An End-to-End System Using Artificial Intelligence (AI) and Augmented Reality (AR) to Support Table Grape Cultivation

人工知能(AI)と拡張現実(AR)を用いたぶどう栽培支援

山梨大学大学院

医工農学総合教育部

博士課程学位論文

２０２２年3月

BUAYAI　PRAWIT

SUMMARY

Trimming the inflorescence and thinning the berries are two critical processes in table grape cultivation. This is because bunch compactness, bunch shape, and berry size, which are controlled mainly by these two tasks, all have a significant impact on the market value of table grape production. The inflorescence trimming and berry thinning should be carried out during appropriate period and farmers have limited time to complete the tasks. Especially in case of berry thinning, the appropriate time period is 2-3 weeks and it overlaps with raining season during which the berries grow quickly and the bunches soon become densely packed and it becomes impossible to do thinning without hurting the neighborhood berries. Since berry thinning requires high skill and is the key task to decide the final shape of the grape bunch and the size and quality of each berry, which has a dominant impact on market value, instead of training part-time farmers, the farmers have to perform berry thinning tasks by themselves. Consequently, one skilled farmer usually needs to thin berries for more than 3000 bunches each season.

This dissertation addresses the challenging issues in applying state-of-the-art computer vision (CV) and augmented reality (AR) technology to assist inflorescence trimming and berry thinning tasks in table grape cultivation. An end-to-end approach was implemented by considering the whole process from preparing data, training artificial intelligence (AI) models, deploying AI models on the server, selecting the appropriate AR device for farmers, and designing the user interface for showing the relevant information to farmers. Image augmentation technique has been designed to tackle the difficulty of collecting sufficient images for training an AI model generalized to detect grape berries with high accuracy under various unconstrained capturing environment. Experiments have been conducted to select and fine tune the state-of-the-art AI models for the particular tasks. Novel post-processing techniques have been proposed to fulfill the requirement for each task. A server-side approach is adopted to make the AI model capable of working with various devices and reducing the computation on edge devices such as mobile phones and smart glasses. Furthermore, the user experience has been taken into consideration in visualizing the predict results to farmers in adherence with the intent of empower the farmer and do not disturb the regular tasks. The proposed end-to-end supporting system consists of an AI server and an Optical See Through Head Mounted Display (OSTHMD). Images are captured with the cameras installed on OSTHMD and sent to the AI server via pocket WiFi. The AI server accounting for intensive prediction computing then sends the results back to OSTHMD, showing the results and instructions to the farmer. This dissertation presents three main technologies for constituting such grape cultivation supporting system. The first is an automatic inflorescence measurement technology for supporting the inflorescence trimming task. The second is the automatic berry counting technology for supporting the berry thinning task. The third is the technology for automatically identifying berry to be removed for berry thinning task.

In most cases, only 20%–30% of an inflorescence is required to produce a bunch of grapes, and the ideal length varying by grape variety. Trimming inflorescences efficiently requires a farmer to accurately assess their length using their eyes, which is difficult for inexperienced farmers. While one to two weeks is the optimal time for inflorescence trimming, grape growers can significantly benefit from automated inflorescence measuring operating on a wearable device. The proposed novel end-to-end inflorescence measurement technology enables farmers to accomplish table grape trimming efficiently. It uses 2D images of the trimming scene only without requiring extra calibrators or high sophisticated preprocess, such as the existing methods based on 3D model reconstruction. The experiment results demonstrate that the proposed approach could reach an accuracy of 88.02% in inflorescence measurement and the inference time are fast enough for real-world working circumstances. An OSTHMD was employed to capture images and guide farmers without obstructing their trimming tasks. An interview to the farmers who used the proposed technology indicates that they are satisfied with the visualization design and that it aided them in intuitively comprehending the current and target lengths. As the result, the proposed system could significantly improve inflorescence trimming operations, and according to several farmers, the proposed system transformed a laborious process into a delightful one.

Berry thinning is a critical step in table grape cultivation. It is a necessary procedure for eliminating undesirable berries and provide sufficient space for remaining berries to grow into ideal size and taste. Karoglan et al. discovered that combining bunch and berry thinning increased mean cluster weight, total phenols, flavan-3-ols, anthocyanins, and a variety of other phenolic chemicals. The number of berries in the bunch is the essential criterion in the berry thinning task, and the optimal berry count range vary by the table grape variety. On the other side, counting berries during berry thinning takes time for both expert and novice farmers, and it is especially difficult for novice farmers. The proposed novel end-to-end berry number prediction technology succeeded in predicting the numbers of berries in the operating bunch accurately by making use of the state-of-the-art deep learning technology. Since a deep neural network (DNN) requires massive training data, a novel data augmentation technique simulating the thinning process is proposed to create a customed grape dataset for gaining a good instance segmentation result. To focus on the working bunch only and avoid detecting surrounding bunches, the structure of the state-of-the-art instance segmentation model is modified to integrate the location feature. The proposed location-sensitive Hybrid Task Cascade (HTC) model can also be applied for other object detection problems that require detecting a particular object from an image comprised of multiple objects of similar features. A set of features, together with their extraction algorithms, is designed for predicting the number of berries in a bunch (3D counting) using the berries detected on a single 2D image. Experimental results show that the average prediction error of the proposed method is below ±2.5 berries, which is considered to be smaller enough for being used for supporting real thinning task.

Unskilled farmers may have difficulty deciding which berry should be removed. As the standard criteria, a grape bunch is divided into several layers vertically and each layer should consist of an appropriate number of berries to form a beautiful bunch at the harvest. Implementing such criteria in real task is difficult even for experienced farmers. The proposed automatic removing berry identification technology combine DNN with a novel attention forcing mechanism to learn the knowledge from the skilled farmers. The operation scenes of skilled farmers recorded before and after removing a berry are used to create attention forcing images to train the DNN. The validation experiment shows that the prediction accuracy could reach 88.02%, which is considered to be sufficient in real applications. Experiments have been carried out on the real grape field by evaluating the quality of the grape bunches thinned by skilled farmers without using the supporting technology and normal participants (office workers) using the proposed supporting technology at harvest by experts of grape cultivation. The experiment shows outstanding results that using the proposed method can achieve 8.18% higher quality e than thinning berries without the proposed method. AI model can fit the good training data while discarding the noise, thus it enables farmers to thin the berries more consistently by preventing human error. Moreover, a post-processing technique was proposed to decide the best timing to update the predicted results to the farmers. The interview to the farmers shows that it can improve the user experience by showing consistent results to farmers, and can restrain farmers from eye fatigue and improve operating performance. In addition, the proposed attention forcing mechanism is compatible with the general DNN models for image classification, succeeded in training the image classification model to predict the berry removal with high accuracy. It can also be applied to other image classification issues that require enforcing the model to consider a particular area.

CONTENTS

INTRODUCTION

1.1 Background

Agriculture now is becoming one of the most important application areas of smart technology. This dissertation aims to present smart farming technologies to support the cultivation of table grapes. Figure 1.1 shows the annual management of table grapes. The crucial process in table grape cultivation are inflorescence trimming and berry thinning, which decide those important factors affecting the market value of table grape production, such as bunch compactness, bunch form, and berry size (Buayai, Saikaew, and Mao 2021; Creasy and Creasy 2018; Ivorra et al. 2015). These two tasks need to be performed within a short period when the grape has a rapid growth rate, which overwhelm the skilled farmers by the workload. Therefore, the farm owners would appreciate any technologies enabling unskilled farmers to perform these tasks while ensure the quality of productions. This dissertation addresses a challenging issue: how to use state-of-the-art Artificial Intelligence (AI) and Augmented Reality (AR) technology to support the inflorescence trimming and berry thinning tasks in table grape cultivation.

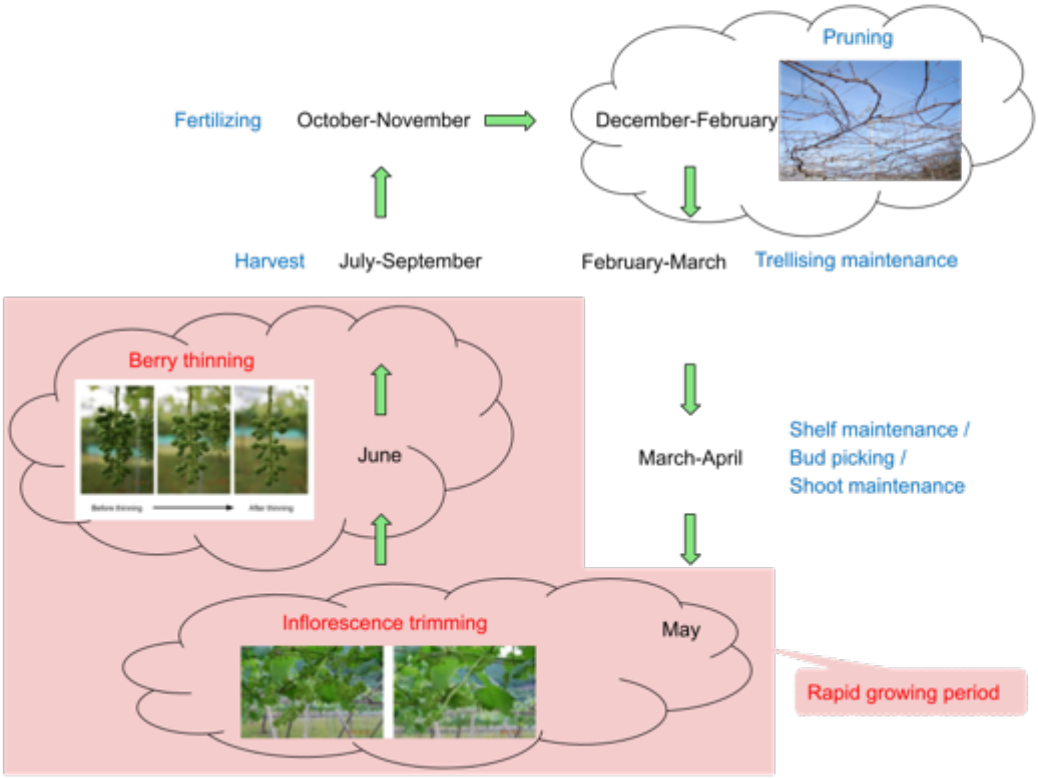


Figure 1.1 Annual management of table grapes.

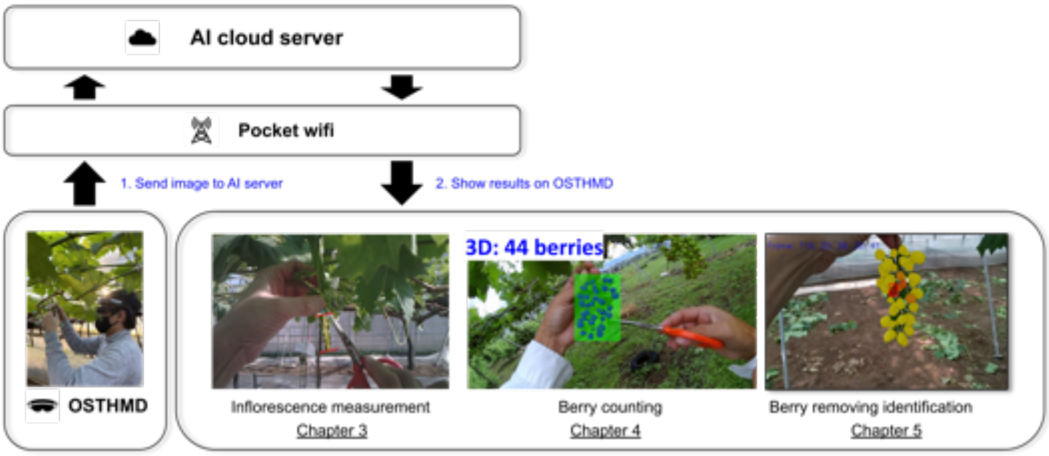
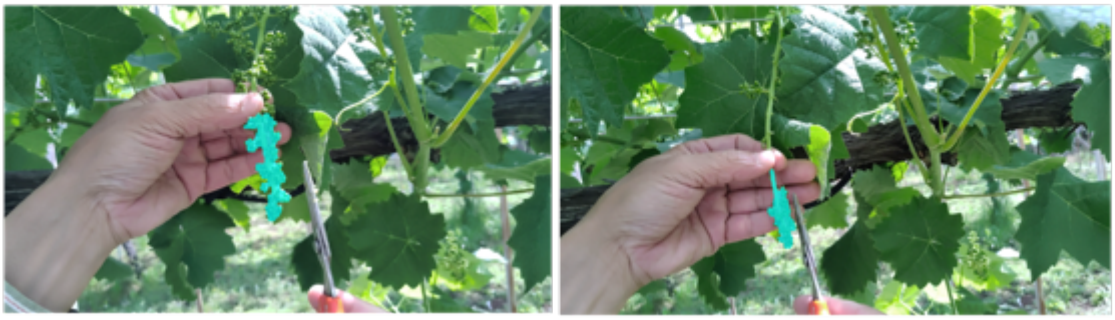

AI cloud server
Pocket wifi
1. Send image to AI server
2. Show results on OSTHMD
OSTHMD
3D: 44 berries
Inflorescence measurement
Chapter 3
Berry counting
Chapter 4
Berry removing identification
Chapter 5

Figure 1.2 The proposed end-to-end system using Artificial Intelligence (AI) and Augmented Reality (AR) to support table grape cultivation.

A novel end-to-end system using deep neural network and Optical-See-Through Head-Mounted-Display (OSTHMD) to support these two tasks. Figure 1.2 depicts the application of the proposed technology in a real table grape field. An OSTHMD is used to capture the images of the grape. Then the captured image is sent to the AI cloud server via the pocket WiFi, which uses high-speed internet such as 4G or 5G. The AI server then sends the result to OSTHMD and gives the farmer visual guidance. The OSTHMD makes it possible to avoid disrupting farmers' regular tasks. This architecture applies to both inflorescence trimming and berry thinning tasks. The inflorescence measurement system (chapter 3) was proposed for the inflorescence trimming task. The berry counting system (chapter 4) and berry removal recommendation (chapter 5) were proposed for the berry thinning task.

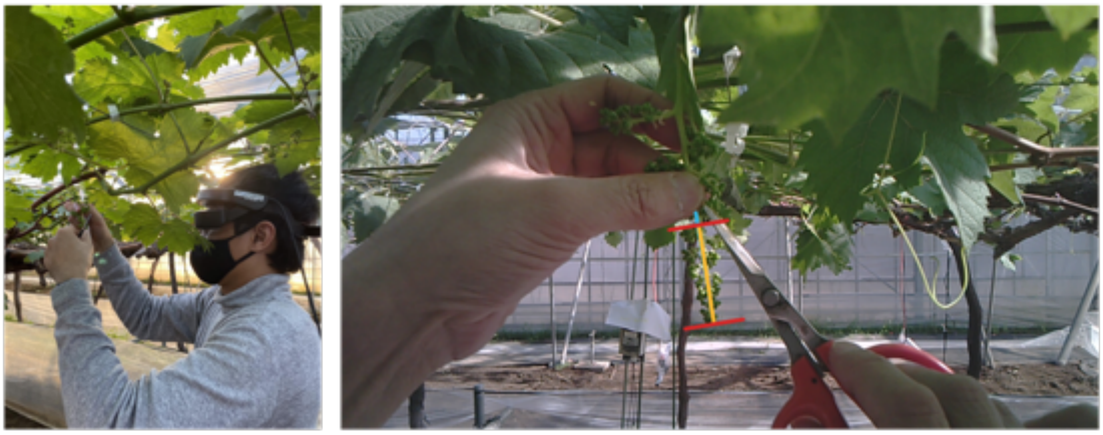
The inflorescence trimming process is required to control factors such as bunch compactness, bunch form, and berry size and produce high-quality table grapes because it can eliminate nutrient competition in a bunch and makes it less susceptible to disease growth (Barbedo 2019; Creasy and Creasy 2018; Okamoto 2007; Santos et al. 2020). Figure 1.3 shows the inflorescence during the trimming process. After trimming, the inflorescence should have an appropriate length. Usually, just 20–30% of an inflorescence is sufficient to produce a full bunch of grape (Jackson 2000), and the ideal target length is empirically decided by the grape variety (Mitsui 2019). Efficient trimming requires a farmer to measure the length of the inflorescences accurately using only the eyes, which is difficult for novice farmers. Therefore, while the appropriate period for inflorescence trimming is limited to one to two weeks (Jackson 2000), an automatic inflorescence measurement technology running on a wearable device can greatly benefit grape farmers.



a) Detected inflorescence before trimming b) Detected inflorescence after trimming

Figure 1.3 The inflorescence trimming process. After trimming, an appropriate length from the tip is kept and all other parts are removed.

A novel end-to-end inflorescence measurement technique and AR system to table grape trimming have been proposed to support the inflorescence trimming task. Figure 1.4 depicts the application of the proposed technology in real inflorescence trimming tasks. The light blue line (Figure 1.4 [b]) is the detected length, the yellow line (Figure 1.4 [b]) is the desired length, and the red line is the position above which the inflorescence should be trimmed.



a) OSTHMD b) OSTHMD's visual guadance for farmer

Figure 1.4 Applying the proposed end-to-end inflorescence measurement technique and AR system to table grape trimming. The detected length (light blue line [b]), the desired length (yellow line [b]), and the position (red line[b]) above which the inflorescence should be trimmed are visualized on OSHMD.

Besides inflorescences trimming, berry thinning is also an essential task in table grape production for producing high-quality grapes. Berry thinning is necessary to remove unnecessary berries. It benefits not only table grape but also wine grape production. Karoglan et al. (Karoglan et al. 2014) found that a combination of bunch thinning and berry thinning reduced the grape yield but increased the mean cluster weight, total phenols, flavan-3-ols, and anthocyanins, as well as many individual phenolic compounds. Likewise, the grape bunch becomes more open and less inclined to disease development (Barbedo 2019; Creasy and Creasy 2018; Santos et al. 2020).

Figure 1.5 shows a bunch during the thinning process. After thinning, the bunch should have a compact and well-balanced shape, and each berry should have sufficient space to grow to the desired size without interfering with others. A successful practice by skilled grape farmers in Japan for achieving such a requirement is using the number of berries in the working bunch to guide the thinning process. Given the desired overall shape of the bunch and the full size of grown berries, the number of berries in the working bunch is a good indicator of whether sufficient space has been created through thinning. The ideal range of berry numbers for typical table grape varieties in Japan is shown in Table 1.1. However, counting berries during berry thinning is time-consuming and is especially difficult for inexperienced farmers. Furthermore, the suitable period for berry thinning is limited to one to two weeks, when there is still enough space among berries to allow unnecessary berries to be cut without hurting those to be kept, before the grapes start to accumulate sugar (Jackson 2000). For the above reasons, an automatic berry-counting technology is desired by grape farmers.

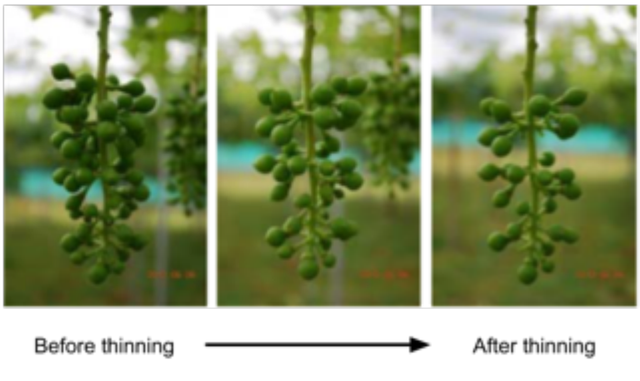
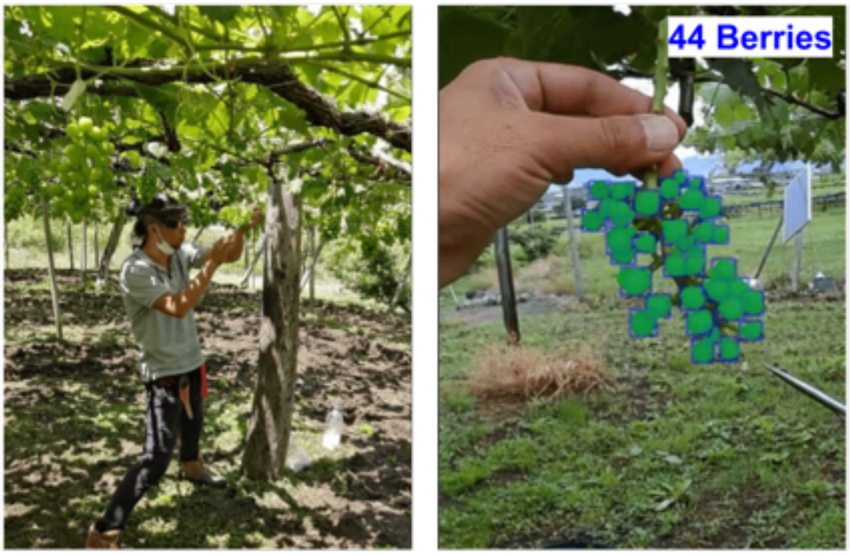


Figure 1.5 The berry-thinning process. Before thinning, the bunch was crowded with berries. After thinning, the bunch had a fine shape and a lesser likelihood of berry decay.

