

DEVELOPMENT OF AN AUTOMATED DEBRIS DETECTION SYSTEM FOR WILD
BLUEBERRY HARVESTERS USING A CONVOLUTIONAL NEURAL NETWORK TO
IMPROVE FRUIT QUALITY

by

Anup Kumar Das

Submitted in partial fulfilment of the requirements
for the degree of Master of Science

at

Dalhousie University
Halifax, Nova Scotia
October 2020

© Copyright by Anup Kumar Das, 2020

DEDICATION

I would like to dedicate my M.Sc dissertation to my parents (Bablu Kumar Das and Fulkumary Das).

Author

Anup Kumar Das

TABLE OF CONTENTS

| | |
|---|------|
| LIST OF TABLES | viii |
| LIST OF FIGURES | xi |
| ABSTRACT..... | xiii |
| LIST OF ABBREVIATIONS USED | xiv |
| ACKNOWLEDGEMENTS..... | xvii |
| CHAPTER 1: INTRODUCTION | 1 |
| 1.1 Objectives | 3 |
| CHAPTER 2: REVIEW OF LITERATURE | 4 |
| 2.0 Wild Blueberry Cropping System..... | 4 |
| 2.1 Wild Blueberry Harvesting System | 4 |
| 2.2 Factors Affecting Wild Blueberry Fruit Quality..... | 5 |
| 2.2 Machine Learning Over Deep Learning | 6 |
| 2.2.1 Supervised Deep Learning..... | 7 |
| 2.3 Convolutional Neural Network..... | 8 |
| 2.3.1 AlexNet | 8 |
| 2.3.2 ZFNet | 9 |
| 2.3.3 GoogleNet | 9 |
| 2.3.4 VGGNet..... | 10 |
| 2.3.5 ResNet..... | 10 |

| | |
|--|-----------|
| 2.3.6 DenseNet..... | 11 |
| 2.4 Modern Object Detection Algorithm YOLOv3 | 12 |
| 3.0 Conclusion | 14 |
| CHAPTER 3: TRAINING AND OPTIMIZING A CNN FOR DEBRIS DETECTION DURING MECHANICAL WILD BLUEBERRY HARVESTING | 15 |
| Abstract | 15 |
| 3.0 Introduction..... | 16 |
| 3.1 Materials and Methodology | 19 |
| 3.1.1 Data Collection and Preparation | 19 |
| 3.1.2 Image Augmentation..... | 20 |
| 3.1.3 You Only Look Once Version 3 (YOLOv3) Family | 23 |
| 3.1.4 YOLO Parameter Settings | 24 |
| 3.1.5 YOLO Training and Testing..... | 25 |
| 3.1.6 Model Evaluation..... | 26 |
| 3.1.7 Statistical Analysis..... | 27 |
| 3.0 Result and Discussion..... | 27 |
| 3.2.1 Effects of Data Augmentation on Accuracy | 30 |
| 3.2.2 T-test on Testing Images of T1 Dataset..... | 31 |
| 3.2.3 T-test on Testing Images of T2 Dataset..... | 32 |
| 3.0 Conclusions..... | 34 |

CHAPTER 4: DEVELOPMENT OF A REAL-TIME DEBRIS DETECTION SYSTEM (HARDWARE & SOFTWARE) FOR MECHANICAL WILD BLUEBERRY HARVESTER. 36

| | |
|---|----|
| Abstract..... | 36 |
| 4.0 Introduction..... | 37 |
| 4.1 Material and Methodology..... | 40 |
| 4.1.1 Video Acquisition Hardware Development..... | 40 |
| 4.1.2 Camera Acquisition Software Development | 41 |
| 4.1.3 Determine the Minimum Processing Time Required for a Debris Detection System to Perform in Real-time..... | 43 |
| 4.1.4 Determine Detection Speed of CNNs for Developing Debris Detection System..... | 44 |
| 4.1.4.1 Data Collection and Preparation | 44 |
| 4.1.4.2 Hardware Setup..... | 46 |
| 4.1.4.3 Darknet Installation..... | 47 |
| 4.1.4.4 Measurement of Prediction Time..... | 48 |
| 4.1.4.5 Statistical Analysis for Determining the Effect of Models and Hardware on Prediction Time | 48 |
| 4.1.5 Validation of Models | 49 |
| 4.1.6 Development of a Debris Detection System..... | 50 |
| 4.2 Result and Discussion..... | 51 |
| 4.2.1 Validation of Models | 53 |
| 4.2.1.1 Evaluation of Model-1 Under Different IoU and Confidence Thresholds | 53 |
| 4.2.1.2 Evaluation of the Model-2 Under Different IoU and Confidence Thresholds | 55 |

| | |
|--|----|
| 4.3 Conclusion | 58 |
| CHAPTER 5: EVALUATION OF OPTIMIZED CNN MODEL FOR DEBRIS DETECTION ON THE IMAGES CAPTURED DURING HARVESTING | |
| Abstract | 60 |
| 5.0 Introduction..... | 60 |
| 5.1 Material and Methodology..... | 62 |
| 5.1.1 Data Collection | 62 |
| 5.1.3 Statistical Analysis..... | 64 |
| 5.2 Results and Discussion | 64 |
| 5.3 Conclusion | 68 |
| CHAPTER 6: GENERAL CONCLUSION AND FUTURE RECOMMENDATION | |
| 6.1 General Conclusion..... | 69 |
| 6.2 Future Recommendation..... | 70 |
| REFERENCES | 73 |
| Appendix a: Class average precisions (%) of YOLOv3, YOLOv3-SPP, and YOLOv3-Tiny for leaves and stems class on non augmentation dataset..... | |
| | 89 |
| Appendix b: Class average precisions (%) of YOLOv3, YOLOv3-SPP, and YOLOv3-Tiny for ripe berries and dirt class on non augmentation dataset..... | |
| | 90 |
| Appendix c: F1-score and mAP(%) of YOLOv3, YOLOv3-SPP, and YOLOv3-Tiny on non augmentation dataset..... | |
| | 91 |
| Appendix d: Class average precisions (%) of YOLOv3 and YOLOv3-SPP for leaves and stems class on T1 augmentation dataset | |
| | 92 |

| | |
|---|----|
| Appendix e: F1-score and mAP(%) of YOLOv3, YOLOv3-SPP on T1 dataset..... | 93 |
| Appendix f: Class average precisions (%) of YOLOv3 and YOLOv3-SPP for leaves and stems class on T2 augmentation dataset | 94 |
| Appendix g: F1-score and mAP(%) of YOLOv3, YOLOv3-SPP on T2 augmentation dataset... | 95 |
| Appendix h: Prediction time on four different hardware and model combinations | 96 |
| Appendix i: Automatic detection of leaves, stems, green berries, ripe berries, stems, and dirt with corresponding ground truth..... | 97 |

LIST OF TABLES

CHAPTER 3

| | |
|---|----|
| Table 3-1. Description of data augmentation techniques..... | 21 |
| Table 3-2. Description of original and augmented dataset | 23 |
| Table 3-3. Training and testing dataset..... | 23 |
| Table 3-4. Initialization parameters of the YOLO networks | 25 |
| Table 3-5. Precision, Recall, F1-score, and mAP (%) of the YOLOv3-SPP, YOLOv3 and YOLOv3-Tiny under 0.5 IoU and 0.25 confidence threshold..... | 27 |
| Table 3-6. Average precision (%) of YOLOv3, YOLOv3-SPP and YOLOv3-Tiny..... | 27 |
| Table 3-7. Comparison of F1-score of YOLOv3, YOLOv3-SPP and YOLOv3-Tiny using the Tukey method at 5% ($\alpha = 0.05$) level of significance..... | 28 |
| Table 3-8. Comparison of class average precisions (%) of YOLOv3, YOLOv3-SPP and YOLOv3-Tiny for leaves and stems using the Tukey method at 5% ($\alpha = 0.05$) level of significance | 28 |
| Table 3-9. Comparison of class average precision (%) of YOLOv3, YOLOv3-SPP, and YOLOv3-Tiny for green berries and ripe berries using the Tukey method at 5% ($\alpha = 0.05$) level of significance | 29 |
| Table 3-10. Comparison of class average precision of YOLOv3, YOLOv3-SPP, and YOLOv3-Tiny for dirt using the Tukey method at 5% ($\alpha = 0.05$) level of significance | 29 |
| Table 3-11. Precision, Recall, F1-score and mAP (%) of the YOLOv3-SPP and YOLOv3 under 0.5 IoU and 0.25 confidence threshold | 30 |
| Table 3-12. Average precision (%) of YOLOs with data augmentation under 0.5 IoU and 0.25 confidence threshold | 31 |
| Table 3-13. Comparison of F1-score and mAP (%) of YOLOv3-SPP and YOLOv3 using the t-test at 5% ($\alpha = 0.05$) level of significance..... | 31 |
| Table 3-14. Comparison of class average precisions (%) of YOLOv3-SPP and YOLOv3 for leaves and stems using the t-test at 5% ($\alpha = 0.05$) level of significance | 32 |
| Table 3-15. Comparison of class average precision of YOLOv3-SPP and YOLOv3 for green berries and ripe berries using the t-test at 5% ($\alpha = 0.05$) level of significance..... | 32 |

| | |
|--|----|
| Table 3-16. Comparison of class average precision of YOLOv3-SPP and YOLOv3 for dirt using the t-test at 5% ($\alpha = 0.05$) level of significance | 32 |
| Table 3-17. Comparison of F1-score and mAP (%) YOLOv3-SPP and YOLOv3 using the t-test at 5% ($\alpha = 0.05$) level of significance..... | 32 |
| Table 3-18. Comparison of class average precisions (%) of YOLOv3-SPP and YOLOv3 for leaves and stems using the t-test at 5% ($\alpha = 0.05$) level of significance | 33 |
| Table 3-19. Comparison of class average precisions (%) of YOLOv3-SPP and YOLOv3 for green berries and ripe berries using the t-test at 5% ($\alpha = 0.05$) level of significance..... | 33 |
| Table 3-20. Comparison of class average precision of YOLOv3-SPP and YOLOv3 for dirt using the t-test at 5% ($\alpha = 0.05$) level of significance | 33 |

CHAPTER 4

| | |
|---|----|
| Table 4-1. Number of objects per image in different categories of images..... | 45 |
| Table 4-2. Dataset Description | 45 |
| Table 4-3. Computer architecture specification about four different hardware..... | 46 |
| Table 4-4. Results of ANOVA on the detection time on four different hardware at 5% ($\alpha = 0.05$) level of significance | 51 |
| Table 4-5. Result of MMC using Tukey method to identify significant difference on models and hardware at 5% ($\alpha = 0.05$) level of significance | 52 |
| Table 4-6. Evaluations of Model-1 under IoU (0.5) and threshold (0.25)..... | 53 |
| Table 4-7. The average precision (%) and mAP (%) of the Model-1 under different IoU and confidence threshold | 54 |
| Table 4-8. True positive, False positive, False negative, Precision, Recall and F1-score of Model-1 under different IoU and confidence threshold | 55 |
| Table 4-9. The average precision (%) and mAP (%) of the Model-2 under different IoU and confidence threshold | 55 |
| Table 4-10. True positive, False positive, False negative, Precision, Recall and F1-score of Model-2 under different IoU and confidence threshold | 56 |

CHAPTER 5

| | |
|---|----|
| Table 5-1. Pair-wise t-test for manually and automatically leaves, green berries, ripe berries, stems, and, dirt detection..... | 66 |
|---|----|

LIST OF FIGURES

CHAPTER 3

| | |
|---|----|
| Figure 3-1. Location of fields in google map | 19 |
| Figure 3-2. Image labeling using custom software developed with the Lazarus compiler..... | 20 |
| Figure 3-3. 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 = 1.2 (g) gamma correction = 1.30 (h) sharpening = 10 (i) brightness = 20 (j) contrast = 20 (k) saturation = 15 | 22 |
| Figure 3-4. Detection comparison of YOLOv3 (a), YOLOv3-SPP(b), and YOLOv3-Tiny(c) trained on non augmented dataset..... | 29 |
| Figure 3-5. Detection results of YOLOv3-SPP on a testing image..... | 34 |

CHAPTER 4

| | |
|---|----|
| Figure 4-1. Camera setup at the side and rear conveyor..... | 41 |
| Figure 4-2. (a) Field of view to the rear conveyor at 48.5 cm camera height; (b) field of view of side conveyor at 34.0 cm camera height..... | 42 |
| Figure 4-3. Location of fields in google map | 44 |
| Figure 4-4. Image labeling using custom software developed with the Lazarus compiler..... | 45 |
| Figure 4- 5. Flow chart of YOLOv3 network architecture | 47 |
| Figure 4-6. Flowchart of measuring prediction time of an image. | 48 |
| Figure 4-7. GUI of a debris detection system. | 53 |
| Figure 4-8. Validation of models under different IoU and confidence thresholds. | 57 |

CHAPTER 5

| | |
|---|----|
| Figure 5-1. Flowchart of saving detection results of YOLOv3-SPP. | 63 |
| Figure 5-2. Automatic detection of leaves, stems, green berries, ripe berries, and dirt using GUI | 64 |
| Figure 5-3. Automatic detection vs manual detection (ground truth). Automatic detection vs manual detection of green berries (a), ripe berries (b), leaves (c), stems (d) and, dirt (e)..... | 65 |
| Figure 5-4. Automatic detection of leaves, stems, green berries, ripe berries, and dirt on the image was represented as 1, 2, 3, 4, and 5, respectively. | 67 |
| Figure 5-5. Image labeling using custom software developed with the Lazarus compiler..... | 67 |

ABSTRACT

Improving wild blueberry fruit quality has become increasingly important to producers due to the tightening profit margin facing the industry. The continuous development of field management practices (i.e., application of fungicides, herbicides, fertilizers, pollination, and pruning, etc.), has improved the wild blueberry field vegetation by significant increases in plant densities, plant height, and fruit yield. This increased plant debris causes additional leaves and stems to enter into the fruit storage bins during mechanical harvesting resulting in a potential reduction of fruit quality. An experimental image dataset (1000 images) was collected from a mechanical harvester in two commercially managed fields in central Nova Scotia. Three different deep learning algorithms (YOLOv3, YOLOv3-Tiny, and YOLOv3-SPP) were implemented and compared for developing the real-time debris detection system. The image dataset was augmented using five different color-based data augmentation techniques (sharpening, brightness, contrast, gamma correction, saturation). T1 dataset containing 2000 images was prepared by changing sharpness, brightness, contrast, gamma, and saturation value of 1000 images by factors of 10, 20, 20, 1.3 and 1.5 respectively and mixing with 1000 experimental images. Similarly, T2 dataset containing 3000 images was prepared by changing the gamma value of images by factors of 0.70, 0.80, 0.90, 1.10 and 1.20 and mixing with experimental images and T1 dataset. YOLOv3-SPP achieved 73.03% of mAP and 74.38% of mAP after training on T1 and T2 dataset. YOLOv3-SPP improved mAP by 5.89% from previous accuracy (YOLOv3-SPP: mAP = 68.49%). The two best performing models (Model-1: YOLOv3-SPP, mAP: 74.38%; Model-2: YOLOv3-SPP, mAP: 73.03%) were tested on four different types of CPU, GPU, and embedded based computers. A 4x2 factorial design was used to choose the most appropriate hardware and model for developing the system. Results determined YOLOv3-SPP (mAP: 73.03%) operated using Intel® Core™ i9-7900X CPU @ 3.30 GHz, GeForce RTX™ 2080 Ti @ 1665 MHz on a desktop computer achieved the fastest detection (33.30 ms) with the highest average frame rate (30.03 FPS). Model-1 and Model-2 achieved 85.90% and 86.10% of validation mAP respectively at 0.10 IoU and 0.10 confidence threshold. A paired t-test was implemented to investigate the difference between automatic detection with ground truths. Results showed that the developed system successfully detected green berries, ripe berries and leaves during lab evaluation at 5% level of significance. However, stems, and dirt classes were not successfully detected from ground truths at 5% level of significance.

In this study YOLOv3-SPP was implemented for developing debris detection system which performed comparatively better than YOLOv3 or YOLOv3-Tiny. This system can be incorporated in a control system to automate brush adjustment on the basis of feedback from conveyors of mechanical wild blueberry harvesters. This system can be a valuable addition for enhancing berry cleaning efficiency and improving fruit quality.

LIST OF ABBREVIATIONS USED

ABBREVIATIONS

AMD- Advanced Micro Devices

ANN- Artificial Neural Network

ANOVA- Analysis of Variance

API- Application Programming Interface

ARM- Advanced Reduced Instruction Set Computing Machine

AVX- Advanced Vector Extensions

BBO- Biogeography-Based Optimization

BGR- Blue Green Red

BPNN- Back Propagation Neural Network

CA- Canada

CNN- Convolution Neural Network

CNNs- Convolution Neural Networks

COCO- Common Objects in Context

CPU- Central Processing Unit

CUDA- Compute Unified Device Architecture

CUDNN- Compute Unified Deep Neural Network

DBE- Doug Bragg Enterprises Ltd.

DDR3- Double Data Rate 3

DDR4- Double Data Rate 4

DL- Deep Learning

DNN- Deep Neural Network

eMMC- Embedded Multi-Media Controller

FCN- Fully Convolutional Network

FDDB- Face Detection Data Set and Benchmark

FN- False Negative

FP- False Positive
FPS- Frames Per Second
FRFE- Fractional Fourier Entropy
FSCABC- Fitness-Scaled Chaotic Artificial Bee Colony
GB- Gigabyte
GHz- Gigahertz
GPU- Graphics Processing Unit
GT- Ground Truth
GUI- Graphical User Interface
HDD- Hard Disk Drive
IHGA- Hybrid Genetic Algorithm
ILSVRC- Large Scale Visual Recognition Challenge
IoU- Intersection Over Union
JPG- Joint Photographic Experts Group
KNN- K-Nearest Neighbors
kSVM- Kernel Support Vector Machine
LBP- Local Binary Patterns
LPDDR4- Low Power Double Data Rate 4
mAP- Mean Average Precision
MHz- Megahertz
ML- Machine Learning
MMC- Multiple Mean Comparison
ms- Milliseconds
ms⁻¹- Meters Per Second
NMS- Non-Maximum Suppression
NSERC- Natural Science and Engineering Research Council
NY- New York
OPENCV- Open Computer Vision

OpenCV- Open Computer Vision
OPENMP- Open Multi-Processing
PCA- Principal Component Analysis
PMRA- The Pest Management Regulatory Agency
RCNN- Region-Based Convolutional Neural Network
R-CNN- Region-Based Convolutional Neural Network
ResNet- Residual Neural Network
R-FCN- Region-Based Fully Convolutional Network
RGB- Red Green Blue
rpm- Revolutions Per Minute
RPN- Region Proposal Network
SPP- Spatial Pyramid Pooling
SSD- Single Shot MultiBox Detector
SVM- Support Vector Machine
TB- Terabyte
TP- True Positive
UK- United Kingdom
USA- United States of America
USB- Universal Serial Bus
W- Watt
WE- Wavelet Entropy
YOLO- You Only Look Once
YOLOv2- You Only Look Once version 2
YOLOv3- You Only Look Once version 3

ACKNOWLEDGEMENTS

First and foremost, I am deeply grateful to the Almighty Lord Krishna who took away my ignorance with the torch of knowledge and blessed me throughout the life. I would like to thank him for showing constant grace and mercy that made me possible to complete my master's study successfully.

I am extremely grateful to my supervisor, Prof. Dr. Qamar Zaman for his scholastic supervision, constant guidance, appreciation, and inspiration that helped me possible to fight against all odds. I would like to offer my gratitude and special thanks to my co-supervisor, Dr. Travis Esau for his supervision, encouragement, critical but constructive comments, valuable suggestions, and supports throughout the study. I would like to express my gratitude to my committee members, Dr. Aitazaz A. Farooque and Dr. Arnold Schumann for their guidance and suggestions during my experiments in this study.

I would like to offer my special thanks to Doug Bragg Enterprises (DBE) Limited, the Natural Science and Engineering Research Council (NSERC) of Canada, and the New Brunswick Department of Agriculture for the financial support for my study. I would also like to acknowledge the assistance of Dr. Peter Havard (Head of Engineering Department), Dr. Christopher Cutler (Associate Dean Research), and Mrs. Pamela Sutherland (Graduate Secretary). I would like to extend my sincere thanks to Craig MacEachern (Research Assistant at the Department of Engineering) for helping me during the data collection and reviewing my thesis documents. I am also extremely grateful to Md. Sultan Mahmud for his unconditional support, instruction, and inspiration from the beginning of my study.

I am also grateful to the precision agriculture research team at Dalhousie's Faculty of Agriculture, Derrick Ouma, Jack Lynds and Connor Mullins for their help during data collection.

I would like to especially thank Mr. Hassan Nabi, Mrs. Irin Arju, Dr. Nasif Sarowar, Mr. Mahboob Rahman, Dr. Zahidul Alam, and their family for showing unconditional love, care, and support during the years of my study.

Finally, I greatly appreciate and sincerely thank my parents and family members for their sacrifice, care, constant encouragement, and supports. I would like to thank all of my teachers in school, college, and university for their guidance and the great lessons. I would also like to thank my friends Muhammad Saad, Hassan Afzaal, Nazar Hussain, Sabiha Antora, Patrick Hennessy, and senior Ahmad Hanmbal Khan for their help.

CHAPTER 1: INTRODUCTION

The wild blueberry (*Vaccinium angustifolium* Ait.) is a naturally growing horticultural crop in Atlantic Provinces of Canada and the Maine State of USA. Canada produced around 87.94 million kg of berries during the year 2019 valued at \$88.89 million (Statistics Canada 2019). The wild blueberry producers are concerned about the berry quality after the harvesting process; the quality wild blueberries are free from poor quality berries and foreign materials (New Brunswick, Canada 1996). Spann et al. (2010) stated that debris in the harvested citrus fruits could increase transportation and processing costs. Fruit processors also expect debris free berry from the wild blueberry producers (Esau et al., 2018). However, the possibilities of debris removal during hand harvesting (i.e., hand raking) are limited as it requires extensive labour and adequate time for harvesting operation (Kinsman, 1993; Donahue et al., 1999). Therefore, the high labour cost, declining availability of farm labour, and the short harvesting window are the major factors behind the design and development of efficient mechanical harvester (Yarborough, 1991). Due to the continuous development of field management practices (i.e., application of fungicides, herbicides, fertilizers, pollination, and pruning etc.), the blueberry field vegetation has been changed resulted in significant increases in plant densities, plants height, and fruit yield (Esau et al., 2018). The increased amount of plant leaves and foliage reduces the berry/fruit quality when it enters into the storage bin during mechanical harvesting due to inappropriate debris separation technology (Esau et al., 2018). The majority of operators (53.8%) run the debris cleaning brush at a shallow adjustment depth which can allow debris to accumulate during harvesting (Esau, 2019). Improper adjustment of the brush enhances the risk of berry damage and allows the picker teeth to accumulate debris in wet and weedy field conditions resulting in reduction of fruit quality (Esau, 2019). Esau et al. (2018) tested a dual blower fan system and successfully removed 98.8% and

98.6% of debris (weight basis) at dry and wet field conditions respectively during mechanical wild blueberry harvesting. However, debris occasionally gets stuck in air inlet vanes of commercial blower fan system and hampers the debris cleaning performance leading to reduced field efficiency of harvesters (Esau, 2019). Therefore, an automatic debris separation technology in mechanical wild blueberry harvester may be a solution to separate and monitor debris, by automatically adjusting the debris cleaning brush during harvesting. It may save time required for cleaning at the processing plant and improve fruit quality by reducing berry shrinkage. Additionally, high-quality berry production has become a priority due to the market competition of highbush blueberries with wild blueberries on a global scale. Therefore, the development of an automatic debris separation technology is utmost important to ensure quality berries in the storage bin.

