figure 12) and Single Shot Detection (SSD) [99] are the representative works of the one-stage detection methods. Compared to the two-stage detection networks, one-stage detection networks have better performance in computational efficiency but relatively lower accuracy on detection and bounding box localisation [185].

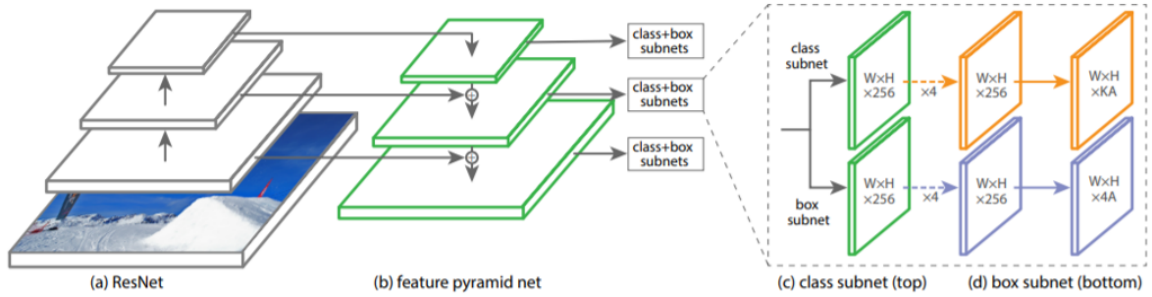
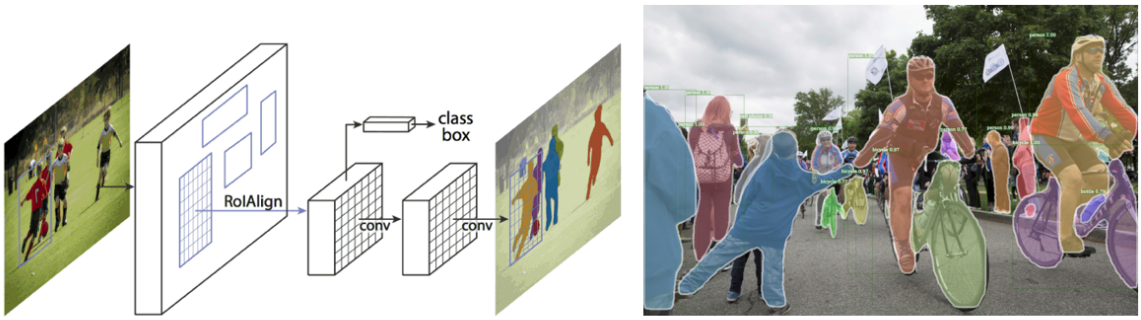


Figure 13: Architecture of RetinaNet [95], which applies FPN and focal-loss training to improve the accuracy on object detection.

To improve the performance of one-stage detection methods in terms of detection and localisation accuracy, multi-scale detection by using FPN [94], focal loss training [95] and multiple prior anchor box are applied in the following improved network models of the YOLO and SSD. YOLO-V3 network applies a three levels FPN structure to increase recall of model on small objects. Meanwhile, nine prior anchor boxes based on training sets are used to improve the bounding box localisation accuracy. By introducing such measurements, one-stage detection can achieve equal accuracy compared to the two-stage detection network but with faster running speed.

c. Instance Segmentation

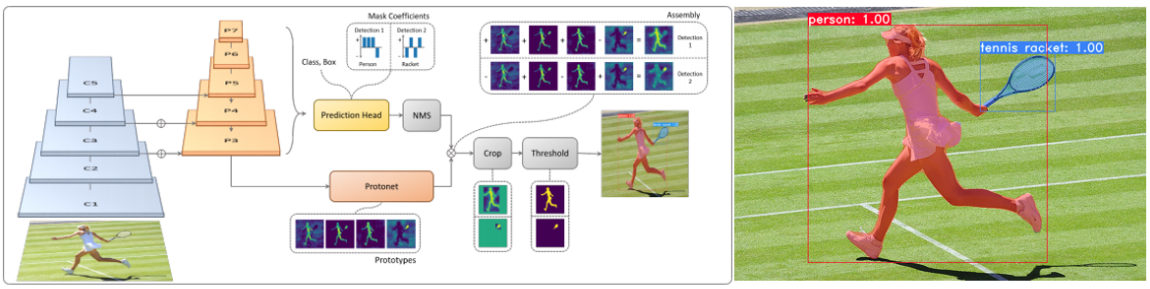
Semantic segmentation can obtain shape of objects while it cannot distinguish which pixels belongs to which objects within the image. Object detection can only returns of bounding box of each object within image while lose the details of objects' shape. Instance segmentation can perform object detection and also obtain the shape of each object within the image. A common approach of achieving



(a) Network architecture of the Mask-RCNN (b) Instance segmentation by using Mask-RCNN

Figure 14: Architecture of Mask-RCNN [52], which combines the architecture of Faster-RCNN and a instance segmentation.

the instance segmentation is to apply a mask generation branch network to perform segmentation of objects after detection. Mask-RCNN [52] (as shown in Figure 14) use this principle, adding a mask segmentation branch in RoI classification step, to obtain a binary mask of each objects. However, because of such working principle, two-stage instance segmentation cannot achieve real-time running speed of algorithm.



(a) Network architecture of the YOLACT network (b) Instance segmentation by using YOLACT

Figure 15: Architecture of YOLACT [19], which combines a detection branch and a protonet branch to generate mask for each objects within images.

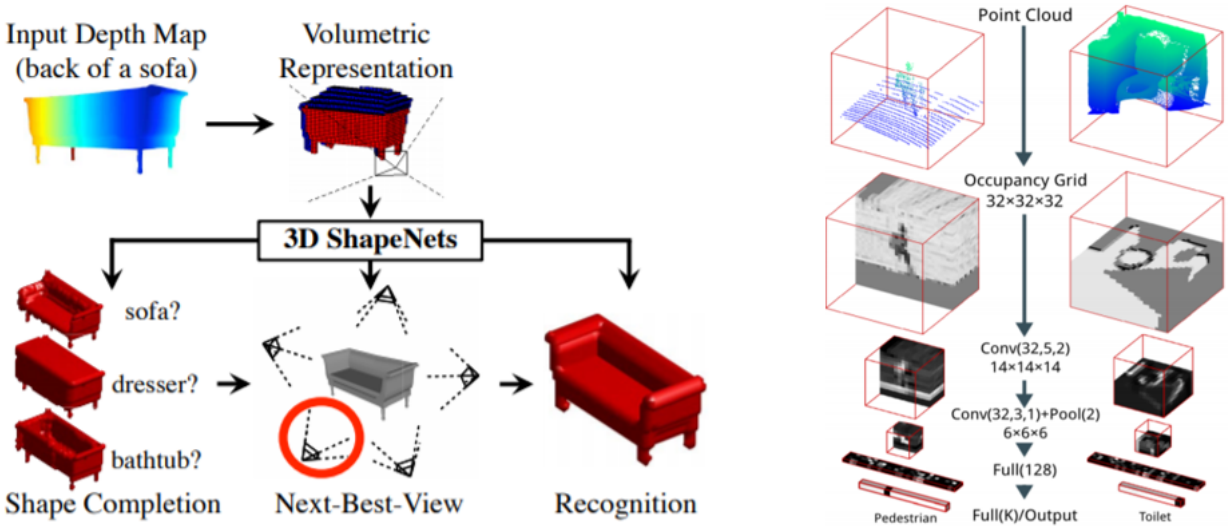
To achieve instance segmentation function by using a one-stage detection network is more challenging compared to the two-stage detection network, as one stage detection network does not have RoI proposing stage, which can directly perform instance segmentation on correspond region of objects on feature maps. Until to now, two different types of one-stage instance segmentation network are developed, which are Single Pixel Reconstruction Network (SPRNet) [175] and YOLACT [19]. SPRNet [175] achieves the one-step instance segmentation by introducing a multi-scale feature fusion branch to generate a mask of objects from a single pixel within the feature maps. That is, an Atrous Spatial Pyramid Pooling (ASPP) [22], which uses several different dilate convolution kernels to fuse

multi-scale information within feature maps, to encode information of a fixed-size neighbour region into the single pixel in the feature maps. Another approach of one-stage instance segmentation is YOLACT [18, 19] (as shown in Figure 15), which applies a detection branch and a protonet branch to achieve segmentation on each objects' shape.

2.3.4 Deep-Learning in 3D Data

The representations of 3D sensory data include point clouds, voxel, mesh, multi-view images and so on [63]. The most commonly used representation of 3D sensory data are point clouds. Point clouds of objects or environment can be directly obtained by using 3D range sensors such as stereo-camera, RGB-D camera, and LiDAR. However, point clouds is highly sparse, unordered, and variant to the transformation. Based on different representation of data used by algorithms, deep-learning based network architectures on 3D data processing can be grouped into voxel-based models, multi-view based models, and point-based models.

a. Voxel-Based Models



(a) Network architecture of the 3D ShapeNet (b) Network architecture of the VoxelNet

Figure 16: Architecture of 3D shapeNet [168] and VoxelNet [188], which process the voxel-based representation of 3D data.

A straight forward approach to process point clouds is to voxelise the 2D or 3D point clouds into grids or voxels (as shown in Figure 16). Firstly, point clouds can be transformed to voxels based on give resolution conveniently. Secondly, Voxels can preserve spatial distribution of objects or scene

and can be processed by using 3D convolution operation. The limitation of voxels-based CNN is that such algorithms require significantly large computational resources and memory requirement to store and process the information within data when resolution become higher. The representative works of voxel-based deep-learning methods are 3D shapenets [168] and VoxelNet [188].

b. Multi-View Based Models

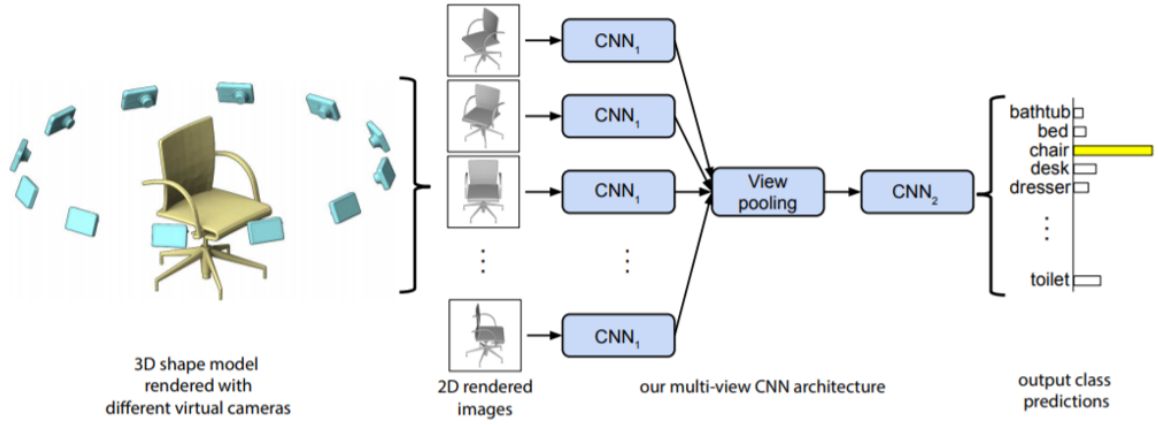


Figure 17: Architecture of multiview-CNN [147], which applies objects images from multiple view-angles and 2D CNN architecture to perform classification on 3D data.

Multi-view CNN methods [121, 147] use images of a object from multiple view-angle as input to perform objects classification (as shown in Figure 17). Multi-view CNN can directly applies CNN architectures which developed in 2D image processing to extract features from multi-view images. The limitation of multi-view based models is that occlusion of objects and choose of view-angle of camera can effect the performance of the algorithm.

c. Point-Based Models

Voxel-based models and multi-view models require conversion of representation before data is processing by algorithms. Point-based models can directly process the raw input point clouds from the 3D range sensors. PointNet [120]u ses a CNN to extract high-dimensional vectors on each of points (as shown in Figure 18), a asymmetric function to process unordered input, and a multi-layer perceptron classifier to use local and global Information. However, random sampling of point clouds cannot fully describe the details of the objects. Pointnet++ [122] further improves the sampling strategies on the point clouds and uses multi-resolution grouping to fuse the local and global information. Such

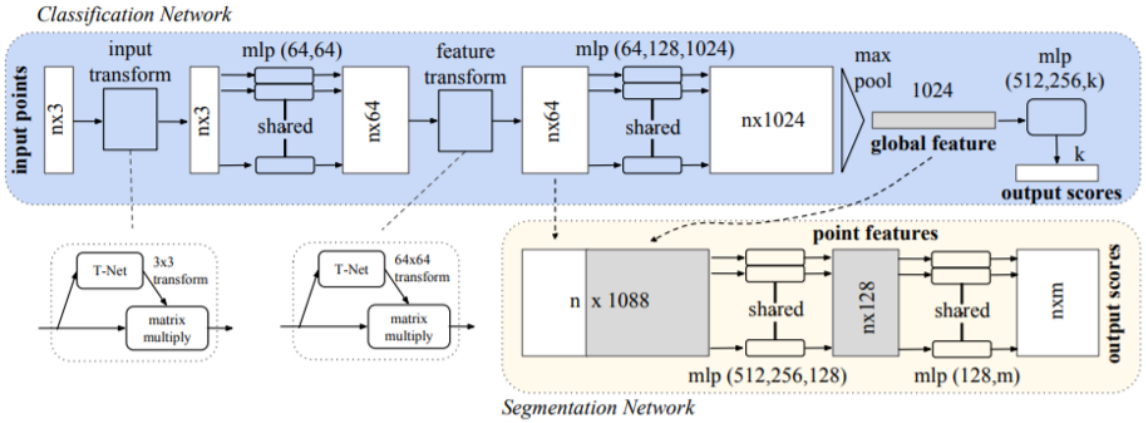


Figure 18: Architecture of Pointnet [120]. Pointnet applies a asymmetric function and 2D CNN architecture to process unordered point clouds.

measurements improve the performance of the network on classification accuracy, but increase the computational efficiency and slow down the running speed.

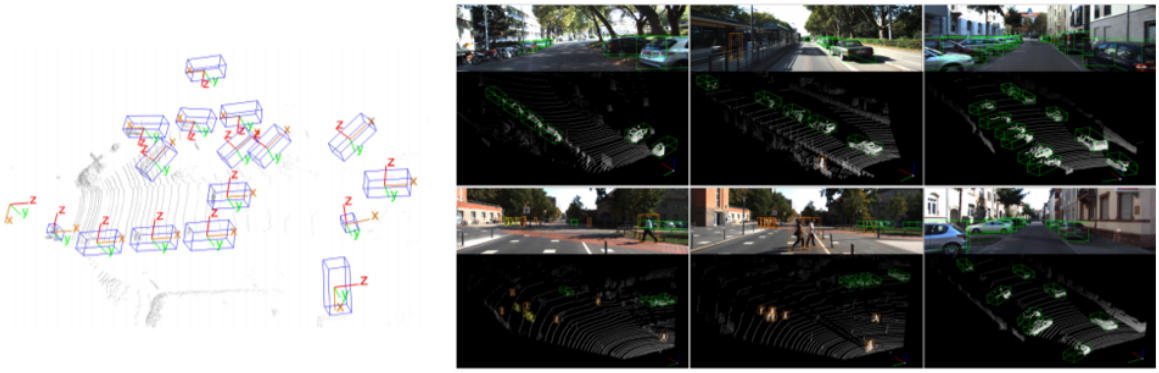


Figure 19: Pointnet based methods are also widely applied in LiDAR data processing in autonomous driving applications (results shown in image is from Frustum-convnet [164]).

Later work [102] further combines voxels-based methods and point-based methods to improve the computational efficiency and accuracy of model. PointNet based architectures are also applied in 3D object detection. Frustum PointNet [119] uses a 2D object detection to propose points of RoI and PointNets to perform OBB regression and instance segmentation (as shown in Figure 19).

2.3.5 Deep-Learning in Harvesting Robots

a. Adaptive Feature Learning on Pixel-Level Segmentation

Deep-learning based visual processing algorithm have been widely studied in many agricultural tasks, such as yield estimation, monitoring, and vision system for harvesting. Bargoti and Underwood [13]

applied sliding windows and Multi-Layer Perceptron (MLP) to perform pixel-level segmentation on input images, Circle Hough Transform (CHT) [8] and Watershed Segmentation (WS) [128] were used to perform detection on segmented results. Authors claimed this method as adaptive feature learning algorithm for fruit recognition, which improves the generalisation and accuracy of models when dealing with image data in uncontrolled environments. Similar framework is also applied in recognition system for almonds [59], apples [11, 60], and trunks [14]. Adaptive feature learning on pixel-level segmentation is the extension of traditional machine learning and multi-features based methods, while the hand-crafted features are replaced with automatic extracted features by the supervised learning [178]. The computational efficiency and accuracy of the models are still limited by the RoI searching strategies and shallow network architecture.

Semantic segmentation based network can be seem as an extension method of the adaptive-feature learning on pixel-level segmentation. Li et al. [91] applied FCN model in cotton segmentation. Lin et al. [93] applied FCN-8s network to perform semantic segmentation on images of including guava fruits and trees. The segmented results from the FCN-8s requires further processing (clustering) to estimate boundary and pose of each fruit. Kestur et al. [74] proposed a similar principle which applied FCN and post-processing to detect mango in orchard environments. Although pixel-level segmentation based on semantic segmentation can improve the accuracy on object segmentation, such methods still limited by performance of traditional post-processing techniques when dealing with densely arranged and occluded fruits.

b. Deep-Learning Architecture in Fruit Recognition

With the successful of deep-learning based applications in many computer vision tasks, deep-learning based methods are also adopted in fruit recognition tasks [70]. Sa et al. [134] adopted Faster-RCNN models on fused RGB and NIR images, to perform recognition of sweet pepper, rockmelon, apple, and avocado. Bargoti and Underwood [12] used network architecture of Faster-RCNN to perform detection of apple, almond, and mango by using RGB images, and accuracy of > 0.9 on F_1 score (see [70] for detail of F_1 score) of object detection was achieved. Liu et al. [101] applied a modified Faster-RCNN model to fuse images from RGB and NIR sensors and perform detection of kiwifruits. The modified Faster-RCNN on multi-source images achieved 0.907 on Average Precision (AP) (see [79] for details of AP evaluation) on kiwifruit detection. Yu et al. [36, 176] utilised a Mask-RCNN model to perform detection and instance segmentation of strawberry in unstructured environments. The au-

thor claimed that 95.78% on AP and 89.95% on instance segmentation accuracy were achieved in experiments. Except for two-stage detection networks, one-stage detection networks are also widely studied and applied by researchers in agriculture applications. Koirala et al. [78] developed a modified and light-weighted YOLO-based one-stage detection for yield estimation of mango in orchard environments and achieved 0.89 on F_1 score. Liang et al. [92] applied an SSD model in on-tree detection of mango. Experimental results showed that the SSD model achieves accurate detection performance (0.911 of F_1 score) and real-time running (35 Frame Per Second (FPS) on 400 x 400 (pixels) image). Tian et al. [154, 155] combined DenseNet [57] and YOLO-V3 model to perform monitoring of apple growth in different stages under orchard environments, the modified YOLO models showed better accuracy on fruit detection and growth-stages classification compared to the original YOLO-V3 model. Similar works of applying detection neural network architectures in fruit recognition can also be found in works [68, 77, 100, 145].

2.4 Recent Development on Harvesting Robots

A typical visual-guided robotic harvesting system is comprised by several sub-systems, including one or multiple robotic arms, end-effector, vision system, and computational device. This section reviews the classic works of robotic system developed in automatic fruit harvesting.

a. Robotic System for Strawberry Harvesting (Japan, 2010)

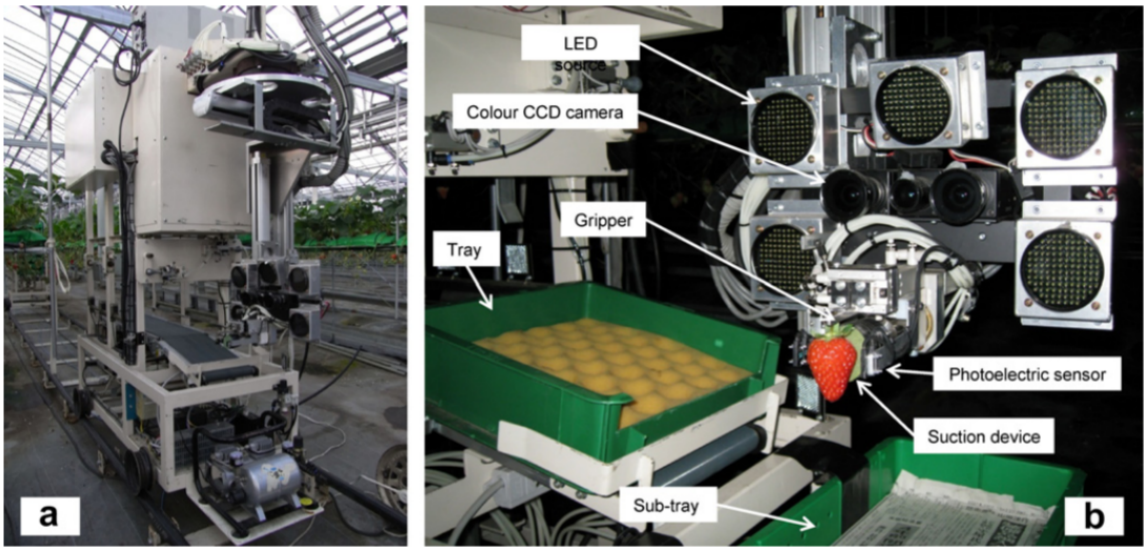


Figure 20: Robotic system for autonomous strawberries harvesting developed in Japan [51], 2010

Hayashi et al. [51] developed a robotic system for strawberry harvesting in greenhouse environments (as shown in Figure 20). The developed system includes a moving platform, a machine vision unit, an end-effector, and a central control unit. Machine vision unit applied a stereo-camera to localise the 3D position of the strawberries, the detection is performed by using colour information and decision tree in HSI colour space. The grasping of strawberries is performed by controlling manipulator translating towards the target. The detection rate of vision system in experiments is 60%, the success rate of harvesting is 41.3%, and the average running time of grasping one fruit is 11.5s.

b. Robotic System for Apple Harvesting (China, 2011)

Figure 21: Robotic system for autonomous apple harvesting developed in China [24], 2011

Zhao et al. [24] developed a visual-guided robotic apple harvesting system, which included a five Degree of Freedom (DoF) robotic arm, a multi-sensory end-effector, a vision processing and centre control system, and a moving platform (as shown in Figure 21). The multi-sensory gripper includes a pressure sensor, a collision sensor, and a RGB camera in eye-in-hand. Robotic system uses RGB images and a SVM with Radiu Basis Function (RBF) kernel function based classifier [67, 162] to perform fruit recognition and motion control on X and Y direction. Then end-effector is translated to the target until pressure sensor touch the fruit. The success rate of robotic system and average picking time of an apple are 77% and 15s, respectively.

c. Robotic System for Tomato Harvesting (China, 2015)

Feng et al. [32] developed a visual-guided system for tomato harvesting in greenhouse environments (as shown in Figure 22). The developed system includes a multi-sensors based vision system, a

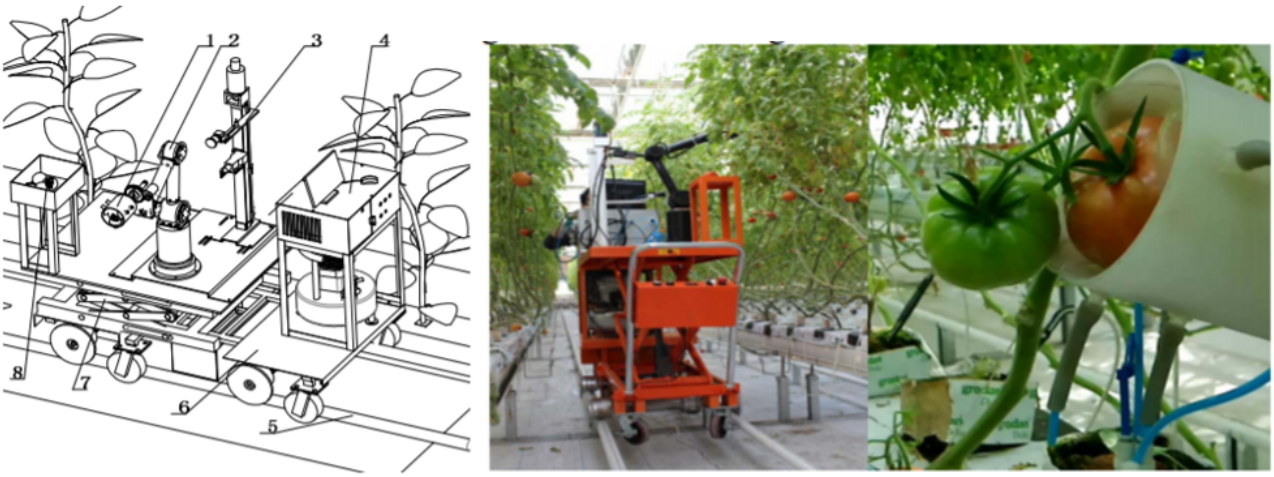
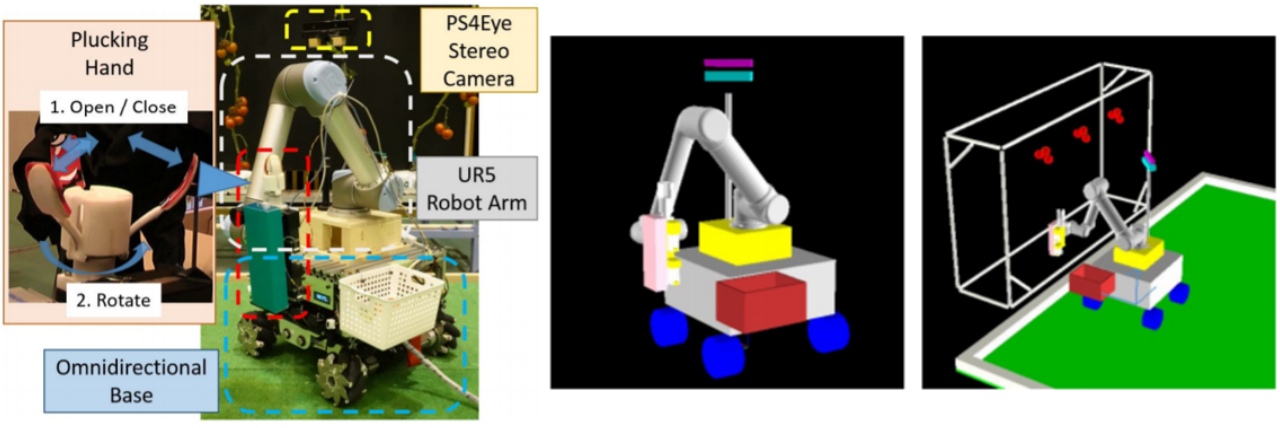


Figure 22: Robotic system for autonomous tomatoes harvesting developed in China [32], 2015

four-joints robotic arm, and a sleeve-shaped grasper. The multi-sensors vision system includes a RGB camera for fruit detection and a line laser generator for fruit localisation. The fruit recognition processing is performed by using a hand-crafted decision tree in HSI colour space. The success rate of harvesting in experiments was 83.9%, the average running speed of a picking iteration is 24s.

d. Robotic System for Tomato Harvesting (Japan, 2016)



(a) System design of the tomatoes robot (b) Robots and operation scene

Figure 23: Robotic system for autonomous tomatoes harvesting developed in Japan [173], 2016

Yaguchi et al. [173] developed a robotic system for automatic tomato harvesting in the 2nd tomato robot competition (as shown in Figure 23). This robotic system includes a commercialised robotic arm (Universal-Robot-5), a 3D stereo camera (Xtion PRO LIVE, ASUS), and a customised rotational plucking gripper. Fruit detection algorithm applies colour information of HSI colour space to perform

pixel-level segmentation and Random sample consensus (RANSAC) to search spherical shape on segmented point clouds. The grasping is achieved by translating robotic arm to the position of fruit centre. The success rate and average running speed of single iteration are 60% and 23s, respectively.

e. Robotic System for Sweet-Pepper Harvesting (Australia, 2017)

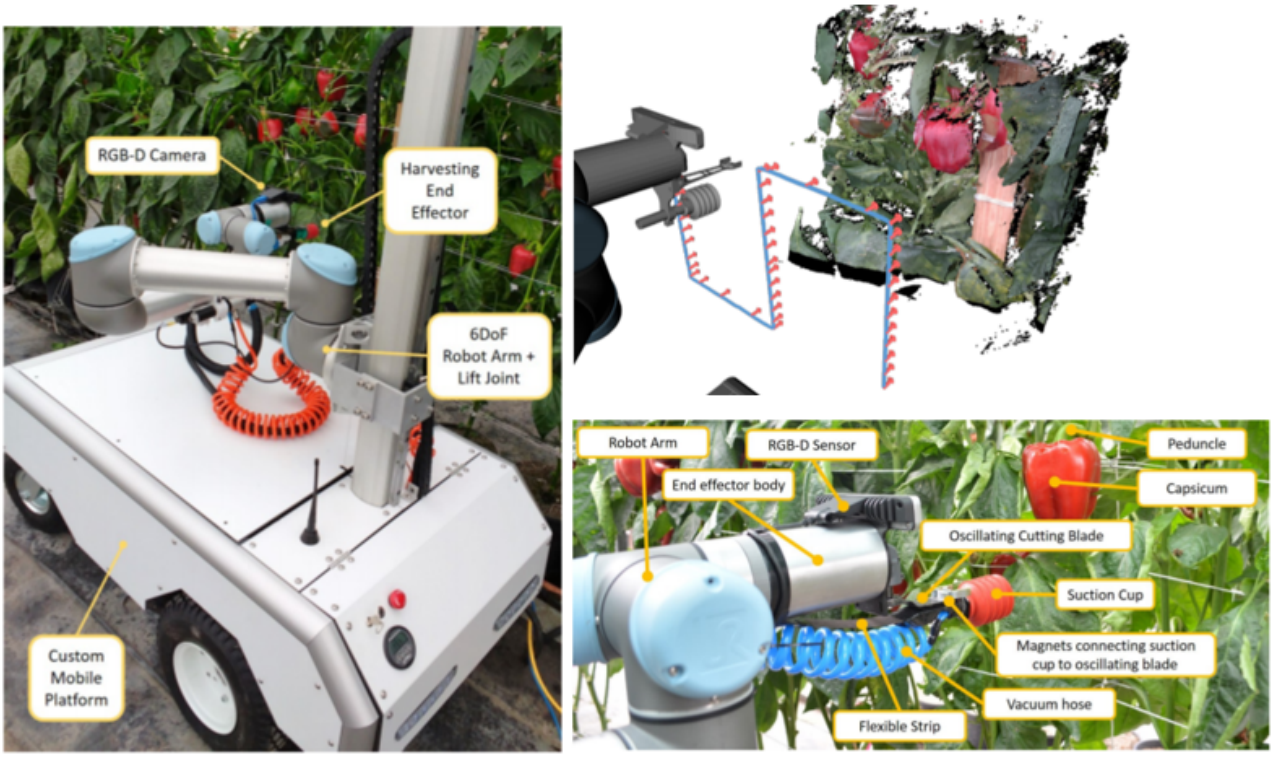
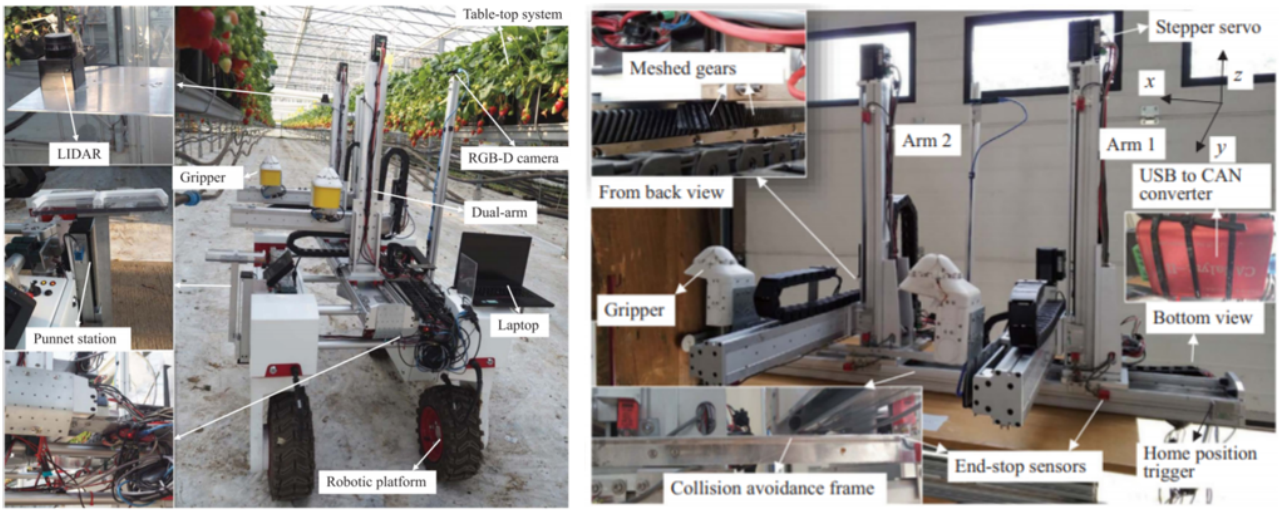


Figure 24: Robotic system for autonomous sweet pepper harvesting developed in Australia [86], 2017

Lehnert et al. [86] developed a robotic system for autonomous sweet pepper system in greenhouse environments (as shown in Figure 24). The developed system includes a small oving vehicle, a commercialised robotic arm (Universal-Robot-5) on a prismatic lift joint (Thomson LM80), a RGB-D camera (Intel Realsense SR300), and a customised end-effector. Vision system applies RGB information to perform fruit detection on coloured 3D cloud points. Vision system also estimates the proper grasping poses of each fruit rather than simple translating to the targets. The authors also explore a shape matching approach [87] to estimate proper grasping pose. In experiments, the developed system achieved 58% on success rate of robotic harvesting in modified crop.



(a) System design of the robot (b) Setup of the manipulator and gripper

Figure 25: Robotic system for autonomous strawberries harvesting developed in Norway [171], 2020

f. Robotic System for Strawberry Harvesting (Norway, 2020)

Xiong et al. [171] developed a automatic strawberry harvesting robot, which is comprised by a moving platform, vision system (RGB-D camera and 2D LiDAR), a duel-arm manipulator, and end-effector on each manipulator (as shown in Figure 25). Vision system of the robot applies a HSV colour space based processing algorithm for fast detection of strawberries in workspace. Vision and control system also include modelling the obstacle within workspace in scanning to avoid collision and harvest strawberries in order. Topological map of the farm is also included in the control system, which can guide the robot to the given position automatically. The developed robotic system achieved high success rate on harvesting and the average iteration time is 4.6s - 6.1s.

g. Robotic System for Sweet-Pepper Harvesting (Europe & Israel , 2020)

Arad et al. [4] developed a sweet pepper harvesting system, as shown in Figure 26. The developed robotic system includes a automated cart with a scissor lift, a Fanuc LR Mate 200iD robot arm, an end-effector with eye-in-hand RGB-D camera and LED lighting system. Robotic vision system applies artificial lighting to reduce environmental variances within workspace. A HSI colour space based fruit detection algorithm [5] is used to detect sweet peppers and stem within workspace. Robotic arm is controlled to move to several view-points around stem, to determine the proper angle of the fruits. The success rate of harvesting is 60% for the modified crop conditions and 18% in current crop conditions. The average single iteration time per fruit is 24s.

(a) System design of the robot (b) Led illumination and gripper design

Figure 26: Robotic system for autonomous sweet pepper harvesting developed by Europe & Israel [4], 2020

2.5 Challenges and Future of Vision System for Harvesting Robots

From the above reviews on recent development of the robotic harvesting system, we conclude that the vision system applied in autonomous harvesting applications is still limited in the following aspects:

1. Many of developed robotic system were designed to be operated in greenhouse environments for sweet pepper, strawberry, tomato, or cucumber. Traditional machine vision methods can perform well in indoor environments when algorithms are properly designed. However, for robotic harvesting of crops in orchard environments, such as apple and citrus, environmental variances can severely influence the accuracy and robustness of vision system.

2. Current robotic vision system requires to deploy multiple modules to perform multiple visual tasks, such as fruit detection, ripeness detection, stem segmentation, and grasping pose detection or estimation. Deploying of multiple algorithms increases requirements of computational resources and reduce operational efficiency. With development in deep-learning based vision processing algorithm, introducing of multi-functional network architectures can largely improves the efficiency and robustness of the system. For example, a single deep-learning network can perform fruit detection and instance segmentation, and semantic segmentation of scene within workspace.

3. The performance of current robotic harvesting systems is limited when performed in unmodified crop conditions. This is due to the limited capability of robotic vision system in environment modelling and understanding. In unmodified crop conditions, obstacles within workspace can block sight of visual sensor, fruits may also presented in dense clutter. Robotic vision sys-

tem is required to accurately detect the target fruits and modelling obstacles or other objects of interest within workspace. Robotic control system can rely on such information to plan a proper path for robotic arm to successfully harvest every fruits within the range.

Except of aforementioned challenges, robotic harvesting robots also requires improvements in adaptive end-effector and manipulator design, high-DoF robotic arm controlling, autonomous task planning, navigation, and driving of moving platform.

2.6 Conclusion

This chapter reviews the related techniques and works in the development of visual-guided harvesting robots, which includes visual sensors, vision processing algorithms, and the recent development of harvesting robots. The commonly used visual sensor includes 2D imaging sensors and 3D imaging or range sensors. 2D imaging sensor can obtain texture, colour, and shape information of the objects, while 3D sensors can further obtain spatial information of targets within the workspace. Vision processing algorithms include traditional machine-learning based methods and deep-learning based methods. Traditional machine-learning based methods apply hand-crafted features to encode objects' appearance and machine-learning based classifier to perform classification or detection. Comparatively, deep-learning based methods can extract and learn the features of objects by backward propagation training, which largely improves the accuracy, generalisation, and robustness of the algorithm in different conditions. We also compare different strategies which are applied in traditional machine-learning based methods and deep-learning based methods in multiple visual tasks, including object detection and segmentation. Moreover, we review the development of agricultural harvesting robots from 2010 to 2020. With the continuous advancements of techniques in vision processing, sensors, mechanism, and others, agricultural harvesting robots show a huge potential to play an important role in the future development in the agriculture industry. Recently, with the promotion of low-cost and advanced techniques in 3D range sensors, such as RGB-D cameras (Kinect-v2 and AZURE KINECT by Microsoft and RealSense series by Intel, a detail performance comparison between different RGB-D cameras is included in work [159]) and LiDAR, introducing of open source computer vision libraries in both traditional and deep-learning (such as TensorFlow and PyTorch), automatic harvesting technologies are experienced a dramatical development. The overall benefits of these technologies are promising for future smart farm to secure food production.

3 Fast Implementation of Real-time Fruit Detection in Apple Orchards Using Deep Learning

Fruit recognition in orchard environment is a challenge task since there are many variances presented in the working space. Recently, deep CNN has shown promising performance in many vision dependent agriculture applications, such as crop yield estimation, monitoring, and harvesting. However, CNN algorithms always requires manual labelling on large number of training data, which is a labour-intensive and time-consuming task. In this work, a fast implementation framework of CNN is developed and validated to perform fruit detection. The developed framework comprises a automatic image labelling algorithm and a one-stage detection network LedNet. The automatic image labelling algorithm utilises the multi-scale pyramid and clustering classifier to assist fast labelling of training data. LedNet applies FPN and ASPP to improve the performance of model. A modified light-weight backbone is used to improve computational efficiency of the network. Experimental results shows LedNet can achieve 0.821 and 0.853 on recall and accuracy on apple detection in orchard environment, respectively. The weights size and average running time of the LedNet on an 640*480 image by using Nvidia GTX-1070 are 7.4M and 28ms, respectively. The experiment results show that LedNet can perform real-time apple detection in the orchard robustly and efficiently.

Computers and Electronics in Agriculture 168 (2020) 105108

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Fast implementation of real-time fruit detection in apple orchards using deep learning

Hanwen Kang, Chao Chen*

Department of Mechanical and Aerospace Engineering, Monash University, Melbourne, Australia

ARTICLE INFO

Keywords:
Fruit detection
Deep learning
Real-time
Data labelling
Robotic harvesting

ABSTRACT

To perform robust and efficient fruit detection in orchards is challenging since there are a number of variances in the working environments. Recently, deep-learning have shown a promising performance in many visual-guided agriculture applications. However, deep-learning based approaches requires labelling on training data, which is a labour-intensive and time-consuming task. In this study, a fast implementation framework of a deep-learning based fruit detector for apple harvesting is developed. The developed framework comprises an auto label generation module and a deep-learning-based fruit detector 'LedNet'. The Label Generation algorithm utilises the multi-scale pyramid and clustering classifier to assist fast labelling of training data. LedNet adopts feature pyramid network and atrous spatial pyramid pooling to improve the detection performance of the model. A lightweight backbone is also developed and utilised to improve computational efficiency. From the experimental results, LedNet achieves 0.821 and 0.853 on recall and accuracy on apple detection in orchards, and its weights size and inference time are 7.4 M and 28 ms, respectively. The experimental results show that LedNet can perform real-time apple detection in orchards robustly and efficiently.