To tackle the time constraint problem, a novel end-to-end automatic berry-counting technology for supporting the berry-thinning process has been proposed. Figure 1.6 depicts the application of the proposed technology in real berry-thinning tasks. After AI server predicted berry numbers in a single working bunch, the detected berries and estimate the number of berries using 3D counting (including hidden berries) are shown on OSTHMD.



a) OSTHMD **b) OSTHMD's information for farmer**

Figure 1.6 Applying the proposed end-to-end automatic berry-counting technique to table grape thinning. The detected berries and estimate the number of berries using 3D counting (including hidden berries) are shown on OSTHMD.

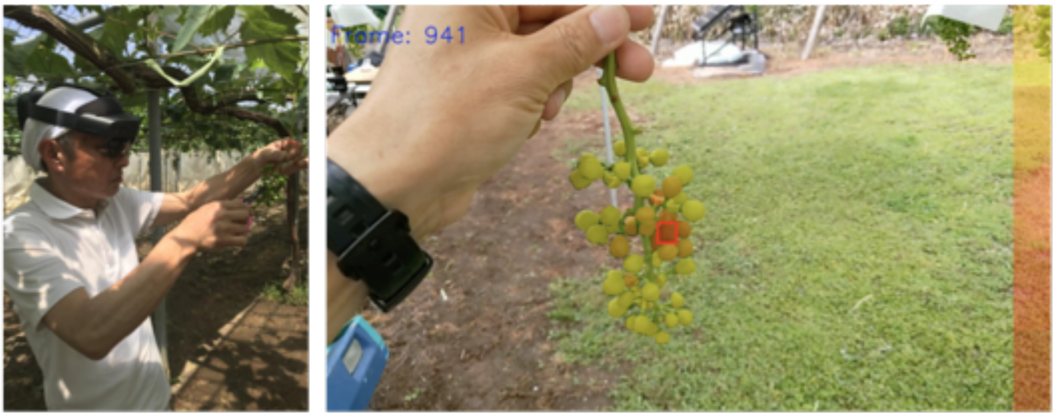
Table 1.1 The expected number of berries in the bunch according to grape variety (Mitsui 2019).

Grape variety	**Expected number of berries**
Fujiminori	28–30
Pione	32
Black beet	32
Kyoho	35–40

This dissertation not only tackles the berry counting problem in the berry thinning task by proposing automatic berry-counting technology which supports both novice and professional farmers, but also tackle decision-making problems to empower novice farmers. To determine which berry should be removed, it requires rich viticulture experience (Mitsui 2019). The standard criteria include considering the number of berries per layer, the position of berries in neighborhood, and the overall bunch shape. Skilled farmers usually making decision by imaging how the bunch looks when it fully-growth. Thus, deciding the berry to be removed is particularly difficult for novice farmers. Moreover, it's difficult and time-consuming to train novice farmers to perform berry thinning. For the above reasons, automatic identifying the berries to be removed in table grape thinning is highly desired by the table grape cultivation industry.

To tackle this issue, this dissertation introduces an image preprocessing technique called ‘attention forcing’ which turns the removing berry identification into an image classification problem to which the state-of-the-art deep neural network model can be applied. In object detection and classification using deep learning, the features of the target object are learned from the training data, and then detection and classification are performed based on these features. However, to determine whether a particular berry should be removed or not, it is necessary to consider not only the features, such as size and shape of the berry itself, but also its positional relationship with neighboring berries, the density of neighboring berries, the distribution of berries in the entire bunch, and the shape of the bunch. Therefore, conventional object detect models cannot provide the expected estimation results for identifying the berry to be removed. With the proposed attention forcing method, the probability that a berry is the target to be removed is represented as an Attention Forcing (AF) image in which only that berry is changed to a different color (white) from the other berries, and the estimation of the probability that the berry is the target to be removed is replaced by the probability estimation of whether the corresponding AF image is the correct image or not. In addition, considering that berries at berry thinning stage do not have color differences among berries, and that geometric information such as size, shape, and position, as well as global contextual information, are more important, the outlines for berries other than the target one are also drawn in the AF image to make it easier to capture this information. The generation of AF image is performed as the downstream of the instance segmentation task which outputs the location of berries and bunch (bounding box), the mask of berries and bunch. The mechanical behind attention forcing is simple yet efficient.

Figure 1.7 depicts the application of the proposed automatic removing berry identification technique to real berry removing task. Considering a huge deep neural network (DNN) model is required for obtaining the mask of berries, a server-based approach is adopted. The AI server is responsible for executing the instance segmentation model and the automatic removing berry identification. Afterward, the result is sent back to OSTHMD to show the detected berries and their probabilities to be removed.



a) OSTHMD **b) OSTHMD's information for farmer**

Figure 1.7 Applying the proposed automatic removing berry identification technique in table grape thinning. The detected berries and their probabilities for being removed are visualized with colors on OSTHMD. The bounding box indicate the berry with highest probability of being removed.

1.2 Contributions

To summarize, this dissertation presents novel solutions to handle real pain problems and improve the efficiency in table grape cultivation which determines the final product quality. End-to-end inflorescence measurement for supporting table grape trimming is proposed. The contributions of this approach are summarized as follows:

1. A novel solution for estimating accurately the length of the operating inflorescence by combining state-of-the-art DNN models and originally designed image processing algorithms.
2. A novel grape inflorescence trimming support system using a cloud computing approach and OSTHMD, enabling naïve farmers to perform inflorescence trimming efficiently.

There are two challenges in berry thinning. The first is the berry counting problem, and the second is to determine which berry should be removed. Firstly, end-to-end automatic berry counting technique for supporting table grape thinning is proposed. Its contributions can be summarized as:

1. A novel data augmentation technique that can automatically generate training datasets that simulate the berry-thinning process. Because berry thinning is conducted once a year during a short period, collecting a large training dataset corresponding to different weather and location conditions is highly difficult but extremely important. The proposed method makes it possible to generate automatically a large annotated training dataset from a small dataset.
2. A novel location-sensitive object detection model, realized as an extension of the state-of-the-art instance segmentation DNN model, to detect the berries in a working bunch only. Location

invariant is a property of DNN models inherently realized through the pooling layers, making it possible to detect all objects with the learned features regardless of their locations in the images. Such a property, however, is undesirable for proposed berry-thinning support purpose, as it means the berries of not only the working bunch but all bunches in an image will be detected. The problem was solved by integrating location information into the Hybrid Task Cascade (HTC) instance segmentation model (Chen, Ouyang, et al. 2019).

3. A novel method to estimate the number of berries in a bunch from one single 2D image of the bunch. Because grape berries have a round shape and no distinguishing features that can be tracked individually, it is difficult and computationally expensive to track and count all individual berries. Proposed method succeeded in achieving a high prediction accuracy that can withstand practical use via a set of originally designed features detected from single 2D images.

Finally, an approach for automatically identifying the berry to be removed in table grape thinning using a DNN and a novel attention forcing technique was proposed. The contribution of this approach can summarize as follows:

1. A novel image preprocessing technique called 'attention forcing' which turns the removing berry identification problem into an image classification problem. By creating the AF images from the instant segmentation results of the grape bunch, it allows DNN to automatically consider the candidate berry from its shape, size, nearby density, and position among its neighbors as the classification criteria.
2. A novel post-processing technique to make the system present consistent results to the farmer. After the removing berry prediction result is visualized, the farmer needs to recognize where it is in the bunch by holding the bunch for a few seconds. The proposed method allows to update the prediction results only when the famer changed the view of the bunch.

The system introduced in this dissertation can drastically improve the efficiency of table grape cultivation. The proposed method empowers beginner farmers to start inflorescence trimming and berry thinning without in-person coaching by expert farmers. Apart from novice farmers, the professorial farmer also needs the proposed system to improve their working productivity, especially the berry counting technique. Moreover, this dissertation also considers user experience to prevent eye fatigue and improve working performance.

1.3 Structure of the Dissertation

The structure of this dissertation is shown in Figure 1.2. The dissertation is organized as follows. Chapter 2 introduces related works including fruits and vegetables size prediction methods, berry detection, berry counting and removing berry prediction model. Chapter 3 proposes end-to-end inflorescence measurement for supporting table grape trimming with augmented reality. Chapter 4 offers an end-to-end automatic berry-counting technique in berry thinning. Automatic identification of berry to be removed in table grape thinning is introduced in Chapter 5. To confirm that the proposed system is suitable to use in the real grape field environment, Chapter 6 presents the experiments comparing the operation time, accuracy and final product quality between the unskilled farmers using the proposed system and the skilled farmers without using the proposed system. Chapter 7 concludes this dissertation, discussing the limitations, and introduces future work.

RELATED WORKS

This dissertation presents a smart farming system to support the cultivation of table grapes by addressing a challenging issue: how to use state-of-the-art Artificial Intelligence (AI) and Augmented Reality (AR) technology to support the inflorescence trimming and berry thinning tasks in table grape cultivation. Therefore, this chapter provides a brief survey of AI and AR techniques used in agricultural applications. In the remainder of the chapter, the existing methods related to the proposed system are introduced from four perspectives: fruits and vegetables size prediction, berry detection, berry counting, and decision-making for removing berry identification.

2.1 Artificial intelligence in agriculture

Smart agriculture is now gaining significant attention for tackling the challenges of agricultural production in terms of productivity, environmental impact, food security and sustainability (Gebbers and Adamchuk 2010; Kamilaris and Prenafeta-Boldú 2018). Since the global population is continuously increasing (Kitzes et al. 2008), food production must be increased massively, while sustainable farming procedures are required to protect the natural ecosystems (FAO 2009). To solve these issues, it is vital to have a better understanding of agricultural ecosystems that are complex, multivariate, and unpredictable by continual monitoring, measurement, and analysis of many physical characteristics and phenomena (Kamilaris and Prenafeta-Boldú 2018). Kamilaris and Prenafeta-Boldú summarize that the analysis of giant agricultural data (Kamilaris, Kartakoullis, and Prenafeta-Boldú 2017) and the use of new information and communication technologies (ICT) (Kamilaris et al. 2017) have several advantages to enhancing the existing tasks of management and decision/policy making by context, situation and location awareness.

The use of computer vision could address various challenges in agriculture domain (Liaghat and Balasundram 2010; Ozdogan et al. 2010). Analysing the images capturre in aricultural environment is an influential research topic of computer vision, and various modern computer vision technologies, such as those for image identification/classification, object detection, instance segmentation, anomaly detection, etc., have been used in diverse agricultural applications (Saxena and Armstrong 2014; Singh et al. 2016; Teke et al. 2013).

The conventional machine learning (ML) techniques used for analyzing images, such as K-means, support vector machines (SVM), artificial neural networks (ANN) (Kamilaris and Prenafeta-Boldú 2018) relies mainly on the so called hand crafted features which are extracted with well designed algorithms while a lately acquiring momentum is deep learning (DL) technology (Lecun and Bengio 1995; Lecun, Bengio, and Hinton 2015). DL, similar to ANN but with deeper network layers, allows to learn the features automatically. It enables more extensive learning capabilities and hence higher performance and precision. The Convolutional Neural Network (CNN) is a type of DL that is easier to train and generalize than networks with full connectivity between adjacent layers. It gained numerous practical successes is widely adopted in the computer vision community (Lecun, Bengio, and Hinton 2015).

The typical architecture of a CNN is shown in Figure 2.1. The feature maps are extracted by applying image convolution operation between input and convolution kernel. The pooling operation, such as max pooling (taking the maximum value of a local region) and average pooling (taking the avaerage of a local region) is the down-sampling method introduced to reduce the amount of data while aggregate useful information and discouraging overfitting problems (Hawkins 2004). The convolution and pooling also makes it possible to tackle the invariance of shift, scale, and distortion. Figure 2.2 depicts the convolution and pooling operations. Through the fully connection, the extracted features are passed to various downstream tasks. The padding enlarges the input with zero value to tackle information lost in the border. The stride is the step size to move the convolution kernel over the input to control the convolution result.

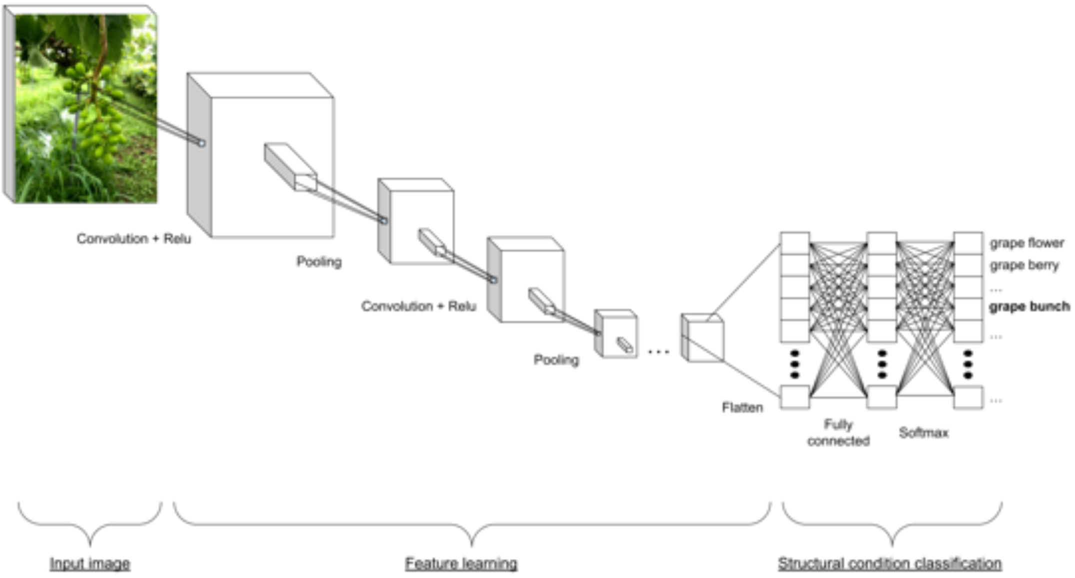
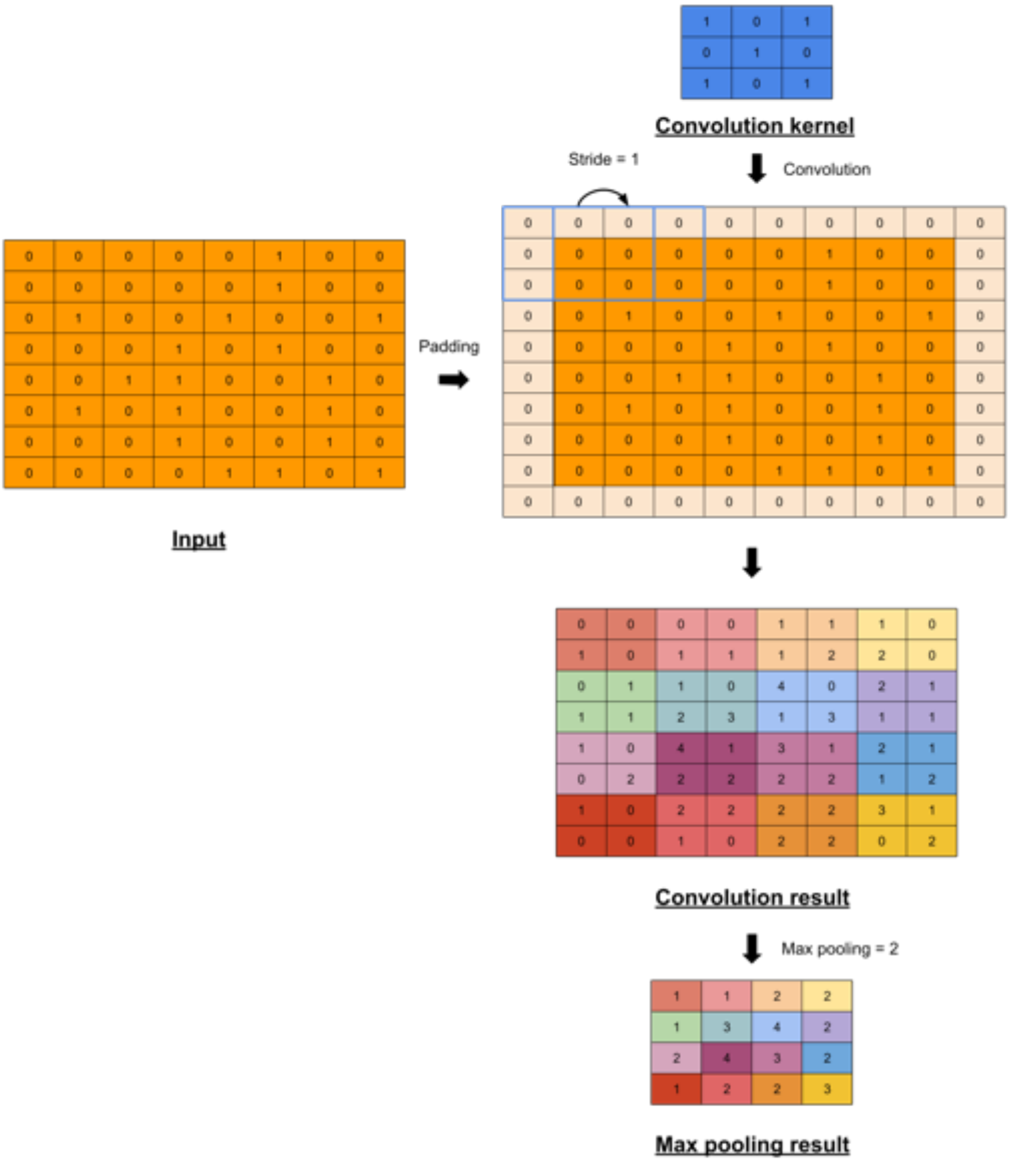


Figure 2.1 The typical architecture of CNN.

There are many agricultural applications using CNN. This section briefly introduces three downstream tasks related to the proposed system: image classification, object detection, and instance segmentation.

Firstly, image classification is to classify a whole image without detecting or locating some particular regions or objects in the image. The pioneering of modern CNN, LeNet-5, was used in hand-written digit image classification. LeNet-5 comprises seven trainable layers containing two convolutional layers, two pooling layers, and three fully-connected layers. Other typical examples of image classification models are Resnet (He et al. 2016), and EfficientNet (Tan and Le 2019). There are various image classification applications of CNN in agriculture (Kamilaris and Prenafeta-Boldú 2018). For instance, Dyrmann, Karstoft and Midtiby used CNN for classifying leaves of different plant species (Dyrmann, Karstoft, and Midtiby 2016), Kussul et al. proposed to classify crop type between wheat, maize, soybeans, sunflower, and sugar beet (Kussul et al. 2017). In this dissertation, the image classification technique based on Resnet model (He et al. 2016) was adopted to classify the berry to be removed during the berry thinning.



Convolution kernel

1	0	1
0	1	0
1	0	1

Input

0	0	0	0	0	1	0	0
0	0	0	0	0	1	0	0
0	1	0	0	1	0	0	1
0	0	0	1	0	1	0	0
0	0	1	1	0	0	1	0
0	1	0	1	0	0	1	0
0	0	0	1	0	0	1	0
0	0	0	0	1	1	0	1

Padding

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0
0	0	1	0	0	1	0	0	1	0
0	0	0	0	1	0	1	0	0	0
0	0	0	1	1	0	0	1	0	0
0	0	1	0	1	0	0	1	0	0
0	0	0	0	1	0	0	1	0	0
0	0	0	0	0	1	1	0	1	0
0	0	0	0	0	0	0	0	0	0

Convolution result

0	0	0	0	1	1	1	0
1	0	1	1	1	2	2	0
0	1	1	0	4	0	2	1
1	1	2	3	1	3	1	1
1	0	4	1	3	1	2	1
0	2	2	2	2	2	1	2
1	0	2	2	2	2	3	1
0	0	1	0	2	2	0	2

Max pooling result

1	1	2	2
1	3	4	2
2	4	3	2
1	2	2	3

Figure 2.2 Procedure of a two-dimensional CNN.

The other two main tasks are object detection and instance segmentation. Unlike the image classification that the whole image is classified as one class. Object detection and instance segmentation can recognize multiple objects in the image. Moreover, they provide the location of individual object as a bounding box in the image. The difference between object detection and instance segmentation is shown in Figure 2.3 and Figure 2.4. The instance segmentation provides the mask information (Mask) while object detection only provides the location of the object (BBox). There are mainly two approaches in object detection and instance segmentation: single-stage and multi-stage.

As shown in Figure 2.3 and Figure 2.4, in single-stage approach, after extracting the features maps by the backbone CNN model such as Resnet (He et al. 2016), the model directly predicts the class probability, bounding box coordinates (x_1, y_1, x_2, y_2), and mask of the objects. Well known single-stage object detection models include the YOLO(You Look Only Once) family, YOLO v1 (Redmon et al. 2016), YOLO v2 (Redmon and Farhadi 2017), YOLO v3 (Redmon and Farhadi 2018), YOLO v4 (Bochkovskiy, Wang, and Liao 2020), YOLO v5 (Jocher et al. 2021), and CornerNet (Law and Deng 2020). Examples of single-stage instance segmentation are YOLACT (Bolya et al. 2019) and YOLACT++ (Bolya et al. 2022).

