

Machine vision system for real-time debris detection on mechanical wild blueberry harvesters



Anup Kumar Das^{a,b}, Travis J. Esau^{a*}, Qamar U. Zaman^a, Aitazaz A. Farooque^c, Arnold W. Schumann^d, Patrick J. Hennessy^a

^a Department of Engineering, Faculty of Agriculture, Dalhousie University, Truro, NS B2N 5E3, Canada

^b Department of Agricultural and Biosystems Engineering, North Dakota State University, Fargo, ND 58102, United States

^c School of Sustainable Design Engineering, University of Prince Edward Island, Charlottetown, PE C1A 4P3, Canada

^d Citrus Research and Education Center, University of Florida, Lake Alfred, FL 33850, United States

ARTICLE INFO

Editor: Stephen Symons

Keywords:

Debris detection
Deep learning
Fruit quality
Image processing
YOLOv3

ABSTRACT

Improving fruit quality is important to wild blueberry producers. Plant debris like leaves and stems caused a reduction in fruit quality during mechanical harvesting. A debris detection system was developed using two Logitech C920 webcam cameras mounted before and after the blower fan on a commercial mechanical wild blueberry harvester. Images were collected from two commercially managed wild blueberry fields located in central Nova Scotia to develop a dataset of 1000 images. Two convolutional neural networks (CNNs), YOLOv3 and YOLOv3-SPP, were used and compared for performance analysis. The CNNs were trained using no augmented images, images with five augmentations (sharpening, brightness, contrast, gamma correction, and saturation), and images with five augmentations that were more heavily weighted towards gamma. Four different computer hardware packages (Hewlett-Packard, Shuttle XPC, Jetson TX2, and GPU-based Desktop) were used to determine the appropriate hardware and CNN model combination for real-time performance. YOLOv3 and YOLOv3-SPP were trained to mAP scores of 72.87 and 74.38% using the dataset with more gamma-augmented images. This was an improvement over the dataset with no augmentation (67.75; 68.49%) and equally numbered augmentations (71.26%; 73.03%). YOLOv3-SPP combined with an Intel® Core™ i9-7900X CPU and Nvidia GeForce RTX™ 2080 Ti GPU in a desktop computer achieved the fastest detection rate of 33.30 ms. The detection speed on all other hardware configurations exceeded 33.30 ms, indicating that a powerful desktop GPU is required for real-time performance of this task. With adequate processing hardware, the developed technology could be integrated into a control system to automatically adjust brushes based on conveyor information from mechanical wild blueberry harvesters to make cleaning berries easier and enhance fruit quality.

1. Introduction

Wild blueberry (*Vaccinium angustifolium* Ait.) is an important economic crop of Northeastern North America and is commercially managed in Atlantic Canada, Quebec, and Maine. More than 80% of commercially managed fields are harvested mechanically following a two-year production cycle [1]. Wild blueberry is different from cultivated highbush blueberry (*Vaccinium corymbosum*) as it grows naturally from rhizomes and spreads under the soil at about 0.05 to 0.08 m per year [2]. Wild blueberry is richer in antioxidants than the cultivated highbush blueberry [3]. Wild blueberries have health benefits including anti-aging and anti-inflammatory effects [3] which help to reduce the risk of cardiovascular disease and cancer [4]. However, the wild

blueberry is comparatively smaller in size than the highbush blueberry [5]. The modern mechanical wild blueberry harvester is typically mounted on an agricultural farm tractor and fitted with a rake-type rotational berry picking head and two conveyors (side and rear) for conveying harvested berries to the storage bin. The storage bins, once filled, are loaded from the field onto a trailer for delivery to a receiving shed and then to a processing facility for final product cleaning and grading. To reduce the debris in the harvested fruit, a blower fan is mounted at the interchange between the side and rear conveyor on the harvester. The fan helps to remove debris from berries while harvesting the crop [6]. Continuous improvements in field management practices (i.e., application of fungicides, herbicides, fertilizers, pollination, and pruning) have increased wild blueberry plant biomass and fruit yield

* Corresponding author.

E-mail address: tesau@dal.ca (T.J. Esau).

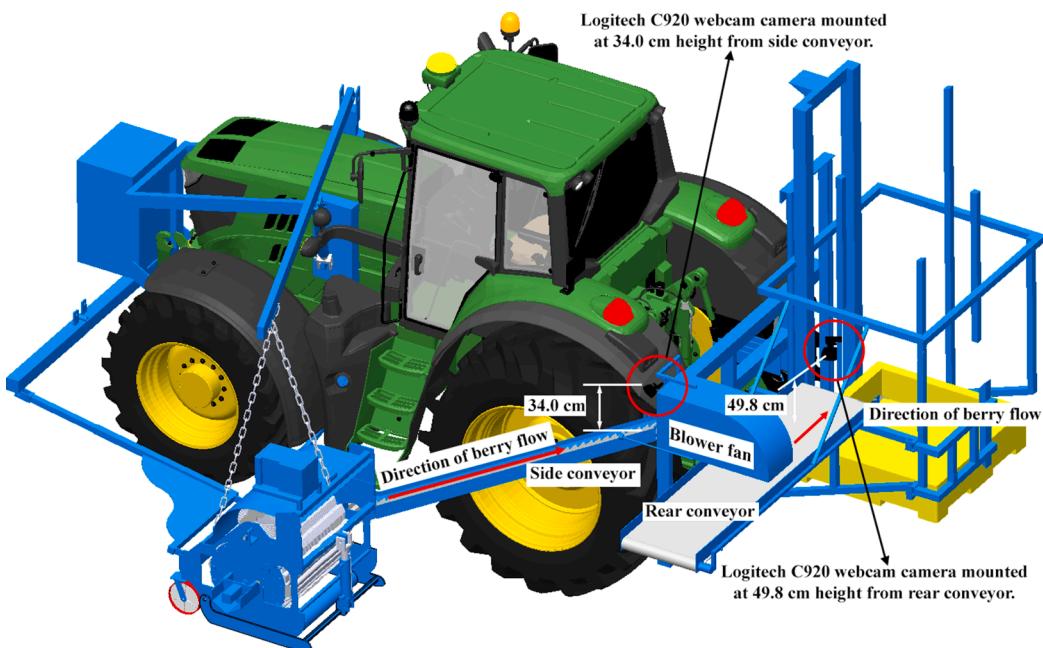


Fig. 1. Camera setup at the side and rear conveyor.



Fig. 2. Location of fields in Google Maps™. (a) Debert site, (b) East Mines site (Adapted from [51]).

Table 1
Enhancement of gamma, sharpness, brightness, contrast, and saturation values of images for augmenting dataset.

Dataset name	Filter Used	Values	Subset images
Augmentation-1	Gamma correction (0.1 to 6.99)	0.7	200
		0.8	
		0.9	
		1.1	
		1.2	
	Sharpness (1 to 99)	10	200
Augmentation-2	Brightness (-255 to 255)	20	
	Contrast (-127 to 127)	20	
	Gamma correction (0.1 to 6.99)	1.3	
	Saturation (-255 to 255)	15	
Augmentation-3	Augmentation-1		
	Augmentation-2		

[6]. As a result of increased biomass, additional blueberry leaves, stems, and foreign debris are harvested and transferred into the berry storage bins, harming fruit quality at the processing facilities [6]. The ratio of additional leaf, dirt, and stem debris is required to lie below a tolerable bound and should be separated from ripe berries during harvesting to improve fruit quality. Esau et al. [6] tested four different blower fan

speeds (0, 14, 18, and 23 m·s⁻¹) and successfully removed 98.8 and 98.6% of debris by weight using the 23 m·s⁻¹ fan speed under low and high plant moisture conditions, respectively. However, debris occasionally gets stuck in air inlet vanes of commercial blower fan systems and hampers the debris cleaning performance leading to reduced field efficiency of harvesters. A debris cleaning brush located on the harvester picking reel is primarily responsible for cleaning debris during mechanical berry picking. A machine vision system can be implemented as a feedback control system for automating the debris cleaning brush speed based on different field conditions and can prevent debris from entering the storage bin. The system can also be incorporated for further upgrading from a dual blower fan system to a multi-air channel-based blower fan system for optimizing debris cleaning performance and enhancing harvester efficiency.

Machine vision systems are becoming the core of many of the latest technological developments in precision agriculture due to their automatic decision-making ability [7,8]. Zaman et al. [9] developed a computer vision-based automated wild blueberry fruit yield monitoring system. The system achieved a high correlation between actual and predicted fruit yield with an R² value of 0.99. Chang et al. [10] developed a color co-occurrence matrix-based machine system to identify bare spots, wild blueberry plants, and weeds and apply agrochemicals in a spot-specific manner and achieved overall 94.9% accuracy based on

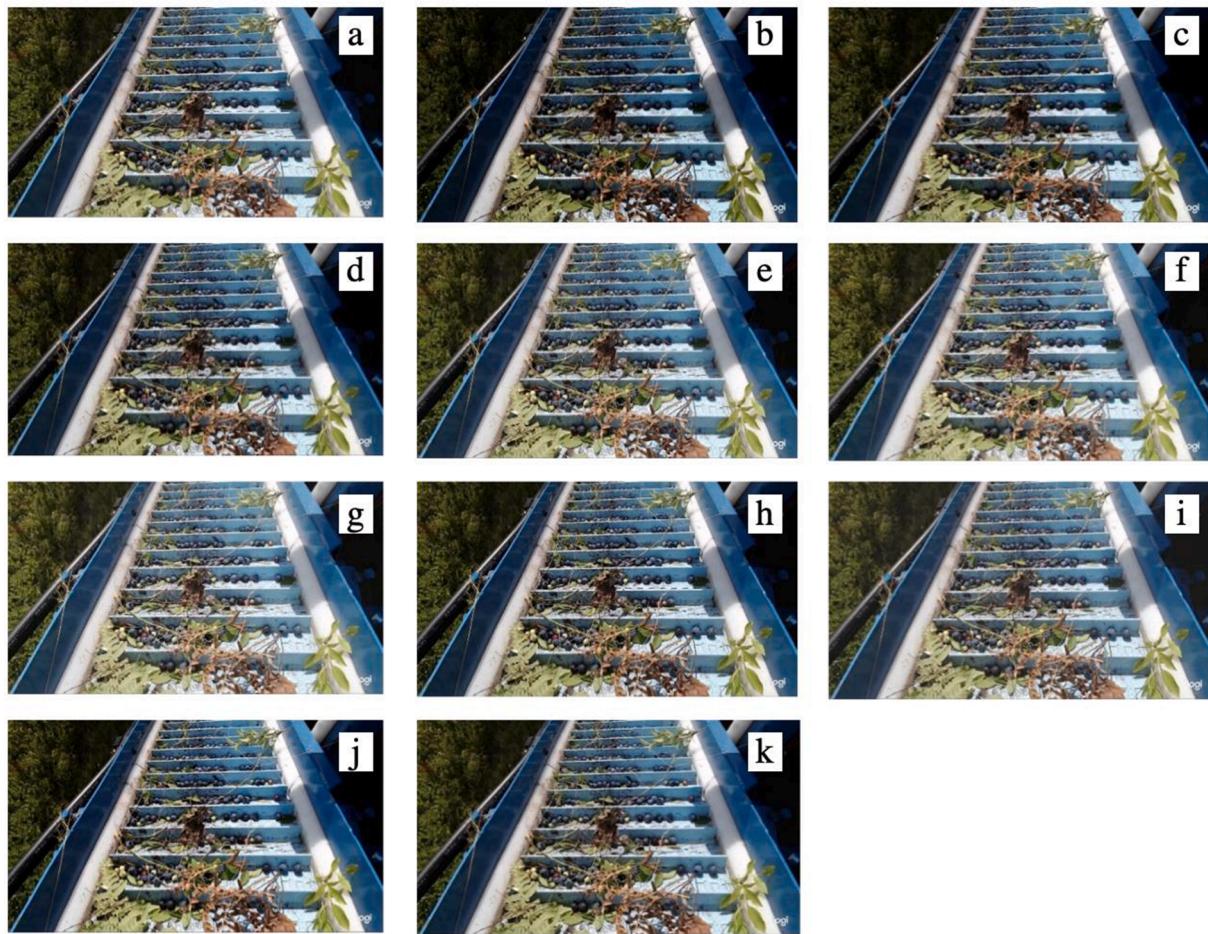


Fig. 3. Effects of the image augmentation techniques (a) Original image = no augmentation, (b) gamma correction = 0.70, (c) gamma correction = 0.80, (d) gamma correction = 0.90, (e) gamma correction = 1.10, (f) gamma correction = 1.20, (g) gamma correction = 1.30, (h) sharpening = 10, (i) brightness = 20, (j) contrast = 20, (k) saturation = 15.