1. Introduction

Robotic fruits harvesting is one of the most challenging task in the automatic agriculture (Zhao et al., 2016). A typical fruit-harvesting robot comprises two subsystems: a vision system and manipulator system (Lehnert et al., 2016). The vision system detects and localises fruits and guides the manipulator to detach fruits from trees. However, a robust and efficient fruit detection algorithm in orchards is challenging as there are many variances such as illumination changing and occlusion between fruits, branches and leaves. Previous studies (Hashimoto, 2003; Kapach et al., 2012) have pointed out that a robust and efficient vision system is the key to the success of the robotic fruits harvesting.

In recent years, deep-learning has become state of the art in many tasks within computer vision, including image classification (He et al., 2016), segmentation (Wang et al., 2018), and object detection (Redmon and Farhadi, 2017). Compared to the traditional machine-learning approaches, deep-learning has strong adaptability to variances within the working scene (Kamilaris and Prenafeta-Boldu, 2018), making it a promising approach in many vision tasks. Deep-learning based object detection can be classified into two classes (Lin et al., 2017): two-stage detector and one-stage detector. The representative of the two-stage detectors is the Region Convolution Neural Network (RCNN), including RCNN (Girshick et al., 2014), Fast/Faster RCNN (Ren et al., 2015), and Mask RCNN (He et al., 2017). A RCNN model has two network branches: a Region Propose Network (RPN) branch and a classification branch. RPN proposes the Region Of Interest (ROI) of foreground class, while the classification branch classifies and estimates boundary box for each ROI. Compared to the traditional ROI searching strategies such as exhaust searching and selective searching (Uijlings et al., 2013), RPN makes the ROI searching a trainable task, improving the performance and computational efficiency of the model. The one-stage detector was developed more recently than the two-stage detector. It combines the RPN branch and classification branch into a single network, leading to more concise architecture and better computational efficiency. You Only Look Once (YOLO) (Redmon et al., 2016; Redmon and Farhadi, 2017, 2018) is one of the most representative aspects of the one-stage detector, it achieves state of the art performance in object detection with high computation speed. One-stage detector has been applied in many vision-based robotic tasks, such as automatic driving (Laroca et al., 2018), UAV monitoring (Tijtgat et al., 2017), and automation agriculture (Zhong et al., 2018).

Deep-learning based detection algorithms are gradually applied in sensing in agriculture environment. The authors of Sa et al. (2016) adopted the Faster-RCNN on the detection of multiple class fruits, including apple, sweet pepper, and melon. Faster-RCNN achieved a better

* Corresponding author.
E-mail addresses: hanwen.kang@monash.edu (H. Kang), chao.chen@monash.edu (C. Chen).

https://doi.org/10.1016/j.compag.2019.105108
Received 29 July 2019; Received in revised form 12 November 2019; Accepted 12 November 2019
Available online 27 November 2019

detection performance and running speed compared to the previous work (McCool et al., 2016). In Hao et al. (2016), the authors adopted the graph-based segmentation and Neural Network (NN) model to perform the maize tassel segmentation. In Bargoti and Underwood (2017), the authors utilised a convolution neural network to perform semantic segmentation of the apple fruit, then a Watershed Segmentation (WS) and Circular Hough Transform (CHT) algorithms are used to detect the apple. Later, the authors of Bargoti and Underwood (2017) utilised the Faster-RCNN model to perform the in-field fruit detection, and they achieved the detection of apple and mangos with F_1 score higher than 0.9. In Li et al. (2017), the authors utilised the Fully Convolution Network (FCN) to perform the semantic segmentation of cotton, and their results showed that FCN outperformed the traditional segmentation algorithms. The authors of Yang et al. (2019) developed an FPN strengthened Mask-RCNN in the strawberry detection under a non-structured environment, the good results in both detection and instance segmentation tasks are shown from their work. The authors of Majeed et al. (2018) adopted the segnet (Badrinarayanan et al., 2017) to perform the segmentation of the apple branch in the orchard, an accuracy of 0.93 on branch segmentation was reported in their work. Lin et al. (2019) used the FCN (Long et al., 2015) to perform the semantic segmentation of the guava fruit and branch, then estimating the grasping posture of the fruit based on the segmentation. Their result generated better results compared to the traditional vision algorithms. In Tian et al. (2019), a customized YOLO-V3 network was applied to apple detection. The authors modified the YOLO-V3 by using the DenseNet (Huang et al., 2017) to enhance the feature extraction ability, and the designed one-stage detector outperformed Faster-RCNN and original YOLO-V3 in their work. In Koirala et al. (2019), the authors adopted the YOLO architecture in the yield estimation of mango fruit, accurate detection performance was reported from their work. In addition, deep-learning is also being applied in many agriculture applications(Kamilaris and Prenafeta-Boldu, 2018), such as yield estimation using remote sensing (Kussul et al., 2017) and crop monitoring using UAV (Yang et al., 2019).

The deep-learning based algorithms rely on backward propagation to train its parameters (Srivastava et al., 2015). With a proper architecture and training mechanism, the deep-learning model can fit the training data in good approximation and generalization. However, a significant difference between working scene and training data may lead to poor performance of the trained model (Shin et al., 2016). Transfer learning, which applies field data to adjust the pre-trained network to fit a specific task, is widely used in many applications (Weiss and Khoshgoftaar, 2016). However, labelling on training data is a time-consuming and labour-intensive task (Papandreou et al., 2015). Therefore, self-labelled algorithms which can programmatically generate the label on the training data has became an important issue in the implementation of the deep-learning approaches(Ratner et al., 2017).

In this study, a fast implementation framework of deep-learning based fruit detector is developed. This framework includes two components: an auto label generation module and a one-stage detector LedNet. The auto label generation module is used to accelerate the labelling on training data, the LedNet is used to perform the real-time detection of fruits in orchards. The pipeline of the framework is shown in Fig. 1. The auto label generation is achieved by a Clustering-RCNN (C-RCNN) algorithm to perform the quasi-good label prediction. The LedNet utilises the Feature Pyramid Network (FPN) and Atrous Spatial Pyramid Pooling (ASPP) to enhance the detection performance. Meanwhile, to improve computation efficiency of the model, a light-weight backbone is developed in this work.

The rest of the paper is organised as follows. Section 2 and Section 3 introduce the label generation method and detector LedNet, respectively. Section 4 discusses the experiments result. Finally, the conclusions and future work are presented.

2. Auto label generation

C-RCNN adopts the principle of the RCNN, separating the detection task into ROI proposal and classification/regression, which are introduced in Section 2.1 and 2.2, respectively.

2.1. Selective searching for ROI proposing

Segmentation on Multi-level Pyramid: Input image is scaled to $\frac{1}{4}$, $\frac{1}{8}$ and $\frac{1}{16}$ to form the multi-level pyramid in terms of improving the detection performance of objects in different scales. Colour Coherence Vector (CCV) (Pass et al., 1996) and Histogram of Gradient (HoG) (Ng and Henikoff, 2003) are utilised to encode colour and geometry features of objects, which are denoted as V_{ccv} and V_{hog}, respectively. In the forward inference, the concatenated feature vector $V_o = [V_{ccv}, V_{hog}]$ of length N is calculated, producing a $H \times W \times N$ feature map on each pyramid level. H and W are the width and height of feature maps, respectively. Assuming there are m-classes $C = [c_1, c_2, ..., c_m]$ and the probability of a feature vector V_o belongs to the class i is donated as $p(V_o|c_1)$. The classifier assigns the V_o to the class i with the highest probability, as follow:

$$\textbf{index}\ i = argmax(p(V_o|c_1), p(V_o|c_2), ..., p(V_o|c_m)) \tag{1}$$

Considering there are multiple distribution kernels within each object class, the Gaussian Mixture Models (GMM) is utilised to model the internal distribution within each class. Assuming there are k-number distributions $D = [d_1, d_2, ..., d_k]$ in the class C_i, the probability that a feature vector V_o belongs to the class C_i is:

$$p(V_o|C_i) = max(p(V_o|d_1), p(V_o|d_2), ..., p(V_o|d_k)) \tag{2}$$

The training of GMM follows the work (Greenspan et al., 2001).

Centre Detection: (1) and (2) are used to perform segmentation on feature maps of each pyramid level. Fruits are assigned as foreground class, other objects are assigned as background classes. Then, the centre detection algorithm is utilised to search the centre of each ROI. Firstly, pixel-connection is used to segment the foreground pixel into Independent Candidate Patch (ICP). Each pyramid level accepts ICP with an acceptable number of foreground pixels (for example, 250–2000). To deal with fruits under occlusion conditions, the CHT is utilised to perform centre detection within each ICP. The workflow of the centre detection is shown in Fig. 2. Finally, a pre-set anchor box is assigned to each centre as output ROIs.

2.2. ROI Post-processing

ROI Classification: Since clustering-based classifier has limited accuracy in the object classification. To increase the accuracy of proposed ROIs, a modified resnet network is utilised, which is introduced in Section 3.2. A resize operation on proposed ROIs is performed before classification. Then, the ROI which is classified as not belong to the fruit class will be deleted from the ROI list.

ROI Regression: boundary box regression can be recast as a template matching, which is expressed in (3). That is, given a template and flowing mask A and B. The template mask A is the average appearance of the apple. K stands the Normalised Cross-Correlation (NCC) value (Yoo and Han, 2009) between template and flowing masks. Eq. (3) estimates the scale S, rotation matrix R, and offset T to perform the matching.

$$\max\{K\}, \quad K = NCC(A, S * (R * B + T)) \tag{3}$$

R is a 2×2 identity matrix as no rotation applied in the matching, while the NCC is expressed as follows:

$$NCC(f(x,y), t(x,y)) = \frac{\sum_{x,y}[f(x,y) - \bar{f}][t(x,y) - \bar{t}]}{\sqrt{\sum_{x,y}[f(x,y) - \bar{f}]^2 \sum_{x,y}[t(x,y) - \bar{t}]^2}} \tag{4}$$

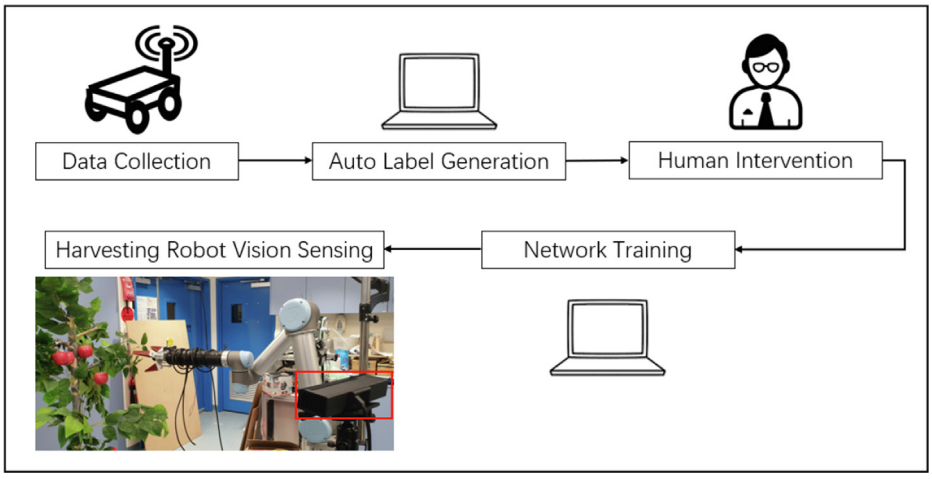


Fig. 1. Pipeline of the framework, from data collection, auto label generation to the detector training.

In the (4), $f(x, y)$ and $t(x, y)$ the template and flowing mask are shown. $\bar{f}$ and $\bar{t}$ are the mean of the $f(x, y)$ and $t(x, y)$, respectively. Eq. (3) is solved by local searching of the S and T to maximise the NCC between the template and mask. The NCC value between the template and flowing masks is returned as the confidence score for each ROI.

3. LedNet: network model

3.1. Network architecture

Multi-level Feature Fusion: Compared to the YOLO-V1 (Redmon et al., 2016) and YOLO-V2 (Redmon and Farhadi, 2017), YOLO-V3 (Redmon and Farhadi, 2018) improves the detection performance on objects of different scales by using FPN. Different levels of network always comprise different information. For example, feature maps on a lower level and higher level of the network comprise spatial and semantic information of objects, respectively. FPN can fuse the features of objects from different levels of the network to improve the feature extraction ability of the model on detection. LedNet utilises a three-level FPN to process feature maps from C3 (1/8), C4 (1/16) and C5 (1/32) of the backbone, which is shown in Fig. 3. Feature maps from different levels are fused by using an adding operation. Each level of FPN is used to detect objects within a specific scale range. For example, C5 level and C3 level of the LedNet detect objects in large scale and small scale, respectively. Each level of FPN has two subnets for boundary box regression and classification. The design of regression subnet follows the work in (Redmon and Farhadi, 2018) and two anchor-boxes are utilised on each level of FPN. (see Figs. 4–6).

Multi-scale Feature Fusion: One difference between the one-stage detector and the two-stage detector is the way of encoding features of ROIs. The two-stage detector uses the corresponding area of ROIs on feature maps to perform the classification and boundary box estimation. The one-stage detector encodes features of ROIs by using the fixed size convolution kernel. However, such processing cannot properly cover the corresponding area of ROIs, as the fixed size convolution kernel may over-cover or under-cover the area of ROIs. LedNet utilises the ASPP to encode the properly area of ROIs for the following classification and boundary box estimation. The ASPP is developed and utilised in GoogLeNet (Szegedy et al., 2015) for classification and DeepLab (Szegedy et al., 2016) for segmentation. The principle of the ASPP is to use dilation convolution kernel with different rate to encode the multi-scale features into feature maps. In the LedNet, an ASPP which includes a 1x1 convolution kernel, three 3x3 dilation convolutions (dilation rates are 1,3,6), and a 3 x 3 max-pooling operation is applied. Three dilation convolution kernels are used to cover the possible area of ROIs on feature maps. The max-pooling layer in the ASPP block aims to amplify the signal of small objects, to increase the recall of such objects in the detection. From the experiment, max-pooling operation may also introduce noise in the detection. Hence, a 3 x 3 convolution kernel is applied after max-pooling layer to filter such noises.

Backbone: LedNet can use different classification networks as a backbone, such as resnet and Darknet. To improve computation efficiency of the model on the embedded computing device, a light-weight backbone is developed, which is included in the following section.

3.2. Light-weight backbone

The light-weight backbone LW-net has nine bottleneck resnet blocks and five down-sampling blocks. Both the resnet block and down-sampling block adopt the residual shortcut design. The shortcut of the

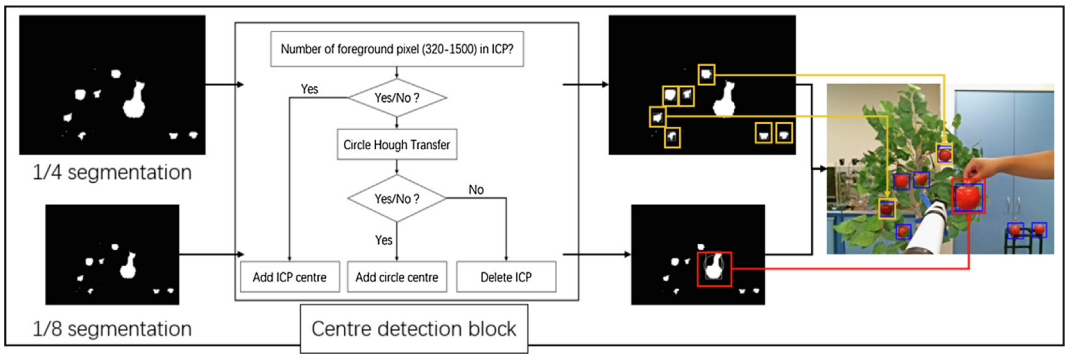


Fig. 2. centre detection algorithm for detecting extracting the potential centre from the segmentation. Yellow box is detected from 1/4 scale segmentation, while the red box is from the 1/8 segmentation. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

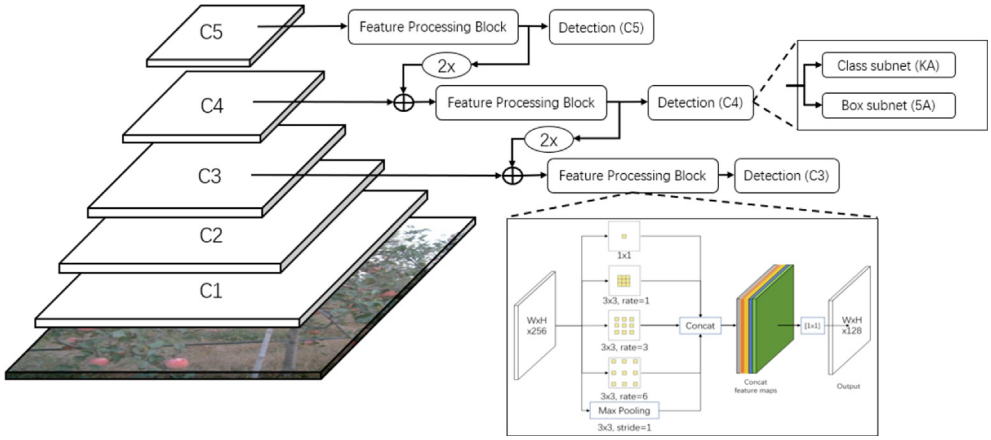


Fig. 3. The architecture of the LedNet, it utilises the 3-levels FPN and ASPP is used in the feature processing block. A and K are the number of pre-set anchor-box on each pyramid and object classes, respectively.

resnet block pass the information of the input feature map, while the shortcut of the down-sampling blocks pass the information of the feature tensor which is processed by the max-pooling of the input feature maps. The stride of the second convolution layer in down-sampling block is 2 to keep the consistency of the size of the feature map. The down-sampling block is used to replace the max-pooling layer to minimise the information loss during down-sampling. The LW-net is pre-trained with Cifar-10/100. The total weights size of the LW-net is 3.46 MB, and the validation accuracy of the model on the Cifar-10 and Cifar-100 is 92.7% and 71.6%, respectively.

3.3. Data processing and training

3.3.1. Data acquisition

800 images are collected from the orchard at Qingdao, China by using Kinect-v2. Another 300 images of apples in different scenes are collected to increase the diversity of the training data. In addition, 100 images of scenes without apples are also included in the training data. In total, 800 images are used as training data, while the remaining images are used for validation.

3.3.2. Data Augmentation

The distance between camera to apple trees is between 0.5 and 1.5 m. The average size of Fuji apples are between 80 mm and 100 mm. Therefore, most of the apples are presented as small objects in the training data. To avoid under-fitting of the model leaded by imbalance distribution of object scale, a objects amplification algorithm is utilised. Firstly, a patch (round 160–480 pixels) is cropped from the image and resized to the training resolution under possibility of 0.5. Then, this image will take another possibility of 0.5 to repeat the previous procedure. The distribution of object scale in training data before and after augmentation is shown in Table 1. The training resolution is 320 × 320, to increase number of images in each training batch. Other augmentation methods, including saturation, brightness, contrast, rotation, and flip are also utilised.

3.3.3. Focal-loss training

The training loss of LedNet includes three terms: confidence score, boundary box regression, and classification. The boundary box regression and classification follow the same equation which are used in the YOLO-v3, while the training of the confidence score utilise the focal-loss and MSE in the training (for a confidence score of objects of the foreground-class and background-class, respectively), as follows:

$$Loss_{obj} = \sum^{obj} -\alpha(1-p_t)^{\gamma} log(p_t) + \sum^{noobj} \beta(p_t)^2 \quad (5)$$

p_t and γ are confidence score and focusing parameter, respectively. α and β are the weights of the foreground-class term and background-class term, respectively. The distribution of the loss value of Mean

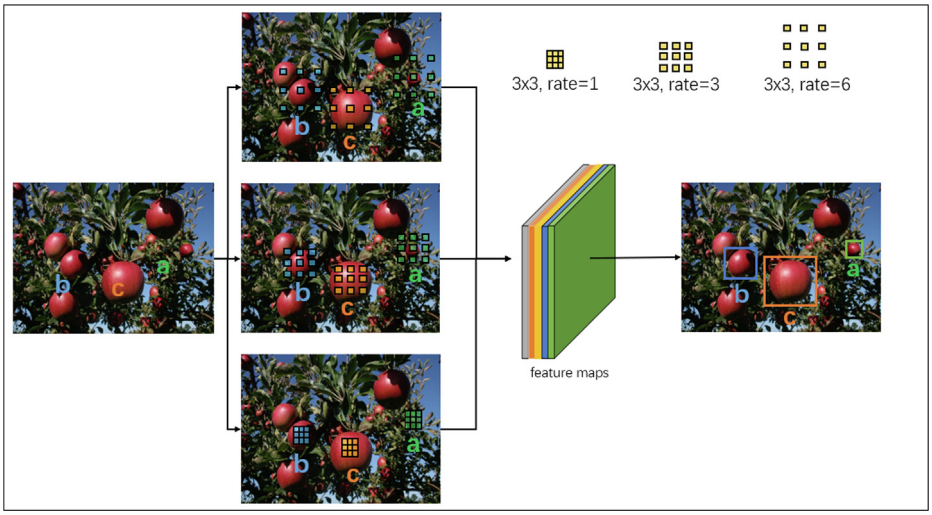


Fig. 4. Demonstration of ASPP of multi-scale feature fusion. Dilation convolution kernels with different rate are used to cover the ROI region to improve the feature extraction ability.

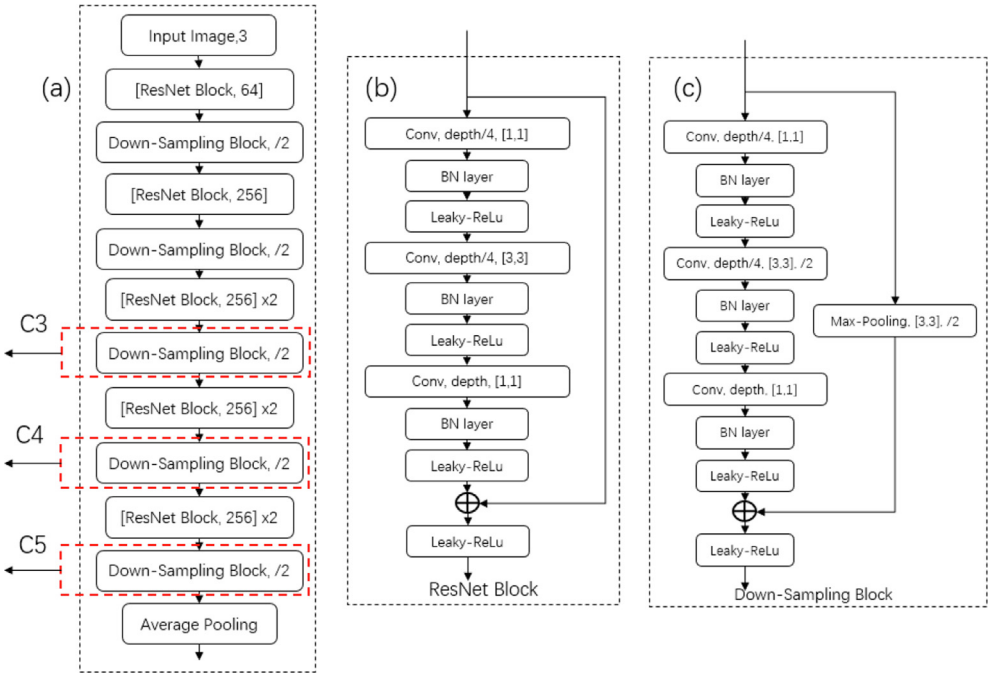


Fig. 5. Architecture of LW-net (a), it has 9 resnet blocks (b) and 5 down-sampling blocks (c).

Square Error (MSE) and focal-loss with different value of γ along the p_t are shown in Fig. 7. For foreground-class objects which have a predicted value of p_t larger than 0.5 are well-classified samples. In this region, focal loss has similar loss value compared to the MSE. When foreground-class objects are in the under-classified region ($p_t < 0.5$), focal-loss provides a much larger loss compared to the MSE. MSE is utilised to train the confidence score of background-class objects, as extremely imbalance between the number of foreground-class objects and background-class objects is presented in the training of the one-stage detector. α, β and γ are set as 1,0.5 and 2 based on the experiment results. The Adam-optimiser is applied and the learning rate and decay rate of optimiser are 0.001 and 0.9/epoch, respectively.

Table 1
Distribution of different scale of object in the training dataset.

Iteration	Augmentation	Small	Median	Large
150 epochs	Yes	41%	38%	21%
150 epochs	No	62%	30%	8%

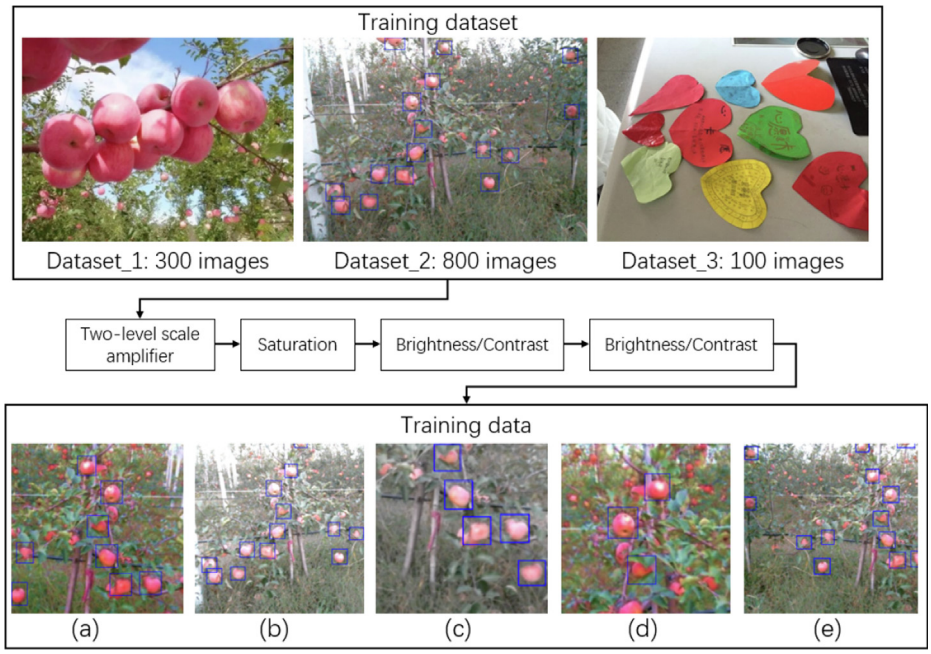


Fig. 6. Training data is comprised by dataset-1 (supplement data), dataset-2 (orchard data), and dataset-3 (other scenes). The data augmentation results are shown in (a)–(e).

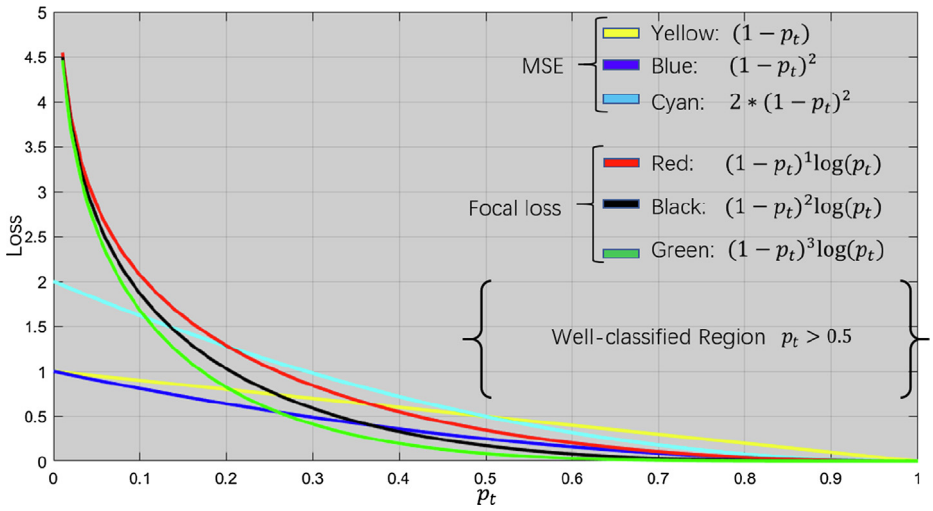


Fig. 7. Loss value of focal loss function and MSE function along the object confidence score p_t.

4. Experiment and discussion

4.1. Evaluation methods

In the experiment, AP is applied as the evaluation index. The AP evaluation includes several important indexes: Intersection of Union (IoU), *Precision*, and *Recall*. IoU calculates the overlap ratio between the boundary box of the prediction (*pred*) and ground-truth (*gt*). *Precision* measures the accuracy of the prediction, while the *Recall* measures how good the detector finds all *gt*. The formulation of the above three indexes is as follows.

$$IoU = \frac{Area_{pred} \cap Area_{gt}}{Area_{pred} \cup Area_{gt}} \tag{6}$$

$$Precision = \frac{TruePositive(TP)}{TruePositive(TP) + FalsePositive(FP)} \tag{7}$$

$$Recall = \frac{TruePositive(TP)}{TruePositive(TP) + FalseNegative(FN)} \tag{8}$$

The Precision-Recall curve forms the $P-R$ curve and the Area Under Curve (AUC) is the *AP* value. The *m* in the AP_m is stand the threshold(%) of IoU value between *pred* and *gt*. Another commonly used evaluation index F_1 score is also used in the evaluation, as formulated below.

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{9}$$

4.2. Experiment on auto label generation

4.2.1. Implementation detail

C-RCNN algorithm requires sampling to train the classifier for segmentation on the multi-level pyramid for ROI proposal. The foreground objects include apple, leaf, branch and land, and the threshold for being background is 0.3. That is, the region will be classified as background when the possibility of this region belonging to any foreground objects is lower than 0.3. The data sampling is to choose a pixel within a foreground objects. Then, the corresponding feature vector of the neighbour region (for example, 48 x 48) of this pixel will be calculated and saved for the training. Each class of foreground object requires 50 to 100 samples to train the classifier.

4.2.2. Performance evaluation

150 images are randomly selected to perform the evaluation of the C-RCNN algorithm. The evaluation results are shown in Table 2 and Fig. 8.

(a) to (c) of Figs. 8 show the labels generated by the C-RCNN on the orchard data. variances of illumination conditions and fruit appearances are the major factors that affect the performance of the label generation. Uneven-light minimising and colour enhancement measurements are utilised to process the images. On the orchard images, C-RCNN achieves 73% and 66% on the recall and accuracy, respectively. The average IoU of the generated label is 69%. (d) to (f) of Figs. 8 show the labels generated by the C-RCNN on the supplement data. There are many human-made objects in the supplement images data which cannot be fully included in the training. Therefore, to perform label generation on those image data is more complicated than the case on orchard data. In this condition, there is a considerable degeneration on the performance of the C-RCNN algorithm in the label generation. The recall, accuracy and average IoU of the generated labels are 68%, 62% and 63%, respectively. The computational time of the C-RCNN on each image is between 0.5 and 1.8s, depending on numbers of apples in images. The task with the highest computational consumption is the segmentation on the multi-level pyramid for ROI proposal, which takes 0.4–1.4 s for processing.

Table 2
Performance evaluation of Auto Label Generation.

Dataset	F_1	Recall	Accuracy	IoU	Mean time
Orchard	0.68	73%	66%	69%	0.8–1.6 s
Supplement	0.63	68%	62%	63%	0.5–1.8 s

From the experimental results, C-RCNN can perform well in the orchard data, but its performance shows a degeneration when applied in the supplement data. Some approaches can be utilised to improve the performance of the C-RCNN, such as including more objects in training and applying more feature descriptors in the segmentation. However, such measurements would increase the complexity of the algorithm. The average number of apples within an image in the training data is between 10 and 25. Therefore, the total number of apples in the training data is between 10,000 and 25,000. Manual labelling of all apples in the training data is labour-intensive and time-consuming. With the assistance of the C-RCNN algorithm, labelling of training data is accomplished within two days.

4.3. Self comparison on LedNet

Influence on Data Augmentation: Data augmentation is important in network training. Two different data augmentation methods are utilised to evaluate the influence of the data augmentation to the detection performance. The first method 'method-A' is introduced in Section 3.3.2, it utilises two-level scale amplification to balance the

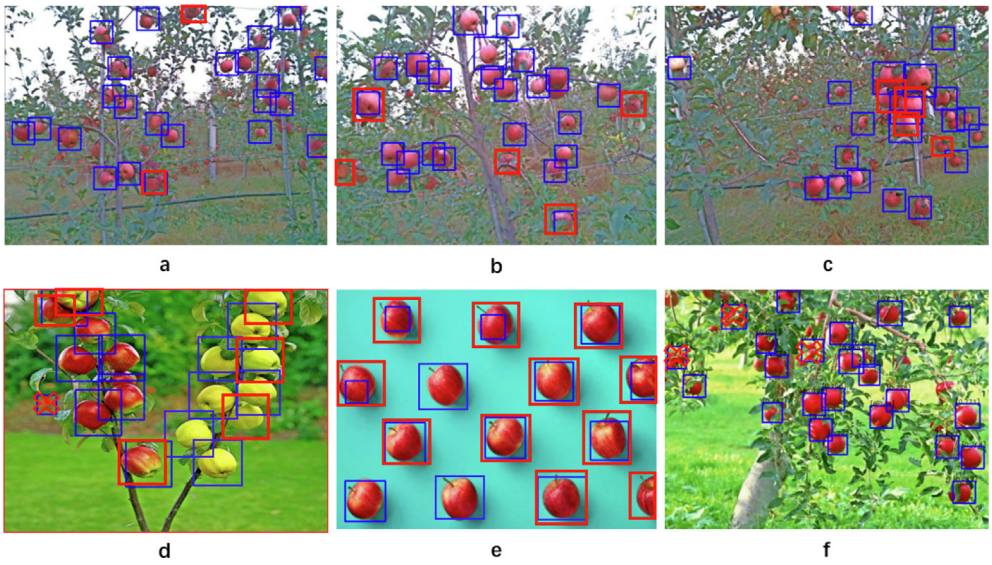


Fig. 8. Label prediction result using C-RCNN algorithm. Blue boxes are detected by the C-RCNN algorithm, red boxes and cross are the examples of adjustment of boundary-box and deletion after manual revision. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 3
Evaluation on different augmentation methods.

Methods	AP_{50}	AP_{small}	AP_{median}	AP_{large}	IoU
method-A	0.826	0.832	0.817	0.763	86.7%
method-B	0.797	0.818	0.778	0.652	78.3%

Table 4
Evaluation on different loss functions.

Loss function	AP_{50}	$F1$	Recall	Accuracy	IoU
Focal loss	0.826	0.832	0.82	0.85	86.3%
MSE	0.811	0.816	0.817	0.831	85.4%

Table 5
Evaluation on different backbones.

Models	AP_{50}	$F1$	Recall	Accuracy	IoU
LedNet (LW-net)	0.826	0.834	0.821	0.853	86.3%
LedNet (resnet-50)	0.834	0.84	0.833	0.854	86.4%
LedNet (resnet-101)	0.843	0.849	0.841	0.864	87.2%
LedNet (darknet-53)	0.833	0.842	0.83	0.857	86.3%

Table 6
Computation time and weights size of the LedNet with different backbones (on GTX-1080Ti).

Model	Inference time	weights size
LedNet (LW-net)	28 ms	7.4 M
LedNet (resnet-50)	38 ms	112 M
LedNet (resnet-101)	46 ms	188 M
LedNet (darknet-53)	36 ms	176 M

Table 7
Detection performance comparison between different detectors.

Model	AP_{50}	F_1	Recall	Accuracy	IoU
LedNet (LW-net)	0.826	0.834	0.821	0.853	86.3%
LedNet (resnet-101)	0.843	0.849	0.841	0.864	87.2%
YOLO-V3	0.803	0.803	0.801	0.82	84.2%
YOLO-V3 (Tiny)	0.782	0.783	0.776	0.796	82.4%
Faster-RCNN (VGG)	0.814	0.818	0.814	0.835	86.3%

distribution of object scale in training data. The second method 'method-B' utilises usual data augmentation method to train the network. The performance of LedNet by using different data augmentation methods is shown in Table 3.

The performance of the LedNet trained with 'method-A' is significantly better than the LedNet trained with 'method-B', which are 0.826 and 0.797 on AP_{50}, respectively. Meanwhile, LedNet trained with 'method-A' achieves a balance performance on the detection of objects in different scales. Comparably, the detection performance of the LedNet trained with 'method-B' shows a considerable reduction on the detection of objects in the median and large scale. The LedNet trained with 'method-A' has higher accuracy in boundary box localisation than the LedNet trained with 'method-B', which are 86.7% and 78.3%, respectively.

Influence on Training Loss: This experiment compares the performance of LedNet by training with focal loss and MSE. The evaluation results are shown in Table 4.

From the experiment, the LedNet trained with focal loss achieves better performance than the network trained with MSE. The recall and accuracy of the LedNet trained with focal loss are 0.82 and 0.85, respectively. Comparatively, the recall and accuracy of LedNet trained with MSE are 0.817 and 0.831, respectively. Overall, the LedNet trained with focal loss achieves 0.832 on F_1 score, which is 1.6% higher than the LedNet trained MSE.

Influence on Backbone: LedNet can utilise different networks as the backbone. This experiment compares the performance of the LedNet with different backbones, including LW-net, resnet-50, resnet-101, and darknet-53. The evaluation results of comparison in detection performance and computational efficiency are shown in Tables 5 and 6, respectively.

Table 5 shows that LedNet with resnet-101 outperforms the other models, it achieves 0.849, 0.84 and 0.86 on $F1$ score, accuracy and recall, respectively. Meanwhile, LedNet with resnet-101 also achieves higher accuracy on boundary box localisation compared to the other models. The experimental results indicate that the better performance of the backbone can improve the detection performance of the model. On the other hand, LedNet with light-weight backbone LW-net shows a balance performance on fruit detection and computational efficiency. It achieves 0.834, 0.821, and 0.853 on $F1$ score, recall, and accuracy, respectively. The weight size and computational time of the LedNet with LW-net are 7.4 M and 28 ms. Comparably, the weight size and computational time of the LedNet with resnet-101 are 188 M and 46 ms. The LedNet with resnet-101 and LW-net are applied in the comparison to state of the art as they outperform in comparison of

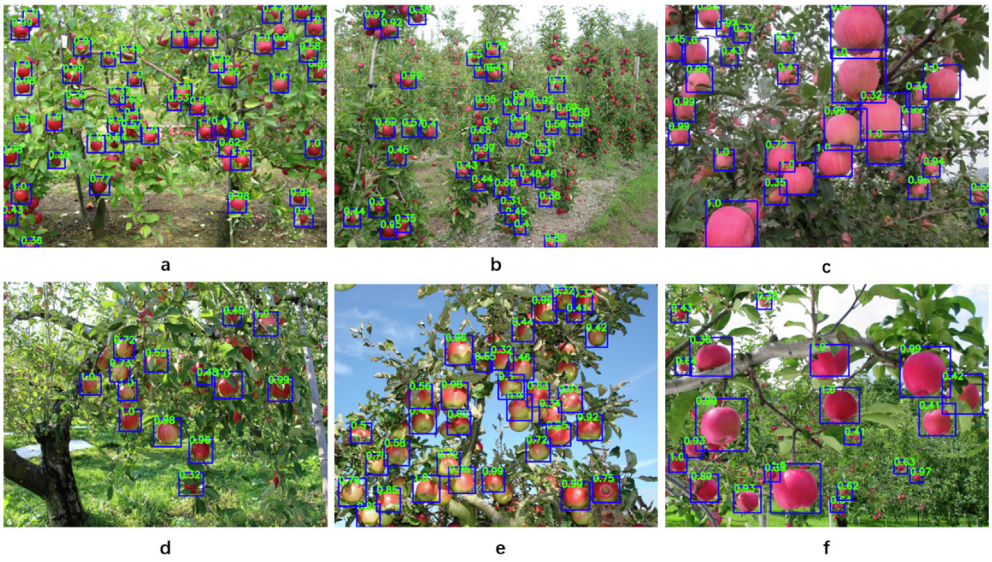


Fig. 9. Detection results of apple using LedNet(LW-net) in supplement dataset (green number is the confidence score). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

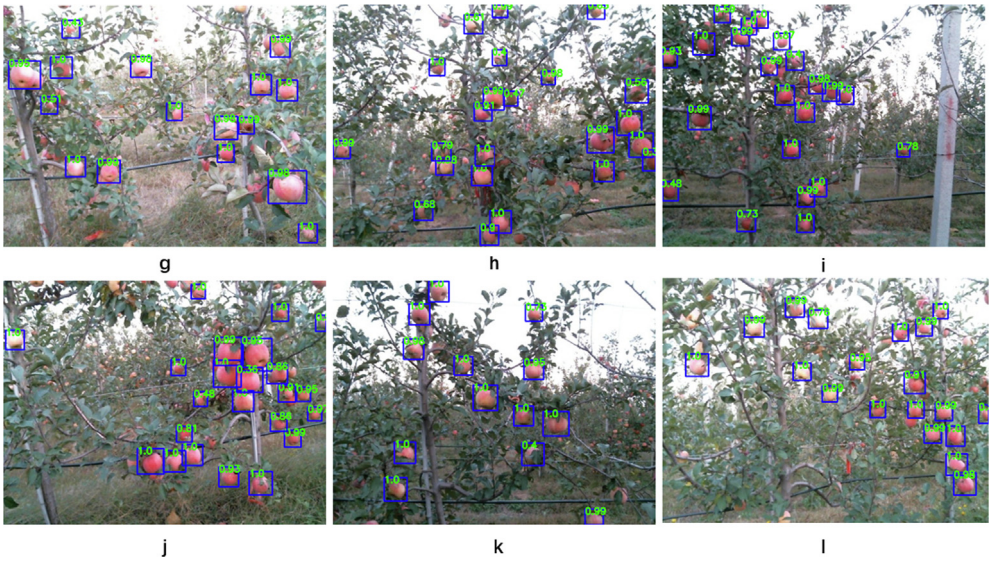


Fig. 10. Detection results of apple using LedNet(LW-net) in orchard dataset (green number is the confidence score). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 8
Computation time and weights size of different models on GTX-1080TI (resolution of images are 640 x 480)

Model	Average time	weights size
LedNet (LW-net)	28 ms	7.4 M
LedNet (resnet-101)	46 ms	188 M
YOLO-V3	45 ms	248 M
YOLO-V3 (Tiny)	30 ms	35.4 M
Faster-RCNN (VGG)	145 ms	533 M

detection performance and computational efficiency, respectively.