The multi-stage approach is shown in Figure 2.5. There are many stages of the prediction through the cascade structure. The main idea of the multi-stage approach is resampling the CNN features maps through the pooling layer (Pooling). First, the region proposals (RPN) are predicted using features maps from the backbone CNN model. The RPN will give the regions of interest (ROI) area, which is usually the foreground objects, while discarding the background. Then only the features maps in ROIs from the RPN network are used to predict the class, bounding box, and mask through pooling layer. By cascading pooling operation, prediction accuracy can be increased. Examples of multi-stage object detection include Faster R-CNN (Ren et al. 2017), Cascade R-CNN (Cai and Vasconcelos 2019). Examples of multi-stage instance segmentation include Mask-RCNN (He et al. 2017), Hybrid task cascade for instance segmentation (Chen, Pang, et al. 2019), DetectoRS (Qiao, Chen, and Yuille 2020).

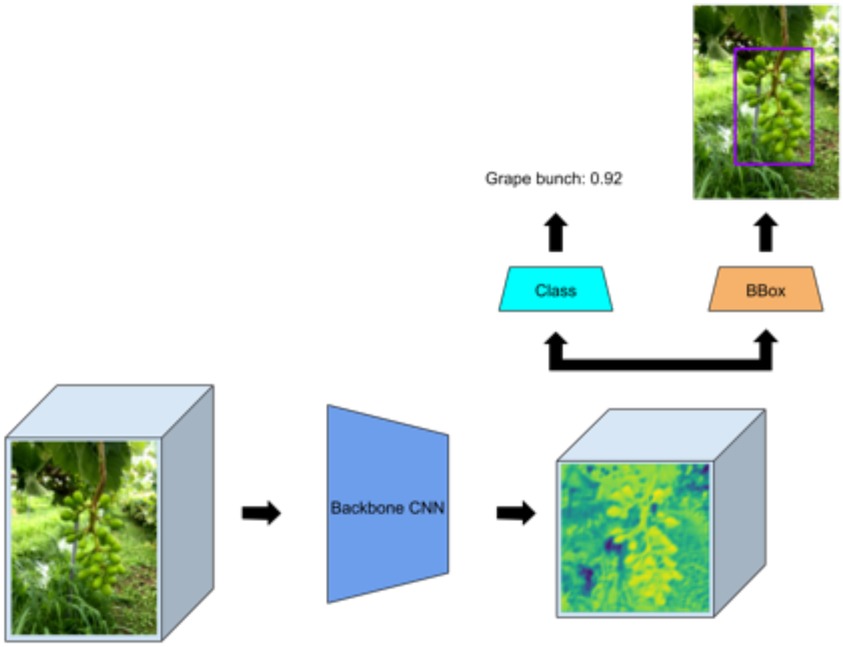


Figure 2.3 Single-stage object detection.

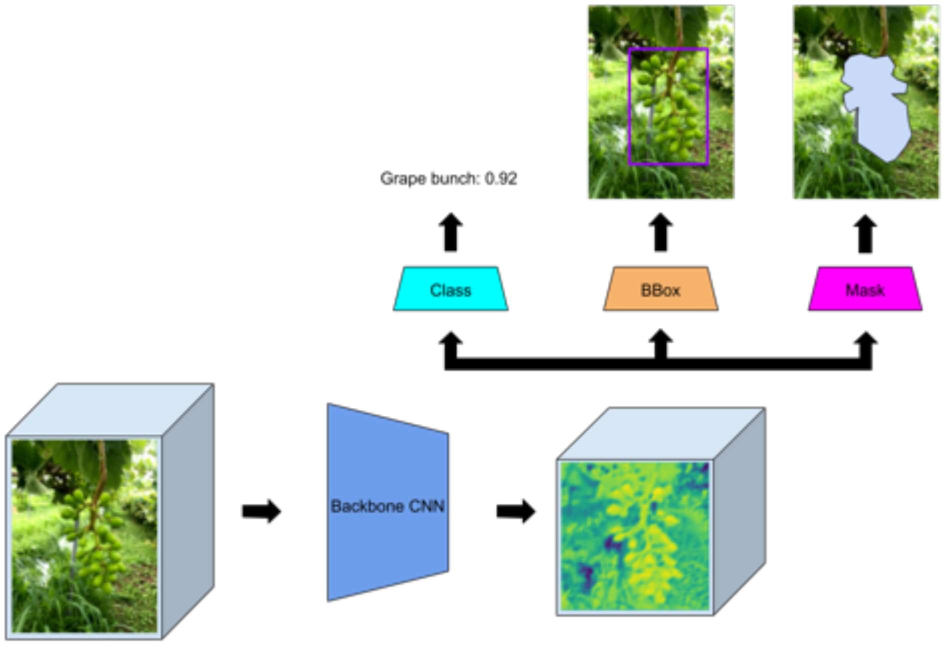


Figure 2.4 Single-stage instance segmentation.

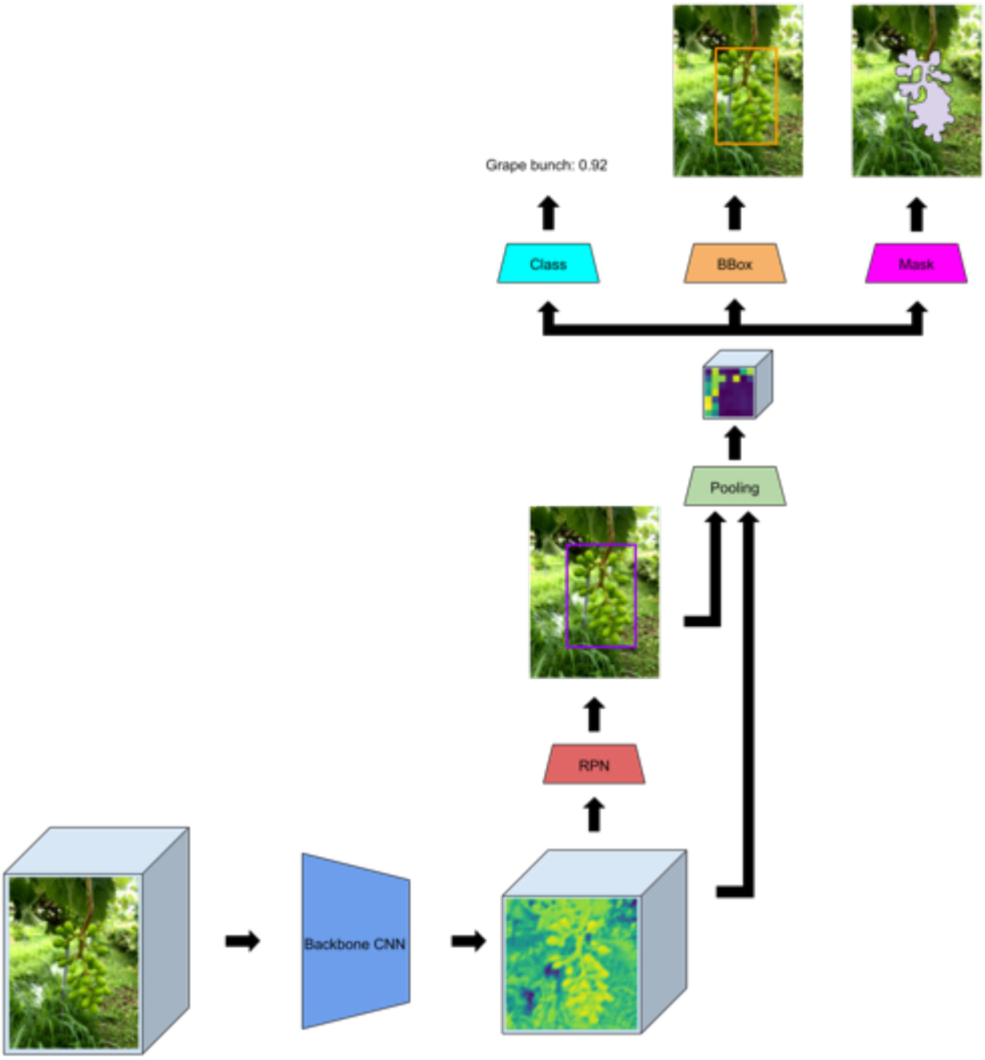


Figure 2.5 Multi-stage instance segmentation.

There are numerous research have been conducted to apply object detection and instance segmentation technologies in agriculture field (Kamilaris and Prenafeta-Boldú 2018). Yield prediction is one of the most popular applications. Technologies for automatically detecting various fruits, such as grapes (Duan et al. 2017; Nellithimaru and Kantor 2019; Santos et al. 2020; Zabawa et al. 2019), and sweet peppers (Sa et al. 2016), have been developed. Object detection should be used in case only bounding box information is required because object detection has lower complexity than instance segmentation which makes the model size smaller and can run faster. In inflorescence measurement proposed for supporting table grape, the size of calibrator, the diameter of the scissor's crew is needed. It can be done by selecting the longer side of the bounding box. Hence the object detection technique, YOLO v5 (Jocher et al. 2021), was used in this task. However, the mask information is required for the downstream tasks, such as computing the inflorescence length from the inflorescence mask or predicting the berry counting using extracted features from berry masks. Therefore instance segmentation techniques, HTC (Chen, Pang, et al. 2019) and DetectoRS (Qiao, Chen, and Yuille 2020), which are state-of-the-art models, were adopted at the implementation.

2.2 Augmented reality in agriculture

Augmented reality (AR) is a technology that provides immersive experiences by superimposing virtual objects over real-world objects (Azuma 1997). There are typically three ways to experience AR: using head mounted display such as Microsoft HoloLens$^{\text{TM}}$, Epson Moverio$^{\text{TM}}$, using hand hold mobile devices such as smart phone or some special equipment in a setup environment (Bimber and Raskar 2006; Danielsson, Holm, and Syberfeldt 2020). Modern agriculture system is usually highly sophisticated and benefit the technologies from various fields, such as optimal farm management, precise climate forecast, and nutrition science (Xi, Adcock, and McCulloch 2018). There are many prospects where augmented reality can impact.

For the AR applications using handhold devices, a geographic information systems (GIS) to help viticulturists accurately understand the parameters that affect their yields and quality of grapes from different vineyards via laptop PC was introduced in 2005 (King, Piekarski, and Thomas 2005). The weed economic thresholds system (Vidal and Vidal 2010) using image recognition to identify and quantify weeds by species and software for herbicide selection based on weed density was introduced in 2010. The identification system which suggests appropriate pesticides and treatments for pest management using AR via mobile phone was developed (Nigam, Kabra, and Doke 2011). Okayama and Miyawaki developed a smart garden system that uses AR to visualize guidance of farming operations via tablet PC (Okayama and Miyawaki 2013). There are other prototypes to identify a plant and provide helpful information to the farmer regarding that plant through a mobile device using AR technology (Katsaros and Keramopoulos 2017; Neto and Cardoso 2013).

However, for the tasks in which a farmer needs both hands to perform their general operations, handhold devices should be avoided. HMD based approach provides an ideal solution to those applications. As AR applications using HMD, Huuskonen and Oksanen developed a technology to provide situational awareness in supervising autonomous tractors (Huuskonen and Oksanen 2019). Santana-Fernández et al. employed AR to support farmers in navigating the field optimally (Santana-Fernández, Gómez-Gil, and del-Pozo-San-Cirilo 2010). Another guidance system is designed to support night-time farming (Kaizu and Choi 2012). A soil sample collecting supporting system using AR was also invented (Huuskonen and Oksanen 2018). Recently, Inoue et al. has developed a system that allows viticulturists to investigate trellising-style vineyards' vegetation condition using HMD. Nevertheless, no related works have employed AR technology using HMD to support inflorescence trimming and berry thinning tasks in table grape cultivation.

2.3 Fruit and vegetable size prediction

It is reported that the sizes of fruits and vegetables are a major factor in deciding yield at harvest (Kaack and Lindhard Pedersen 2010; Li et al. 2015; Stajnko, Lakota, and Hočevar 2004; Tijskens et al. 2020). Table 2.1 shows summary of fruit/vegetable size prediction. Existing size prediction methods can be classified into two major approaches: indirect and direct. For the indirect method, environmental factors and growing time have been employed to estimate size. The diameter and length of apples are simulated using physiological development time (PDT), an indicator that combines important environmental factors, such as temperature, evaporation potential, and photoperiod (Kaack and Lindhard Pedersen 2010; Li et al. 2015). Tijskens et al. (Tijskens et al. 2020) adapted the von Bertalanffy model (Bertalanffy 1938) to include the growth rate constant, the time after fruit set, the time of development, and the reference diameter to estimate tomato diameter. The indirect method is suitable for yield estimations, representing how many products the farm can produce in the season. Nevertheless, it is not suitable when it is necessary to know individual fruit lengths precisely. For the direct approach, computer vision is the main technology employed to measure plant size. Stajnko et al. (Stajnko, Lakota, and Hočevar 2004) used a thermal camera to capture images of apples. After applying a thresholding operation, they use the longest segment to calculate the diameter of the apple. However, the thermal image is processed offline; the farmer cannot get the apple diameter in real time. The other existing studies concerning the measurement of plant structure have been developed based on 3D image technologies, such as structure from motion (SfM) (Lu et al. 2020; Yang and Han 2020) and 3D laser scanner (Dassot, Fournier, and Deleuze 2019; Huang, Zheng, and Gui 2021). A laboratory setup or special capture device is needed for the 3D-based plant structure measurement. Besides, 3D reconstruction is time consuming and usually cannot be performed in real time. For the above reason, existing studies are not suitable to operate in the real field. Chapter 3 of this dissertation proposes a method for measuring the inflorescence to supporting table grape trimming with AR technology on a real grape yard in real time without special equipment setup. It presents a novel solution for estimating accurately the length of the operating inflorescence by combining state-of-the-art DNN models and originally designed image processing algorithms. Furthermore, a novel grape inflorescence trimming support system using a cloud computing approach and OSTHMD, enabling naïve farmers to perform inflorescence trimming efficiently.

Table 2.1 Summary of fruit/vegetable size prediction.

Metric	**Indirect**	**Direct**			
	Li et al.	**Stajnko et al.**	**Yang et al.**	**Huang et al.**	**Proposed**
Based approach	Environmental factors and growing time	Thermal camera	Structure from motion	3D laser scanner	DNN from single image
Suitable for yield estimations	✅	✅	✅	✅	✅
Suitable for precise individual fruit size measurement	❌	✅	✅	✅	✅
Not requiring laboratory setup	✅	✅	❌	❌	✅
Real-time processing	✅	❌	❌	❌	✅
Accuracy	MAE 0.20 cm	R^2 ~0.70	RMSE 2.43	R^2 0.74	MAE 0.19 cm

2.4 Berry detection

In this section, berry detection approaches are summarized in Table 2.2. Considering the round shape of the grape, the circle Hough Transform (CHT) has been employed to detect grape berries. Roscher at el. (Roscher et al. 2014) introduced the CHT to detect grape berries in the natural scene, while Liu at el. (Liu, Whitty, and Cossell 2015) employed the CHT for preprocessing in the 3D reconstruction of a grape bunch from a single image. In addition, Rudolph at el. (Rudolph et al. 2019) applied the CHT during post-processing to filter the flower bunch detected from the DNN network. However, a major problem of the CHT-based approach is that it cannot detect berries partially occluded by other berries. Reis et al. (Reis et al. 2012) and Luo at el. (Luo et al. 2016) proposed a system for detecting grape bunches in the natural environment based on the color mapping approach. Aquino et al. (Aquino et al. 2015, 2017; Aquino, Barrio, et al. 2018; Aquino, Millan, et al. 2018) proposed a method for estimating the number of grapevine berries and flowers using image analysis based on the h-maxima transform. Nuske at el. (Nuske et al. 2014) and Perez and Zavala (Pérez-Zavala et al. 2018) use feature descriptors, such as histograms of gradients (HoGs), fast retina keypoint (FREAK), local binary patterns (LBPs), and scale-invariant feature transform (SIFT), to detect the berries. However, the above approaches may not operate in a natural field with unconstrained illumination conditions and shadows. Such a problem can be solved with a DNN, because the image feature can be trained in the model, not just by using the specific range of the color value or specific hand-craft features to distinguish the objects (Santos et al. 2020; Zabawa et al. 2019). Typically, the approaches based on semantic segmentation are not designed to count object instances in the image, as the result of semantic segmentation is pixel-wise and overlapping objects of the same class cannot be distinguished. Zabawa et al. (Zabawa et al. 2019) tried to solve such a problem when applying semantic segmentation to grape berry detection by introducing a new edge class object surrounding the individual grape berry. However, because the edge is a small object, it is difficult to detect all edge pixels surrounding the berry. The method based on instance segmentation was designed to give an output comprised of the bounding box, classification, and pixel mask, thus immediately counting the individual objects. Santos et al. (Santos et al. 2020) used instance segmentation to detect a grape bunch, but the detection of berries was not addressed in their study. Most importantly, none of the recent DNN-based approaches (Nellithimaru and Kantor 2019; Santos et al. 2020; Zabawa et al. 2019) introduced a method for focusing on a particular bunch, which is crucial in supporting the table grape-thinning task. This dissertation proposes a method integrating location information into the HTC model (Chen, Ouyang, et al. 2019) to focus only on a particular bunch. The extended model is efficient enough to detect partially occluded berries real time in the natural scenes without requiring black background.

Table 2.2 Summary of berry detection.

Metric	**Roscher et al.**	**Aquino et al.**	**Pérez-Zavala et al.**	**Santos et al.**	**<u>Proposed</u>**
Approach based	CHT	h-maxima transform	feature descriptors	DNN	DNN
Component detected	Berry	Berry, Bunch	Berry, Bunch	Bunch	Berry, Bunch
Can detect partially occluded berries	❌	✅	✅	❌	✅
Not requiring black background	✅	❌	✅	✅	✅
Real-time processing	❌	✅	❌	❌	✅
Focus only on working bunch	❌	✅	❌	❌	✅
Accuracy	R^2 0.88	Recall 0.89, Precision 0.95	Bunch recall 0.80, Bunch precision 0.80, Berry recall 0.88, Berry precision 0.99	F_1 0.91	CD 96.55%, MC 2.79%

2.5 Berry counting

In this section, berry counting approaches are summarized in Table 2.3. While the proposed method can operate using a mobile device in a real field, existing research dealing with the number of berries in 3D bunches has required a laboratory setup or special capture devices. For instance, Liu et al. (Liu, Whitty, and Cossell 2015) required a plain background to apply Otsu's binarization while rotating the grape bunch. Ivorra et al. (Ivorra et al. 2015) needed constant light intensity, so they installed a stereo camera using four pairs of fluorescent tubes to afford the illumination. Scholer et al. (Schöler and Steinhage 2015) installed a laser scanner on a robot arm to scan the grape bunch. Because 3D reconstruction operation is time consuming, it is not appropriate for real grapevine yard application. Besides, it is difficult and computationally expensive to track and count all individual berries because grape berries have a round shape and no distinguishing features that can be tracked individually. This dissertation presents the novel method to estimate the number of berries in a bunch from one single 2D image of the bunch. The proposed method succeeded in achieving a high prediction accuracy than 3D laser scanner approach. That can withstand practical use via a set of originally designed features detected from single 2D images. Moreover, the proposed system can operate in actual grape field in real time without requiring laboratory setup.

Table 2.3 Summary of berry counting.

Metric	**Liu et al.**	**Ivorra et al.**	**Schöler and Steinhage**	**<u>Proposed</u>**
Based approach	Single image	Stereo vision	3D laser scanner	Single image
Not requiring laboratory setup	❌	❌	❌	✅
Real-time processing	❌	❌	❌	✅
Operate in real grape field	❌	❌	❌	✅
Accuracy	Average absolute error 12.4%	R^2 0.71	MAE 13.02 berries	MAE 2.81 berries

2.6 Automatic identifying berries to be removed

Automatic decision-making is one of the most important and but challenging technology in smart agriculture and researches have been conducted for various fruits and vegetables such as grape (Botterill et al. 2017), strawberry (Xiong et al. 2020), sweet pepper (Arad et al. 2020), apple (Karkee et al. 2014). Harvesting (Arad et al. 2020; Font et al. 2014; Ji et al. 2012; Silwal et al. 2017; Xiong et al. 2020) and pruning (Botterill et al. 2017; Karkee et al. 2014) are the main applications for automatic decision-making tasks which are summarized in Table 2.4.

Some of the automatic decision-making are made by the color and shape of fruits and vegetables. For instance, an autonomous strawberry harvesting robot can decide which strawberry should be harvested by a color thresholding technique considering the significant differences of color between ripe strawberries, green strawberries, and green plants (Xiong et al. 2020). A sweet pepper harvesting robot can identify harvestable sweet pepper based on the color threshold by selecting yellow-colored ones while ignoring the green-colored sweet pepper (Arad et al. 2020). With the apple harvesting robot proposed by Silwal et al., using color thresholding (Ji et al. 2012), Circular Hough Transformation (CHT) and Blob Analysis (BA) in an iterative procedure (Silwal et al. 2017), was employed to identify ripe apple.