Machine vision based automated technology incorporated with machine learning (ML) are gaining popularity nowadays for their application in different agricultural cropping systems (Granitto et al., 2002; Piedad et al., 2018; Tu et al., 2018). The major concern of ML is that it requires the feature of a specified engineering field and domain expertise for feature extraction (LeCun et al., 2015). The accuracy and computational cost of ML algorithm can vary due to inappropriate feature extraction (Al-Hiary et al., 2011; Chang et al., 2012; Rehman et al., 2018). Conversely, Deep Learning (DL) is a current state of the art technique that can extract the required features automatically from raw data including image, audio, video, and speech (LeCun et al., 2015). Cavallo et al. (2018) developed a system using convolutional neural networks (CNNs) based computer vision technique for evaluating packaged fresh-cut lettuce quality and achieved 83% of classification accuracy. Wang et al. (2018) developed a system for evaluating the internal quality of blueberry by feeding hyperspectral transmittance images to CNNs (ResNet, ResNeXt) and achieved F1 scores of 87.84% and 89.05%, respectively. Researchers also used CNNs in

different fields of agriculture including detection of various plant diseases (Mohanty et al., 2016; Amara et al., 2017; Khan et al., 2018), weed (Dyrmann et al., 2016; Milioto 2017; Sharpe et al., 2018), and fruit counting (Rahnemoonfar et al., 2017) with higher success rates.

The wild blueberry industry faces significant fruit quality losses with mechanical harvesting due to inefficient as well as inappropriate debris separating strategies/technologies. A CNN based automated debris detection can be powerful technology for separating debris from ripe berries. The system can be implemented in a feedback control system for automating the debris cleaning brush on the basis of different field conditions which can prevent debris to enter into the storage bin. The system can also be incorporated for further upgrading from dual blower fan system to multi air channel-based blower fan system for optimizing debris cleaning performance and enhance harvester efficiency. Therefore, a CNN based automated debris separation technology can be a solution for the wild blueberry industry to improve quality of the harvested fruit.

1.1 Objectives

1. Training and optimizing a CNN for debris detection during mechanical wild blueberry harvesting.
2. Development of a real-time debris detection system (hardware & software) for mechanical wild blueberry harvester.
3. Evaluation of optimized CNN model for debris detection on the images captured during harvesting.

CHAPTER 2: REVIEW OF LITERATURE

2.0 Wild Blueberry Cropping System

Wild blueberry (*Vaccinium angustifolium* Ait.) is a unique crop native to Northeastern North America that produced approximately 147 million kg of berries only in New Brunswick, Nova Scotia, Prince Edward Island, and Quebec provinces of Canada valued at \$90.70 million (Statistics Canada 2016). These perennial crops are forced to become a biennial crop by regular pruning during the sprout year (Hall et al., 1979); vegetative growth happens during the first year, pollination and production of berry occur during the next year (Eaton, 1988). The wild blueberry plants are developed from rhizomes that spread under the soil about 0.05 to 0.08 meters (m) per year (Kinsman, 1993) with height of the plants ranging from 0.10 to 0.38 m (Farooque et al., 2013). Wild blueberry ripening begins in late July to August of the production year (Hall et al., 1979) and their harvesting starts from the first week of August to the end of the same month in Nova Scotia (Ali, 2016). After harvesting of the berries, the fields are pruned in Fall or Spring by either burning or mowing (Malay, 2000). Majority of the wild blueberry fields (above 80%) in Canada are harvested with commercial mechanical harvesters (PMRA, 2005) but still, some rough terrain fields are hand raked (Yarborough, 1991).

2.1 Wild Blueberry Harvesting System

Hand raking is a tedious and backbreaking wild blueberry harvesting method (Gidge, 1995). The berry picking efficiency of hand rakers is about 80% and responsible for 20% losses on an average (Kinsman, 1993). However, hand raking cannot separate field debris (stems and leaves) from the harvested berries (Kinsman, 1993; Gidge, 1995; Donahue et al., 1999).

The research into developing a wild blueberry mechanical harvester started in the early 1950s (Dale et al., 1994). As a result, a machine manufacturer (Bragg Lumber Company, Collingwood, Nova Scotia) got success in developing a viable mechanical harvester in 1979 (Dale et al., 1994). They initiated a hydraulic for controlling the rotational speed of header (Malay, 2000). Doug Bragg Enterprises Ltd. (DBE) is the leading manufacturers of wild blueberry harvester in Northeastern North America (Esau et al., 2018). A study in 1983 proposed that picking efficiency of Bragg mechanical harvester can be 68% and 76% in weedy and clean smooth fields, respectively (Hall et al. 1983, Malay, 2000). The modern DBE harvester head is configured with 16 rotating rakes resulting into low fruit loss and less debris (Esau et al., 2018). Farooque et al. (2014) experimented the effect ground speed and header revolution of DBE harvester and achieved higher berry picking efficiency (i.e., > 90%). However, for the three levels of harvesting time (early, middle, and late) the picked fruits exhibit physical and chemical changes during harvesting (Ali, 2016). The operation of the mechanical harvester at 0.44 ms^{-1} and 28 revolutions per minute (rpm) results in observable berry loss (Ali, 2016). Ali et al. (2016) developed an effective relationship among harvest time, machine parameters, fruit quality for minimizing loss while harvesting. Fruit losses are higher in the late season compared to the early and middle season and the combination 0.33 m s^{-1} and 26 rpm can be the profitable combination in harvesting season (Ali, 2016).

2.2 Factors Affecting Wild Blueberry Fruit Quality

Wild Blueberry field management practices have improved over the past few decades resulting in the significant increase of berry yield and plant foliage (Yarborough and Ismail, 1985; Litten et al., 1997; Esau et al., 2014). This increased plant density produces more debris during harvesting process that reduces fruit quality by entering the storage bin of the harvested fruits (Esau et al., 2018). The quality berries are indicative of a small number of broken berries and less

amount of debris in the storage bins (Hall et al., 1983). The amount of debris in the harvested bins may depend on field conditions, the number of picker bar teeth, and the blower fan speed (Esau et al., 2018). Esau et al. (2018) tested the performance of a harvester and reported that the tested harvester produced a higher percentage of debris at wet harvest conditions compared to the dry harvest conditions. They observed that the harvester using head of 63-tooth picker bars harvested lesser debris as compared to that with 65-tooth picker bar. They also observed, after increasing blower fan speeds from 18 to 23 ms⁻¹, that the percentage of debris in storage bin did not significantly change. Field debris is to be removed by a winnowing operation at a processing stage (Donahue et al., 1999). Field debris including dirt, small berry, and foreign material increases berry shrinkage that reduces the berry quality and acceptability in the processing and cleaning lines (McIsaac, 1998).

2.2 Machine Learning Over Deep Learning

Machine learning has been widely applied for food and fruit quality evaluation in the agricultural sector (Nakano, 1997; Nagata et al., 1998; Piedad et al., 2018). However, feature extraction is a major concern in ML (LeCun et al., 2015). Kim et al. (2009) developed four classification models for classifying grapefruit peel diseases using the textural feature. They observed that the model used eleven intensity (I-11) texture features that provided lower overall accuracy (81.7%) than other models. Chang et al. (2012) and Rehman et al. (2018) also observed significant classification accuracy and computational time difference due to inappropriate feature extraction. Feature extraction in ML requires feature engineering, whereas DL can extract feature automatically from the raw data (LeCun et al., 2015). The automatic feature extraction from the raw data in DL reduces the need of feature engineering (LeCun et al., 2015).

2.2.1 Supervised Deep Learning

Supervised learning is the most common form of learning approach in DL and ML. In this approach, a machine experiences the known labelled data with its category (LeCun et al., 2015). Fan et al. (2014) developed a deep neural network (Pyramid CNN) for face recognition using a supervised learning approach. They compared the traditional face representation method (LBP+PCA) with Pyramid CNN and achieved 97.3% of recognition accuracy in the labelled Faces in the Wild database. Yang et al. (2015a) developed a CNN to recognize human activity using time series signal data and observed that the CNN performed better than other classifiers such as Support Vector Machine (SVM), k-nearest neighbors (KNN), Means and Variance, and Deep Belief Network. A detector based on a DL algorithm had been proposed to detect retouched face images (Bharati et al., 2016). The authors reported that the supervised deep model outperformed another unsupervised deep Boltzmann machine and Kee and Farid algorithm. Dubrovina et al. (2018) proposed a supervised CNN for tissue segmentation and classification using mammography images. Do et al. (2018) presented a new supervised DL approach to detect objects and their affordances in real-time. Their proposed technique was able to process images end to end with higher inference speed which would take 150 milliseconds (ms) for processing each image. Sa et al. (2018) developed a detector using a dense semantic segmentation approach based on CNN for detecting background, weeds, crops from multispectral images. Their model successfully detected weeds with 80% of F1-score. However, there was another kind of learning approach used for clustering or determining the distribution of data known as unsupervised learning (Xie et al., 2016).

2.3 Convolutional Neural Network

Convolutional Neural Network is a multilayer DL network designed to process data including image, audio, and video (LeCun et al., 2015). The convolutional layer of CNN works to extract features and uses a pooling layer to map the extracted feature(s). Convolutional Neural Network can be used either image classification or object detection. Image classification is widely used to classify different objects based on the image category. Image classification algorithms are limited to understand the locations of objects in images. However, understanding the location of objects on images is important for solving several computer vision problems including identifying disease location, counting. On the other hand, object detection algorithms are efficient and reliable to classify as well as understand the locations of the target objects in images. Convolutional Neural Network in past years has achieved success in image classification and object detection in different fields of agriculture (Steen et al., 2016; Mohanty et al., 2016; Amara et al., 2017; Rahnemoonfar et al., 2017). Several researchers have worked on CNN to improve efficiency and reduce the computational cost by changing the structure (Szegedy et al., 2016). AlexNet, ZFNet, GoogleNet VGGNet, ResNet, and DenseNet are commonly used CNN architecture (Dyrmann et al., 2016; Mohanty et al., 2016; Huang et al., 2017; Rahnemoonfar et al., 2017; Enciso-Aragón et al. 2018; Fu et al., 2018; Wang et al., 2018).

2.3.1 AlexNet

Krizhevsky et al. (2012) developed AlexNet won the “Large Scale Visual Recognition Challenge” (ILSVRC) in 2012. AlexNet was tested on ImageNet Fall 2011 dataset resulted into 15.3% of top-5 error rate. The architecture was similar to LeNet-5 (LeCun, 1989) but deeper with five convolutional layers, three fully connected layers, and ended with a soft max layer. Enciso-

Aragón et al. (2018) developed a quality inspection system for lemons based on AlexNet architecture and classified fresh and spoiled lemons having accuracies of 98.25% and 93.73%, respectively. They estimated the parameter of lemons such as damages, weight, diameter by using fuzzy logic to categorized fruit quality as low, medium, and high. The architecture also achieved success in the agricultural sector for plant disease detection (Mohanty et al., 2016), obstacle detection (Steen et al., 2016), and plant recognition (Lee et al., 2015; Yalcin et al., 2017).

2.3.2 ZFNet

Zeiler and Fergus (2014) developed a CNN architecture known as ZFNet, which performed in ILSVRC-13 and achieved top-5 error rate of 14.8%. It was improved from AlexNet architecture by adding convolutional layers, tweaking hyperparameter, reducing filter, and stride size (Tahir et al., 2018). Fu et al. (2018) implemented ZFNet for the development of a Kiwi fruit detector in field images. The proposed detector performed outstandingly in the images of varying light conditions and achieved 92.3% of recognition rate.

2.3.3 GoogleNet

A 22-layer deep CNN architecture was developed by Szegedy et al. (2015) securing first place in ILSVRC-14 with top-5 error rate of 6.67%. The parameter of this architecture was optimized nine times more than the previous AlexNet architecture (Szegedy et al., 2016). A plant detector was developed by Mohanty et al. (2016) using CNN based GoogleNet architecture experiment on a PlantVillage dataset containing 54,306 plant leaves images for identifying crop-disease and achieved accuracy around 98.86% for classifying 38 different crop-disease. Dyrmann et al. (2017) developed an automatic weed detector using GoogleNet architecture that detected 46% of weeds due to the overlap of the weeds with plants.

2.3.4 VGGNet

Simonyan and Zisserman (2014) developed a very deep CNN and secured first runner up position in ILSVRC-14. The network is homogeneous in structure; however, employed about 138 million parameters that require high computational power. Dyrmann et al. (2016) modified the VGGNet architecture to develop a plant and weed detector. Their system achieved an overall 94.4% of classification accuracy and 100% detection rate. A real-time mango detector was developed by Liang et al. (2018) using VGGNet based SSD (Single Shot Multi Box Detector) network. Their system outperformed Faster Region-based CNN (Faster R-CNN) and achieved an F1-score of 0.911 at 35 frames per second (FPS). Zeng (2017) developed a system to classify 26 categories of fruit and vegetables using VGGNet. Their model achieved 95.6% of classification and recognition accuracy and performed better than another ML classifier including KNN and KSVM. This architecture also got success in land cover and crop type classifications (Kussul et al., 2017) and mango yield estimation (Stein et al., 2016).

2.3.5 ResNet

He et al. (2016) developed ResNet that became the winner of ILSVRC 15 and achieved top-5 error rate of 3.57% on the official validation set. The architecture was improved by implementing batch normalization and skip connector. Rahneemoonfar et al. (2017) developed a fruit counting system using a modified version of ResNet architecture trained on the dataset containing 24,000 synthetic images (artificially generated) and tested it on real (field) images. Their system outperformed other area-based techniques including the shallow neural network and original Inception-ResNet. The system achieved 91% validation accuracy on real images and 93% validation accuracy on synthetic images.

Xie et. (2017) developed a CNN architecture known as ResNeXt that secured second place in the ILSVRC 2016 and achieved top-5 error of 3.03%. ResNeXt and ResNet both use the same number parameter. A ResNeXt based detection system was developed by Wang et al. (2018) for evaluating the internal quality of blueberry by feeding hyperspectral transmittance images to CNN; it achieved F-1 score of 89%. ResNeXt took only 6.5 milliseconds (ms) for classifying each testing sample. Fuentes et al. (2017) used ResNeXt as a deep feature extractor and combined it with Faster R-CNN resulting into 71% mean average precision (mAP) for classifying tomato disease and pest.

2.3.6 DenseNet

DenseNet is a modern deep CNN architecture developed by Huang et al. (2017). A plant disease identification system was developed by Too et al. (2018) to identify 38 plant diseases using DenseNet. The author trained the four different architectures including DenseNet, VGG16, Inception V4, and ResNet using the PlantVillage dataset containing 54,306 images of healthy and infected plant leaves. DenseNet outperformed the other architecture such as VGG16, Inception V4, ResNet, and achieved 99.75% accuracy in the test dataset.

Many researchers used CNN architecture for image classification in agriculture where few researchers implemented it to understand the location of objects on the scene. Bargoti & Underwood (2016) and Stein et al. (2016) both used Faster R-CNN and VGG16 model for fruit detection. On the other hand, Dyrmann et al. (2017) and Sharpe et al. (2018) implemented DetectNet CNN based on GoogLeNet architecture for weed detection. However, Faster Region-based CNN uses selective search which requires higher computation cost than DetectNet.

2.4 Modern Object Detection Algorithm YOLOv3

YOLOv3 is a modern DL based object detector uses darknet-53 framework which is consisted of 53 convolutional layers. In common object in context (COCO) dataset YOLOv3 performed 3.8x times faster than RetinaNet and 3x times faster than SSD model (Redmon & Farhadi, 2018). Benjdira et al. (2019) developed a system for detecting car using CNNs. They compared the YOLOv3 with Faster R-CNN and observed YOLOv3 performed faster (0.056 ms) than Faster R-CNN (1390 ms) for processing a single image. The experimental results showed that YOLOv3 performed better than Faster R-CNN for detecting single object (car) having an accuracy of 99.07%. Liu et al., (2018a) also compared the performances YOLOv2 with YOLOv3 by training and testing both models on the dataset contained 15 common categories object. YOLOv3 outperformed YOLOv2 for detecting ships and bridges and took an average of 70 ms for processing each image. Ćorović et al. (2018) also experimented for multiclass object detection in real-time. They developed a system for detecting traffic participants including cars, trucks, pedestrians, traffic signs, and traffic lights. They trained the model on Berkley Deep Drive dataset (Yu et al., 2018) and tested on 300 traffic images of city Novi Sad, Serbia. The testing result showed that the 120 epochs model achieved highest accuracy (F-Score = 59%, mAP = 46.60%, average IoU = 45.98%). This system could run in real-time (~25 FPS) on 1920x1080 pixels video streaming. A real-time face detection system was developed by Yang et al. (2018) using YOLOv3. The system was trained with Celeb Faces (Yang et al., 2015b), FDDB (Jain et al., 2015), WIDER FACE (Yang et al., 2016) datasets. This system took an average of 0.03 ms for detecting the faces in an image. Shinde et al. (2018) also developed a system for detecting human behavior or action in video. They categorized human actions and labelled the actions from different videos and

divided the dataset into training and testing sets. This system performed outstanding in quantitative analysis having mAP and F-Score 89.88% and 88.34%, respectively.

Manual labelling or annotating of data is one of the main constraints for dataset preparation. An automatic image annotation system was proposed by Tumas et al. (2018) by using YOLOv3. A YOLOv3 as a detector was trained using the images that only had pedestrian. Then the experimental video of it was fed to the system after extracting frames; it was passed through the detector. The detector produced a probability for each frame and then frames were annotated by the system when the probability met the condition ($P \geq 0.7$). The system detected 50 pedestrians from 114 in test images.

Kim et al. (2018) compared 13 different variants of YOLO, SSD, RCNN, R-FCN, and SqueezeDet model to select an optimized model for the embedded application. They built GPU based system for comparison of these models. They analyzed models based on time complexity and average precision measure. The analyzed result showed that YOLOv-314 model outperformed other models which met the requirement of precision detection. They also observed that the embedded platform Nvidia Jetson TX2 provided more flexibility for deploying model than Movidius.

Nvidia Jetson TX2 platform was also used by Nagaraj et al. (2017) for YOLO deployment. They compared DetectNet with YOLO for detecting various street objects; however, YOLO performed better than DetectNet with 0.25 mAP. Similarly, Bhandary et al. (2017) compared Nvidia Jetson TX2 embedded platform with a desktop computer environment to investigate faster processing time and observed that the desktop computer environment was faster than Nvidia Jetson TX2. Therefore, the above literature showed a great potential of DL for classifying and detecting objects in the different fields of study.

3.0 Conclusion

Several researchers applied DL technology for fruit quality estimation, plant disease detection, weed or plant detection, and fruit counting in the agricultural sector. With the fact that field debris including weeds, grass, wild blueberry leaves, stems (with leaves, without leaves), and dirt are the major constraint for ensuring high fruit quality during harvesting. Esau et al., (2018) tested a dual fan blower system and experimented with four different blower fan speeds for investigating optimum blower fan speed to improve berry cleaning efficiency during harvesting. Esau (2019) also extended the research and determined the effect of debris removal brush on debris cleaning performance. Esau (2019) reported a brush bristle length of 120 mm performed better for debris cleaning however the shorter bristle length of the brush (shorter than 110 mm) contributed to significant debris buildup at the rear cross member. Esau (2019) recommended adjustment of the cleaning brush for shorter bristle lengths (shorter than 120 mm) to achieve optimum debris cleaning on the picking reel. However, no study has been conducted for adjusting of debris cleaning brush automatically using machine vision system. A CNN based debris detection system can be used for debris monitoring and assisting a feedback control system in order to adjust the debris cleaning brush automatically on the basis of the debris presence in side and rear conveyor. Therefore, a CNN based debris detection system can be a valuable addition in berry separation technology to improve the quality of the fruit.

CHAPTER 3: TRAINING AND OPTIMIZING A CNN FOR DEBRIS DETECTION DURING MECHANICAL WILD BLUEBERRY HARVESTING

Abstract

Wild blueberry (*Vaccinium angustifolium* Ait.) is an economically important horticultural crop in the Atlantic Provinces of Canada and Maine, USA. The continuous development of field management practices has changed the wild blueberry's productiveness through significant increases in plant densities, plant height, and fruit yield. This increased plant biomass causes additional leaf and stem debris to enter the fruit storage bins during mechanical harvesting, resulting in a reduction of fruit quality. CNN based automatic debris separation technology can be a viable solution to monitor the quality of berries during harvesting. Three CNNs (YOLOv3, YOLOv3-SPP, and YOLOv3-Tiny) were implemented and compared for developing the real-time debris detection system. An experimental dataset comprising of 1,000 images were collected from two commercially managed fields in central Nova Scotia. The image datasets were labeled in Darknet format to ensure the compatibility of detection models. A total of five classes including leaves, stems, green berries, ripe berries, and dirt were created and labeled in the images. The YOLOv3-SPP achieved the highest mAP (68.49%), YOLOv3 achieved the second-highest mAP (67.75%), YOLOv3-Tiny achieved the lowest mAP (60.36%) in the experiment. A total of five different data augmentation techniques (sharpening, brightness, contrast, gamma correction, saturation) were employed, and a combination of these techniques significantly improving the CNNs accuracy. YOLOv3-SPP achieved 73.03% and 74.38% of mAP on augmented datasets. The mAP of YOLOv3-SPP improved by 4.54%, and 5.89 % when trained and tested on T1, T2 datasets, respectively. The accuracy improved because data augmentation techniques were introduced new data and allowed models to learn from the different varieties of data. The

YOLOv3-SPP achieved slightly better accuracy than the YOLOv3 and YOLOv3-Tiny network in this experiment.

3.0 Introduction

Wild blueberry (*Vaccinium angustifolium* Ait.) is an economically important crop of Eastern Canada. The large majority of commercially managed fields (above 80%) are harvested mechanically following a two-year production cycle (PMRA, 2005). 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 side and rear conveyor on the harvester. The fan helps to remove debris from berries on the go while harvesting the crop (Esau et al., 2018). Continuous improvements in field management practices (i.e., application of fungicides, herbicides, fertilizers, pollination, and pruning, etc.) have increased plant biomass and fruit yield. (Esau et al., 2018). This increased plant biomass causes additional leaf and stem debris to enter the fruit storage bins during mechanical harvesting resulting in reduction of fruit quality (Esau et al., 2018).

Computer vision-based technologies are gaining popularity in different fields of agriculture (Granitto et al., 2002; Piedad et al., 2018; Tu et al., 2018). Zaman et al. (2008) 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 R^2 value of 0.99. Chang et al. (2012) 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 three co-occurrence matrices and selected textural features. However, a feature extraction approach needs feature engineering skills (LeCun et al., 2015) and inappropriate feature selection leads to an increase in false-positives and decreases the precision of classifiers (Chang et al., 2012; Rehman et al., 2018).

Deep learning (DL) techniques can learn features by themselves from the pixels of images where feature extraction and classification are taken care by the network (LeCun et al., 2015). Several researchers used Convolutional Neural Networks (CNNs) for estimating fruit quality (Cavallo et al., 2018; Wang et al., 2018, Jahanbakhshi et al., 2020). Wang et al. (2018) used two varieties of CNNs (ResNet and ResNeXt) for classifying sound and damaged highbush blueberries and achieved 88.44% and 87.84% accuracy, respectively. Jahanbakhshi et al. (2020) implemented a CNN based DL technique for classifying healthy and damaged sour lemons. The CNN outperformed traditional machine learning approaches including k-nearest neighbour, artificial neural network, Fuzzy, support vector machine, and decision tree. Schumann et al. (2019) also used four varieties of the YOLO (you only look once) family of CNNs for recognizing three stages wild blueberry maturity (unripe green, unripe red, and ripe blue) and achieved the best accuracy with 85.3% of mAP and 28.3 milliseconds inference time in the validation dataset. Researchers improved classification and detection accuracy of a CNN by optimizing hyperparameters resulted in saving of computational cost (Wang et al. 2019, Lu et al. 2019). However, the optimization of CNN is challenging and generally done by weight initialization, stochastic gradient descent optimization, batch normalization, shortcut connections, and data augmentation (Gu et al. 2018). Wang et al., (2019) 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, mirroring. These techniques were employed by several researchers for improving the overall learning procedure and performance (Zhang et al., 2017; Sladojevic et al., 2016). Zhang et al. (2019) 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. The proposed CNN achieved 5% higher accuracy compared to the state-of-the-art approaches: PCA + kSVM (Zhang & Wu, 2012), PCA + FSCABC algorithm (Zhang et al., 2014), WE + BBO (Wang et al., 2015), FRFE + BPNN (Wang et al., 2016), FRFE + IHGA (Lu & Li, 2017). Kang and Chen, (2019) applied image augmentations techniques including image flipping, color saturation, contrast, and brightness adjustment, and translation to minimize the unbalanced distribution in the training data. A CNN based recognition system was developed by Sladojevic et al. (2016) 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 (Perez and Wang, 2017; Namozov and Im Cho, 2018).