Table 2

Description of training and testing (no augmentation, T1, and T2) datasets.

Dataset name	Dataset description	Number of images	Training images
No augmentation	Original dataset	1000	900
T1	Original dataset + Augmentation-2	2000	1800
T2	Original dataset + Augmentation-1 + Augmentation-2	3000	2700

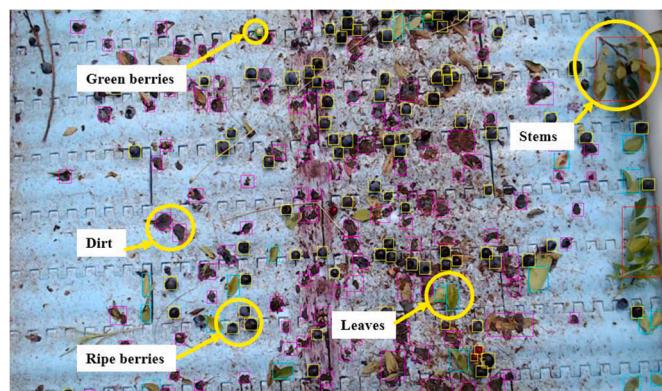


Fig. 4. Labeling of leaves, stems, dirt, green berries, and ripe berries on images using custom software by Schumann et al. [56].

three co-occurrence matrices and selected textural features. However, these machine vision approaches cannot learn visual features automatically. They require manual feature engineering skills. Inappropriate selection of features decreases accuracy and increases the computation time for the classifier [10]. Deep learning (DL) approaches are powerful for automatic feature extraction and image classification [11]. Several researchers used Convolutional Neural Network (CNN) based DL approaches for estimating fruit quality [12,13,14]. Wang et al. [14] classified intact and damaged highbush blueberries with 88.44% and 87.84% precision, respectively using two modern CNN architectures, ResNet [15] and ResNeXt [16]. An automatic lettuce quality evaluation system was developed by Cavallo et al. [12] using a deep CNN. The system could identify color-distorted lettuce through the packaging with 83% classification accuracy [12]. Ni et al. [17] developed a pipeline using DL techniques to classify blueberry fruit traits, including estimating maturity, evaluating compactness, and counting the number of berries per cluster. They implemented a Mask R-CNN model for segmenting and classifying berries based on fruit maturity and achieved average precision of 90.6% for segmentation and 78.3% for

Table 3

Specifications of four different hardware platforms (Hewlett-Packard, Shuttle XPC, Jetson TX2, and Desktop) used in determining detection speed.

Specifications	Hewlett-Packard (Hardware-1)	Shuttle XPC (Hardware-2)	Jetson TX2 (Hardware-3)	Desktop (Hardware-4)
CPU	Intel® Core™ i5-4300 U CPU @ 1.90 GHz	Intel® Core™ i7-6700 K CPU @ 4.00 GHz	ARM Cortex-A57 (quadcore) @ 2 GHz + NVIDIA Denver2 (dual core) @ 2 GHz	Intel® Core™ i9-7900X CPU @ 3.30 GHz
GPU	NA	NA	256-core Pascal™ @ 1300MHz	NVIDIA Turing™ GeForce RTX™ 2080 Ti @ 1665 MHz
Memory	8 GB DDR3@ 1600 MHz	8 GB DDR4@ 2400 MHz	8GB LPDDR4 @ 1866 MHz	32 GB DDR4 @ 2800 MHz
Storage	218 GB HDD	375 GB HDD	32 GB eMMC 5.1	1 TB SSD
Power (W)	65	300	15	850

classification, respectively. Mask R-CNN yielded mean average precision (mAP) of 71.6 and 78.3% for testing and validation datasets, respectively. They recommended DL for estimating blueberry fruit traits as well as classification and segmentation of blueberries. Gonzalez et al.

[18] proposed a DL network-based system for quantifying blueberries. They used Mask R-CNN with a backbone of ResNet50 and achieved mAP of 90% and 75.90% using an intersection over union threshold of 0.5 for berry segmentation and classification, respectively. Qiao et al. [19] detected and classified the early decay of blueberries using a Deep Residual 3D CNN. They observed that the proposed network achieved a 96.69% detection rate and performed better than the AlexNet and GoogleNet CNNs. However, few studies have been conducted using machine learning for the detection of blueberry bruises [20,21,22], and classification of blueberries at different growth stages [23] which achieved satisfactory results.

You Only Look Once (YOLO) is a CNN architecture for object detection within images. The accuracy and speed of YOLO were improved in YOLOv2 [24] and YOLOv3 [25]. YOLOv3-SPP is an improved version of YOLOv3 that incorporates spatial pyramid pooling (SPP) into the backbone of the YOLO network to enhance spatial features [26]). MacEachern et al. [27] detected maturity stage in wild blueberries using YOLOv3, YOLOv3-Tiny, YOLOv3-SPP, and YOLOv4. They observed that YOLOv3-SPP outperformed other CNNs and achieved a mAP of 77.90% for detecting blue, red, and green berries. An automatic tomato fruit detection system was developed by Liu et al. [28] using YOLOv3 called YOLO-Tomato that achieved 96.40% average precision (AP). Tian et al. [8] also implemented YOLOv3, incorporated with DenseNet developed by Huang et al. [29], for detecting young, expanding, and ripe apples in the orchard. Their implementation of YOLOv3 achieved an 81.70% F1-score. A densely connected SPP approach was implemented on YOLOv2 by Huang et al. [26] to improve detection accuracy. This approach improved the mean average precision

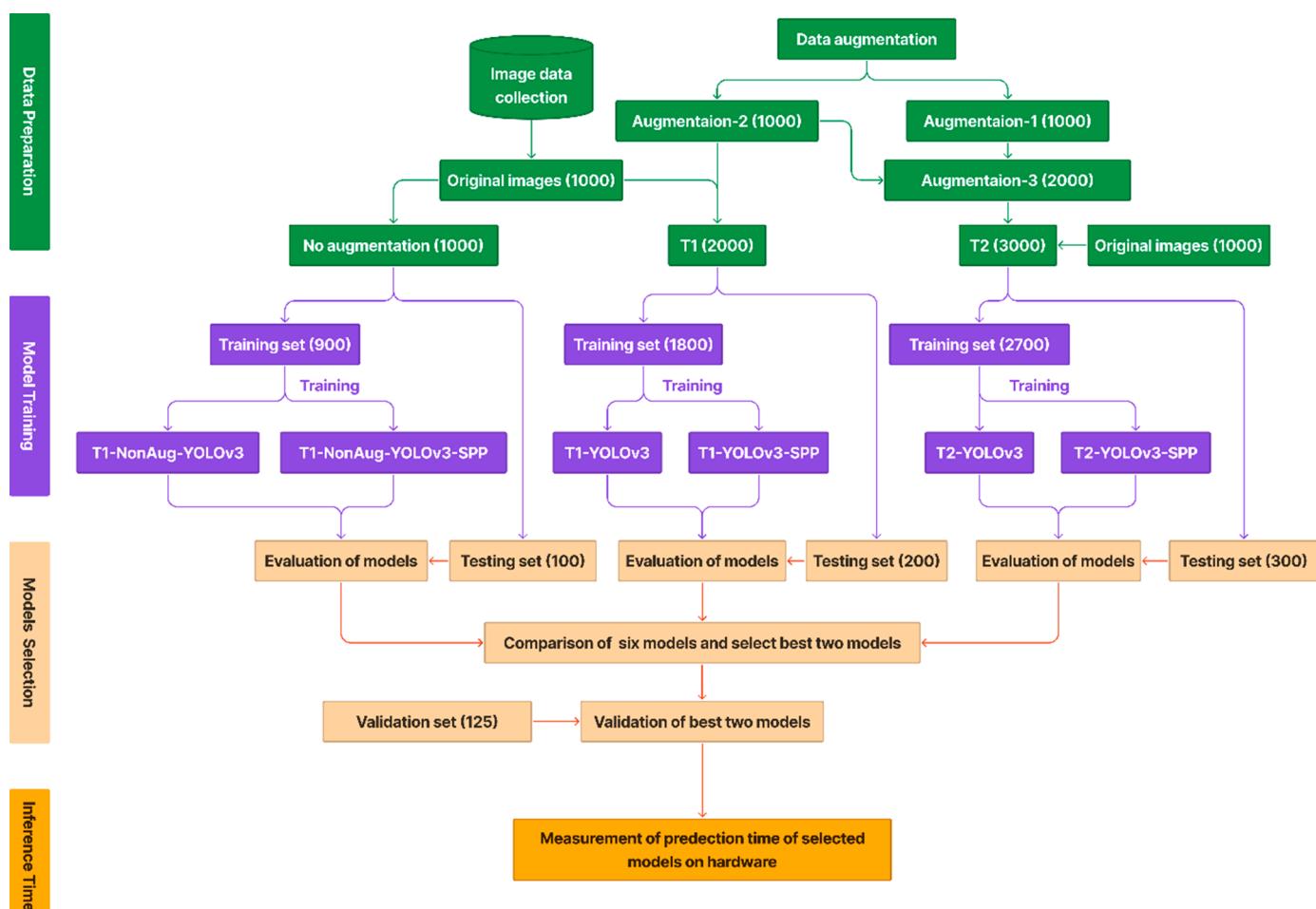


Fig. 5. Flowchart of the training and evaluation procedures used in this study.

Table 4

Precision, Recall, F1-score, and mAP of the YOLOv3 and YOLOv3-SPP, at 0.5 IoU and 0.25 confidence threshold.