There is a study that applies advanced techniques in automated decision-making in pruning task based on an AI (Botterill et al. 2017). This research developed a robot system for the automatic pruning of grapevines. The AI system decides which grape cane needs to be pruned. The features used to classify grape cane are length, position, the angle from the head, distance below wires, and whether they grow from the head, the trunk, or another cane. Since the study use hand-craft features, it cannot be adapted to unconstrained image capturing environments.

To summarize, none of the existing decision making technology work on grape berry thinning. Furthermore, the above studies consider only the candidate object individually. The attributes of neighborhood objects as long as their position or density weren't used as the classification criteria. It's challenging to design the procedure that enables the AI learn to classify which berry should be removed. The AI should automatically consider the candidate berry from not only its own shape, size, but also the berry density and position in its neighborhood. Chapter 5 of this dissertation proposes an approach for automatically identifying the berry to be removed in table grape thinning using a DNN and a novel attention forcing technique. The novel image preprocessing technique, “Attention Forcing” (AF), turns the removing berry identification problem into an image classification problem. By creating the AF images from the instant segmentation results of the grape bunch, it allows DNN to automatically consider the candidate berry from its shape, size, nearby density, and position among its neighbors as the classification criteria. Furthermore, the proposed method allows updating the prediction results only when the farmer changes the view of the bunch by using a novel post-processing technique to make the system present consistent results to the farmer.

Table 2.4 Summary of automatic decision-making in fruits / vegetables cultivation.

Metric	**Xiong et al.**	**Arad et al.**	**Ji et al.**	**Silwal et al.**	**Karkee et al.**	**Botterill et al.**	**<u>Proposed</u>**
Fruits and vegetables	Strawberry	Sweet pepper	Apple	Apple	Apple	Grape	Grape
Task	Harvesting	Harvesting	Harvesting	Harvesting	Branch pruning	Cane pruning	Berry thinning
Approach based	Color thresholding	Color thresholding	Color thresholding	CHT and BA	Rule-based	AI with handcrafted features	AI with learnable features
Accuracy	50%-97.1%, depending on the growth situations	61%	89%	84%	Removing long branches 85%	-	88.02%

2.7 Summary

This chapter introduced AI and AR techniques in agricultural applications. Researches related to fruits and vegetables size prediction are reviewed. Then, existing technologies which can be employed for berry detection, which is a mandatory process for the subsequent tasks such as berry counting and removing berry identification, were introduced. Existing works related to berry counting and the current challenges of the topic, especially how to deal with the unconstrained image capturing condition of real environment were stated. Finally, the related works for decision-making in fruits and vegetables cultivation are also introduced. Since the main aim of the proposed method is to develop a smart farming system to support the cultivation of table grapes using AI and AR in real grape fields, this dissertation provides comprehensive evaluation metrics, including accuracy, product quality, time efficiency, and user experience, while other related works focus on evaluating the system with fewer metrics.

END-TO-END INFLORESCENCE MEASUREMENT FOR SUPPORTING TABLE GRAPE TRIMMING WITH AUGMENTED REALITY

This chapter introduces an end-to-end inflorescence measurement technique proposed for supporting table grape trimming with augmented reality. Figure 3.1 shows the framework of the proposed technique. The framework comprises four parts. First, DetectoRS, a state-of-the-art instance segmentation DNN model with Recursive feature pyramid and Switchable atrous convolution (Qiao, Chen, and Yuille 2020), is used to detect the inflorescence, and the scissors from the image captured with the camera on the OSTHMD. Second, a state-of-the-art real-time object detection network called You only look once version 5 (YoloV5) (Jocher et al. 2021) is used to detect the calibrator, which is the scissors' screw. Third, an originally designed inflorescence length-estimating algorithm is applied to the detected inflorescence area to compute the current length and the desired length of the operating inflorescence, and the final step is to visualize the estimated lengths on the OSTHMD. The remainder of this chapter goes into the specifics of each system component.

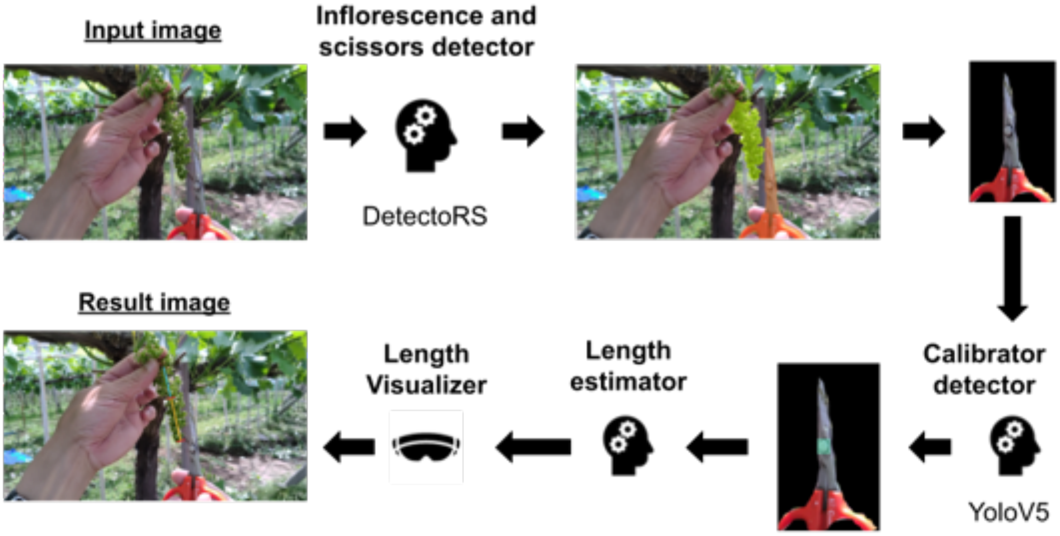
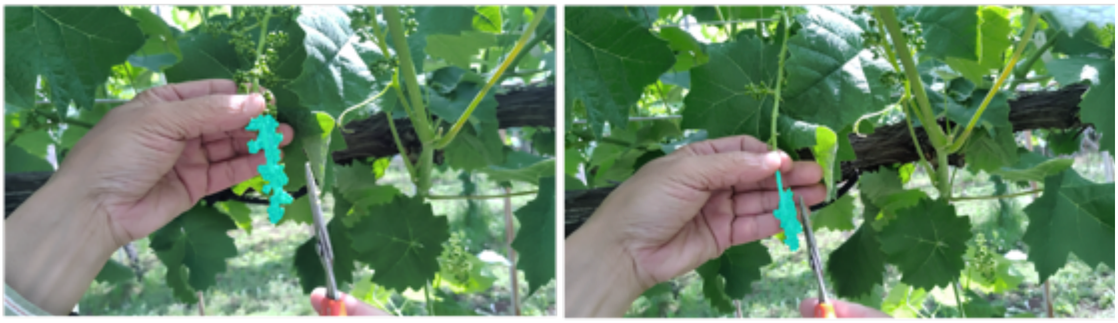


Figure 3.1 The framework of the proposed end-to-end inflorescence measurement technique and AR system for table grape trimming.

3.1 Methodology

3.1.1 Detecting the inflorescence and scissors

To compute the length of the inflorescence, it is necessary to detect the inflorescence area as precisely as possible. In addition, as will be introduced in 3.1.2, the area information of the scissors is needed for detecting its screw as the calibrator. For this reason, the state-of-the-art instant segmentation model DetectoRS was employed, which provides the pixel mask in addition to the bounding box of the detected object for detecting both the inflorescence and scissors. DetectoRS uses a multi-stage model architecture with which the detector is trained sequentially, applying the output of the detector as a training set for the subsequent stage. Such an architecture significantly improves detection accuracy (Cai and Vasconcelos 2019). As shown in Figure 1.3, during the trimming process, the farmer usually uses one hand to hold the inflorescence at a position above the remaining part of the inflorescence. Therefore, the DetectoRS is trained to detect the inflorescence under this holding position, as shown in Figure 3.2. This ensures that the remaining part of the inflorescence is detected dynamically during the trimming process.



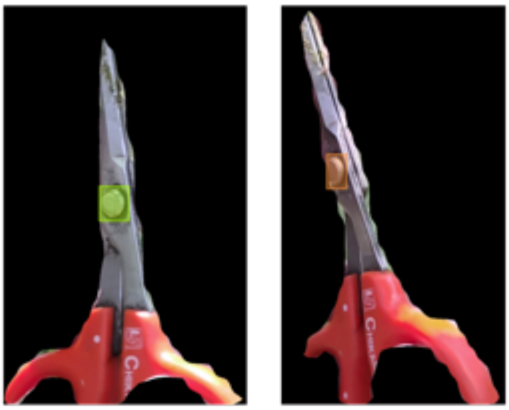
(a) Detected inflorescence before trimming **(b) Detected inflorescence after trimming**

Figure 3.2 Examples of the detected inflorescence are at different stages of trimming.

3.1.2 Detecting the calibrator

The calibrator, the scissors' screw, is used to calculate the inflorescence length. Figure 3.3 shows the detected screw on the scissors. The main idea of using the screw as the calibrator is based on the fact that the round shape of the screw makes it possible to obtain the diameter of the screw invariant to the orientation of the scissors by using the longer side of the bounding box given by YoloV5, as can be observed in Figure 3.3 (a) and (b). Besides, scissors are a necessary tool for the farmer to trim the inflorescence. When a farmer would like to know the length of an inflorescence, they only need to place scissors on the same plane of the inflorescence. Thus, the proposed method does not require an additional calibrator that interferes with the farmer's task during the trimming process.

I detect the scissors with DetectoRS first and then detect the calibrator from the detected scissors image. Such a two-step approach makes it possible to raise the detection accuracy by discarding unnecessary information and focusing only on the region of interest. Using the scissors mask, The original input image is masked and cropped to obtain a smaller image only consisting of the scissors, as shown in Figure 3.3, and then use this image as the input to YoloV5 for detecting the screw as the calibrator. YoloV5 does not provide the object mask as the output, but it is known to be the fastest object detection model. Because only the bounding box is needed to compute the diameter of the screw, YoloV5 is used for taking its advantage in processing time.



(a) Front-facing scissors **(b) Rotated scissors**

Figure 3.3 Detected scissors' screw.

3.1.3 Estimating the current and target inflorescence lengths

Figure 3.4 shows the flowchart of the proposed inflorescence measurement technique. As described in previous paragraphs, first, the inflorescence and the scissors are detected using DetectoRS (Qiao, Chen, and Yuille 2020). Then, the detected inflorescence and detected scissors are processed separately. The connected components on the inflorescence mask are computed. If more than one connected component are found, then the convex hull of these components is generated to get a single merged region (Figure 3.5[c]). Thereafter, the major axis (Burger and Burge 2009) of the inflorescence mask (Figure 3.5[d]) is computed and the inflorescence length is computed from this major axis.

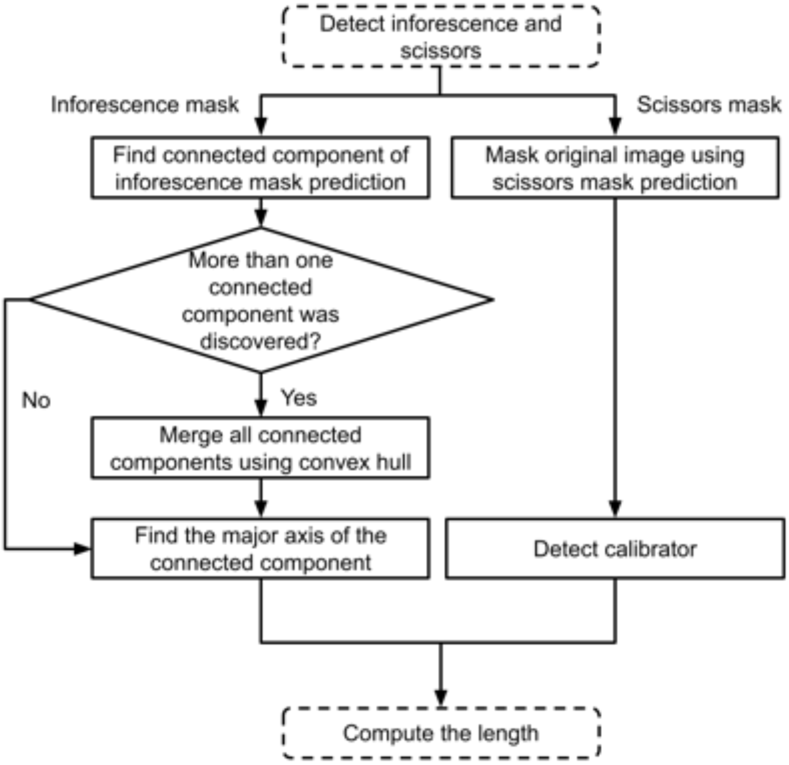
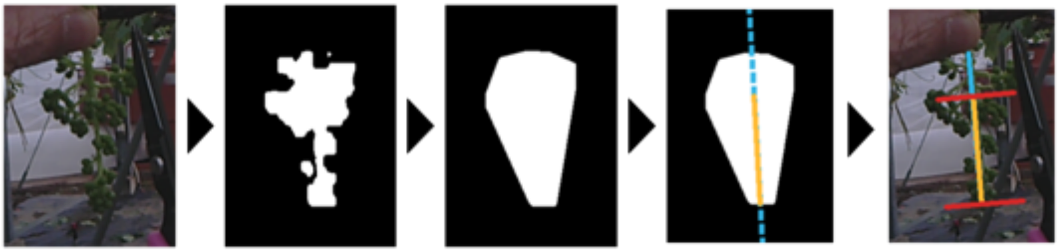

Detect inforescence and scissors
Inforescence mask
Scissors mask
Find connected component of inforescence mask prediction
Mask original image using scissors mask prediction
More than one connected component was discovered?
No
Yes
Merge all connected components using convex hull
Find the major axis of the connected component
Detect calibrator
Compute the length

Figure 3.4 Flowchart of the proposed inflorescence measurement technique.



(a) Input **(b) Mask** **(c) Convex hull** **(d) Inflorescence length computing** **(e) Result**

Figure 3.5 The proposed inflorescence measurement steps.

Figure 3.6 depicts the parameters used to calculate the inflorescence length. The parameters that are known from previous steps are as follows: detected bottom and top points, which are the bottom and top intersection between the mask and the line passing through the major axis of the inflorescence mask; target bottom point, which is the same as the detected bottom point; the angle between the y axis and the major axis in the *radian*; the target length in centimeters (*cm*), which is set by the grape variety; the diameter of the scissors' screw in *cm*, which is measured from the real scissors' screw; and the size of the scissors' screw in *pixels*, which is obtained from the calibrator detection step. The unknown values that need to be calculate thereafter are as follows: the detected current inflorescence length in *cm*, the target inflorescence length in *pixels*, and the target top point coordinates.

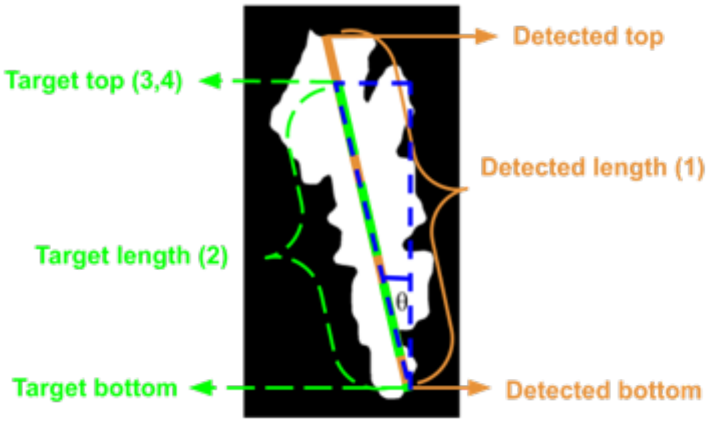


Figure 3.6 The proposed inflorescence measurement.

First, $DetectedLen_{cm}$, the detected current inflorescence length in *cm*, is calculated as follows:

$$DetectedLen_{cm} = \frac{ScrewDim_{cm} \times DetectedLen_{px}}{max\ (ScrewWidth_{px}, ScrewHeight_{px})} \tag{3.1}$$

Here, $ScrewDim_{cm}$ is the diameter of the screw in *cm*; $DetectedLen_{px}$ is the computed length of the detected inflorescence mask in pixels; $ScrewWidth_{px}$ and $ScrewHeight_{px}$ are the width and length of the screw in pixels, respectively; and max is the function to take the maximum of the two values. $TargetLen_{px}$, the target length in pixels, is calculated as:

$$TargetLen_{px} = \frac{TargetLen_{cm} \times DetectedLen_{px}}{DetectedLen_{cm}} \tag{3.2}$$

Here, $TargetLen_{cm}$ is the length of the target inflorescence in *cm*. Next, $TargetLen_{px}$ from (3.2) is used to calculate the target position, which is the top point of the target length and needs to be shown to the farmer, using (3.3) and (3.4).

$$TargetTopX_{px} = TargetBotX_{px} + (\sin(\theta) \times TargetLen_{px}) \tag{3.3}$$

$$TargetTopY_{px} = TargetBotY_{px} + (\cos(\theta) \times TargetLen_{px}) \tag{3.4}$$

Here, $TargetTopX_{px}$ and $TargetTopY_{px}$ are the target top point coordinates and θ is the angle between the y axis and the major axis, which has a range from $-\pi/2$ to $\pi/2$. The image to be shown to the farmer on the OSTHMD is shown in Figure 3.5(f). The desired length is shown as a yellow line, while the detected current inflorescence length is indicated with a light blue line.

3.1.4 Visualizing lengths on the OSTHMD

Figure 3.7 is the proposed user interface design on the OSTHMD. The top bar is designed to show the result, such as the detected current length, the desired length, and the network status. Hence, the farmer will be notified if there is a network connection trouble. Because displaying the entire input image on the whole screen in real time would cause occlusion to the real inflorescence and the farmer's operation, a small window is placed at the right bottom of the screen to show the input image to enable the farmer to confirm whether all parts of the operating bunch have been captured. By placing the result widget to the right of the center area, it makes the center area of the screen transparent to real-world objects, so that the farmer can see the real working inflorescence in the middle of the screen while also able to refer to the information in the result widget comfortably. Moreover, the guideline (red lines) was introduced to emphasize the top end of the desired length to help the farmer confirm intuitively where the position till which, the upper part of the inflorescence, should be trimmed.

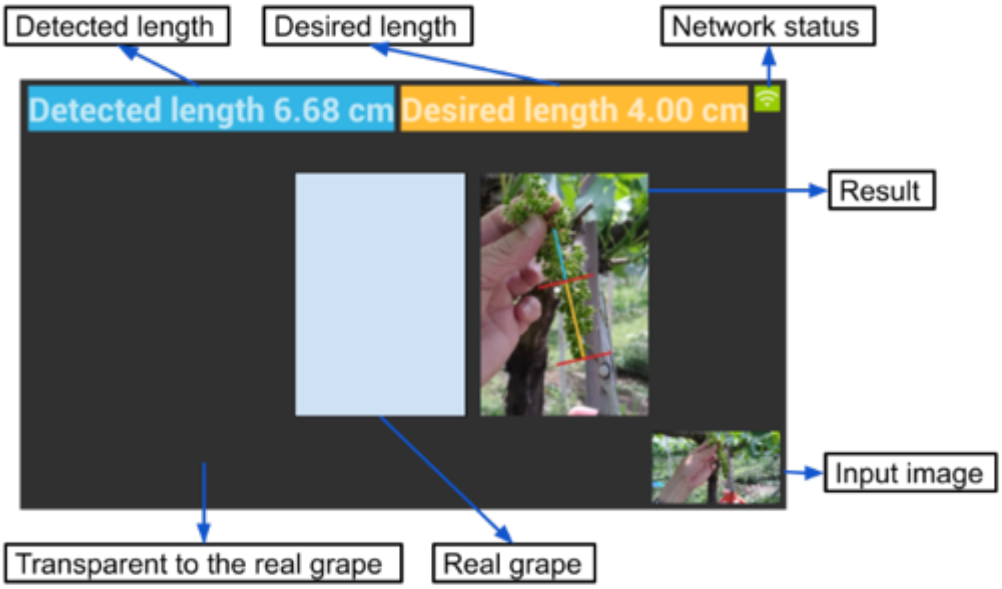


Figure 3.7 The proposed user interface on OSTHMD.

3.2 Experimental results and discussion

3.2.1 Dataset

Two farmers were asked for cooperation by installing cameras on their heads to capture the working scene throughout the grape-trimming task. Then, 200 inflorescence images were manually annotated to train DetectoRS (Qiao, Chen, and Yuille 2020) to detect the inflorescence and scissors. Each image has a resolution of 1,920 × 1,080 pixels, and each was rescaled to 1,333 × 800 pixels. After successfully training the inflorescence and scissors detection model, the model for scissors detection was used to detect the scissors and then the images were cropped to only contain the scissors' bounding box. The scissors' screw on these cropped images were annotated manually to train YoloV5 (Jocher et al. 2021) to detect the calibrator. Each scissors image was rescaled to 320 × 320 pixels to train the calibrator detection model. In the current implementation, the models were trained on a single Titan RTX GPU for more than 10 hours. 140 annotated images are used for training, and another 60 images are used for the test. The detection evaluation follows the approach from the Pascal VOC Challenges (Mottaghi et al. 2014) are 0.99 mAP (mean average precision) and 0.9971 mAP on DetectoRS, and YoloV5, respectively.