Improving wild blueberry fruit quality has become increasingly important to producers due to the tightening profit margin facing the industry. Therefore, A CNN based automatic debris separation technology can be a viable solution to monitor the quality of berries during harvesting.

3.1 Materials and Methodology

3.1.1 Data Collection and Preparation

Video data was collected from two different commercial fields in central Nova Scotia including the Debert site (45.4418°N, 63.4496°W) and the East Mines site (45.42713°N, -63.48186°W) on 15th August 2019 through the 31st August 2019 from morning to evening (Figure 3-1).



Figure 3-1. Location of fields in google map

The videos were captured using two Logitech C920 webcam cameras (Logitech International S.A., Lausanne, Switzerland) mounted before and after the blower fan on the Doug Bragg Enterprises Ltd. (DBE) commercial mechanical wild blueberry harvester. The cameras were connected via two Universal Serial Bus (USB) version 2.0 active extension cables, each measuring 2 m in length to an Intel® Core™ i5-4300U @ 1.90 GHz and an Intel® Core™ i5-8250U @ 1.60 GHz central processing unit (CPU) based computers (Hewlett-Packard, Palo Alto, California, United States, and Dell Incorporation, Round Rock, Texas, United States) operated with Windows 10 (Microsoft Corp. Redmond, Washington, United States) 64-bit system. Cameras were mounted at 34.00 cm and 48.39 cm height above side and rear conveyor respectively and pointed downward by 45 and 90 degrees to the side and rear conveyors, respectively. The videos were recorded using

acquisition software Logitech Capture (Logitech International S.A., Lausanne, Switzerland) during harvesting. The camera resolution was set to 1280x720 (Horizontal x Vertical) pixels at 25 frames per second while the autofocus was turned off and image adjustment and antiflicker were turned on to ensure no blurry video acquisition. Video frames were extracted at 1s intervals using a 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. The blurry and empty frames were removed manually after extraction, and all the extracted frames were renamed to prepare a parent dataset. The dataset, consisting of 1000 images, was generated through the random selection of images from the parent dataset. The dataset was labelled using custom software developed with the Lazarus compiler (<https://www.lazarus-ide.org/>). Ripe berries, green berries, leaves, stems, and dirt were labelled based on classifiable appearance in the images (Figure 3-2).

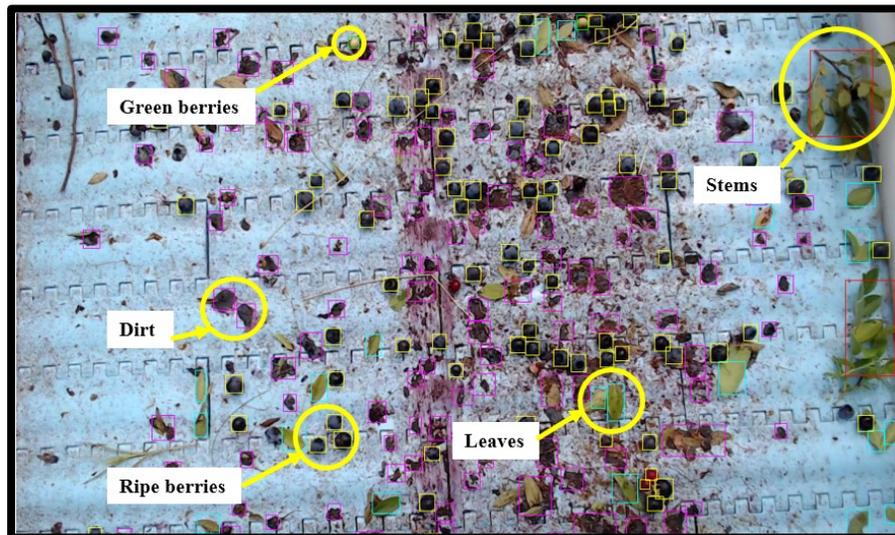


Figure 3-2. Image labeling using custom software developed with the Lazarus compiler

3.1.2 Image Augmentation

Images of the dataset were augmented using five different color augmentation techniques such as gamma correction, sharpness, brightness, contrast, saturation. Gamma correction increases

or decreases the luminance value in an image (Jiang et al., 2015). Six different gamma corrections (0.70, 0.80, 0.90, 1.10, 1.20 and 1.30) were used (Table 3-1). The image variation was ensured by setting gamma correction 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. The Sharpness enhancement improves the edges of objects in an image and makes sensitive to detection (Khan et al., 2020). Image sharpness was improved by setting the strength value to 10 (Table 3-1). Brightness enhancement increases the pixel value of all channels of an image evenly (Huang et al., 2019). The brightness of images was enhanced by the adding factor of 20 shown in Table 3-1. Contrast enhancement stretches or compresses the range of brightness value of an image and helps to stand out the target objects from the background (Huang et al., 2019). The contrast of images was enhanced by the factor of 20 shown in (Table 3-1). The saturation enhancement intensifies the color of an image make it colorful and vibrant. The saturation of images was enhanced by a factor of 15 shown in (Table 3-1). Figure 3-3 showed a sample of normal and augmented images in the dataset.

Table 3-1. Description of data augmentation techniques

| Dataset Name | Augmentation techniques | Subset images | Total Images |
|---------------------|--------------------------------|----------------------|---------------------|
| Augmentation-1 | Gamma correction = 0.70 | 200 | 1000 |
| | Gamma correction = 0.80 | 200 | |
| | Gamma correction = 0.90 | 200 | |
| | Gamma correction = 1.10 | 200 | |
| | Gamma correction = 1.20 | 200 | |
| Augmentation-2 | Sharpness = 10 | 200 | 1000 |
| | Brightness = 20 | 200 | |
| | Contrast = 20 | 200 | |
| | Gamma correction = 1.30 | 200 | |
| | Saturation = 15 | 200 | |
| Augmentation-3 | Augmentation-1 | 1000 | 2000 |
| | Augmentation-2 | 1000 | |

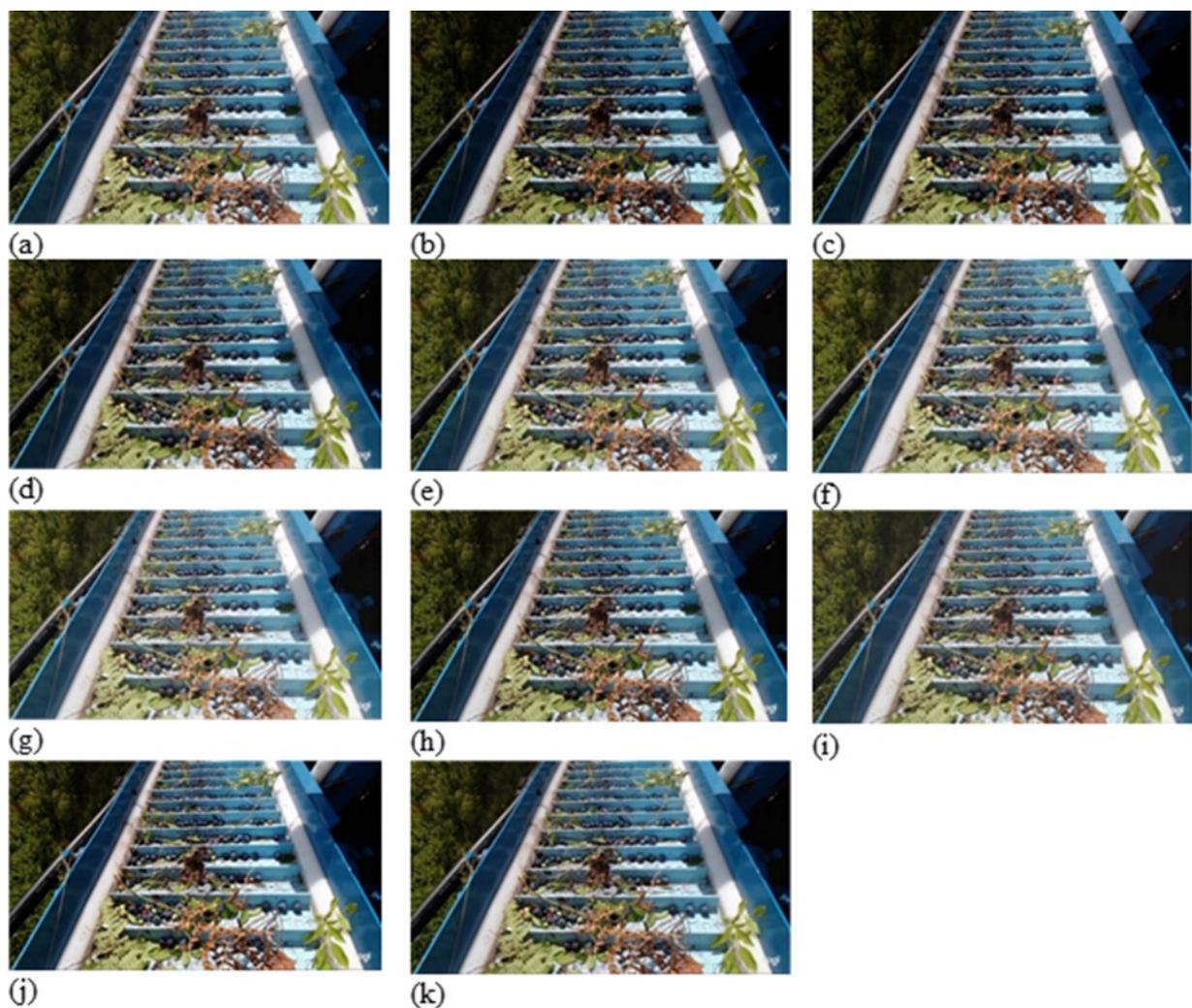


Figure 3-3. Image augmentation techniques (a) Original image (no augmentation) (b) gamma correction = 0.70 (c) gamma correction = 0.80 (d) gamma correction = 0.90 (e) gamma correction = 1.10 (f) gamma correction = 1.2 (g) gamma correction = 1.30 (h) sharpening = 10 (i) brightness = 20 (j) contrast = 20 (k) saturation = 15

The T1 dataset was prepared by mixing the original dataset with the Augmentation-2 dataset. Similarly, the T2 dataset was prepared by mixing the original dataset with Augmentation-1 and Augmentation-2 dataset is shown in Table 3-2. Ninety percent of the image of datasets was used for training and 10% was used for validation (Table 3-3). The training and testing dataset ratio (9:1) was also reported by previous study. Schumann et al. (2019) used 90% images for training and the remaining 10% for testing YOLOv3 and successfully detected wild blueberry maturity (unripe green, unripe red, and ripe blue) with higher detection accuracy (mAP: 85.3%).

Table 3-2. Description of original and augmented dataset

| Dataset Name | Dataset description | Number of images |
|-----------------|---|------------------|
| No augmentation | Original dataset | 1000 |
| T1 | Original dataset + Augmentation-2 | 2000 |
| T2 | Original dataset + Augmentation-1 + Augmentation-2 | 3000 |

Table 3-3. Training and testing dataset

| Dataset Name | Training Image | Testing Image |
|-----------------|----------------|---------------|
| No augmentation | 900 | 100 |
| T1 | 1800 | 200 |
| T2 | 2700 | 300 |

3.1.3 You Only Look Once Version 3 (YOLOv3) Family

YOLOv3 is a CNN based object detector developed by Redmon & Farhadi (2018) using the darknet-53 framework. The network consists of 53 convolutional layers that have been improved from the previous framework Darknet-19. In a common object in context (COCO) dataset, YOLOv3 performed 3.8 times faster than RetinaNet and 3 times faster than the Single Shot MultiBox Detector (Redmon & Farhadi 2018). YOLOv3 network is faster than the Faster R-CNN because it doesn't use a proposal region for detecting objects. YOLOv3 generates bounding box coordinates and probabilities of each class directly through regression. YOLOv3 divides an input image into $S \times S$ grids and when a center of a target object falls in any cell of the grid, that cell is responsible for detecting target objects. It detects objects at three different scales of the bounding box and each of the bounding boxes has five components (x, y, w, h, c) . Coordinates (x, y) represent the center, and (w, h) represents the height and width of a bounding box where C represents a confidence score. The confidence score represents the probability of having an object in a bounding box. The confidence score is calculated by (Eq. 3-1).

$$C = Pr(class_i) * IoU_{pred}^{truth} \quad (3-1)$$

The $Pr(class_i)$ determines whether the center point of leaves, stems, ripe berries, green berries, and dirt falls in the cell of a grid. The $Pr(class_i)$ is 1 when the center point of an object is present in a grid cell otherwise, it is zero. The IoU_{pred}^{truth} is the ratio of the predicted bounding box of leaves, stems, ripe berries, green berries, and dirt with the ground truth bounding box. The IoU value confirms the detection of leaves, stems, ripe berries, green berries, and dirt were weather true positive or false positive. When the predicted bounding box has an IoU greater than the given IoU threshold, the prediction is considered as a true positive, otherwise, it is considered a false positive. Then a Non-Maximum Suppression (NMS) method is used to keep the most confident boxes and eliminate duplicate and redundant bounding boxes of the same objects by setting a confidence threshold. The $Pr(class_i|object)$ determines the class probability of leaves, stems, ripe berries, green berries, and dirt. Finally, conditional class probabilities are multiplied with individual bounding box confidence values to determine the classification score of each object, which is calculated by using (Eq. 3-2)

$$Pr(class_i|object) * Pr(object) * IoU_{pred}^{truth} = Pr(class_i) * IoU_{pred}^{truth} \quad (3-2)$$

YOLOv3 (Redmon and Farhadi, 2018) is the full-size standard version of YOLOv3. YOLOv3-SPP is the enhanced and revised version of YOLOv3. YOLOv3-Tiny is the scaled-down network of YOLOv3 which is less accurate but faster than YOLOv3 (Redmon and Farhadi, 2018).

3.1.4 YOLO Parameter Settings

The input image size was set to 1280x736 pixels because image width and height must be divisible by 32 for YOLOv3 training (Table 3-4). Another reason is the camera resolution 1280x736 pixels with 34 cm and 48.39 cm camera depth would cover the maximum field of view at the side and rear conveyor, respectively. Training batch size represented the number of images

that would be processed in one batch while $\frac{batch}{subdivision}$ determined the number of images that would be processed by the GPU at a time (Table 3-4). Max batches represented the number of training iterations. Models were trained up to 12000 iterations (Table 3-4). The initial learning rate confirmed the rate at which the model would learn; lower the better. The initial learning rate controls error each time when model weights are updated. The steps sizes indicated that the initial learning rate (0.001) would be multiplied with scales (0.20, 0.10, 0.10) at 7000th, 8000th, and 9000th iterations (Table 3-4). Other parameters of YOLOs were kept as the default in the configuration files.

Table 3-4. Initialization parameters of the YOLO networks

| | |
|-----------------------|------------------|
| Size of input images | 1280x736 |
| Batch size | 64 |
| Subdivision | 16 |
| Max batches | 12000 |
| Initial learning rate | 0.001 |
| Steps | 7000,8000,9000 |
| Scales | 0.20, 0.10, 0.10 |

3.1.5 YOLO Training and Testing

Three YOLO networks (YOLOv3, YOLOv3-Tiny, and YOLOv3-SPP) were successfully trained and tested on a GeForce RTX™ 2080 Ti @ 1665 MHz graphics processing unit (GPU) card (NVIDIA, Santa Clara, CA) computer installed with 64-bit Ubuntu 16.04 (Canonical Group Ltd, London, UK). The networks were trained and tested on four different datasets using transfer learning. The selected training parameters were kept the same for all networks throughout the training and testing process. A total of 13 different binary weight files were produced from each network. The twelve different weight files were produced from 1000th to 12000th iteration at every 1000 iterations and the remaining one weight file (yolo_best.weight) was generated and saved automatically by the YOLO networks which yielded the best Mean average precision (mAP)

during training and testing. Then these YOLO weights were used to record the precision, recall, mean average precision, and F1-score values for the networks.

3.1.6 Model Evaluation

The detection performance models were evaluated by the precision, recall, mAP, and F1-score value. Precision is the ratio of true detections of target classes to the identified detections and is calculated by (Eq. 3-3)

$$\text{Precision} = \frac{TP}{TP+FP} \quad (3-3)$$

Recall is the ratio of true detection of target classes among all the ground truth detections and is calculated by (Eq. 3-4)

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3-4)$$

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 YOLO fails to detect objects.

F1-score is the harmonic mean of precision and recall and is calculated by (Eq. 3-5)

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3-5)$$

Mean average precision is integral to all the precision values of all target classes and is calculated by equation (Eq. 3-6)

$$\text{mAP} = \int_0^1 p(o) do \quad (3-6)$$

3.1.7 Statistical Analysis

One-way analysis of variance (ANOVA) performed using Minitab 17 (Minitab Inc. NY, USA) with a 5% ($\alpha = 0.05$) level of significance to check the effect of models (YOLOv3, YOLOv3-SPP, and YOLOv3-Tiny) on accuracies such as F1-score, average precisions, and mAP. The testing images of datasets were divided into five subsets and used for the evaluation of models. The accuracies obtained from the models were recorded. Tukey method and T-test were performed to find the mean differences.

3.0 Result and Discussion

Models were evaluated at 0.5 IoU and 0.25 confidence threshold. Table 3-5 showed detection results of YOLOv3-SPP, YOLOv3, and YOLOv3-Tiny when trained and tested on a non augmented dataset. The average precision of targets classes were shown in Table 3-6.

Table 3-5. Precision, Recall, F1-score, and mAP (%) of the YOLOv3-SPP, YOLOv3 and YOLOv3-Tiny under 0.5 IoU and 0.25 confidence threshold

| Models | Precision | Recall | F1-score | mAP (%) |
|-------------|-----------|--------|----------|---------|
| YOLOv3 | 0.85 | 0.54 | 0.66 | 67.75 |
| YOLOv3-SPP | 0.83 | 0.55 | 0.66 | 68.49 |
| YOLOv3-Tiny | 0.79 | 0.50 | 0.61 | 60.36 |

Table 3-6. Average precision (%) of YOLOv3, YOLOv3-SPP and YOLOv3-Tiny

| Models | Leaves | Stems | Green Berries | Ripe Berries | Dirt |
|-------------|--------|-------|---------------|--------------|-------|
| YOLOv3 | 69.22 | 55.66 | 72.02 | 80.56 | 61.31 |
| YOLOv3-SPP | 72.00 | 56.14 | 74.90 | 79.18 | 60.24 |
| YOLOv3-Tiny | 65.07 | 51.88 | 52.63 | 77.33 | 54.88 |

The overall accuracy (F1-score), mAP, and average class precision of one model were compared with other models (Table 3-7). The P-value (<0.001) and (0.004) indicated that means of F1-score and mAP of models were significantly different (Table 3-7). Tukey results showed mean F1-score

of YOLOv3-Tiny was significantly different than the mean F1-score of YOLOv3 and YOLOv3-SPP ($P < 0.001$). However, there was no significant difference between the mean F1-score of YOLOv3 and the mean F1-score of YOLOv3-SPP.

Table 3-7. Comparison of F1-score of YOLOv3, YOLOv3-SPP and YOLOv3-Tiny using the Tukey method at 5% ($\alpha = 0.05$) level of significance

| Models | Mean (F1-score) | P-Value | Mean (mAP) | P-Value |
|-------------|-----------------|---------|------------|---------|
| YOLOv3 | 0.65A | <0.001 | 68.57A | 0.004 |
| YOLOv3-SPP | 0.66A | | 69.12A | |
| YOLOv3-Tiny | 0.61B | | 61.92B | |

Table 3-8 showed mAP of the leaves of YOLOv3-SPP (72.34%) was significantly different than YOLOv3-Tiny (65.48%) but not significantly different than YOLOv3 (69.62%) ($P = 0.003$). However, mAP of stems of YOLOv3 (58.53%), YOLOv3-SPP (57.48%), and YOLOv3-Tiny (56.33%) were not significantly different ($P = 0.958$). Table 3-9 showed mAP of the green berry of YOLOv3 (72.26%) was significantly different than YOLOv3-Tiny (55.13%) ($P < 0.001$) but not significantly different than YOLOv3-SPP (75.45%). On the other hand, the mAP of ripe berries of YOLOv3 (80.71%), YOLOv3-SPP (79.57%), and YOLOv3-Tiny (77.52%) were not significantly different ($P = 0.170$). Table 3-10 showed mAP of dirt of YOLOv3 (61.70%) was not significantly different than YOLOv3-SPP (60.80%) but significantly different than YOLOv3-Tiny (54.11%) ($P = 0.001$)

Table 3-8. Comparison of class average precisions (%) of YOLOv3, YOLOv3-SPP and YOLOv3-Tiny for leaves and stems using the Tukey method at 5% ($\alpha = 0.05$) level of significance

| Models | Mean (Leaves) | P-value | Mean (Stems) | P-value |
|-------------|---------------|---------|--------------|---------|
| YOLOv3 | 69.62AB | 0.003 | 58.53A | 0.958 |
| YOLOv3-SPP | 72.34A | | 57.48A | |
| YOLOv3-Tiny | 65.48B | | 56.33A | |

Table 3-9. Comparison of class average precision (%) of YOLOv3, YOLOv3-SPP and YOLOv3-Tiny for green berries and ripe berries using the Tukey method at 5% ($\alpha = 0.05$) level of significance

| Models | Means (Green Berries) | P-Value | Mean (Ripe Berries) | P-Value |
|---------------|------------------------------|----------------|----------------------------|----------------|
| YOLOv3 | 72.26A | 0.001 | 80.71A | 0.170 |
| YOLOv3-SPP | 75.45A | | 79.57A | |
| YOLOv3-Tiny | 55.13B | | 77.52A | |

Table 3-10. Comparison of class average precision of YOLOv3, YOLOv3-SPP and YOLOv3-Tiny for dirt using the Tukey method at 5% ($\alpha = 0.05$) level of significance

| Models | Mean (Dirt) | P-value |
|---------------|--------------------|----------------|
| YOLOv3 | 61.70A | 0.001 |
| YOLOv3-SPP | 60.80A | |
| YOLOv3-Tiny | 54.11B | |

The detection of leaves, stems, green berries, ripe berries, and dirt of models was visually compared in Figure 3-4 after training models in non augmented dataset. The detection of leaves, stems, green berries, ripe berries, and dirt on images were represented as 1, 2, 3, 4, and 5, respectively.