Dataset	Models	Precision (%)	Recall (%)	F1-score (%)	mAP (%)
No augmentation	YOLOv3-SPP	83	55	66	68.49
	YOLOv3	85	54	66	67.75
T1	YOLOv3-SPP	87	53	66	73.03
	YOLOv3	90	46	61	71.26
T2	YOLOv3-SPP	86	54	66	74.38
	YOLOv3	89	47	62	72.87

Table 5

Class Average Precision (%) of YOLOv3 and YOLOv3-SPP on three different datasets.

Dataset	Models	Leaves (%)	Stems (%)	Green Berries (%)	Ripe Berries (%)	Dirt (%)
No augmentation	YOLOv3-SPP	72.00	56.14	74.90	79.18	60.24
	YOLOv3	69.22	55.66	72.02	80.56	61.31
T1	YOLOv3-SPP	77.6	67.05	72.82	83.58	64.10
	YOLOv3	74.15	63.8	72.26	82.91	63.19
T2	YOLOv3-SPP	81.11	74.55	70.61	81.99	63.65
	YOLOv3	77.40	65.31	75.54	82.51	63.61

Table 6

Comparisons of accuracies of the YOLOv3-SPP and YOLOv3 networks when trained and tested with different augmentation datasets.

Dataset	Models	Mean F1-score (%)	P-values	Mean mAP (%)	P-values
No augmentation	YOLOv3-SPP	66	0.565	69.12	0.702
	YOLOv3	65		68.57	
T1	YOLOv3-SPP	64	0.003	75.56	0.15
	YOLOv3	61		71.84	
T2	YOLOv3-SPP	65	<0.001	75.71	0.076
	YOLOv3	62		73.19	

(mAP) by 1.2% from original the YOLOv2 and achieved a higher detection speed [26]. However, Pham et al. [30] observed the sensitivity of YOLO networks for detecting various small objects (lower than 10 pixels) on aerial and satellite images. They also achieved equivalent mAP scores from YOLOv3 and YOLOv3-SPP networks on both aerial images (color images: YOLOv3: mAP = 73.11%, YOLOv3-SPP: mAP = 75.04%; infrared images: YOLOv3: mAP = 71.01%, YOLOv3-SPP: mAP = 73.70%, two class color images: YOLOv3: mAP = 97.87%, YOLOv3-SPP: mAP = 95.58%) and satellite images (YOLOv3: mAP = 78.93%, YOLOv3-SPP: mAP = 77.34%).

Researchers improved CNNs by optimizing hyperparameters resulting in saving computational costs [31,32]. However, the optimization of CNNs is challenging and generally done by weight initialization, stochastic gradient descent optimization, batch normalization, shortcut connections, and data augmentation [33]. Wang et al. [34] used YOLOv3 for monitoring the behavior of egg breeders in real-time. They configured the value of the subdivision and batch size of YOLOv3 manually and set the learning rate by observing the value loss function during training. The optimized YOLOv3 achieved an overall accuracy of 92.09% on the validation dataset. CNN optimization using data augmentation is an artificial process of enlarging a dataset for training a CNN with varied data. The augmentation techniques include image rotations, dataset partitioning, image cropping, scaling, transposing, and mirroring. These techniques were employed by several researchers for improving the overall learning procedure and performance [35,36]. Zhang et al. [36] developed a 13-layer CNN for classifying fruits and improved accuracy by applying different data augmentation techniques including image rotation, gamma correction, and noise injection on the training dataset. The CNN achieved an accuracy of 94.94% for fruit classification when the CNN was trained on an augmented dataset. Kang and Chen [37] applied image augmentations techniques including image flipping, color saturation, contrast, brightness adjustment, and translation to minimize the unbalanced distribution in the training data. A CNN-based recognition system was developed by Sladojevic et al. [35]

for classifying 13 different types of plant diseases. They observed that augmented images improved the performance of the model to 96.30% classification accuracy. Data augmentation techniques can also reduce the overfitting of models [38,39].

Deep learning networks require computational power for the processing of large datasets. This limitation in DL approaches influenced several researchers to investigate rich computing resources for solving various classification and detection problems in agriculture [7,40,41]. Liu et al. [28] compared tomato average detection time of YOLO-Tomato and Faster R-CNN using a computer configured with Intel (Santa Clara, CA, USA) i5, 64-bit 3.30 GHz quad-core central processing unit (CPU), and an NVIDIA (Santa Clara CA, USA) GeForce GTX 1070 Ti graphics processing unit (GPU). The author observed that YOLO-Tomato was faster than Faster R-CNN and took on average 54 milliseconds (ms) for detecting a single image. Tian et al. [8] also compared the average detection time of young, expanding, and ripe apples with YOLOv3 and Faster R-CNN. They observed that YOLOv3 detected 8.864 times faster than Faster R-CNN on an NVIDIA Tesla V100 server. Instead of using powerful computing resources, Liu & Wang [42] used an NVIDIA Jetson TX2 low-power embedded computer board for developing a real-time broken corn monitoring system addressing power and space limitations on a corn harvester. They implemented the YOLOv3-Tiny CNN and achieved 89.77% accuracy with 10 frames per second (FPS). Researchers also used YOLO-based CNNs on embedded hardware, NVIDIA Jetson TX2 and NVIDIA Jetson Nano, in different agricultural fields for weed detection [43], fish detection [44], and fruit detection [45] with high success rates. However, very few researchers implemented CNN on the central processing unit (CPU) based computers in agriculture. Quiroz and Alférrez [46] developed a legacy blueberries recognition system equipped with AMD Ryzen5 2500 U CPU-based computer, and the system achieved 86% accuracy on testing images. Kim et al. [47] experimented with 13 different DL object detection algorithms including YOLO on CPU and GPU and observed that a GPU accelerated image processing speed better than a CPU.

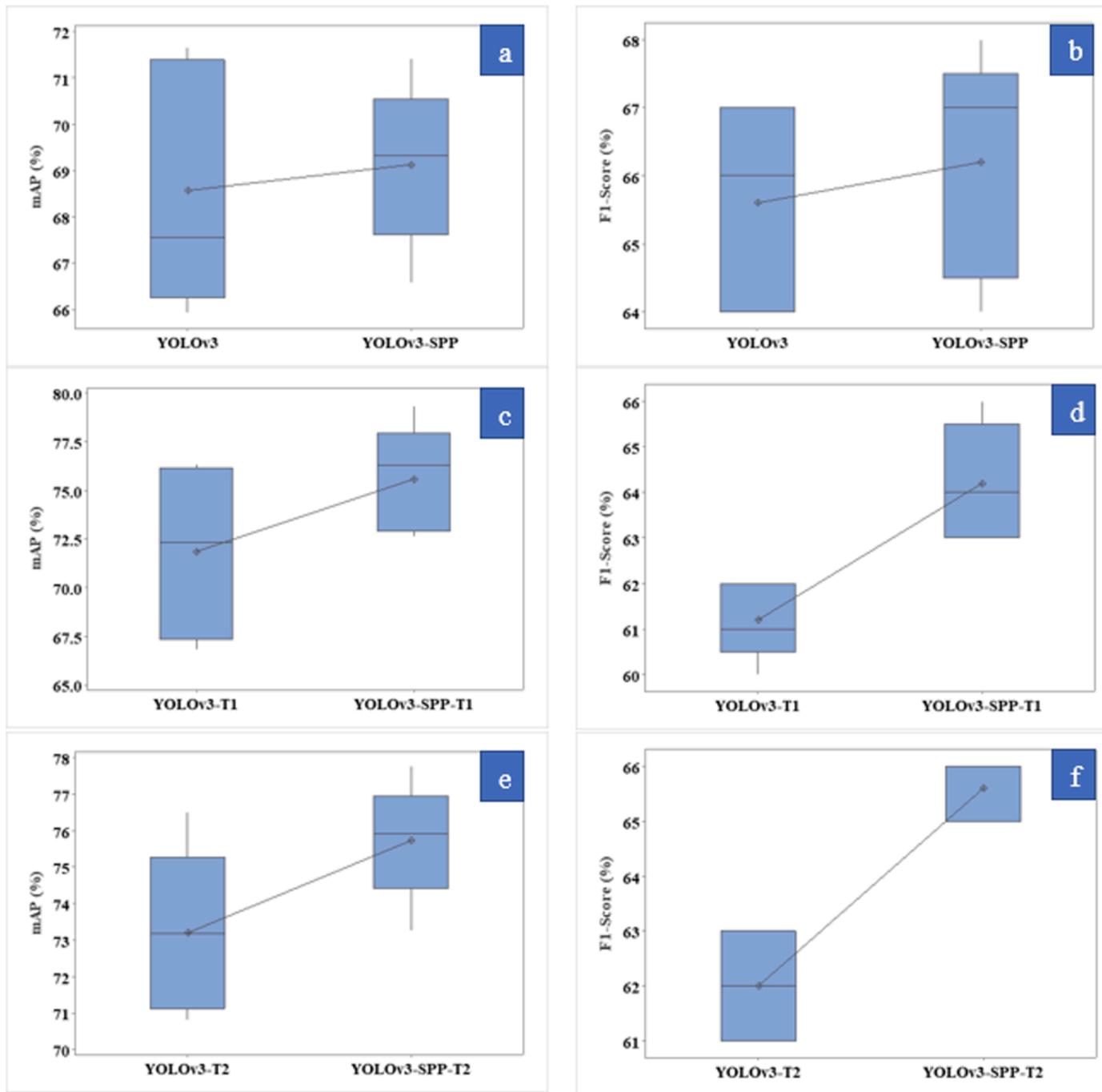


Fig. 6. Boxplot of YOLOv3 and YOLOv3-SPP on testing dataset of no augmentation (a,b), T1 (c,d), and T2 (e,f).

Modern wild blueberry harvesters were designed for mounting on an agricultural farm tractor. The modern harvesters, manufactured by Doug Bragg Enterprises, Ltd. (DBE), are tractor driven and operated by a hydraulic system that is controlled from the tractor cabin using an electric controller. The electric controller is powered by a 12-volt DC battery that can provide a maximum of 120 W of power. Thus, space and power limitations in the tractor cabin are a constraint without additional modifications and upgrades to the electrical system. Therefore, carefully selecting appropriate computer hardware will aid the commercial implementation of a real-time debris separation system for removing debris to improve fruit quality.

This study implements two CNNs (YOLOv3 and YOLOv3-SPP) for developing a machine vision system for a mechanical wild blueberry harvester to separate debris from ripe berries. The wild blueberry

industry faces fruit quality losses with mechanical harvesting due to current debris separating strategies/technologies that lack real-time feedback. The developed system can be implemented in real-time to alert operators of anomalies with collected berries. The developed machine vision system could be used as part of a feedback loop to precisely control the position and rotational speed of the debris-cleaning brush. The developed system could also be incorporated into commercial berry processing facilities for automatic detection and separation of leaves, stems, dirt, and green berries from ripe harvested berries to reduce manual labor efforts currently required. Feedback control can be added to further automate the debris cleaning brush to optimize debris removal. The system can also be incorporated to alert operators of clogged or damaged debris cleaning blower fans. Therefore, a CNN-based automated debris separation technology can be a solution for

Table 7
Comparisons of class AP between YOLOv3-SPP and YOLOv3.