3.2.2 Implementation details

The hyper-parameters for training DetectoRS (Qiao, Chen, and Yuille 2020) and YoloV5 (Jocher et al. 2021) are shown in Table 3.1. The diameter of the scissors' screw (Dcm_{screw}) in this experiment is 0.8 centimeter. The desired length ($Lcm_{desired}$) of the inflorescence in this experiment is 4 centimeters. Figure 1.4 shows the communication structure of the proposed system. The OSTHMD, which is the Epson Moverio BT-2000, is connected to the pocket local 5G via the WIFI IEEE 802.11 b/g/n/a. The OSTHMD sends the captured image of 1,280 × 720 pixels along with a few additional parameters, such as the diameter of the scissors' screw and the desired length of the inflorescence, to the AI Server using a REST API. The data throughput was increased by using security authentication via JSON Web Token (JWT). With JWT, every time when send the request data to the AI server, the security information can be validated without database access or requiring additional memory capacity. As this architecture accesses a remote AI server from the table grape farm, the time interval for sending request data to the AI server from OSTHMD is set to 1 second.

Table 3.1 Hyper-parameters for training DetectoRS (Qiao, Chen, and Yuille 2020) and YoloV5 (Jocher et al. 2021).

Property	**Inflorescence and scissors detection model's value (DetectoRS)**	**Calibrator detection model's value (YoloV5)**
Optimizer	Stochastic gradient descent	Stochastic gradient descent
Iterations (K)	20	20
Decay	0.0001	Not set
Momentum	0.9	Not set
Batch size	1	16
Learning rate	0.00125	0.001

3.2.3 Evaluation of inflorescence measurement technique

The following mean absolute error (MAE) (Géron 2019) is used as the accuracy evaluation metric in this experiment:

$$MAE = \frac{1}{m}\sum_{i=1}^{m}\left|pd^{(i)} - gt^{(i)}\right| \tag{3.5}$$

Where m is the total number of inflorescences, $pd^{(i)}$ is the predicted length of the i^{th} inflorescence in the dataset, and $gt^{(i)}$ is its ground truth length.

The experiment was conducted at a real table grape yard. The local 5G network was installed at the site to connect the OSTHMD (Epson Moverio BT-2000) to AI server. Then, the participants are asked to report the length of the operating inflorescence shown on the OSTHMD while manually measuring the inflorescence length (ground truth) with a ruler to obtain the ground truth.

For 77 real inflorescences, all inflorescences were correctly detected. Then, the accuracy of proposed method was computed with (3.5). The MAE is only 0.19 *cm*, and the farmers commented that the estimation accuracy is high enough for use in real applications.

Because real-time processing is required to support efficient trimming, the models' inference time was observed. 1,041 inflorescence images were used to evaluate the inference time. The average time to detect the inflorescence and scissors using DetectoRS (Qiao, Chen, and Yuille 2020) is 0.16 seconds. The average time to detect the calibrator using YoloV5 (Jocher et al. 2021) is 0.01 seconds. Hence, the total average time for using DNN models is 0.17 seconds, which is around 5.88 frames per second (fps). From the above experiment result, it can be concluded that the two-step approach, which is to detect the scissors first and then use only the scissors image to detect the calibrator, is reasonable from the perspective of time efficiency. This is because YoloV5 (Jocher et al. 2021) was designed to detect an object in real time. Because only scissors area is used and other unnecessary area on the input image are discarded, the model size could be reduced by using the input image of a small size (312 × 312 pixels) while still maintaining a high detection accuracy. Moreover, the response time from the OSTHMD to the AI server was investigated. The local 5G network, which is still in the experiment stage, was employed in the experiment. 11,052 inflorescence images were sent (requests) over the local 5G network. The average response time is 1.25 seconds/image. It can be expected that a commercial-ready 5G system will significantly reduce the response time.

3.2.4 Evaluation of user experience

The qualitative evaluation was conducted by interviewing the grape farmers who performed trimming task with the proposed technology. The interview results show that they are satisfied with the design of the visualization shown in Figure 3.7. The information window, which was placed at the top of the screen, was easy to refer. Placing the result widget near the middle of the screen improved the visibility and adding the guidelines in the result widget enabled them to understand the current length and target length intuitively. Therefore, proposed system can improve the operability of inflorescence trimming. The response time is sufficient for real working scenarios. Some of the farmers said that using proposed system turned a boring task into a joyful task.

3.2.5 Limitation

One limitation of the proposed technology is that it requires the farmers to place the scissors at the same depth as the inflorescence. Another limitation is that the estimation accuracy may decrease if the shape of an inflorescence is severely curved. This is because proposed method approximates the length of the inflorescence with the major axis of the mask, while the circumscribed area of a severely curved inflorescence may not be well approximated with a major axis. Figure 3.8 shows an example of such a case.

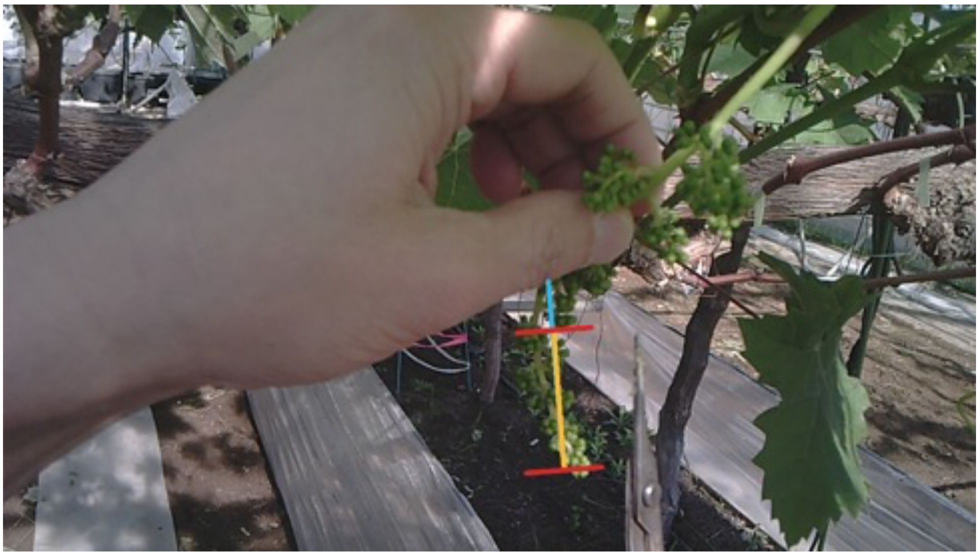


Figure 3.8 An example of severely curved inflorescence, with which the proposed method failed to predict its length accurately.

3.3 Summary

In this chapter, a technology proposed for building a functional application for supporting the grape trimming task in e real table grape farm environment is introduced. The novel end-to-end inflorescence measurement technology allows farmers to perform table grape trimming efficiently, as it is a significant task influencing the market value of table grapes. The proposed approach uses 2D images of the trimming scene without requiring any extra calibrators except for sessors which is a tool required for the trimming task. The experiment results demonstrate that proposed method could achieve an outstanding result in inflorescence measurement. The measurement accuracy and the inference time are sufficient for use in the real table grape environment. The OSTHMD was employed to capture images and offer guidance to farmers without interrupting their trimming tasks.

I plan to improve further the accuracy of the measurement by designing a new algorithm for dealing with highly curved inflorescences. I also plan to improve the user experience by using Microsoft HoloLens^{TM}. By using Microsoft HoloLens^{TM}, a wider field of view and overlaying on the real object can be expected.

END-TO-END AUTOMATIC BERRY COUNTING FOR TABLE GRAPE THINNING

Figure 4.1 depicts the framework of the proposed end-to-end automatic berry-counting technique. The framework consists of three parts: a DNN model that takes a captured 2D image as the input and detects the berries in a working bunch, a feature extractor that computes a set of carefully designed features from the detected berries, and a regression model that predicts the number of berries in the whole 3D bunch using the features from the feature extractor. For the DNN model, the HTC (Chen, Pang, et al. 2019), a state-of-the-art instance segmentation model, was made an extension to detect the berries only in the working bunch and to exclude other bunches. A new data augmentation technique is proposed to generate a large dataset to train this extended DNN model. To predict the number of berries in the whole 3D bunch, a set of features together with their extraction algorithms is carefully designed, and six different regression models are investigated. The details of each part of the framework are given in the remainder of this section.

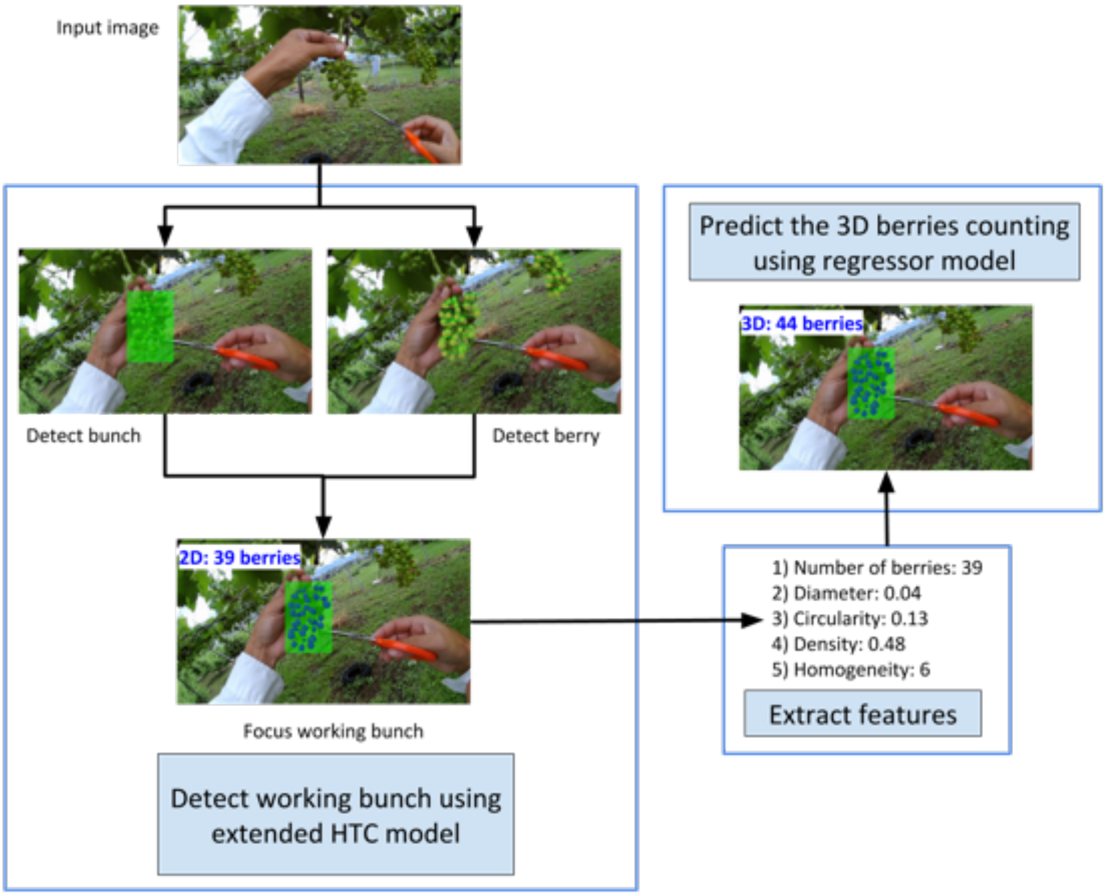
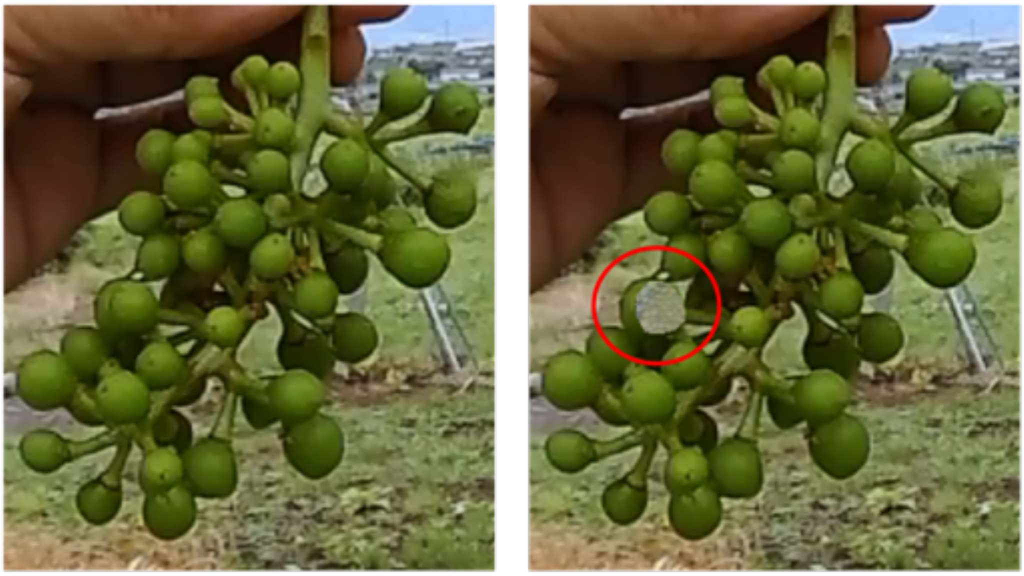


Figure 4.1 The framework of the proposed end-to-end automatic berry-counting technique for table grape thinning.

4.1 Methodology

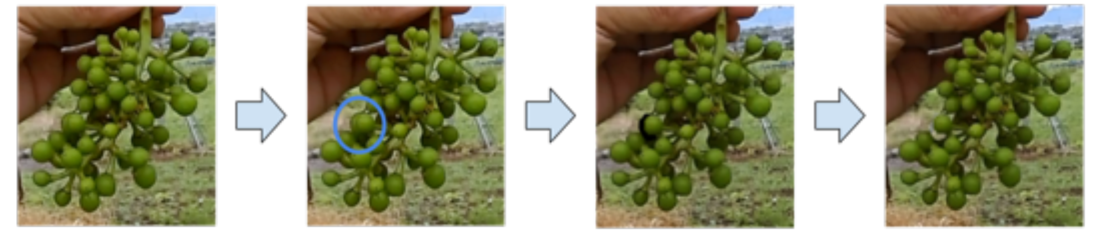
4.1.1 Data augmentation

Deep learning models have gained huge success in object detection tasks (Cai and Vasconcelos 2019; Chen, Ouyang, et al. 2019; He et al. 2017; Lin et al. 2017). However, to train a successful model, a large amount of labeled data is required. Because berry thinning is performed once a year during a short period, it is difficult to collect a sufficient number of images for training a model that can accurately detect berries during the whole process of berry thinning. Moreover, for the training of an instant segmentation model, the masks of individual grape berries are required. Generating such annotated data with manual labeling requires a huge amount of labor. To solve this problem, this study proposes a new data synthesis method to generate sufficient data from a small training set. The basic idea is to generate the images simulating the thinning process by removing berries gradually from an existing image. As shown in Figure 4.2, removing a front berry may result in an unnatural appearance of the berries partially occluded by this front berry. To avoid such an artifact, the proposed method first identifies the berry behind the front berry by computing the circularity of the berries. If the circularity is below a given threshold, we can judge that it is a partially occluded berry and it can be removed. To make the synthesized image look as natural as possible, a state-of-the-art image inpainting technology using a deep convolutional neural network (Iizuka, Simo-Serra, and Ishikawa 2017) is employed to fill the region of the removed berry. Figure 4.3 shows the process of proposed synthesis method. First, a partially occluded berry is identified by computing the circularity (Figure 4.3 [b]); then, this berry is removed (Figure 4.3 [c]); finally the removed region is filled with inpainting technology (Figure 4.3 [d]). This process can be repeated until all the partially occluded berries are removed, simulating the images captured during the thinning process. Two examples of synthesized images are shown in Figure 4.4.



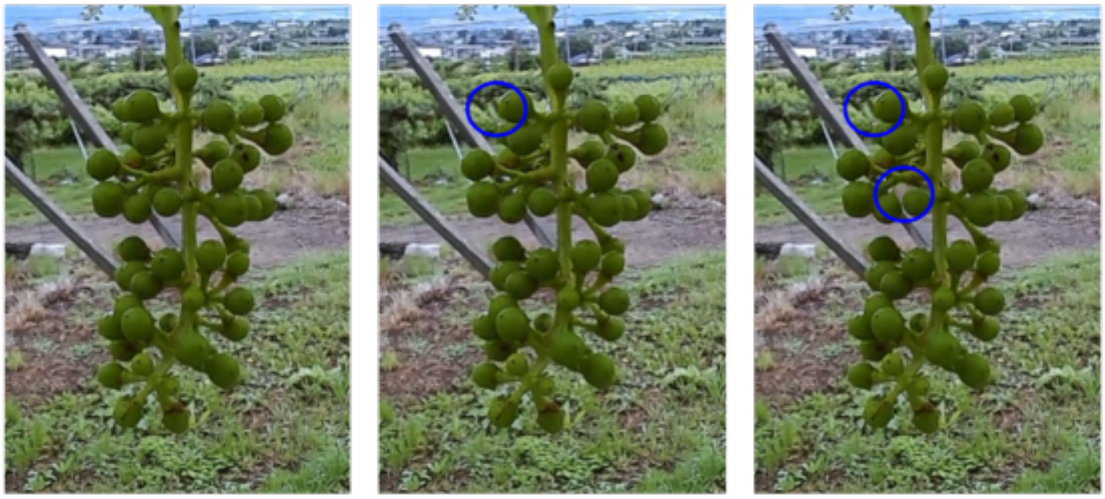
a) Original **b) Synthesized problem**

Figure 4.2 The problem occurs when synthesizing the image by removing the circular berry. The red circle is the inpainting area in which the berry was eliminated.



a) Original b) Identify target c) Remove berry d) Fill with inpainting

Figure 4.3 The process to synthesize new image data using the image inpainting technique.



a) Original b) Synthesized c) Synthesized

Figure 4.4 Comparison of the original image and its synthesized image. The blue circle is the inpainting area in which a berry was eliminated.

4.1.2 Automatic focusing on working bunch

1) Location sensitive HTC model

As depicted in Figure 1.6, this research aims to support farmers in effectively performing grape thinning by visualizing the number of berries in a working bunch. Therefore, the DNN model used for detecting berries should meet three requirements. First, it should be able to detect the berries only in the working bunch without detecting the berries in other bunches in the captured images. Second, it should detect the berries with a high accuracy without detecting the same berry multiple times. Third, as will be introduced in Part C of this section, the geometry features of berries are needed to predict the number of berries in a 3D bunch; therefore, it is desirable to obtain the accurate mask of individual berries. The third requirement indicates that it is necessary to use an instance segmentation DNN model. The second requirement cannot be met by any existing DNN models, as a DNN model is actually designed to be location-invariant to detect all objects with the learned features regardless of their locations. To solve the problem, this study proposes a new location-sensitive model by integrating explicit location information into the HTC, the state-of-the-art instance segmentation model proposed by Chen et al. (Chen, Ouyang, et al. 2019). Because the location information can also be viewed as a kind of feature distinguishing the berries in the working bunch from other objects in the image, the integration of location information into the DNN model can actually improve the detection accuracy, which contributes to meeting the first requirement.

Figure 4.5 depicts the network architecture of the original HTC model (Chen, Ouyang, et al. 2019). It consists of the CNN backbone network ('Backbone CNN') for extracting features; the region proposal network ('RPN') for predicting the location of objects in the input image; the pooling layer ('Pooling'), which is the cropped features from the backbone network using the mapping location from the RPN; classification branches ('Class') that predict the classes of objects; bounding box branches ('BBox') that predict the locations of objects in the input image; mask branches ('Mask') that predict the pixel-level masks of objects; and a semantic branch ('Semantic') that predicts pixel-level stuff segmentation for the whole image.