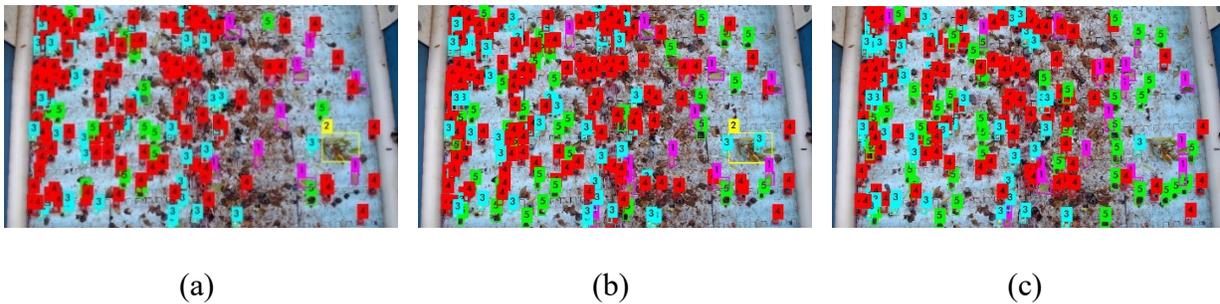


Figure 3-4. Detection comparison of YOLOv3 (a), YOLOv3-SPP(b), and YOLOv3-Tiny(c) trained on non augmented dataset.

3.2.1 Effects of Data Augmentation on Accuracy

The effect of data augmentation on the accuracy of YOLOv3-SPP and YOLOv3 was explored. Table 3-11 showed the detection results of YOLOv3-SPP and YOLOv3 when trained and tested on augmented datasets (T1 and T2).

Table 3-11. Precision, Recall, F1-score, and mAP (%) of the YOLOv3-SPP and YOLOv3 under 0.5 IoU and 0.25 confidence threshold

| Dataset Name | Models | Precision | Recall | F1-score | mAP (%) |
|--------------|------------|-----------|--------|----------|---------|
| T1 | YOLOv3-SPP | 0.87 | 0.53 | 0.66 | 73.03 |
| | YOLOv3 | 0.90 | 0.46 | 0.61 | 71.26 |
| T2 | YOLOv3-SPP | 0.86 | 0.54 | 0.66 | 74.38 |
| | YOLOv3 | 0.89 | 0.47 | 0.62 | 72.87 |

The class accuracy of models was also explored after testing models on T1 and T2 dataset. Table 3-11 showed detection results of leaves, stems, green berries, ripe berries, and dirt. The average precision of leaves, stems, ripe berries, and dirt was improved 5.60%, 10.91%, 4.40%, 3.86% when YOLOv3-SPP trained and tested on T1 dataset (Table 3-12). Similarly, the average precision of leaves, stems, ripe berries, and dirt was improved by 9.11%, 18.41%, 2.81%, 3.41% when YOLOv3-SPP trained, and tested on the T2 dataset (Table 3-12). On the other hand, the average precision of leaves, stems, green berries, ripe berries, and dirt was improved by 4.93%, 8.14%, 0.24%, 2.35%, 1.88% respectively when YOLOv3 trained and tested on T1 dataset and improved 8.18%, 9.65%, 3.52%, 1.95%, 2.30% respectively when trained and tested on T2 dataset (Table 3-12).

Table 3-12. Average precision (%) of YOLOs with data augmentation under 0.5 IoU and 0.25 confidence threshold

| Dataset Name | Models | Leaves | Stems | Green Berries | Ripe Berries | Dirt |
|--------------|------------|--------|-------|---------------|--------------|-------|
| T1 | YOLOv3-SPP | 77.60 | 67.05 | 72.82 | 83.58 | 64.10 |
| | YOLOv3 | 74.15 | 63.80 | 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 |

3.2.2 T-test on Testing Images of T1 Dataset

The result showed mean F1-score of YOLOv3-SPP (0.64) was significantly higher ($P = 0.003$) than the mean F1-score of YOLOv3 (0.61). Although mAP of YOLOv3-SPP (75.56%) was not significantly different than mAP of YOLOv3 ($P = 0.150$) (Table 3-13)

Table 3-13. Comparison of F1-score and mAP (%) of YOLOv3-SPP and YOLOv3 using the t-test at 5% ($\alpha = 0.05$) level of significance

| Models | Mean (F1-score) | P-value | Mean (mAP) | P-value |
|------------|-----------------|---------|------------|---------|
| YOLOv3-SPP | 0.64 | 0.003 | 75.56 | 0.150 |
| YOLOv3 | 0.61 | | 71.84 | |

Table 3-14 showed mAP of the leaf of YOLOv3 (74.22%) was significantly different than YOLOv3-SPP (81.84%) ($P < 0.001$). The YOLOv3-SPP yielded significantly higher mAP (81.84%) for leaves class than the YOLOv3 (74.22%). However, mAP of stems of YOLOv3 (75.77%) and YOLOv3-SPP (65.69%) was not significantly different ($P = 0.292$) (Table 3-15). Similarly, mAP of green berries (74.84%), ripe berries (83.32%), and dirt (62.03%) of YOLOv3-SPP were not significantly different than mAP of green berries (72.90%), ripe berries (83.05%), and dirt (63.36%) of YOLOv3 respectively (Table 3-15) (Table 3-16).

Table 3-14. Comparison of class average precisions (%) of YOLOv3-SPP and YOLOv3 for leaves and stems using the t-test at 5% ($\alpha = 0.05$) level of significance

| Models | Mean (Leaves) | P-value | Mean (Stems) | P-value |
|------------|---------------|---------|--------------|---------|
| YOLOv3-SPP | 81.84 | <0.001 | 75.77 | 0.292 |
| YOLOv3 | 74.22 | | 65.69 | |

Table 3-15. Comparison of class average precision of YOLOv3-SPP and YOLOv3 for green berries and ripe berries using the t-test at 5% ($\alpha = 0.05$) level of significance

| Models | Mean (Green Berries) | P-value | Mean (Ripe Berries) | P-value |
|------------|----------------------|---------|---------------------|---------|
| YOLOv3-SPP | 74.84 | 0.338 | 83.32 | 0.762 |
| YOLOv3 | 72.90 | | 83.05 | |

Table 3-16. Comparison of class average precision of YOLOv3-SPP and YOLOv3 for dirt using the t-test at 5% ($\alpha = 0.05$) level of significance

| Models | Mean (Dirt) | P-value |
|------------|-------------|---------|
| YOLOv3-SPP | 62.03 | 0.567 |
| YOLOv3 | 63.36 | |

3.2.3 T-test on Testing Images of T2 Dataset

The result showed mean F1-score of YOLOv3-SPP was significantly higher than the mean F1-score of YOLOv3 ($P < 0.001$). YOLOv3-SPP resulted in 0.65 of mean F1-score while YOLOv3 resulted in 0.62 of mean F1-score. Although mAP of YOLOv3-SPP (75.71%) was not significantly different than mAP of YOLOv3 (73.19%) ($P = 0.076$) (Table 3-17)

Table 3-17. Comparison of F1-score and mAP (%) of YOLOv3-SPP and YOLOv3 using the t-test at 5% ($\alpha = 0.05$) level of significance

| Models | Mean (F1-score) | P-value | Mean (mAP) | P-value |
|------------|-----------------|---------|------------|---------|
| YOLOv3-SPP | 0.65 | <0.001 | 75.71 | 0.076 |
| YOLOv3 | 0.62 | | 73.19 | |

The mAP of the leaves (80.19%), green berries (74.89%), ripe berries (82.55%), and dirt (65.33%) of YOLOv3-SPP was not significantly different than mAP of the leaves (77.46%), green berries (74.14%), ripe berries (82.64%) and dirt (63.80%) of YOLOv3 (Table 3-18) (Table 3-19)

(Table 3-20). However, mAP of stems (75.63%) of YOLOv3-SPP was significantly higher than mAP of stems (66.08%) of YOLOv3 ($P = 0.025$) (Table 3-18). The YOLOv3-SPP yielded significantly higher mAP (75.63%) for stems class than YOLOv3 (66.08%) (Table 3-18).

Table 3-18. Comparison of class average precisions (%) of YOLOv3-SPP and YOLOv3 for leaves and stems using the t-test at 5% ($\alpha = 0.05$) level of significance

| Models | Mean (Leaves) | P-value | Mean (Stems) | P-value |
|------------|---------------|---------|--------------|---------|
| YOLOv3-SPP | 80.19 | 0.096 | 75.63 | 0.025 |
| YOLOv3 | 77.46 | | 66.08 | |

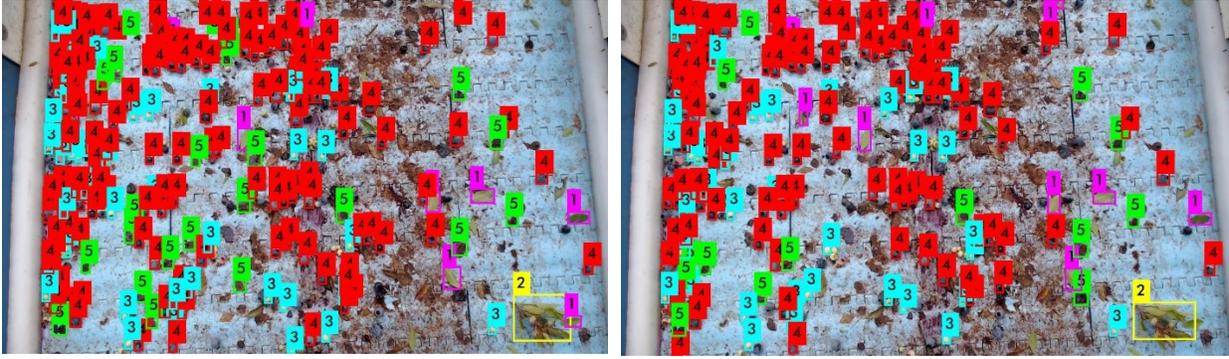
Table 3-19. Comparison of class average precisions (%) of YOLOv3-SPP and YOLOv3 for green berries and ripe berries using the t-test at 5% ($\alpha = 0.05$) level of significance

| Models | Mean (Green Berries) | P-value | Mean (Ripe Berries) | P-value |
|------------|----------------------|---------|---------------------|---------|
| YOLOv3-SPP | 74.89 | 0.854 | 82.55 | 0.938 |
| YOLOv3 | 74.14 | | 82.64 | |

Table 3-20. Comparison of class average precision of YOLOv3-SPP and YOLOv3 for dirt using the t-test at 5% ($\alpha = 0.05$) level of significance

| Models | Mean (Dirt) | P-value |
|------------|-------------|---------|
| YOLOv3-SPP | 65.33 | 0.265 |
| YOLOv3 | 63.80 | |

Figure 3-5 showed the detection results of two models YOLOv3-SPP (mAP-73.03%) and YOLOv3-SPP (mAP-74.38%) after training YOLOv3-SPP on T1 and T2 dataset. Both models successfully detected most target classes. However, models struggled to detect dirt at 0.5 IoU and 0.25 confidence threshold. The probable reason may be the lower average precision of dirt classes of both models. The variation of dirt shape in the dataset is likely the reason for lower accuracy across the models. Since the lower confident bounding boxes (less than 0.5 IoU) were screen models by default, experimenting on the effect of different IoU and confidence scores on detection accuracy can be a valuable addition to this experiment in the future.



(a) YOLOv3-SPP (T3)

(b) YOLOv3-SPP (T2)

Figure 3-5. Detection results of YOLOv3-SPP on a testing image.

The detection of leaves, stems, green berries, ripe berries, and dirt on images were represented as 1, 2, 3, 4, and 5 respectively.

3.0 Conclusions

Results concluded that YOLOv3-SPP and YOLO-v3 both performed well in non-augmented and augmented datasets. However, Tukey MMC showed mean F1-scores of YOLOv3-SPP (0.64 and 0.65) were significantly higher than YOLOv3 (0.61 and 0.62) on testing images of T1 and T2 dataset (Table 3-13) (Table 3-17). The mAPs of YOLOv3 and YOLOv3-SPP were not significantly different (Table 3-14) (Table 3-17) but YOLOv3-SPP (mAP: 73.03%; mAP: 74.38%) yielded slightly higher mAP than YOLOv3 (mAP: 71.26%; mAP: 72.87%) on both T1 and T2 datasets (Table 3-11). However, YOLOv3 and YOLOv3-SPP achieved quite similar mAP in different experiments. The most likely reason for yielding similar results could be the sensitivity to small object detection of YOLO networks. Pham et al., (2020) experimented with different YOLO networks for detecting various small objects (lower than 10 pixels) on aerial and satellite images. They also received equivalent mAP score 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%). However, detection accuracy of YOLO was improved by adding an extra residual block detection layer and an up sampling layer in the YOLO network. Results showed the overall performance (F1-scores) of YOLOv3-SPP models were slightly higher than YOLOv3 models. All the target classes were labelled based on visual appearance where stems attached with leaves were labelled as stems instead of their own class. This strategy of labelling 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. In this study, YOLOv3-SPP models achieved comparatively better results than models which can be selected and implemented for the debris detection system.

CHAPTER 4: DEVELOPMENT OF A REAL-TIME DEBRIS DETECTION SYSTEM (HARDWARE & SOFTWARE) FOR MECHANICAL WILD BLUEBERRY HARVESTER

Abstract

Wild blueberry fruit quality has always been economically important to wild blueberry producers due to the competitive global market and tightening profit margins currently facing the industry. Continuous development of field management practices (i.e., application of fungicides, herbicides, fertilizers, pollination, and pruning, etc.) has increased plant biomass. This increased plant biomass significantly reduces fruit quality which highlights the need to develop a real-time automatic debris separation system during harvesting. Three different hardware and two CNN models were implemented to detect leaves, stems, ripe berries, green berries, and dirt on images and compared with the detection time of each combination of model and hardware for investigating a real-time performance. Nine different IoU and confidence thresholds (0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, and 0.9) were examined for determining the effect of threshold values on overall detection of target classes. The result showed that Hardware-4 and Model-2 yielded the fastest detection time (33.30 ms) and the highest average frame rate (30.03 FPS). Model-1 executed on Hardware-1 resulted in the longest detection time (12126.90 ms) and lowest average frame rate (0.80). Hardware-3 and Hardware-4 both achieved lower processing time 831.20 ms and 33.30 ms respectively than the camera passing time of a scene at side conveyor (2276.67 ms) and rear conveyor (1105.21 ms). Model-1 and Model-2 achieved 71.10% and 72.36% of mAP respectively when validated under 0.10 IoU and 0.25 confidence threshold. Model-1 and Model-2 reached maximum mAP of 85.90% and 86.10% respectively at 0.10 IoU and 0.10 confidence threshold. Therefore, a YOLOv3-SPP based debris detection system could meet the demand for real-time debris detection with greater accuracy.

4.0 Introduction

Wild blueberry (*Vaccinium angustifolium* Ait.) is a commercially managed native fruit crop in northeastern North America. Plant biomass and fruit yields have increased significantly due to improved field management practices over the past decades (Esau et al., 2018). As a result of increased biomass, additional blueberry leaves, stems, and foreign debris are harvested and transferred into the berry storage bins, causing a negative impact on fruit quality at the processing facilities (Esau et al., 2018). Esau et al. (2018) tested four different blower fan speeds (0 ms^{-1} , 14 ms^{-1} , 18 ms^{-1} , 23 ms^{-1}) and successfully removed 98.8% and 98.6% of debris (weight basis) using 23 ms^{-1} fan speed under low and high plant moisture conditions respectively. 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.

There are many types of automated technologies used in agriculture today, among them machine learning and deep learning are becoming the core of many of the latest technological developments in precision agriculture due to their automatic decision-making ability (Koirala et al., 2019; Tian et al., 2019a). However, the traditional machine learning approach cannot learn features automatically from image pixels. Feature extraction requires manual feature engineering skills (LeCun et al., 2015). Inappropriate selection of features decreases accuracy and increases the computation time for the classifier (Chang et al. 2012). Deep learning (DL) approaches are powerful for automatic feature extraction while classification is taken care by the network itself (LeCun et al., 2015). Several researchers used Convolutional Neural Network (CNN) based DL approaches for estimating fruit quality (Cavallo et al., 2018; Wang et al., 2018, Jahanbakhshi et al., 2020). Wang et al. (2018) classified sound and damaged highbush blueberry with 88.44% and 87.84% precision respectively using two modern CNN architectures ResNet and ResNeXt,

originally developed by He et al. (2016) and Xie et al. (2017) respectively. An automatic lettuce quality evaluation system was developed by Cavallo et al. (2018) using a deep CNN. The system could identify color distorted lettuce through the packaging with 83% classification accuracy.

CNN can also be implemented for classification as well as detection at the same time for understanding the location of objects within an image. The image can then be classified into two categories such as two-stage detection methods and the single-stage detection methods. Two-stage detection networks such as Faster Region-based Convolutional Neural Network (R-CNN) uses region proposal network (RPN) for generating proposals with multiple scales and aspect ratios and then feeds them to a Fast R-CNN detector for final class prediction with localization (Ren et al., 2015). On the other hand, a single-stage detection CNN network such as YOLOv3 uses a fully convolutional neural network (CNN) for generating bounding boxes and class probabilities (Redmon & Farhadi 2018). Some researchers found that a one-stage object detector outperformed a two-stage detection network (Tian et al., 2019b, Liu et al. 2020). An automatic tomato fruit detection system was developed by Liu et al., (2020) using YOLOv3 called YOLO-Tomato that achieved 96.40 % of average precision and performed than Faster R-CNN. Tian et al., (2019b) also implemented YOLOv3, incorporating with DenseNet developed by Huang et al., (2017) for detecting young, expanding, and ripe apples in the orchard. The proposed YOLOv3 achieved 81.70% of accuracy (F1-score) and performed better than Faster R-CNN with VGG16 net. However, DL either one-stage object detector or two-stage detection networks require plenty of computational power due to the processing of a large dataset. This limitation in DL approaches influenced several researchers to invigilate rich computing resources for solving various classification and detection problems in agriculture (Andrea et al., 2017, Shadrin et al., 2019, Koirala et al., 2019). Huang et al., (2017) compared tomato average detection time of YOLOv3

and Faster R-CNN using a computer configured with Intel i5 (Santa Clara, CA, USA), 64-bit 3.30 GHz quad-core central processing unit (CPU), and a NVIDIA GeForce GTX 1070Ti graphics processing unit (GPU). The author observed YOLOv3 took on average 0.045 seconds for detecting a single image and achieved 94.06% of average precision while Faster R-CNN had a longer detection time than YOLOv3 which was 177 milliseconds (ms) slower than YOLOv3. Tian et al., (2019a) also compared average detection time of young, expanding, and ripe apple images feeding to YOLOv3 and Faster R-CNN and observed YOLOv3 detected 8.864 times faster than Faster R-CNN on a NVIDIA Tesla V100 server. Instead of using rich computing resources another researcher Liu et al., (2019) used a NVIDIA TX2 low-power embedded computer board for developing a real-time broken corn monitoring system addressing power and space limitation on the corn harvester. They implemented YOLOv3–tiny architecture as a backbone of their system and achieved 89.77% of accuracy with 10 frames per second (FPS). Researchers also used YOLO on embedded Hardware, NVIDIA TX2, Nvidia Jetson Nano in different fields of agriculture for weed detection (Partel et al., 2019), Fish Detection (Liu et al., 2018b), and fruit detection (Mazzia et al., 2020) with higher success rates. However, very few researchers implemented CNN on the central processing unit (CPU) based computers in agriculture. Quiroz et al., (2020) developed legacy blueberries recognition system equipped with AMD Ryzen5 2500U CPU-based computer, and the system achieved 86% of accuracy on testing images.

The modern wild blueberry harvesters were designed for working with an agricultural tractor. The modern DBE harvesters are tractor driven and operated by the hydraulic system which is controlled from the tractor cabin using an electric controller. The electric controller is powered by a 12V direct current battery that could provide maximum 120 W power. Thus, space and power limitations in the tractor cabin are the major constraint to install additional computing resources

inside of it. This constrains leads to investigate appropriate computing resources for developing a real-time debris detection system. Therefore, finding an appropriate and rich computing source and selection of appropriate CNN could help hardware and software development of real-time debris separation systems for separating debris in real-time to improve fruit quality.

4.1 Material and Methodology

4.1.1 Video Acquisition Hardware Development

Video acquisition hardware consisted of two Logitech C920 webcam cameras (Logitech International S.A., Lausanne, Switzerland). The cameras were mounted on the side and rear conveyor of Doug Bragg Enterprises Ltd. (DBE) mechanical wild blueberry harvester at different heights to capture 8-bit RGB 1280x720 (Horizontal x Vertical) pixels video. The side and rear cameras were connected to an Intel® Core™ i5-4300U CPU @ 1.90 GHz central processing unit-based laptop using a built-in 1.5 m Universal Serial Bus (USB) version 2.0 camera cable. The camera was mounted at the side conveyor was pointed downward at a 45° angle. The camera height at the side conveyor was adjusted to 34.00 cm to get the maximum field of view. The rear camera was mounted at 49.80 cm working depth above the rear conveyor and pointed downward at a 90° angle. (Figure 4-1).

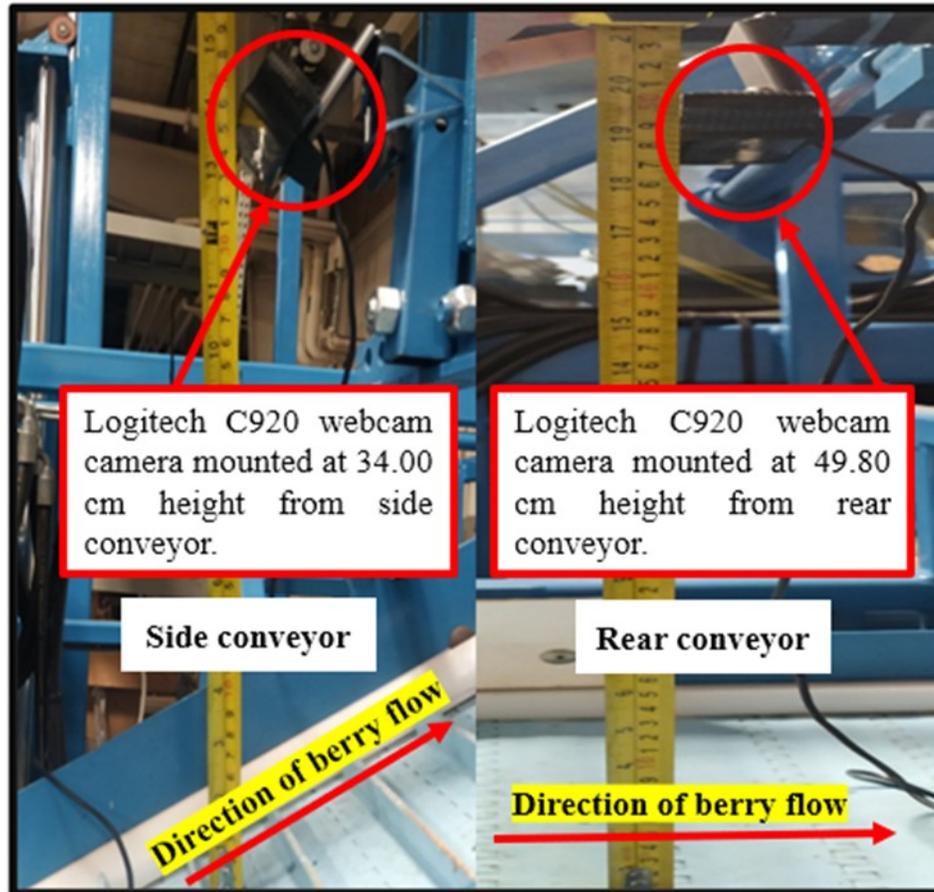


Figure 4-1. Camera setup at the side and rear conveyor

4.1.2 Camera Acquisition Software Development

A video acquisition graphical user interface (GUI) was developed for Windows 10 (Microsoft Corp. Redmond, Washington, United States) 64-bit computer system using Python 3.7.2 programming language (Python Software Foundation, Wilmington, Delaware, United States) and python version of open computer vision library (OpenCV). OpenCV-Python is a python wrapper of OpenCV C++ (Bradsk 2000) and was used because of syntax simplicity, code readability, and offering the HighGUI module for developing a simple GUI program.