4.4. Comparison to state of the art

Performance Evaluation: A comparison between LedNet, YOLO-V3, YOLO-V3(tiny), and Faster-RCNN is included in this experiment. YOLO-V3 is the state of the art in the one-stage detector. YOLO-V3 uses FPN in the network architecture to improve the detection performance on different scale objects. YOLO-V3 (Tiny) is the light-weight version of the YOLO-V3, which utilises a light-weight backbone and 2-level FPN to improve the real-time detection performance. On the other hand, Faster-RCNN is state of the art in the two-stage detector. Compared to the one-stage detector, Faster R-CNN has better detection performance while its computation efficiency is lower than the YOLO-V3 as it comprises two tasks within the detection. The performance of the different detectors are shown in Table 7. Figs. 9 and 10 show the detection on the validation dataset by using LedNet (LW-net).

From the experimental results, the two-stage detector Faster-RCNN outperforms the one-stage detector YOLO-V3 in apple detection, including the areas of recall, accuracy, and boundary box localisation accuracy. The experimental results indicate that RPN allows Faster-RCNN to encode information about objects from the proper area of the ROI within the feature maps, leading to a better detection performance compared to the YOLO-V3 model. LedNet with resnet-101 achieves 0.849, 0.841, and 0.864 on F_1 score, recall, and accuracy of the detection, respectively. The evaluation results show that LedNet outperforms the YOLO-V3 and Faster-RCNN in the evaluation. With the

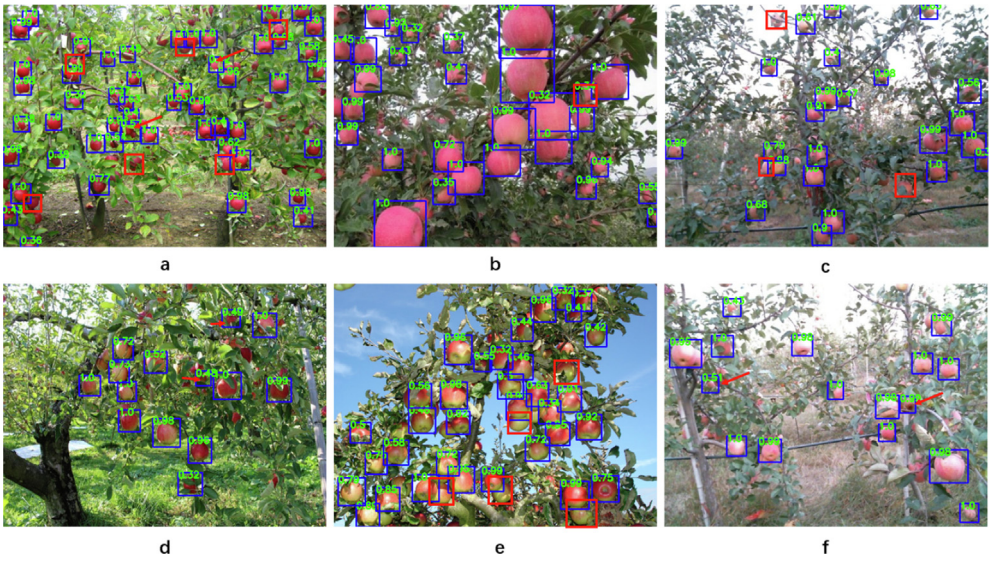


Fig. 11. Detection of apple in occlusion and overlapping conditions.

introduction of the ASPP, LedNet can encode the information of objects from the multi-scale behaviour to improve the detection performance of the network model. Meanwhile, LedNet utilises the more powerful backbone such as resnet-101, to achieve a better performance compared to the Faster R-CNN in fruit detection of apples in orchards.

Computation Efficiency: The comparison of computation efficiency between different detectors is shown in Table 8. The weights size and computational time of the LedNet(LW-net) on an image (640 x 480) with GTX-1080Ti are 7.4 M and 28 ms, respectively. YOLO-V3 (tiny) achieves similar results; the weights size and computational time of YOLO-V3 (tiny) is 35.4 M and 30 ms, respectively. Compared to the YOLO-V3 (tiny), LedNet (LW-net) achieves a better detection performance. The F_1 score and IoU of LedNet with LW-net are 0.826 and 86.3%, which are 4.4% and 3.9% higher than the YOLO-V3 (tiny), respectively. LedNet with resnet-101 achieves similar computation efficiency compared to the YOLO-V3, from the experimental results. Faster-RCNN is a two-stage detector, which includes an RPN and classification network. Therefore, the computational time of Faster-RCNN is 145 ms, which is much longer than the one-stage detector YOLO and LedNet.

4.5. Detection under occlusion and overlapping

Fruit overlapping and occlusion between fruits, branches, and leaves are challenging issues in the fruit detection in orchards. The detection results are shown in Fig. 11, blue boxes are the apples detected by the LedNet, while the red boxes and red arrows are the false-negative and false-positive of the apple detection, respectively. As shown in the results, the apples which are mostly obscured by branches or leaves are still detected by the LedNet, leading to the false-positives in the detection. The apples which overlap with each other may lead to missing detection by LedNet, leading to the false-negatives in detection. From the experimental results shown in Table 7 and Fig. 11, LedNet achieves a good performance in apple detection under the occlusion and overlapping conditions.

5. Conclusion and future work

In this study, a fast implementation framework of a deep-learning based fruit detection algorithm was developed. This framework includes a label generation module and a fruit detector LedNet. The label generation module utilised the multi-level pyramid and clustering-based classifier to assist fast labelling of training data. LedNet utilised the FPN and ASPP to enhance feature extraction ability and detection performance of the model. Meanwhile, a light-weight backbone was developed to improve computation efficiency of the model. From the experimental results, LedNet with resnet-101 achieved 0.841 and 0.864 on recall and accuracy on the fruit detection of apples in orchards, respectively. The LedNet with light-weight backbone achieved 0.821 and 0.853 on recall and accuracy on the apple detection, and the weights size and average computational time are 7.4 M and 28 ms, respectively. Future work will focus on embedding more functions into the LedNet, including growth monitoring, yield estimation and ripeness detection. Moreover, future work will investigate the automatic labelling generation techniques such as generative adversarial network, to further reduce human intervention during the network training.

Declaration of Competing Interest

The authors declare no conflict of interest.

Acknowledgement

This work is supported by ARC ITRH IH150100006 and THOR TECH PTY Ltd. We acknowledge Zijue Chen and Hongyu Zhou for their assistance in the data collection. And we also acknowledge Zhuo Chen for her assistance in preparation of this work.

References

Badrinarayanan, Vijay, Kendall, Alex, Cipolla, Roberto, 2017. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 39 (12), 2481–2495.

Bargoti, Suchet, Underwood, James, 2017a. Deep fruit detection in orchards. In: 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp. 3626–3633.

Bargoti, Suchet, Underwood, James P, 2017. Image segmentation for fruit detection and yield estimation in apple orchards. J. Field Robot. 34 (6), 1039–1060.

Girshick, Ross, Donahue, Jeff, Darrell, Trevor, Malik, Jitendra, 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587.

Greenspan, Hayit, Goldberger, Jacob, Eshet, Itay, 2001. Mixture model for face-color modeling and segmentation. Pattern Recogn. Lett. 22 (14), 1525–1536.

Hashimoto, Koichi, 2003. A review on vision-based control of robot manipulators. Adv. Robot. 17 (10), 969–991.

He, Kaiming, Gkioxari, Georgia, Dollár, Piotr, Girshick, Ross, 2017. Mask r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969.

He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, Sun, Jian, 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778.

Huang, Gao, Liu, Zhuang, Van Der Maaten, Laurens, Weinberger, Kilian Q, 2017. Densely

connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708.
Kamilaris, Andreas, Prenafeta-Boldu, Francesc X., 2018. Francesc X Prenafeta-Boldu. Deep learning in agriculture: a survey. Comput. Electron. Agric. 147, 70–90.
Kapach, Keren, Barnea, Eh.ud., Mairon, Rotem, Edan, Yael, Ben-Shahar, Oh.ad., 2012. Computer vision for fruit harvesting robots–state of the art and challenges ahead. Int. J. Comput. Vision Robot. 3 (1/2), 4–34.
Koirala, A., Walsh, K.B., Wang, Z., McCarthy, C., 2019. Deep learning for real-time fruit detection and orchard fruit load estimation: benchmarking of 'mangoyolo'. Precis. Agric. 1–29.
Kussul, Nataliia, Lavreniuk, Mykola, Skakun, Sergii, Shelestov, Andrii, 2017. Deep learning classification of land cover and crop types using remote sensing data. IEEE Geosci. Remote Sens. Lett. 14 (5), 778–782.
Laroca, Rayson, Severo, Evair, Zanlorensi, Luiz A., Oliveira, Luiz S., Gonçalves, Gabriel Resende, Schwartz, William Robson, Menotti, David, 2018. A robust real-time automatic license plate recognition based on the yolo detector. In: 2018 International Joint Conference on Neural Networks (IJCNN). IEEE, pp. 1–10.
Lehnert, Christopher, Sa, Inkyu, McCool, Christopher, Upcroft, Ben, Perez, Tristan, 2016. Sweet pepper pose detection and grasping for automated crop harvesting. In: 2016 IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp. 2428–2434.
Li, Yanan, Cao, Zhiguo, Xiao, Yang, Cremers, Armin B, 2017. Deepcotton: in-field cotton segmentation using deep fully convolutional network. J. Electron. Imaging 26 (5), 053028.
Lin, Guichao, Tang, Yunchao, Zou, Xiangjun, Xiong, Juntao, Li, Jinhui, 2019. Guava detection and pose estimation using a low-cost rgb-d sensor in the field. Sensors 19 (2), 428.
Lin, Tsung-Yi, Goyal, Priya, Girshick, Ross, He, Kaiming, Dollár, Piotr, 2017. Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988.
Long, Jonathan, Shelhamer, Evan, Darrell, Trevor, 2015. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440.
Hao, Lu., Cao, Zhiguo, Xiao, Yang, Li, Yanan, Zhu, Yanjun, 2016. Region-based colour modelling for joint crop and maize tassel segmentation. Biosyst. Eng. 147, 139–150.
Majeed, Yaqoob, Zhang, Jing, Zhang, Xin, Longsheng, Fu., Karkee, Manoj, Zhang, Qin, Whiting, Matthew D, 2018. Apple tree trunk and branch segmentation for automatic trellis training using convolutional neural network based semantic segmentation. IFAC-PapersOnLine 51 (17), 75–80.
McCool, Christopher, Sa, Inkyu, Dayoub, Feras, Lehnert, Christopher, Perez, Tristan, Upcroft, Ben, 2016. Visual detection of occluded crop: For automated harvesting. In: 2016 IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp. 2506–2512.
Ng, Pauline C., Henikoff, Steven, 2003. Sift: Predicting amino acid changes that affect protein function. Nucl. Acids Res. 31 (13), 3812–3814.
Papandreou, George, Chen, Liang-Chieh, Murphy, Kevin P, Yuille, Alan L., 2015. Weakly- and semi-supervised learning of a deep convolutional network for semantic image segmentation. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1742–1750.
Pass, Greg, Zabih, Ramin, Miller, Justin, 1996. Comparing images using color coherence vectors. In: ACM Multimedia, vol. 96. Citeseer, pp. 65–73.
Ratner, A., Bach, S., Varma, P., Ré, C, 2017. Weak supervision: The new programming paradigm for machine learning.
Redmon, Joseph, Divvala, Santosh, Girshick, Ross, Farhadi, Ali, 2016. You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788.
Redmon, Joseph, Farhadi, Ali, 2017. Yolo9000: better, faster, stronger. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7263–7271.
Redmon, Joseph, Farhadi, Ali, 2018. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.
Ren, Shaoqing, He, Kaiming, Girshick, Ross, Sun, Jian, 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pages 91–99, 2015.
Sa, Inkyu, Ge, Zongyuan, Dayoub, Feras, Upcroft, Ben, Perez, Tristan, McCool, Chris, 2016. Deepfruits: A fruit detection system using deep neural networks. Sensors 16 (8), 1222.
Shin, Hoo-Chang, Roth, Holger R., Gao, Mingchen, Le, Lu, Ziyue, Xu, Nogues, Isabella, Yao, Jianhua, Mollura, Daniel, Summers, Ronald M, 2016. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. IEEE Trans. Med. Imaging 35 (5), 1285–1298.
Srivastava,Rupesh K., Greff, Klaus, Schmidhuber, Jürgen, 2015. Training very deep networks. In: Advances in Neural Information Processing Systems, pp. 2377–2385.
Szegedy, Christian, Liu, Wei, Jia, Yangqing, Sermanet, Pierre, Reed, Scott, Anguelov, Dragomir, Erhan, Dumitru, Vanhoucke, Vincent, Rabinovich, Andrew, 2015. Going deeper with convolutions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–9.
Szegedy, Christian, Vanhoucke, Vincent, Ioffe, Sergey, Shlens, Jon, Wojna, Zbigniew, 2016. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826.
Tian, Yunong, Yang, Guodong, Wang, Zhe, Wang, Hao, Li, En, Liang, Zize, 2019. Apple detection during different growth stages in orchards using the improved yolo-v3 model. Comput. Electron. Agric. 157, 417–426.
Tijtgat, Nils, Van Ranst, Wiebe, Goedeme, Toon, Volckaert, Bruno, De Turck, Filip, 2017. Embedded real-time object detection for a uav warning system. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2110–2118.
Uijlings, Jasper RR, Van De Sande, Koen EA, Gevers, Theo, Smeulders, Arnold W.M., 2013. Selective search for object recognition. Int. J. Comput. Vision 104 (2), 154–171.
Wang, Panqu, Chen, Pengfei, Yuan, Ye, Liu, Ding, Huang, Zehua, Hou, Xiaodi, Cottrell, Garrison, 2018. Understanding convolution for semantic segmentation. In 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, pp. 1451–1460.
Weiss, Karl, Khoshgoftaar, Taghi M, Wang, DingDing, 2016. A survey of transfer learning. J. Big Data 3 (1), 9.
Yang, Qi, Shi, Liangsheng, Han, Jinye, Zha, Yuanyuan, Zhu, Penghui, 2019. Deep convolutional neural networks for rice grain yield estimation at the ripening stage using uav-based remotely sensed images. Field Crops Res. 235, 142–153.
Yoo, Jae-Chern, Han, Tae Hee, 2009. Fast normalized cross-correlation. Circuits Syst. Signal Process. 28 (6), 819.
Yang, Yu., Zhang, Kailiang, Yang, Li, Zhang, Dongxing, 2019. Fruit detection for strawberry harvesting robot in non-structural environment based on mask-rcnn. Comput. Electron. Agric. 163, 104846.
Zhao, Yuanshen, Gong, Liang, Huang, Yixiang, Liu, Chengliang, 2016. A review of key techniques of vision-based control for harvesting robot. Comput. Electron. Agric. 127, 311–323.
Zhong, Yuanhong, Gao, Junyuan, Lei, Qilun, Zhou, Yao, 2018. A vision-based counting and recognition system for flying insects in intelligent agriculture. Sensors 18 (5), 1489.

4 Fruit Detection and Segmentation for Apple Harvesting Using Visual Sensor in Orchards

Autonomous harvesting shows a promising prospect in the future development of the agriculture industry. Vision system is one of the most challenging components in the autonomous harvesting technologies. CNN has shown superior performance in many computer vision tasks. However, CNN always requires large computational resource during the operation. In this work, a light-weight multi-task CNN DasNet-v1, which combines architecture of semantic segmentation network and a one-stage detection network, is proposed to perform real-time detection and semantic segmentation of fruits and branches in orchard environments. DasNet-v1 uses ASPP and gate-FPN to enhance feature learning in different scale. From the experimental results, the detection and segmentation network with ResNet-101 backbone outperformed on the detection and segmentation tasks, achieving an F_1 score of 0.832 on the detection of apples and 87.6% and 77.2% on the semantic segmentation of apples and branches, respectively. The weights size and average computational time of the light-weight version of the DasNet-v1 are 12.8 M and 32ms by using a Nvidia GTX-1070, respectively. The experimental results show that the Detection and Segmentation Network can effectively perform the real-time detection and segmentation of apples and branches in orchards.

sensors

MDPI

Article

Fruit Detection and Segmentation for Apple Harvesting Using Visual Sensor in Orchards

Hanwen Kang and Chao Chen *

Laboratory of Motion Generation and Analysis, Faculty of Engineering, Monash University, Clayton, VIC 3800, Australia; hanwen.kang@monash.edu

* Correspondence: chao.chen@monash.edu

Received: 3 September 2019; Accepted: 19 October 2019; Published: 22 October 2019

Abstract: Autonomous harvesting shows a promising prospect in the future development of the agriculture industry, while the vision system is one of the most challenging components in the autonomous harvesting technologies. This work proposes a multi-function network to perform the real-time detection and semantic segmentation of apples and branches in orchard environments by using the visual sensor. The developed detection and segmentation network utilises the atrous spatial pyramid pooling and the gate feature pyramid network to enhance feature extraction ability of the network. To improve the real-time computation performance of the network model, a lightweight backbone network based on the residual network architecture is developed. From the experimental results, the detection and segmentation network with ResNet-101 backbone outperformed on the detection and segmentation tasks, achieving an F_1 score of 0.832 on the detection of apples and 87.6% and 77.2% on the semantic segmentation of apples and branches, respectively. The network model with lightweight backbone showed the best computation efficiency in the results. It achieved an F_1 score of 0.827 on the detection of apples and 86.5% and 75.7% on the segmentation of apples and branches, respectively. The weights size and computation time of the network model with lightweight backbone were 12.8 M and 32 ms, respectively. The experimental results show that the detection and segmentation network can effectively perform the real-time detection and segmentation of apples and branches in orchards.

Keywords: deep learning; machine vision; real-time fruit detection; semantic segmentation; visual sensor; automated harvesting robot

1. Introduction

Apple harvesting is a labour-intensive, time-consuming, and costly task. The ageing population and cost of the human resources has led to a decreasing of available labour force for the agriculture harvesting [1]. Therefore, automatic harvesting robots that can automatically work in the field are becoming a promising technology in the future development of the agriculture industry. Different from the traditional automatic harvesting of crops, automatic harvesting of fruits such as apple is in a more complicated case [2]. Robotic harvesting of fruit requires the vision system to detect and localise the fruit. Furthermore, to increase the success rate and reduce the damage rate of automatic fruit harvesting, information of the fruit pose [3] and stem–branch joint location and orientation [4] are also required. This demands the robotic vision system to accurately and robustly extract the geometry and semantic information from the working scene in the orchard environment [5]. Recently, with the advancements of the depth camera technologies, the harvesting robotic vision system is able to model and present the working scene in the three-dimensional form [6]. However, it is still challenging to robustly and accurately perform semantic processing of the visual data in the orchard environment, such as detection and segmentation of the fruit and branch, due to various factors such as illumination

variance, occlusion, and variations of object appearance. To overcome these, it is crucial to develop a highly effective and robust vision algorithm for the fully automatic harvesting [3]. In this work, a multi-function Deep Convolution Neural Network (DCNN) is developed to perform the real-time detection and semantic segmentation of apples and branches in orchards. Firstly, to enhance the feature extraction ability of the network, the Gated Feature Pyramid Network (GFPN) and the Atrous Spatial Pyramid Pooling (ASPP) are utilised in the developed Detection and Segmentation Network (DaSNet). Secondly, to facilitate the fast computation of the network on the embedded computing device, a lightweight network (lw-net) is developed based on the residual network architecture. In the experiment, we evaluated the performance and efficiency of the DaSNet with different backbones and different detector architectures. The comparison between the DaSNet and other deep-learning based detection and segmentation works was also included.

The rest of the paper is organised as follows. Section 2 reviews the related work of fruit detection techniques. Section 3 introduces the DaSNet model in detail. The experiment and discussion are demonstrated and analysed in Sections 4 and 5 concludes the work.

2. Related Work

Fruit detection has been studied extensively in the past few decades. Several kinds of sensors have been applied in the automation of fruit harvesting [7], including RGB/RGB-D camera, laser sensor, thermal imaging sensor, and spectral imaging sensor. This work focuses on reviewing the techniques which are developed for RGB image processing. Imaging detection can be classified into two groups: conventional machine-learning based algorithms and deep-learning based algorithms. The former methods use the image feature descriptors to encode the feature information, and then apply the machine-learning based classifier to perform the segmentation or detection the fruit within the image [8]. There are many expert-coded feature descriptors that have been developed, such as the histogram of gradient [9], the colour coherence vector [10], and the local binary patterns [11]. Similarly, many machine-learning based classifiers have been developed, such as the clustering, the Support Vector Machine (SVM) and the neural network. Traditional machine-learning based algorithms have been widely applied in automatic agriculture applications. Zhou et al. [12] proposed a colour feature-based logistic regression classifier to detect apples in the orchard environment. Song et al. [13] applied a Bayes classifier and SVM to learn the colour and texture features, in order to detect peppers with an RGB camera. Luo et al. [14] and Wang et al. [15] utilised the colour feature-based and texture-feature based AdaBoost classifier to perform the fruit detection.

Recently, DCNN shows promising performance in many computer vision tasks, including object classification [16], object detection [17], and image segmentation [18]. Compared to traditional machine-learning based algorithms, DCNN achieves a more robust and accurate performance due to its strong feature extraction ability and autonomous learning mechanism [19]. There are two kinds of DCNN model that have been developed to perform object detection: two-stage detectors and one-stage detectors. Region Convolution Neural Network (RCNN) is one of the most successful works of the two-stage detector, including RCNN [20], Fast/Faster RCNN [21,22], and Mask RCNN [23]. RCNN contains two sub-tasks networks: the Region Proposal Network (RPN) and the classification network. The RPN searches the location of Region of Interest (ROI), while the classification network predicts the class of ROI and regresses the boundary box of the ROI candidates. RCNN has been widely applied in many vision-guided automatic agriculture applications. Sa et al. [24] applied Faster RCNN on multi-vision sensor to detect peppers, rock-melons and apples. Bargoti and Underwood [25] adopted the faster-RCNN model on the detection of apples and mangos in orchards. Yu et al. [26] applied the mask RCNN to perform the detection and segmentation of the strawberry in the greenhouse, in order to guide the automatic harvesting of strawberries. Another DCNN model, called "one-stage detector" was developed more recently. Representative methods of one-stage detector are Single Shot Detection (SSD) [27,28] and You Only Look Once (YOLO) [29–31]. The one-stage detector combines RPN and classification network into a single architecture, which largely reduces

the computation cost of the forward inference. The one-stage detector has been gradually studied and applied in the vision-guided automatic agriculture applications such as the yield estimation and automatic harvesting. Tian et al. [32] modified an improved YOLO-V3 network to perform real-time detection of apples, which is developed to monitor and evaluate the growing of apples in orchards. Koirala et al. [33] applied a lightweight YOLO network to perform the yield estimation of mangos in orchards, and reported an F_1 score of 0.89 in their work.

Semantic segmentation is another essential computer vision task, which predicts a class label in each of pixel of the image [18]. Compared to object classification task, which predicts the image class while losing the spatial information of the objects, semantic segmentation can preserve the spatial information of the objects and predict its shape within the image [34]. Typical deep-learning based semantic segmentation network applies the auto-encoder architecture to encode the image data and generate the semantic segmentation of the image. Rather than using the sliding windows strategies to classify all pixels within the image [35], deep-learning based semantic segmentation network can predict the labels for all pixels within the image in a single forward inference. Many semantic segmentation network architectures have been developed for different applications, such as the Full Convolution Network (FCN) [36], SegNet [37], and the DeepLab [38–40], which are designed for general applications, and Unet and Vnet [34,41], which are designed for medical image analysis. Semantic segmentation network has also been applied in many agriculture applications. McCool et al. [42] developed a multi-feature classifier to perform the semantic segmentation on peppers, which is used to guide the robotic harvesting of peppers. Bargoti and Underwood [43] developed a multi-Layered Perception to segment apples for yield estimation in orchards. Li et al. [44] applied an FCN model to perform the automatic ground-based in-field cotton segmentation. Lin et al. [5] applied an FCN model to perform the segmentation of guava fruits and branches and estimate the pose of guava fruits based on the segmented information to guide the robotic harvesting. It was reported that FCN model achieved higher segmentation accuracy than traditional algorithms.

3. Material and Methods

3.1. Vision Sensing System

The developed apple harvesting robot includes a UR-5 robotic arm and an RGB-D visual sensor, as shown in Figure 1. The RGB-D camera applied in this work is the Kinect-v2, which is developed by Microsoft Inc. The Kinect-v2 is comprised of an RGB camera and an infrared(IR) depth camera, and can capture colour images with the resolution in the range between 640 × 480 and 1920 × 1080. The IR depth camera of the Kinect-v2 can capture depth images with the resolution of 424 × 512. During working, the depth image is resized to be consistent with the colour image size and fused. Based on the previous in-field experiments and the computation ability of the applied embedded computing device, the resolution of 640 × 480 is used in this work. In the experiment, the Kinect-v2 was controlled using the ROS-kinetic in Ubuntu 16.04 with the libfreenet2 SDK tool.

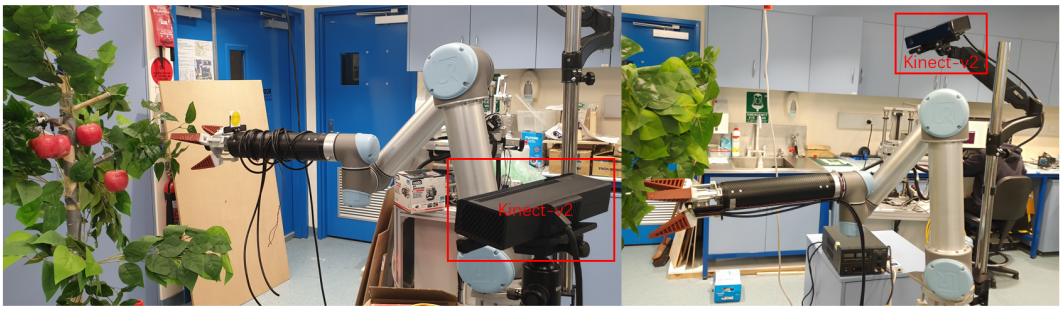


Figure 1. The developing apple harvesting robot. The robot comprises a RGB-D camera for vision sensing. A universal robot arm (UR5) is applied as the manipulator.

3.2. Network Architecture

3.2.1. Gated-FPN for Multi-level Fusion

Figures 2–4 show the three developed architectures of the DaSNet, which are named DaSNet-A, DaSNet-B and DaSNet-C, respectively. DaSNet adopts multi-level feature pyramid network to receive the feature tensors from the C3 layer (1/8), C4 layer (1/16) and C5 layer (1/32) of the backbone network. From the previous study of the representation of the feature in the DCNN model, features in different levels of the network contain different information of the objects [45]. The lower level network (such as C3) mainly includes the spatial information of the objects, while the higher level network (such as C5) mainly includes the semantic information of the objects. A recent study by Yao et al. [46] has pointed out that direct fusion of the different levels of the network can lead to the spatial shift of the feature and unbalance gradient propagation in the network training. Therefore, they developed Gated-FPN to minimise the effect of the above issues. Similarly, GFPN design is adopted in the DaSNet architecture to enhance the feature expression of the model. The GFPN design adopted in the DaSNet is inspired by the work of the Long Short-term Memory (LSTM) [47] and the Gated Recurrent Unit (GRU) [48], which adopt gate network to enable the network to selectively memorise or forget the information within the sequences data. The GFPN in DaSNet adopts a channel-wise multiplication on each channel of the input feature tensors, to allow the network to adjust the weights of the feature in the feature maps. The weights used for the channel-wise multiplication is pre-activated by the sigmoid function, to allocate the value range of the weights from zero to one. A batch-normalisation layer is added after the gate as our experiment shows that the batch-normalisation layer can improve the performance of the network model. The GFPN in the DaSNet allows the selective representation of the feature between different levels, which can minimise the spatial shift of the feature maps from different levels and balance the backward propagated gradients. The architecture of the GFPN used in DaSNet is shown in Figure 5.

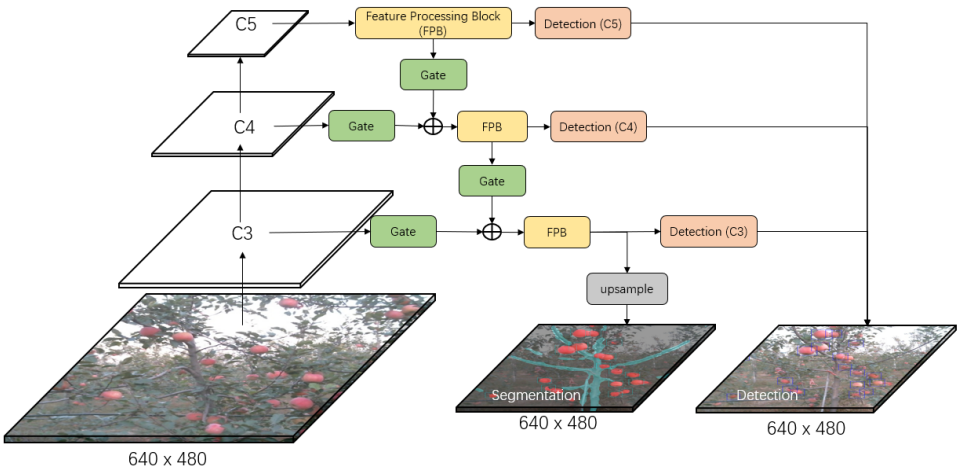


Figure 2. The architecture of DaSNet-A model. DaSNet-A has the common GFPN and the feature processing block for processing of detection and segmentation tasks. The output of the feature processing block in C3 level is upsampled to the original image size to generate the semantic segmentation prediction.

Three different DaSNet architectures are developed in this work, as shown in Figures 2–4, respectively. The details about these three network architectures and its correspond training methods are included in Section 4.1.

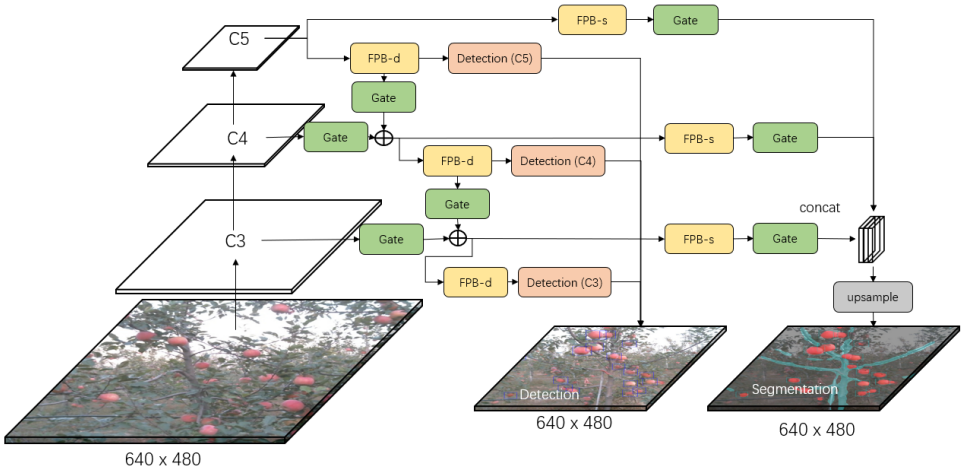


Figure 3. The architecture of DaSNet-B model. DaSNet-B has the common GFPN but the independent feature processing block for processing of detection and segmentation tasks. FPB-s stands for the feature processing block of the segmentation branch, while FPB-d stands for the feature processing block of the detection branch.

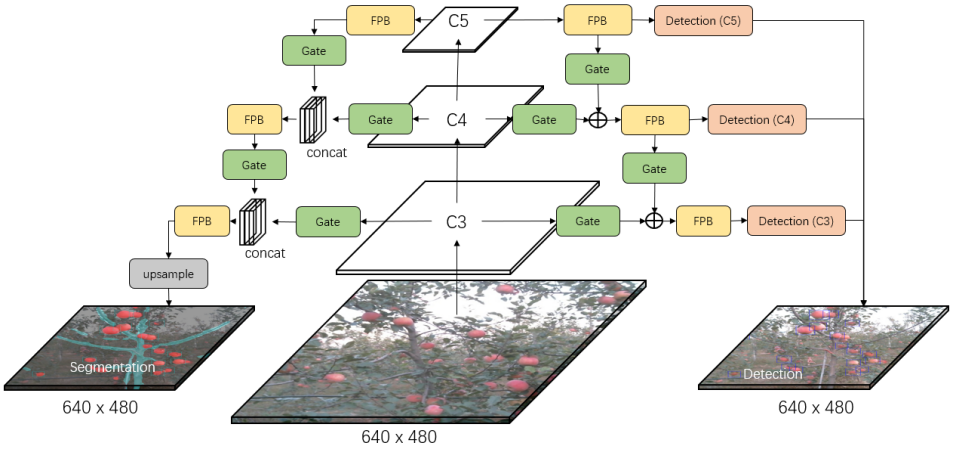


Figure 4. The architecture of DaSNet-C model. DaSNet-C has the independent GFPN and the feature processing block for processing of detection and segmentation tasks separately.

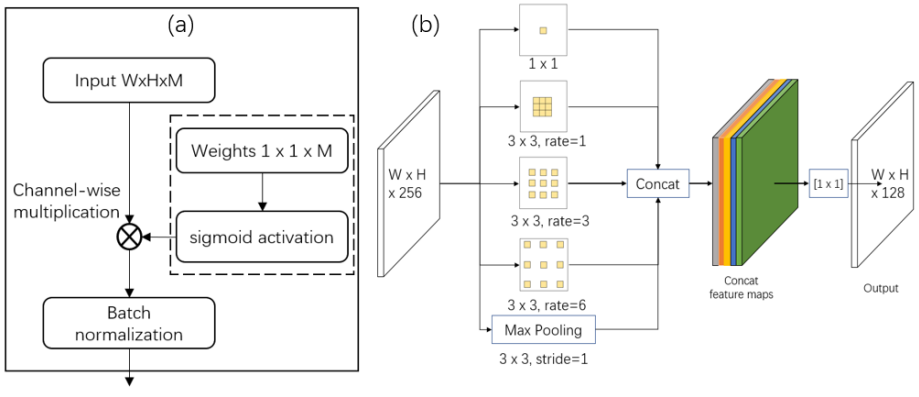


Figure 5. (**a**) The architecture of gate in the GFPN; and (**b**) the architecture of ASPP in the feature processing block.

3.2.2. ASPP for Multi-Scale Fusion

Each level of the GFPN in the DaSNet adopts a feature processing block to process the feature maps before it is fed into the detection and segmentation branch. DaSNet utilises the ASPP to enhance

the feature extraction of the multi-scale information of the objects; the architecture of the feature processing block is shown in Figure 5. The ASPP has been applied in many previous works on object detection [46] and segmentation [49]. It relies on the dilation convolution with different dilation rate to encode the multi-scale information of the objects into a single pixel within the feature maps, as shown in Figure 6. The ASPP adopted in the DaSNet applies three branches, which include three 3×3 dilation convolutions with dilation rates as 1, 3 and 6, and the feature maps on each branch of the ASPP have 64 channels. In addition, another branch, which adopts a 3×3 max-pooling layer, is also applied in the ASPP, as our experiment suggested that max-pooling can improve the detection and segmentation performance of the model. To keep the consistency of the channel number of the feature maps on each branch of the ASPP, a convolution layer with a 1×1 kernel is adopted after the max-pooling. All the branches of the ASPP are concatenated to generate the combined feature map. Then, a 1×1 convolution layer is applied to fuse the information within the combined feature map and reduce the channel number of the feature maps to 128.

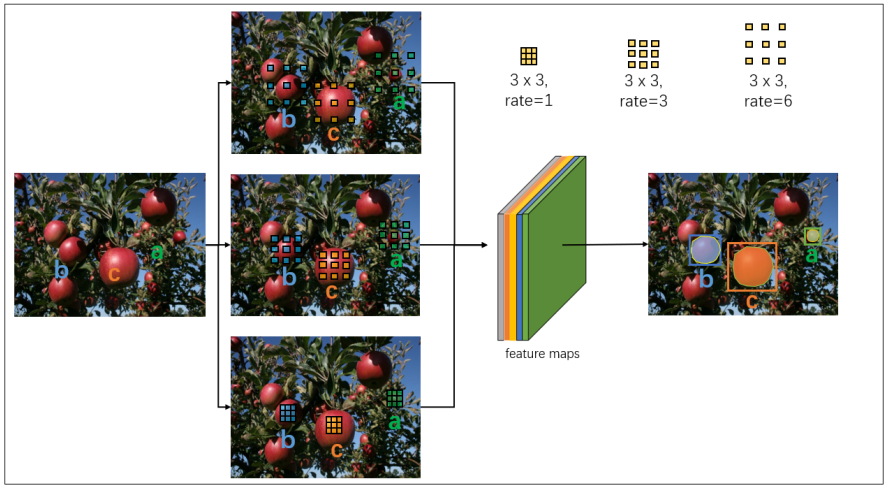


Figure 6. The ASPP adopts the dilation convolution with different dilation rates to encode the multi-scale information of objects.

3.2.3. Lightweight Designed Backbone

To reduce the computing cost and facilitate the real-time applications of the DaSNet model in the embedded computing device, a lightweight backbone network (lw-net0 is developed based on the residual network architecture [50]. The lw-net adopts the bottleneck residual network block design to reduce the weight size and computation complexity of the network inference. Meanwhile, to reduce the feature information loss during the under-sampled pooling operation, the max-pooling layer of the original residual network architecture is replaced with a modified down-sampling block design. Both the bottleneck residual network block and the down-sampling block comprise two branches: the body branch and the shortcut branch, which are shown in Figure 7. The bottleneck residual network block design shortcuts to add the input feature map from the input of the block to the output of the block. The down-sampling block applies a max-pooling layer in the shortcut branch to perform the pooling of the input feature maps. Meanwhile, the second convolution layer in the body branch of the down-sampling block applies a convolution layer with stride 2 to perform the pooling of the feature maps as well. The lw-net architecture includes nine bottleneck residual network blocks and five down-sampling blocks. In the experiment, the lw-net was pre-trained on the Cifar-10 and the Cifar-100 datasets. The validation accuracies of the lw-net on the Cifar-10 and the Cifar-100 datasets were 92.7% and 71.6%, respectively. The size of the total weight of the lw-net is only 3.46 MB. In the experiment, the lw-net was further trained with the collected orchard image data, with the resolution of the training image set to 128×128 and the object classes were apple, branch and background. In addition to the developed lw-net, for acceleration of network computation purpose, some other

state-of-the-art classification networks were also applied as the backbone for the DasNet model. In the experiment, ResNet-50/ResNet-101 [50] and Darknet-53 [31], which were pre-trained with ImageNet dataset, were applied as the backbone network of the DasNet model.