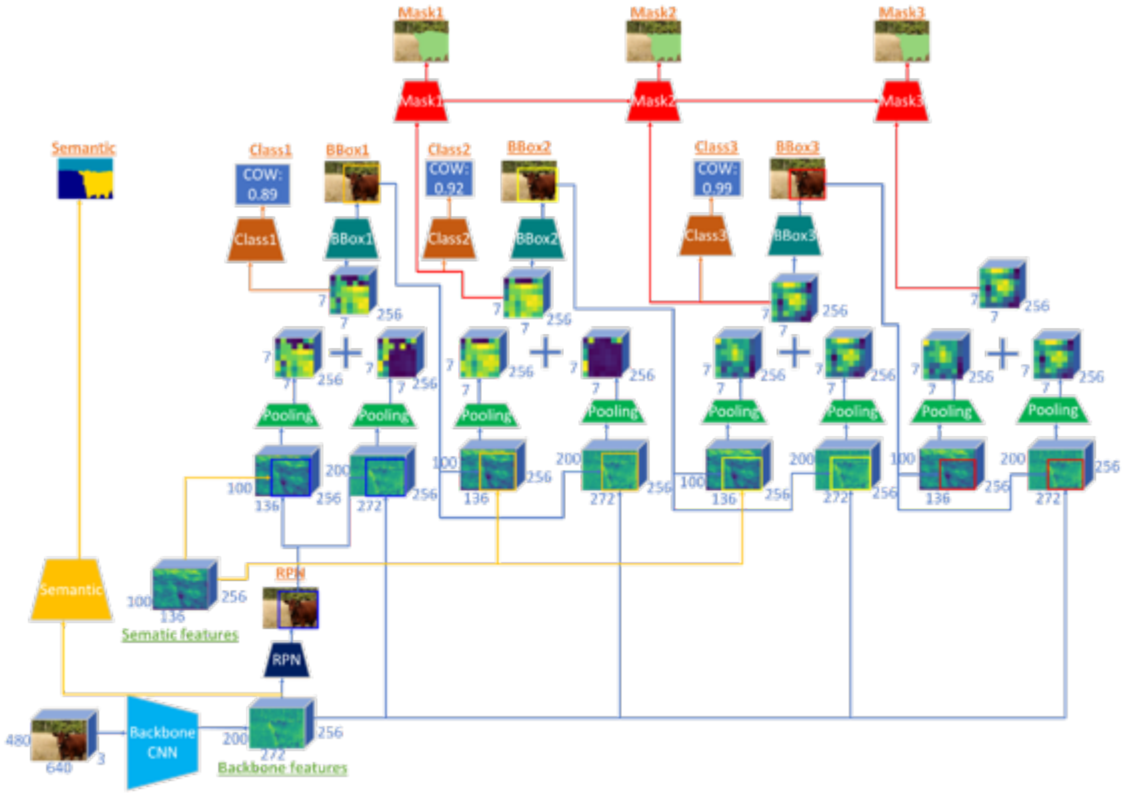


Figure 4.5 The structure of the HTC network (Chen, Ouyang, et al. 2019).

Figure 4.6 and Figure 4.7 show the proposed location-sensitive HTC models with location features integrated into the Fully Connected (FC) layer and into the HTC itself, respectively. In both models, the semantic segmentation branch from the original HTC has been excluded because only have two kinds of objects, the bunch and the berries, need to be considered and stuff segmentation of the whole image is not needed. Figure 4.6 shows the location feature from the RPN, BBox1, and BBox2, which were represented in terms of (x1, y1, x2, y2) and were fed as the input to the Classes and BBoxes, along with the features from the FC layer.

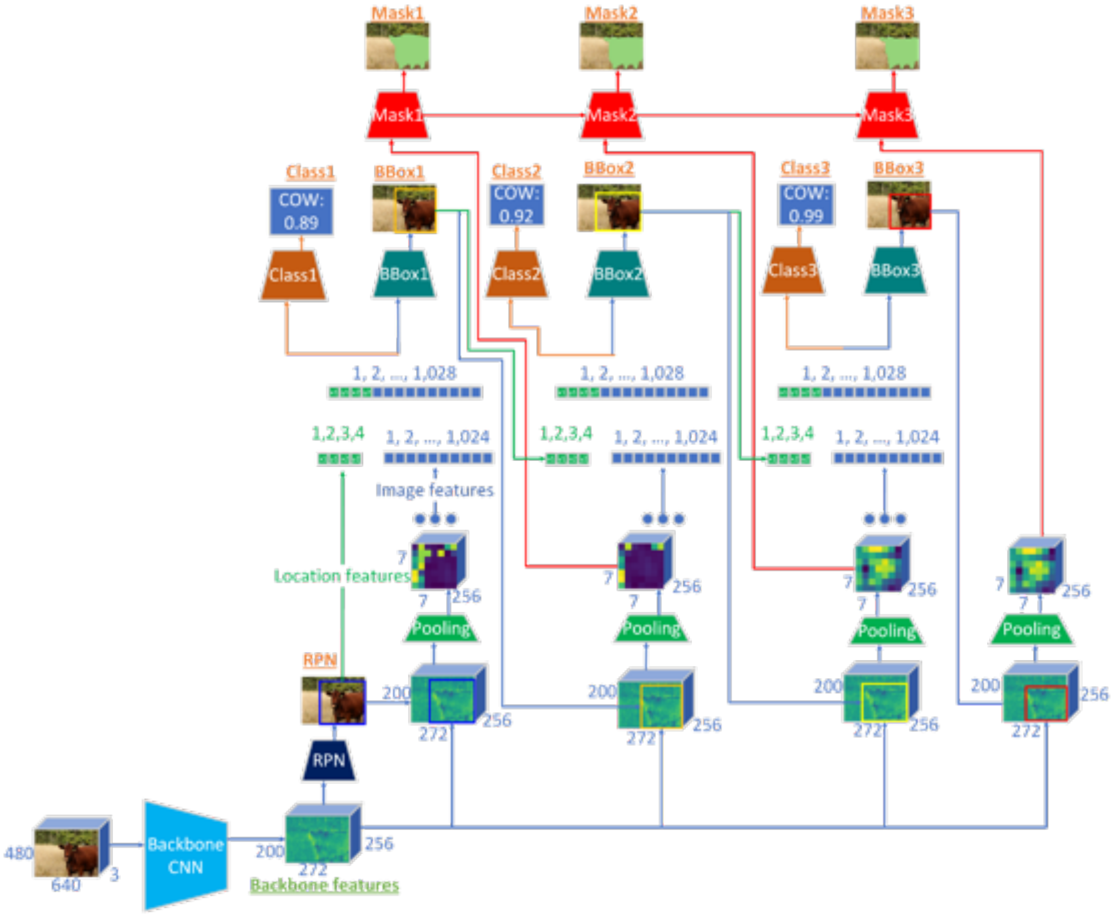


Figure 4.6 The proposed location-sensitive HTC network that integrates location features at the FC layer.

As shown in Figure 4.7, the second method is to add the new network head, named the supplementary classification head (SCLASS), to the HTC network. the location features (from the RPN, BBox1, and BBox2) and the feature from the FC layer (from the pooling layer) are combined as the input of the SCLASS branch. Because the new supplementary classification branch has been incorporated into the network architecture, defining a new supplementary loss for this branch is necessary. The HTC is a multi-stage approach; that is, at each stage t, for all sampled regions of interest (RoIs), the box branches estimate the bounding box regression offset, the classification branches estimate the classification score, and the mask branches estimate the pixel-wise masks for positive RoIs. By adding the new supplementary classification branches, the overall loss function, taking the form of multi-task learning, is defined as follows:

$$L = \sum_{t=1}^{T} \alpha_t \left(L_{bbox}^{t} + L_{cls}^{t} + L_{scls}^{t} + L_{mask}^{t}\right) \tag{4.1}$$

Where L_{bbox}^{t} is the loss of the bounding box predictions and L_{cls}^{t} is the loss of the classification at stage t, which is the same as that of the Cascade R-CNN (Cai and Vasconcelos 2019). L_{scls}^{t} is the proposed loss of the classification on the new supplementary classification branch at stage t. L_{mask}^{t} is the loss of mask prediction at stage t, which employs binary cross entropy (BCE), as in the Mask R-CNN (He et al. 2017). The coefficient α_t is used to balance the supplements of several stages and tasks. The hyper-parameter settings have been adopted from the HTC (Chen, Ouyang, et al. 2019) with $\alpha = [1,0.5,0.25]$ and $T = 3$ by default.

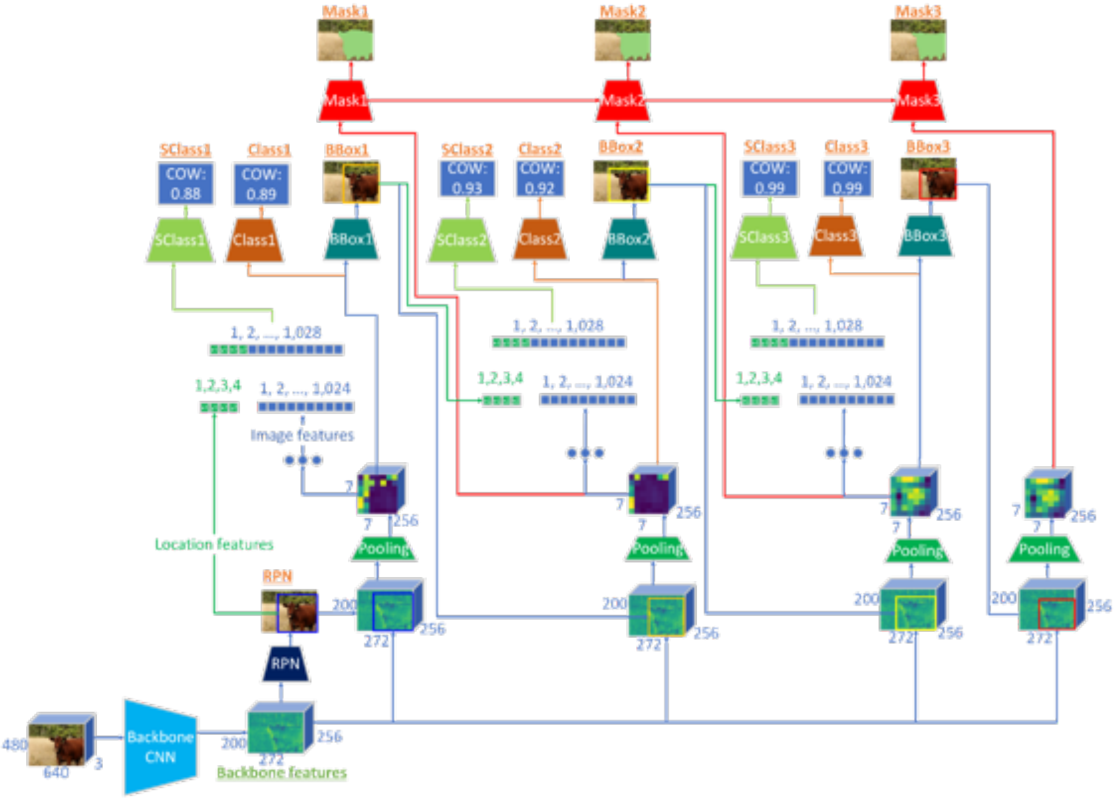


Figure 4.7 The proposed location-sensitive HTC network, which has a new 'supplementary classification branch' (SCLASS) taking the location feature and the feature from the FC Layer as the inputs.

The bounding box regression loss for each RoI in (4.2) is defined over a tuple of the bounding box ground truth $v = (v_x, v_y, v_w, v_h)$ and a predicted tuple $b = (b_x, b_y, b_w, b_h)$ for each class, where x, y, w, and h are the position (x, y) and size (w, h) of the RoI. L_1 is the Manhattan distance defined in (4.3) as in the Fast R-CNN (Girshick 2015).

$$L_{bbox}(b,\ v) = \sum_{i \in \{x,\ y,\ w,\ h\}} smooth_{L_1}(b_i - v_i) \tag{4.2}$$

$$smooth_{L_1}(x) = \begin{cases} 0.5x^2 & if\ |x| =< 1 \\ |x| - 0.5 & otherwise \end{cases} \tag{4.3}$$

The classification and supplementary classification loss are defined by cross entropy (CE), where p is the predicted probability computed by a softmax at the FC layer, while u is the ground truth for each class. CE loss measures the performance of a classification model whose output is a probability value between 0 and 1. CE loss increases as the predicted probability diverges from the actual label. A perfect model would have a CE loss of 0, where CE is defined as follows:

$$CE(p,\ u) = -\sum_{i}^{K} u_i log p_i \tag{4.4}$$

Where K is the number of classes in the model. In BCE loss, where the number of classes K equals 2, CE can be calculated as:

$$BCE(p,\ u) = -(ulogp + (1-u)log(1-p)) \tag{4.5}$$

The mask branch has a Km^2 dimensional output for each RoI, which encodes the K binary masks of resolution $m \times m$, one for each of the K classes. He et al. (He et al. 2017) applied a per-pixel sigmoid and defined L_{mask} as the average BCE loss:

$$L_{mask}(m_{pred}, m_{gt}) = BCE(m_{pred}, m_{gt}) \tag{4.6}$$

For a RoI, m_{pred} is a predicted mask and m_{gt} is a ground-truth class u.

2) Post-processing

With the extended HTC models, the working bunch can be detected in most cases, but occasionally, bunches other than the working bunch may be detected. Figure 4.8 shows an example in which three bounding boxes have been obtained and two of them are overlapping with each other and are actually detected from the same bunch. To exclude these results further, the post-filtering the bounding boxes is proposed by using the probability of estimation and the size of the bounding box obtained from the proposed location-sensitive HTC model. The post-processing procedure is depicted in Figure 4.10. First, bounding boxes with a low probability of estimation are removed. The overlapping bounding boxes are sorted by the size of bounding box. Afterward, intersection over union (IoU) is used to remove the bounding box with smaller overlapping. Finally, the bounding box nearest the image's center is selected. Figure 4.9 shows the result by applying the proposed post-processing technique to the results shown in Figure 4.8.

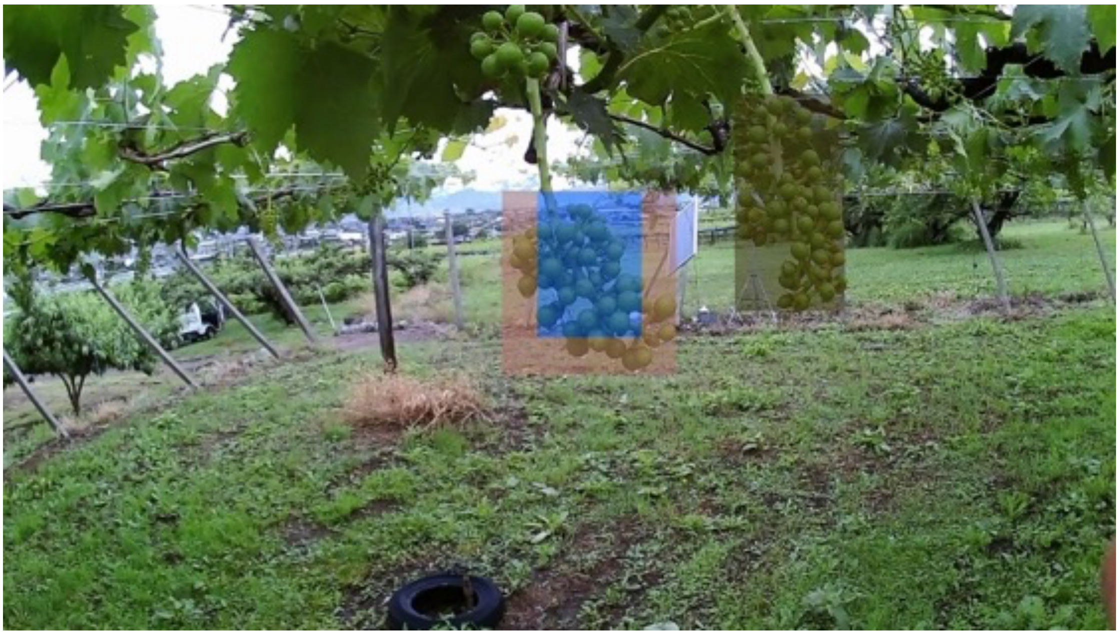


Figure 4.8 An example in which the proposed location-sensitive HTC still detected other bunches in addition to the working bunch and output multiple overlapping bounding boxes for the same bunch.

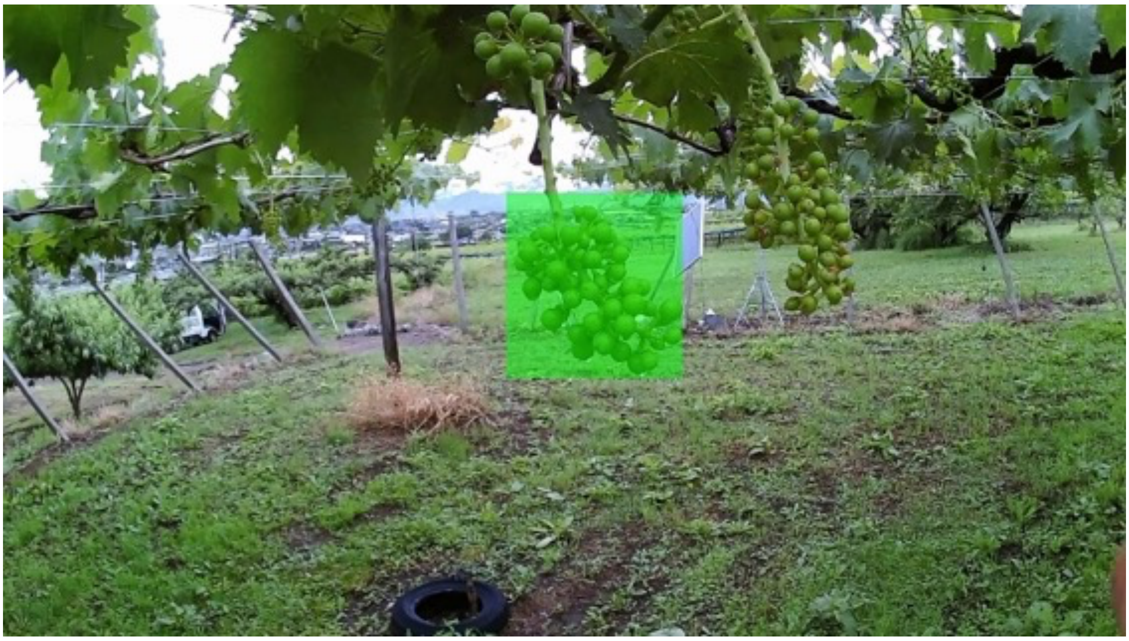


Figure 4.9 An example in which only the working bunch is detected by applying the proposed location-sensitive HTC and post-processing.

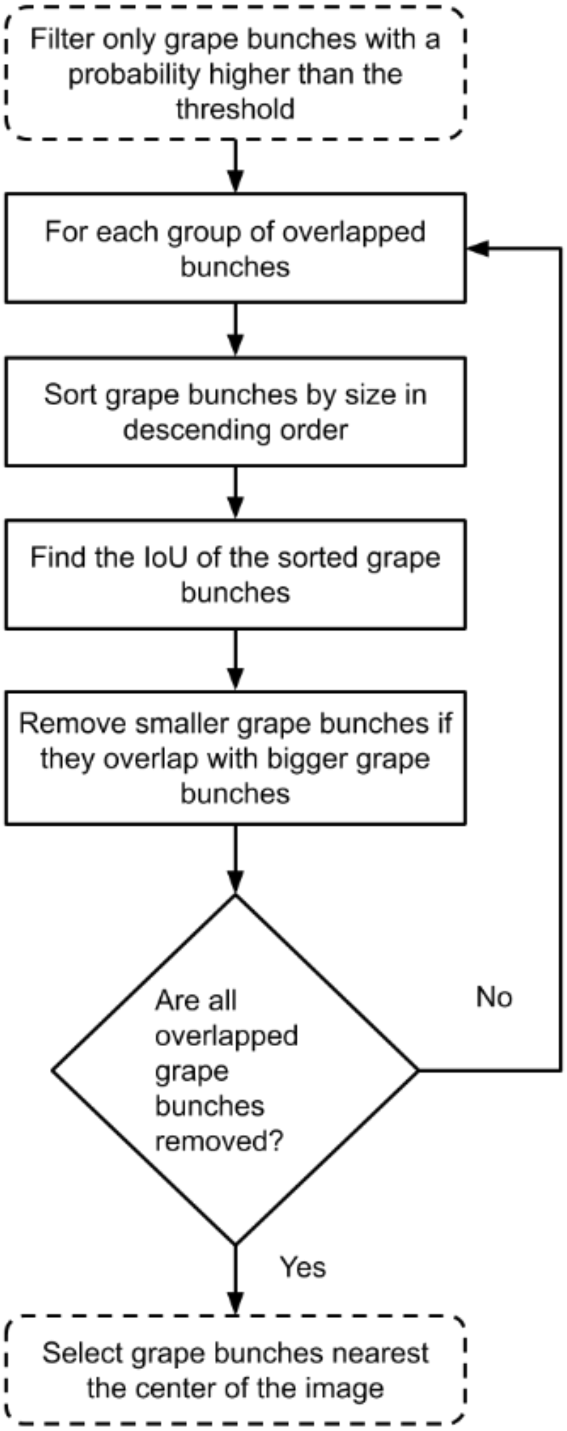

Filter only grape bunches with a probability higher than the threshold
For each group of overlapped bunches
Sort grape bunches by size in descending order
Find the IoU of the sorted grape bunches
Remove smaller grape bunches if they overlap with bigger grape bunches
Are all overlapped grape bunches removed?
No
Yes
Select grape bunches nearest the center of the image

Figure 4.10 Flowchart of post-processing filtering for the further elimination of berries not included in the working bunch.

4.1.3 Automatic berry number prediction using a single image

Predicting the number of berries in a whole 3D bunch using a single image is highly challenging, as the number of visible berries can differ significantly depending on the view directions. Based on careful observation, it was found that the relationship between the number of berries in the whole 3D bunch and the number of berries visible in the captured images could be affected by multiple factors. The following five features are empirically used to computed from the 2D images as the inputs to the regression model for predicting the number of berries in a 3D bunch.