Dataset	Classes	Models	Mean AP (%)	P-values
No augmentation	Leaves	YOLOv3-SPP	72.34	0.075
		YOLOv3	69.62	
	Stems	YOLOv3-SPP	57.48	0.881
		YOLOv3	58.53	
	Green Berries	YOLOv3-SPP	75.45	0.488
		YOLOv3	72.26	
	Ripe Berries	YOLOv3-SPP	79.57	0.545
		YOLOv3	80.71	
	Dirt	YOLOv3-SPP	60.8	0.627
		YOLOv3	61.7	
T1	Leaves	YOLOv3-SPP	81.84	<0.001
		YOLOv3	74.22	
	Stems	YOLOv3-SPP	75.77	0.292
		YOLOv3	65.69	
	Green Berries	YOLOv3-SPP	74.84	0.338
		YOLOv3	72.9	
	Ripe Berries	YOLOv3-SPP	83.32	0.762
		YOLOv3	83.05	
	Dirt	YOLOv3-SPP	62.03	0.567
		YOLOv3	63.36	
T2	Leaves	YOLOv3-SPP	80.19	0.096
		YOLOv3	77.46	
	Stems	YOLOv3-SPP	75.63	0.025
		YOLOv3	66.08	
	Green Berries	YOLOv3-SPP	74.89	0.854
		YOLOv3	74.14	
	Ripe Berries	YOLOv3-SPP	82.55	0.938
		YOLOv3	82.64	
	Dirt	YOLOv3-SPP	65.33	0.265
		YOLOv3	63.8	

the wild blueberry industry to improve the quality of the harvested fruit.

This paper aims to determine which CNN is more accurate for this task, what the minimum hardware requirements are for implementing a real-time debris detection system on the harvester, and whether the image complexity significantly affects the processing speed. Furthermore, this paper examines which image augmentation techniques are best suited for optimizing the accuracy of the two CNNs for this task. YOLOv3 was originally implemented on the common objects in context (COCO) dataset for the detection of 80 distinct types of objects. This study contributes by adding five new objects for detection including leaves, stems, green berries, ripe berries, and dirt, aiming to improve berry separation technology in mechanical wild blueberry harvesters. The architecture of the YOLO networks was kept the same, but the parameters in the configuration files of the networks, such as the learning rate, were changed to optimize them for training on the datasets in this study. Furthermore, the best-performing models were tested on four different types of computer hardware to investigate detection speeds and recommend a model and hardware for deploying a real-time debris detection system for a wild blueberry harvester.

2. Materials and methods

2.1. Description of equipment and computer systems

Two Logitech C920 webcams (Logitech International S.A., Lausanne, Switzerland) were used for video data collection. The cameras were both connected to the selected computers via USB 2.0 active extension cables. The first selected computer featured an Intel® Core™ i5-4300 U central processing unit (CPU) @ 1.90 GHz while the second utilized an Intel® Core™ i5-8250 U CPU @ 1.60 GHz (Hewlett-Packard, Palo Alto, CA, United States, and Dell Incorporation, Round Rock, Texas, United States). Both computers were operated with 64-bit Windows 10 (Microsoft Corp. Redmond, WA, United States). The first camera was mounted at the side conveyor and was pointed downward at a 45° angle to give a clear view of the conveyed berries and debris. The side camera angle was chosen at 45° to obtain a wide field of view on the side

conveyor (30.0 cm × 118.0 cm), which would allow the CNNs sufficient time to detect targets during operation. The angle was also used in several other studies for developing CNN-based fruit detection systems [48,49]. The camera height at the side conveyor was adjusted to 34.0 cm to encompass the complete length of the side conveyor. The rear camera was mounted at 49.8 cm working depth above the rear conveyor and pointed downward at a 90° angle (Fig. 1). The purpose of the side and rear cameras was to capture videos while conveying harvested berries from the side to the rear conveyor, respectively. Still frames were extracted from the videos and used for CNN training, testing, and validation. Images from both cameras (side and rear) were used in preparing the training, testing, and validation datasets.

2.2. Data collection

Video data with 1280 × 720 pixel resolution was gathered from two commercial wild blueberry fields in central Nova Scotia, Canada: Debert (45.4418°N, 63.4496°W) and East Mines (45.4271°N, 63.4819°W) from August 16 to August 26, 2019 (Fig. 2) following traditional harvesting scheduling of daytime operation in sunny conditions. The temperature ranged from 17.2 to 27.4 °C, relative humidity was 35–81%, and the wind speed was 2–24 km.h⁻¹ on August 16. On August 26, the temperature ranged from 13.3 to 17.7 °C, the relative humidity was 63–81% with a wind speed of 5–14 km.h⁻¹ [50]. Both fields were in their vegetative sprout year of the biennial crop production cycle in 2019. The fields had been commercially managed in previous years, with regular fertilization, weed and disease control, and biannual pruning by mowing. The videos were captured using the Logitech C920 webcams pointed at conveyors before and after the blower fan on a Doug Bragg Enterprises Ltd. (DBE) commercial mechanical wild blueberry harvester. Video frames were extracted at 1 s (s) intervals using video-to-image converter software (Free Video to JPG Converter, Ver.: 5.0.101.201). The videos were captured from twelve different plots in the two fields during harvesting. Plots of 0.91 m width (harvester head) and 100 m length were created using a measuring tape along the route of the harvester. The blurry and empty frames were removed manually after extraction, and all the extracted frames were renamed to prepare a parent dataset. The compiled dataset, consisting of 1000 images, was generated from the parent dataset.

2.3. Data augmentation

Data was augmented using a batch processing tool (IrfanView, Ver: 4.54) running on an Intel® Core™ i7-6700 K CPU @ 4.00 GHz Windows 7 64-bit machine (Microsoft Corp, Redmond, WA, USA). Five different color-based filters were used for generating augmentation datasets: gamma correction (0.1 to 6.99), sharpness (1 to 99), brightness (-255 to 255), contrast (-127 to 127), saturation (-255 to 255). The augmented datasets were named Augmentation-1, Augmentation-2, and Augmentation-3 (Table 1). The Augmentation-1 dataset contained five subsets each subset with 200 images. The five subsets were modified using five different gamma values (0.70, 0.80, 0.90, 1.10 and 1.20) (Table 1). Gamma correction increases or decreases the luminance value in an image [52]. The image variation was ensured by setting the gamma filter value to 0.70, 0.80, and 0.90 to obtain darker images and 1.10, 1.20, and 1.30 to obtain brighter images (Fig. 3). Similarly, the Augmentation-2 dataset also contained five subsets each containing 200 images. The subset images were enhanced by applying sharpness, brightness, contrast, gamma correction, and saturation filter values to 0.70, 0.80, 0.90, 1.10, and 1.20, respectively (Table 1). The sharpness enhancement improves the object edge in an image and renders them more sensitive to detection [53]. Image sharpness was increased by setting the filter value to 10 (Table 1). Brightness enhancement increases the pixel value of all channels of an image evenly [54]. The brightness of images was modified by setting the filter value to 20. Contrast

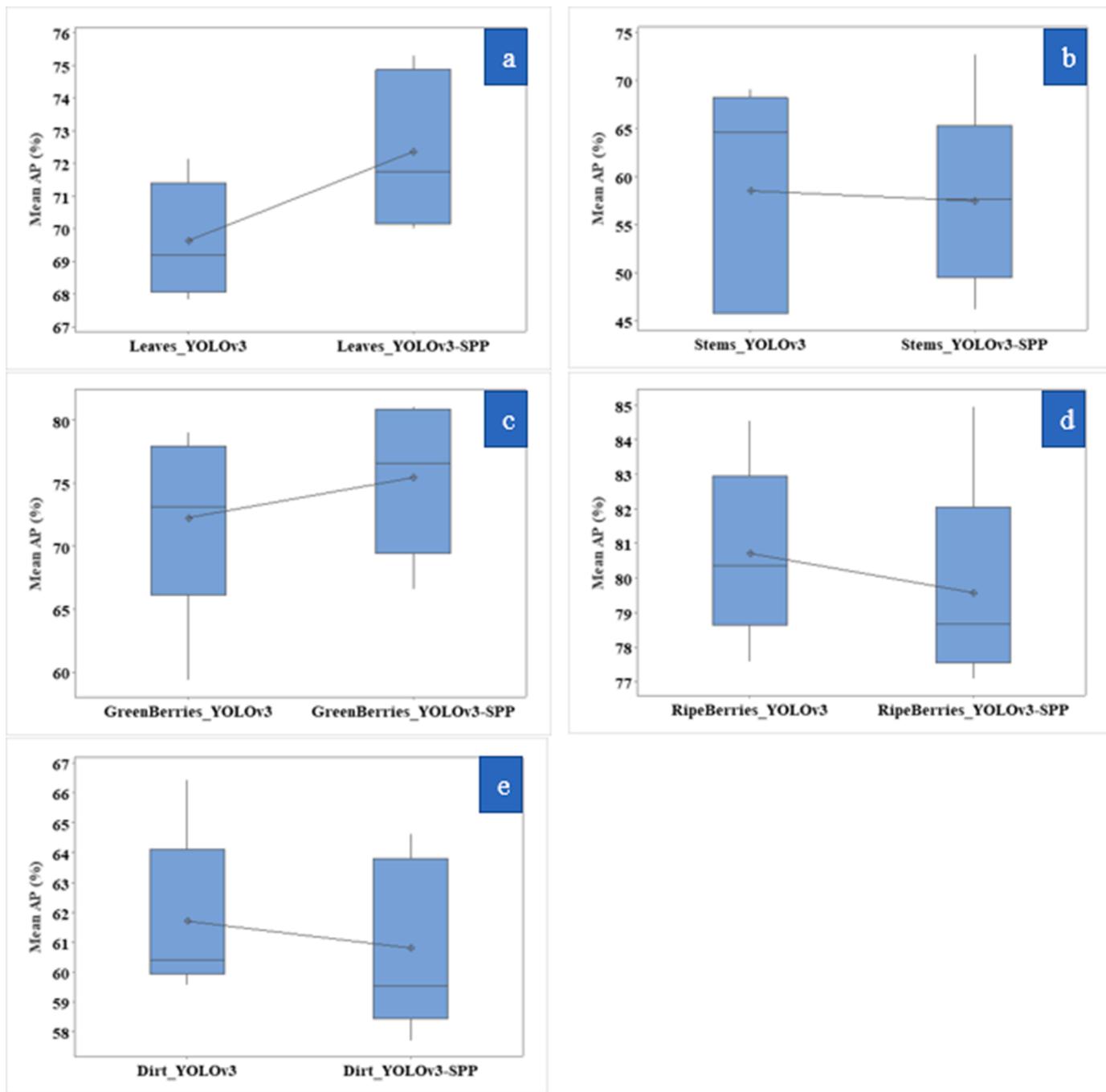


Fig. 7. Box plot of YOLOv3 and YOLOv3-SPP for leaves (a) stems (b) green berries (c) ripe berries (d) and dirt (e).

enhancement stretches or compresses the range of brightness value of an image and helps distinguish the target objects from the background [54]. The image contrast was enhanced by adopting a filter value of 20 as shown in (Table 1). The saturation enhancement intensifies the color of an image making it colorful and vibrant. Image saturation was enhanced by adopting a filter value of 15 (Table 1). The Augmentation-3 dataset was created by combining the Augmentation-1 and Augmentation-2 datasets.