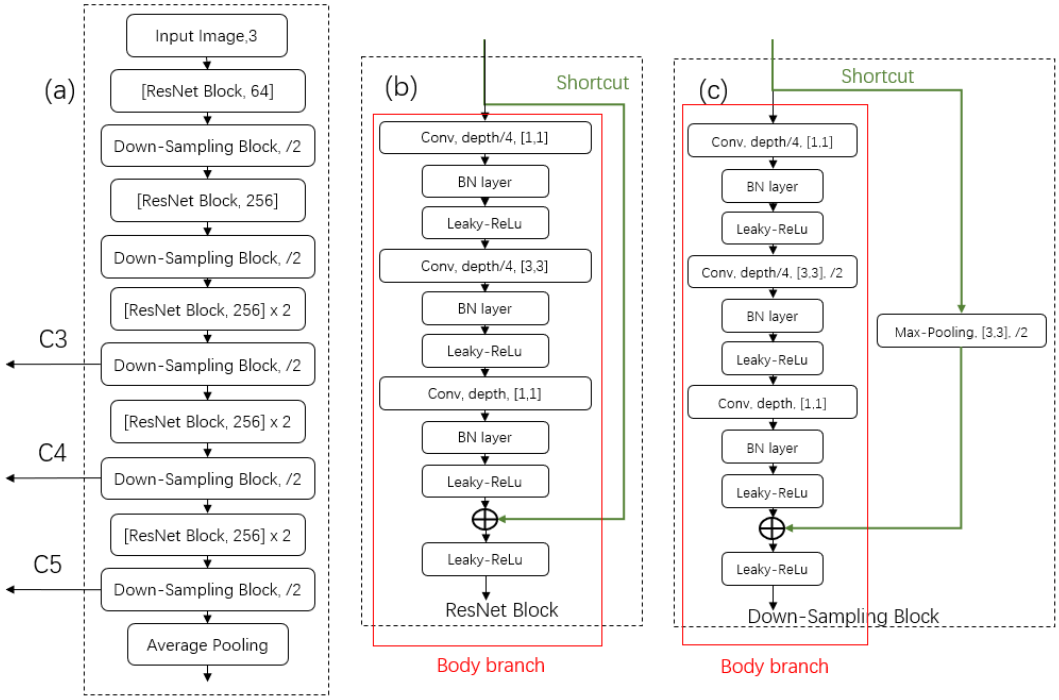


Figure 7. The architecture of the lw-net: (**a**) the lw-net architecture; and (**b**,**c**) the bottleneck residual network block and down-sampling block architectures, respectively.

3.3. Training Data and Method

3.3.1. Data Collection

The data were collected by using the Kinect-v2 in the orchard located in Qingdao, China. The collection time of the image data was from 08:00 to 18:00. We collected 800 images from the orchards in total. The ground truth of the object detection was labelled by using the "LabelImg", which is publicly available on Github. The ground truth of the semantic segmentation was labelled by using the windows drawing tool and surface pen. In the following experiment, 600 out of 800 images were used to train the network, while the other images served as the validation data.

3.3.2. Training Method

The data were collected from a mobile vehicle that works in the orchard. The distance between apple trees to the vehicle was between 0.8 and 1.5 m, which is also the working distance of the developed apple harvesting robot. The diameter of apples in the orchard were 80–100 mm. At the working distance of 0.8–1.5 m, most apples in the training data were presented in the form of small scale objects. This unbalanced distribution of the scale of the objects may lead to under-fitting issue during the training of the anchor-box based detector. Therefore, an augmentation method that can minimise the unbalanced distribution of the scale of the objects within the training data was utilised in this work. The resolution of the original image in the training dataset is 640 $\times$ 480, while the resolution of the image used for the network training is 320 $\times$ 320. The reason for applying 320 $\times$ 320 as the training resolution is to increase the number of the image in each training batch, which can stabilise the training process of the batch normalisation layer [51]. During the training, the augmentation algorithm had the probability of 0.5 to crop a patch whose size is between 160 $\times$ 160 and 320 $\times$ 320 from the original image and resize the cropped patch to the training resolution. Then, this step had

the probability of 0.5 to be repeated another time to further amplify the small objects within the training data. Other image augmentations, including image flip, colour saturation, contrast and brightness adjustment, and translation, were also applied during the training. Several examples of the training data that were processed by the applied augmentation are shown in Figure 8. To analyse the distribution of the scale of the objects within the training data before and after the augmentation process, a statistical analysis was performed, which is shown in Table 1.

Table 1. Statistic of different scale of object in the training data during the network training.

Index	Iteration	If Augmentation	Small	Median	Large
1	100	Yes	67%	32%	1%
2	100	No	87%	13%	0%
3	1K	Yes	58%	37%	5%
4	1K	No	85%	12%	1%
5	10K	Yes	51%	40%	9%
6	10K	No	89%	11%	0%

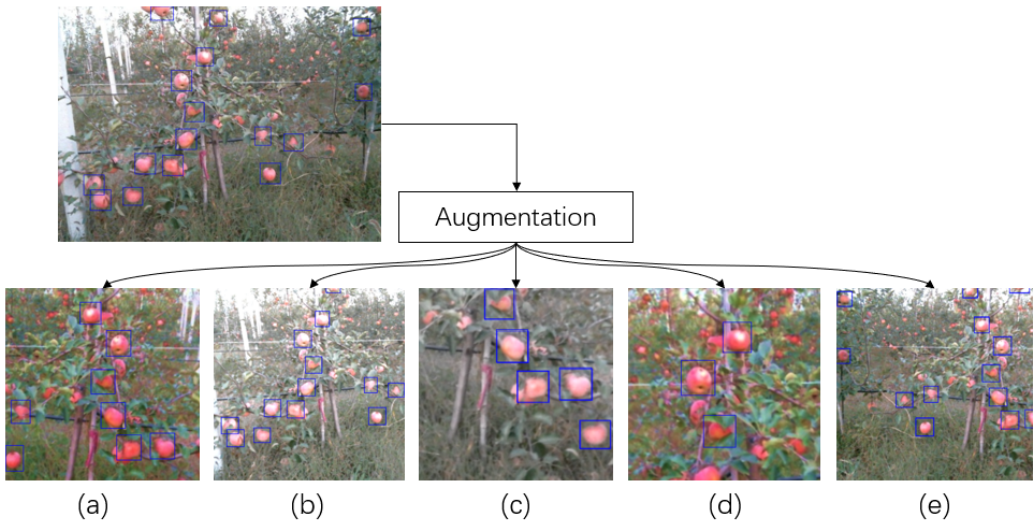


Figure 8. Examples of augmentation method applied in the network training using: (**a**) crop, flip and saturation adjustment; (**b**) brightness and contrast adjustment; (**c**) crop and flip; (**d**) crop and saturation; and (**e**) flip.

From the statistical result of the training data shown above, the applied augmentation algorithm can minimise the unbalanced distribution of the scale of the objects in the training data. However, considering the number of objects in the large scale in the training data is still limited compared to the objects in the small and medium scale in the training data, some open-source image data were collected into the training dataset to further balance the distribution of the scale of the objects. During the training, the Adam-optimiser was used to train the DasNet; the learning rate and decay rate used in training were 0.01 and 0.9/epoch based on our previous experiment results.

4. Experiment and Discussion

The DaSNet code was implemented in Tensorflow 1.11 and trained on the Nvidia GTX-1080Ti. The Kinect-v2 was controlled using the ROS-kinetic on the Ubuntu 16.04. The pre-trained weight and implement code of ResNet-50/101 in Tensorflow [52], YOLO-V3/YOLO-V3(tiny) in tensorflow [53], Faster-RCNN in caffe [54], and FCN-8s(ResNet-50/ResNet-101) in Tensorflow [55] were from the Github publicly code library.

Mean Intersection of Union (MIoU) [56] was used to evaluate the performance of the network on semantic segmentation. The Average Precision (AP_{IoU}) [57], and F_1 score [58] were used to evaluate the performance of the network on object detection.

4.1. Experiment on Network Architecture and Training

4.1.1. Experiment on Network Architectures

DaSNet is developed to perform the detection and semantic segmentation of the multi-class objects within a single network architecture. The object detection task includes the prediction of the confidence score, the boundary box, and the class of the objects, while the semantic segmentation only includes the classification on each pixel within the feature maps. Meanwhile, object detection predicts the objects from the separate level of GFPN (C3, C4, and C5), while the prediction of the semantic segmentation is generated from the upsampled feature maps of the C3 level in the GFPN. Therefore, there may be a significant difference in the distribution of the feature maps between the object detection task and semantic segmentation task.

Three architectures were developed to explore the optimal design of the network, which allows the model to fit the feature distribution of different tasks within a single network. These three models are named as DaSNet-A, DaSNet-B and DaSNet-C and shown in Figures 2–4, respectively. DaSNet-A has the common GFPN and feature processing blocks for both object detection and semantic segmentation. The prediction of the object detection is generated from the C3, C4 and C5 levels of the GFPN, while the semantic prediction is generated from the upsampled feature maps of the C3 level of the GFPN. DaSNet-B has the common GFPN but the independent feature processing blocks for the object detection and semantic segmentation. DaSNet-C has the independent GFPN and feature processing blocks for the object detection and semantic segmentation.

Two different training strategies were utilised based on the characters of the different network architectures. The first method "M1" is to train the network on the detection and semantic segmentation tasks simultaneously, while the second method "M2" is to train the network on the detection and semantic segmentation tasks separately. DaSNet-A was trained with the M1 method since the object detection and semantic segmentation share the major body of the network model. DaSNet-B and DaSNet-C were trained with M1 and M2 methods to explore which training strategies are optimal for such network architectures. During the training of DaSNet-B and DaSNet-C with the M2 method, the weights of the detection branch and backbone were frozen, and only the weights of the segmentation branch were involved.

4.1.2. Experiment Results and Discussion

The experimental results of the comparison between the different network architectures and training methods are shown in Table 2.

Table 2. Experiment on DaSNet architectures and training methods (G stand for the GFPN) (* F stands for fruit, B stands for branch).

Index	Model	Method	AP_{50}	F_1	$MIoU_F^*$	$MIoU_B^*$
1	DaSNet-A(G)	M1	0.792	0.796	0.849	0.683
2	DaSNet-B(G)	M1	0.803	0.8	0.857	0.703
3	DaSNet-C(G)	M1	0.819	0.819	0.86	0.76
4	DaSNet-B(G)	M2	**0.827**	0.821	**0.865**	0.757
5	DaSNet-C(G)	M2	0.823	**0.824**	0.864	**0.762**
6	DaSNet-B(FPN)	M2	0.799	0.792	0.832	0.722

GFPN vs. **FPN**: Experiments 4 and 6 showed the performance evaluation of DaSNet-B with GFPN and FPN, respectively. The results show that GFPN improved the AP_{50} from 79.9% to 82.7%, and increased the F_1 score from 79.2% to 82.1%. Similar results are also shown in the semantic segmentation: DaSNet-B with GFPN had a higher MIoU score than DaSNet-B with the FPN. DaSNet-B with GFPN achieved 0.865 and 0.757 on the segmentation of apples and branches, respectively. DaSNet-B with FPN achieved 0.832 and 0.722 on the segmentation of apples and branches, respectively. The experiment showed that GFPN can increase the performance of the network in both tasks of object detection and semantic segmentation.

Network Architectures: Experiments 1–3 showed the performance of DaSNet-A, -B and -C on the detection and segmentation when M1 training strategy was applied. From the experiment results, DaSNet-C outperformed in the comparison of the performance of the three network architectures. DaSNet-C had the independent GFPN and feature processing block for semantic segmentation and object detection, which allowed the network to fit the feature distribution of the different tasks properly. DaSNet-A had the common GFPN and feature processing blocks for both tasks, which limited the ability of the network to fit the feature distribution of the different tasks. From the experiment results, DaSNet-A showed the least efficient performance of the detection and segmentation.

Training Methods: Experiments 2–5 compared the performance of the networks when different training methods were applied. Experiments 3 and 5 compared the performance of DaSNet-C trained with M1 and M2 methods. Since DaSNet-C model had the independent branch for object detection and semantic segmentation, similar results are shown in the comparison. Experiments 2 and 4 compared the performance of DaSNet-B with M1 and M2 methods. From the experiment results, the DaSNet-B showed better performance when the M2 training method was applied. The reasons that contribute to the results are summarised as follows. When M1 training method was applied, backward propagation of different feature distribution on the different tasks may lead to the under-fitting of the weights on both tasks. When M2 training method was applied, the training of the segmentation task only focuses on the update of the weight in the segmentation branch. From the experiment results, this measurement improved the training quality and performance of the network.

Implementation efficiency: Table 3 shows the weights size and inference time of the DaSNet-A, -B and -C. Considering the aspect of implementation efficiency and performance, DaSNet-B achieved the equal detection (0.827 and 0.821 on AP_{50} and F_1 score, respectively) and segmentation performance (86.5% and 75.7% for apple and branch segmentation, respectively) compared to DaSNet-C. It kept a similar implementation efficiency compared to DaSNet-A model (weight size is 12.8 M and inference time is 32 ms). Therefore, the DaSNet-B architecture was considered as the best performing candidate in the experiment, and it was applied as the DaSNet in the following experiments.

Table 3. Time efficiency and weights size of developed model (tested on GTX1080Ti).

Index	Model	Inference Time	Weights Size
1	DaSNet-A(G)	**30** ms	**9.6** M
2	DaSNet-B(G)	32 ms	12.8 M
3	DaSNet-C(G)	40 ms	15.8 M

4.2. Experiment on Detection Performance

This experiment compared the performance and implementation efficiency of the DaSNet-B when different backbone networks were applied, including the ResNet-50, ResNet-101, Darknet, and the developed lw-net. Meanwhile, a comparison between the DaSNet and other state-of-the-art works, includeing YOLO-V3, YOLO-V3 (tiny) and Faster RCNN, was also performed. The experimental results are listed in Table 4.

Table 4. Experiment of prediction performance on validation set (640 × 480) with GTX1080Ti.

Index	Model	AP_{50}	F_1	Weights Size	Time
1	DaSNet-B(lw-net)	0.827	0.821	**12.8** M	**32** ms
2	DaSNet-B(ResNet-50)	0.831	0.825	112 M	47 ms
3	DaSNet-B(ResNet-101)	**0.836**	**0.832**	188 M	72 ms
4	DaSNet-B(Darknet-53)	0.832	0.827	176 M	50 ms
5	YOLO-V3(Darknet-53) [31]	0.80	0.797	248 M	48 ms
6	YOLO-V3(Tiny) [31]	0.782	0.776	35.4 M	38 ms
7	Faster-RCNN (VGG-16) [22]	0.817	0.813	533 M	136 ms

Experiments 1, 5, 6 and 7 compared the performance of the object detection of DaSNet-B (lw-net), YOLO-V3, YOLO-V3 (Tiny) and Faster RCNN. The experimental results show that DaSNet-B (lw-net) outperformed the three other network models. DaSNet-B (lw-net) achieved 82.7% on AP evaluation and 0.821 on F_1 score. Meanwhile, DaSNet-B model also outperformed in the implementation efficiency. The weights size and inference time of DaSNet-b Model are 12.8 M and 32 ms, respectively. Experiment 6 was the performance evaluation of the tiny version of the YOLO-V3, which optimises YOLO-V3 in terms of the calculation complexity and time efficiency. From the experiment results, DasNet-B with lw-net achieved better performance on detection task, while kept the equal implementation efficiency with the tiny version of YOLO-V3 network. Experiments 1–4 showed the performance evaluation of the DaSNet-B with different backbone, including lw-net, ResNet, and darknet. As shown in the results, DaSNet-B with ResNet-101 backbone performed the best within the test, achieving 83.6% on AP and 0.832 on F_1 score. The experimental results indicate that the powerful backbone network can increase the performance of the detection network. However, the implementation efficiency of DaSNet-B (ResNet-101) showed a decrease as the backbone increase the computation complexity of the inference. The weights size and inference time of the DaSNet-B (ResNet-101) are 188 M and 72 ms, respectively. During implementation, DaSNet can use different backbone network based on the computation hardware and the design requirement.

4.3. Experiment on Segmentation Performance

Semantic segmentation returns a multi-class mask to predict the label of each pixel of the input RGB image, which is a critical task for sensing and understanding the working environment. In this experiment, two different architectures modified from DaSNet-B model were tested. The first model uses the concatenate operation at the C3 layer to fuse the feature maps for semantic label prediction, which is named as the DaSNet-B-Concat model. The second model uses an adding operation to fuse the feature maps for semantic label prediction, which is named as DaSNet-B-Add model. The architecture of these two models are shown in Figure 9. The reason behind this experiment was to explore which operation could generate better performance on the semantic segmentation. Similar to the evaluation of the detection results, several pre-trained backbone networks were applied in this experiment. The FCN-8s network was applied as the baseline algorithm to form the comparison. FCN-8s network uses ResNet-50 and ResNet-101 as the backbone. The FCN-8s network was first trained with PASCAL-VOC2012 and then trained using the orchard data. The results of the experiment are shown in Table 5. The demonstration results of segmentation and detection of apple and branch are shown in Figure 10.

Experiments 1 and 2 compared the performance of the DaSNet-B-add and the DaSNet-B-concat model. From the experiment results, the DaSNet-B-concat achieved a better MIoU score compared to the DaSNet-B-add model. DaSNet-B-concat achieved 86.5% and 75.7% on the segmentation of apples and branches, i.e., 0.7% and 2.6% higher than the DaSNet-B-add model, respectively. Experiments 2–5 compared the performance of the DaSNet-B-concat model with the different backbone networks. Similar to the experimental results for the detection evaluation, the DaSNet-B-concat model with the ResNet-101 backbone outperformed in the segmentation evaluation. Meanwhile, the DaSNet-B-concat model with the lw-net backbone also showed a good performance on the segmentation of apples and branches, at 86.5% and 75.7%, respectively. Experiments 3, 4, 6, and 7 compared the performance of

the semantic segmentation between DaSNet and FCN-8s. From the experiment results, the DaSNet achieved a higher accuracy on the segmentation task compared to the FCN-8s. Especially, the MIoU score of the branch segmentation achieved by the DaSNet was 4.5% and 3.9% higher than the MIoU score achieved by FCN-8s. Since DaSNet adopts ASPP and GFPN to enhance the extraction ability in terms of the multi-scale information and multi-level information, DaSNet achieved a better segmentation performance compared to the FCN-8s model. From the experiment results, the segmentation of the branch is more challenging than the segmentation of apples, as there are many branches which are blocked by the leaves. In this condition, DaSNet could still segment the majority structure of the branch from the background accurately, as shown in Figures 10 and 11.

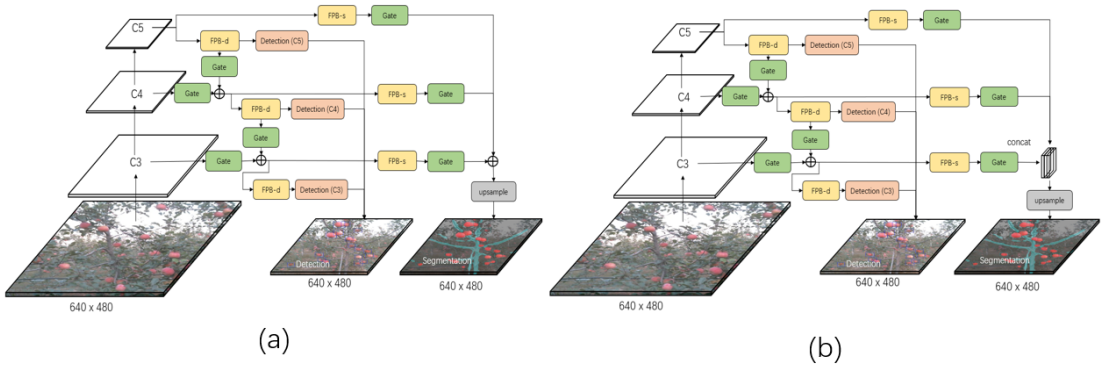


Figure 9. The architecture of DaSNet-B-Add (**a**) and DaSNet-B-concat (**b**). DaSNet-B-Add model uses adding operation to fuse the feature maps for segmentation, while DaSNet-B-Concat model uses concatenate operation.

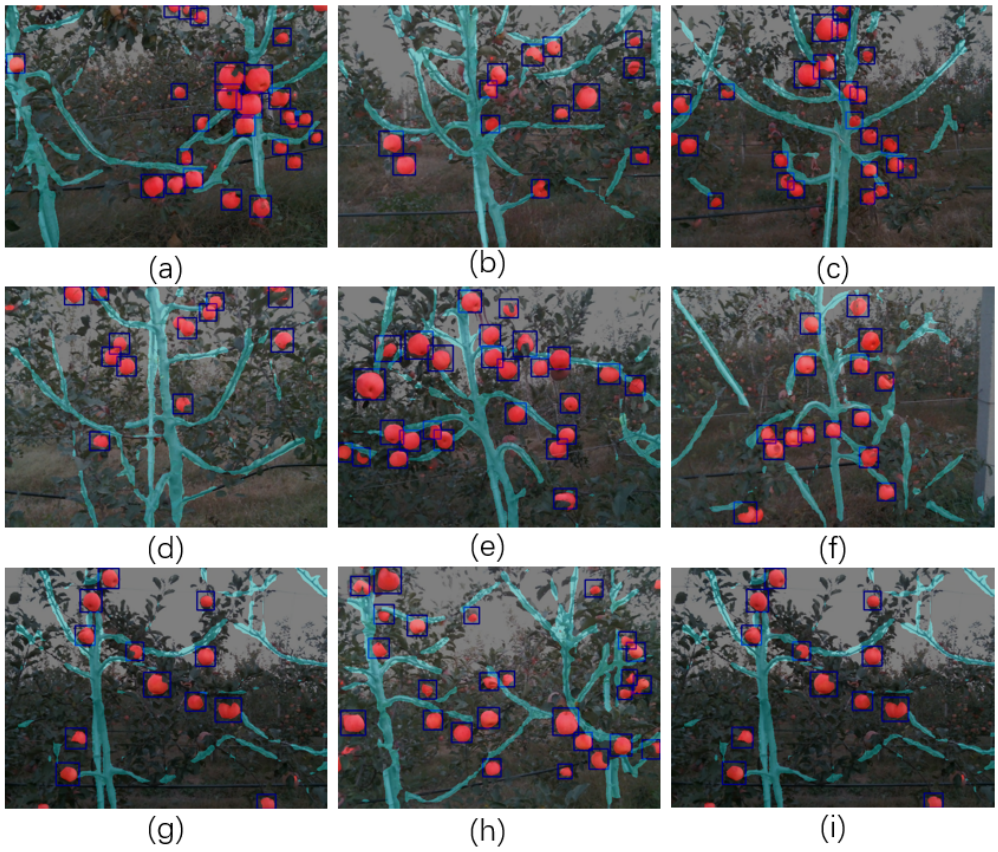


Figure 10. (**a**–**i**) show the segmentation and detection of apples and branches in orchard environment by using the DaSNet.

Table 5. Experiment of semantic segmentation performance on validation set (640 × 480) with GTX1080Ti (* F stands for the Fruit, while B stands for the branch).

Index	Model	$MIoU_F^*$	$MIoU_B^*$
1	DaSNet-B-Add(lw-net)	0.858	0.731
2	DaSNet-B-Concat(lw-net)	0.865	0.757
3	DaSNet-B-Concat(ResNet-50)	0.87	0.763
4	DaSNet-B-Concat(ResNet-101)	**0.876**	**0.772**
5	DaSNet-B-Concat(Darknet-53)	0.868	0.762
6	FCN-8s(ResNet-50) [36]	0.853	0.718
7	FCN-8s(ResNet-101) [36]	0.861	0.733

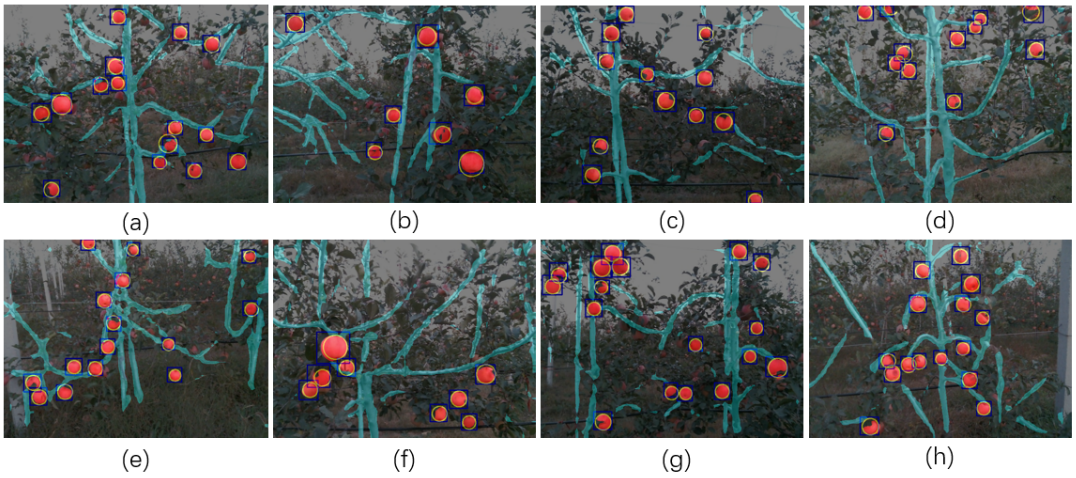


Figure 11. (**a**–**h**) show the instance segmentation (* yellow circle) of apples by using the circle hough transform based on the detection and segmentation results of the DaSNet

After the segmentation by using DaSNet, Circle Hough Transform (CHT) was further used to segment apples and its instance mask in each detected boundary box. Several processed image by using the DaSNet and the CHT are shown in Figure 11. The blue box indicates the detected boundary boxes and the yellow circle indicates the detected apple in each of boundary box. With the semantic label and spatial information on each pixel within the image, further processing such as estimation of the fruit pose [5], branch reconstruction [59] and the estimation of the stem–branch joint [3] can be applied accordingly. Such post-processing techniques will be included in future work of our study.

5. Conclusions

This work developed a multi-function network DaSNet to perform the real-time detection and semantic segmentation of the apple and branch in the orchards. The DasNet utilises the GFPN to fuse the information from different levels of the model and adopts the ASPP to enhance the feature extraction of multi-scale information of the objects. To improve the real-time computing performance of the model in the embedded computing device, a lightweight backbone network based on the residual network architecture was developed. Based on the different characters of the semantic segmentation and object detection, three different DasNet architectures and corresponding training strategies were developed and evaluated in the experiment. The comparison of the performance between the DasNet and the other state-of-the-art works in object detection and semantic segmentation was included in the experiment. From the results, the DaSNet with ResNet-101 backbone performed the best in both semantic segmentation and object detection tasks. It achieved an F_1 score of 0.832 on the detection of apples and 87.6% and 77.2% on the segmentation of apples and branches, respectively. The DaSNet with lightweight backbone lw-net achieved a good detection and segmentation performance while it outperformed in the computation efficiency. It achieved an F_1 score of 0.821 on the detection of apples

and 86.8% and 75.7% on the segmentation of apples and branches, respectively. The weights size and inference time of the network model was 12.8 MB and 32 ms, respectively. Overall, the developed DasNet can perform real-time detection and segmentation in the orchards.

Author Contributions: H.K. contributed to developing the algorithm pipeline, data labelling and revision, programming and writing, C.C. provided significant suggestions on the development and contributed to the writing and editing.

Funding: This work was supported by ARC ITRH IH150100006 and Thor Technology.

Acknowledgments: We acknowledge Zijue Chen and Hongyu Zhou for their assistance in the data collection. We also acknowledge Zhuo Chen for her assistance in the writing and polishing.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. ABARES. *Australian Vegetable Growing Farms: An Economic Survey, 2016–17 and 2017–18*; Australian Bureau of Agricultural and Resource Economics (ABARE): Canberra, Australia, 2018.
2. Kamilaris, A.; Prenafeta-Boldú, F.X. Deep learning in agriculture: A survey. *Comput. Electron. Agric.* **2018**, *147*, 70–90. [CrossRef]
3. Bac, C.W.; Hemming, J.; Van Tuijl, B.; Barth, R.; Wais, E.; van Henten, E.J. Performance evaluation of a harvesting robot for sweet pepper . *J. Field Robot.* **2017**, *34*, 1123–1139. [CrossRef]
4. Li, J.; Karkee, M.; Zhang, Q.; Xiao, K.; Feng, T. Characterizing apple picking patterns for robotic harvesting. *Comput. Electron. Agric.* **2016**, *127*, 633–640. [CrossRef]
5. Lin, G.; Tang, Y.; Zou, X.; Xiong, J.; Li, J. Guava detection and pose estimation using a low-cost RGB-D sensor in the field. *Sensors* **2019**, *19*, 428. [CrossRef]
6. Vit, A.; Shani, G. Comparing RGB-D Sensors for Close Range Outdoor Agricultural Phenotyping. *Sensors* **2018**, *18*, 4413. [CrossRef] [PubMed]
7. Zhao, Y.; Gong, L.; Huang, Y.; Liu, C. A review of key techniques of vision-based control for harvesting robot. *Comput. Electron. Agric.* **2016**, *127*, 311–323. [CrossRef]
8. Khalid, S.; Khalil, T.; Nasreen, S. A survey of feature selection and feature extraction techniques in machine learning. In Proceedings of the 2014 Science and Information Conference, London, UK, 27–29 August 2014; pp. 372–378.
9. Ke, Y.; Sukthankar, R. PCA-SIFT: A more distinctive representation for local image descriptors. *CVPR (2)* **2004**, *4*, 506–513.
10. Pass, G.; Zabih, R.; Miller, J. Comparing Images Using Color Coherence Vectors. In Proceedings of the Fourth ACM International Conference on Multimedia, Boston, MA, USA, 18–22 November 1996; pp. 65–73.
11. Ahonen, T.; Hadid, A.; Pietikainen, M. Face description with local binary patterns: Application to face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 2037–2041. [CrossRef]
12. Zhou, R.; Damerow, L.; Sun, Y.; Blanke, M.M. Using colour features of cv.'Gala'apple fruits in an orchard in image processing to predict yield. *Precis. Agric.* **2012**, *13*, 568–580. [CrossRef]
13. Song, Y.; Glasbey, C.; Horgan, G.; Polder, G.; Dieleman, J.; Van der Heijden, G. Automatic fruit recognition and counting from multiple images. *Biosyst. Eng.* **2014**, *118*, 203–215. [CrossRef]
14. Luo, L.; Tang, Y.; Zou, X.; Wang, C.; Zhang, P.; Feng, W. Robust grape cluster detection in a vineyard by combining the AdaBoost framework and multiple color components. *Sensors* **2016**, *16*, 2098. [CrossRef] [PubMed]
15. Wang, C.; Lee, W.S.; Zou, X.; Choi, D.; Gan, H.; Diamond, J. Detection and counting of immature green citrus fruit based on the local binary patterns (lbp) feature using illumination-normalized images. *Precis. Agric.* **2018**, *19*, 1062–1083. [CrossRef]
16. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
17. Liu, W.; Wang, Z.; Liu, X.; Zeng, N.; Liu, Y.; Alsaadi, F.E. A survey of deep neural network architectures and their applications. *Neurocomputing* **2017**, *234*, 11–26. [CrossRef]

18. Guo, Y.; Liu, Y.; Georgiou, T.; Lew, M.S. A review of semantic segmentation using deep neural networks. *Int. J. Multimed. Inf. Retr.* **2018**, *7*, 87–93. [CrossRef]
19. Zhang, Q.; Yang, L.T.; Chen, Z.; Li, P. A survey on deep learning for big data. *Inf. Fusion* **2018**, *42*, 146–157. [CrossRef]
20. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 24–27 June 2014; pp. 580–587.
21. Girshick, R. Fast r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 1440–1448.
22. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2015; pp. 91–99.
23. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
24. Sa, I.; Ge, Z.; Dayoub, F.; Upcroft, B.; Perez, T.; McCool, C. Deepfruits: A fruit detection system using deep neural networks. *Sensors* **2016**, *16*, 1222. [CrossRef]
25. Bargoti, S.; Underwood, J. Deep fruit detection in orchards. In Proceedings of the 2017 IEEE International Conference on Robotics and Automation (ICRA), Singapore, 29 May–3 June 2017; pp. 3626–3633.
26. 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.* **2019**, *163*, 104846. [CrossRef]
27. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 21–37.
28. Fu, C.Y.; Liu, W.; Ranga, A.; Tyagi, A.; Berg, A.C. Dssd: Deconvolutional single shot detector. *arXiv* **2017**, arXiv:1701.06659.
29. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 779–788.
30. Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
31. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
32. Tian, Y.; Yang, G.; Wang, Z.; Wang, H.; Li, E.; Liang, Z. Apple detection during different growth stages in orchards using the improved YOLO-V3 model. *Comput. Electron. Agric.* **2019**, *157*, 417–426. [CrossRef]
33. Koirala, A.; Walsh, K.; Wang, Z.; McCarthy, C. Deep learning for real-time fruit detection and orchard fruit load estimation: Benchmarking of 'MangoYOLO'. *Precis. Agric.* **2019**, *20*, 1107–1135. [CrossRef]
34. Milletari, F.; Navab, N.; Ahmadi, S.A. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In Proceedings of the 2016 Fourth International Conference on 3D Vision (3DV), Stanford, CA, USA, 25–28 October 2016; pp. 565–571.
35. Uijlings, J.R.; Van De Sande, K.E.; Gevers, T.; Smeulders, A.W. Selective search for object recognition. *Int. J. Comput. Vis.* **2013**, *104*, 154–171. [CrossRef]
36. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
37. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [CrossRef] [PubMed]
38. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 834–848. [CrossRef] [PubMed]
39. Chen, L.C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv* **2017**, arXiv:1706.05587.
40. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 801–818.

41. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
42. McCool, C.; Sa, I.; Dayoub, F.; Lehnert, C.; Perez, T.; Upcroft, B. Visual detection of occluded crop: For automated harvesting. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 2506–2512.
43. Bargoti, S.; Underwood, J.P. Image segmentation for fruit detection and yield estimation in apple orchards. *J. Field Robot.* **2017**, *34*, 1039–1060. [CrossRef]
44. Li, Y.; Cao, Z.; Xiao, Y.; Cremers, A.B. DeepCotton: In-field cotton segmentation using deep fully convolutional network. *J. Electron. Imaging* **2017**, *26*, 053028. [CrossRef]
45. Zeiler, M.D.; Fergus, R. Visualizing and understanding convolutional networks. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 818–833.
46. Yao, J.; Yu, Z.; Yu, J.; Tao, D. Single Pixel Reconstruction for One-stage Instance Segmentation. *arXiv* **2019**, arXiv:1904.07426.
47. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
48. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
49. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.
50. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
51. Wu, Y.; He, K. Group normalization. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
52. Silberman, N.; Guadarrama, S. TensorFlow-Slim Image Classification Model Library. Available online: https://github.com/tensorflow/models/tree/master/research/slim (accessed on 21 May 2019).
53. Tensorflow-yolo-v3. Available online: https://github.com/mystic123/tensorflow-yolo-v3 (accessed on 17 January 2019).
54. Ren, S.; He, K.; Girshick, R.; Sun, J. Py-Faster-Rcnn. Available online: https://github.com/rbgirshick/py-faster-rcnn (accessed on 29 December 2018).
55. TF Image Segmentation: Image Segmentation Framework. Available online: https://github.com/warmspringwinds/tf-image-segmentation (accessed on 14 March 2018).
56. Garcia-Garcia, A.; Orts-Escolano, S.; Oprea, S.; Villena-Martinez, V.; Garcia-Rodriguez, J. A review on deep learning techniques applied to semantic segmentation. *arXiv* **2017**, arXiv:1704.06857.
57. Everingham, M.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes (voc) challenge. *Int. J. Comput. Vis.* **2010**, *88*, 303–338. [CrossRef]
58. Powers, D.M. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *J. Mach. Learn. Technol.* **2011**, *2*, 37–63.
59. Wang, Q.; Zhang, Q. Three-dimensional reconstruction of a dormant tree using rgb-d cameras. In Proceedings of the 2013 Kansas City, Kansas City, MI, USA, 21–24 July 2013; p. 1.

5 Fruit Detection, Segmentation and 3D Visualisation of Environments in Apple Orchards

Robotic vision is required to understand the working environments from the sensory data and guide the robotic arm to detach the fruits. Our previous work developed a semantic segmentation and detection combined one-stage network DasNet-v1. From the harvesting experiments, it is important for vision detector to segment correspond region of each fruit as it can provide abundant information of shape, size, and other information, especially for those overlapped fruits. However, Semantic segmentation returns the mask for each class instead of each object. In this work, a improved multi-task one-stage detector DasNet-v2 is further developed. Compared to the DasNet-v1, DasNet-v2 combines instance segmentation branch in detection branch, to perform instance segmentation on each fruit. The semantic segmentation branch of DasNet can be used to segment the elements in background, such as branches or leaves in orchards. From the experiment results, DaSNet-v2 achieves 0.868, 0.88 and 0.873 on recall and precision of detection, and accuracy of instance segmentation on fruits, and 0.794 on the accuracy of branches segmentation, respectively. The average running time and weight size of light-weight DasNet-v2 are 55ms and 8.1M by using a Nvidia GTX-1070, respectively. Experimental results show DasNet-v2 can robustly and efficiently perform the vision sensing for robotic harvesting in apple orchards.

Computers and Electronics in Agriculture 171 (2020) 105302

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Fruit detection, segmentation and 3D visualisation of environments in apple orchards

Hanwen Kang, Chao Chen*

Department of Mechanical and Aerospace Engineering, Monash University, Melbourne, Australia

ARTICLE INFO

Keywords:
Fruit detection
Fruit segmentation
Branch segmentation
Deep learning
Robotic harvesting

ABSTRACT

Development of an accurate and reliable fruit detection system is a challenging task. There are many complex conditions in orchard environments, such as changing illumination, appearance variation, and occlusion. Robotic vision is required to understand the working environments from the sensory data and guide the robotic arm to detach the fruits. In our previous work, a deep neural network DaSNet-v1 was developed to perform detection and segmentation on fruits and branches in orchard environments. However, semantic segmentation returns the mask for each class instead of each object. Segmentation on each fruit is important as it can provide abundant information of each object, especially for those overlapped fruits. This work presents an improved deep neural network DaSNet-v2, which can perform detection and instance segmentation on fruits, and semantic segmentation on branches. DaSNet-v2 is tested and validated by experimental results obtained from field-testing in an apple orchard. From the experiment results, DaSNet-v2 with resnet-101 achieves 0.868, 0.88 and 0.873 on recall and precision of detection, and accuracy of instance segmentation on fruits, and 0.794 on the accuracy of branches segmentation, respectively. DaSNet-v2 with light-weight backbone resnet-18 achieves 0.85, 0.87 and 0.866 on recall and precision of detection, and accuracy of instance segmentation on fruits, and 0.775 on the accuracy of branches segmentation, respectively. The average running time and weight size of light-weight DaSNet-v2 are 55 ms and 8.1 M, respectively. Experimental results show DaSNet-v2 can robustly and efficiently perform the vision sensing for robotic harvesting in apple orchards.

1. Introduction

Nowadays, with the increasing cost and difficulty in availability of the labour resource (ABARES, 2018), the agricultural industry requires transformation from the labour-intensive industry to the technology-intensive industry. Robotic technology has shown a promising prospect in terms of improving the efficiency and yield of agriculture production. Different from the harvesting equipments which are designed to perform autonomous harvesting of commercial crops such as wheat and soybean in the structured working environments, to design a robotic system for automatic harvesting of fruits in orchard environments is much more challenging (Vasconez et al., 2019). Among the challenging issues in developing a fruit harvesting robot, the vision system is a crucial issue since it senses the working environment and guides the robotic arm to detach the fruits. Due to the complex conditions in real working environments, issues such as densely arranged branches and fruits in orchards should be taken into account when designing a fruit harvesting robots. In other words, fruit harvesting robots are required to understand the working environment to increase the rate of success during the harvesting (Zhao et al., 2016). Meanwhile, other environmental factors, such as various illumination conditions, changing object appearances, and occlusion or overlap of objects, can also critically affect the performance of the robotic vision system. In the previous work (Kang and Chen, 2019), a multi-task deep neural network DaSNet-v1 was developed, which can perform detection and semantic segmentation on fruits and branches in orchard environments. However, semantic segmentation can only segment the images into different classes while lacking capability of segmenting each object within the class (which also known as instance segmentation). Instance segmentation of each fruit is important as it can provide geometric property (shape and size) of each fruit, and such information can be used to compute the poses (RGB-D camera applied) of the objects. Therefore, further development of the techniques to obtain instance segmentation of each object is demanded.