At first, the size of the frame was defined inside of the functions (VideoCapture()) for two Logitech C920 webcam cameras (Logitech International S.A., Lausanne, Switzerland). The frame size was set to 1280x720 pixels. Then the index value of cameras was passed through the functions for capturing frames. A method (isOpenned()) was used to confirm camera initialization which checks whether cameras were connected properly. Then frames were captured by the two frame capturing functions where functions return values (True) confirmed frame readings. Finally, the captured frames were displayed at the 1 ms rate to make a video stream. A user-based event was added to the GUI to interrupt video streaming.

The camera acquisition program was executed on the computer system. The field of view for the conveyors was marked using a marker and measured with a measuring tape. The field of view of side conveyor was 30.00 cm x 118.00 cm (horizontal x vertical) where each pixel covered approximately 0.23 cm x 0.16 cm (horizontal x vertical) (Figure 4-2). The rear camera covered 60.70 cm x 37.50 cm (horizontal x vertical) at the rear conveyor and each pixel occupied approximately 0.04 cm x 0.05 cm (horizontal x vertical) of the field of view (Figure 4-2).

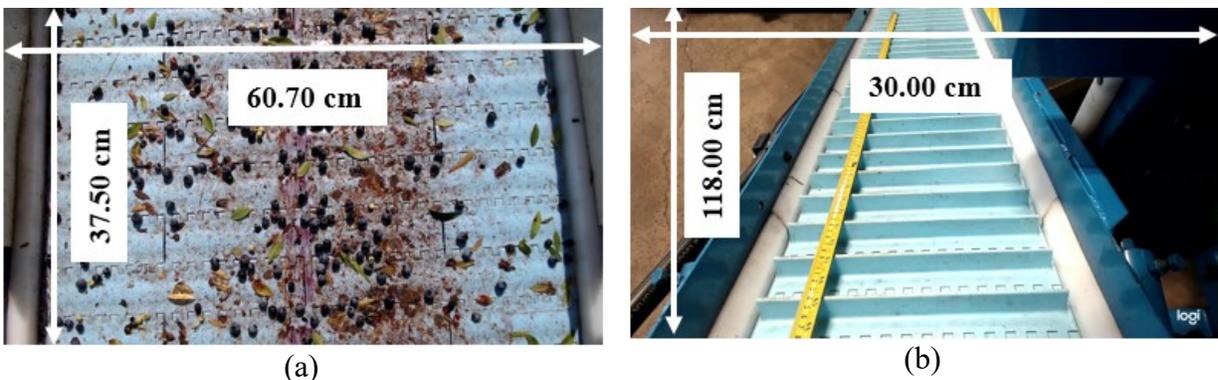


Figure 4-2. (a) Field of view to the rear conveyor at 48.5 cm camera height; (b) field of view of side conveyor at 34.0 cm camera height

4.1.3 Determine the Minimum Processing Time Required for a Debris Detection System to Perform in Real-time

Conveyors are one of the important parts of the wild blueberry harvester as they convey harvested berries from picking head to the storage bin. The conveyors are operated by a hydraulic motor which is powered by the central hydraulic system. A sprocket pulley is responsible for the rolling plastic flighted belt in the clockwise direction. The maximum shaft revolution per min (rpm) of side and rear conveyor motor was measured at 110 rpm and 72 rpm respectively using a tachometer on the reel of the conveyor belts. The pulley diameter of the side and rear conveyor belt was measured to be 9.00 cm. The width of the side and rear conveyor was measured at 27.00 and 59.60 cm respectively. The linear speed of the side and rear conveyor belt was also measured at 0.52 m s⁻¹ and 0.34 ms⁻¹ respectively using equation (4-1).

The time required for a scene to pass the cameras at the side and rear conveyors was 2276.67 ms and 1105.21 ms respectively, which calculated by the equation (4-2)

$$v \text{ (ms}^{-1}\text{)} = \frac{2\pi \times r}{60} \times N \text{ (rpm)} \quad (4-1)$$

where

v is the linear velocity expressed in ms⁻¹

N is the angular velocity expressed in rpm

r is the radius expressed in m

$$\text{Time needs to pass a scene (ms)} = \frac{\text{Vertical field of view}}{\text{conveyor belt speed}} \times 1000 \quad (4-2)$$

Where,

Vertical field of view in cm

Conveyor belt speed in cm s^{-1}

4.1.4 Determine Detection Speed of CNNs for Developing Debris Detection System

4.1.4.1 Data Collection and Preparation

Two commercial fields in central Nova Scotia termed as Debert site (45.4418°N, 63.4496°W) and the East Mines site (45.42713°N, -63.48186°W) were selected for capturing experimental videos using two Logitech C920 webcam cameras (Logitech International S.A., Lausanne, Switzerland) mounted on the side and rear conveyors of a mechanical wild blueberry harvester during harvesting (Figure 4-3). The videos were captured from twelve different plots in the two fields during harvesting and extracted using a video to image converter software (Free Video to JPG Converter, Ver.: 5.0.101.201).



Figure 4-3. Location of fields in google map

A dataset containing a total of 125 images was prepared and sorted based on the total number of objects. Images were categorized as simple images, simple-moderate images, moderate, moderate-complex, and complex. Five different subsets were prepared from the dataset with each subset containing 25 images, having 5 images from each category of image (Table 4-1). The dataset was labelled using custom software developed with the Lazarus compiler (<https://www.lazarus-ide.org/>) and labelled ripe berries, green berries, leaves, stems, and dirt based

on classifiable appearance in the images (Figure 4-4). The number of objects was labelled corresponding to the classes shown in (Table 4-2).

Table 4-1. Number of objects per image in different categories of images

| Image categories | Number of images | Number of objects range |
|------------------------|------------------|-------------------------|
| simple images | 25 | 0-49 |
| simple-moderate images | 25 | 50-99 |
| moderate | 25 | 100-199 |
| moderate-complex | 25 | 200-299 |
| complex | 25 | 300-399 |

Table 4-2. Dataset Description

| Classes | Number of objects |
|---------------|-------------------|
| Leaves | 3290 |
| Stems | 169 |
| Ripe berries | 2494 |
| Green berries | 9418 |
| Dirt | 5840 |



Figure 4-4. Image labeling using custom software developed with the Lazarus compiler

4.1.4.2 Hardware Setup

A total of four different configured hardware were used in this experiment as shown in Table 4-1. Hardware Jetson TX2 was flashed using Jetpack version 3.0 to enable compute unified device architecture (CUDA version 8.0) and loading GPU related files and dependencies. A Linux based operating system, Ubuntu 14.04 64 bit (Canonical Ltd., London, UK) was installed on the host computer. Jetson TX2 was flashed from the host computer using 2.0 m long standard USB type A cable by holding the recovery button and frequently pressing the reset button of Jetson TX2. Finally, the additional driver and packages were installed on Jetson TX2 via a secure shell using an ethernet connection from the host computer. Similarly, CUDA version 10.20 was installed on the Hardware-4 to enable network inference and recognition of images. On the other hand, only C++ API of OpenCV was adopted on Hardware-1 and Hardware-2 to use CPU power for network inference and recognition on images.

Table 4-3. Specification of four different hardware

| Specifications | Hewlett-Packard (Hardware-1) | Shuttle XPC (Hardware-2) | Jetson TX2 (Hardware-3) | Desktop (Hardware-4) |
|----------------|--------------------------------------|--------------------------------------|--|--|
| CPU | Intel® Core™ i5-4300U CPU @ 1.90 GHz | Intel® Core™ i7-6700K CPU @ 4.00 GHz | ARM Cortex-A57 (quadcore) @ 2GHz + NVIDIA Denver2 (dualcore) @ 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 |

4.1.4.3 Darknet Installation

Darknet-53 works as a feature extractor in YOLOv3 networks which performs 1.5 times faster than ResNet-101 and 2 times faster ResNet-152 frameworks (Redmon & Farhadi 2018). The CPUs of Hardware-1 and Hardware-2 were accessed by enabling OPENCV, Advanced Vector Extensions (AVX), and Open Multi-Processing (OPENMP) and disabling GPU, CUDA Deep Neural Network (CUDNN) library, CUDNN_HALF in the make file of Darknet. 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 make the file of Darknet. The configuration file of YOLOv3 was kept as same as used in Chapter-3 except batch size was changed to 1 for testing because batch = 1 helps DL models to use less memory and achieve real-time performance (Bianco et al., 2018). YOLOv3 evaluates the whole image at once and generates bounding boxes and class probabilities (Figure 4-5).

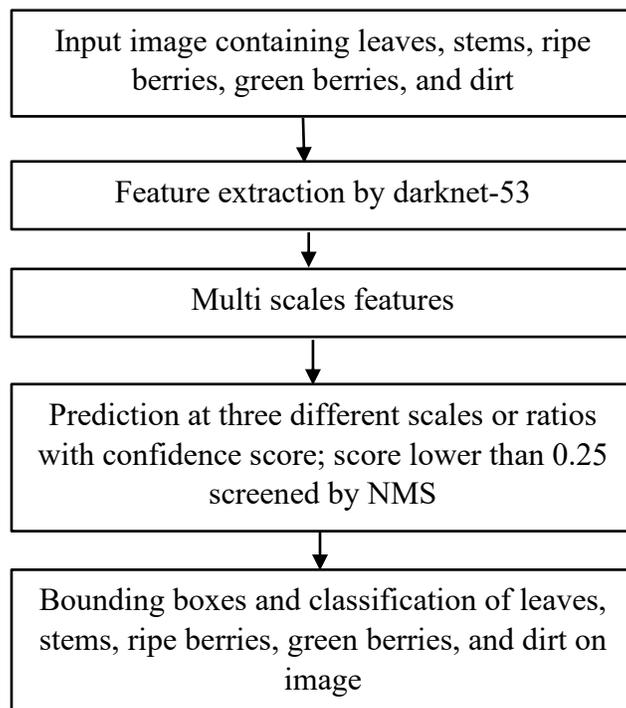


Figure 4- 5. Flow chart of YOLOv3 network architecture

4.1.4.4 Measurement of Prediction Time

The best two models were selected from chapter-3 and used for network inference. The models were denoted as Model-1 and Model-2. Model-1 (YOLOv3-SPP, mAP: 74.38%) was trained on 2700 images and tested on 300 images where Model-2 (YOLOv3-SPP, mAP: 73.03%) was trained and tested on 1800 and 200 images. These two models were used to measure the detection time of images on four different computer architectures. Each subset of the dataset was randomly inputted to the YOLOv3-SPP networks installed on four different hardware and the average prediction time per image was calculated (Figure 4-6).

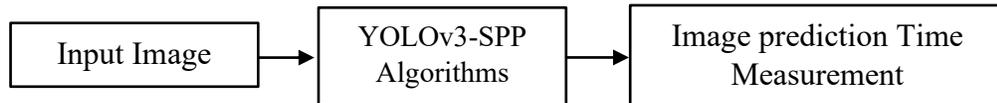


Figure 4-6. Flowchart of measuring prediction time of an image.

4.1.4.5 Statistical Analysis for Determining the Effect of Models and Hardware on Prediction Time

A 2x4 factorial design was performed using Minitab 17 (Minitab Inc. NY, USA) on four different hardware for determining the effect of hardware and models on the prediction time of objects in images. An analysis of variance (ANOVA) was performed with ($\alpha = 0.05$) level of significance to check the effect of hardware and models on prediction time. The data was collected for eight different treatments (Hardware-1 x Model-1, Hardware-1 x Model-2, Hardware-2 x Model-2, Hardware-2 x Model-1, Hardware-3 x Model-2, Hardware-3 x Model-1, Hardware-4 x Model-1, Hardware-4 x Model-2) where each treatment was replicated five times. Tukey's multiple mean comparisons (MMC) were performed to find out means difference.

4.1.5 Validation of Models

Models were validated using popular evaluation metrics. Model-1 and Model-2 were validated using a total of 125 images and results were recorded. Precision is the ratio of true detections of leaves, stems, ripe berries, green berries, and dirt to the identified detections was calculated by (Eq. 4-3)

$$\text{Precision} = \frac{TP}{TP+FP} \quad (4-3)$$

Recall is the ratio of true detection of leaves, stems, ripe berries, green berries, and dirt among all the ground truth detections was calculated by (Eq. 4-4)

$$\text{Recall} = \frac{TP}{TP+FN} \quad (4-4)$$

True positive (TP) is the number of correct detection 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 of not detection of leaves, stems, ripe berries, green berries, and dirt where ground truths are present.

F1-score is the harmonic mean of precision and recall was calculated by (Eq. 4-5)

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (4-5)$$

Mean average precision (mAP) is integral to all the precisions values of all target classes was calculated by (Eq. 4-6)

$$\text{mAP} = \int_0^1 p(o) do \quad (4-6)$$

The IoU value confirms whether the detection of leaves, stems, ripe berries, green berries, and dirt were true positive or false positive. When predicted bounding boxes have an IoU greater than given IoU threshold, prediction is considered as a true positive otherwise, false positive. Selection

of IoU value is important before detection thus, nine different IoU threshold (0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, and 0.90) values were experimented for determining the effect of threshold values on detection. IoU can be calculated by (Eq. 4-7)

$$\text{IoU} = \frac{B_1 \cap B_2}{B_1 \cup B_2} \quad 4-7$$

Where

B_1 = Ground truth bounding box

B_2 = Predicted bounding box

4.1.6 Development of a Debris Detection System

The video acquisition system was upgraded for developing the debris detection system. The best model was selected from the validation of models and deployed in Intel® Core™ i9-7900X CPU @ 3.30 GHz, GeForce RTX™ 2080 Ti @ 1665 MHz desktop computer. C++ API of OpenCV was compiled with CUDA to enable Nvidia GPUs and CUDNN since the deep neural network library of OpenCV- Python supports CPU computation by default. The code development was divided into three sections such as network definition, video processing, and display output.

The network was defined by the loading configuration file and binary trained weight of YOLOv3-SPP using `cv2.dnn.readNetFromDarknet()` method. The GPU computation was confirmed by calling `setPreferableBackend (cv2.dnn.DNN_BACKEND_CUDA)`, `net.setPreferableTarget (cv2.dnn.DNN_TARGET_CUDA)` methods after loading YOLO network. Images were preprocessed using `cv2.dnn.blobFromImages()` method which created a four-dimensional binary object for use in YOLOv3-SPP. The method took six arguments such as image, scale factor, output size, mean, swapRB, and crop. The first argument image was used for taking image input; the

second argument scaled pixel of unsigned 8-bit images from 0 to 1 dividing by 255, the third argument set up output images to 1280 x 736 pixels. The fifth argument swapped image channel order from BGR to RGB by setting True value and the fourth argument was set to (0,0,0) since image normalization was not considered during YOLO training. The last argument crop was set to false to preserve the original aspect ratio of the image. Then the blobs of images passed through the network to obtain a prediction with a confidence score. An NMS technique was applied to filter out better and refined bounding boxes. Finally, the detection result was displayed using imshow() function. The debris detection GUI was developed for displaying information of leaves, stems, ripe berries, green berries, and dirt on the computer screen during harvesting to monitor fruit quality on the side and rear conveyor at the same time.

4.2 Result and Discussion

The result of the ANOVA indicated that the Hardware ($P < 0.001$) and Hardware*Models (0.015) had a significant effect on processing time however only Models had no contribution on processing time. Tukey MMC was performed on the interaction factor (Hardware*Models) only because the interaction effect was significant (Table 4-4).

Table 4-4. Results of ANOVA on the detection time on four different hardware at 5% ($\alpha = 0.05$) level of significance

| Source | P-Value |
|-----------------|---------|
| Hardware | < 0.001 |
| Models | 0.724 |
| Hardware*Models | 0.015 |

Interaction of Hardware-1 and Model-1 resulted in the highest mean (12126.90 ms) and the interaction of Hardware-4 Model-2 yielded the lowest mean (33.30 ms) as shown in Table 4-5.

Table 4-5 also showed that the processing speed of a single image on the CPU was higher than the GPU. DL models performed better in GPUs (Hardware -3, Hardware-4) than CPUs (Hardware-1, Hardware-2) because GPUs have high bandwidth and parallel computing capabilities (Kim et al., 2018). Kim et al. (2018) 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. 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 models (Table 4-5). GPU based hardware resulted in a higher frame rate because the GPU can process an image faster than CPU (Kim et al., 2018).

Table 4-5. Result of MMC using Tukey method to identify significant difference on models and hardware at 5% ($\alpha = 0.05$) level of significance

| Hardware*Model | Prediction time (Mean) (unit in ms) | Frame rate (FPS) |
|-----------------------|--|-----------------------------|
| Hardware-1 Model-1 | 12126.90 A | 0.08 |
| Hardware-1 Model-2 | 12056.40A | 0.08 |
| Hardware-2 Model-2 | 3625.60B | 0.28 |
| Hardware-2 Model-1 | 3608.60B | 0.28 |
| Hardware-3 Model-2 | 868.70C | 1.15 |
| Hardware-3 Model-1 | 831.20C | 1.20 |
| Hardware-4 Model-1 | 34.10D | 29.33 |
| Hardware-4 Model-2 | 33.30D | 30.03 |

Figure 4-7 showed a GUI of debris detection system developed in OpenCV-Python. GUI shows detection of targets at side and rear conveyor. GUI showing detection of leaves, stems, green berries, ripe berries, and dirt on videos was represented as 1, 2, 3, 4, and 5, respectively.

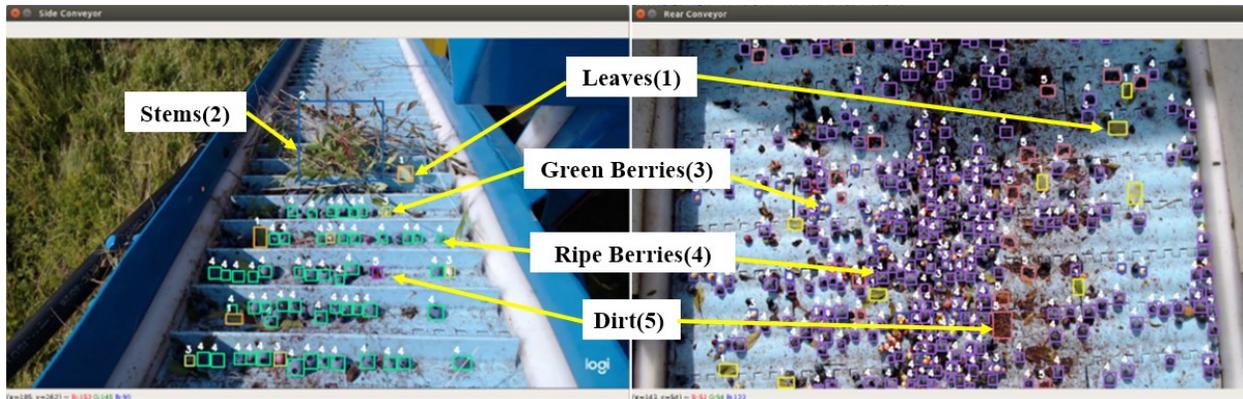


Figure 4-7. GUI of a debris detection system.

4.2.1 Validation of Models

Model-1 and Model-2 were validated using the dataset containing 125 images. The validation results showed that both models achieved 83% precision (Table 4-6). All the target classes except the dirt class achieved comparatively lower average precision (59.89%) during Model-2 validation than Model-1 validation. Model-1 achieved comparatively higher recall (65%) and F1-score (0.65) values than Model-2 however, they yielded comparatively lower mAP (71.10%) values than the mAP of Model-2 (72.36%) (Table 4-6).

Table 4-6. Evaluations of Model-1 under IoU (0.5) and threshold (0.25)

| Models | Leaves | Stems | Green Berries | Ripe Berries | Dirt | Precision | Recall | F1-score | mAP(%) |
|---------|--------|-------|---------------|--------------|-------|-----------|--------|----------|--------|
| Model-1 | 78.49 | 70.07 | 65.42 | 78.64 | 62.86 | 0.83 | 0.53 | 0.65 | 71.10 |
| Model-2 | 79.48 | 81.33 | 65.20 | 75.91 | 59.89 | 0.83 | 0.48 | 0.61 | 72.36 |

4.2.1.1 Evaluation of Model-1 Under Different IoU and Confidence Thresholds

The performance of Model-1 was recorded with different IoU and confidence threshold values. Table 4-7 showed that with the increase of IoU and confidence thresholds from 0.10 to

0.90, average precision values increased for all target classes. The threshold value 0.10 provided comparatively higher average precision of target classes and the lower average precision was achieved at 0.90 IoU and confidence threshold. The target classes leaves, stems, green berries, ripe berries, and dirt achieved 78.49%, 70.07%, 65.42%, 78.64%, and 62.86% of average precision respectively at the threshold of 0.50 (Table 4-7). Model-1 yielded the highest 85.90% of mAP at threshold 0.50 and 0.13% of mAP at a threshold of 0.90 (Table 4-7). Model-1 reached 71.10% of mAP at the threshold of 0.50 (Table 4-7).

Table 4-7. The average precision (%) and mAP (%) of the Model-1 under different IoU and confidence threshold

| IoU& Confidence | 0.10 | 0.20 | 0.30 | 0.40 | 0.50 | 0.60 | 0.70 | 0.80 | 0.90 |
|--------------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Leaves | 86.02 | 85.82 | 85.01 | 83.17 | 78.49 | 63.75 | 34.56 | 6.64 | 0.09 |
| Stems | 85.20 | 84.57 | 84.14 | 78.21 | 70.07 | 45.81 | 20.29 | 4.44 | 0.38 |
| Green Berries | 88.89 | 88.44 | 86.19 | 80.22 | 65.42 | 35.72 | 13.83 | 2.02 | 0.04 |
| Ripe Berries | 90.79 | 90.63 | 90.19 | 88.34 | 78.64 | 55.50 | 29.50 | 6.83 | 0.11 |
| Dirt | 78.60 | 78.22 | 77.24 | 72.79 | 62.86 | 42.82 | 18.35 | 2.37 | 0.01 |
| mAP | 85.90 | 85.54 | 84.55 | 80.55 | 71.10 | 48.72 | 23.31 | 4.46 | 0.13 |

Table 4-8 showed that Model-1 reached 83%, 53%, and 65% of precision, recall, and F1-score respectively at 0.50 IoU and confidence threshold. When the threshold was increased from 0.50 to 0.90, precision, recall, and F1-score were decreased. However, the performance of model-1 increased when the threshold decreased from 0.50 to 0.10 (Table 4-8). Maximum precision, recall, and F1-score were achieved at 0.10 IoU threshold and 0.10 confidence score (Table 4-8). The value of true positive decreased from 12284 to 311 when the threshold changed from 0.10 to 0.90 (Table 4-8). Conversely false positive and false negative detections significantly increased

with the increase of threshold from 0.10 to 0.90 which lead to decreased performance of the model as well as detection of leaves, stems, green berries, ripe berries, and dirt in an image (Table 4-8).

Table 4-8. True positive, False positive, False negative, Precision, Recall and F1-score of Model-1 under different IoU and confidence threshold

| IoU& Confidence | 0.10 | 0.20 | 0.30 | 0.40 | 0.50 | 0.60 | 0.70 | 0.80 | 0.90 |
|--------------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| True positive | 12284 | 12271 | 12216 | 12019 | 11246 | 9413 | 6623 | 2872 | 311 |
| False positive | 1392 | 1405 | 1460 | 1657 | 2377 | 4263 | 7053 | 10804 | 13365 |
| False negative | 8927 | 8940 | 8995 | 9192 | 9965 | 11798 | 14588 | 18339 | 20900 |
| Precision | 0.90 | 0.90 | 0.89 | 0.88 | 0.83 | 0.69 | 0.48 | 0.21 | 0.02 |
| Recall | 0.58 | 0.58 | 0.58 | 0.57 | 0.53 | 0.44 | 0.31 | 0.14 | 0.01 |
| F1-score | 0.70 | 0.70 | 0.70 | 0.69 | 0.65 | 0.54 | 0.38 | 0.16 | 0.02 |

4.2.1.2 Evaluation of the Model-2 Under Different IoU and Confidence Thresholds

Table 4 -9 showed that 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 reached 79.48%, 81.33%, 65.20%, 75.91%, and 59.89% of average precision at standard IoU (0.50) (Table 4-9). At the lowest threshold (0.10) leaves, stems, green berries, ripe berries, and dirt classes achieved best average precision 87.57%, 89.98%, 87.70%, 90.38%, and 75.85% respectively and decreased with the increase of the threshold value (Table 4-9).