1. Number of berries
2. Diameters of berries
3. Circularity of berries
4. Density of berries
5. Homogeneity of berry distribution

The berry number feature $Feat_{nberries}$ is set to the number of detected berries (N_{cd}) in the single image as follows.

$$Feat_{nberries} = N_{cd} \tag{4.7}$$

The diameter feature $Feat_{diameter}$ can be computed as the average diameter of all berries detected in the 2D image with (4.8).

$$Feat_{diameter} = \frac{\sum_{i=1}^{N_{cd}} B\,erry_{diameter}(i)}{\sum_{i=1}^{N_{cd}} B\,erry_{area}(i)} \tag{4.8}$$

Here, $Berry_{diameter}$ is the diameter of individual berries. The distance between the camera and the grape in each image is not fixed, and the absolute diameter value changes with the distance. The feature scale invariant is made by normalizing the diameter with the berry area denoted as $Berry_{area}$.

The circularity feature ($Feat_{circularity}$) indicates how many partially occluded berries are among the detected berries. Generally, the occluded berries have a non-circular shape. The circularity of a berry (Friel 2000) can be computed with (4.9) from the berry's area $Berry_{area}$ and perimeter $Berry_{perimeter}$.

$$Berry_{circularity} = \frac{4\pi Berry_{area}}{Berry_{perimeter}^{2}} \tag{4.9}$$

The circularity feature $Feat_{circularity}$ is then computed as the proportion of the number of occluded berries ($Berry_{circularity}$ less than the threshold) over the total number of detected berries (N_{cd}) with (4.10).

$$Feat_{circularity} = \frac{\sum_{i=1}^{N_{cd}} \begin{cases} 1 & if\ Berry_{circularity}(i) \leq 0.7 \\ 0 & otherwise \end{cases}}{N_{cd}} \tag{4.10}$$

The detected non-occluded berries should have a round shape with a circularity value close to 1.0. The number of partially occluded berries can be estimated by counting the number of berries whose circularity is smaller than a given threshold, which is empirically set to 0.7 in my experiment.

The density feature $Feat_{density}$ is computed with (4.11) as the proportion of the berries' area $Berries_{area}$, which is the summation of the areas of all detected berries, over the bunch area $Bunch_{area}$, as detected by the location-sensitive HTC model trained with bunch images.

$$Feat_{density} = \frac{\sum_{i=1}^{N_{cd}} Berries_{area}(i)}{Bunch_{area}} \tag{4.11}$$

The larger the $Feat_{density}$, the more berries are likely to be occluded in the current image. Therefore, $Feat_{density}$ also gives a reasonable indication of the number of occluded berries.

The homogeneity feature $Feat_{homo}$ indicates how uniform the distribution of the detected berries is in the image. The distribution of berries was found that it can be non-uniform in the images, which means severe occlusion can occur locally even though the overall density is low. Therefore, together with the density, the homogeneity feature also plays an important role in accurately predicting the number of berries in a 3D bunch. To compute the homogeneity feature, a method based on Gaussian smoothing (Forsyth and Ponce 2002) is employed. The mask image was considered with the berry area set to white (255) and the other set to black (0). If the berries are uniformly distributed, that is, if each berry is surrounded by the background and no berries are overlapping or close to each other, then after repeatedly applying Gaussian smoothing, the berry area will be gradually blended with the background. Thus, an image of uniformly gray pixel values can be obtained. On the contrary, if the berries are not uniformly distributed, then the image should consist of a large background area and an area with dense overlapping berries. Then, some background areas and berry areas would remain unchanged, even after repeatedly applying Gaussian smoothing. Therefore, the difference between the images at different stages of repeated Gaussian smoothing should give a good measure of the homogeneity. In current implementation, the difference between the images is computed after applying Gaussian smoothing once and the image after applying Gaussian smoothing 11 times and then adding the difference of all pixels together to get the $Feat_{homo}$.

To predict the number of berries in a 3D bunch using the above five features, the six representative regression models were experimented: kernel ridge regression (KRR) (Murphy 2012), support vector regression (SVR) (Bishop 2006), random forest regression (RFR) (Breiman 2001), gradient boosting (GB) (Friedman 2001), stochastic gradient descent (SGD) (Bottou 2010), and artificial neural network (ANN) (Géron 2019).

4.2 Experiment results and discussion

4.2.1 Dataset and implementation details

Two farmers were asked for help by installing cameras on their heads to capture the working scene during the grape-thinning task. Then, 2,701 berries were manually labeled in 60 images from 10 different bunches. Each image has a resolution of 1,920 × 1,080 pixels, and each was rescaled to have a minimum size of 800 pixels and a maximum size of 1,333 pixels. In the current implementation, the models were trained and evaluated on a single Titan RTX GPU for more than 10 hours. Table 4.1 shows the hyper-parameter applied to the instance segmentation models.

Table 4.1 Hyper-parameters applied to the instance segmentation models.

Property	**Value**
Optimizer	Stochastic gradient descent (Bottou, Curtis, and Nocedal 2018)
Learning rate	0.00125 (Goyal et al. 2017)
Decay	0.0001
Momentum	0.9
Batch size	1
Epoch	500

4.2.2 Evaluation metrics

Because the aim of the first step is to detect grape berries accurately, the accuracy is measured by computing the IoU between the mask of the detected grape berry and that of the ground truth grape berry. Similar to Zabawa et al. (Zabawa et al. 2019), two quantitative measures, Correctly Detect (*CD*) and Miss-Classification (*MC*) are used, which are computed with (4.12) and (4.13), respectively.

$$CD = \left(\frac{N_{cd}}{N_{gt}}\right) \times 100 \tag{4.12}$$

$$MC = \left(\frac{N_{id}}{N_{ad}}\right) \times 100 \tag{4.13}$$

Here, N_{cd} , N_{gt}, N_{id}, and N_{ad} are the number of correctly detected berries, manually labeled berries, falsely detected berries, and all detected berries, respectively. In other words, CD is the percentage of correctly detected grape berries over the manually labeled grape berries, and MC is the percentage of falsely detected berries over all detected berries. The IoU threshold is used to determine whether the detected object is correctly or falsely detected. In the experiment, the threshold is set to 0.5, which follows the approaches from the Pascal VOC Challenges (Mottaghi et al. 2014). The annotation application used in this study is the COCO Annotator application (Brooks 2019).

4.2.3 Evaluation of data augmentation technique

The proposed data augmentation technique synthesizes images by removing berries with a circularity below a given threshold. Table 4.2 shows the number of synthetic images with different circularity thresholds. In terms of the limited diversity of background images and storage resources, we decided to use the circularity threshold 0.6, which can synthesize 956 images from 60 annotated images, for the experiment.

Table 4.2 The number of synthesized images and berries with different circularity thresholds.

Circularity threshold	**Number of synthesized images**	**Number of berries**
0.5	164	7,325
0.6	**956**	**41,704**
0.7	19,952	970,780

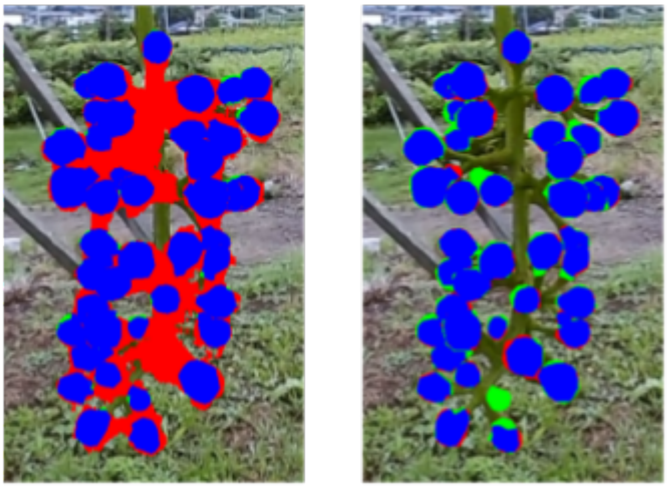
I compare the results with/without using the proposed augmentation method, and six-fold cross-validation was applied. As shown in Table 4.3, the 60 annotated images are divided into six folds, each of which contains 10 images. The number of images synthesized with the proposed techniques from each fold is also shown in the table.

During cross-validation, 50 original images of five folds and their corresponding synthesized images are used for training, and the 10 original images are used for validation.

Table 4.3 The number of synthesized images from each fold.

Fold	**Number of original images**	**Number of synthesized images**
1	10	185
2	10	130
3	10	103
4	10	114
5	10	332
6	10	92

Table 4.4 shows the results of the validation, which is the average of the validation results of all six folds. The HTC (Chen, Ouyang, et al. 2019) model using HRNet (Sun et al. 2019) as the backbone model was used. It can be observed that using the proposed augmentation affords a notable performance over not using augmentation (MC decrease of 51.38%). Although the CD decreases by 2.17% compared to not using augmentation, the decrease in MC is a huge improvement. Figure 4.11 shows the detected results without augmentation (left) and with the proposed augmentation (right). The red mask is the detected mask that did not overlap with the ground truth mask (false positive), blue is the detected mask that did overlap with the ground truth mask (true positive), and the green mask is the ground truth that did not overlap with the detected mask (false negative). It is obvious that the proposed method can reduce a large number of false-positive results by trading a small loss of true-positive results. The reason the proposed method can generate images for training an effective model is that it does not destroy the context information in the image. The synthesized image simulates the real pictures taken during the thinning process. The proposed method provides a simple yet efficient approach to prevent model overfitting.



a) HTC without augmentation   b) HTC with A-Berry (Proposed)

Figure 4.11 Comparison of the detected result between HTC without augmentation (a) and HTC with the proposed augmentation (b). The red mask is the detected mask that did not overlap with the ground truth mask, blue is the detected mask that did overlap with the ground truth mask, and the green mask is the ground truth mask that did not overlap with the detected mask.

Table 4.4 Comparison of training the HTC (Chen, Ouyang, et al. 2019) model using HRNet (Sun et al. 2019) as the backbone model, with and without augmentation.

Methods	CD (%)	MC (%)
Without augmentation	**98.72**	54.17
With augmentation	96.55	**2.79**

4.2.4 Evaluation of location-sensitive HTC model

This section presents the results of the proposed location-sensitive models. Figure 1.6 shows the experiment results of the HTC (Chen, Ouyang, et al. 2019) and the proposed location-sensitive models using HRNet (Sun et al. 2019) as the backbone model. The results show that combining the explicit location feature with the fully connected feature (Figure 4.6) improves model accuracy. Integrating the explicit location information with the fully connected features at the new supplementary classification branch and training the model using new supplementary classification loss (Figure 4.7) improves model accuracy more than simply integrating the location features in the original branch of the HTC (Figure 4.6).

Furthermore, the 956 synthesized images used for the six-fold cross-validation, as shown in Table 4.2 and Table 4.3, are used to evaluate the average number of berries detected from the non-working bunch (Avg_{NWB}) using the metric given in (4.14).

$$Avg_{NWB} = \frac{1}{Fold}\sum_{i=1}^{Fold} Berries_{NWB}(i) \tag{4.14}$$

Here, $Fold$ is the number of folds, which is 6 in this study, and $Berries_{NWB}(i)$ is the number of non-working bunch berries for each fold i. the proposed SCLASS (Figure 4.7) is compared with the conventional HTC (Chen, Ouyang, et al. 2019) using (4.14). The proposed models can reduce the average number of berries detected from the non-working bunch from 5.33 to 0.33, which can prevent the counting of berries that do not belong to the working bunch. The reason the proposed models can reduce the number of unexpectedly detected berries is that the location features help the model partially learn the location of the object. The explicit location information is what the conventional model uses to specify the feature map location from the pooling layer. However, the explicit location has never been used as an input feature for object classification or prediction in the conventional model. The proposed models make use of a feature that is already available without requiring additional data annotation costs. Especially, the experiment results shown in Table 4.5 and Table 4.6 also demonstrate that the proposed models do not consume much more time than the original model. Figure 4.12 shows the results of the HTC (Chen, Ouyang, et al. 2019) and the proposed methods. The red circle is the berry that the proposed methods could detect but that the HTC (Chen, Ouyang, et al. 2019) could not.

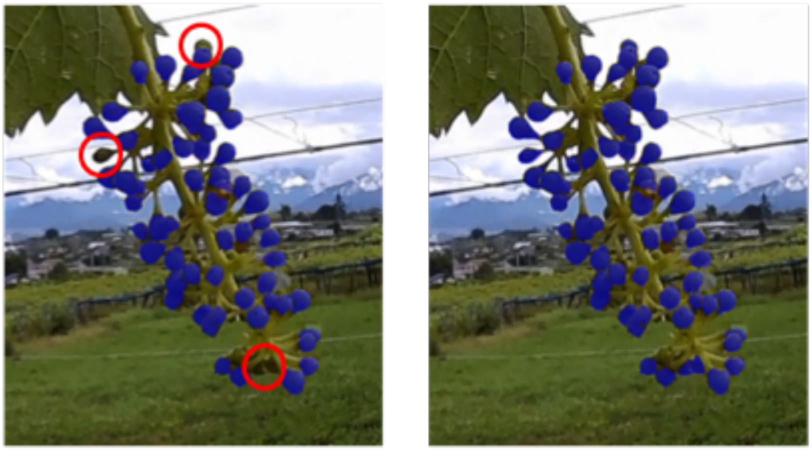
Table 4.5 Comparison of the average processing times of 60 images for different stages shown in Figure 4.1 between the HTC (Chen, Ouyang, et al. 2019) model and the proposed models, using HRNet (Sun et al. 2019) as the backbone model.

Stages	**Processing time**		
	HTC (Chen, Ouyang, et al. 2019) **(s)**	**Proposed FC (Figure 4.6) (s)**	**Proposed SCLASS (Figure 4.7) (s)**
Detect working bunch	0.493	0.501	0.517
Extract features	0.363	0.368	0.371
Predict berry number in 3D bunch	0.00847	0.00867	0.01025

Table 4.6 Comparison of the number of trainable parameters and the number of floating-point operations per second (FLOPs) between the HTC (Chen, Ouyang, et al. 2019) model and the proposed models, using HRNet (Sun et al. 2019) as the backbone model. The size of the input image is 1,280 x 800 pixels.

Model	**Parameters (M)**	**FLOPs (G)**
HTC (Chen, Ouyang, et al. 2019)	82.68	516.14
Proposed FC (Figure 4.6)	82.68	516.14
Proposed SCLASS (Figure 4.7)	83.13	516.58

There exists a trade-off between the accuracy and the computational complexity when selecting a DNN model. The state-of-the-art instance segmentation model has made an extension to take advantage of obtaining accurate mask information about individual berries to compute the features required for predicting berry numbers. As shown in Table 4.5, it takes about 0.9 seconds on average to process one frame on a high-end graphics processing unit (Titan RTX GPU), which makes the method more suitable to be implemented as a remote application. However, during the experiment, it was found that the farmers did not actually need to confirm the number of berries in every frame. Therefore, it is possible to implement a user-friendly application even on an embedded AI computing device or mobile device by only computing and visualizing the berry numbers whenever any berries are removed.



a) HTC **b) Proposed**

Figure 4.12 Comparison of the detected berry between HTC (Chen, Ouyang, et al. 2019) and the proposed method. The blue mask is the detected berry mask; the red circle is the berry that the proposed method can detect, but that HTC (Chen, Ouyang, et al. 2019) cannot detect.

4.2.5 Evaluation of post-processing technique

This section shows the evaluation results of post-processing to exclude the berries that do not belong to the working bunch. Figure 4.13 shows an example of the berries (in blue color) detected by the location-sensitive HTC model but that were identified as not belonging to the working bunch during post-processing. Figure 4.14 shows that the grape berries not belonging to the working bunch are discarded and only the berries in the working bunch are counted. The 2,535 different berry images are used to evaluate the efficiency of the proposed post-processing method using the metric given in (4.15).

$$Ab(B_{pd}) = \begin{cases} 1 & : B_{pd} = B_{gt} \text{ } and \text{ } COUNT(B_{pd}) = 1 \\ 0 & : otherwise \end{cases} \tag{4.15}$$

Here, Ab is the number of accurately detected working bunch, B_{pd} is the predicted working bunch, and B_{gt} is the ground truth working bunch. The result is manually checked for each image and compute the proposed method's average accuracy using (4.16).

$$A(x)_{avg} = \left(\frac{\sum_{i=1}^{N} x_i}{N}\right) \times 100 \tag{4.16}$$

Here, $A(x)_{avg}$ is the average accuracy of x, x_i is the accuracy (Ab) for image i, and N is the number of images in this experiment. The proposed method was found that it could select the working bunch with an accuracy of 100% for all images.

Experimental results show that combining the proposed location-sensitive models with the post-processing method can well meet the main purpose of the research, that is, the end-to-end automatic counting of berries in a working bunch without counting the berries from other bunches. The proposed method succeeds in tackling this problem.

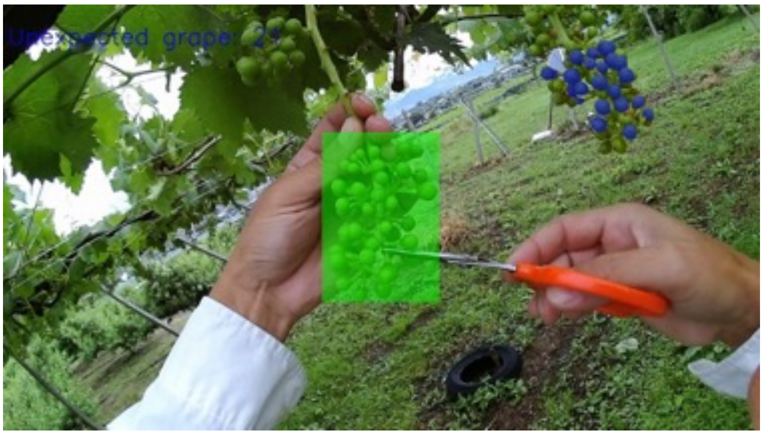


Figure 4.13 The berries (blue) that have been discarded because they do not belong to the working bunch (green).

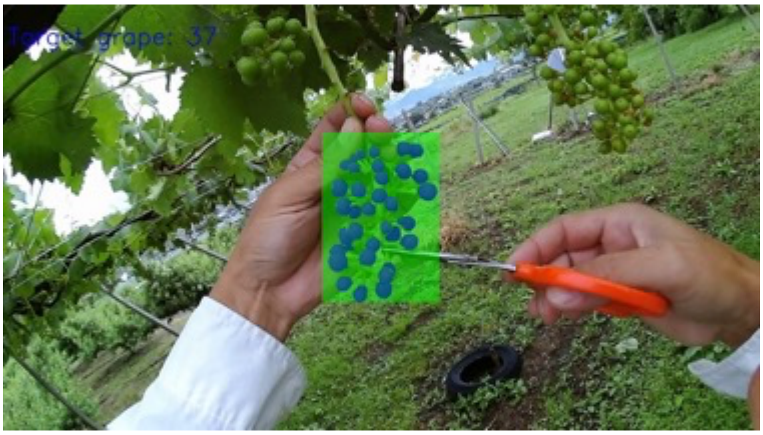


Figure 4.14 The final result (blue) after post-processing, including only the berries in the working bunch.

4.2.6 Evaluation of automatic berry number prediction

To evaluate the regression models using the proposed features described in Section 4.1.3, the dataset was collected by taking images while farmers thin grape berries from start to end. The farmers were asked to rotate the bunch during the process to capture as many images from different perspectives as possible. The actual numbers of berries in the 3D bunch (3D counting) were manually counted as the ground truth. Input features for regression models were extracted using the method proposed in Section 4.1.2. Table 4.7 shows number of training and test data with three datasets took for two years. Figure 4.15 depicts the proposed method to collect 3D berry number counting datasets. The light blue circles represent the events done by the farmer. The red lines represent the ignored frames. The green lines are the frames used. The example shows when starting berry thinning task, the ground truth (real number of berries in the bunch) is 45 berries. After finish berry thinning, the ground truth is 43 berries. The beginning and almost ending frames were discarded to prevent the bad example data from farmers' unexpected movement.

Totally 3 datasets are created. The first two datasets, namely AI_berry3dcounting_2020 and AI_berry3dcounting_2021, were created using the detected result from the AI model. That is, the bounding box and mask of berries on the 2D images were obtained using the instance segmentation model (AI). Since typically detected berries counting should be more than 30 berries per bunch, the frame containing the detected berries counting below 30 berries per bunch was discarded from the dataset. The third data set, Human_berry3dcounting_2021, was created by manually annotating the berries and the bunch every frame by a human and has a high quality.

Table 4.7 Number of training and test data with various dataset.