In this work, an improved multi-task deep neural network DaSNet-v2 is developed to perform multi-task vision sensing for robotic harvesting in apple orchards. Firstly, DaSNet-v2 combines multi-task into the network architecture, which can perform detection and instance segmentation on fruits, and semantic segmentation on branches. Secondly, the network architecture of DaSNet-v2 is optimised compared to the DaSNet-v1 to

* Corresponding author.
E-mail addresses: hanwen.kang@monash.edu (H. Kang), chao.chen@monash.edu (C. Chen).

https://doi.org/10.1016/j.compag.2020.105302
Received 21 November 2019; Received in revised form 19 February 2020; Accepted 20 February 2020
Available online 17 March 2020

obtain better performance on detection and segmentation. Additional, a light-weight designed DaSNet-v2 (light-weight backbone applied) is trained and validated in this work to ensure the computational availability of the model on the embedded computing devices. DaSNet-v2 is tested and validated by experimental results obtained from field-test in an apple orchard. 3D visualisation of experimental results by means of the DaSNet-v2 is also illustrated in this work.

The rest of the paper is organised as follows. Section 2 reviews the related works. Sections 3 and 4 introduce the methodology and experiment of the work, respectively. In Section 5, the conclusions and future work are presented.

2. Literature review

Vision sensing in fruit orchards has been extensively studied. Currently, there are two classes of approaches: traditional machine-learning based algorithms and deep-learning based algorithms. Traditional machine-learning based algorithms apply feature descriptors to extract object features from the sensory data and machine-learning based classifier to perform classification, detection, or segmentation (Kapach et al., 2012). There are numbers of work which have applied traditional machine-learning based algorithms on vision sensing in agricultural applications (Vibhute and Bodhe, 2012; Zhao et al., 2016). Nguyen et al. (2016) applied colour features and geometric features to encode the appearance of the red apples. Then a clustering algorithm based on Euclidean distance in feature space is used to segment and detect the fruits from the input images. The similar processing techniques of performing segmentation and detection in vision sensing in orchard environment are also presented in several works (Zhou et al., 2012; McCool et al., 2016; Lin et al., 2019a; Liu et al., 2018). Recently, Wang and Lihong (2018) applied multiple image features and Latent Dirichlet Allocation (LDA) model to perform unsupervised instance/semantic segmentation of the plants and fruits in the greenhouse environments.

The development of deep-learning algorithms is more recent. Compared to the traditional machine-learning based algorithms, deep-learning based algorithms have demonstrated higher accuracy on detection, and segmentation (Han et al., 2018). Deep-learning based algorithms can be classified into two classes: two-stage detector and one-stage detector (Lin et al., 2017). The representative work of the two-stage detector is the Region Convolution Neural Network (RCNN), which includes fast/faster-RCNN (Girshick, 2015; Ren et al., 2015) and mask-RCNN (He et al., 2017). Faster-RCNN applies Region Proposal Network (RPN) and Region of Interest (RoI) pooling to combine the RoI searching and classification into a single network architecture, which increases the computational efficiency of the model. Mask-RCNN further combines instance segmentation into the detection network, which allows the network to segment the corresponding area for each object within the images. On the other hand, the representative work of the one-stage detector is You Only Look Once (YOLO) (Redmon and Farhadi, 2018) and Single Shot Detection (SSD) (Liu et al., 2016). One-stage detector predicts the object on each grid of feature maps, and it achieves similar performance with improved computational efficiency compared to the RCNN. Recently, Single Pixel Reconstruction Network (SPRNet) (Yao et al., 2019) improves the one-stage detector by introducing the instance segmentation into the network architecture, which allows one-stage detector to perform multi-task vision sensing similar to the mask-RCNN.

Recently, deep-learning based algorithms are being studied and applied in many agricultural applications (Kamilaris and Prenafeta-Boldú, 2018). Sa et al. (2016),Bargoti and Underwood (2017) applied faster-RCNN in multiple-classes fruit detection, and accurate detection performance was reported from both of work. Liu et al. (2019) applied faster-RCNN on detection of kiwifruit by using RGB and NIR images, an average-precision of 0.904 was reported from their work. Yu et al. (2019) applied mask-RCNN in the application of strawberry harvesting in a non-structured environment. Tian et al. (2019) applied YOLO-v3 in the monitoring of apple growth during different stages, an F_1 score of 0.817 was achieved in their work. Kang and Chen (2019) combined the semantic segmentation and detection into a one-stage detector, to perform the fruit detection and branch segmentation in the apple orchard for robotic harvesting. Other deep-learning based algorithms such as Fully Convolution Network (FCN) (Long et al., 2015) are also being studied and applied in performing vision sensing in the agriculture applications (Lin et al., 2019b; Xu et al., 1873).

3. Methodologies and materials

3.1. Network architecture

DaSNet-v2 follows the network architecture design of the one-stage detection network (such as YOLO), as shown in Fig. 1. It applies a 5-

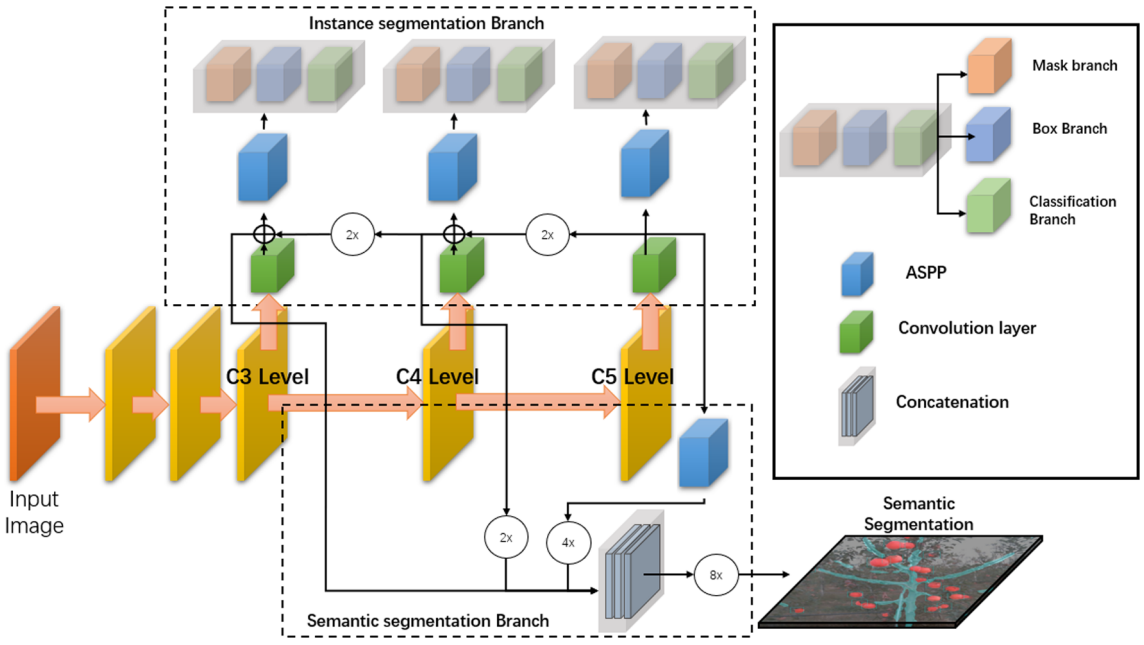


Fig. 1. DaSNet-v2 includes an instance segmentation branch and a semantic segmentation branch for detection and segmentation on apples and branches.

levels network for images classification as the backbone, which generates feature maps of 1/8, 1/16, and 1/32 size of the input image from the C3, C4 and C5 levels, respectively. A 3-level Feature Pyramid Network (FPN) is applied in the DaSNet-v2 to receive and fuse the feature maps from the C3, C4 and C5 level of the backbone to generate the detection and segmentation of fruits and branches. Feature maps from different levels of FPN are in different resolutions and contain the information or features of the objects in different scales. Therefore, the feature maps from different levels of FPN are used to detect the objects in different scales. For example, feature maps in lower-level of the FPN (C3 level) are used to detect the objects in small-scale while feature maps in higher-level (C5 level) are used to detect the objects in large-scale. Meanwhile, feature maps from the higher-level of the network contain more semantic information of objects, which can improve the accuracy of object classification in the detection. Therefore, the feature maps from the C5 level and C4 level of are two times upsampled and be added to the feature maps of the C4 level and C3 level by the FPN, respectively.

On each level of FPN, an instance segmentation branch which predicts the bounding boxes and masks for objects is applied. Besides, a semantic segmentation branch which is used to segment branches from images is grafted on the FPN. Semantic segmentation branch receives the feature maps of the C3, C4, and C5 levels of the FPN (as shown in Fig. 1). By combining the outputs of the instance segmentation branch and the semantic segmentation branch, the processing results of input images are generated.

3.1.1. Instance segmentation branch

Instance segmentation branch is applied to predict, classify and segment the objects from output feature maps of each level of FPN. The instance segmentation branch of DaSNet-v2 follows the design developed in SPRNet. The instance segmentation branch includes the boxes branch, classes branch, and masks branch to predict bounding boxes, classes, and masks of objects, respectively. SPRNet applies a shared decoder to reconstruct the instance masks for objects from individual positive pixels within the feature maps. To encode the multi-scale rich features of an object into a single pixel within the feature maps, the Atrous Spatial Pyramid Pooling (ASPP) is applied before the mask branch in the SPRNet. Our implemented instance segmentation branch in DaSNet-v2 (as shown in Fig. 2) is different from the SPRNet. DaSNet-v2 applies ASPP before the instance segmentation branch (including boxes branch, classes branch, and masks branch) as our implementation suggest that such setup can improve the localisation accuracy of the

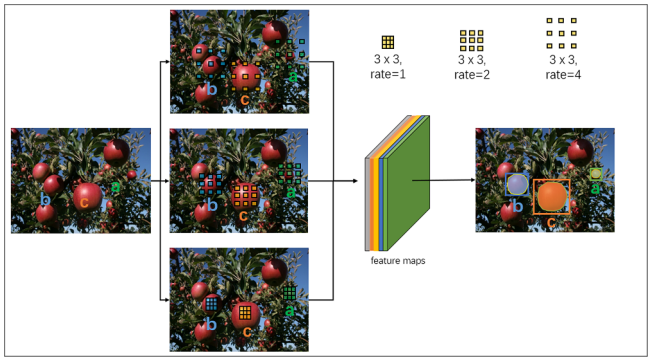


Fig. 3. ASPP applies atrous convolution kernels with given dilate rates to encode extract features of objects at different scales.

bounding boxes. The mask branch of DaSNet-v2 is simplified compared to the SPRNet in terms of improving computational efficiency.

The applied ASPP (as shown in Fig. 3) uses three dilation convolution kernels (3×3) with dilation rate 1, 2 and 4 and a 1×1 kernel to encode the multi-scale features into a single pixel (The implementation suggests that ASPP with large dilation rate may introduce redundant information which can lead to low recall on detection of overlapped objects). The reconstructed masks from the mask branch will be rescaled to the size of the predicted object box. Each level of FPN in the DaSNet-v1 has two preset anchor boxes (3-levels in total). The experimental results in Table 2 show that such setup can efficiently cover the changing shape of bounding boxes in apple detection.

3.1.2. Semantic segmentation branch

Instance segmentation branch can detect and segment the fruits from the input images to stand the location, size and shape of the fruits in working space. However, such information is limited to guide a robot to perform successful harvesting in the orchards setting. There are many obstacles which are presented in the working space of orchards, such as densely arranged branches. To provide more information for robots to understand the current working space, a semantic segmentation branch is applied to perform the branches segmentation from the input images.

The DaSNet-v2 applies 'Encoder-Decoder with atrous convolution' which were developed in Deeplab-v3+ (Chen et al., 2018) to perform the branches segmentation. The semantic segmentation branch of the DaSNet-v2 receives the feature maps of the C3, C4, and C5 levels of the

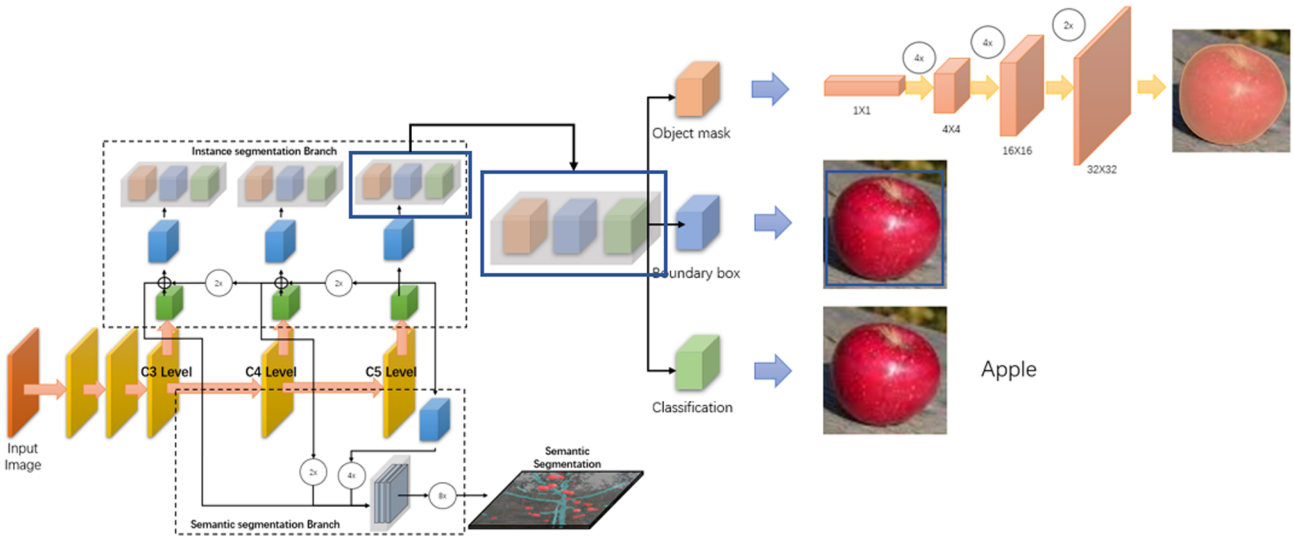


Fig. 2. Network architecture design of the instance segmentation branch applied in the DaSNet-v2, which includes a box branch, a classification branch, and a mask branch.

FPN. The feature maps of C5 level are processed by the ASPP to extract features in different scales. To introduce detail features of objects, the feature maps of C3 and C4 level are concatenated with the processed feature maps from the C5 level. The output tensor of the semantic segmentation branch is 8 times upsampled to match the size of the input images. Different from the DaSNet-v1, the semantic segmentation branch of the DaSNet-v2 only perform segmentation on branches (including branches and trunks), since segmentation of fruits has been included in the instance segmentation branch. .

3.1.3. Compared to the DaSNet-v1

Compared to the DaSNet-v1, the performance of DaSNet-v2 is improved in the following points. Firstly, DaSNet-v2 improves the network model by introducing the instance segmentation into the detection branch. This improvement allows the vision system to provide geometric information (such as shape) of each object. Secondly, DaSNet-v2 optimises the architecture design of the FPN and semantic segmentation branch, compared to the DaSNet-V1. On the one hand, DaSNet-v2 adopts a simplified FPN design, which improves training efficiency and performance of the model. On the other hand, DaSNet-v2 adopts the 'Encoder-Decoder with atrous convolution' from the Deeplab-v3+ to improve the accuracy of branches segmentation.

3.2. Visualisation of working space

In the fruit orchards which is not optimised for robotic operation, the branch and fruits are presented randomly, which can heavily affect the performance of harvesting robots. Densely arranged branches can obstruct the path of robotic arms or even damage the robotic arm (Megalingam et al., 2017). Besides, densely arranged fruits and different types of the stem-branch joint of fruits may also affect the success rate of fruit harvesting (Lin et al., 2019b). To provide a more intuitive understanding of the working environments and guide the manipulator and gripper, 3D modelling and visualisation of the working space in orchards are important (Comba et al., 2018).

DaSNet-v2 can detect and segment the fruit and branches in the orchard environments. For the fruit class, different colours are assigned to the detected fruits to stand their shape and corresponding area. For the branches, a unified colour is assigned to the segmented mask. Other elements within the working space such as ground, fence and leaves are presented in black pixel. Leaf segmentation is not included in the task of DaSNet-v2 since our previous experiments suggest that leaves only block the sight of vision system without obstructing path for picking during the harvesting. PPTK tool-kit (Heremaps, 2018) is used to visualise the 3D point clouds of the working space, an example of 3D visualisation of an orchard environment is shown in Fig. 4.

3.3. Implementation details

3.3.1. Data augmentation

Data augmentation plays an important role in the training of the deep-learning model. To avoid network over-fitting to the training data, extensive image augmentations are introduced. The applied data augmentation includes random crop, random scale (from 50% to 150%), random flip (horizontal only), and random rotation (± 10°). Further, the randomly adjust of brightness (0.5–1.5) and saturation (0.5–1.5) of images in HSV colour space are also applied in the augmentation.

3.3.2. Training method

Focal Loss (FL) (Lin et al., 2017) is used in the training to balance the uneven distribution of the foreground class objects (obj) and background class objects (noobj). The focal loss can be expressed as:

$$FL(p) = \sum^{obj} -\alpha(1-p)^{\gamma} log(p) + \sum^{noobj} -\beta(p)^{\gamma} log(1-p) \quad (1)$$

p is the confidence score of the object. α, β, and γ are the inner parameters to adjust the profile of the loss function. We set α, β, and γ as 8, 0.5, and 2 in the training, respectively. Other training tasks including regression of bounding boxes and classification follow the design of the YOLO-V3 (Redmon and Farhadi, 2018). Cross-entropy loss is used in the training of the instance segmentation and semantic segmentation tasks. Adam-optimizer is used in the training of the network model. The learning rate, decay rate, and batch size used in the training are 0.01, 0.9 and 32, respectively. The backbone weights of the network are fixed in the first 100 epochs of the training. Then the overall network is trained for another 50 epochs.

3.3.3. Other details

The programming of DaSNet-v2 was performed by using TensorFlow-slim image classification model library (Silberman and Guadarrama, 2016) in Ubuntu 16.04. 3D visualisation of the point cloud is achieved by using PPTK tool-kit. The DaSNet-v2 is trained on the GTX-1080Ti (Nvidia, United States) and be tested on Jetson-TX2 (Nvidia, United States) and GTX-1080Ti. Intel RealSense D-435 RGB-D camera (Intel, United States) is used to perform vision sensing in the field-test. It is controlled by using the realsense-ros SDK (Intel-Corp, 2018) in ROS-kinetic on Ubuntu 16.04.

To ensure the computation availability of the DaSNet-v2 model on the Jetson-TX2, a light-weight modified resnet-18 (Kang and Chen, 2019) (as shown in Fig. 5) was used as the backbone in the model of DaSNet-v2. Resnet-50 and resnet-101 (He et al., 2016) were also used as the backbone in the model of DaSNet-v2. The implemented code and ImageNet pre-trained weights of the resnet-50 and resnet-101 were from the Github publicly code library (Taylor et al., 2018), the resnet-18 was pre-trained on Cifar (Krizhevsky et al., 2009).

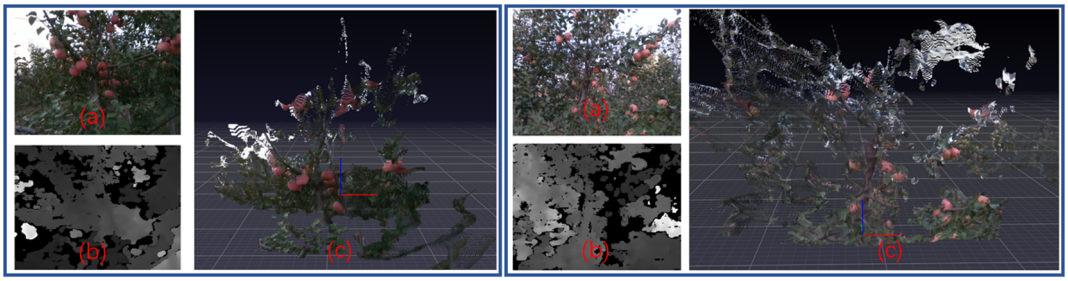


Fig. 4. (a), (b), and (c) of figure (i) and (ii) are the RGB images, depth images, and point clouds in 3D space, respectively.

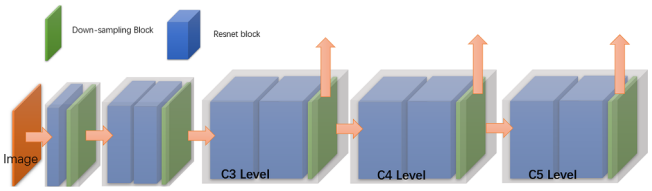


Fig. 5. Architecture of the resnet-18, it applies bottleneck designed resnet block to reduce the weight size and improve the computational efficiency.

Table 1
Numbers of image data from different dataset in training, validation, and test set.

	Device	Type	Number	Training	Validation	Test
A	C-615	RGB	427	227	50	150
B	D-435	RGB	382	192	50	140
C	D-435	RGB-D	468	148	50	270
Total	–	–	1277	567	150	560

4. Experiment and discussion

4.1. Data collection

Both RGB-D images and RGB images were collected from the apple orchard located at Qingdao, China. The collection time of the image data was from 10:00 am to 21:00 pm through the day by using the Intel RealSense D-435 RGB-D camera and Logitech webcam-C615 (Logitech, Switzerland). The images were collected at the distance of 0.5–1.5 m from the camera to apple trees, which is the distance from the camera to trees during the robotic harvesting. There were about 400 RGB-D images and more than 800 RGB images which were collected during the field-test in the apple orchard (as shown in Table 1). Image data A, B and C were collected by using handheld webcam C-615, depth-camera D-435 on the robotic arm (see Fig. 6), and handheld depth-camera D-435, respectively. We used 148 RGB-D images (only RGB information was used in training) and 419 RGB images as the training set and applied another 150 images as the validation set. The rest of the images were used to evaluate the performance of the trained model.

Table 2
Comparison of performance on detection and instance segmentation among different networks models on GTX-1080Ti, image size (640 × 480).

Model	F_1	Recall	Precision	IoU_{box}	IoU_{mask}	Time
Faster-RCNN	0.852	0.836	0.872	0.858	–	127 ms
YOLO-v3	0.86	0.852	0.87	0.851	–	43 ms
DaSNet-v1	0.863	0.857	0.875	0.856	–	54 ms
Mask-RCNN	0.868	0.86	0.882	0.863	0.878	158 ms
DaSNet-v2	0.873	0.868	0.88	0.861	0.873	70 ms

Table 3
Comparison of performance on detection by DaSNet-v2 with different backbones on GTX-1080Ti, image size (640 × 480).

Backbone	F_1	Recall	Precision	IoU_{box}	IoU_{mask}	Time
Resnet-18	0.857	0.85	0.87	0.858	0.866	54 ms
Resnet-50	0.868	0.861	0.876	0.859	0.872	64 ms
Resnet-101	0.873	0.868	0.88	0.861	0.873	70 ms

4.2. Evaluation method

The performance evaluation includes three tasks, which are the accuracy evaluation on fruit detection and segmentation, and branch segmentation. To evaluate accuracy of fruit detection, Intersection of Union (IoU) and F_1 score are used as performance metric in this work. IoU computes a ratio of the intersection and the union of two sets (Garcia-Garcia et al., 2017). In the case of detection and segmentation, the sets can be bounding boxes or masks of prediction and ground-truth. IoU measures the localisation accuracy of bounding box or accuracy of segmentation. When assessing the performance of detection, we compute the IoU by using bounding boxes (denoted in IoU_{box}) between prediction and ground truth. The predicted objects with IoU_{box} and confidence score higher than 0.5 will be treated as true-positive. F_1 score measures the detection performance by using the *Recall* and *Precision*. *Recall* measures the fraction of true-positive objects that are successfully detected, while *Precision* measures the fraction of true-positive objects in the predictions. The expression of the *Precision*, *Recall* and F_1 score are listed as follow:

$$Precision = \frac{TruePositive(TP)}{TruePositive(TP) + FalsePositive(FP)} \quad (2)$$

Fig. 6. (a) Setting of the apple orchard, (b) apple harvesting robot and mobile platform, (c) setting of depth camera.

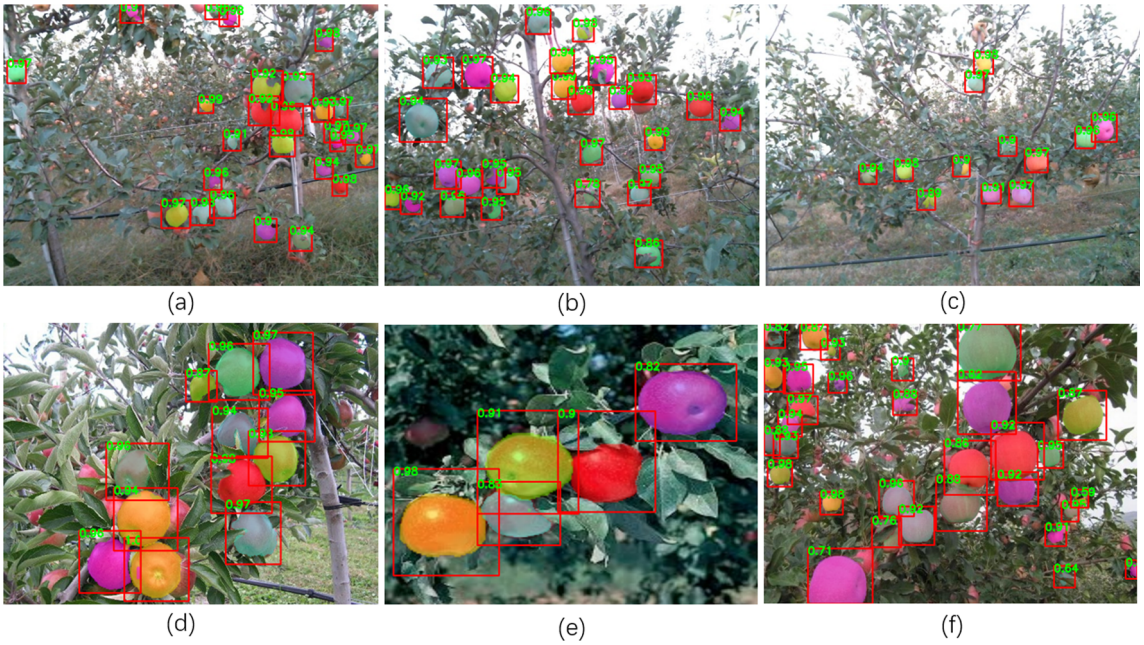


Fig. 7. Instance segmentation of fruits by using DaSNet-v2 with resnet-101. Each fruit is drawn in a distinguished colour, green numbers are the confidence values of detected objects within the boxes. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

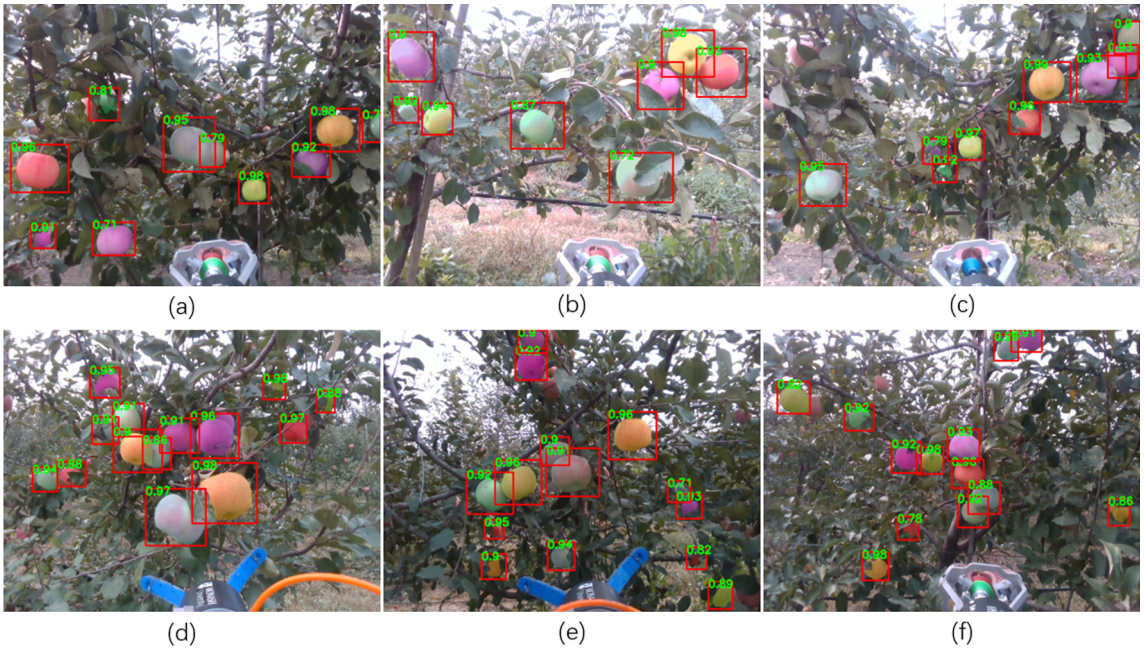


Fig. 8. Instance segmentation of fruit by using DaSNet-v2 with resnet-101, the images are collected by depth-camera under the view of robotic arm during harvesting. Each fruit is drawn in a distinguished colour.

Table 4
Comparison of performance on semantic segmentation of branches among different models, image size (640 × 480).

Model (Backbone)	IoU_{branch}	Time
FCN-8s (Resnet-101)	0.757	52 ms
DaSNet-v1 (Resnet-101)	0.772	54 ms
DaSNet-v2 (Resnet-101)	0.794	70 ms

Table 5
Comparison of performance of DaSNet-v2 on semantic segmentation with different backbones, image size (640 × 480).

Backbone	IoU_{branch}	Time
Resnet-18	0.775	54 ms
Resnet-50	0.788	64 ms
Resnet-101	0.794	70 ms

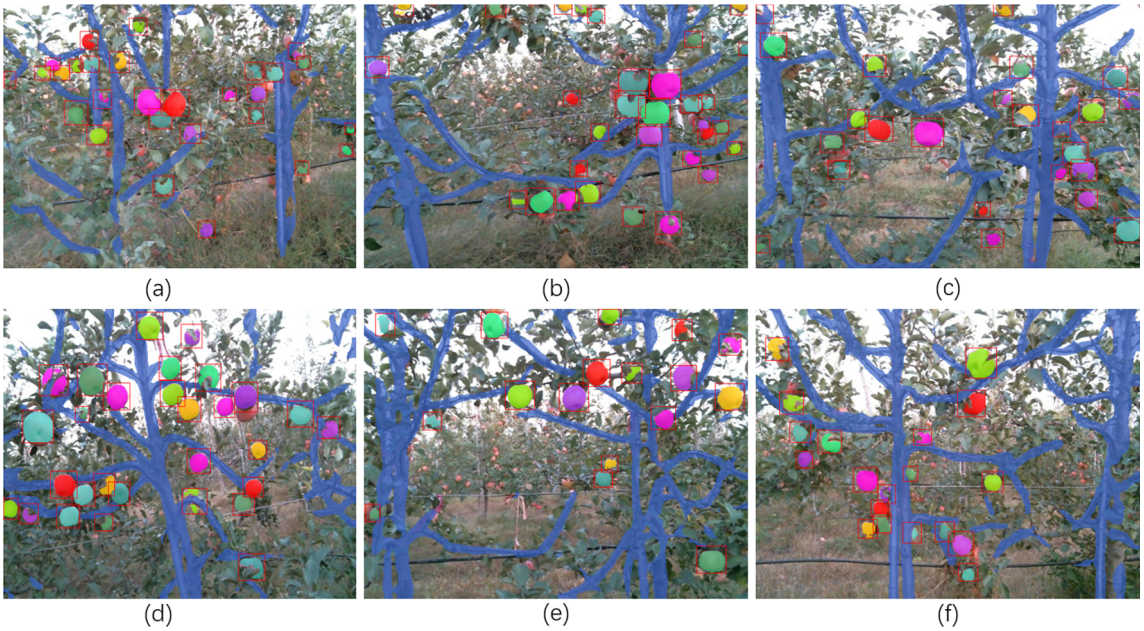


Fig. 9. Detection and segmentation of fruits and branches by using the DaSNet-v2 in the orchard. Fruits are drawn in distinguished colours, branches are drawn in the colour of blue. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

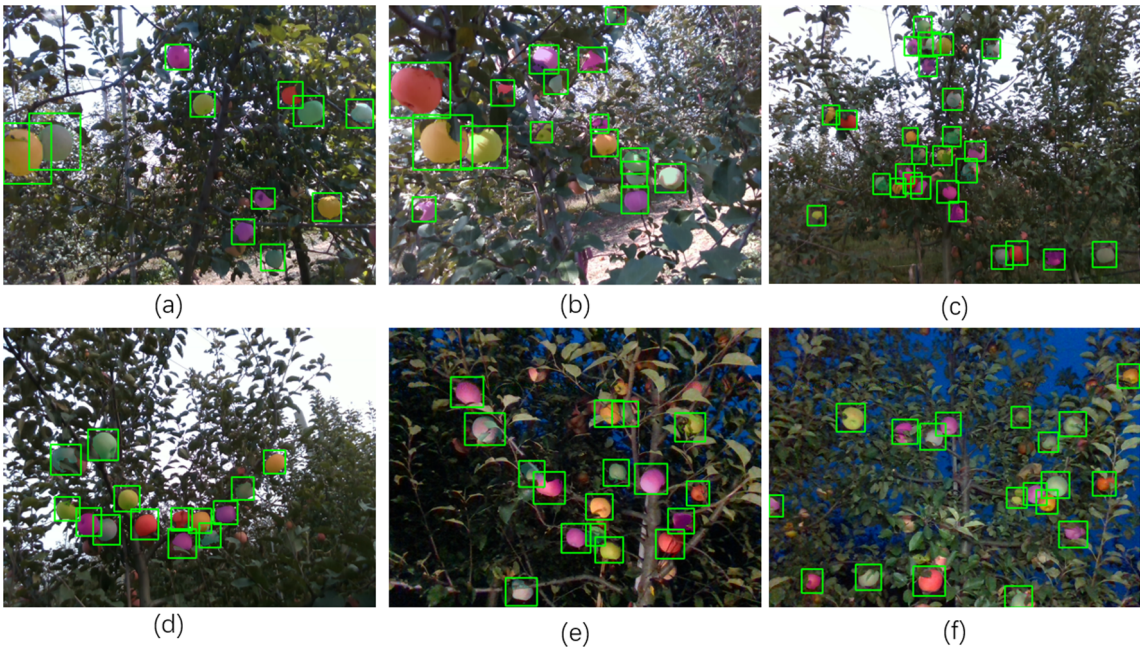


Fig. 10. Detection and segmentation of fruits by using the DaSNet-v2 in different times. (a) and (b) are images taken in 11:00 am-13:00 pm, (c) and (d) are images taken between 4:00 pm to 6:00 pm, (e) and (f) are images taken between 7:00 pm to 9:00 pm under artificial lighting.

$$Recall = \frac{TP}{TP + FalseNegative\,(FN)} \tag{3}$$

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4}$$

In the performance evaluation, the F_1 score is calculated by averaging the F_1 score of each image within the test set. The IoU is also used to evaluate the accuracy of instance segmentation and semantic segmentation of network models on fruits and branches, which are denoted as IoU_{mask} and IoU_{branch}, respectively.

4.3. comparison to state of the art

4.3.1. Evaluation of detection and instance segmentation

A series of experiments were conducted to compare the detection performance among the DaSNet-v2 and DaSNet-v1, YOLO-v3, faster-RCNN and the mask-RCNN. YOLO-v3 is the representative work of the one-stage detector, which applies darknet-53 as the backbone and a 3-level FPN in the model. The implemented code of YOLO-v3 is from Github publicly code library (Kapica, 2019). Faster-RCNN and mask-RCNN are the representative works of the two-stage detector. The implemented code of faster-RCNN (Jia et al., 2014) use VGG-19 (Simonyan and Zisserman, 2014) as the backbone, while FPN is not applied in the model. The implemented code of mask-RCNN (Abdulla,

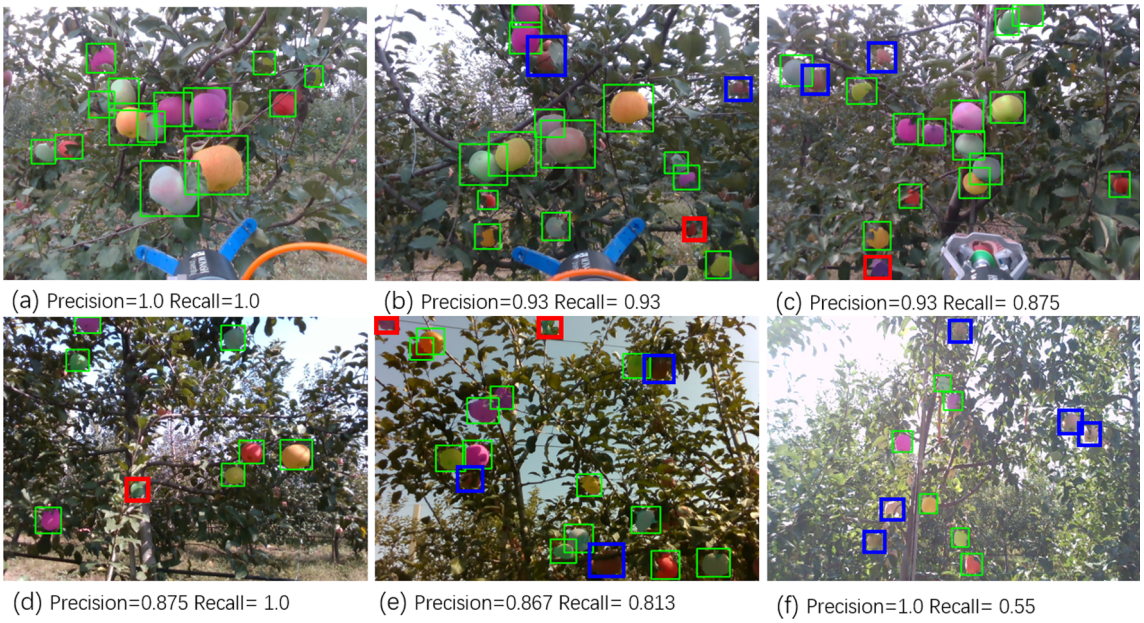


Fig. 11. Success and Failure detection by using DaSNet-v2 in the orchard. Green, red, and blue boxes represent the True-Positive (TP), False-Positive (FP), and False-Negative (FN) of detection, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 6
Comparison of computational efficiency of network models on Jetson-TX2, image size (640 × 480).

Model	Weight Size	Time
YOLO-v3 (darknet-53)	246 M	235 ms
DaSNet-v1 (resnet-101)	192 M	306 ms
DaSNet-v2 (resnet-18)	8.1 M	342 ms
DaSNet-v2 (resnet-101)	187 M	437 ms
Faster-RCNN (VGG-19)	533 M	1.1 s
Mask-RCNN (resnet-101)	244 M	1.3 s

2017) (FPN applied), DaSNet-v1, and DaSNet-v2 applies resnet-101 as backbone. In the experiment, all the network models were trained and tested on our collected training set and test set. We set 0.5 as the threshold for confidence and IoU_{box} in all network models. The experimental results among different network models are shown in Table 2. The comparison results among DaSNet-v2 with different backbones are shown in Table 3.