Table 4-9. The average precision (%) and mAP (%) of the Model-2 under different IoU and confidence threshold

| IoU& Confidence | 0.10 | 0.20 | 0.30 | 0.40 | 0.50 | 0.60 | 0.70 | 0.80 | 0.90 |
|--------------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Leaves | 86.57 | 86.26 | 85.58 | 84.05 | 79.48 | 65.38 | 36.16 | 5.99 | 0.12 |
| Stems | 89.98 | 89.53 | 88.30 | 85.87 | 81.33 | 53.78 | 16.61 | 1.56 | 0.02 |
| Green Berries | 87.70 | 87.22 | 84.93 | 79.89 | 65.20 | 37.25 | 13.62 | 1.93 | 0.04 |

| IoU& Confidence | 0.10 | 0.20 | 0.30 | 0.40 | 0.50 | 0.60 | 0.70 | 0.80 | 0.90 |
|--------------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Dirt | 75.85 | 75.51 | 74.25 | 69.54 | 59.89 | 41.27 | 16.86 | 2.11 | 0.02 |
| mAP | 86.10 | 85.74 | 84.54 | 81.32 | 72.36 | 50.07 | 22.44 | 3.73 | 0.07 |

Model-2 behaved similar to Model-1 at 0.10 IoU threshold and 0.10 confidence score, except that it detected comparatively lower true positive and higher false positive than Model-1. Model-1 detected a total of 10181, 2110, and 11030 true positive, false positive, and false negatives respectively at the threshold of 0.50 and the number of false positives increased with the increasing of the threshold (Table 4-10). The precision (0.91), recall (0.53), and F1-score (0.67) were achieved at the threshold of 0.10 (Table 4-10). When the threshold value increased from the standard value (0.50) false negative detection increased and the overall performance of models also decreased. The models yielded comparatively higher F1-score at 0.10 threshold than the threshold of 0.50.

Table 4-10. True positive, False positive, False negative, Precision, Recall and F1-score of Model-2 under different IoU and confidence threshold

| IoU& Confidence | 0.10 | 0.20 | 0.30 | 0.40 | 0.50 | 0.60 | 0.70 | 0.80 | 0.90 |
|--------------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| True positive | 11136 | 11129 | 11072 | 10880 | 10181 | 8474 | 6040 | 2645 | 281 |
| False positive | 1124 | 1131 | 1188 | 1380 | 2110 | 3786 | 6220 | 9615 | 11979 |
| False negative | 10075 | 10082 | 10139 | 10331 | 11030 | 12737 | 15171 | 18566 | 20930 |
| Precision | 0.91 | 91 | 0.90 | 0.89 | 0.83 | 0.69 | 0.49 | 0.22 | 0.02 |
| Recall | 0.53 | 0.52 | 0.52 | 0.51 | 0.48 | 0.40 | 0.28 | 0.12 | 0.01 |
| F1-score | 0.67 | 0.66 | 0.66 | 0.65 | 0.61 | 0.51 | 0.36 | 0.16 | 0.02 |

The results showed that with the increases in the threshold value from 0.50, the overall performance of both models decreased (Figure 4-8). The probable reason was small anchors of

objects such as leaves stems, dirt, green berries, and ripe berries which had a lower average IoU than larger object IoUs (Yan et al., 2019). Because of this, the models might struggle to consider objects as true positive, and overall performance was decreased at a higher threshold. The true positives started to increase and false positives and false negatives started to decrease with a decrease of the threshold from 0.50. The maximum true positive achieved for both models using the same dataset was at 0.10 threshold value. False positives and false negatives reached a peak at the threshold of 0.90.

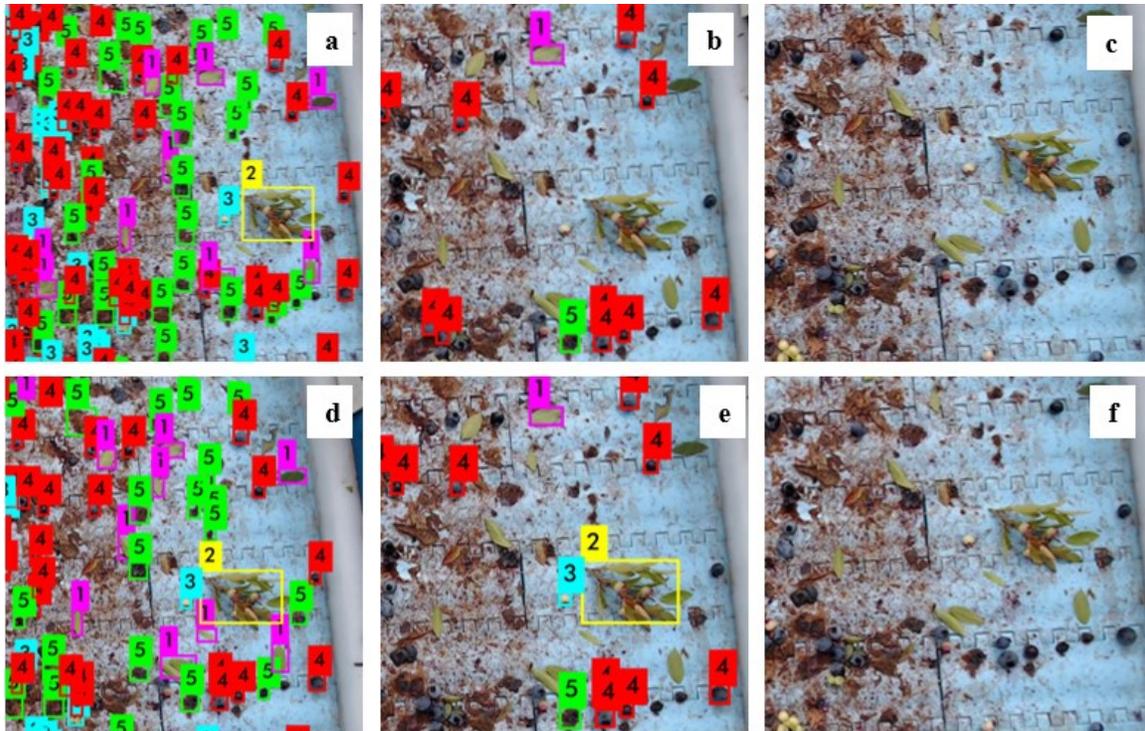


Figure 4-8. Validation of models under different IoU and confidence thresholds.

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 is shown in the first row and second row respectively. Figure (a) and (d) showed detection at 0.10 IoU and 0.10 confidence threshold. Figure (b) and (e) showed detection at 0.50 IoU and 0.50 confidence threshold. Figure (c) and (f) showed detection at 0.90 IoU and 0.90 confidence threshold.

4.3 Conclusion

Automatic separation of leaves, stems, and dirt from the fruits using a CNN can be an essential step for improving wild blueberry fruit quality and enhancing processing facilities. A DL based real-time automatic separating system requires not only a great detection accuracy but also needs high-speed image processing hardware. Two CNN models and four different hardware were investigated to select an appropriate combination of model and hardware which achieve the lowest average image prediction time and highest frame rate. A desktop configured with Intel® Core™ i9-7900X CPU @ 3.30 GHz and GeForce RTX™ 2080 Ti @ 1665 MHz GPU combined with a CNN model YOLOv3-SPP(mAP: 73.03%) achieved the lowest prediction time (33.30 ms) with the highest frame rate (30.03 FPS) when tested on 125 different images where image complexity was considered by varying the number of objects from 0 to 399 images. The processing time of a single image must be lower than the camera passing time of a scene. Hardware-4 processed images faster than other hardware in our experiment which achieved lower processing time than the camera passing time of a scene. The overall performance of both models reached, increased when threshold values were near to 0.50. The threshold values such as 0.10, 0.20, 0.30, and 0.40 enhanced the overall performance of models which, can be used in future experiments for real-time detection. Model-1 detected a greater number of true positive (11246) than Model-2 (10181) and a smaller number of false positive (1392) and false negative (8927) compared to false positive (2110) and false negative (11030) of Model-2 at 0.50 IoU and confidence threshold. In terms of true positive detection, Model-1 outperformed Model-2. However, validation mAP of Model-2 (72.36%) was slightly better than Model-1 (71.10%). The probable reason is the models were trained on two different datasets (Model-1: 2700 images; Model-2: 1800 images) however validated on a common dataset will need future investigation. A real-time automatic debris

separation system can be developed using these two models and Hardware-4 which would be a valuable addition to the wild blueberry industry for improving fruit quality by separating debris during harvesting.

CHAPTER 5: EVALUATION OF OPTIMIZED CNN MODEL FOR DEBRIS DETECTION ON THE IMAGES CAPTURED DURING HARVESTING

Abstract

A total of 300 images of which, 150 were of the side conveyor and 150 were of the rear conveyor were selected from a dataset. The images were selected randomly from the 1000 images which were collected from two different fields in Central Nova Scotia. Thirty different sample datasets each contained ten images; five images from side conveyor and five images from the rear conveyor. The images were labelled in darknet format to ensure the compatibility of the CNN (YOLOv3-SPP) for evaluation. The system was developed with the YOLOv3-SPP and incorporated with GeForce RTX™ 2080 Ti @ 1665 MHz graphics processing unit (GPU) card (NVIDIA, Santa Clara, CA) hardware. The system achieved $R^2 = 0.86$, $R^2 = 0.81$, $R^2 = 0.77$, $R^2 = 0.34$, and $R^2 = 0.25\%$ for green berries, ripe berries, leaves, stem and dirt respectively. There was no significant difference observed between system detection of green berries, ripe berries and leaves and manual detection of green berries ripe berries and leaves and yielded $P = 0.593$, $P = 0.061$ and $P = 0.641$, respectively. However, there was a significant difference between system detection of stems, and dirt with the manual detection of stems, and dirt and achieved $P < 0.001$ and $P < 0.001$ respectively. These results can help to deploy the system for real-time detection of leaves, stems, green berries, ripe berries, and dirt to monitor wild blueberry fruit quality.

5.0 Introduction

Wild blueberry (*Vaccinium angustifolium* Ait.) is a commercially managed horticultural crop native to northeastern North America. It is a biennial crop thus, vegetation is grown in the first year, and pollination and pruning occur in the following year. Wild blueberry fields were developed from deforested farmland, abandoned cropland, and fields. However, the improved

management practices (application of fungicides, herbicides, fertilizers, pollination, and pruning, etc.) have changed the wild blueberry vegetation resulting in significant increases in plant densities, plant height, and fruit yield (Esau et al., 2018). This increased plant foliage reduces the fruit quality by entering into the storage bin during mechanical wild blueberry harvesting.

Application of computer vision-based technologies are growing fast in different field of agriculture including yield estimation (Zaman et al., 2008), weed detection (Esau et al., 2018, Rehman et al., 2018), leaf disease detection (Mahmud et al., 2019), etc. because of its robust classification and automatic decision-making ability. Zaman et al. (2008) developed an automated yield monitoring system using computer vision techniques and achieved a high correlation between actual and predicted fruit yield with an R^2 value of 0.99. Chang et al. (2012) developed an automated yield monitoring system for the double head mechanical wild blueberry harvester and observed high correlations ($R^2 = 0.94$; $P < 0.001$; $R^2 = 0.95$; $P < 0.001$) between the percentage of blue pixels of wild blueberry and actual fruit yield. The system achieved a higher R^2 value of 0.95 between actual and predicted fruit yield for two different fields. Mahmud (2019) also developed an artificial neural network (ANN) based on computer vision technology to detect powdery mildew disease in strawberry fields. The system yielded R^2 of 0.93, 0.88, and 0.92 between manual and automatic powdery mildew disease detection for three different fields.

Computer vision-based technologies incorporated with deep learning (DL) can extract features automatically (LeCun et al., 2015). Manual feature extraction requires feature engineering skill and selection of redundant features may lead to decreased classification accuracy and increased computation time (Chang et al., 2012; Mahmud et al., 2019).

Convolutional Neural Networks (CNNs) are a multilayer DL network that has been implemented by several researchers in precision agriculture (Granitto et al., 2002; Piedad et al.,

2018; Tu et al., 2018). Schumann et al. (2019) also developed a wild blueberry maturity (unripe green, unripe red, and ripe blue) recognition system using YOLO (you only look once) CNNs and achieved 85.3% of mAP. Wang et al. (2018) classified sound and damaged highbush blueberries using two varieties of CNNs (ResNet and ResNeXt) and achieved 88.44% and 87.84% accuracy, respectively. Sun et al. (2019) developed soybean yield prediction using Long Short-Term Memory network and achieved R^2 of 0.74 between predicted and actual yield in season and R^2 of 0.78 between predicted and actual yield in the end of the season. Gutiérrez et al. (2019) also developed a mango yield prediction system using a CNN and achieved R^2 of 0.75 for field-based mango counting and R^2 of 0.83 for image-based counting. Convolutional Neural Networks can also be used for monitoring wild blueberry fruit quality by separating debris from harvested berries. Automatic separation of debris during harvesting can increase cleaning line speeds at processing facilities and reduce fruit shrinkage. Thus, the automatic separation of debris in situ can be a valuable addition for the wild blueberry industry as a whole.

5.1 Material and Methodology

5.1.1 Data Collection

A dataset containing 1000 images was collected from two different commercial fields in Central Nova Scotia termed as Debert site (45.4418°N, 63.4496°W) and the East Mines site (45.42713°N, -63.48186°W). Images were collected from 15th August 2019 to 31st August 2019 from morning to evening. The images were captured using two Logitech C920 webcam cameras (Logitech International S.A., Lausanne, Switzerland) mounted before and after the blower fan on a Doug Bragg Enterprises Ltd. (DBE) commercial mechanical wild blueberry harvester. A total of 300 images, of them 150 images of side conveyor, and 150 images of the rear conveyor, were selected

randomly from the dataset. Thirty different subsets of the dataset were produced from 300 images and each set contained 5 images of the side conveyor and 5 images of the rear conveyor. The subset images were labeled in Darknet format to make it compatible with YOLOv3-SPP testing. The objects of images such as ripe berries, green berries, leaves, stems, and dirt were labelled using custom software developed with the Lazarus compiler (<https://www.lazarus-ide.org/>) based on the classifiable appearance. The number of ground truths of ripe berries, green berries, leaves, stems, and dirt in each subset was counted and recorded using a bash script.

5.1.2 YOLOv3-SPP Evaluation on the Images

The model (YOLOv3-SPP; mAP-74.38%) was selected from the previous chapter-4. The YOLOv3-SPP was implemented for developing a debris detection system and evaluated using the subsets of the dataset. The fastest hardware was selected from chapter-4 and was used for evaluating the subset images. The hardware consisted of GeForce RTX™ 2080 Ti @ 1665 MHz graphics processing unit card (NVIDIA, Santa Clara, CA) installed with 64-bit Ubuntu 16.04 (Canonical Group Ltd, London, UK). The batch size in the configuration file of YOLOv3-SPP was set to 1 because batch = 1 helps DL models to use less memory and achieve real-time performance (Bianco et al., 2018). The detection results of ripe berries, green berries, leaves, stems, and dirt were saved as a text (.txt) file for each subset of the dataset and then counted and recorded using a bash script.

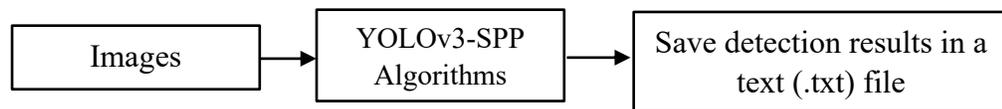


Figure 5-1. Flowchart of saving detection results of YOLOv3-SPP.

Figure 5-2 showed the Graphical User Interface (GUI) of the debris detection system. The two windows of GUI represented side and rear conveyor view respectively. The system was detecting

the target classes from the video feeds. GUI showing detection of leaves, stems, green berries, ripe berries, and dirt on videos was represented as 1, 2, 3, 4, and 5, respectively.

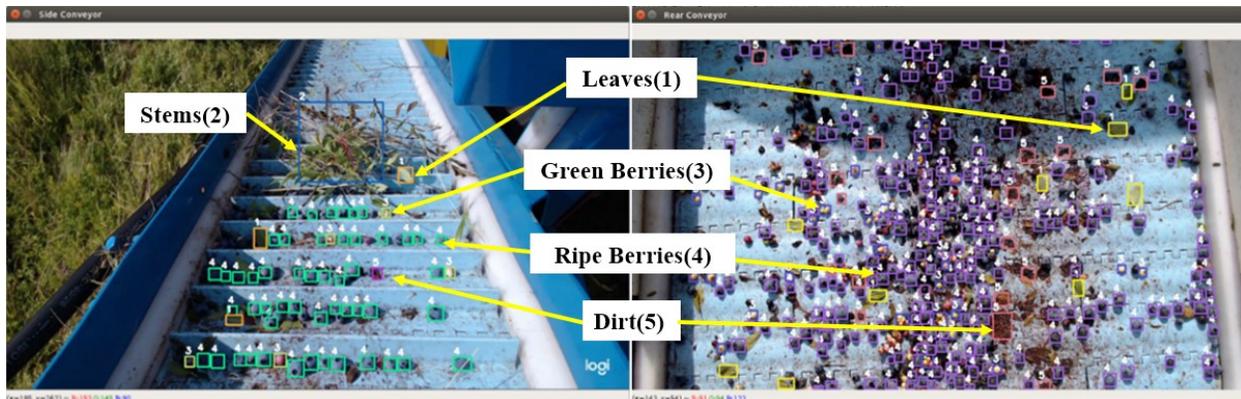


Figure 5-2. Automatic detection of leaves, stems, green berries, ripe berries, and dirt using GUI

5.1.3 Statistical Analysis

A linear regression analysis was used to compare the manual detections (ground truths) with automatic system detection for 30 different subsets of the dataset using Minitab 19 (Minitab Inc. NY, USA) statistical software. The coefficient of determination (R^2) was calculated to find out the correlation of automatic detection with ground truths. A paired t-test was also used to compare the mean of two measurements in Minitab version 19.

5.2 Results and Discussion

The linear regression analysis results showed correlations of automatic detection of leaves stems green berries ripe berries and dirt with ground truths (Figure 5-3). The system showed strong correlation ($R^2 = 0.86$, $P < 0.001$, $N = 30$) between automatic detection and ground truths of green berries (Figure 5-3, (a)). The automatic detection of ripe berries was also strongly correlated ($R^2 = 0.81$, $P < 0.001$, $N = 30$) with ground truths of ripe berries (Figure 5-3, (b)). The automatic detection of leaves was strongly correlated ($R^2 = 0.77$, $P < 0.001$, $N = 30$) with ground truths of

leaves (Figure 5-3, (c)). Lower correlation ($R^2 = 0.34$, $P = 0.001$, $N = 30$) was achieved for stems detection by the system (Figure 5-3, (d)). Similarly, lower correlation ($R^2 = 0.25$, $P = 0.005$, $N = 30$) was achieved for dirt by the system (Figure 5-3, (e)).

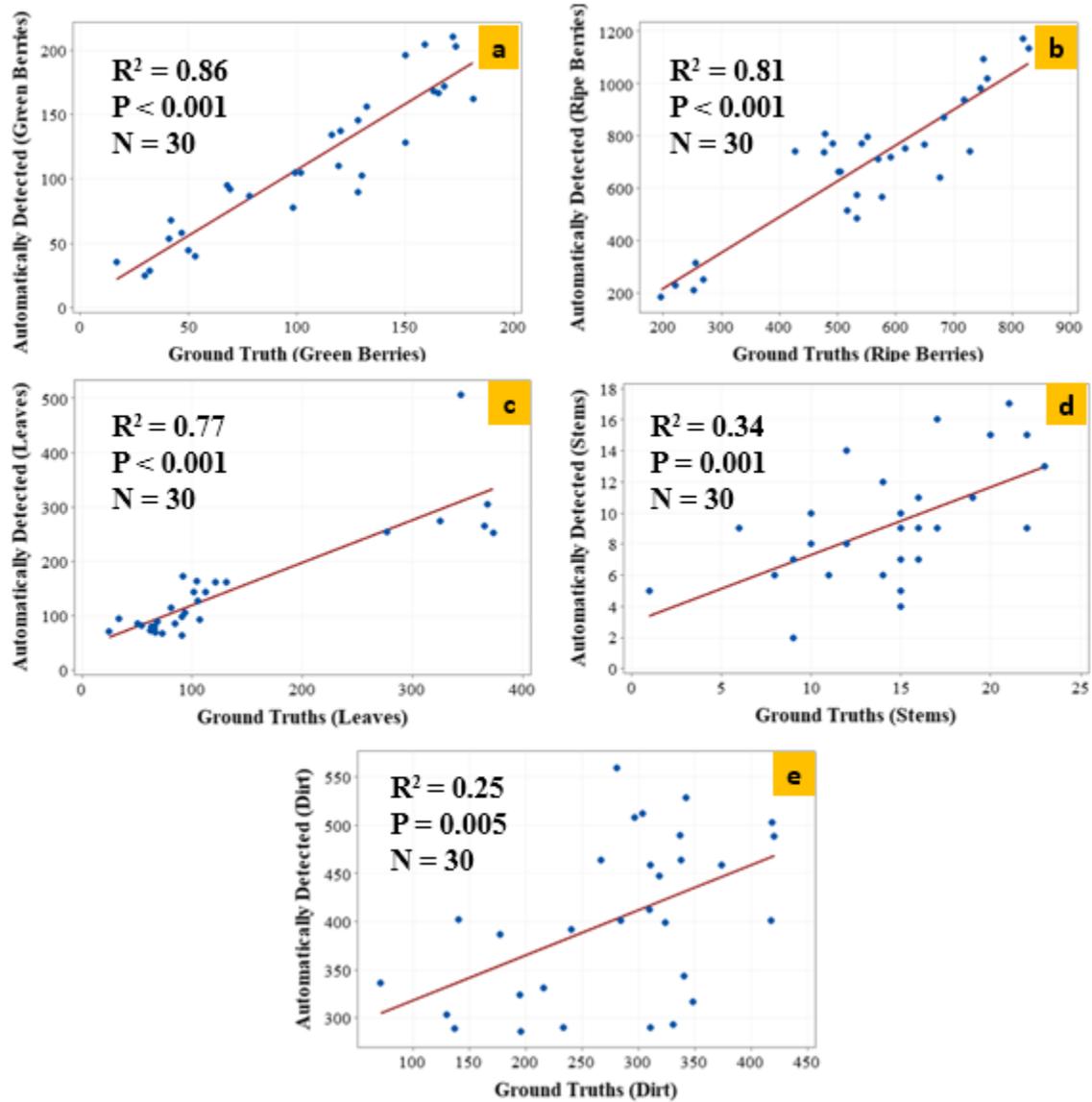


Figure 5-3. Automatic detection vs manual detection (ground truth). Automatic detection vs manual detection of green berries (a), ripe berries (b), leaves (c), stems (d) and, dirt (e).