The T1 dataset was prepared by mixing the experimental image dataset with the Augmentation-2 dataset. Similarly, the T2 dataset was prepared by mixing the experimental dataset with the Augmentation-1 and Augmentation-2 datasets (Table 2). Ninety percent of the images were used for training and 10% were used for testing (Table 2). Data were divided into training and testing sets of non-augmentation, T1, and

T2 datasets using a custom Python script. Images were listed sequentially, and the script allocated every tenth image for testing while leaving the remaining images for training. The objective of using a 9:1 ratio for training and testing was to ensure that the models were robust enough to recognize targets on unseen testing images with high accuracy by training with large training datasets. This ratio was also adopted by several researchers for CNN-based detections in various aspects of agriculture [8,55,56]. The testing dataset was separate from the training dataset and was not used during training. During splitting, augmented images were used in the testing dataset. The reason that augmented images were included for preparing the testing dataset was to investigate model performance in darker and lighter conditions. Furthermore, models were trained with both original and augmented images. The testing images with no augmentation, T1, and T2 were further divided

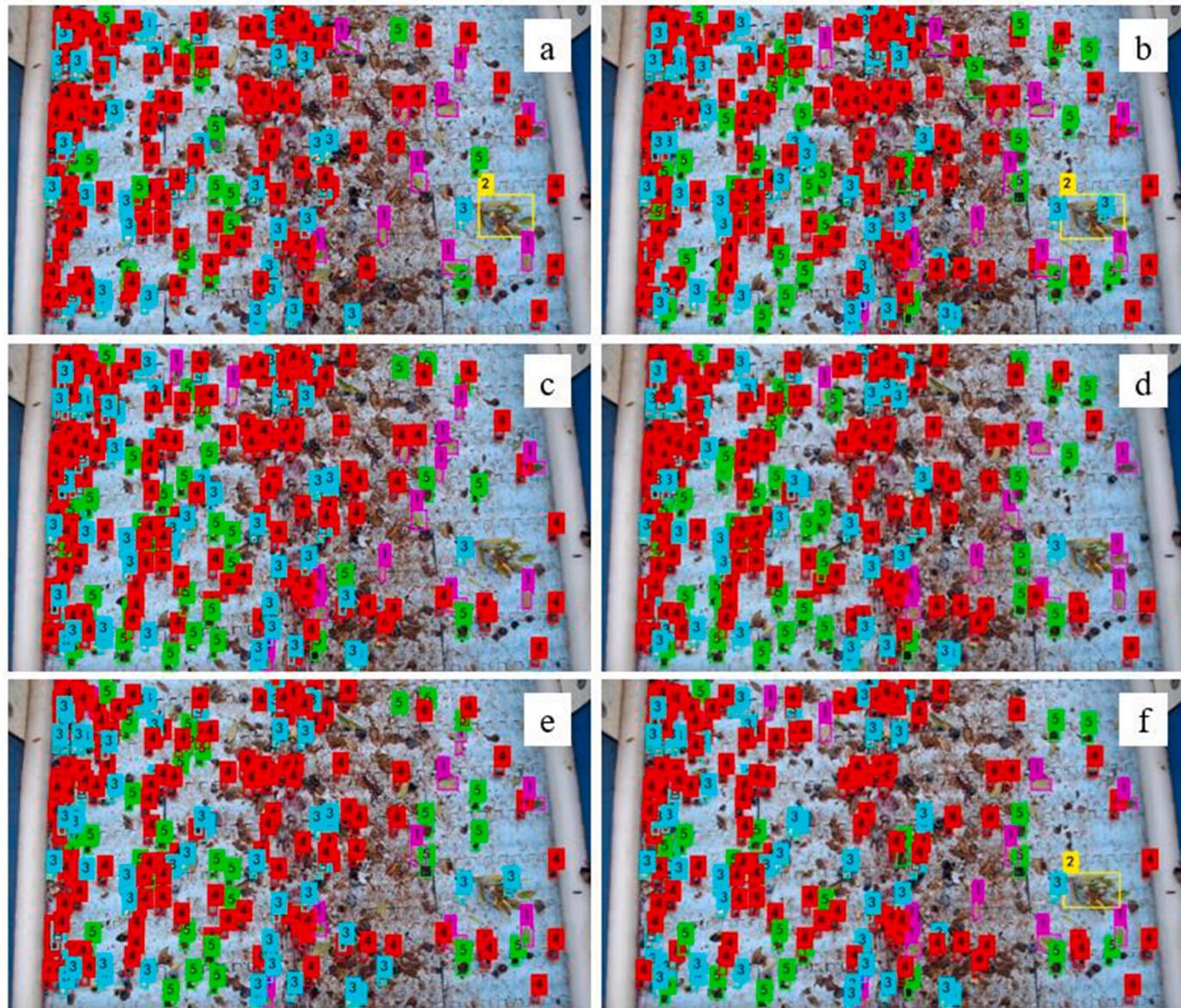


Fig. 8. Side by side comparison of detection results of YOLOv3 and YOLOv3-SPP on a testing image. (a), (c), and (e) represent detection of YOLOv3 on the testing images of no augmentation, T1, and T2 datasets, respectively. Similarly, (b), (d), and (f) represent the detection of YOLOv3-SPP on the testing images of no augmentation, T1 and T2 datasets, respectively. The detection of leaves, stems, green berries, ripe berries, and dirt on images were represented as 1, 2, 3, 4, and 5, respectively.

into five subsets and used for the evaluation and comparison of CNNs. One-way analysis of variance (ANOVA) was performed using Minitab 17 (Minitab Inc. NY, USA) with a 5% ($\alpha = 0.05$) level of significance.

A validation dataset containing a total of 125 images was also prepared and sorted based on the total number of objects labeled in an image. Images were categorized: simple, simple-moderate, moderate, and complex, with a total object count ranging from 0 to 49, 50 to 99, 100 to 199, 200 to 299, and 300 to 399, respectively. The number of labeled objects for each category was calculated using a bash script. A total of 1128, 1620, 3944, 6616, and 7865 labeled objects were found for simple, simple moderate, moderate, moderate complex, and complex categories, respectively. The simple images were mostly collected at the start of the harvesting process, which contributed to preparing the dataset considering different scenarios. Ripe berries, green berries, leaves, stems, and dirt were labelled based on classifiable appearance in the images (Fig. 4).

2.4. Training of CNNs (YOLOv3 and YOLOv3-SPP) on datasets

Two different YOLO networks (YOLOv3 [57], and YOLOv3-SPP (Z. [26])) were successfully trained and tested on the Darknet framework [57] using a GeForce RTX™ 2080 Ti @ 1665 MHz graphics processing unit (GPU) card in a computer installed with 64-bit Ubuntu 16.04 (Canonical Group Ltd, London, UK). Both networks were trained using the Darknet framework available at Github (<https://github.com/AlexeyAB/darknet>). The input image size was set to 1280×736 pixels to ensure the image features were properly represented and to ensure compatibility with the Darknet framework. Training batch size represented the number of images utilized per iteration was set to 64. The batch images were split into 16 mini-batches to reduce memory usage during training [58]. Max batches represents the number of total training iterations. According to Bochkovskiy [59], the appropriate max batches is determined by multiplying the total number of classes by 2000, the recommended number of training samples per class. Due to the smaller number of labelled samples per class in the datasets, more

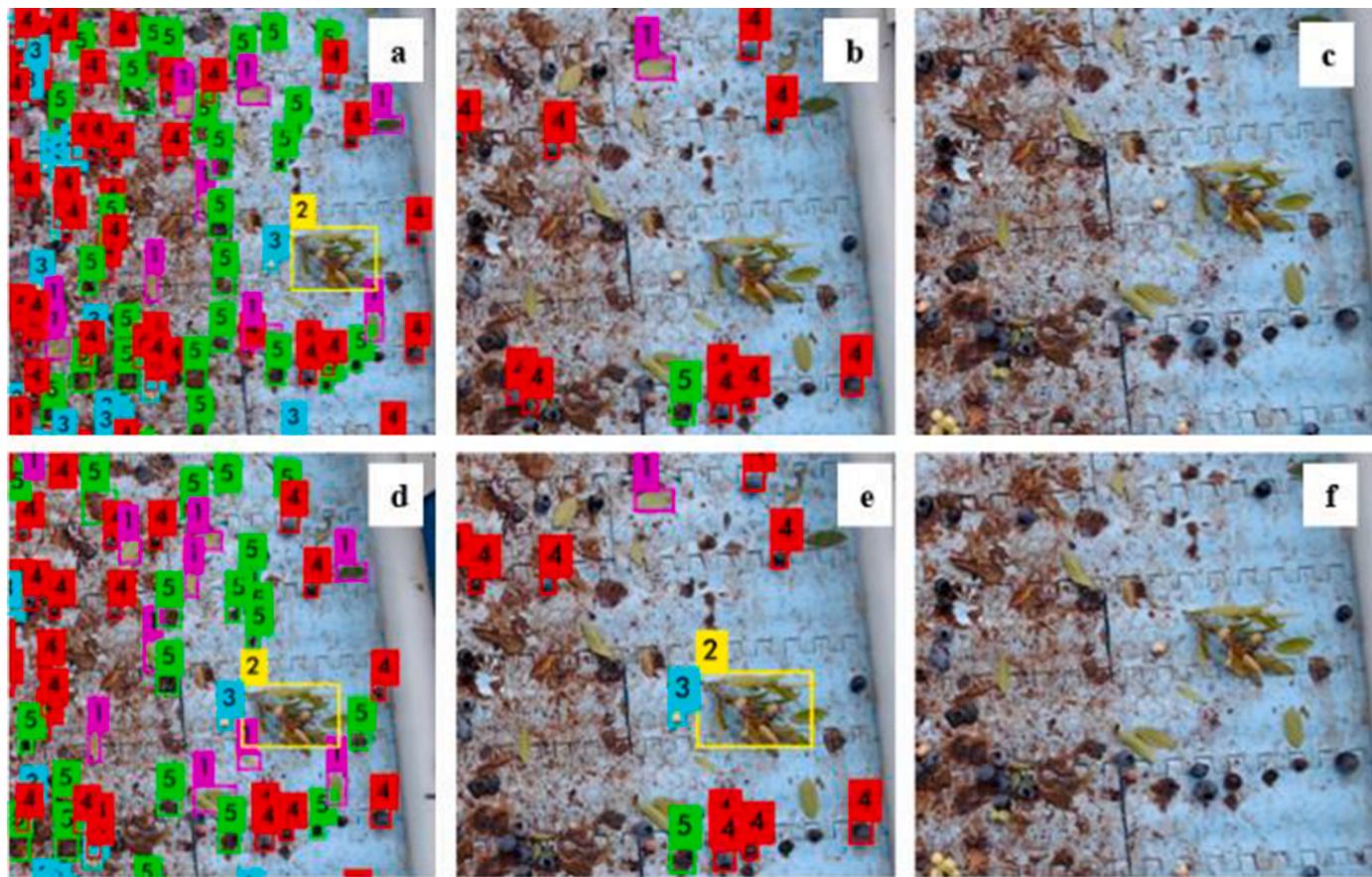


Fig. 9. Validation of models under different IoU and confidence thresholds. Detection of leaves, stems, green berries, ripe berries, and dirt on videos were represented as 1, 2, 3, 4, and 5, respectively. The detection result of Model -1 and Model-2 are shown in the first row and second row, respectively. (a) and (d) show detections at 0.10 IoU and 0.10 confidence threshold. (b) and (e) show detections at 0.50 IoU and 0.50 confidence threshold. (c) and (f) show detections at 0.90 IoU and 0.90 confidence threshold.