As shown in Table 2, DaSNet-v2 and mask-RCNN outperform other network models in terms of the fruit detection. The F_1 score of DaSNet-v2 and mask-RCNN are 0.873 and 0.868, respectively. The implemented code of faster-RCNN does not adopt FPN in the network model. Experimental results show that it has lower recall on small-scale

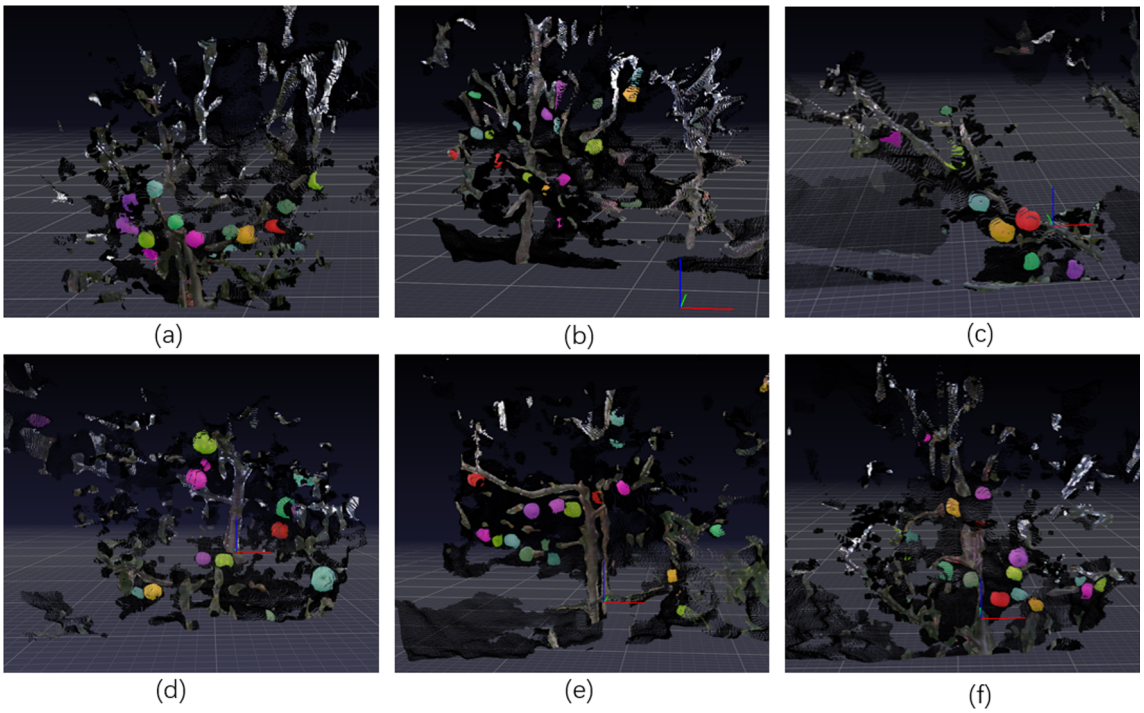


Fig. 12. 3D visualisation of the processed orchard by using PPTK. The fruits are drawn in distinguished colours, branches are drawn in original colour.

objects compared to the other network models. Therefore, a lower score on recall and F_1 score are reported on fruit detection by using faster-RCNN, which are 0.836 and 0852, respectively. As shown in (a-c) of Fig. 7, many fruits in images are presented in small-scale, especially in the images which are collected from orchard environments. Compared to the DaSNet-v1, DaSNet-v2 optimises the network architecture and training procedures. Therefore, a higher score on both recall and precision are obtained by DaSNet-v2, which are 0.868 and 0.88, respectively. Compared to the faster-RCNN, mask-RCNN with FPN design achieves a higher score on both recall and precision of detection, which are 0.86 and 0.882, respectively. In terms of the instance segmentation, mask-RCNN and DaSNet-v2 achieve similar score on the accuracy of instance segmentation, which are 0.878 and 0.873, respectively. Figs. 7 and 8 show the examples of detection and instance segmentation of apples by using DaSNet-v2.

From the experimental results shown in Table 2, one-stage detectors have better computational efficiency compared to two-stage detectors. The average computational time of faster-RCNN and mask-RCNN are 127 ms and 158 ms, respectively. While the average computational time of YOLO-v3, DaSNet-v1, and DaSNet-v2 are 43 ms, 54 ms, and 70 ms, respectively. DaSNet-v2 achieves similar performance on fruit detection compared to the mask-RCNN with better computational efficiency. Table 3 shows the performance of DaSNet-v2 with different backbones. To apply the DaSNet-v2 in the embedded computational device such as Jetson-TX2, a light-weight backbone Resnet-18 is adopted in the DaSNet-v2. Experimental results show that DaSNet-v2 with Resnet-18 can achieve similar performance on recall and precision of detection compared to the YOLO-v3. The recall and precision of DaSNet-v2 with Resnet-18 are 0.85 and 0.87, respectively. The weight size and average computational time of DaSNet-v2 with Resnet-18 are 8.1 MB and 54 ms (as shown in Table 6), respectively.

4.3.2. Evaluation of semantic segmentation

This experiment compares the performance of semantic segmentation between the DaSNet-v2, DaSNet-v1 and the FCN-8s. The implemented code of FCN-8s with resnet-101 is from Github publicly code library (Pakhomov et al., 2017). The experimental results are shown in Table 4.

From the experimental results shown in Table 4, the accuracy of semantic segmentation on branches achieved by DaSNet-v2 is improved compared to the DaSNet-v1 and FCN-8s. The IoU value on branches segmentation achieved by FCN-8s, DaSNet-v1 and DaSNet-v2 are 0.757, 0.772 and 0.794, respectively. Compared to the DaSNet-v1, DaSNet-v2 only applies ASPP on the feature maps from C5 level, as experimental results suggest ASPP on lower-level (such as C3 and C4) will introduce noise and lead to under-fitting of the model. Compared to the FCN-8s, DaSNet-v2 achieves 3.7% higher value on IoU_{branch}. Table 5 compares the accuracy on branches segmentation by DaSNet-v2 with different backbones. Experimental results show that backbones with better performance can improve the accuracy of branches segmentation. The IoU_{branch} achieved by DaSNet-v2 with Resnet-18, Resnet-50, and Resnet-101 are 0.775, 0.788 and 0.794, respectively.

The average computational time of DaSNet-v2 is increased compared to the DaSNet-v1, which is due to the increasing computational consumption on instance segmentation branch. Although DaSNet-v2 has shown an improved ability on branches segmentation, to classify tree under various conditions accurately is still a challenging task. The segmentation of branches and fruits by using DaSNet-v2 are shown in Fig. 9.

4.4. Visual sensing in orchards

There are various factors which are presented in orchards environments, such as illumination variation, overlapped fruits or branches, and appearance variation. These factors can heavily affect the accuracy of detection and segmentation. The DaSNet-v2 was tested in the apple orchard in different setup (including operation time and mode), the results which are processed by DaSNet-v2 are visualised in Figs. 8 and 10. Several examples of success or fail detections by DaSNet-v2 in orchard environments are shown in Fig. 11. The detection errors include two types: false-positive and false-negative, which are linked to the precision and recall of detection, respectively. From experimental results shown in Fig. 11, false-positive in detection mainly caused by false detection on leaves or branches. The reasons that lead to false-negative in detection are varied. Environment factors such as strong sunlight reflection, shadow, and appearances variation of fruits in colours, shape, occlusion, or view-angle can lead to the false-negative in detection. These factors can cause inaccurate in the detection, while the experimental results in Table 2 shows that DaSNet-v2 achieves high recall and precision on detection of apples in orchard environments.

The developed fruit harvesting robot applies Jetson-TX2 as computation centre to process vision sensing and robot control. The comparison of weight size and average computation time of different network models on Jetson-TX2 are shown in Table 6. It can be seen that the one-stage detectors,such as YOLO-v3 and DaSNet-v2, have better computational efficiency compared to the two-stage detector.

4.5. 3D visualisation of orchards

The collected RGB-D images from the orchard are processed by using the DaSNet-v2 and visualised by using the PPTK, which are shown in Fig. 12. As shown in figures, 3D point clouds with semantic information added can clearly describe the working environments of the harvesting robot in orchard environments. These information can be used to construct the 3D map of working spaces (Lang et al., 2013) and compute the pose of each fruit (Wong et al., 2017), which can increase the success rate of robotic harvesting (Bac et al., 2014). Such works will be included in the future works of development of intelligent robotic system for fruit harvesting.

5. Conclusion and future works

In this study, a multi-function deep neural network DaSNet-v2 was proposed and validated. DaSNet-v2 combines an instance segmentation branch and a semantic segmentation branch into the architecture of the one-stage detection network, which allows DaSNet-v2 to perform detection and segmentation on each fruit, and semantic segmentation on branches. Besides, DaSNet-v2 adopts FPN and ASPP to improve the performance on detection and segmentation of fruits and branches. To improve the computational efficiency of network model running on embedded computational devices, DaSNet-v2 with a light-weight backbone resnet-18 was trained and validated in this work. In the experiments, DaSNet-v2 was tested and validated by experimental results obtained from field tests in an apple orchard. DaSNet-v2 with resnet-101 achieved 0.868 and 0.88 on recall and precision of detection, 0.873 on the accuracy of fruits segmentation, and 0.794 on the accuracy of branches segmentation, respectively. DaSNet-v2 with resnet-18 achieved 0.85 and 0.87 on recall and precision of detection, 0.866 on the accuracy of fruits segmentation, and 0.757 on the accuracy of branches segmentation, respectively. The weight size and average computational time of DaSNet-v2 with resnet-18 to process an image (640 × 480) on GTX-1080Ti are 8.1 M and 54 ms, respectively. From the experiment results, DaSNet-v2 demonstrated a robust and efficient performance on vision sensing in orchards. Future work will focus on developing the orchard reconstruction algorithm based on the DaSNet-v2, corresponding control strategy for guiding the automatic robotic fruit harvesting will also be included.

CRediT authorship contribution statement

Hanwen Kang: Conceptualization, Methodology, Software, Data curation, Visualization, Validation, Writing - original draft. **Chao Chen:**

Conceptualization, Writing - review & editing, Validation, Supervision, Project administration.

Declaration of Competing Interest

The authors declare no conflict of interest.

Acknowledgement

This work is supported by ARC ITRH IH150100006 and THOR TECH PTY Ltd. We acknowledge Zijue Chen and Hongyu Zhou for their assistance in the data collection. And we also acknowledge Zhuo Chen for her assistance in preparation of this work.

References

ABARES, 2018. Australian vegetable growing farms: an economic survey, 2016-17 and 2017-18. Australian Bureau of Agricultural and Resource Economics (ABARE): Canberra.

Abdulla Waleed, 2017. Mask r-cnn for object detection and instance segmentation on keras and tensorflow, 2017. https://github.com/matterport/Mask_RCNN. [Online; accessed Sep-2019].

Bac, C. Wouter, van Henten, Eldert J., Hemming, Jochen, Edan, Yael, 2014. Harvesting robots for high-value crops: State-of-the-art review and challenges ahead. J. Field Robot. 31 (6), 888–911.

Bargoti, Suchet, Underwood, James, 2017. Deep fruit detection in orchards. In: 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp. 3626–3633.

Chen Liang-Chieh, Zhu Yukun, Papandreou George, Schroff Florian, Adam Hartwig, 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 801–818.

Comba, Lorenzo, Biglia, Alessandro, Aimonino, Davide Ricauda, Gay, Paolo, 2018. Unsupervised detection of vineyards by 3d point-cloud uav photogrammetry for precision agriculture. Comput. Electron. Agric. 155, 84–95.

Garcia-Garcia Alberto, Orts-Escolano Sergio, Oprea Sergiu, Villena-Martinez Victor, Garcia-Rodriguez Jose, 2017. A review on deep learning techniques applied to semantic segmentation. arXiv preprint arXiv:1704.06857.

Girshick Ross, 2015. Fast r-cnn. In: Proceedings of the IEEE international Conference on Computer Vision, pp. 1440–1448.

Han, Junwei, Zhang, Dingwen, Cheng, Gong, Liu, Nian, Dong, Xu., 2018. Advanced deep-learning techniques for salient and category-specific object detection: a survey. IEEE Signal Process. Mag. 35 (1), 84–100.

He Kaiming, Zhang Xiangyu, Ren Shaoqing, Sun Jian, 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778.

He Kaiming, Gkioxari Georgia, Dollár Piotr, Girshick Ross, 2017. Mask r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969.

Heremaps, 2018. heremaps/pptk, URL https://github.com/heremaps/pptk. [Online; accessed July-2019].

Intel-Corp, 2018. Intel realsense sdk 2.0, https://github.com/IntelRealSense/realsense-ros.

Jia Yangqing, Shelhamer Evan, Donahue Jeff, Karayev Sergey, Long Jonathan, Girshick Ross, Guadarrama Sergio, Darrell Trevor, 2014. Caffe: Convolutional architecture for fast feature embedding. arXiv preprint arXiv:1408.5093.

Kamilaris, Andreas, Prenafeta-Boldú, Francesc X., 2018. Deep learning in agriculture: A survey. Comput. Electron. Agric. 147, 70–90.

Kang, Hanwen, Chen, Chao, 2019. Fruit detection and segmentation for apple harvesting using visual sensor in orchards. Sensors 19 (20), 4599.

Kapach, Keren, Barnea, Ehud, Mairon, Rotem, Edan, Yael, Ben-Shahar, Oh.ad., 2012. Computer vision for fruit harvesting robots–state of the art and challenges ahead. Int. J. Comput. Vision Robot. 3 (1/2), 4–34.

Kapica Pawel, 2019. tensorflow-yolov3, https://github.com/mystic123/tensorflow-yolo-v3. [Online; accessed july-2019].

Krizhevsky Alex, Hinton Geoffrey, et al., 2009. Learning multiple layers of features from tiny images.

Lang, Dagmar, Friedmann, Susanne, Paulus, Dietrich, 2013. Semantic 3d octree maps based on conditional random fields. MVA 13, 185–188.

Lin, Guichao, Tang, Yunchao, Zou, Xiangjun, Xiong, Juntao, Fang, Yamei, 2019a. Color-, depth-, and shape-based 3d fruit detection. Precision Agric. 1–17.

Lin, Guichao, Tang, Yunchao, Zou, Xiangjun, Xiong, Juntao, Li, Jinhui, 2019b. Guava detection and pose estimation using a low-cost rgb-d sensor in the field. Sensors 19 (2), 428.

Lin, Tsung-Yi, Goyal, Priya, Girshick, Ross, He, Kaiming, Dollár, Piotr, 2017. Focal loss for dense object detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2980–2988.

Liu, Wei, Anguelov, Dragomir, Erhan, Dumitru, Szegedy, Christian, Reed, Scott, Cheng-Yang, Fu., Berg, Alexander C, 2016. Ssd: Single shot multibox detector. In: European Conference on Computer Vision. Springer, pp. 21–37.

Liu, Xiaoyang, Jia, Weikuan, Ruan, Chengzhi, Zhao, Dean, Yuwan, Gu., Chen, Wei, 2018. The recognition of apple fruits in plastic bags based on block classification. Precis. Agric. 19 (4), 735–749.

Liu Zhihao, Wu Jingzhu, Fu Longsheng, Majeed Yaqoob, Feng Yali, Li Rui, Cui Yongjie, 2019. Improved kiwifruit detection using pre-trained vgg16 with rgb and nir information fusion. IEEE Access.

Long, Jonathan, Shelhamer, Evan, Darrell, Trevor, 2015. Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440.

McCool Christopher, Sa Inkyu, Dayoub Feras, Lehnert Christopher, Perez Tristan, Upcroft Ben, 2016. Visual detection of occluded crop: For automated harvesting. In: 2016 IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp. 2506–2512.

Megalingam Rajesh Kannan, Vivek Gedela Vamsy, Bandyopadhyay Shiva, Rahi Mohammed Juned, 2017. Robotic arm design, development and control for agriculture applications. In: 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS). IEEE, pp. 1–7.

Nguyen, Tien Thanh, Vandevoorde, Koenraad, Wouters, Niels, Kayacan, Erdal, De Baerdemaeker, Josse G., Saeys, Wouter, 2016. Detection of red and bicoloured apples on tree with an rgb-d camera. Biosyst. Eng. 146, 33–44.

Pakhomov Daniil, Premachandran Vittal, Allan Max, Azizian Mahdi, Navab Nassir, 2017. Deep residual learning for instrument segmentation in robotic surgery. arXiv preprint arXiv:1703.08580.

Redmon Joseph, Farhadi Ali, 2018. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767.

Ren Shaoqing, He Kaiming, Girshick Ross, Sun Jian, 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99.

Sa, Inkyu, Ge, Zongyuan, Dayoub, Feras, Upcroft, Ben, Perez, Tristan, McCool, Chris, 2016. Deepfruits: A fruit detection system using deep neural networks. Sensors 16 (8), 1222.

Silberman N., Guadarrama, S., 2016. Tensorflow-slim image classification model library, URL https://github.com/tensorflow/models/tree/master/research/slim.

Simonyan Karen, Zisserman Andrew, 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

Taylor Robie, Hongkun Yu, Wu Neal, 2018. tensorflow-resnet. URL https://github.com/tensorflow/models/tree/master/research/resnet. [Online; accessed Nov-2018].

Tian, Yunong, Yang, Guodong, Wang, Zhe, Wang, Hao, Li, En, Liang, Zize, 2019. Apple detection during different growth stages in orchards using the improved yolo-v3 model. Comput. Electron. Agric. 157, 417–426.

Vasconez, Juan P., Kantor, George A., Auat Cheein, Fernando A., 2019. Human–robot interaction in agriculture: A survey and current challenges. Biosyst. Eng. 179, 35–48.

Vibhute, Anup, Bodhe, S.K., 2012. Applications of image processing in agriculture: a survey. Int. J. Comput. Appl. 52 (2).

Wang, Yi, Lihong, Xu., 2018. Unsupervised segmentation of greenhouse plant images based on modified latent dirichlet allocation. PeerJ 6, e5036.

Wong, Jay M., Kee, Vincent, Le, Tiffany, Wagner, Syler, Mariottini, Gian-Luca, Schneider, Abraham, Hamilton, Lei, Chipalkatty, Rahul, Hebert, Mitchell, Johnson, David M.S., et al., 2017. Segicp: Integrated deep semantic segmentation and pose estimation. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, pp. 5784–5789.

Xu, Hui, Chen, Guodong, Wang, Zhenhua, Sun, Lining, Fan, Su., 1873. Rgb-d-based pose estimation of workpieces with semantic segmentation and point cloud registration. Sensors 19 (8), 2019.

Yao Jinghan, Yu Zhou, Yu Jun, Tao Dacheng, 2019. Single pixel reconstruction for one-stage instance segmentation. arXiv preprint arXiv:1904.07426.

Yu, Yang, Zhang, Kailiang, Yang, Li, Zhang, Dongxing, 2019. Fruit detection for strawberry harvesting robot in non-structural environment based on mask-rcnn. Comput. Electron. Agric. 163, 104846.

Zhao, Yuanshen, Gong, Liang, Huang, Yixiang, Liu, Chengliang, 2016. A review of key techniques of vision-based control for harvesting robot. Comput. Electron. Agric. 127, 311–323.

Zhou, Rong, Damerow, Lutz, Sun, Yurui, Blanke, Michael M, 2012. Using colour features of cv.'gala'apple fruits in an orchard in image processing to predict yield. Precision Agric. 13 (5), 568–580.

6 Visual Perception and Modelling for Autonomous Apple Harvesting

Visual perception and modelling are essential tasks in many vision-dependent robotic tasks. Chapter 4 and 5 developed a efficient and robust multi-task one-stage detection network DasNet for fruit recognition by using RGB or RGB-D cameras. In this chapter, a robotic vision system to perform fruit recognition, modelling, and environment modelling for autonomous apple harvesting is presented. The developed vision framework applies Dasnet to perform fruit recognition and background segmentation, and a fruit modelling algorithm to estimates grasping pose of each fruit. Environment modelling algorithm applies Octrees to present the occupied obstacles within the workspace. During the operation, robotic control system computes a collision-free path and move robotic arm to pick the fruits based on vision processing results. The developed control framework is tested in both simulated and real orchard environments. Experimental results show that fruit recognition and modelling algorithm can accurately localise the fruits and compute the grasp pose in various situations. The success rate and average cycling time are 0.91 and 12s, respectively.

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 17, 2020, accepted March 28, 2020, date of publication March 31, 2020, date of current version April 14, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2984556

Visual Perception and Modeling for Autonomous Apple Harvesting

HANWEN KANG, HONGYU ZHOU, AND CHAO CHEN
Laboratory of Motion Generation and Analysis, Faculty of Engineering, Monash University, Clayton, VIC 3800, Australia

Corresponding author: Chao Chen (chao.chen@monash.edu)

This work was supported by the Australian Research Council and THOR TECH PTY LTD., under Grant ARC ITRH IH150100006.

ABSTRACT Visual perception and modelling are essential tasks in many vision-dependent robotic tasks. This work presents a robotic vision system to perform fruit recognition, modelling, and environment modelling for autonomous apple harvesting. The fruit recognition applies a deep-learning model Dasnet to perform detection and segmentation on fruits, and segmentation on branches. Fruit modelling localises the centre and computes the grasp pose of each fruit based on Hough Transform. Environment modelling adopts Octrees to represent the occupied space within the working environment of the robot. The robot control computes the path and guide manipulator to pick the fruits based on the computed 3D model of the crop. The developed method is tested in both laboratory and orchard environments. Test results show that fruit recognition and modelling algorithm can accurately localise the fruits and compute the grasp pose in various situations. The Dasnet achieves 0.871 on F_1 score of the fruit detection. Fruit modelling achieves 0.955 and 0.923 on the accuracy of the fruit centre estimation and grasp orientation, respectively. To illustrate the efficiency of the vision system in autonomous harvesting, a robotic harvesting experiment by using industry robotic arm in a controlled environment is conducted. Experimental results show that the proposed visual perception and modelling can efficiently guide the robotic arm to perform the detachment and success rate of harvesting is improved compared to the method which does not compute the grasp pose of fruits.

INDEX TERMS Robotic harvesting, robot vision systems, robotics and automation, computer vision.

I. INTRODUCTION

Robotic harvesting is a promising technology of agriculture in the future. Compared to the autonomous harvesting of the structured crops, visual-guided autonomous harvesting in unstructured orchards is more challenging [1]. Autonomous harvesting Robot requires to detect the fruits, estimate the pose of the fruits, calculate the path for robotic arms to pick the fruits based on surrounding environments. Among these challenges, robotic vision is the key to the success of the harvesting [2]. Since robots cannot perform harvesting if vision systems cannot accurately detect and localise the fruits. Meanwhile, the presented obstacles such as branches could also obstacle the path of robotic arms during the harvesting, which may lead to the fail of the harvesting or even damage the manipulator. Therefore, an efficient robotic vision system is a crucial step towards the development of the fully-functional harvesting robot.

The associate editor coordinating the review of this manuscript and approving it for publication was Okyay Kaynak.

This work developed a robotic visual perception and modelling algorithm for autonomous apple harvesting. The proposed vision system includes a Detection And Segmentation Network (Dasnet) [3], a fruit modelling algorithm, and an environment modelling algorithm. The following highlights are presented in this paper:

- Development of a multi-purpose deep convolution neural network Dasnet, which can perform detection/instance segmentation on fruits, and semantic segmentation on branches by using a one-stage detection network architecture.
- An efficient fruit modelling algorithm and an environment modelling algorithm are presented. Fruit modelling algorithm estimates the geometry properties of the fruits and compute the proper grasp pose for detachment. Environment modelling algorithm applies octrees to represent the obstacles within the working environments and be used in the motion planning of the robotic arm.
- An efficient control framework which guide robot to perform autonomous harvesting is also presented in this work.

The developed visual perception and modelling algorithm was tested in controlled laboratory and orchard environments, showing a robust and efficient performance to applied in the autonomous apple harvesting.

The rest of the paper is organised as follow. Section II reviews the related works. Sections III and IV introduce the design of the robotic system and the visual processing approach, respectively. Experimental methods and results are presented in Section V. In Section VI, conclusion and future works are included.

II. RELATED WORKS

A: FRUIT RECOGNITION

Fruit recognition is the key issue in the vision-guided autonomous fruit harvesting. Robotic vision systems can apply different visual sensors [4], such as RGB imaging sensors, RGB-D cameras, Light Detection And Ranging (LiDAR), thermal imaging sensors, and spectral cameras. This section focus on reviewing the approaches which are developed for fruit recognition by using RGB imaging sensors. Fruit recognition method on RGB images can be divided into two types: traditional machine-learning based algorithms and deep convolution neural network based algorithms. Traditional machine-learning based algorithms use feature descriptors to encode the appearance of the objects, and then use machine-learning algorithms [5] such as Support Vector Machine (SVM), Clustering, and Random forest to perform the classification. There are many feature descriptors such as Colour Coherence Vector (CCV), Histogram of Gradient (HoG), Scale Invariance Feature Transform (SIFT), and Local Binary Pattern (LBP) which have been developed in the previous studies [6]. Traditional machine-learning based algorithms have been widely applied in the fruit recognition [7]. Nguyen *et al.* [8] applied colour features and geometric descriptor to perform detection of apple fruit by using an RGB-D camera. Similarly, Lin *et al.* [9] developed a fruit recognition system by using feature descriptor of HSV colour and geometry to describe the appearance of multiple class of fruits, and an machine-learning based classifier was applied to perform the fruit recognition. Wang and Xu [10] describe the objects by using multiple image feature descriptors, a Latent Dirichlet Allocation (LDA) based classifier is used to perform unsupervised learning and segmentation on plants and fruits in greenhouse environments. Arad *et al.* [11] applied colour and shape features to perform sweet pepper detection and segmentation in the greenhouse with artificial illumination.

The performance of the traditional machine-learning based algorithms is limited due to the limited expressibility of the feature descriptor. Deep convolution neural network based algorithms are developed more recently and show advance and robust performance in the core tasks of machine vision [12]. In the object recognition task, Region Convolution Neural Network (RCNN) [13] based two-stage networks and You Only Look Once (YOLO) [14] or Single Shot Detection (SSD) [15] based one-stage networks are state of the art approaches. RCNN based networks apply two-stage network architecture, which applies a Region Proposal Network (RPN) to search the Region of Interest (ROI) and a classification network branch to perform the classification, bounding box regression and even segmentation [16]. In contrast, one-stage networks apply the fully convolution network architecture, combining the RPN and classification branch into a single network, which largely increases the computation efficiency compared to the two-stage networks. Deep convolution neural network based algorithms have been applied in many vision based agriculture applications [17], such as yield estimation [18], branch pruning [19], and autonomous harvesting [20]. Sa *et al.* [21] applied faster-RCNN to perform detection of multiple class of fruits in orchard and greenhouse environments, and high accuracy on fruit detection was reported in their work. Similarly, Bargoti and Underwood [22] applied faster-RCNN on yield estimation of apples, mango and almonds, a $F1$-score of 0.9 was reported in their work. Yu *et al.* [23] applied the Mask-RCNN to perform instance segmentation in the vision-guided autonomous harvesting of strawberry in greenhouse environments. Tian *et al.* [24] applied an improved YOLO-dense on real-time in the monitoring of apple growth in different stages in orchard environments. Koirala *et al.* [25] applied a light-weight YOLO in mango yield estimation, their system was tested in both day and night (artificial lighting) conditions. Lin *et al.* [26] and Li *et al.* [27] adopted Fully Convolution Network (FCN) [28] to perform semantic segmentation on the guava fruits and cotton, respectively. Kang and Chen [29] developed a customised YOLO-based network for apple detection in the application of autonomous apple harvesting.

B: FRUIT AND ENVIRONMENT MODELLING

Fruit modelling is an important issue in autonomous harvesting. Yaguchi *et al.* [30] applied stereo-camera to obtain RGB and depth image of the working environments. The tomatoes were fitted by using Random Sample Consensus (RANSAC), and robotic arm grasps the tomatoes by translating forward to the centroid of the fruits. Lehnert *et al.* [31], [32] applied Kinect-Fusion algorithm [33] to combine information from multiple frames. The fused three-dimensional point clouds were used to detect the sweet pepper and guide the motion of the robotic manipulator. The model and grasp pose of the Sweet pepper was fitted by using super-ellipsoid or surface normal. Xiong *et al.* [34], [35] applied three-dimensional point clouds to compute the safety operation space of robots to perform robotic harvesting on the strawberry. Lin *et al.* [26] applied a RANSAC-based sphere fitting method [36] to estimate the centre of the guava fruits. An FCN network was used in stem localisation for determining the grasp pose of each fruit. Arad *et al.* [11] developed a method to grasp the sweet pepper by observing the fruits and stems from multiple view-angle. Previous work mainly focuses on computing the centre and grasp pose of the fruits while ignoring the environment

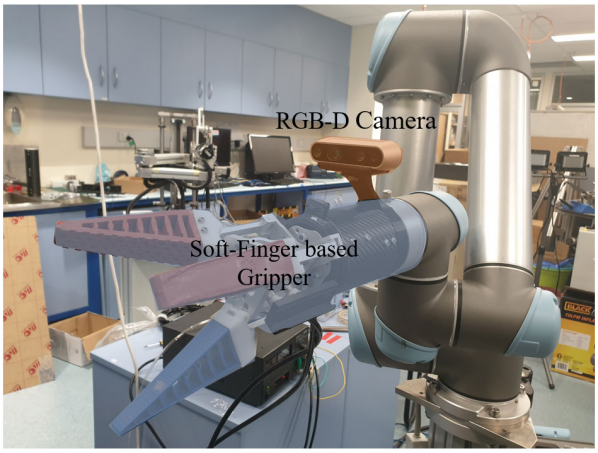


FIGURE 1. The developed prototype of apple harvesting robot.

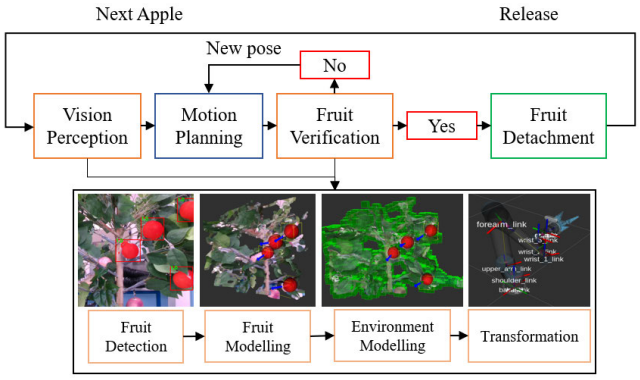


FIGURE 2. The four stages of the autonomous harvesting cycle.

modelling. In the orchard environment, the presented obstacles such as branches can also affect the success rate of the robotic harvesting [4].

Three-dimensional environments modelling is an essential task in many robotic applications. Roth-Tabak and Jain [37] discretised the three-dimensional environments by using the equal-sized grid of cubic volume. The drawback of this method is that such operation requires enormous memory consumption when large outdoor scenes are presented and fine resolution is required. Another commonly used method of modelling the occupied space in three-dimensional environments is point clouds. Cole and Newman [38] applied three-dimensional point clouds in an outdoor Three-dimensional Simultaneous Localisation And Mapping (SLAM) system. Point clouds contain many detail information which cannot be used in the robotic action planning [39]. Other modelling approaches, such as the elevation map [40], [41] and the surface representations [42], [43], require certain assumptions when these methods are applied [44]. Octrees-based representation is also widely used in three-dimensional environment modelling [45]. Octrees is a hierarchical data structure for spatial subdivision in 3D space. Each voxel within an Octrees is recursively subdivided into eight small sub-volumes until the minimum size of the voxel is reached. Octrees-based methods have several advantages compared to the other methods in terms of memory efficiency and adjustable in representation resolution.

III. SYSTEM DESIGN AND OPERATION

A. HARDWARE CONFIGURATION

The robot (see Figure 1) is constructed by a standard industry 6-DoF robotic arm (UR5) made by Universal Robots, A custom-designed end-effector, an Intel Realsense D-435 RGB-D camera, a main computer with high-performance Graphic Processing Unit (GPU) (Nvidia GTX-1080Ti or Jetson-X2), an Arduino-based Programmable Logic Controller (PLC) (Arduino Mega-2560). The output of the RGB-D camera is sent to the main computer and be used continuously to detect and localise the fruits within the working environments. Arduino-based PLC is connected to the main computer to receive the control instructions and controls the end-effector to grasp or release the fruits. The Kinetic version of the Robot Operating System (ROS) [46] on Ubuntu 16.04 LTS operating system is employed for controlling the robot operation.

B. SOFTWARE DESIGN

1) SUBSYSTEM BLOCK

This section outlines the system design for autonomous apple harvesting robot. The robotic system can be broken down into vision processing block, manipulator block, and grasping block. Vision processing block receives colour and depth images continuously from the RGB-D camera through the RealSense-ros communication package. A deep neural network Dasnet and a 3D pose estimation algorithm are adopted to localises and computes the pose of each fruit based on the received image data. Manipulator block controls the UR5 to move to the given pose. The ROS MoveIt! Library [47] with TRAC-IK inverse kinematic solver [48] is used in motion planning for an improved solution. Trajectory execution of robotic arm is performed by using Universal Robots ROS controller package. Grasping block controls the end-effector to grasp and release based on the given order. The connection between Arduino-based PLC and the main computer is performed by using the python port library. The call of each subsystem block follows the designed operation procedures, which is introduced in the following section.

2) OPERATION PROCEDURES

The overall operation procedure of robotic harvesting is shown in Figure 2, which includes four steps below.

1) **Vision Perception:** this function detects apples from the working environments. The 6-DoF pose of each apple and environmental model are computed and sent to the central control for motion planning.
2) **Motion Planning:** this function computes the path which guides the robotic arm to move to the observation pose of each apple.

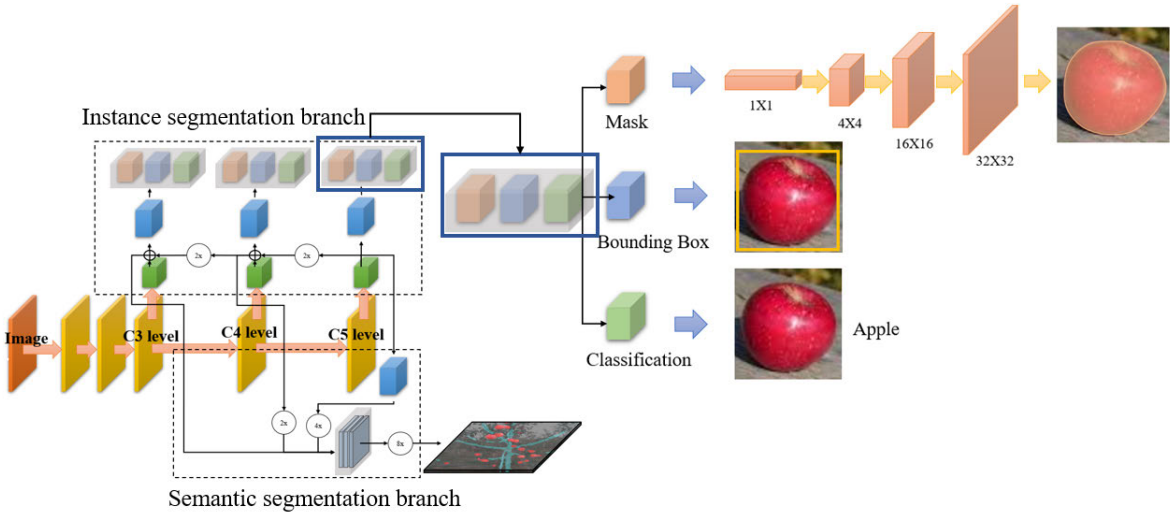


FIGURE 3. Dasnet includes a instance segmentation branch and a semantic segmentation branch for detection and segmentation on apples and branches.

3) **Fruit Verification:** this function verifies whether the chosen apple is within the grasp range or not. If no, the robotic arm will move to the new observation pose based on the newly received fruit pose from the verification step.
4) **Fruit Detachment:** End-effector detaches the apple from the plant.

This work focus on visual perception and modelling (used in steps 1 and 3), which includes the fruit recognition (Section IV-A), fruit pose computation (Section IV-B), and obstacles modelling (Section IV-C). The transformation of detected fruits and obstacles from the camera frame to the robot frame is introduced in Section IV-D. The workflow of the visual perception and modelling is shown in Figure 2.

IV. PERCEPTION AND PLANNING

A. FRUIT RECOGNITION

1) NETWORK ARCHITECTURE

A one-stage detection network Dasnet is applied to process the colour images from the RGB-D camera. The Dasnet follows one-stage detection network architecture to perform fruit detection and instance segmentation. It applies a 50 layers residual network [49] (resnet-50) as the backbone, which generates the feature maps with 1/8, 1/16 and 1/32 size of the original images (as shown in Figure 3). A three levels Feature Pyramid Network (FPN) is applied to fuse the multi-levels features from the C3, C4 and C5 level of the backbone. On each level of the FPN, an instance segmentation branch is applied to perform confidence estimation, bounding box regression, object classification, and mask generation. The branch design of confidence estimation, bounding box regression and object classification follows the design of the YOLO-V3, while the mask generation branch follows the design which is developed in Single Pixel Reconstruction network (SPRNet [50]). Before the instance segmentation branch of each FPN level, an Atrous Spatial Pyramid Pooling (ASPP) [51] is used to fuse the multi-scales features within the feature maps.

Another semantic segmentation branch is embedded on the backbone to perform segmentation on branches. Semantic segmentation branch receives feature maps from the C5, C4 and C3 level from the backbone network. The feature maps of the C5 level and C4 level are four times and two times upsampled and concatenated with the feature maps from the C3 level. The concatenated feature maps of the C5, C4 and C3 are eight times upsampled to match the size of the input image.

2) TRAINING DETAILS

The programming of the Dasnet is performed by using slim image classification library in TensorFlow 1.11. The programming of the resnet-50 is from the GitHub public code [52]. There are 1277 images which are collected from several apple orchards located at the Qingdao (Figure 9 shows the mobile platform used in Qingdao test), China and Melbourne, Australia. The 567 and 150 of these images are used for network training and validation. The other 560 images are used to evaluate the performance of the trained network. Focal-loss [53] and cross-entropy loss are used in the training of confidence estimation and mask generation tasks, respectively. The training of other tasks (bounding box regression and object classification) follows the design described in YOLO-V3. Adam-optimiser in Tensorflow is used as the optimiser during the training with learning rate and decay rate as 0.01 an 0.9/epoch, respectively. The training of the network is performed on Nvidia GTX-1080Ti for 150 epochs.

B. FRUIT POSE COMPUTATION

We apply a rather simple but efficient algorithm in this section to compute 6-DoF of each detected fruit. The 6-DoF pose

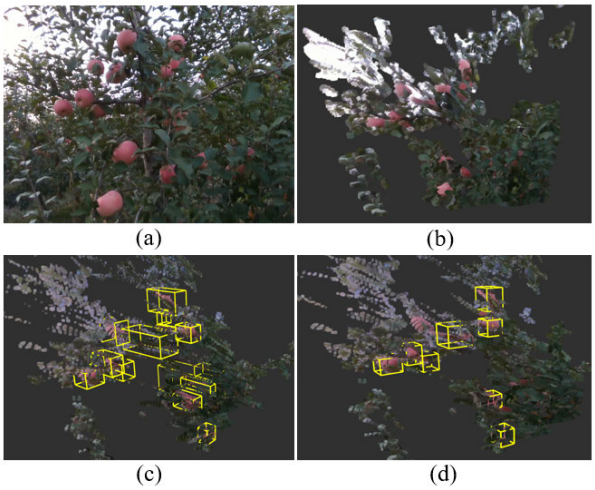


FIGURE 4. (a) RGB image, (b) point clouds, (c) bounding box of fruits before denosing (d) bounding box of fruits after denosing.

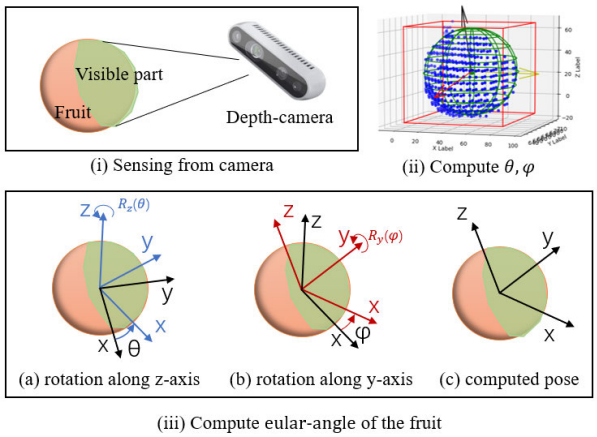


FIGURE 5. Workflow of the fruit pose estimation. The 6-DoF pose of each fruit is computed based on spatial distribution of points. Then ZYX-Eular angle is applied to represent he orientation of the fruit.

estimation includes two-steps: central estimation and orientation computation. The first step applies a 3D Sphere Hough Transform (3D-SHT) to estimate the centre of the fruit. Then, the 6-DoF pose of the given fruit is computed based on the spatial distribution of the point cloud.

1) POINTS DENOISING

In the outdoor environments, the accuracy of RGB-D camera can be affected by many factors, such as lighting, colour and distance. Additional, the depth sensing of fruits can be easily affected by the adjacent objects, such as branch or leaves (as shown in Figure 4). We implement a distance based denoising method on points processing before pose computation. That is, the points of fruit are divided into inlier and outlier based on the Euclidean distance to the point cloud centre. Then, the fruits without the sufficient number of points or have severely in-balance length on X, Y, Z axis is deleted from the detection list. Rather than using all points in the computation which is time-consuming, a spatial down-sampling on point clouds is applied to speed up the running rate.

2) CENTRE ESTIMATION

We assume that apples are in the shape of sphere, which can be expressed as:

$$(x - c_x)^2 + (y - c_y)^2 + (z - c_z)^2 = r^2 \tag{1}$$

c_x, c_y, c_z, and r are the centre position and radius length of the sphere, respectively. The 3D-SHT follows the principle of the circle hough transform, applying vote framework to estimate the geometric properties of the sphere from point clouds. In the first step, we descretise the searching space of the centre position into separate grid based on given resolution, which are denoted as c_x^p, c_y^q, and c_z^n, respectively. Then, we calculate the radius length r_{est} for each point within the point cloud of a fruit in terms of every pair of c_x^p, c_y^q, and c_z^n within the searching range. If r_{est} is in the range of r, one vote is added to the grid c_x^p, c_y^q, c_z^n, and r_{est}. Finally, the grid with the highest number of votes is chosen as the central and radius of the fruit.