Table 5-1 showed manual and automatic detection of classes. The automatic detection of green berries was not significantly different than manual detection ($P = 0.593$) (Table 5-1). Results showed there was no significant difference automatic detection of ripe berries ($P = 0.061$) with the manual (ground truth) detections of ripe berries (Table 5-1). The P-value (0.641) of leaves indicated that there was no significant difference between automatic and manual detection (ground truth) of leaves. However, the P-values < 0.001 , and < 0.001 indicated that there was a significant detection difference between ground truth and automatic detection of stems and dirt detection respectively (Table 5-1).

Table 5-1. Pair-wise t-test for manually and automatically leaves, green berries, ripe berries, stems, and dirt detection

| Classes | Detection | N | Mean | StDev | 95% CI | P-value |
|---------------|-----------|----|--------|--------|------------------|-----------|
| Green-Berries | Automatic | 30 | 113.40 | 56.10 | (93.90, 133.00) | 0.593 |
| | Manual | 30 | 106.00 | 50.87 | (86.43, 125.57) | |
| Ripe-Berries | Automatic | 30 | 695.90 | 266.80 | (606.20, 785.60) | 0.061 |
| | Manual | 30 | 574.90 | 221.90 | (485.20, 664.60) | |
| Leaves | Automatic | 30 | 145.90 | 97.90 | (107.80, 183.90) | 0.641 |
| | Manual | 30 | 133.30 | 109.80 | (95.20, 171.30) | |
| Stems | Automatic | 30 | 9.16 | 3.71 | (7.54, 10.78) | < 0.001 |
| | Manual | 30 | 14.26 | 14.26 | (12.68, 15.88) | |
| Dirt | Automatic | 30 | 403.00 | 83.90 | (371.20, 434.80) | < 0.001 |
| | Manual | 30 | 280.70 | 89.80 | (248.90, 312.40) | |

Figure 5-3 and Figure 5-4 showed automatic detection and manual detection (ground truth) of leaves, stems, green berries, ripe berries, and dirt respectively. In Figure 5-4 the system counted a total of 188 objects while a total of 263 objects were labelled (Figure 5-4).

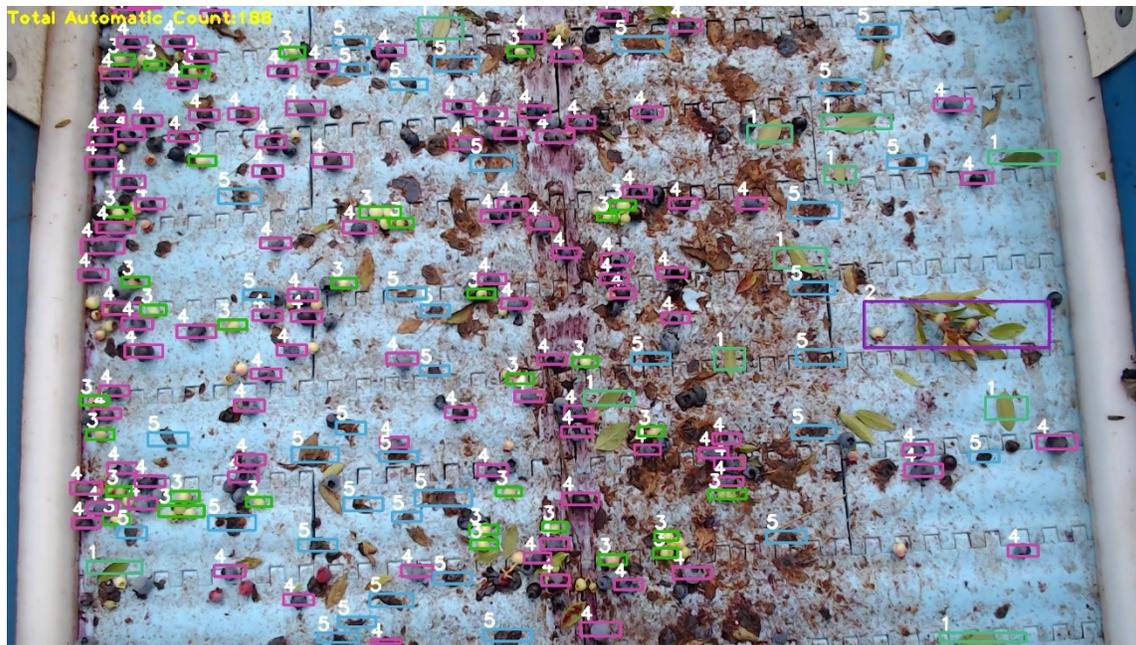


Figure 5-4. Automatic detection of leaves, stems, green berries, ripe berries, and dirt on the image was represented as 1, 2, 3, 4, and 5, respectively.

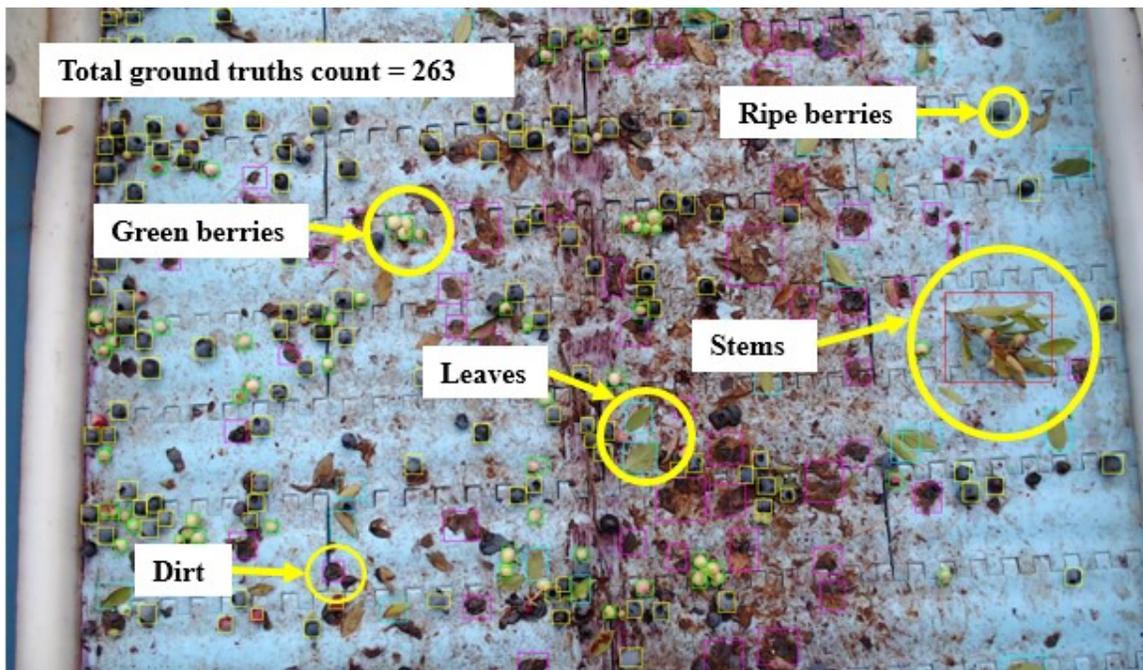


Figure 5-5. Image labeling using custom software developed with the Lazarus compiler

5.3 Conclusion

Results concluded that automatic and manual detections of green berries, ripe berries and leaves were not significantly different. However, a significant difference was observed between automatic and manual detection of stems, and automatic and manual detection of dirt. The correlations $R^2 = 0.86$, $R^2 = 0.81$, $R^2 = 0.77$, $R^2 = 0.34$ and $R^2 = 0.25$ were achieved for green berries, ripe berries, leaves, stems and dirt respectively. This developed system can be implemented and tested in realtime during wild blueberry harvesting which could be an important and valuable addition for the future development of an automatic debris detection system.

CHAPTER 6: GENERAL CONCLUSION AND FUTURE RECOMMENDATION

Wild blueberries are grown on over 55,000 hectares in Canada and more than 80% of that area is harvested mechanically (PMRA, 2005). Due to the continuous development of field management practices (i.e., application of fungicides, herbicides, fertilizers, pollination, and pruning, etc.) augmented plant characteristics have resulted in increased debris in the harvester's handling systems. This increases the quantity of plant debris and reduces berry quality during harvesting. A CNN based automatic debris separation technology can be a viable solution to monitor the quality of berries during harvesting.

6.1 General Conclusion

In chapter-3, three CNNs (YOLOv3, YOLOv3-SPP, and YOLOv3-Tiny) were implemented and compared for detecting leaves, stems, green berries, ripe berries, and dirt. The CNNs were trained and tested using an experimental dataset comprised of 1,000 images. The dataset was collected from 12 different plots in two commercially managed fields, the Debert site (45.4418°N, 63.4496°W) and the East Mines site (45.4412°N, -63.48186°W) in central Nova Scotia. The performance of CNN models was evaluated using precision, recall, F1-score, and mAP (%) values. The YOLOv3-SPP, YOLOv3, and YOLOv3-Tiny achieved 68.49%, 67.75%, and 60.36% of mAP when trained and tested on the non-augmentation datasets. A total of five different color augmentation techniques (sharpening, brightness, contrast, gamma correction, and saturation) were used for determining the effects of data augmentation on accuracy. The combination of these techniques improved CNN accuracy (mAP) by 5.89%. YOLOv3-SPP achieved slightly higher mAP (74.38%) when trained and tested on the augmented dataset (T2).

In chapter-4, The two models (mAP-74.38%; mAP-73.03%) were selected from chapter-3 and implemented on four different hardware to detect leaves, stems, ripe berries, green berries, and dirt within images. The processing time of each hardware and model combination was compared for investigating real-time performance. The combination of Hardware-4 and Model-2 yielded fastest detection (33.30 ms) and achieved highest average frames per second (30.03 FPS). The slowest detection was achieved from Model-1 and Hardware-1 (12126.90 ms) which yielded the lowest average frame rate (0.80 FPS). The Model-2 achieved comparatively better mAP (72.36%) than Model-1 (71.10%) under 0.5 IoU and 0.25 confidence threshold. The effect of IoU and confidence threshold values on the overall detection of target classes were also examined by changing nine different IoU and confidence thresholds (0.10, 0.20, 0.30, 0.40, 0.50, 0.60, 0.70, 0.80, and 0.90). The maximum mAP was achieved from Model-1 (85.90%) and Model-2 (86.10%) under 0.10 IoU and 0.10 confidence threshold.

In chapter-5, the automatic detection was compared to the ground truth data. The system was developed using a YOLOv3-SPP (mAP-74.38%) model incorporating with GeForce RTX™ 2080 Ti @ 1665 MHz graphics processing unit card (NVIDIA, Santa Clara, CA) hardware. The system yielded 0.86, 0.81, 0.77, 0.34, and 0.25 of R^2 for green berries, ripe berries, leaves, stems, and dirt respectively. Among of the five target classes, the system detected green berries, ripe berries and leaves successfully at 5% level of significance i.e., significant difference was not observed between system and ground truth detection for green berries, ripe berries and leaves.

6.2 Future Recommendation

The result of this study suggests the ability to install the developed machine vision system on a mechanical wild blueberry harvester to monitor wild blueberry fruit quality. The developed

machine vision system has the potential to automate the speed and position of the debris cleaning brush to optimize berry quality. A feedback loop using the developed machine vision system could relay information from the hydraulic control system to precisely control the debris cleaning brush. The developed system could also be incorporated onto commercial berry processing units for automatic separation of leaves, stems, dirt, and green berries from ripe harvested berries and used for saving manual labour and costs required for manual separation. The system could be implemented for developing a multi air channel-based blower fan system for optimizing debris cleaning performance and enhancing harvester efficiency. The result of this study recommends evaluating the system in the wild blueberry fields during harvesting. The CNN based detection technology was introduced in this system but, the incorporation of object tracking technology in the system could be valuable for monitoring debris as well as tracking during harvesting. It could likewise be used for fruit and debris yield prediction. The fruit and debris yield can be calculated by multiplying average weight of fruit and debris with the total count of harvested berry and debris by the system, respectively. The existing GUI could be improved by adding buttons for opening and closing cameras, checkboxes for opening a specific camera, and a display section to show output information. The GUI could be updated by using an advanced python module including PyQt, Tkinter, and wxPython (Summerfield, 2017).

Several factors were not considered in this study which might affect detection accuracy of the developed system. The labelling of target objects was done by visual appearance, where labelling of dirt and stems class could be improved. Dirt, including sand, silt and clay on conveyors were labelled based on appearance. The variable size of dirt particles might have caused lower detection accuracy (59.89%) for the dirt class shown in chapter-4. High resolution cameras with zoom lenses could be used for better detection of small dirt particles during harvesting. Stems

attached with leaves which were labelled as stems rather than their own class, Labelling leaves within stems could provide enough information to discriminate stems from leaves class. Those factors could be considered in future system development in order to improve debris detection.

REFERENCES

- Al-Hiary, H., Bani-Ahmad, S., Reyalat, M., Braik, M., & ALRahamneh, Z. (2011). Fast and accurate detection and classification of plant diseases. *International Journal of Computer Applications*, 17(1), 31-38.
- Ali, S. (2016). *Effect of harvesting time on berry losses during mechanical harvesting of wild blueberries*. Masters Dissertation, Dalhousie University, Canada.
- Amara, J., Bouaziz, B., & Algergawy, A. (2017). A deep learning-based approach for banana leaf diseases classification. In *BTW (Workshops)*, 79-88.
- Andrea, C. C., Daniel, B. B. M., & Misael, J. B. J. (2017). Precise weed and maize classification through convolutional neuronal networks. In *IEEE Second Ecuador Technical Chapters Meeting (ETCM)*, 1-6
- Bargoti, S., & Underwood, J. (2016). Image classification with orchard metadata. In *2016 IEEE International Conference Robotics and Automation (ICRA)*, 5164-5170
- Benjdira, B., Khursheed, T., Koubaa, A., Ammar, A., & Ouni, K. (2019). Car Detection using Unmanned Aerial Vehicles: Comparison between Faster R-CNN and YOLOv3. In *Unmanned Vehicle Systems-Oman (UVS), 2019 1st International Conference on IEEE*, 1-6.
- Bhandary, N., MacKay, C., Richards, A., Tong, J., & Anastasiu, D. C. (2017). Robust classification of city roadway objects for traffic related applications. In *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCOM/IOP/SCI)*, 1-6.

- Bharati, A., Singh, R., Vatsa, M., & Bowyer, K. W. (2016). Detecting facial retouching using supervised deep learning. *IEEE Transactions on Information Forensics and Security*, 11(9), 1903-1913.
- Bianco, S., Cadene, R., Celona, L., & Napoletano, P. (2018). Benchmark analysis of representative deep neural network architectures. *IEEE Access*, 6, 64270-64277.
- Bradski, G. (2000). The opencv library. *Dr Dobb's J. Software Tools*, 25, 120-125.
- Cavallo, D. P., Cefola, M., Pace, B., Logrieco, A. F., & Attolico, G. (2018). Non-destructive automatic quality evaluation of fresh-cut iceberg lettuce through packaging material. *Journal of Food Engineering*, 223, 46-52.
- Chang, Y. K., Zaman, Q. U., Schumann, A. W., Percival, D. C., Esau, T. J., & Ayalew, G. (2012). Development of color co-occurrence matrix based machine vision algorithms for wild blueberry fields. *Applied Engineering in Agriculture*, 28(3), 315-323.
- Ćorović, A., Ilić, V., Durić, S., Marijan, M., & Pavković, B. (2018). The real-time detection of traffic participants using yolo algorithm. In *2018 26th Telecommunications Forum (TELFOR)*, 1-4
- Dale, A. E., Hanson, J., Yarborough, D. E., McNicol, R. J., Stang, E. J., Brennan, R., Hergert, G. B. (1994). Mechanical harvesting of berry crops. *Horticultural Reviews*, 16, 255-382.
- Do, T. T., Nguyen, A., & Reid, I. (2018). Affordancenet: An end-to-end deep learning approach for object affordance detection. In *Robotics and Automation, 2018 IEEE International Conference on IEEE*, 1-5
- Donahue, D.W., A.A. Bushway, K.E. Moore, & B.J. Lagasse. (1999). Maine wild blueberries field winnowing systems. Maine Agricultural and Forest Experiment Station Technical Bulletin 174.

- Dubrovina, A., Kisilev, P., Ginsburg, B., Hashoul, S., & Kimmel, R. (2018). Computational mammography using deep neural networks. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization*, 6(3), 243-247.
- Dyrmann, M., Jørgensen, R. N., & Midtiby, H. S. (2017). Robo weed support-detection of weed locations in leaf occluded cereal crops using a fully convolutional neural network. *Advances in Animal Biosciences*, 8(2), 842-847.
- Dyrmann, M., Mortensen, A. K., Midtiby, H. S., & Jørgensen, R. N. (2016). Pixel-wise classification of weeds and crops in images by using a fully convolutional neural network. In *Proceedings of the International Conference on Agricultural Engineering*, 26-29.
- Eaton, L. J. (1988). *Nitrogen cycling in lowbush blueberry stands*. Doctoral Dissertation, Dalhousie University, Canada.
- Enciso-Aragón, C. J., Pachón-Suescún, C. G., & Jimenez-Moreno, R. (2018). Quality control system by means of CNN and fuzzy systems. *International Journal of Applied Engineering Research*, 13(16), 12846-12853.
- Esau, K. (2019). *Effective debris removal methods for mechanical wild blueberry harvester*. Masters Dissertation, Dalhousie University, Canada.
- Esau, K., Esau, T., Zaman, Q., Farooque, A., & Schumann, A. (2018). Effective use of a variable speed blower fan on a mechanical wild blueberry harvester. *Applied Engineering in Agriculture*, 34(5), 831-840.
- Fan, H., Cao, Z., Jiang, Y., Yin, Q., & Doudou, C. (2014). Learning deep face representation. *arXiv preprint arXiv:1403.2802*.
- Farooque, A. A., Chang, Y. K., Zaman, Q. U., Groulx, D., Schumann, A. W., & Esau, T. J. (2013). Performance evaluation of multiple ground based sensors mounted on a commercial wild

- blueberry harvester to sense plant height, fruit yield and topographic features in real-time. *Computers and Electronics in Agriculture*, *91*, 135-144.
- Farooque, A. A., Zaman, Q. U., Groulx, D., Schumann, A. W., Yarborough, D. E., & Nguyen-Quang, T. (2014). Effect of ground speed and header revolutions on the picking efficiency of a commercial wild blueberry harvester. *Applied Engineering in Agriculture*, *30*(4), 535-546.
- Fu, L., Feng, Y., Majeed, Y., Zhang, X., Zhang, J., Karkee, M., & Zhang, Q. (2018). Kiwifruit detection in field images using Faster R-CNN with ZFNet. *IFAC-PapersOnLine*, *51*(17), 45-50.
- Fuentes, A., Yoon, S., Kim, S. C., & Park, D. S. (2017). A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors*, *17*(9), 2022.
- Gidge, L. (1995). Blueberry harvesting machine and method of harvesting. U.S. Patent No. 5450716.
- Granitto, P. M., Navone, H. D., Verdes, P. F., & Ceccatto, H. A. (2002). Weed seeds identification by machine vision. *Computers and Electronics in Agriculture*, *33*(2), 91-103.
- Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., & Chen, T. (2018). Recent advances in convolutional neural networks. *Pattern Recognition*, *77*, 354-377.
- Hall, I. V., Aalders, L. E., Nickerson, N. L., & Vander Kloet, S. P. (1979). Biological flora of Canada. 1. *Vaccinium angustifolium* Ait., sweet lowbush blueberry. *Canadian Field-Naturalist*, *93*, 415-430.
- Hall, I. V., Craig, D. L., & Lawrence, R. A. (1983). A comparison of hand raking and mechanical harvesting of lowbush blueberries. *Canadian Journal of Plant Science*, *63*(4), 951-954.

- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770-778
- Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700-4708
- Huang, R., Gu, J., Sun, X., Hou, Y., & Uddin, S. (2019). A Rapid recognition method for electronic components based on the improved YOLO-V3 network. *Electronics*, 8(8), 825.
- Jahanbakhshi, A., Momeny, M., Mahmoudi, M., & Zhang, Y. D. (2020). Classification of sour lemons based on apparent defects using stochastic pooling mechanism in deep convolutional neural networks. *Scientia Horticulturae*, 263, 109133.
- Jain, V., & Learned-Miller, E. (2010). *Fddb: A benchmark for face detection in unconstrained settings*, 2(4), 5-15.
- Jiang, G., Wong, C. Y., Lin, S. C. F., Rahman, M. A., Ren, T. R., Kwok, N., & Wu, T. (2015). Image contrast enhancement with brightness preservation using an optimal gamma correction and weighted sum approach. *Journal of Modern Optics*, 62(7), 536-547.
- Kang, H., & Chen, C. (2019). Fruit detection and segmentation for apple harvesting using visual sensor in orchards. *Sensors*, 19(20), 4599.
- Khan, J. A., Chen, Y., Rehman, Y., & Shin, H. (2020). Performance enhancement techniques for traffic sign recognition using a deep neural network. *Multimedia Tools and Applications*, 1-16.
- Khan, M. A., Akram, T., Sharif, M., Awais, M., Javed, K., Ali, H., & Saba, T. (2018). CCDF: Automatic system for segmentation and recognition of fruit crops diseases based on

- correlation coefficient and deep CNN features. *Computers and Electronics in Agriculture*, 155, 220-236.
- Kim, C. E., Oghaz, M. M. D., Fajtl, J., Argyriou, V., & Remagnino, P. (2018). A Comparison of embedded deep learning methods for person detection. *arXiv preprint arXiv:1812.03451*.
- Kim, D. G., Burks, T. F., Qin, J., & Bulanon, D. M. (2009). Classification of grapefruit peel diseases using color texture feature analysis. *International Journal of Agricultural and Biological Engineering*, 2(3), 41-50.
- Kinsman, G. (1993). *The history of the lowbush blueberry industry in Nova Scotia 1950-1990*. Nova Scotia Dept. of Agriculture & Marketing, Truro, Nova Scotia, Canada.
- 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'. *Precision Agriculture*, 20(6), 1107-1135.
- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. *In: NIPS*, 1097-1105.
- Kussul, N., Lavreniuk, M., Skakun, S., & Shelestov, A. (2017). Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geoscience and Remote Sensing Letters*, 14(5), 778-782.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521, 436-444.
- LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4), 541-551.

- Lee, S. H., Chan, C. S., Wilkin, P., & Remagnino, P. (2015). Deep-plant: Plant identification with convolutional neural networks. In *2015 IEEE international conference on image processing (ICIP)*, 452-456
- Liang, Q., Zhu, W., Long, J., Wang, Y., Sun, W., & Wu, W. (2018). A Real-Time Detection Framework for On-Tree Mango Based on SSD Network. In *International Conference on Intelligent Robotics and Applications*, 423-436
- Litten, W. J., Smagula, J., & Durham, S. (1997). Blueberry surprise from phosphorus. Report 401. *Maine Agricultural Experiment Station, Orono, Maine.*
- Liu, G., Nouaze, J. C., Touko Mbouembe, P. L., & Kim, J. H. (2020). YOLO-Tomato: a robust algorithm for tomato detection based on YOLOv3. *Sensors*, 20, 2145.
- Liu, S., Li, X., Gao, M., Cai, Y., Nian, R., Li, P., & Lendasse, A. (2018b). Embedded online fish detection and tracking system via yolov3 and parallel correlation filter. *OCEANS 2018 MTS/IEEE Charleston IEEE*. 1-6.
- Liu, W., Ma, L., Wang, J., Chen, H. (2018a) Detection of multiclass objects in optical remote sensing Images. *EEE Geoscience and Remote Sensing Letters*, 1–5.
- Liu, Z., & Wang, S. (2019). Broken corn detection based on an adjusted yolo with focal loss. *IEEE Access*, 7, 68281-68289.
- Lu, S., Wang, B., Wang, H., Chen, L., Linjian, M., & Zhang, X. (2019). A real-time object detection algorithm for video. *Computers & Electrical Engineering*, 77, 398-408.
- Lu, Z., & Li, Y. (2017). A fruit sensing and classification system by fractional fourier entropy and improved hybrid genetic algorithm. In *5th International Conference on Industrial Application Engineering (IIAE)*. Kitakyushu, Institute of Industrial Applications Engineers, Japan, 293-299

- Mahmud, M. S., Zaman, Q. U., Esau, T. J., Price, G. W., & Prithiviraj, B. (2019). Development of an artificial cloud lighting condition system using machine vision for strawberry powdery mildew disease detection. *Computers and Electronics in Agriculture*, 158, 219-225.
- Mahmud, M. S. (2019). *Development of a machine vision system for strawberry powdery mildew disease detection*. Masters Dissertation, Dalhousie University, Canada.
- Malay, W. J. (2000). *Spatial variability and yield monitor evaluation for carrots and wild blueberries*. Masters Dissertation, Dalhousie University, Canada.
- Mazzia, V., Khaliq, A., Salvetti, F., & Chiaberge, M. (2020). Real-time apple detection system using embedded systems with hardware accelerators: an edge AI application. *IEEE Access*, 8, 9102-9114.
- McIsaac, D. (1998). Wild blueberry production and marketing in nova scotia: *The Wild Blueberry Handbook*; Production Technology Branch, Nova Scotia Department of Agriculture and Marketing: Nappan, NS, 1–12.
- Milioto, A., Lottes, P., & Stachniss, C. (2017). Real-time blob-wise sugar beets vs weeds classification for monitoring fields using convolutional neural networks. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4, 41–48.
- Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, 7, 1419-1428.
- Nagaraj, S., Muthiyan, B., Ravi, S., Menezes, V., Kapoor, K., & Jeon, H. (2017). Edge-based street object detection. *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCOM/IOP/SCI)*, 1-4.