Table 8
Class Average Precision (%) of Model-1 and Model-2 on validation dataset.

Models	Leaves (%)	Stems (%)	Green Berries (%)	Ripe Berries (%)	Dirt (%)
Model-1	78.49	70.07	65.42	78.64	62.86
Model-2	79.48	81.33	65.20	75.91	59.89

than 10,000 iterations ($5 \text{ classes} \times 2000$) may be required. The YOLO networks were trained up to 12,000 iterations to ensure convergence. The initial learning rate was set at 0.001 and then decreased to 0.0002,

0.0001, and 0.0001 by multiplying with three different scales (0.20, 0.10, 0.10) at 7000th, 8000th, and 9000th iterations. The reason for decreasing the learning rate was to help minimize the training loss [60]. The other training parameters of the CNNs were kept as the default in the configuration files.

2.5. Evaluation the CNNs

The detection performance of CNNs was evaluated using precision, recall, mAP, and F1-Score values. True positive (TP) is the number of correct detections of leaves, stems, ripe berries, green berries, and dirt. False positive (FP) is the number of incorrect detections of leaves, stems, ripe berries, green berries, and dirt. False negative (FN) is the number

Table 9
Evaluation of Model-1 under different IoU thresholds.

Thresholds	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
AP _{Leaves}	86.02	85.82	85.01	83.17	78.49	63.75	34.56	6.64	0.09
AP _{Stems}	85.2	84.57	84.14	78.21	70.07	45.81	20.29	4.44	0.38
AP _{Green Berries}	88.89	88.44	86.19	80.22	65.42	35.72	13.83	2.02	0.04
AP _{Ripe Berries}	90.79	90.63	90.19	88.34	78.64	55.5	29.5	6.83	0.11
AP _{Dirt}	78.6	78.22	77.24	72.79	62.86	42.82	18.35	2.37	0.01
mAP	85.9	85.54	84.55	80.55	71.1	48.72	23.31	4.46	0.13
True positive	12,284	12,271	12,216	12,019	11,246	9413	6623	2872	311
False positive	1392	1405	1460	1657	2377	4263	7053	10,804	13,365
False negative	8927	8940	8995	9192	9965	11,798	14,588	18,339	20,900
Precision	90	90	89	88	83	69	48	21	2
Recall	58	58	58	57	53	44	31	14	1
F-score	70	70	70	69	65	54	38	16	2

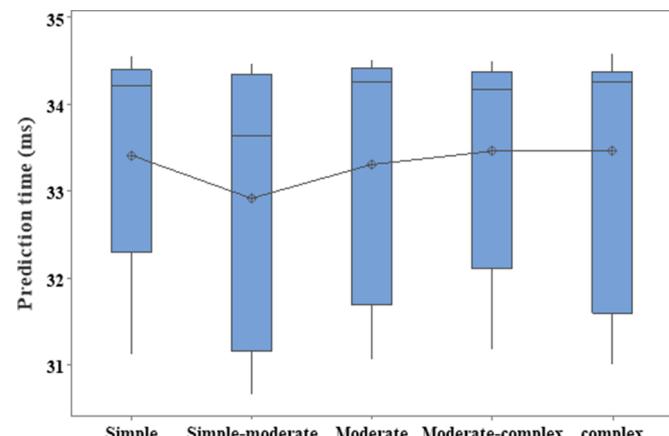
Table 10

Evaluation of Model-2 under different IoU thresholds.

Thresholds	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
APLeaves	86.57	86.26	85.58	84.05	79.48	65.38	36.16	5.99	0.12
APStems	89.98	89.53	88.3	85.87	81.33	53.78	16.61	1.56	0.02
APGreen Berries	87.7	87.22	84.93	79.89	65.2	37.25	13.62	1.93	0.04
APRipe Berries	90.38	90.19	89.66	87.25	75.91	52.68	28.94	7.06	0.16
APDirt	75.85	75.51	74.25	69.54	59.89	41.27	16.86	2.11	0.02
mAP	86.1	85.74	84.54	81.32	72.36	50.07	22.44	3.73	0.07
True positive	11,136	11,129	11,072	10,880	10,181	8474	6040	2645	281
False positive	1124	1131	1188	1380	2110	3786	6220	9615	11,979
False negative	10,075	10,082	10,139	10,331	11,030	12,737	15,171	18,566	20,930
Precision	91	91	90	89	83	69	49	22	2
Recall	53	52	52	51	48	40	28	12	1
F-score	67	66	66	65	61	51	36	16	2

Table 11Results of MMC using the Tukey method to identify significant differences in models and hardware at a 5% ($\alpha = 0.05$) level of significance.

Hardware*Model	Mean Prediction Time (ms)	Frame Rate (FPS)
Hardware-1 Model-1	12,126.90 A	0.08
Hardware-1 Model-2	12,056.40 A	0.08
Hardware-2 Model-2	3625.60 B	0.28
Hardware-2 Model-1	3608.60 B	0.28
Hardware-3 Model-2	868.70 C	1.15
Hardware-3 Model-1	831.20 C	1.2
Hardware-4 Model-1	34.10 D	29.33
Hardware-4 Model-2	33.30 D	30.03

**Fig. 10.** Boxplot on prediction time of categorical images.

where there are leaves, stems, ripe berries, green berries, and dirt on the images but the model does not find the objects. True positive (TP) is the number of correct detections of leaves, stems, ripe berries, green berries, and dirt. False positive (FP) is the number of incorrect detections of leaves, stems, ripe berries, green berries, and dirt. False negative (FN) is the number where there are leaves, stems, ripe berries, green berries, and dirt on the images but the model does not find the objects. Precision is calculated as the ratio of true detections of target classes to the identified detections and is calculated using Eq. (1). Recall is the ratio of true detection of target classes among all the ground truth detections and is calculated using Eq. (2). F1-Score is the harmonic mean of precision and recall and is calculated using Eq. (3). The highest precision at eleven levels of recall was used to determine the AP score (Eq. (4)). Mean average precision (mAP) is the average of average precision values of all target classes. The intersection over union (IoU) value confirms whether the detection of leaves, stems, ripe berries, green berries, and dirt were true positive or false positive (Eq. (5)). When predicted bounding boxes have an IoU greater than the given IoU threshold,

prediction is considered a true positive otherwise, false positive. The intersection over union can be calculated using Eq. (5) where B1 and B2 represent the ground truth bounding box and predicted bounding box, respectively. Total of nine different IoU thresholds (0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, and 0.90) were used to validate CNNs. The difference between IoU of 0.3 and IoU of 0.5 is negligible to the average person [61]. However, some studies were conducted for CNN-based detections using even lower IoU thresholds [62,63,64]. According to Bargoti & Underwood [62], IoU greater than 0.2 resulted in better small fruit detection, but IoU greater than 0.5 resulted in detection errors. Due to the relatively small fruit size, [63] chose an IoU of 0.4.

$$Precision = \frac{TP}{TP + FP} \times 100 \quad (1)$$

$$Recall = \frac{TP}{TP + FN} \times 100 \quad (2)$$

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \times 100 \quad (3)$$

$$AP = \frac{1}{11} \sum_{Re \in \{0.0, 0.1, \dots, 1\}} \max_{\widetilde{Re}: Re \geq \widetilde{Re}} \Pr(\widetilde{Re}) \quad (4)$$

$$IOU = \frac{B_1 \cap B_2}{B_1 \cup B_2} \quad (5)$$

2.6. Measurement of prediction time of selected models on hardware

Two CNNs (YOLOv3 and YOLOv3-SPP) were trained and tested on three datasets (no augmentation, T1, and T2) (Fig. 5). A total of six CNN models were generated and best performing models (Model-1, Model-2) were selected based on mAP and F1-Score. To test the detection speed, the two models were deployed on four different configured hardware platforms (Hardware-1, Hardware-2, Hardware-3, and Hardware-4) as shown in Table 5. The two computers described as Hardware-1 and Hardware-2 are configured with Intel Core i5-4300 U and Intel Core i7-6700 K processor processors running at 1.90 GHz and 4.00 GHz respectively. A GPU-based, ARM Cortex-A57 (quadcore) embedded computer (Nvidia Jetson TX2) @ 2.00 GHz is referred to as Hardware-3. Similarly, a desktop GPU computer, equipped with an Intel Core i9-7900X CPU @ 3.30 GHz CPU and an Nvidia RTX 2080 Ti @ 1665 MHz GPU was referred to as Hardware-4. Only the C++ version of OpenCV was installed on Hardware-1 and Hardware-2 to use CPU power for network inference and recognition on images. Hardware-3 was flashed using Nvidia Jetpack version 3.0 to enable Compute Unified Device Architecture (CUDA version 8.0) for loading GPU-related files and dependencies. Similarly, CUDA version 10.2 was installed on Hardware-4. The GPU and CUDA of Hardware-3 and Hardware-4 were activated by enabling GPU, CUDNN, CUDNN—HALF, and disabling OPENCV, AVX, and OPENMP in the makefile of Darknet (Table 3). The configuration file of YOLOv3 was kept the same as during training

except the batch size was changed to 1 for testing to use less memory and achieve real-time performance [65]. A 4×2 factorial design ($\alpha = 0.05$) with the four hardware configurations and two CNN models was used to check the effect of hardware and models on object prediction time. A total of eight different treatments (Hardware-1 \times Model-1, Hardware-1 \times Model-2, Hardware-2 \times Model-2, Hardware-2 \times Model-1, Hardware-3 \times Model-2, Hardware-3 \times Model-1, Hardware-4 \times Model-1, Hardware-4 \times Model-2) were prepared where each treatment was replicated five times. Data were analyzed using Minitab 17 (Minitab Inc. NY, USA) statistical software. Tukey's multiple mean comparisons (MMC) were performed to determine the difference in means.

3. Results and discussion

3.1. Evaluation of models

Models were evaluated at 0.5 IoU and 0.25 confidence threshold. YOLOv3-SPP achieved comparatively better recall and mAP than YOLOv3 on no augmentation, T1, and T2 datasets, respectively (Table 4). However, YOLOv3 achieved comparatively lower recall on all three datasets (Table 4). The lower recall value of YOLOv3 indicates missed detections of some true positives in images. The recall of YOLOv3-SPP increased by 1%, 7%, and 7% from the recall of YOLOv3 in no augmentation, T1, and T2 datasets, respectively (Table 4). The AP of leaves, stems, ripe berries, and dirt was improved from the no augmentation dataset by 5.60%, 10.91%, 4.40%, 3.86%, and 9.11%, 18.41%, 2.81%, 3.41% when YOLOv3-SPP trained and tested on T1 and T2 datasets, respectively. The AP of leaves, stems, green berries, ripe berries, and dirt was improved by 4.93%, 8.14%, 0.24%, 2.35%, 1.88% and 8.18%, 9.65%, 3.52%, 1.95%, 2.30%, respectively, when YOLOv3 trained and tested on T1 and T2 dataset, respectively (Table 5).