3) ORIENTATION COMPUTATION

Based on the spatial distribution of points to the estimated centre of each fruit, the 3-dof orientation is estimated accordingly. Based on computed geometric properties and the spatial distribution of point clouds of each fruit, an orientation estimation algorithm is used in this section to compute the Euler-angle for the robotic arm to access the fruits. We assume that the point clouds P (number of points is n) identified belonging to a fruit is the unblocked and visible partition of a fruit from the current view-angle of the RGB-D camera. Firstly, the point clouds P in Euclidean coordinate can be remapped in the Spherical coordinate through the following equations.

$$\begin{cases} x = R\cos\theta\sin\varphi, \\ y = R\sin\theta\sin\varphi, \\ z = R\cos\varphi. \end{cases} \tag{2}$$

The θ and φ in Eq. (2) within the range of [-π,π] and $[-\frac{1}{2}\pi, \frac{1}{2}\pi]$, respectively. R is the radius of the sphere. We calculate the angle of point p_i which belongs to the point clouds (as shown in Figure 5), to remap these points on the estimated spherical surface. For a given point $\boldsymbol{p_i} = [x_i, y_i, z_i]^T$ with an estimated centre of the sphere $\boldsymbol{c} = [x_c, y_c, z_c]^T$, we have:

$$\boldsymbol{p_c^i} = \boldsymbol{p_i} - \boldsymbol{c} = [x_c^i, y_c^i, z_c^i]^T \tag{3}$$

Then, the angle θ_i can be calculated through the function shown as follow:

$$\theta_i = Atan2(\frac{y_c^i}{R_{xy}^i}, \frac{x_c^i}{R_{xy}^i}), \tag{4}$$

$$R_{xy}^i = ||\boldsymbol{p_c^i}|| \cdot sin\varphi_i = \sqrt{(x_c^i)^2 + (y_c^i)^2} \tag{5}$$

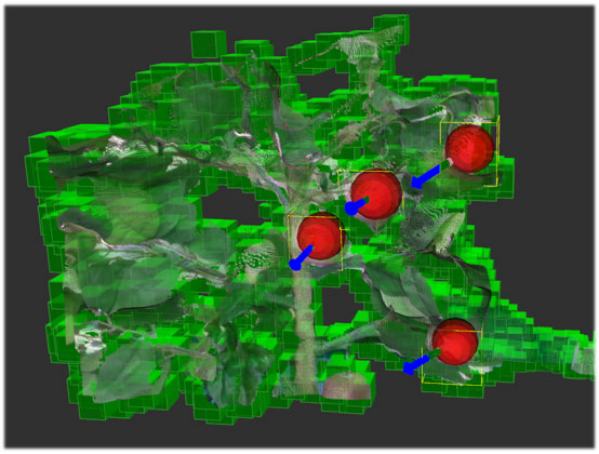


FIGURE 6. Octrees is used in the environment modelling (green boxes), fruits and its pose are represented as the red spheres and blue arrows respectively.

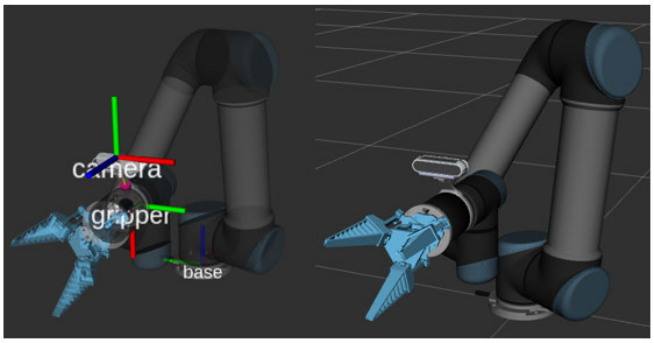


FIGURE 7. Transformations between different links of UR5 are illustrated in this figure.

Similarly, the angle φ_i can be calculated through the function:

$$\varphi_i = Atan2(\frac{z_c^i}{R_{xy}^i}, \frac{R_{xy}^i}{R^i}) \tag{6}$$

Then, based on the computed angle of θ and φ, the point clouds P can be remapped to the point clouds Q which is 2D spherical surface embedded in the 3D Euclidean space, by using Eq. (2). The above step aims to minimise the estimation errors due to the depth sensing. We assume that the point clouds Q is in the 3D-Euclidean space as our experiment results suggest such setup performs well on orientation computation. Therefore, the centroid of the point clouds Q is computed through the following function.

$$C_Q = \frac{1}{n+1}\sum_{i=0}^{n} q_i, q_i \in Q \tag{7}$$

Then, the centroid C_Q is used to compute the angle of θ and φ. We applied ZYX-Eular angle to represent the fruits orientation (as shown in Figure 5). θ and φ are the rotation angle along the Z-axis and Y-axis, respectively. The transformation matrix of the approaching pose of a fruit in the *camera* frame can be formulated as:

$$T_F^C = \begin{bmatrix} \cos\theta\cos\varphi & -\sin\theta & \cos\theta\sin\varphi & x \\ \sin\theta\cos\varphi & \cos\theta & \sin\theta\sin\varphi & y \\ -\sin\varphi & 0 & \cos\varphi & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{8}$$

where F and C are the fruit frame and camera frame, respectively. x, y, z are the fruit centre position in the camera frame. During the operation, we limit the range of the θ and φ from $[-\frac{1}{3}\pi, \frac{1}{3}\pi]$ to secure the robotic arm.

C. ENVIRONMENTS MODELLING

From the previous steps, the semantic segmentation masks of fruits and branches are obtained. Combining the information of the depth map from the RGB-D camera, the point clouds are assigned for each class of objects correspondingly. However, applying point clouds to represent the 3D environments is not an efficient way, since it contains much noisy and detail information which cannot be used in the planning of robotic action [39]. Therefore, octrees-based representation of occupied space in working environments is applied in this section.

Octrees recursively divide a three-dimensional voxel into eight small sub-volumes until the minimum size of the volume is reached. In operation, the resolution of the environment model can be adjusted by given different minimum volume size. Another issue of environment modelling is the error in the range measurement. A typical solution to this problem is applying the probabilistic representation of the occupied space [45], which aims to reduce the measurement error due to the movement of the robot. However, apple harvesting robot only requires one frame in the working environment modelling, which aims to reduce the cycle time of apple harvesting. Therefore, a threshold-controlled binary representation of the voxels is used in the modelling. That is, the voxels with the number of points which is higher than the threshold will be accepted as inlier, while the voxels with the number of points which is lower than the threshold will be rejected as the outlier. During the experiment, we set the minimum size of a voxel as 5 cm and the threshold for voxels filtering as 50. We remove the obstacles which are 10 cm around the fruits, to ensure the success rate on path planning. Harvesting experiments (see Section V-C) show that such setup will not influence the performance of the robot.

D. TRANSFORMATION AND MOTION PLANNING

The transformations between different links of the UR5 robotic arm have been included in the Universal Robots ROS package by using ROS-TF library. The transformation from the frame of the last link (end-effector (ee), which are coincide with gripper frame in Figure 7 and donated as E) to the frame of the first link (base, donated as B) of the UR5 is donated as T_E^B. The transformation from the frame of gripper (G) to the E is denoted as T_G^E. We further applies transformation to link the camera frame C to the gripper frame, which is denoted as T_C^G. In the step of fruit detection,

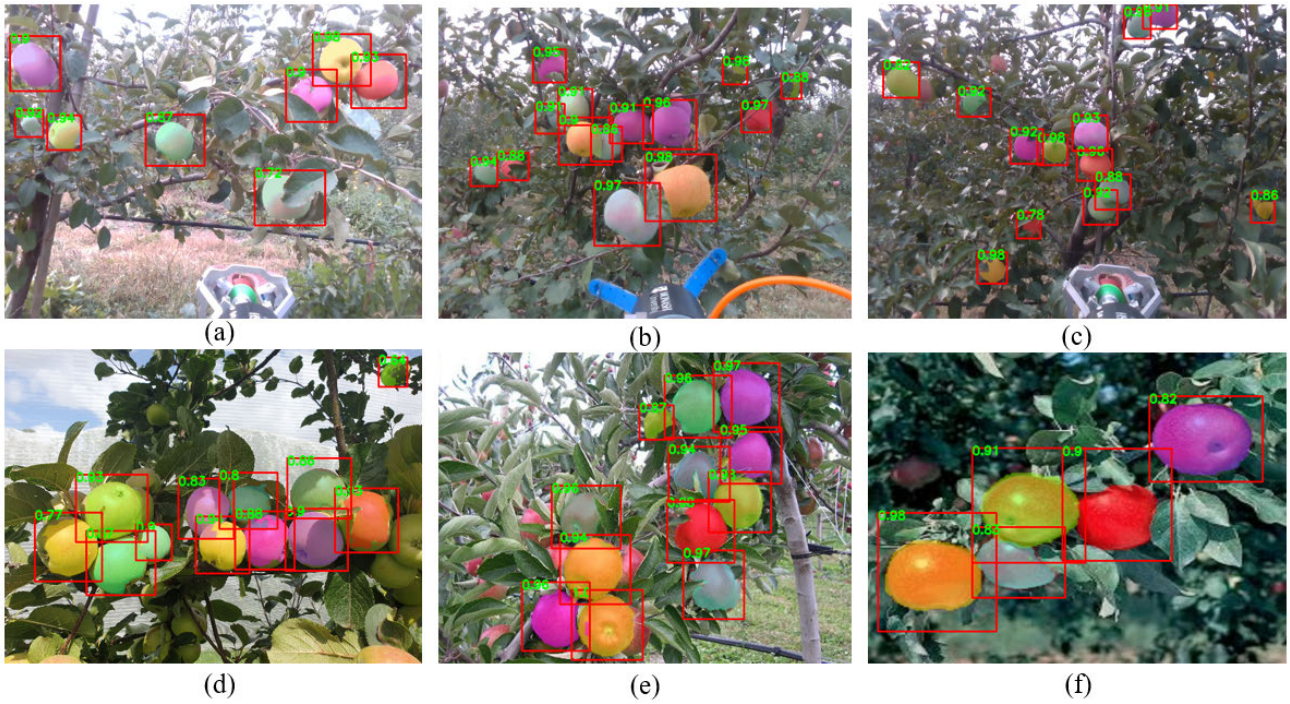


FIGURE 8. Instance segmentation of fruits by using Dasnet. Fruits are drawn in the distinguished colours.

the pose of detected fruits in camera frame (T_F^C) are mapped into the base frame by Eq. (9), as follow.

$$T_F^B = T_E^B T_G^E T_C^G T_F^C \tag{9}$$

We calculate the observation pose by adding an offset in the gripper frame based on the design configurations, to ensure that the camera aims the centre of fruit at the observation pose. The obstacle modelling results are transformed from camera frame to the base frame by using Eq. (9), with the fruit pose replaced with centre of occupied voxels. Axis Aligned Bounding Boxes is used to represent the obstacles within the working environments. Obstacles models are added into the working scene and ROS MoveIt! library [47] with TACKIK inverse kinematic solver [48] is used to compute the path of robotic arm to the given pose. For more complicated cases of motion planning, recent advances in path planning for high DoF robotic arm in complex environments [54], [55] can be further explored. Robotic arm is designed to move to home position to release the fruit after each detachment (as shown in Figure 14) until all the fruits are detached.

V. EXPERIMENT AND DISCUSSION

A. EVALUATION ON FRUIT RECOGNITION

1) EVALUATION METRIC

The evaluation of fruit recognition includes three tasks: fruit detection, segmentation and branches segmentation. In the evaluation of the fruit detection, Intersection of Union (IoU) and F_1-score are applied. In the detection, IoU measures the ratio of the intersection and union between the bounding box of ground-truth and prediction, which is donated as IoU_{BB}. F_1-score measures the detection performance by using the *Recall* and *Precision*. *Recall* measures the fraction of true-positive objects that are successfully detected, while *Precision* measures the fraction of true-positive objects in the predictions. There are four case in the detection, which are True-Positive(TP), False-Positive(FP), True-Negative(TN), and False-Negative(FN). The objects with IoU_{BB} and confidence higher than 0.5 will be identified as the true-positive predictions. The *Precision*, *Recall*, and F_1-score are as follow:

FIGURE 9. Orchard setup and mobile apple harvesting platform.

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{12}$$

The F_1-score of detection is computed by averaging the F_1-score of each image in the test set. In the assessment of the

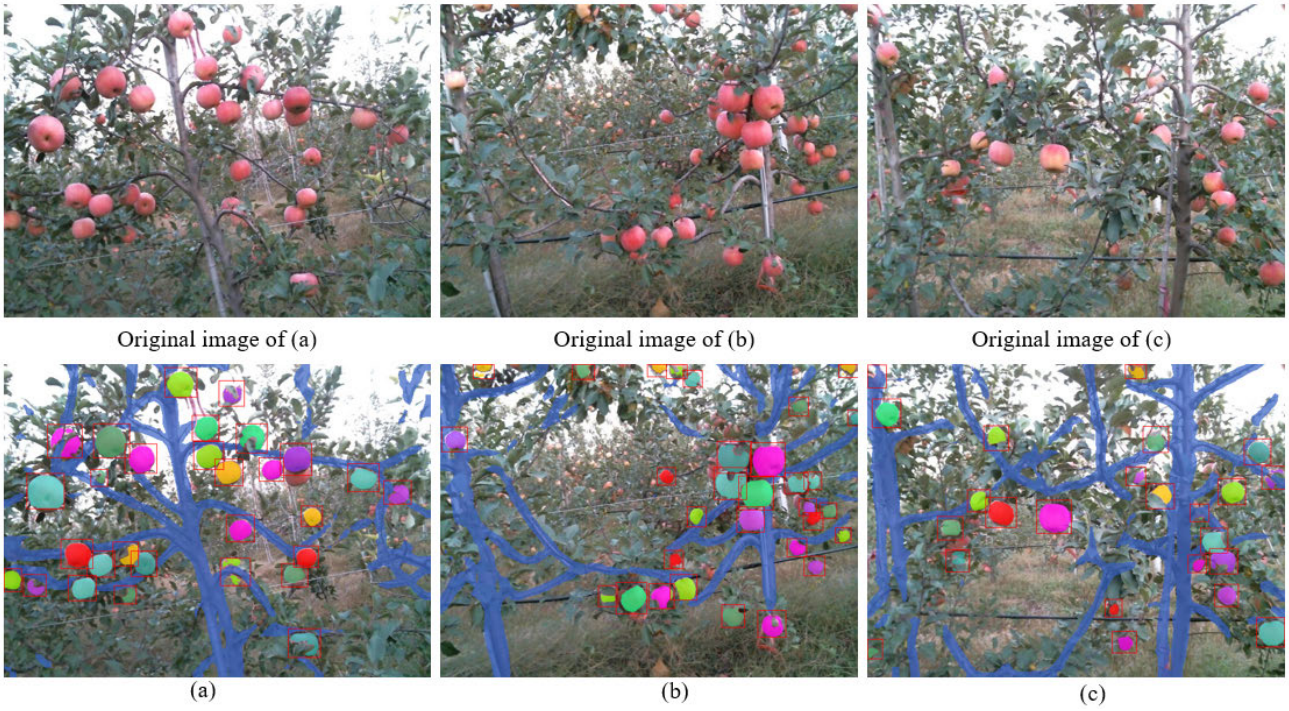


FIGURE 10. Fruits detection and branch segmentation by using the Dasnet. Branches are drawn in the colour of blue.

fruit segmentation and branch segmentation. In this case, IoU measures the ratio of the intersection and union between the ground-truth area and prediction area of the objects. The IoU for evaluation of fruit segmentation and branch segmentation quality are donated as IoU_{fruit} and IoU_{branch}, respectively.

2) DATA VARIETIES

Although our focus is to develop an autonomous apple harvesting system for Fuji apple, generalisability is still an important index in our consideration. To validate the generalisation of the Dasnet on multiple apple varieties, several types of apple such as Fuji (from the orchard located at Qingdao, as shown in Figure 9), pink-lady (from orchards located at Melbourne), and gala (from orchards located at Melbourne) with different colours (green, green-yellow, red-yellow, or pink-red) under different illumination conditions are collected from the different orchards.

3) EXPERIMENTAL ON FRUIT DETECTION

In this experiment, the performance of Dasnet in fruit detection and segmentation was evaluated and compared with the YOLO-V3 and Mask-RCNN. YOLO-V3 applies darknet-53 as backbone. A three-levels FPN design is adopted in the YOLO-V3 to improve detection performance on objects in different scales. The implemented code of YOLO-V3 [56] is from Github shared code. Mask-RCNN is state of the art work in the two-stage detection network. The implemented code of Mask-RCNN [16] applies the multi-level FPN design and resnet-50 is used as the backbone in the model. In the experiment, Dasnet, YOLO-V3, and Mask-RCNN are trained with collected training images. In the assessment, we compared the detection performance among Dasnet, YOLO-V3, and Mask-RCNN and fruit segmentation performance between Dasnet and Mask-RCNN. The experimental results are shown in Table 1. As shown in experimental results, Dasnet, YOLO-V3, and Mask-RCNN achieved similar F_1-score on fruit detection, which were 0.871, 0.864, and 0.87, respectively. Comparably, Dasnet achieved higher recall and precision (0.868 and 0.88) on fruit detection compared to the YOLO-V3 network (0.858 and 0.87). Compared to the Mask-RCNN, Dasnet achieved a similar score on *Precision* and *Recall*. Mask-RCNN achieved a slightly higher accuracy on bounding box localisation (0.866). The IoU_{box} of the Dasnet and YOLO-V3 were 0.861 and 0.847, respectively. In terms of the fruit segmentation, Mask-RCNN performed better than the Dasnet. The IoU_{fruit} achieved by the Mask-RCNN and Dasnet were 0.883 and 0.862, respectively. Table 2 shows the average computational time of the Dasnet, YOLO-V3, and Mask-RCNN on the Nvidia Jetson-TX2. Compared to the Mask-RCNN, Dasnet and YOLO-V3 were more computation efficient. The average computational time of the Mask-RCNN was 0.235 s, 0.477 s, and 1.2 s, respectively.

TABLE 1. Comparison of accuracy of fruit detection and segmentation among different networks.

Model	F_1	Recall	Precision	IoU_{box}	IoU_{fruit}
YOLO-v3	0.864	0.858	0.87	0.847	-
Mask-RCNN	0.87	0.863	0.882	0.866	0.883
Dasnet	0.871	0.868	0.88	0.861	0.862

TABLE 2. Average computational time of different networks on Nvidia Jetson-TX2. the tested image size is 640 × 480 (pixels).

Model	Weight Size	Time
YOLO-v3 (darknet-53)	246 M	235 ms
Dasnet (resnet-50)	187 M	477 ms
Mask-RCNN (resnet-50)	244 M	1.2 s

TABLE 3. Comparison of performance on semantic segmentation of branches among different models.

Model (Backbone)	IoU_{branch}
FCN-8s (Resnet-101)	0.757
Dasnet (Resnet-50)	0.794

Examples of fruit detection and segmentation by using Dasnet on collected orchard images are shown in Figure 8.

4) EXPERIMENTAL ON BRANCH SEGMENTATION

Branch segmentation is another important task of Dasnet. This experiment compares the accuracy of branch segmentation with FCN-8s network. The implemented code of FCN-8s [57] with resnet-101 is from Github shared code. The segmentation results are shown in Table 3. Dasnet applies ASPP to enhance feature extraction on feature maps from the C5 level, which increases the accuracy of the branch segmentation. The IoU_{branch} achieved by FCN-8s and Dasnet were 0.757 and 0.794, respectively. However, when the camera working in a close distance (for example, < 0.5 m) from the trees, many small branches are presented and affect the accuracy of the branch segmentation. Therefore, the branch segmentation is applied when the working distance between 0.8 m-1.5 m (experimental results suggest branch segmentation can work robustly at this distance). Examples of branch segmentation are shown in Figure 10

B. EVALUATION ON MODELLING ALGORITHM

1) EVALUATION METHODS

This experiment evaluates the accuracy of the fruits pose computation by measuring the accuracy within the working space of the harvesting robot. The ground-truth of the fruit centre and orientation are measured manually. In the accuracy assessment of modelling algorithm, the *True* and *False* is determined by comparing the computed value and ground-truth. In the evaluation of the centre estimation accuracy, we set the threshold as 5cm, since the den-effector is capable of detaching the fruits when centre error is less than 5 cm. That is, when the Euclidean distance between the estimated centre and ground-truth is smaller than 5 cm, we consider this estimation is *True*. The threshold to determine the accuracy of pose computation is 10° (Euclidean distance). The experiments are taken in two scenarios. One is in the controlled laboratory environment, and another is on the RGB-D images collected from orchards. In the laboratory experiment, the computation of each object is repeated ten times. If the ratio of the *True* in total number is larger than 0.8, this sample is considered as a true computed case. Additional, Standard Deviation (SD) is used to measure the fluctuating of the computed centre and pose.

TABLE 4. Experimental results on fruit centre estimation.

Distance (m)	0.3-0.5	0.5-0.7	> 0.7
Accuracy	0.955	0.907	0.82
SD(centre) (cm)	1.6	3.2	5.7
SD(radius) (cm)	1.2	2.2	4.3

TABLE 5. Experimental results on pose estimation of fruit modelling.

Distance (m)	0.3-0.5	0.5-0.7	> 0.7
Accuracy	0.923	0.885	0.775
SD(θ) (°)	5.2	8.6	13.7
SD(φ) (°)	4.9	7.6	12.5

2) EXPERIMENT IN LABORATORY SCENARIO

The operating range of the vision system is from 0.3m to 0.7m along the X-axis of the robot coordinate (the minimum sensing distance of the Intel RealSense D-435 is 0.2-0.3 m). We extend the maximum range of visual system up to 1.2m in the experiment. The operating distance is divided into four subsections, which are 0.3-0.5 m, 0.5-0.7 m, and above 0.7 m. In the most conditions of the robotic harvesting, the distance between the RGB-D camera to the objects is in the section 0.3-0.7 m. Table 4 shows the results on centre estimation and Table 5 shows the results on fruit pose computation.

As shown in Tables 4 and 5, fruit modelling algorithm performed accurately within the range of 0.3 m to 0.7 m. From the experimental results, the accuracy of the fruit modelling algorithm dropped off with the increase of the distance from the RGB-D camera to the fruits. The accuracy of the centre estimation in 0.3 m to 0.5 m and 0.5 m to 0.7 m were 0.955 and 0.907, respectively. The accuracy of the pose computation in 0.3 m to 0.5 m and 0.5 m to 0.7 m were 0.923 and 0.885, respectively. Decreased accuracy was due to the decreasing numbers of the point in the point clouds of fruit. Insufficient number of points may cause larger fluctuating on the computation, which is shown in the SD of centre estimation and pose computation (as shown in Tables 4 and 5). To ensure the success of fruit detachment, fruit verification step will measure the fruit centre and pose in a close distance (0.4 m).

Another important issue of fruit modelling is to compute the centre accurately and pose for adjacent or overlapped fruits. As shown in Figure 8, Dasnet can accurately detect and segment the fruit under the adjacent and overlapping conditions. We evaluated the performance of fruit modelling algorithm on the adjacent and overlapped fruits. The modelling results are shown in Figure 11. From the results, it can be seen that fruit modelling can generate a good fit and good estimation of the object pose in the adjacent or overlapped condition.

3) EXPERIMENT IN ORCHARD SCENARIO

We also conducted experiment on collected RGB-D images. The applied 125 RGB-D images were collected from the Fuji

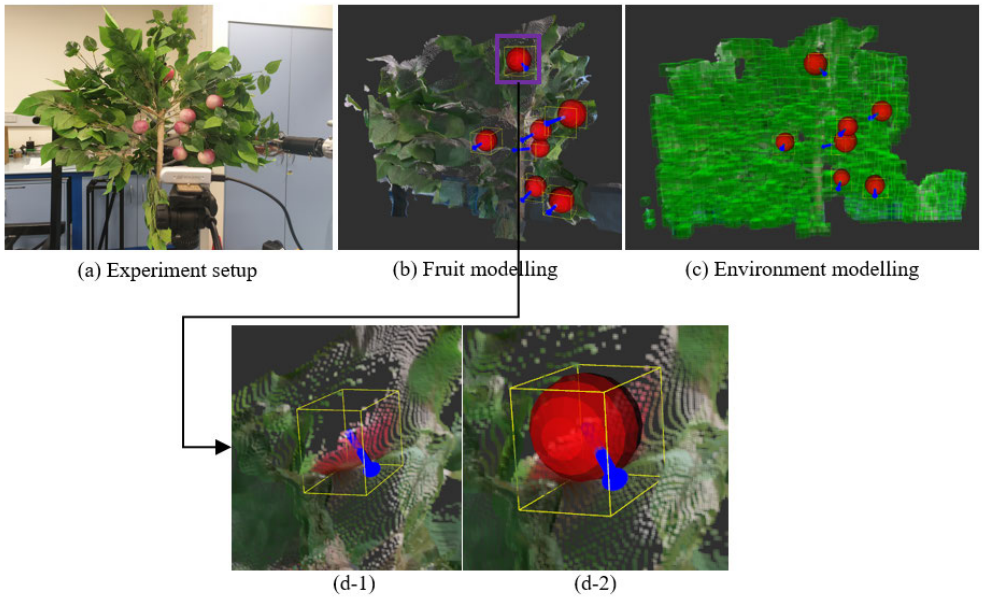


FIGURE 11. Fruit and environment modelling in the laboratory environment. (d-1) and (d-2) show the details of the partial blocked fruit modelling.

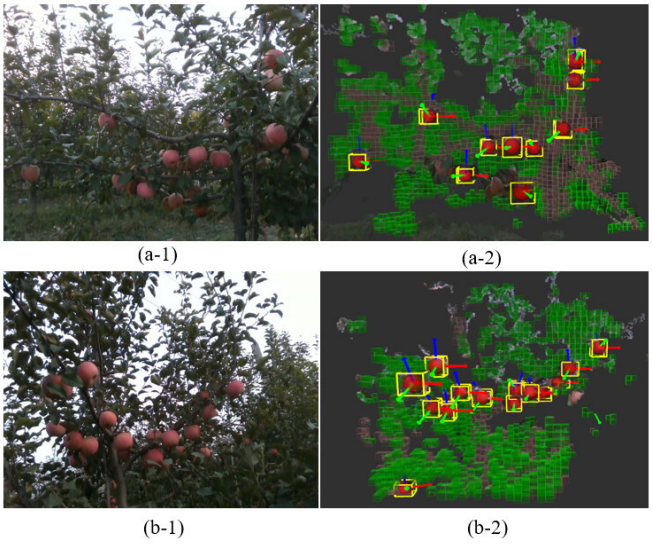


FIGURE 12. Fruit and environment modelling on orchard RGB-D images. Fruits and grasp poses are represented as red sphere and blue arrows. Green and gray boxes stand the occupied space of leaves and branches.

TABLE 6. Experimental results of fruit modelling on orchard images.

Distance (m)	0.3-0.5	0.5-0.7	> 0.7
Accuracy on Centre	0.923	0.85	0.72
Accuracy on Pose	0.875	0.793	0.67

FIGURE 13. Harvesting experiment setup in controlled environment.

apple orchard located at Qingdao, China. During the data collection, the distance from the camera to the fruits was between 0.3-1.0 meters. Fruit modelling is more challenging than performing in the controlled environments. The experimental results of fruit modelling on orchard data are shown in Figure 12 and Table 6.

Compared to the fruit modelling in the laboratory environment, the accuracy of the fruit modelling algorithm in the orchard environment was decreased. The major reason for the lower accuracy of fruit modelling in orchard environment was due to the lower quality of the point clouds. Experimental results showed that data collected from orchard contained more noise than the data collected from the lab. However, in the section of 0.3-0.5 m, the accuracy of the fruit modelling on centre estimation and pose computation were 0.923 and 0.875, respectively. Since the robot will perform the verification step before each detachment, high accuracy of fruit modelling in the section of 0.3-0.5 m can secure the performance of the success rate of the harvesting.

C. EXPERIMENT ON AUTONOMOUS HARVESTING

This section examines the effectiveness of visual perception and modelling by attempting picking apples in the controlled experiment. The experimental setup is illustrated

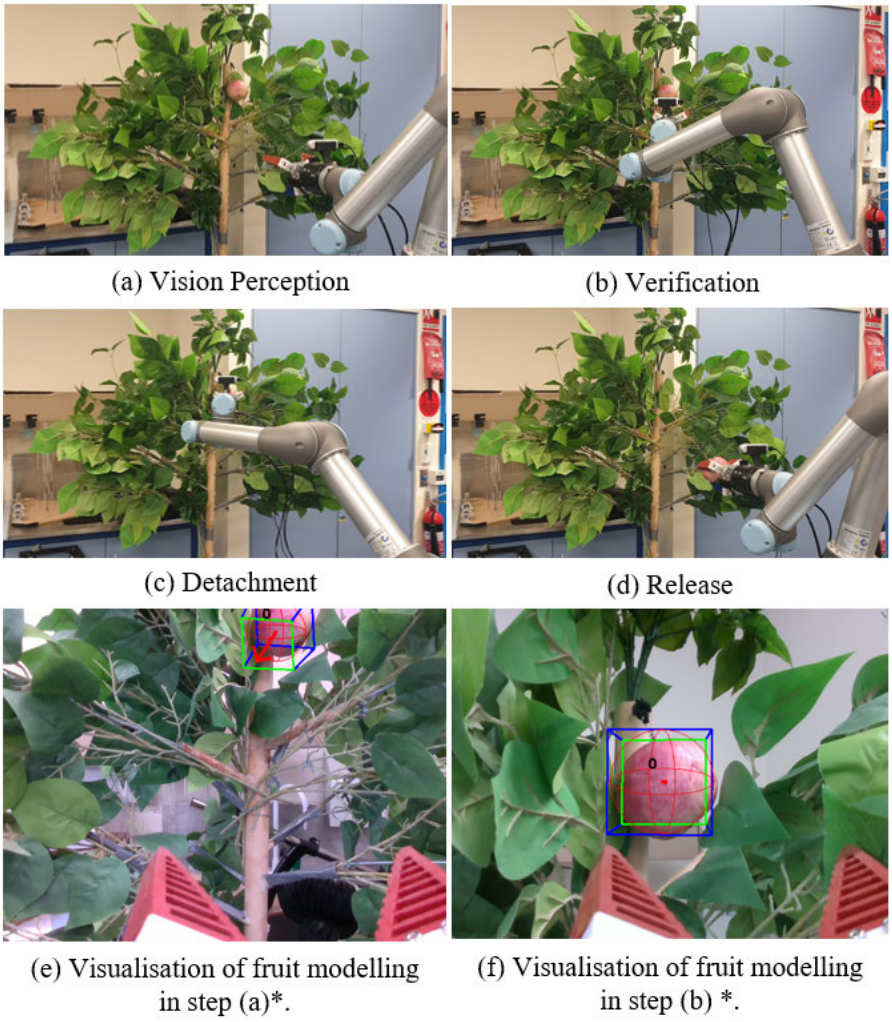


FIGURE 14. Illustration of workflow of the robotic harvesting.

in Figure 13. We simulated the real harvesting environments by locating five to eight apples on the plant in the laboratory environment. The harvesting robot followed the designed harvesting method to generate the 3D model and implement the grasping. Another naive harvesting method was implemented, which harvest apple by translating the robotic arm towards the centroid of the fruits without considering the pose. In total 50 trails with 265 apples were attempted to be harvested with each method. Out of the total apples, 132/265 of apples were adjacent, overlapped, or partially blocked by branches or leaves (type-B), other apples are separated (type-A).

As shown in Table 7, fruit recognition performed well on both types. Fruit modelling achieved high accuracy on type A but a little lower accuracy on type-B. The common issue in type-B fail was centre estimation fail. When fruit is partially blocked by branches or other fruits, the number of points for centre estimation was insufficient. In terms of the harvesting performance, our harvesting robot performed well in the harvesting of both type-A and type-B while the success rate of the naive method on type B harvesting was decreased significantly. The common failure of the naive method in the type-B harvesting was the occlusion of fruit by branches. Our method can guide the robotic arm to grasp the fruits in a proper pose, which increases the success rate of harvesting on type B apple from 0.62 (naive method) to 0.81 (our method). Another common failure of both methods was that the robot could not recognise the apples when fruits

TABLE 7. Harvesting result.

Method	Type-A	Type-B
Recognition rate	1.0	0.92
Modelling rate	0.97	0.88
Harvesting rate (Our method)	0.91	0.81
Harvesting rate (Naive method)	0.90	0.62

are severely blocked by the leaves and branches. This issue can be improved by designing the optimal path for the robot to observing the plants before harvesting. Experiments also suggested that grasp failure is another common reason to fail to harvest. It was showed that fruits might be slipped from the end-effector when fruits are partially covered by leaves. We believe that this issue can be overcome by investigating more appropriate end-effector designs. The average picking time of our method and the naive method was 7 seconds and 12 seconds, respectively. Our method requires more time to process the visual modelling, which cost approximately 3 seconds for modelling of each frame. Meanwhile, since more complex motion is introduced by targeting apples with different poses, more time is consumed in the motion planning and robotic motion.

VI. CONCLUSION

This work developed a fruit recognition and modelling algorithm and an environment modelling algorithm for autonomous apple harvesting robot. A deep neural network Dasnet was applied to perform fruit recognition. Fruit modelling was based on 3D-SHT and spatial distribution of the point clouds of the apples. Environment modelling algorithm adopted Octrees to represent the obstacle within the working environments. The developed vision method was tested in both laboratory and orchard environments. The tested results showed that Dasnet could accurately detect and segment the fruits in the orchards, the F_1-score and IoU which stands for accuracy of fruit detection and segmentation were 0.871 and 0.862, respectively. Fruit modelling also illustrated accurate performance in both environments. Moreover, we conducted a robot harvesting experiment in a controlled environment. Experimental results showed the successful performance in recognition and harvesting apples in various situations.

In future works, the fruit modelling algorithm can be future improved and adapted to be applied in more complex conditions. Another aspect of further work is to include more functions into fruit recognition, such as ripeness and damage detection.

ACKNOWLEDGEMENT

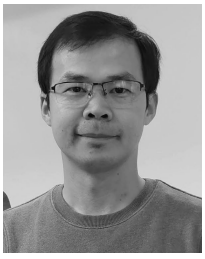
The authors would like to acknowledge Z. Chen for her assistance in preparation of this article.

REFERENCES

[1] C. W. Bac, E. J. van Henten, J. Hemming, and Y. Edan, "Harvesting robots for high-value crops: State-of-the-art review and challenges ahead," *J. Field Robot.*, vol. 31, no. 6, pp. 888–911, Nov. 2014.
[2] A. Vibhute and S. K. Bodhe, "Applications of image processing in agriculture: A survey," *Int. J. Comput. Appl.*, vol. 52, no. 2, pp. 34–40, 2012.
[3] H. Kang and C. Chen, "Fruit detection, segmentation and 3D visualisation of environments in apple orchards," *Comput. Electron. Agricult.*, vol. 171, Apr. 2020, Art. no. 105302.
[4] Y. Zhao, L. Gong, Y. Huang, and C. Liu, "A review of key techniques of vision-based control for harvesting robot," *Comput. Electron. Agricult.*, vol. 127, pp. 311–323, Sep. 2016.
[5] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," *Emerg. Artif. Intell. Appl. Comput. Eng.*, vol. 160, pp. 3–24, 2007.
[6] R. M. Kumar and K. Sreekumar, "A survey on image feature descriptors," *Int. J. Comput. Sci. Inf. Technol.*, vol. 5, pp. 7668–7673, May 2014.
[7] K. Kapach, E. Barnea, R. Mairon, Y. Edan, and O. B. Shahar, "Computer vision for fruit harvesting robots—State of the art and challenges ahead," *Int. J. Comput. Vis. Robot.*, vol. 3, no. 1/2, pp. 4–34, 2012.
[8] T. T. Nguyen, K. Vandevoorde, N. Wouters, E. Kayacan, J. G. De Baerdemaeker, and W. Saeys, "Detection of red and bicoloured apples on tree with an RGB-D camera," *Biosyst. Eng.*, vol. 146, pp. 33–44, Jun. 2016.
[9] G. Lin, Y. Tang, X. Zou, J. Xiong, and Y. Fang, "Color-, depth-, and shape-based 3D fruit detection," *Precis. Agricult.*, vol. 21, no. 1, pp. 1–17, Feb. 2020.
[10] Y. Wang and L. Xu, "Unsupervised segmentation of greenhouse plant images based on modified latent Dirichlet allocation," *PeerJ*, vol. 6, Jun. 2018, Art. no. e5036.
[11] B. Arad, J. Balendonck, R. Barth, O. Ben-Shahar, Y. Edan, T. Hellström, J. Hemming, P. Kurtser, O. Ringdahl, T. Tielen, and B. Tuijl, "Development of a sweet pepper harvesting robot," *J. Field Robot.*, Jan. 2020.
[12] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: A review," *Neurocomputing*, vol. 187, pp. 27–48, Apr. 2016.
[13] 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, Jun. 2017.
[14] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767
[15] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot MultiBox detector," in *Proc. ECCV*, 2016, pp. 21–37.
[16] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2961–2969.
[17] A. Kamilaris and F. X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Comput. Electron. Agricult.*, vol. 147, pp. 70–90, Apr. 2018.
[18] K. Kuwata and R. Shibasaki, "Estimating crop yields with deep learning and remotely sensed data," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2015, pp. 858–861.
[19] Y. Majeed, J. Zhang, X. Zhang, L. Fu, M. Karkee, Q. Zhang, and M. D. Whiting, "Apple tree trunk and branch segmentation for automatic trellis training using convolutional neural network based semantic segmentation," *IFAC-PapersOnLine*, vol. 51, no. 17, pp. 75–80, 2018.
[20] Kang and Chen, "Fruit detection and segmentation for apple harvesting using visual sensor in orchards," *Sensors*, vol. 19, no. 20, p. 4599, 2019.
[21] I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez, and C. McCool, "DeepFruits: A fruit detection system using deep neural networks," *Sensors*, vol. 16, no. 8, p. 1222, 2016.
[22] S. Bargoti and J. Underwood, "Deep fruit detection in orchards," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2017, pp. 3626–3633.
[23] Y. Yu, K. Zhang, L. Yang, and D. Zhang, "Fruit detection for strawberry harvesting robot in non-structural environment based on mask-RCNN," *Comput. Electron. Agricult.*, vol. 163, Aug. 2019, Art. no. 104846.
[24] Y. Tian, G. Yang, Z. Wang, H. Wang, E. Li, and Z. Liang, "Apple detection during different growth stages in orchards using the improved YOLO-V3 model," *Comput. Electron. Agricult.*, vol. 157, pp. 417–426, Feb. 2019.
[25] A. Koirala, K. B. Walsh, Z. Wang, and C. McCarthy, "Deep learning for real-time fruit detection and orchard fruit load estimation: Benchmarking of—MangoYOLO,"' *Precis. Agricult.*, vol. 20, no. 6, pp. 1107–1135, Dec. 2019.
[26] G. Lin, Y. Tang, X. Zou, J. Xiong, and J. Li, "Guava detection and pose estimation using a low-cost RGB-D sensor in the field," *Sensors*, vol. 19, no. 2, p. 428, 2019.
[27] Y. Li, Z. Cao, Y. Xiao, and A. B. Cremers, "DeepCotton: In-field cotton segmentation using deep fully convolutional network," *J. Electron. Imag.*, vol. 26, no. 5, Oct. 2017, Art. no. 053028.
[28] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.
[29] H. Kang and C. Chen, "Fast implementation of real-time fruit detection in apple orchards using deep learning," *Comput. Electron. Agricult.*, vol. 168, Jan. 2020, Art. no. 105108.
[30] H. Yaguchi, K. Nagahama, T. Hasegawa, and M. Inaba, "Development of an autonomous tomato harvesting robot with rotational plucking gripper," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2016, pp. 652–657.