- Nagata, M., & Cao, Q. (1998). Study on grade judgment of fruit vegetables using machine vision. *Japan Agricultural Research Quarterly*, 32(4), 257–265.
- Nakano, K. (1997). Application of neural networks to the color grading of apples. *Computers and Electronics in Agriculture*, 18(2-3), 105-116.
- Namozov, A., & Im Cho, Y. (2018). An efficient deep learning algorithm for fire and smoke detection with limited data. *Advances in Electrical and Computer Engineering*, 18(4), 121-128.
- New Brunswick, Canada (1996). *Producing high quality wild blueberry fruit*. Retrieved on August 18, 2020, from https://www2.gnb.ca/https://www2.gnb.ca/content/gnb/en/departments/10/agriculture/content/crops/wild_blueberries/high_quality_fruit.html
- Partel, V., Kakarla, S. C., & Ampatzidis, Y. (2019). Development and evaluation of a low-cost and smart technology for precision weed management utilizing artificial intelligence. *Computers and Electronics in Agriculture*, 157, 339-350.
- Perez, L., & Wang, J. (2017). The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*.
- Pesticide Risk Reduction Program (PMRA) (2005). Crop profile for wild blueberry in Canada. Annual report, Agriculture and Agri-Food Canada. Available at: http://www4.agr.gc.ca/resources/prod/doc/prog/prrp/pdf/blueberry_e.pdf.
- Pham, M. T., Courtrai, L., Friguet, C., Lefèvre, S., & Baussard, A. (2020). YOLO-Fine: One-stage detector of small objects under various backgrounds in remote sensing images. *Remote Sensing*, 12(15), 2501.

- Piedad, E., Larada, J. I., Pojas, G. J., & Ferrer, L. V. V. (2018). Postharvest classification of banana (*Musa acuminata*) using tier-based machine learning. *Postharvest Biology and Technology*, *145*, 93-100.
- Quiroz, I. A., & Alférez, G. H. (2020). Image recognition of legacy blueberries in a chilean smart farm through deep learning. *Computers and Electronics in Agriculture*, *168*, 105044.
- Rahnemoonfar, M., & Sheppard, C. (2017). Deep count: fruit counting based on deep simulated learning. *Sensors*, *17*(4), 905-916.
- Rangarajan Aravind, K., & Raja, P. (2020). Automated disease classification in (selected) agricultural crops using transfer learning. *Automatika*, *61*, 260-272.
- Redmon, J., & Farhadi, A. (2018). YOLOv3: An incremental improvement. arXiv preprint arXiv:1804.02767.
- Rehman, T. U., Zaman, Q. U., Chang, Y. K., Schumann, A. W., Corscadden, K. W., & Esau, T. J. (2018). Optimising the parameters influencing performance and weed (goldenrod) identification accuracy of colour co-occurrence matrices. *Biosystems Engineering*, *170*, 85-95.
- Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: towards real-time object detection with region proposal networks. *Advances in Neural Information Processing Systems*, 91-99.
- Sa, I., Chen, Z., Popović, M., Khanna, R., Liebisch, F., Nieto, J., & Siegwart, R. (2018). weedNet: Dense semantic weed classification using multispectral images and MAV for smart farming. *IEEE Robotics and Automation Letters*, *3*(1), 588-595.

- Schumann, A. W., Mood, N. S., Mungofa, P. D., MacEachern, C., Zaman, Q., & Esau, T. (2019). Detection of Three Fruit Maturity Stages in Wild Blueberry Fields Using Deep Learning Artificial Neural Networks. In *2019 ASABE Annual International Meeting*, 1
- Shadrin, D., Menshchikov, A., Ermilov, D., & Somov, A. (2019). Designing future precision agriculture: detection of seeds germination using artificial intelligence on a low-power embedded system. *IEEE Sensors Journal*, 19(23), 11573-11582.
- Sharpe, S., Schumann, A., & Boyd, N. (2018). Detection of Carolina Geranium (*Geranium carolinianum*) Growing in Competition with Strawberry Using Convolutional Neural Networks. *Weed Science*, 1-7.
- Shinde, S., Kothari, A., & Gupta, V. (2018). YOLO based Human Action Recognition and Localization. *Procedia Computer Science*, 133, 831-838.
- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., & Stefanovic, D. (2016). Deep neural networks based recognition of plant diseases by leaf image classification. *Computational Intelligence and Neuroscience*, 2016.
- Spann, T. M., & Danyluk, M. D. (2010). Mechanical harvesting increases leaf and stem debris in loads of mechanically harvested citrus fruit. *HortScience*, 45(8), 1297-1300.
- Statistics Canada (2016). Table 32-10-0364-01. Area, production and farm gate value of marketed fruits.
- Statistics Canada (2019). Table 32-10-0364-01. Area, production and farm gate value of marketed fruits.

- Steen, K., Christiansen, P., Karstoft, H., & Jørgensen, R. (2016). Using deep learning to challenge safety standard for highly autonomous machines in agriculture. *Journal of Imaging*, 2(1), 6-14.
- Stein, M., Bargoti, S., & Underwood, J. (2016). Image based mango fruit detection, localisation and yield estimation using multiple view geometry. *Sensors*, 16(11), 1915-1939.
- Summerfield, M. (2007). *Rapid GUI Programming with Python and Qt: The Definitive Guide to PyQt Programming (paperback)*. Pearson Education.
- Sun, J., Di, L., Sun, Z., Shen, Y., & Lai, Z. (2019). County-level soybean yield prediction using deep CNN-LSTM model. *Sensors*, 19(20), 4363.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., & Rabinovich, A. (2015). Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1-9.
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2818-2826.
- Tahir, M. W., Zaidi, N. A., Rao, A. A., Blank, R., Vellekoop, M. J., & Lang, W. (2018). A Fungus Spores Dataset and a Convolutional Neural Network Based Approach for Fungus Detection. *IEEE Transactions on Nanobioscience*, 17(3), 281-290.
- Tian, D., Zhang, C., Duan, X., & Wang, X. (2019b). An automatic car accident detection method based on cooperative vehicle infrastructure systems. *IEEE Access*, 7, 127453-127463.
- Tian, Y., Yang, G., Wang, Z., Wang, H., Li, E., & Liang, Z. (2019a). Apple detection during different growth stages in orchards using the improved YOLO-V3 model. *Computers and Electronics in Agriculture*, 157, 417-426.

- Too, E. C., Yujian, L., Njuki, S., & Yingchun, L. (2018). A comparative study of fine-tuning deep learning models for plant disease identification. *Computers and Electronics in Agriculture*.
- Tu, K., Li, L., Yang, L., Wang, J., & Sun, Q. (2018). Selection for high quality pepper seeds by machine vision and classifiers. *Journal of Integrative Agriculture*, 17, 1999-2006.
- Tumas, P., & Serackis, A. (2018). Automated image annotation based on YOLOv3. In *2018 IEEE 6th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)*, 1-3.
- Wang, J., Wang, N., Li, L., & Ren, Z. (2019). Real-time behavior detection and judgment of egg breeders based on YOLO v3. *Neural Computing and Applications*, 1-11.
- Wang, S., Lu, Z., Yang, J., Zhang, Y., Liu, J., Wei, L., & Dong, Z. (2016). Fractional Fourier entropy increases the recognition rate of fruit type detection. *BMC Plant Biology*, 16.
- Wang, S., Zhang, Y., Ji, G., Yang, J., Wu, J., & Wei, L. (2015). Fruit classification by wavelet-entropy and feedforward neural network trained by fitness-scaled chaotic ABC and biogeography-based optimization. *Entropy*, 17(8), 5711-5728.
- Wang, Z., Hu, M., & Zhai, G. (2018). Application of deep learning architectures for accurate and rapid detection of internal mechanical damage of blueberry using hyperspectral transmittance data. *Sensors*, 18(4), 1126.
- Xie, J., Girshick, R., & Farhadi, A. (2016). Unsupervised deep embedding for clustering analysis. In *International Conference on Machine Learning*, 48, 478-487.
- Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1492-1500.

- Yalcin, H. (2017). Plant phenology recognition using deep learning: Deep-Pheno. In *2017 6th International Conference on Agro-Geoinformatics*, 1-5.
- Yan, J., Wang, H., Yan, M., Diao, W., Sun, X., & Li, H. (2019). IoU-adaptive deformable R-CNN: make full use of iou for multi-class object detection in remote sensing imagery. *Remote Sensing*, *11*(3), 286.
- Yang, J., Nguyen, M. N., San, P. P., Li, X. L., & Krishnaswamy, S. (2015a). Deep convolutional neural networks on multichannel time series for human activity recognition. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 3995–4001.
- Yang, S., Luo, P., Loy, C. C., & Tang, X. (2015b). From facial parts responses to face detection: A deep learning approach. In *Proceedings of the IEEE International Conference on Computer Vision*, 3676-3684.
- Yang, S., Luo, P., Loy, C. C., & Tang, X. (2016). Wider face: A face detection benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5525-5533.
- Yang, W., & Jiachun, Z. (2018). Real-time face detection based on YOLO. In *2018 1st IEEE International Conference on Knowledge Innovation and Invention (ICKII)*, 221-224.
- Yarborough, D. E. & Ismail, A. A. (1985). Hexazinone on weeds on lowbush blueberry growth and yield. *Hortscience*, *20*, 406-407.
- Yarborough, D. E. (1991). Progress towards the development of a mechanical harvester for wild Blueberries. Fact Sheet No. 226 Univ. Maine Coop. Ext. Retrieved from: <http://umaine.edu/blueberries/factsheets/production/progress-towards-the-development-of-a-mechanical-harvester-for-wild-blueberries/>. Accessed: February 02, 2019.

- Yu, F., Xian, W., Chen, Y., Liu, F., Liao, M., Madhavan, V., & Darrell, T. (2018). BDD100K: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*.
- Zaman, Q. U., Schumann, A. W., Percival, D. C., & Gordon, R. J. (2008). Estimation of wild blueberry fruit yield using digital color photography. *Transactions of the ASABE*, 51(5), 1539-1544.
- Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, 818-833.
- Zeng, G. (2017). Fruit and vegetables classification system using image saliency and convolutional neural network. In *2017 IEEE 3rd Information Technology and Mechatronics Engineering Conference (ITOEC)*, 613-617.
- Zhang, Y. D., Dong, Z., Chen, X., Jia, W., Du, S., Muhammad, K., & Wang, S. H. (2019). Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation. *Multimedia Tools and Applications*, 78(3), 3613-3632.
- Zhang, Y., & Wu, L. (2012). Classification of fruits using computer vision and a multiclass support vector machine. *Sensors*, 12(9), 12489-12505.
- Zhang, Y., Wang, S., Ji, G., & Phillips, P. (2014). Fruit classification using computer vision and feedforward neural network. *Journal of Food Engineering*, 143, 167-177.

Appendix

Appendix a: Class average precisions (%) of YOLOv3, YOLOv3-SPP, and YOLOv3-Tiny for leaves and stems class on non augmentation dataset

| Leaves | | | Stems | | | Green Berries | | |
|---------------|------------|-------------|--------------|------------|-------------|----------------------|------------|-------------|
| v3 | SPP | Tiny | v3 | SPP | Tiny | v3 | SPP | Tiny |
| 67.83 | 71.73 | 64.14 | 45.92 | 46.22 | 40.24 | 72.95 | 72.34 | 56.52 |
| 70.67 | 75.29 | 64.69 | 67.27 | 52.86 | 60.07 | 76.84 | 80.62 | 60.45 |
| 72.12 | 74.41 | 70.55 | 69.14 | 72.69 | 68.29 | 59.34 | 66.56 | 61.31 |
| 69.19 | 69.99 | 62.11 | 45.71 | 57.98 | 43.37 | 73.15 | 76.61 | 43.73 |
| 68.29 | 70.3 | 65.93 | 64.62 | 57.64 | 69.67 | 79.03 | 81.11 | 53.63 |

Appendix b: Class average precisions (%) of YOLOv3, YOLOv3-SPP, and YOLOv3-Tiny for ripe berries and dirt class on non augmentation dataset

| Ripe Berries | | | Dirt | | |
|---------------------|------------|-------------|-------------|------------|-------------|
| v3 | SPP | Tiny | v3 | SPP | Tiny |
| 79.72 | 78.00 | 75.74 | 66.42 | 64.63 | 55.42 |
| 80.35 | 78.67 | 76.00 | 60.40 | 59.17 | 55.29 |
| 77.57 | 77.08 | 79.11 | 59.58 | 57.70 | 52.99 |
| 81.37 | 79.14 | 77.52 | 60.30 | 59.52 | 53.50 |
| 84.55 | 84.97 | 79.24 | 61.81 | 62.98 | 53.39 |

Appendix c: F1-score and mAP (%) of YOLOv3, YOLOv3-SPP, and YOLOv3-Tiny on non augmentation dataset

| F1-score | | | mAP | | |
|----------|------------|-------------|--------|------------|-------------|
| YOLOv3 | YOLOv3-SPP | YOLOv3-Tiny | YOLOv3 | YOLOv3-SPP | YOLOv3-Tiny |
| 0.66 | 0.67 | 0.62 | 66.57 | 66.58 | 59.41 |
| 0.64 | 0.65 | 0.60 | 71.11 | 69.32 | 63.3 |
| 0.64 | 0.64 | 0.62 | 67.55 | 69.69 | 66.45 |
| 0.67 | 0.68 | 0.62 | 65.94 | 68.65 | 56.05 |
| 0.67 | 0.67 | 0.61 | 71.66 | 71.40 | 64.37 |

Appendix d: Class average precisions (%) of YOLOv3 and YOLOv3-SPP for leaves and stems class on T1 augmentation dataset

| Leaves | | Stems | | Green Berries | | Ripe Berries | | Dirt | |
|--------|-------|-------|-------|---------------|-------|--------------|-------|-------|-------|
| v3 | SPP | v3 | SPP | v3 | SPP | v3 | SPP | v3 | SPP |
| 71.91 | 81.21 | 52.57 | 65.42 | 71.14 | 72.23 | 81.62 | 82.07 | 61.83 | 61.96 |
| 73.65 | 82.80 | 67.70 | 76.89 | 77.57 | 79.36 | 82.54 | 84.54 | 60.01 | 57.65 |
| 74.24 | 81.95 | 43.67 | 67.27 | 71.82 | 74.68 | 82.61 | 83.14 | 61.76 | 58.72 |
| 74.89 | 82.61 | 82.73 | 89.80 | 75.31 | 76.39 | 83.21 | 81.94 | 65.42 | 65.82 |
| 76.41 | 80.65 | 81.77 | 79.49 | 68.67 | 71.54 | 85.28 | 84.93 | 67.80 | 65.98 |

Appendix e: F1-score and mAP(%) of YOLOv3, YOLOv3-SPP on T1 dataset

| F1-score | | mAP | |
|-----------------|-------------------|---------------|-------------------|
| YOLOv3 | YOLOv3-SPP | YOLOv3 | YOLOv3-SPP |
| 0.60 | 0.63 | 67.82 | 72.58 |
| 0.61 | 0.65 | 72.29 | 76.25 |
| 0.62 | 0.66 | 66.82 | 73.15 |
| 0.61 | 0.63 | 76.31 | 79.31 |
| 0.62 | 0.64 | 75.98 | 76.52 |

Appendix f: Class average precisions (%) of YOLOv3 and YOLOv3-SPP for leaves and stems class on T2 augmentation dataset

| Leaves | | Stems | | Green Berries | | Ripe Berries | | Dirt | |
|---------------|------------|--------------|------------|----------------------|------------|---------------------|------------|-------------|------------|
| v3 | SPP | v3 | SPP | v3 | SPP | v3 | SPP | v3 | SPP |
| 76.88 | 79.55 | 65.73 | 68.95 | 80.97 | 80.11 | 83.29 | 84.73 | 63.28 | 64.52 |
| 76.35 | 79.24 | 62.30 | 75.70 | 62.30 | 68.78 | 80.96 | 78.83 | 63.09 | 63.70 |
| 74.12 | 78.28 | 59.05 | 72.43 | 76.01 | 78.36 | 83.07 | 83.88 | 64.94 | 66.56 |
| 80.60 | 83.51 | 69.25 | 81.25 | 72.43 | 71.08 | 82.04 | 81.51 | 61.52 | 63.30 |
| 79.35 | 80.40 | 74.08 | 79.83 | 78.98 | 76.10 | 83.88 | 83.81 | 66.19 | 68.57 |

Appendix g: F1-score and mAP(%) of YOLOv3, YOLOv3-SPP on T2 augmentation dataset

| F1-score | | mAP | |
|-----------------|-------------------|---------------|-------------------|
| YOLOv3 | YOLOv3-SPP | YOLOv3 | YOLOv3-SPP |
| 0.61 | 0.65 | 74.03 | 75.57 |
| 0.63 | 0.66 | 70.80 | 73.25 |
| 0.61 | 0.65 | 71.44 | 75.90 |
| 0.62 | 0.66 | 73.17 | 76.13 |
| 0.63 | 0.66 | 76.49 | 77.74 |

Appendix h: Prediction time on four different hardware and model combinations

| Hardware | Models | Prediction Time (ms) |
|-----------------|---------------|-----------------------------|
| H1 | M1 | 12.1038 |
| H1 | M2 | 12.0929 |
| H1 | M1 | 12.1779 |
| H1 | M2 | 12.0843 |
| H1 | M1 | 12.1319 |
| H1 | M2 | 12.0719 |
| H1 | M1 | 12.0513 |
| H1 | M2 | 12.0203 |
| H1 | M1 | 12.1695 |
| H1 | M2 | 12.0127 |
| H2 | M1 | 3.5498 |
| H2 | M2 | 3.6579 |
| H2 | M1 | 3.6757 |
| H2 | M2 | 3.6498 |
| H2 | M1 | 3.5390 |
| H2 | M2 | 3.6101 |
| H2 | M1 | 3.6452 |
| H2 | M2 | 3.6112 |
| H2 | M1 | 3.6331 |
| H2 | M2 | 3.5992 |
| H3 | M1 | 0.8365 |
| H3 | M2 | 0.8449 |
| H3 | M1 | 0.8381 |
| H3 | M2 | 0.8371 |
| H3 | M1 | 0.8351 |
| H3 | M2 | 0.8312 |
| H3 | M1 | 0.8244 |
| H3 | M2 | 0.9498 |
| H3 | M1 | 0.8220 |
| H3 | M2 | 0.8806 |
| H4 | M1 | 0.0341 |
| H4 | M2 | 0.0323 |
| H4 | M1 | 0.0335 |
| H4 | M2 | 0.0329 |
| H4 | M1 | 0.0342 |
| H4 | M2 | 0.0327 |
| H4 | M1 | 0.0341 |
| H4 | M2 | 0.0343 |
| H4 | M1 | 0.0344 |
| H4 | M2 | 0.0344 |

Appendix i: Automatic detection of leaves, stems, green berries, ripe berries, stems, and dirt with corresponding ground truth

| Leaves | GT | Stems | GT | Green | GT | Ripe | GT | Dirt | GT |
|---------------|-----------|--------------|-----------|--------------|-----------|-------------|-----------|-------------|-----------|
| 266 | 365 | 9 | 16 | 36 | 17 | 486 | 533 | 332 | 216 |
| 253 | 373 | 6 | 14 | 54 | 41 | 519 | 517 | 291 | 234 |
| 254 | 277 | 14 | 12 | 92 | 69 | 570 | 576 | 290 | 137 |
| 306 | 368 | 16 | 17 | 90 | 128 | 642 | 676 | 286 | 196 |
| 274 | 325 | 4 | 15 | 146 | 128 | 744 | 728 | 304 | 130 |
| 73 | 62 | 15 | 20 | 58 | 47 | 319 | 257 | 324 | 195 |
| 68 | 73 | 10 | 15 | 29 | 32 | 214 | 253 | 317 | 349 |
| 70 | 67 | 10 | 10 | 25 | 30 | 186 | 197 | 294 | 331 |
| 76 | 66 | 7 | 9 | 45 | 50 | 231 | 221 | 291 | 311 |
| 64 | 91 | 9 | 6 | 40 | 53 | 253 | 270 | 344 | 341 |
| 93 | 107 | 6 | 11 | 196 | 150 | 798 | 551 | 401 | 418 |
| 99 | 91 | 6 | 11 | 204 | 159 | 810 | 478 | 448 | 319 |
| 115 | 81 | 2 | 9 | 203 | 173 | 744 | 428 | 490 | 337 |
| 128 | 105 | 5 | 1 | 210 | 172 | 771 | 492 | 560 | 281 |
| 87 | 85 | 6 | 8 | 167 | 165 | 578 | 533 | 459 | 311 |
| 95 | 34 | 15 | 22 | 172 | 168 | 664 | 506 | 529 | 343 |
| 87 | 51 | 13 | 23 | 168 | 163 | 665 | 502 | 508 | 297 |
| 72 | 25 | 11 | 19 | 156 | 132 | 739 | 477 | 464 | 267 |
| 80 | 66 | 17 | 21 | 105 | 102 | 771 | 541 | 512 | 304 |
| 83 | 54 | 9 | 22 | 95 | 68 | 753 | 616 | 503 | 419 |
| 81 | 63 | 9 | 17 | 87 | 78 | 720 | 591 | 459 | 374 |
| 89 | 69 | 9 | 17 | 68 | 42 | 714 | 570 | 399 | 324 |
| 106 | 94 | 11 | 16 | 78 | 98 | 767 | 650 | 489 | 421 |
| 145 | 102 | 7 | 16 | 105 | 99 | 873 | 682 | 464 | 338 |
| 145 | 112 | 5 | 15 | 134 | 116 | 940 | 718 | 413 | 310 |
| 162 | 121 | 8 | 10 | 110 | 119 | 982 | 745 | 401 | 285 |
| 164 | 104 | 8 | 12 | 103 | 130 | 1020 | 900 | 392 | 241 |
| 162 | 131 | 7 | 15 | 128 | 150 | 1095 | 1100 | 387 | 178 |
| 173 | 92 | 9 | 15 | 137 | 120 | 1174 | 990 | 402 | 141 |
| 506 | 344 | 12 | 14 | 162 | 181 | 1134 | 950 | 337 | 72 |