3.2. Comparison of YOLO detection results

YOLOv3-SPP outperformed YOLOv3 in all three datasets with comparatively higher F1-scores (66%, 64% and 65%) and mAP (69.12%, 75.56% and 75.71%) (Table 6, Fig. 6). Additionally, YOLOv3-SPP achieved significantly higher F1-scores than YOLOv3 on T1 and T2 datasets (Table 6). With the T1 and T2 datasets, YOLOv3-SPP resulted in significantly higher mAP than YOLOv3 for the leaves class (<0.001) and the stems class ($P = 0.025$), respectively (Table 7). It also achieved comparatively higher AP than YOLOv3 for leaves ($P = 0.075$) and green berries ($P = 0.488$) on the no augmentation dataset. Similarly, classes of YOLOv3-SPP including stems ($P = 0.292$), green berries ($P = 0.338$) and ripe berries ($P = 0.762$) achieved comparatively better mAP on the T1 dataset than YOLOv3. In the T2 dataset, leaves ($P = 0.096$), green berries ($P = 0.854$), and dirt ($P = 0.265$) classes of YOLOv3-SPP also reached comparatively higher mAP than YOLOv3. Results found that YOLOv3 and YOLOv3-SPP achieved similar mAP in select cases for ripe berries and dirt detection (Fig. 7). In the no augmentation dataset, YOLOv3 and YOLOv3-SPP yielded AP of 79.57% and 80.71% for ripe berries and 60.80% and 61.70% for dirt, respectively. Similar detection results for ripe berries and dirt were also observed in T1 and T2 datasets (Fig. 8). The most likely reason for yielding similar results could be the sensitivity to small object detection of YOLO networks. All the target classes were labelled based on visual appearance where stems with attached leaves were labelled as stems instead of their own class. This strategy of labeling may have led to slight errors in classification. However, the accuracies of models can be enhanced by labeling leaves within stems and checking confusion matrices during testing (Fig. 9).

3.3. Validation of selected models under different IoU thresholds

Model-1 (YOLOv3-SPP; mAP: 73.03%) achieved 83% precision, 53% recall, 65% F-score, and 71.10% mAP, while Model-2 (YOLOv3-SPP, mAP: 74.38%) achieved 48% recall, 61% F-score, and 72.36% mAP on

the validation dataset. Each target class except the dirt achieved comparatively lower AP (59.89%) during Model-2 validation than Model-1 validation (Table 8).

Results showed that with the increase of the IoU threshold from 0.10 to 0.90, AP values decreased for all target classes (Tables 9 and 10). The maximum precision, recall, F1-score, and mAP were achieved at the 0.10 IoU threshold. Conversely, false positive and false negative detections significantly increased with the increase of the threshold from 0.10 to 0.90 which led to decreased performance of the model as well as detection of leaves, stems, green berries, ripe berries, and dirt in an image (Tables 9 and 10). Model-1 yielded the highest mAP (89.50%) at the 0.10 threshold and the lowest mAP (0.13%) at a threshold of 0.90 (Table 9). The number of true positives decreased from 12,284 to 311 when the threshold changed from 0.10 to 0.90 (Table 9). Conversely, the number of false positives and false negatives increased from 1392 to 13,365 and 8927 to 20,900 when the thresholds were changed from 0.10 to 0.90 (Table 9). Model-2 also achieved the highest mAP (86.10%) at the threshold of 0.10 and gradually decreased when the threshold increased up to 0.90. Model-2 detected a total of 11,136, 1124, and 10,075 true positives, false positives, and false negatives, respectively at the threshold of 0.10 (Table 10).

3.4. Statistical analysis for determining the effect of models and hardware on prediction time

The result of the ANOVA indicated that the Hardware (<0.001) and Hardware \times Models (0.015) had a significant effect on processing time. Models alone had no contribution to processing time ($p = 0.724$). Tukey's MMC was performed on the interaction factor (Hardware \times Models) only because the interaction effect was significant. The interaction of Hardware-1 and Model-1 resulted in the highest mean (12,126.90 ms) and the interaction of Hardware-4 and Model-2 yielded the lowest mean (33.30 ms) as shown in Table 11. Table 11 also showed that the processing speed of a single image on the CPU was slower than on the GPU. DL models performed faster in GPUs (Hardware-3, Hardware-4) than in CPUs (Hardware-1, Hardware-2) because GPUs have high bandwidth and parallel computing capabilities [47]. The combination of Hardware-4 and Model-2 resulted in the highest frame rate (30.03 FPS) while Hardware-1 provided the lowest frame rate (0.80 fps) with both models (Table 11). GPU-based hardware resulted in a higher frame rate because the GPU can process an image faster than the CPU. Since Model-2 achieved a comparatively lower prediction time on Hardware-4, an ANOVA was performed to investigate the difference in prediction time on image categories using Model-2 and Hardware-4. The simple, simple-moderate, moderate, and complex images took 33.409, 32.912, 33.301, 33.454, and 33.466 ms, respectively. The ANOVA result showed no significant inference time on image categories ($p = 0.587$). Tukey's method found that mean inference times on image categories were not significantly different. However, the complex image took a slightly higher prediction time (33.466 ms) than other image categories. A simple moderate image resulted in a comparatively lower prediction time (32.912 ms) (Fig. 10).

4. Conclusion

Accurate debris detection is a major challenge during wild blueberry harvesting. An automatic debris detection system was developed to separate debris from berries to improve fruit quality. The system consisted of a camera vision system and a data processing unit. The YOLOv3 and YOLOv3-SPP CNNs were trained at 1280×736 input resolution to detect debris and fruit in images of wild blueberry harvester conveyors. Three image datasets using different degrees of augmentation were used to train the CNNs. YOLOv3-SPP produced higher mAP scores (74.38%, 73.03%, and 68.49%) than YOLOv3 (72.78%, 71.26%, and 67.75%) in all three datasets. Training the YOLOv3-SPP and YOLOv3 using non-augmented images produced lower mAP scores (68.49%, 67.75%)

than training with a dataset containing five equally weighted augmentations (73.03%, 71.26%) and a dataset with augmentations weighted toward gamma (74.38%, 72.28%). This indicates that dataset augmentation is necessary to achieve optimal accuracy for this application. Gamma augmentation in particular helped boost the accuracy of both CNNs. Lowering the IoU threshold helped improved mAP scores but did not have a substantial effect on F-scores at thresholds below 0.3. A desktop computer with an Nvidia GeForce RTX 2080 Ti GPU running YOLOv3-SPP processed images in 33.30 ms, which would be suitable for real-time debris detection. Conversely, the other hardware systems were unsuitable for real-time processing with processing times exceeding 33.30 ms. The number of objects detected in an image has no significant influence on the processing speed. The accuracy and speed of smaller neural network architectures and lower resolutions should be tested on different hardware systems to provide more options for real-time debris detection.

Funding

This research was financially supported by Doug Bragg Enterprises (DBE) Ltd, the Natural Sciences and Engineering Research Council (NSERC) of Canada, and the New Brunswick Canadian Agricultural Partnership (CAP).

CRediT authorship contribution statement

Anup Kumar Das: Conceptualization, Methodology, Validation, Formal analysis, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Travis J. Esau:** Conceptualization, Methodology, Investigation, Supervision, Project administration, Funding acquisition, Writing – review & editing. **Qamar U. Zaman:** Conceptualization, Investigation, Resources, Writing – review & editing, Supervision, Project administration, Funding acquisition. **Aitazaz A. Farooque:** Investigation, Writing – review & editing. **Arnold W. Schumann:** Conceptualization, Investigation, Writing – review & editing. **Patrick J. Hennessy:** Writing – review & editing.

Declaration of Competing Interest

The authors declare no conflict of interest.

Data availability

Data will be made available on request.

Acknowledgments

The authors would like to thank Craig MacEachern (Research Assistant), Derrick Ouma, Jack Lynds, and Connor Mullins (summer students) at Dalhousie University for helping during the data collection and the wild blueberry growers for use of their fields. Also, the authors would like to acknowledge the efforts of the Mechanized Systems and Precision Agriculture Research Teams at Dalhousie's Faculty of Agriculture.