[31] C. Lehnert, I. Sa, C. McCool, B. Upcroft, and T. Perez, "Sweet pepper pose detection and grasping for automated crop harvesting," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2016, pp. 2428–2434.
[32] C. Lehnert, A. English, C. McCool, A. W. Tow, and T. Perez, "Autonomous sweet pepper harvesting for protected cropping systems," *IEEE Robot. Autom. Lett.*, vol. 2, no. 2, pp. 872–879, Apr. 2017.
[33] R. A. Newcombe, A. Fitzgibbon, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, and S. Hodges, "KinectFusion: Real-time dense surface mapping and tracking," in *Proc. 10th IEEE Int. Symp. Mixed Augmented Reality*, Oct. 2011, pp. 127–136.
[34] Y. Xiong, Y. Ge, L. Grimstad, and P. J. From, "An autonomous strawberry–harvesting robot: Design, development, integration, and field evaluation," *J. Field Robot.*, vol. 37, no. 2, pp. 202–224, Mar. 2020.
[35] Y. Ge, Y. Xiong, G. L. Tenorio, and P. J. From, "Fruit localization and environment perception for strawberry harvesting robots," *IEEE Access*, vol. 7, pp. 147642–147652, 2019.
[36] R. Schnabel, R. Wahl, and R. Klein, "Efficient RANSAC for point-cloud shape detection," *Comput. Graph. Forum*, vol. 26, no. 2, pp. 214–226, Jun. 2007.
[37] Y. Roth-Tabak and R. Jain, "Building an environment model using depth information," *Computer*, vol. 22, no. 6, pp. 85–90, Jun. 1989.
[38] D. M. Cole and P. M. Newman, "Using laser range data for 3D SLAM in outdoor environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2006, pp. 1556–1563.
[39] L. Zhang, L. Wei, P. Shen, W. Wei, G. Zhu, and J. Song, "Semantic SLAM based on object detection and improved octomap," *IEEE Access*, vol. 6, pp. 75545–75559, 2018.
[40] R. Hadsell, J. A. Bagnell, D. F. Huber, and M. Hebert, "Accurate rough Terrain estimation with space-carving kernels.," *Robot., Sci. Syst.*, vol. 2009, pp. 1–8, Jun. 2009.
[41] B. Douillard, J. Underwood, N. Melkumyan, S. Singh, S. Vasudevan, C. Brunner, and A. Quadros, "Hybrid elevation maps: 3D surface models for segmentation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 1532–1538.
[42] M. Magnusson, A. Lilienthal, and T. Duckett, "Scan registration for autonomous mining vehicles using 3D-NDT," *J. Field Robot.*, vol. 24, no. 10, pp. 803–827, Oct. 2007.
[43] M. Habbecke and L. Kobbelt, "A surface-growing approach to multi-view stereo reconstruction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.
[44] M. Beul, D. Droeschel, M. Nieuwenhuisen, J. Quenzel, S. Houben, and S. Behnke, "Fast autonomous flight in warehouses for inventory applications," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3121–3128, Oct. 2018.
[45] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Auto. Robots*, vol. 34, no. 3, pp. 189–206, Apr. 2013.
[46] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: An open-source robot operating system," in *Proc. ICRA Workshop Open Source Softw.*, Kobe, Japan, vol. 3, 2009, p. 5.
[47] I. A. Sucan and S. Chitta. (2016). *Moveit*. [Online]. Available: http://moveit.ros.org
[48] P. Beeson and B. Ames, "TRAC-IK: An open-source library for improved solving of generic inverse kinematics," in *Proc. IEEE-RAS 15th Int. Conf. Humanoid Robots (Humanoids)*, Nov. 2015, pp. 928–935.
[49] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[50] J. Yu, J. Yao, J. Zhang, Z. Yu, and D. Tao, "Single pixel reconstruction for one-stage instance segmentation," 2019, *arXiv:1904.07426*. [Online]. Available: http://arxiv.org/abs/1904.07426
[51] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.
[52] R. Taylor, Y. Hongkun, and N. Wu. (Nov. 2018). *Tensorflow-Resnet*. [Online]. Available: https://github.com/tensorflow/models/tree/master/research/resnet
[53] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 318–327, Feb. 2020.
[54] X. Zhang and J. Liu, "Effective motion planning strategy for space robot capturing targets under consideration of the berth position," *Acta Astronautica*, vol. 148, pp. 403–416, Jul. 2018.
[55] X. Zhang, J. Liu, J. Feng, Y. Liu, and Z. Ju, "Effective capture of nongraspable objects for space robots using geometric cage pairs," *IEEE/ASME Trans. Mechatronics*, vol. 25, no. 1, pp. 95–107, Feb. 2020.
[56] P. Kapica. (Jul. 2019). *tensorflow-Yolov3*. [Online]. Available: https://github.com/mystic123/tensorflow-yolo-v3
[57] D. Pakhomov, V. Premachandran, M. Allan, M. Azizian, and N. Navab, "Deep residual learning for instrument segmentation in robotic surgery," 2017, *arXiv:1703.08580*. [Online]. Available: http://arxiv.org/abs/1703.08580

HANWEN KANG received the B.Eng. degree in engineering from Monash University, Melbourne, Australia, in 2017, and the degree in transportation with Central South University, Changsha, China, in 2017. He is currently pursuing the Ph.D. degree with Monash University. Since 2017, he has been a Research Assistant with the Laboratory of Motion Generation and Analysis, Monash University. His research interests include automation, computer vision, deep learning, SLAM, and image processing.



HONGYU ZHOU received the B.Eng. degree from the Harbin Institute of Technology and the M.Eng. degree from Tongji University, China. He is currently pursuing the Ph.D. degree with Monash University, Melbourne, Australia. He has been a Research Assistant with the Laboratory of Motion Generation and Analysis, Monash University, since 2018. His research interests include agricultural robot, adaptive grasping, and tactile sensing.

CHAO CHEN received the B.Eng. degree of mechanical engineering from Shanghai Jiao Tong University, Shanghai, in 1996, and the M.Eng. and Ph.D. degrees in mechanical engineering from McGill University, Montreal, in 2002 and 2006, respectively. From 2006 to 2007, he was a Postdoctoral Fellow with the University of Toronto. From 2007 to 2010, he was a Lecturer with the Department of Mechanical and Aerospace Engineering, Monash University, where he has been a Senior Lecturer, since 2011. He was a Visiting Professor with IRCCyN, Ecole Central de Nantes. He is also an Adjunct Associate Professor with The Chinese University of Hong Kong. His research interests include robotic design and control, theory of mechanisms, robotic exoskeleton, robotic surgery, agricultural robots, and humanoid robots. He has received several awards, including the 1996 Dean's Honor List at Shanghai Jiao Tong University, the 2004 ASME International Scholarship, the 2005 FQRNT Doctorate Fellowship, the 2006–2007 FQRNT Post-Doctorate Fellowship, and the 2017 Year's Innovation Award by Australia Hand Therapy Association.

7 Conclusions and Future Works

7.1 Conclusions

A high-performance machine vision system is the essential requirement for functionable harvesting robots. Improvements in image analysis algorithms allow new advancing in techniques in a wide range of agricultural applications. In this research, we investigate deep-learning based algorithms for the use of machine vision of harvesting robots. A machine vision system which includes a multi-function deep-learning based neural network and an environmental modelling algorithm were developed. The overall developments of the machine vision system were included in four chapters.

Chapter 3 developed a fast network training framework for a customised YOLO-based fruit detector. The developed framework includes a traditional machine-learning based algorithm for orchard data labelling and a YOLO-based fruit detection network LedNet. Image labelling algorithm applies a clustering algorithm to perform pixel-level segmentation on multi-scale pyramid of resized images. LedNet adopts feature pyramid network, ASPP, and a light-weight Resnet to improve the performance and computational efficiency. From the experiments, LedNet achieved 0.821 and 0.853 on recall and accuracy of apple detection in orchard environments. The weights size and average computational time of the LedNet on a 640*480 image are 7.4M and 28ms, respectively.

In Chapter 4, a multi-functional network DaSNet-v1, which can perform real-time detection of fruits and semantic segmentation of branches in orchards environments, was developed. DaSNet-v1 adds a semantic segmentation branch on LedNet, which allows the network to perform multi-tasks on a one-stage detection network. DaSNet-v1 applies the ASPP and the Gated-FPN to enhance feature extraction between different layers and scales. Meanwhile, DaSNet-v1 also applies a light-weight Resnet to ensure the computational efficiency of the model. From the experimental results, DaSNet-v1 achieved an F_1 score of 0.827 on the fruit detection, and 86.5%, 75.7% on the semantic segmentation of fruits and branches, respectively. The weights size and average computational time of the network model are 12.8 M and 32ms (on Nvidia GTX-1070), respectively.

Chapter 5 follows improvements made in Chapter 4, a improved multi-function network model DaSNet-v2 was developed. Instance segmentation on each fruit is important as it can provide abundant information on each object, especially under occlusion and overlapping conditions. Based on the previ-

ously developed DaSNet-v1, DaSNet-v2 applies the instance segmentation design from the SPRNet, allows the model to perform detection, instance segmentation on fruits, and semantic segmentation on workspace with a one-stage detection network. From the experimental results, DaSNet-v2 with Resnet-101 achieves 0.868, 0.88 and 0.873 on recall, the precision of detection, and instance segmentation accuracy on fruits, and 0.794 on semantic segmentation accuracy of branches, respectively. DaSNet-v2 with light-weight Resnet design achieves 0.85, 0.87, and 0.866 on recall and precision of detection, and accuracy of instance segmentation on fruits, and 0.775 on the segmentation accuracy of branches, respectively. The average running time and weight size of light DaSNet-v2 are 55ms and 8.1M, respectively.

Chapter 6 combined progress from the previous three chapters and developed a robotic vision system to perform fruit recognition and modelling, workspace modelling, and high-level control framework for an apple harvesting robot. The framework applies DaSNet-v2 to perform detection and instance segmentation on fruits, and semantic segmentation on the workspace. Fruit modelling localises the centre and computes the grasp pose of each fruit based on Hough Transform. Workspace modelling uses Octrees to represent the occupied space within the working environment of the robot. The robot control computes the path and guides the manipulator to pick the fruits based on the computed 3D model of the crop. From the experimental results, DaSNet-v2 achieves 0.871 on F_1 score of the fruit detection. Fruit modelling achieves 0.955 and 0.923 on the accuracy of the fruit centre estimation and grasp orientation, respectively. Robotic harvesting experiments shows that our developed vision system can efficiently guide the robotic harvesting in a simulated environment. The success rate and average picking time of robotic system are 0.91 and 12s. respectively. In future work, we aims to validate and improve the performance of robotic system in real orchard environment.

7.2 Future Works

7.2.1 Advancements in Machine Vision

The current deep-learning based algorithms are still limited in several aspects, including data labelling computational efficiency and limited functions. Deep-learning methods require a large number of labelled data, which is time-consuming and labour-intensive. Future work can focus on semi-supervised learning methods, which can train the model from a limited number of ground-truth. Secondly, the computational efficiency of deep-learning based methods also limits the use of the model in many

applications. Meanwhile, deep-learning methods which can directly process 2D and 3D sensory data, allowing more functions of models. For example, 3D CNN allows deep-learning model to estimate approaching pose of each fruit within workspace by combining both 2D RGB images and 3D point clouds. Combining the aforementioned points, deep-learning based visual processing algorithms can further optimising the performance of machine vision of harvesting robots.

Workspace modelling is another important task in a machine vision system for harvesting robots. Robotic grasping in unstructured orchard environments is challenging as obstacles or other objects within the workspace can lower the success rate of operation. A well-defined 3D map of workspace allows the robot to plan a proper path for a high DoF robotic arm to grasp the fruits within the workspace.

7.2.2 Fully-Automation System Working in Orchards

The agricultural robotic system in the future will include crop growth monitoring and maintain system, crop yield estimation system, and crop harvesting system. This thesis only focuses on machine vision for harvesting robots. Future work will also investigate other tasks within intelligent farms or orchards such as accurate modelling and monitoring of the orchards, cooperation between unmanned working vehicles and UnmanNed Aerial Vehicle (UAV) to improve the efficiency of the robotic system, and automatic task assignment of corresponding robotic system. Such improvements can largely reduce required human resources for operation and maintenance an orchard. With the optimisation of orchard structures and advances in robotic systems, fully-automated systems have huge potential in future agriculture.

References

[1] ABARES. Australian vegetable growing farms: an economic survey, 2016-17 and 2017-18. *Australian Bureau of Agricultural and Resource Economics (ABARE): Canberra*, 2018.

[2] Timo Ahonen, Abdenour Hadid, and Matti Pietikainen. Face description with local binary patterns: Application to face recognition. *IEEE transactions on pattern analysis and machine intelligence*, 28(12):2037–2041, 2006.

[3] Ethem Alpaydin. *Introduction to machine learning*. MIT press, 2020.

[4] Boaz Arad, Jos Balendonck, Ruud Barth, Ohad Ben-Shahar, Yael Edan, Thomas Hellström, Jochen Hemming, Polina Kurtser, Ola Ringdahl, Toon Tielen, et al. Development of a sweet pepper harvesting robot. *Journal of Field Robotics*, 2020.

[5] Boaz Arad, Polina Kurtser, Ehud Barnea, Ben Harel, Yael Edan, and Ohad Ben-Shahar. Controlled lighting and illumination-independent target detection for real-time cost-efficient applications. the case study of sweet pepper robotic harvesting. *Sensors*, 19(6):1390, 2019.

[6] Arman Arefi, Asad Modarres Motlagh, Kaveh Mollazade, Rahman Farrokhi Teimourlou, et al. Recognition and localization of ripen tomato based on machine vision. *Australian Journal of Crop Science*, 5(10):1144, 2011.

[7] Shebiah Arivazhagan, R Newlin Shebiah, S Selva Nidhyanandhan, and L Ganesan. Fruit recognition using color and texture features. *Journal of Emerging Trends in Computing and Information Sciences*, 1(2):90–94, 2010.

[8] Tim J Atherton and Darren J Kerbyson. Size invariant circle detection. *Image and Vision computing*, 17(11):795–803, 1999.

[9] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[10] Mohd Ali Balafar, Abdul Rahman Ramli, M Iqbal Saripan, and Syamsiah Mashohor. Review of brain mri image segmentation methods. *Artificial Intelligence Review*, 33(3):261–274, 2010.

[11] Suchet Bargoti and James Underwood. Image classification with orchard metadata. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5164–5170. IEEE, 2016.

[12] Suchet Bargoti and James Underwood. Deep fruit detection in orchards. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3626–3633. IEEE, 2017.

[13] Suchet Bargoti and James P Underwood. Image segmentation for fruit detection and yield estimation in apple orchards. *Journal of Field Robotics*, 34(6):1039–1060, 2017.

[14] Suchet Bargoti, James P Underwood, Juan I Nieto, and Salah Sukkarieh. A pipeline for trunk detection in trellis structured apple orchards. *Journal of field robotics*, 32(8):1075–1094, 2015.

[15] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[16] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape context: A new descriptor for shape matching and object recognition. In *Advances in neural information processing systems*, pages

831–837, 2001.

[17] Lisa M Belue and Kenneth W Bauer Jr. Determining input features for multilayer perceptrons. *Neurocomputing*, 7(2):111–121, 1995.

[18] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact++: Better real-time instance segmentation. *arXiv preprint arXiv:1912.06218*, 2019.

[19] Daniel Bolya, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: real-time instance segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9157–9166, 2019.

[20] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. In *European conference on computer vision*, pages 778–792. Springer, 2010.

[21] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017.

[22] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[23] Chengwei. Recent advances in deep learning for object detection. https://www.dlology.com/blog/recent-advances-in-deep-learning-for-object-detection/, 2020.

[24] Zhao De-An, Lv Jidong, Ji Wei, Zhang Ying, and Chen Yu. Design and control of an apple harvesting robot. *Biosystems engineering*, 110(2):112–122, 2011.

[25] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[26] Yubin Deng, Chen Change Loy, and Xiaoou Tang. Image aesthetic assessment: An experimental survey. *IEEE Signal Processing Magazine*, 34(4):80–106, 2017.

[27] Thomas G Dietterich et al. Ensemble learning. *The handbook of brain theory and neural networks*, 2:110–125, 2002.

[28] Paul E Downing, Yuhong Jiang, Miles Shuman, and Nancy Kanwisher. A cortical area selective for visual processing of the human body. *Science*, 293(5539):2470–2473, 2001.

[29] Sahibsingh A Dudani. The distance-weighted k-nearest-neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics*, (4):325–327, 1976.

[30] Mark Everingham and John Winn. The pascal visual object classes challenge 2012 (voc2012) development kit. *Pattern Analysis, Statistical Modelling and Computational Learning, Tech. Rep*, 2011.

[31] Di Feng, Lars Rosenbaum, and Klaus Dietmayer. Towards safe autonomous driving: Capture uncertainty in the deep neural network for lidar 3d vehicle detection. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3266–3273. IEEE, 2018.

[32] Qingchun Feng, Xiaonan Wang, Guohua Wang, and Zhen Li. Design and test of tomatoes harvesting robot. In *2015 IEEE International Conference on Information and Automation*,

pages 949–952. IEEE, 2015.

[33] Alex Flint, Anthony Dick, and Anton Van Den Hengel. Thrift: Local 3d structure recognition. In *9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications (DICTA 2007)*, pages 182–188. IEEE, 2007.

[34] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and Jose Garcia-Rodriguez. A review on deep learning techniques applied to semantic segmentation. *arXiv preprint arXiv:1704.06857*, 2017.

[35] Matt W Gardner and SR Dorling. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. *Atmospheric environment*, 32(14-15):2627–2636, 1998.

[36] Yuanyue Ge, Ya Xiong, Gabriel Lins Tenorio, and Pål Johan From. Fruit localization and environment perception for strawberry harvesting robots. *IEEE Access*, 7:147642–147652, 2019.

[37] Andreas Geiger, Julius Ziegler, and Christoph Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *2011 IEEE intelligent vehicles symposium (IV)*, pages 963–968. Ieee, 2011.

[38] Donald B Gennery. Stereo-camera calibration. In *Proceedings ARPA IUS Workshop*, pages 101–107, 1979.

[39] Zoubin Ghahramani. Unsupervised learning. In *Summer School on Machine Learning*, pages 72–112. Springer, 2003.

[40] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[41] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 38(1):142–158, 2015.

[42] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323, 2011.

[43] Xiao-Feng Gu, Lin Liu, Jian-Ping Li, Yuan-Yuan Huang, and Jie Lin. Data classification based on artificial neural networks. In *2008 International Conference on Apperceiving Computing and Intelligence Analysis*, pages 223–226. IEEE, 2008.

[44] Yanming Guo, Yu Liu, Ard Oerlemans, Songyang Lao, Song Wu, and Michael S Lew. Deep learning for visual understanding: A review. *Neurocomputing*, 187:27–48, 2016.

[45] Yulan Guo, Mohammed Bennamoun, Ferdous Sohel, Min Lu, and Jianwei Wan. 3d object recognition in cluttered scenes with local surface features: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2270–2287, 2014.

[46] Muhammad Salman Haleem, Liangxiu Han, Jano Van Hemert, and Baihua Li. Automatic extraction of retinal features from colour retinal images for glaucoma diagnosis: a review. *Computerized medical imaging and graphics*, 37(7-8):581–596, 2013.

[47] Xian-Feng Hana, Jesse S Jin, Juan Xie, Ming-Jie Wang, and Wei Jiang. A comprehensive review of 3d point cloud descriptors. *arXiv preprint arXiv:1802.02297*, 2018.

[48] Ronny Hänsch, Thomas Weber, and Olaf Hellwich. Comparison of 3d interest point detectors and descriptors for point cloud fusion. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2(3):57, 2014.

[49] M Hassaballah, Aly Amin Abdelmgeid, and Hammam A Alshazly. Image features detection, description and matching. In *Image Feature Detectors and Descriptors*, pages 11–45. Springer, 2016.

[50] Trevor Hastie, Saharon Rosset, Ji Zhu, and Hui Zou. Multi-class adaboost. *Statistics and its Interface*, 2(3):349–360, 2009.

[51] Shigehiko Hayashi, Kenta Shigematsu, Satoshi Yamamoto, Ken Kobayashi, Yasushi Kohno, Junzo Kamata, and Mitsutaka Kurita. Evaluation of a strawberry-harvesting robot in a field test. *Biosystems engineering*, 105(2):160–171, 2010.

[52] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[53] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015.

[54] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[55] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. Rgb-d mapping: Using depth cameras for dense 3d modeling of indoor environments. In *Experimental robotics*, pages 477–491. Springer, 2014.

[56] Wolfgang Hess, Damon Kohler, Holger Rapp, and Daniel Andor. Real-time loop closure in 2d lidar slam. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1271–1278. IEEE, 2016.

[57] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[58] LW Huang and DJ He. Ripe fuji apple detection model analysis in natural tree canopy. *Telkomnika*, 10:1771–1778, 2012.

[59] Calvin Hung, Juan Nieto, Zachary Taylor, James Underwood, and Salah Sukkarieh. Orchard fruit segmentation using multi-spectral feature learning. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5314–5320. IEEE, 2013.

[60] Calvin Hung, James Underwood, Juan Nieto, and Salah Sukkarieh. A feature learning based approach for automated fruit yield estimation. In *Field and service robotics*, pages 485–498. Springer, 2015.

[61] Nazli Ikizler-Cinbis and Stan Sclaroff. Object, scene and actions: Combining multiple features for human action recognition. In *European conference on computer vision*, pages 494–507. Springer, 2010.

[62] Dana E Ilea and Paul F Whelan. Image segmentation based on the integration of colour–texture descriptors—a review. *Pattern Recognition*, 44(10-11):2479–2501, 2011.

[63] Anastasia Ioannidou, Elisavet Chatzilari, Spiros Nikolopoulos, and Ioannis Kompatsiaris. Deep learning advances in computer vision with 3d data: A survey. *ACM Computing Surveys (CSUR)*, 50(2):1–38, 2017.

[64] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[65] SM Sofiqul Islam, Shanto Rahman, Md Mostafijur Rahman, Emon Kumar Dey, and Mohammad Shoyaib. Application of deep learning to computer vision: A comprehensive study. In *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)*, pages 592–597. IEEE, 2016.

[66] Samarth Brahmbhatt James Hays and John Lambert. Cs 6476: Computer vision,project 2: Local feature matching. https://www.cc.gatech.edu/ hays/compvision/proj2/, 2018.

[67] Wei Ji, Dean Zhao, Fengyi Cheng, Bo Xu, Ying Zhang, and Jinjing Wang. Automatic recognition vision system guided for apple harvesting robot. *Computers & Electrical Engineering*, 38(5):1186–1195, 2012.

[68] Weikuan Jia, Yuyu Tian, Rong Luo, Zhonghua Zhang, Jian Lian, and Yuanjie Zheng. Detection and segmentation of overlapped fruits based on optimized mask r-cnn application in apple harvesting robot. *Computers and Electronics in Agriculture*, 172:105380, 2020.

[69] Andrew E Johnson. Spin-images: a representation for 3-d surface matching. 1997.

[70] Andreas Kamilaris and Francesc X Prenafeta-Boldú. Deep learning in agriculture: A survey. *Computers and electronics in agriculture*, 147:70–90, 2018.

[71] Sekitoshi Kanai, Yasuhiro Fujiwara, and Sotetsu Iwamura. Preventing gradient explosions in gated recurrent units. In *Advances in neural information processing systems*, pages 435–444, 2017.

[72] Hanwen Kang and Chao Chen. Fast implementation of real-time fruit detection in apple orchards using deep learning. *Computers and Electronics in Agriculture*, 168:105108, 2020.

[73] Keren Kapach, Ehud Barnea, Rotem Mairon, Yael Edan, and Ohad Ben-Shahar. Computer vision for fruit harvesting robots–state of the art and challenges ahead. *International Journal of Computational Vision and Robotics*, 3(1/2):4–34, 2012.

[74] Ramesh Kestur, Avadesh Meduri, and Omkar Narasipura. Mangonet: A deep semantic segmentation architecture for a method to detect and count mangoes in an open orchard. *Engineering Applications of Artificial Intelligence*, 77:59–69, 2019.

[75] Sameer Khan and Suet-Peng Yong. A comparison of deep learning and hand crafted features in medical image modality classification. In *2016 3rd International Conference on Computer and Information Sciences (ICCOINS)*, pages 633–638. IEEE, 2016.

[76] Kourosh Khoshelham, DR Dos Santos, and George Vosselman. Generation and weighting of 3d point correspondences for improved registration of rgb-d data. *Proc. ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci*, 5:W2, 2013.

[77] Raymond Kirk, Grzegorz Cielniak, and Michael Mangan. L* a* b* fruits: A rapid and robust outdoor fruit detection system combining bio-inspired features with one-stage deep learning networks. *Sensors*, 20(1):275, 2020.

[78] A Koirala, KB Walsh, Z Wang, and C McCarthy. Deep learning for real-time fruit detection and orchard fruit load estimation: Benchmarking of 'mangoyolo'. *Precision Agriculture*, 20(6):1107–1135, 2019.

[79] Anand Koirala, Kerry B Walsh, Zhenglin Wang, and Cheryl McCarthy. Deep learning–method overview and review of use for fruit detection and yield estimation. *Computers and Electronics in Agriculture*, 162:219–234, 2019.

[80] Sethu Hareesh Kolluru. Investigations on the inference optimization techniques and their impact on multiple hardware platforms for semantic segmentation. *arXiv preprint arXiv:1911.12993*, 2019.

[81] Scott Krig. *Computer vision metrics: Survey, taxonomy, and analysis*. Apress, 2014.

[82] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[83] Christoph H Lampert, Matthew B Blaschko, and Thomas Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008.

[84] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[85] Yann LeCun, Patrick Haffner, Léon Bottou, and Yoshua Bengio. Object recognition with gradient-based learning. In *Shape, contour and grouping in computer vision*, pages 319–345. Springer, 1999.

[86] Christopher Lehnert, Andrew English, Christopher McCool, Adam W Tow, and Tristan Perez. Autonomous sweet pepper harvesting for protected cropping systems. *IEEE Robotics and Automation Letters*, 2(2):872–879, 2017.

[87] Christopher Lehnert, Inkyu Sa, Christopher McCool, Ben Upcroft, and Tristan Perez. Sweet pepper pose detection and grasping for automated crop harvesting. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2428–2434. IEEE, 2016.

[88] Chengcai Leng, Hai Zhang, Bo Li, Guorong Cai, Zhao Pei, and Li He. Local feature descriptor for image matching: A survey. *IEEE Access*, 7:6424–6434, 2018.

[89] Chengcai Leng, Hai Zhang, Bo Li, Guorong Cai, Zhao Pei, and Li He. Local feature descriptor for image matching: A survey. *IEEE Access*, 7:6424–6434, 2018.

[90] Lei Li, Qin Zhang, and Danfeng Huang. A review of imaging techniques for plant phenotyping. *Sensors*, 14(11):20078–20111, 2014.

[91] Yanan Li, Zhiguo Cao, Yang Xiao, and Armin B Cremers. Deepcotton: in-field cotton segmentation using deep fully convolutional network. *Journal of Electronic Imaging*, 26(5):053028, 2017.

[92] Qiaokang Liang, Wei Zhu, Jianyong Long, Yaonan Wang, Wei Sun, and Wanneng Wu. A real-time detection framework for on-tree mango based on ssd network. In *International Conference on Intelligent Robotics and Applications*, pages 423–436. Springer, 2018.

[93] Guichao Lin, Yunchao Tang, Xiangjun Zou, Juntao Xiong, and Jinhui Li. Guava detection and

pose estimation using a low-cost rgb-d sensor in the field. *Sensors*, 19(2):428, 2019.

[94] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[95] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[96] Wei-Chao Lin, Chih-Fong Tsai, Zong-Yao Chen, and Shih-Wen Ke. Keypoint selection for efficient bag-of-words feature generation and effective image classification. *Information Sciences*, 329:33–51, 2016.

[97] Tony Lindeberg. Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention. *International Journal of Computer Vision*, 11(3):283–318, 1993.

[98] Tong Liu, Xiaowei Zhang, Ziang Wei, and Zejian Yuan. A robust fusion method for rgb-d slam. In *2013 Chinese Automation Congress*, pages 474–481. IEEE, 2013.

[99] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[100] Xiaoyang Liu, Dean Zhao, Weikuan Jia, Wei Ji, Chengzhi Ruan, and Yueping Sun. Cucumber fruits detection in greenhouses based on instance segmentation. *IEEE Access*, 7:139635–139642, 2019.

[101] Zhihao Liu, Jingzhu Wu, Longsheng Fu, Yaqoob Majeed, Yali Feng, Rui Li, and Yongjie Cui. Improved kiwifruit detection using pre-trained vgg16 with rgb and nir information fusion. *IEEE Access*, 2019.

[102] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-voxel cnn for efficient 3d deep learning. In *Advances in Neural Information Processing Systems*, pages 963–973, 2019.

[103] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[104] G Lowe. Sift-the scale invariant feature transform. *Int. J*, 2:91–110, 2004.

[105] Lufeng Luo, Yunchao Tang, Xiangjun Zou, Chenglin Wang, Po Zhang, and Wenxian Feng. Robust grape cluster detection in a vineyard by combining the adaboost framework and multiple color components. *Sensors*, 16(12):2098, 2016.

[106] Zoltan-Csaba Marton, Dejan Pangercic, Nico Blodow, and Michael Beetz. Combined 2d–3d categorization and classification for multimodal perception systems. *The International Journal of Robotics Research*, 30(11):1378–1402, 2011.

[107] Christopher McCool, Inkyu Sa, Feras Dayoub, Christopher Lehnert, Tristan Perez, and Ben Upcroft. Visual detection of occluded crop: For automated harvesting. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2506–2512. IEEE, 2016.

[108] SS Mehta and TF Burks. Vision-based control of robotic manipulator for citrus harvesting.

Computers and Electronics in Agriculture, 102:146–158, 2014.

[109] Ajmal S Mian, Mohammed Bennamoun, and Robyn Owens. Keypoint detection and local feature matching for textured 3d face recognition. *International Journal of Computer Vision*, 79(1):1–12, 2008.

[110] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571. IEEE, 2016.

[111] Marius Muja, Radu Bogdan Rusu, Gary Bradski, and David G Lowe. Rein-a fast, robust, scalable recognition infrastructure. In *2011 IEEE international conference on robotics and automation*, pages 2939–2946. IEEE, 2011.

[112] Raul Mur-Artal, Jose Maria Martinez Montiel, and Juan D Tardos. Orb-slam: a versatile and accurate monocular slam system. *IEEE transactions on robotics*, 31(5):1147–1163, 2015.

[113] Eduardo A Murillo-Bracamontes, Miguel E Martinez-Rosas, Manuel M Miranda-Velasco, Horacio L Martinez-Reyes, Jesus R Martinez-Sandoval, and Humberto Cervantes-de Avila. Implementation of hough transform for fruit image segmentation. *Procedia Engineering*, 35:230–239, 2012.

[114] Fionn Murtagh. Multilayer perceptrons for classification and regression. *Neurocomputing*, 2(5-6):183–197, 1991.

[115] Loris Nanni, Stefano Ghidoni, and Sheryl Brahnam. Handcrafted vs. non-handcrafted features for computer vision classification. *Pattern Recognition*, 71:158–172, 2017.

[116] Yasin Osroosh, Lav R Khot, and R Troy Peters. Economical thermal-rgb imaging system for monitoring agricultural crops. *Computers and Electronics in Agriculture*, 147:34–43, 2018.

[117] Hetal N Patel, RK Jain, and Manjunath V Joshi. Fruit detection using improved multiple features based algorithm. *International journal of computer applications*, 13(2):1–5, 2011.

[118] Taihú Pire, Thomas Fischer, Javier Civera, Pablo De Cristóforis, and Julio Jacobo Berlles. Stereo parallel tracking and mapping for robot localization. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1373–1378. IEEE, 2015.

[119] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 918–927, 2018.

[120] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.

[121] Charles R Qi, Hao Su, Matthias Nießner, Angela Dai, Mengyuan Yan, and Leonidas J Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5648–5656, 2016.

[122] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017.

[123] Jurij Rakun, Denis Stajnko, and Damjan Zazula. Detecting fruits in natural scenes by using

spatial-frequency based texture analysis and multiview geometry. *Computers and Electronics in Agriculture*, 76(1):80–88, 2011.

[124] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[125] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.

[126] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[127] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[128] Jos BTM Roerdink and Arnold Meijster. The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta informaticae*, 41(1, 2):187–228, 2000.

[129] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[130] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011.

[131] Dennis W Ruck, Steven K Rogers, and Matthew Kabrisky. Feature selection using a multilayer perceptron. *Journal of Neural Network Computing*, 2(2):40–48, 1990.

[132] Radu Bogdan Rusu, Nico Blodow, and Michael Beetz. Fast point feature histograms (fpfh) for 3d registration. In *2009 IEEE international conference on robotics and automation*, pages 3212–3217. IEEE, 2009.

[133] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, and Michael Beetz. Learning informative point classes for the acquisition of object model maps. In *2008 10th International Conference on Control, Automation, Robotics and Vision*, pages 643–650. IEEE, 2008.

[134] Inkyu Sa, Zongyuan Ge, Feras Dayoub, Ben Upcroft, Tristan Perez, and Chris McCool. Deepfruits: A fruit detection system using deep neural networks. *Sensors*, 16(8):1222, 2016.

[135] Samuele Salti, Federico Tombari, and Luigi Di Stefano. A performance evaluation of 3d keypoint detectors. In *2011 International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pages 236–243. IEEE, 2011.

[136] Allah Bux Sargano, Plamen Angelov, and Zulfiqar Habib. A comprehensive review on handcrafted and learning-based action representation approaches for human activity recognition. *applied sciences*, 7(1):110, 2017.

[137] Jochen Schmidt, Heinrich Niemann, and Sebastian Vogt. Dense disparity maps in real-time with an application to augmented reality. In *Sixth IEEE Workshop on Applications of Computer Vision, 2002.(WACV 2002). Proceedings.*, pages 225–230. IEEE, 2002.

[138] Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceed-*

ings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4104–4113, 2016.

[139] Paul Scovanner, Saad Ali, and Mubarak Shah. A 3-dimensional sift descriptor and its application to action recognition. In *Proceedings of the 15th ACM international conference on Multimedia*, pages 357–360, 2007.

[140] Nicu Sebe, Ira Cohen, Ashutosh Garg, and Thomas S Huang. *Machine learning in computer vision*, volume 29. Springer Science & Business Media, 2005.

[141] Chang Shu, Xiaoqing Ding, and Chi Fang. Histogram of the oriented gradient for face recognition. *Tsinghua Science and Technology*, 16(2):216–224, 2011.

[142] Yongsheng Si, Gang Liu, and Juan Feng. Location of apples in trees using stereoscopic vision. *Computers and Electronics in Agriculture*, 112:68–74, 2015.

[143] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pages 746–760. Springer, 2012.

[144] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[145] Zhenzhen Song, Longsheng Fu, Jingzhu Wu, Zhihao Liu, Rui Li, and Yongjie Cui. Kiwifruit detection in field images using faster r-cnn with vgg16. *IFAC-PapersOnLine*, 52(30):76–81, 2019.

[146] Aristeidis Sotiras, Christos Davatzikos, and Nikos Paragios. Deformable medical image registration: A survey. *IEEE transactions on medical imaging*, 32(7):1153–1190, 2013.

[147] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015.

[148] Shiliang Sun and Jin Zhou. A review of adaptive feature extraction and classification methods for eeg-based brain-computer interfaces. In *2014 International Joint Conference on Neural Networks (IJCNN)*, pages 1746–1753. IEEE, 2014.

[149] Johan AK Suykens and Joos Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999.

[150] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

[151] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[152] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[153] Johan WH Tangelder and Remco C Veltkamp. A survey of content based 3d shape retrieval

methods. In *Proceedings Shape Modeling Applications, 2004.*, pages 145–156. IEEE, 2004.

[154] Yunong Tian, Guodong Yang, Zhe Wang, En Li, and Zize Liang. Detection of apple lesions in orchards based on deep learning methods of cyclegan and yolov3-dense. *Journal of Sensors*, 2019, 2019.

[155] Yunong Tian, Guodong Yang, Zhe Wang, Hao Wang, En Li, and Zize Liang. Apple detection during different growth stages in orchards using the improved yolo-v3 model. *Computers and electronics in agriculture*, 157:417–426, 2019.

[156] Sergey Tulyakov, Stefan Jaeger, Venu Govindaraju, and David Doermann. Review of classifier combination methods. In *Machine learning in document analysis and recognition*, pages 361–386. Springer, 2008.

[157] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[158] Anup Vibhute and Shrikant K Bodhe. Applications of image processing in agriculture: a survey. *International Journal of Computer Applications*, 52(2), 2012.

[159] Adar Vit and Guy Shani. Comparing rgb-d sensors for close range outdoor agricultural phenotyping. *Sensors*, 18(12):4413, 2018.

[160] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep learning for computer vision: A brief review. *Computational intelligence and neuroscience*, 2018, 2018.

[161] Chenglin Wang, Won Suk Lee, Xiangjun Zou, Daeun Choi, Hao Gan, and Justice Diamond. Detection and counting of immature green citrus fruit based on the local binary patterns (lbp) feature using illumination-normalized images. *Precision agriculture*, 19(6):1062–1083, 2018.

[162] Jin-jing Wang, De-an Zhao, Wei Ji, Jun-jun Tu, and Ying Zhang. Application of support vector machine to apple recognition using in apple harvesting robot. In *2009 International Conference on Information and Automation*, pages 1110–1115. IEEE, 2009.

[163] Rui Wang, Martin Schworer, and Daniel Cremers. Stereo dso: Large-scale direct sparse visual odometry with stereo cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3903–3911, 2017.

[164] Zhixin Wang and Kui Jia. Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection. *arXiv preprint arXiv:1903.01864*, 2019.

[165] Daniel J Withey and Zoltan J Koles. A review of medical image segmentation: methods and available software. *International Journal of Bioelectromagnetism*, 10(3):125–148, 2008.

[166] Ryan W Wolcott and Ryan M Eustice. Robust lidar localization using multiresolution gaussian mixture maps for autonomous driving. *The International Journal of Robotics Research*, 36(3):292–319, 2017.

[167] Chee Sun Won, Dong Kwon Park, and Soo-Jun Park. Efficient use of mpeg-7 edge histogram descriptor. *ETRI journal*, 24(1):23–30, 2002.

[168] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of*

the IEEE conference on computer vision and pattern recognition, pages 1912–1920, 2015.

[169] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[170] Zhonghong Xie, Changying Ji, Xiaoqing Guo, and Shougang Ren. An object detection method for quasi-circular fruits based on improved hough transform. *Transactions of the Chinese Society of Agricultural Engineering*, 26(7):157–162, 2010.

[171] Ya Xiong, Yuanyue Ge, Lars Grimstad, and Pål J From. An autonomous strawberry-harvesting robot: Design, development, integration, and field evaluation. *Journal of Field Robotics*, 37(2):202–224, 2020.

[172] Rui Xu and Donald Wunsch. Survey of clustering algorithms. *IEEE Transactions on neural networks*, 16(3):645–678, 2005.

[173] Hiroaki Yaguchi, Kotaro Nagahama, Takaomi Hasegawa, and Masayuki Inaba. Development of an autonomous tomato harvesting robot with rotational plucking gripper. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 652–657. IEEE, 2016.

[174] Hongpeng Yin, Yi Chai, Simon X Yang, and Gauri S Mittal. Ripe tomato recognition and localization for a tomato harvesting robotic system. In *2009 International Conference of Soft Computing and Pattern Recognition*, pages 557–562. IEEE, 2009.

[175] Jun Yu, Jinghan Yao, Jian Zhang, Zhou Yu, and Dacheng Tao. Sprnet: Single-pixel reconstruction for one-stage instance segmentation. *IEEE Transactions on Cybernetics*, 2020.

[176] Yang Yu, Kailiang Zhang, Li Yang, and Dongxing Zhang. Fruit detection for strawberry harvesting robot in non-structural environment based on mask-rcnn. *Computers and Electronics in Agriculture*, 163:104846, 2019.

[177] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[178] Matthew D Zeiler, Graham W Taylor, and Rob Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *2011 International Conference on Computer Vision*, pages 2018–2025. IEEE, 2011.

[179] Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 3dmatch: Learning local geometric descriptors from rgb-d reconstructions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1802–1811, 2017.

[180] Jun Zhao, Joel Tow, and Jayantha Katupitiya. On-tree fruit recognition using texture properties and color data. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 263–268. IEEE, 2005.

[181] Yuanshen Zhao, Liang Gong, Yixiang Huang, and Chengliang Liu. A review of key techniques of vision-based control for harvesting robot. *Computers and Electronics in Agriculture*, 127:311–323, 2016.

[182] Yuanshen Zhao, Liang Gong, Yixiang Huang, and Chengliang Liu. Robust tomato recognition for robotic harvesting using feature images fusion. *Sensors*, 16(2):173, 2016.

[183] Yuanshen Zhao, Liang Gong, Bin Zhou, Yixiang Huang, and Chengliang Liu. Detecting tomatoes in greenhouse scenes by combining adaboost classifier and colour analysis. *biosystems engineering*, 148:127–137, 2016.

[184] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11):3212–3232, 2019.

[185] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, and Xindong Wu. Object detection with deep learning: A review. *IEEE transactions on neural networks and learning systems*, 30(11):3212–3232, 2019.

[186] Wang Zhiqiang and Liu Jun. A review of object detection based on convolutional neural network. In *2017 36th Chinese Control Conference (CCC)*, pages 11104–11109. IEEE, 2017.

[187] Rong Zhou, Lutz Damerow, Yurui Sun, and Michael M Blanke. Using colour features of cv.'gala'apple fruits in an orchard in image processing to predict yield. *Precision Agriculture*, 13(5):568–580, 2012.

[188] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018.

[189] Quanwen Zhu, Long Chen, Qingquan Li, Ming Li, Andreas Nüchter, and Jian Wang. 3d lidar point cloud based intersection recognition for autonomous driving. In *2012 IEEE Intelligent Vehicles Symposium*, pages 456–461. IEEE, 2012.

[190] Xiaojin Zhu and Andrew B Goldberg. Introduction to semi-supervised learning. *Synthesis lectures on artificial intelligence and machine learning*, 3(1):1–130, 2009.