References

- [1] PMRA. (2005). Crop profile for wild blueberry in canada. Annual report. Pesticide risk reduction program. *Agri. And agri-food canada*. Available at: http://www4.agr.gc.ca/resources/prod/doc/prog/prpp/pdf/blueberry_e.pdf.
- [2] G. Kinsman, The History of the Lowbush Blueberry Industry in Nova Scotia 1950–1990, Nova Scotia Dept. of Agriculture & Marketing, 1993.
- [3] C.D. Kay, B.J. Holub, The effect of wild blueberry (*vaccinium angustifolium*) consumption on postprandial serum antioxidant status in human subjects, *Br. J. Nutr.* 88 (4) (2002) 389–397.
- [4] V. Lobo, A. Patil, A. Phatak, N. Chandra, Free radicals, antioxidants and functional foods: impact on human health, *Pharmacogn. Rev.* 4 (8) (2010) 118.
- [5] D. Yarborough, Establishment and management of the cultivated lowbush blueberry (*vaccinium angustifolium*), *Int. J. Fruit Sci.* 12 (1–3) (2012) 14–22.
- [6] K. Esau, T. Esau, Q. Zaman, A. Farooque, A. Schumann, Effective use of a variable speed blower fan on a mechanical wild blueberry harvester, *Appl. Eng. Agric.* 34 (5) (2018) 831–840.
- [7] A. Koirala, K. Walsh, Z. Wang, C. McCarthy, Deep learning for real-time fruit detection and orchard fruit load estimation: benchmarking of “MangoYOLO.”, *Precis. Agric.* 20 (6) (2019) 1107–1135.
- [8] Y. Tian, G. Yang, Z. Wang, H. Wang, E. Li, Z. Liang, Apple detection during different growth stages in orchards using the improved YOLO-V3 model, *Comput. Electron. Agric.* 157 (2019) 417–426.
- [9] Q. Zaman, D. Percival, R. Gordon, A. Schumann, et al., Estimation of wild blueberry fruit yield using digital color photography, *Acta Hortic.* 824 (2009) 57–65.
- [10] Y. Chang, Q. Zaman, A. Schumann, D. Percival, T. Esau, G. Ayalew, Development of color co-occurrence matrix based machine vision algorithms for wild blueberry fields, *Appl. Eng. Agric.* 28 (3) (2012) 315–323.
- [11] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, *Nature* 521 (7553) (2015) 436–444.
- [12] D.P. Cavallo, M. Cefola, B. Pace, A.F. Logrieco, G. Attolico, Non-destructive automatic quality evaluation of fresh-cut iceberg lettuce through packaging material, *J. Food Eng.* 223 (2018) 46–52.
- [13] A. Jahanbakhshi, M. Momeny, M. Mahmoudi, Y.D. Zhang, Classification of sour lemons based on apparent defects using stochastic pooling mechanism in deep convolutional neural networks, *Sci. Hortic.* 263 (2020), 109133.
- [14] Z. Wang, M. Hu, G. Zhai, Application of deep learning architectures for accurate and rapid detection of internal mechanical damage of blueberry using hyperspectral transmittance data, *Sensors* 18 (4) (2018) 1126.
- [15] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
- [16] S. Xie, R. Girshick, P. Dollár, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1492–1500.
- [17] X. Ni, C. Li, H. Jiang, F. Takeda, Deep learning image segmentation and extraction of blueberry fruit traits associated with harvestability and yield, *Hortic. Res.* 7 (2020). <https://doi.org/10.1038/s41438-020-0323-3>.
- [18] S. Gonzalez, C. Arellano, J.E. Tapia, Deepblueberry: quantification of blueberries in the wild using instance segmentation, *IEEE Access* 7 (2019) 105776–105788.
- [19] S. Qiao, Q. Wang, J. Zhang, Z. Pei, Detection and classification of early decay on blueberry based on improved deep residual 3D convolutional neural network in hyperspectral images, *Sci. Program.* 2020 (2020) 8895875. <https://doi.org/10.1155/2020/8895875>.
- [20] S. Fan, C. Li, W. Huang, L. Chen, Data fusion of two hyperspectral imaging systems with complementary spectral sensing ranges for blueberry bruising detection, *Sensors* 18 (12) (2018) 4463.
- [21] Y. Jiang, C. Li, F. Takeda, Nondestructive detection and quantification of blueberry bruising using near-infrared (NIR) hyperspectral reflectance imaging, *Sci. Rep.* 6 (1) (2016) 1–14.
- [22] J. Kuzy, Y. Jiang, C. Li, Blueberry bruise detection by pulsed thermographic imaging, *Postharvest Biol. Technol.* 136 (2018) 166–177.
- [23] H. Li, W.S. Lee, K. Wang, Identifying blueberry fruit of different growth stages using natural outdoor color images, *Comput. Electron. Agric.* 106 (2014) 91–101.
- [24] J. Redmon, A. Farhadi, YOLO9000: better, faster, stronger, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 7263–7271.
- [25] Redmon, J., & Farhadi, A. (2018). Yolov3: an incremental improvement. *arXiv*. <https://doi.org/10.48550/arXiv.1804.02767>.
- [26] Z. Huang, J. Wang, X. Fu, T. Yu, Y. Guo, R. Wang, DC-SPP-YOLO: dense connection and spatial pyramid pooling based YOLO for object detection, *Inf. Sci.* 522 (2020) 241–258 (Ny).
- [27] C.B. MacEachern, T.J. Esau, A.W. Schumann, P.J. Hennessy, Q.U. Zaman, Deep learning artificial neural networks for detection of fruit maturity stage in wild blueberries, in: Proceedings of the ASABE Annual International Virtual Meeting, 2021, p. 1.
- [28] G. Liu, J.C. Nouaze, P.L. Touko Mbouembe, J.H. Kim, YOLO-tomato: a robust algorithm for tomato detection based on YOLOv3, *Sensors* 20 (7) (2020) 2145.
- [29] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700–4708.
- [30] M.T. Pham, L. Courtrai, C. Friguet, S. Lefevre, A. Baussard, YOLO-fine: one-stage detector of small objects under various backgrounds in remote sensing images, *Remote Sens.* 12 (15) (2020) 2501 (Basel).
- [31] S. Lu, B. Wang, H. Wang, L. Chen, M. Linjian, X. Zhang, A real-time object detection algorithm for video, *Comput. Electr. Eng.* 77 (2019) 398–408.
- [32] Shafiee, M.J., Chywl, B., Li, F., & Wong, A. (2017). Fast YOLO: a fast you only look once system for real-time embedded object detection in video. *arXiv*. <https://doi.org/10.48550/ARXIV.1709.05943>.
- [33] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroody, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, et al., Recent advances in convolutional neural networks, *Pattern Recognit.* 77 (2018) 354–377.
- [34] J. Wang, N. Wang, L. Li, Z. Ren, Real-time behavior detection and judgment of egg breeders based on YOLO v3, *Neural Comput. Appl.* 32 (10) (2020) 5471–5481.
- [35] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, D. Stefanovic, Deep neural networks based recognition of plant diseases by leaf image classification, *Comput. Intell. Neurosci.* (2016) 3289801. <https://doi.org/10.1155/2016/3289801>.

- [36] Y.D. Zhang, Z. Dong, X. Chen, W. Jia, S. Du, K. Muhammad, S.H. Wang, Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation, *Multimed. Tools Appl.* 78 (3) (2019) 3613–3632.
- [37] H. Kang, C. Chen, Fruit detection and segmentation for apple harvesting using visual sensor in orchards, *Sensors* 19 (20) (2019) 4599.
- [38] A. Namozov, Y. Im Cho, An efficient deep learning algorithm for fire and smoke detection with limited data, *Adv. Electr. Comput. Eng.* 18 (4) (2018) 121–128.
- [39] Perez, L., & Wang, J. (2017). The effectiveness of data augmentation in image classification using deep learning. *arXiv*. <https://doi.org/10.48550/ARXIV.1712.04621>.
- [40] C.C. Andrea, B.B.M. Daniel, J.B.J. Misael, Precise weed and maize classification through convolutional neuronal networks, in: Proceedings of the IEEE Second Ecuador Technical Chapters Meeting (ETCM), 2017, pp. 1–6.
- [41] D. Shadrin, A. Menshchikov, D. Ermilov, A. Somov, Designing future precision agriculture: detection of seeds germination using artificial intelligence on a low-power embedded system, *IEEE Sens. J.* 19 (23) (2019) 11573–11582.
- [42] Z. Liu, S. Wang, Broken corn detection based on an adjusted YOLO with focal loss, *IEEE Access* 7 (2019) 68281–68289.
- [43] V. Partel, S.C. Kakarla, Y. Ampatzidis, Development and evaluation of a low-cost and smart technology for precision weed management utilizing artificial intelligence, *Comput. Electron. Agric.* 157 (2019) 339–350.
- [44] S. Liu, X. Li, M. Gao, Y. Cai, R. Nian, P. Li, T. Yan, A. Lendasse, Embedded online fish detection and tracking system via YOLOv3 and parallel correlation filter, in: Proceedings of the OCEANS 2018 MTS/IEEE Charleston, 2018, pp. 1–6.
- [45] V. Mazzia, A. Khaliq, F. Salvetti, M. Chiaberge, Real-time apple detection system using embedded systems with hardware accelerators: an edge AI application, *IEEE Access* 8 (2020) 9102–9114.
- [46] I.A. Quiroz, G.H. Alférrez, Image recognition of legacy blueberries in a chilean smart farm through deep learning, *Comput. Electron. Agric.* 168 (2020), 105044.
- [47] C.E. Kim, M.M.D. Oghaz, J. Fajtl, V. Argyriou, P. Remagnino, A comparison of embedded deep learning methods for person detection (2018). *arXiv*. <https://doi.org/10.48550/ARXIV.1812.03451>.
- [48] R. Kirk, G. Cielniak, M. 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) (2020) 275.
- [49] C. Zhou, J. Hu, Z. Xu, J. Yue, H. Ye, G. Yang, A novel greenhouse-based system for the detection and plumpness assessment of strawberry using an improved deep learning technique, *Front. Plant Sci.* 11 (2020) 559.
- [50] ECCC. (2019). Historical climate data-environment and climate change canada (ECCC). Retrieved August 4, 2021 from <https://climate.weather.gc.ca>; Canadian Daily Climate Data [Database]. Gov. Canada.
- [51] LLC, G. (2019). Google maps™: map of wild blueberry sites used for image data collection in summer 2019. Available online: <https://www.google.ca/maps/@45.4360162,-63.4581544,14z>. (accessed on 29 September 2021).
- [52] G. Jiang, C. Wong, S. Lin, M. Rahman, T. Ren, N. Kwok, H. Shi, Y.H. Yu, T. Wu, Image contrast enhancement with brightness preservation using an optimal gamma correction and weighted sum approach, *J. Mod. Opt.* 62 (7) (2015) 536–547.
- [53] J.A. Khan, Y. Chen, Y. Rehman, H. Shin, Performance enhancement techniques for traffic sign recognition using a deep neural network, *Multimed. Tools Appl.* 79 (29) (2020) 20545–20560.
- [54] R. Huang, J. Gu, X. Sun, Y. Hou, S. Uddin, A rapid recognition method for electronic components based on the improved YOLO-V3 network, *Electronics* 8 (8) (2019) 825 (Basel).
- [55] W. Chen, C. Ju, Y. Li, S. Hu, X. Qiao, Sugarcane stem node recognition in field by deep learning combining data expansion, *Appl. Sci.* 11 (18) (2021) 8663.
- [56] A.W. Schumann, N.S. Mood, P.D. Mungofa, C. MacEachern, Q. Zaman, T. Esau, Detection of three fruit maturity stages in wild blueberry fields using deep learning artificial neural networks, in: Proceedings of the ASABE Annual International Meeting, 2019, p. 1.
- [57] Redmon, J., Bochkovskiy, A., & Sinigardi, S. (2019). *Darknet: yOLOv3-neural network for object detection*. Available online: <https://github.com/AlexeyAB/darknet>. (accessed on 15 March 2020).
- [58] Y. Huang, J.C. Zheng, S.D. Sun, C.F. Yang, J. Liu, Optimized YOLOv3 algorithm and its application in traffic flow detections, *Appl. Sci.* 10 (9) (2020) 3079.
- [59] Bochkovskiy, A. (2019). AlexeyAB/darknet: windows and Linux version of darknet yolo v3 v2 neural networks for object detection (tensor cores are used). In GitHub, 2019. <https://github.com/AlexeyAB/darknet>. (accessed on 15 March 2020).
- [60] J. Huang, H. Zhang, L. Wang, Z. Zhang, C. Zhao, Improved YOLOv3 model for miniature camera detection, *Opt. Laser Technol.* 142 (2021), 107133.
- [61] O. Russakovsky, L.J. Li, L. Fei-Fei, Best of both worlds: human-machine collaboration for object annotation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2121–2131.
- [62] S. Bargoti, J. Underwood, Deep fruit detection in orchards, in: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2017, pp. 3626–3633.
- [63] I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez, C. McCool, Deepfruits: a fruit detection system using deep neural networks, *Sensors* 16 (8) (2016) 1222.
- [64] S.M. Sharpe, A.W. Schumann, J. Yu, N.S. Boyd, Vegetation detection and discrimination within vegetable plasticulture row-middles using a convolutional neural network, *Precis. Agric.* 21 (2) (2020) 264–277.
- [65] S. Bianco, R. Cadene, L. Celona, P. Napoletano, Benchmark analysis of representative deep neural network architectures, *IEEE Access* 6 (2018) 64270–64277.