Dataset	Training data	Test data
AI_berry3dcounting_2020	13,285	3,322
AI_berry3dcounting_2021	92,578	23,147
Human_berry3dcounting_2021	1,748	438

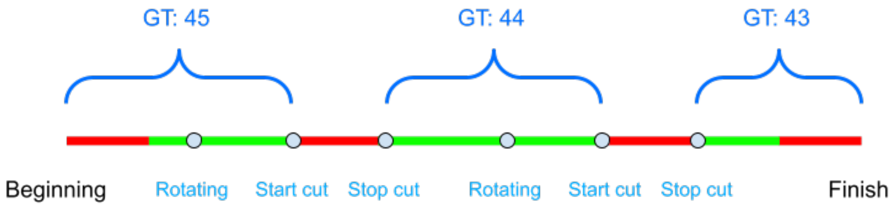


Figure 4.15 The proposed method to collect 3D counting dataset. The light blue circles represent the events done by the farmer. The red lines represent the ignored frames. The green lines are the frames used. The example shows when starting berry thinning task, the ground truth (real number of berries in the bunch) is 45 berries.

The evaluation metric in this experiment is the mean absolute error (MAE) (Géron 2019), shown below.

$$MAE(X, h) = \frac{1}{m} \sum_{i=1}^{m} \left| h(x^{(i)}) - y^{(i)} \right| \tag{4.17}$$

Where m is the number of instances in the dataset, $x^{(i)}$ is is a vector of all the feature values (excluding the label) of the i^{th} instance in the dataset, $y^{(i)}$ is its label (the desired output value for that instance), X is a matrix containing all the feature values (excluding labels) of all instances in the dataset, h is the regression model (also called a hypothesis), and $MAE(X, h)$ is the cost function measured on the set of example X using hypothesis h. In this study, $y^{(i)}$ is the ground truth of the berry numbers in the 3D bunch and $x^{(i)}$ is the five features computed from the i^{th} 2D images. For h, six different models are tested.

The results by six regression models are shown in Table 4.8. The results show that using RFR (Breiman 2001) can archive the most accurate estimation for all three datasets. The reason RFR obtains the best accuracy can be explained by the fact that a random forest is good at reducing the variance in the forest estimator by combining diverse trees, which complies with the large variance in features computed from 2D images. For the same bunch, the number of berries visible on a 2D image can vary by more than 10 berries for the images captured from different perspectives. Such a fact makes the berry number-prediction task highly difficult.

Table 4.8 MAE of berry number prediction for different regression models using the proposed features computed from 2D images.

Regression model	**AI_berry3dcounting_2020**	**Human_berry3dcounting_2021**	**AI_berry3dcounting_2021**
Kernel ridge regression (KRR) (Murphy 2012)	6.12	3.91	Out of memory
Support vector regression (SVR) (Bishop 2006)	4.64	2.92	3.82
Random forest regression (RFR) (Breiman 2001)	**3.79**	**2.81**	**3.65**
Gradient boosting (GB) (Friedman 2001)	4.57	2.95	3.78
Stochastic gradient descent (SGD) (Bottou 2010)	5.48	3.00	3.98
Artificial neural network (ANN) (Géron 2019) (Géron 2019)	5.03	2.82	3.93

In a practical scenario, when farmers are thinning grapes, the number of berries in a bunch begins at a larger number and reaches a smaller number (target number). Therefore, in the experiment, The MAE was computed as the function of 3D counting to validate the effect of a prediction model during the thinning process. Figure 4.16 ~ Figure 4.18 show such result for the three datasets. During the real thinning process, when the number of berries in the bunch is much larger than the target number, the estimation accuracy is relatively unimportant. However, when the number of berries in the bunch approaches the target number, the estimation accuracy becomes critical for avoiding over-thinning. In Figure 4.16, for AI_berry3dcounting_2020 dataset, MAE decreases when 3D counting decreases. MAE starts from 6.44 for the 3D counting range of 76–91 and decreases to 2.91 for the 3D counting range of 42–58. In Figure 4.17, for AI_berry3dcounting_2021 dataset, MAE decreases when 3D counting decreases too. MAE starts from 7.71 for the 3D counting range of 58–69 and decreases to 3.25 for the 3D counting range of 34–45. In Figure 4.18, for Human_berry3dcounting_2021 dataset, MAE starts from 6.10 for the 3D counting range of 59–68 and decreases to 2.48 for the 3D counting range of 35–46.

Because the target number of berries in a bunch for major table grape varieties is less than 40, as shown in Table 1.1, this experiment result demonstrates that the proposed method can fit real practical scenario usage well. The farmers involved in the experiment are highly satisfied with the performance of the proposed technique.

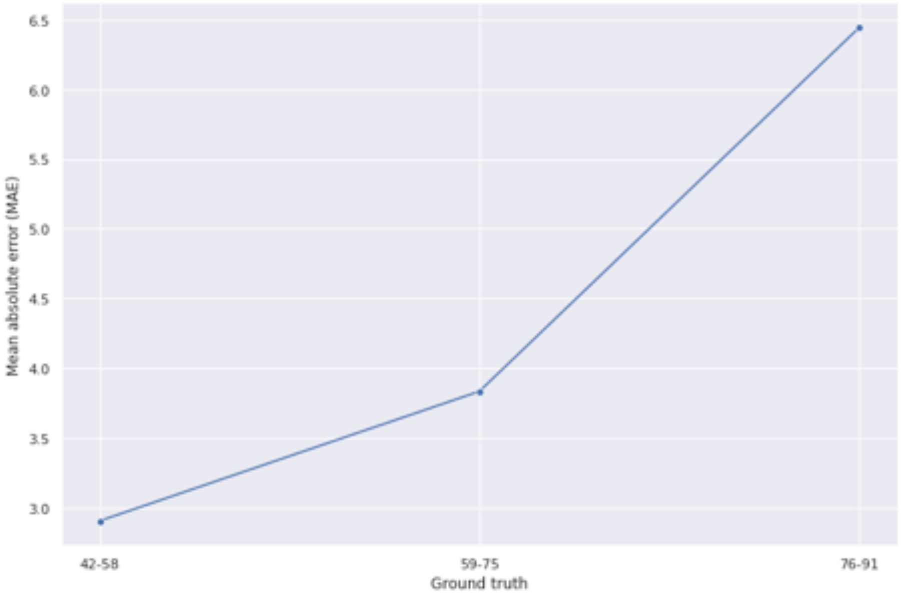


Figure 4.16 MAE as a function of ground truth berry number for AI_berry3dcounting_2020 dataset using Random forest regression model.

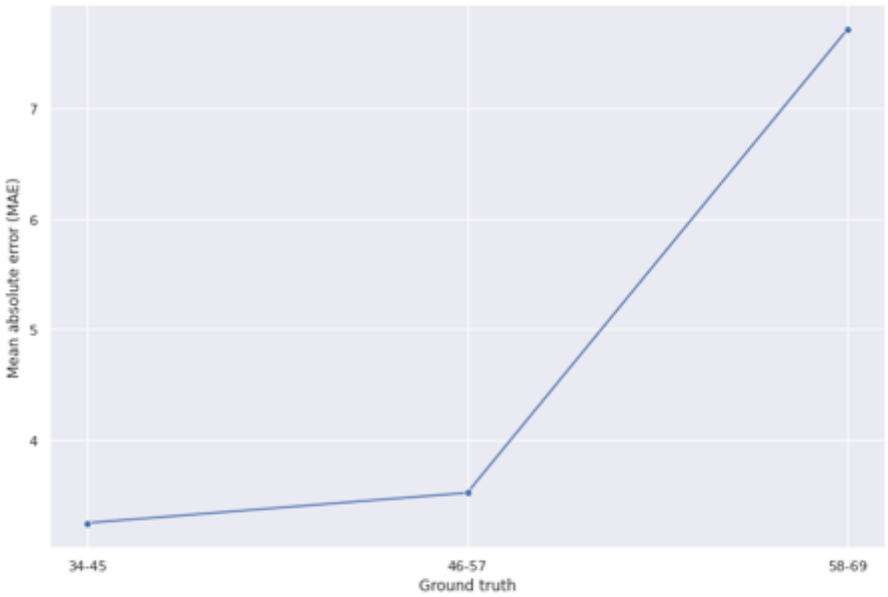


Figure 4.17 MAE as a function of ground truth berry number for AI_berry3dcounting_2021 dataset using Random forest regression model.

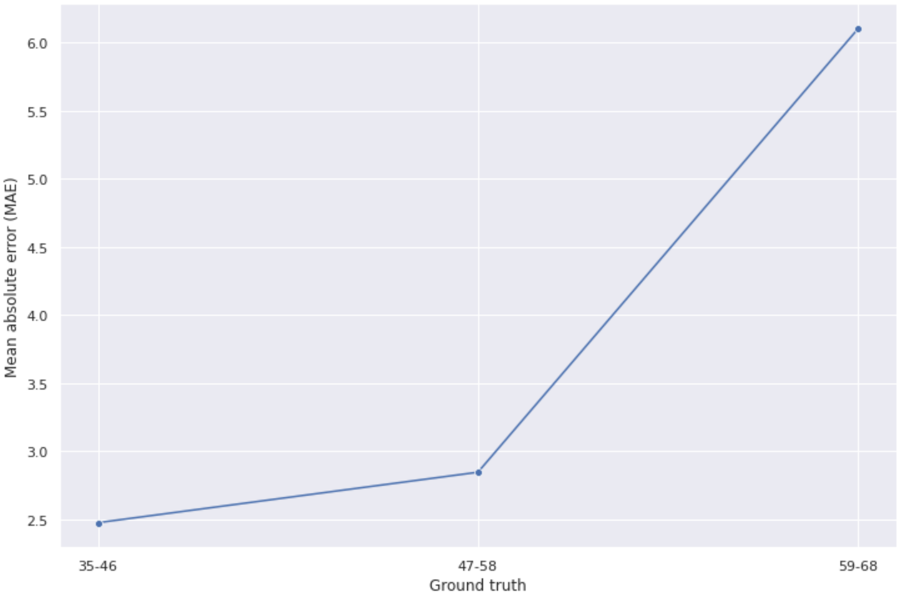


Figure 4.18 MAE as a function of ground truth berry number for Human_berry3dcounting_2021 dataset using Random forest regression model.

4.3 Summary

The proposed technology is for building a practical application for the real grapevine farm environment. The novel end-to-end berry number prediction technology enables farmers to perform berry thinning, which is a crucial task affecting the market value of table grapes, efficiently. By integrating the location feature into the state-of-the-art instance segmentation DNN model, focusing the berry detection on the working bunch only has succeeded. The proposed location-sensitive HTC model can also be used for other object detection problems that require detecting a particular object from an image consisting of multiple objects of similar features. Berry number prediction using the originally designed features can also be applied to the image-based counting of other kinds of fruits or vegetables.

AUTOMATIC IDENTIFYING THE BERRIES TO BE REMOVED IN TABLE GRAPE THINNING USING A DEEP NEURAL NETWORK WITH ATTENTION FORCING TECHNIQUE

Figure 5.1 shows the pipeline of the proposed technique for identifying the berry to be removed during the berry thinning. First, an instance segmentation model is executed to detect the working bunch and all berries in the bunch on the images sent from OSTHMD. Then a post-processing step is followed to pass the detection results of the frames with sufficient changes to the previous frame only to the downstream. This detection post processing is for avoiding updating the prediction result every frame even though the farmer does not change his/her view of the working bunch. Practically, the farmer needs time to recognize where is the berry should be removed in two or three seconds. If the prediction result is updated too fast, it can get the farmer confused and tired. Moreover, detection post-processing can avoid computation redundancy. Using the results from the detection post processing, AF image generation was applied to prepare input image for removing berry identification model. All the AF images are fed to removing berry identification model directly without being saved to the storage. This is because DNN model can process numerous images simultaneously by organizing them into batches, supported by modern neural network frameworks. Finally, the system gathers the removal probabilities for each berry, then generates the visualization image and sends it back to OSTHMD.

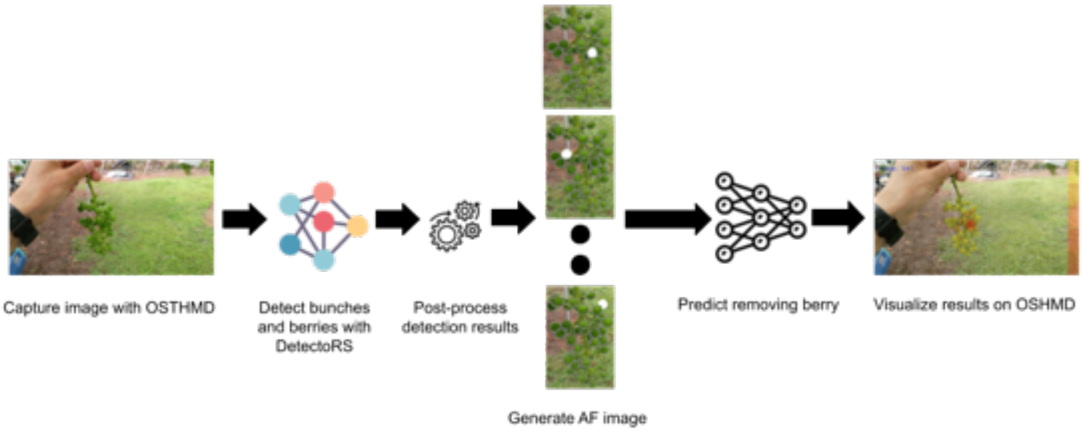

Capture image with OSTHMD
Detect bunches and berries with DetectoRS
Post-process detection results
Generate AF image
Predict removing berry
Visualize results on OSHMD

Figure 5.1 Proposed removing berry identification pipeline.

5.1 Methodology

5.1.1 The detection post-processing improving user experience

The detection post-processing is depicted in Figure 5.2. The bounding box of the detected bunch determines the region to crop the bunch area from the original image. Then the similarity score between this cropped image and the one of previous image are computed using the structural similarity index measure (SSIM) (Wang et al. 2004). If the similarity score exceeds the given threshold, the previous result is sent back to OSTHMD. Otherwise, generation of AF images is performed. The generated AF images are sent to the removing berry identification model and new identification result will be sent to OSTHMD. The similarity threshold was set to 0.34, which is the average SSIM score among 14,782 video frames from 54 different bunches throughout the berry thinning task was captured with OSTHMD.

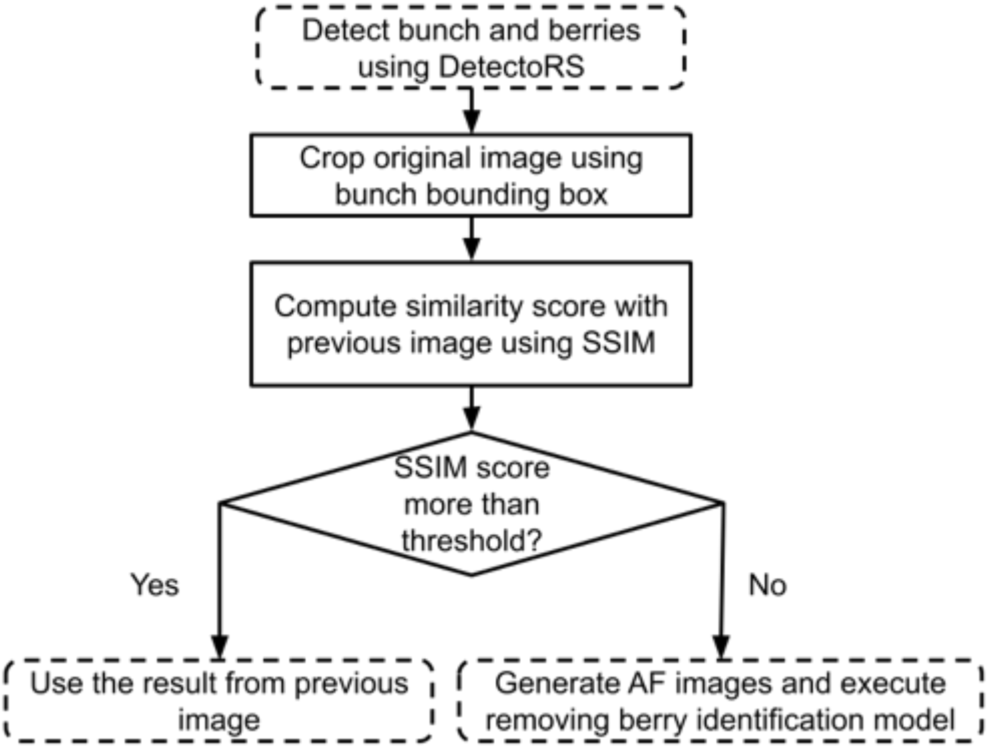

Detect bunch and berries using DetectoRS
Crop original image using bunch bounding box
Compute similarity score with previous image using SSIM
SSIM score more than threshold?
Yes
No
Use the result from previous image
Generate AF images and execute removing berry identification model

Figure 5.2 Proposed detection post-processing for improving user experience.

5.1.2 Attention Forcing (AF) image generation

It is challenging to enable the DNN to learn the characteristic of the berries that should be removed or not. Traditional methods based on hand crafted features, such as Color thresholding (Arad et al. 2020; Font et al. 2014; Ji et al. 2012; Xiong et al. 2020), Circular Hough Transformation (CHT), or Blob Analysis (BA) (Silwal et al. 2017), are not suitable for the purpose because there are no difference in those features between candidate berry and other berries. Moreover, whether a berry should be removed or not depends on not only its own features but also other semantic features such as the berry distribution in neighborhood or the whole bunch. Therefore, AF, an image preprocessing technique was proposed to encode those semantic features. With AF technique, the fact that a berry is the target to be removed is represented as an image (AF image) in which only that berry is changed to a color (white in current implementation) different from that of other berries. Thus, estimating the probability that a berry is the target to be removed is replaced by estimating the probability whether the corresponding AF image is the correct image or not. Figure 5.3 shows the AF image generation algorithm. First, DetectoRS (Qiao, Chen, and Yuille 2020), as introduced in 3, is used to detect the bunch and berries. From the detected results, a berry mask image, which is a binary image with the berry areas indicated in white was created. Then erosion morphology operation was applied to this binary image to shrink the area of each berry for a few pixels (pixels in current implementation). After that, the difference between the original berry mask image and the result from erosion operation was computed. As the result, the edge of the detected berries can be obtained. Next, to remove unnecessary information, both the original image and the edge image are cropped using the detected bunch's bounding box. Finally, for each of the detected berries, an AF image is created by changing the berry’s area in the cropped original image to white color and then adding the cropped edge image to it after changing the edge color to gray. The number of AF images created is the same as the number of berries and all AF images are fed to removing berry identification model for predicting the probability of being the berry to be removed.

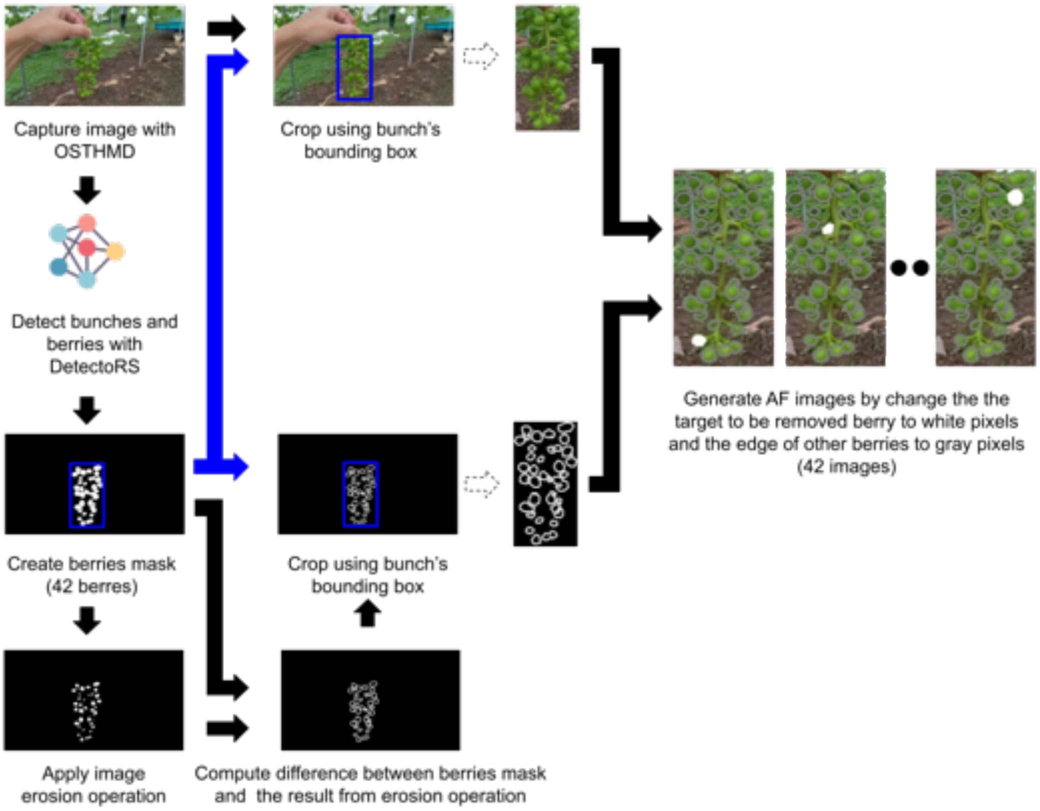

Capture image with OSTHMD
Crop using bunch's bounding box
Detect bunches and berries with DetectoRS
Generate AF images by change the the target to be removed berry to white pixels and the edge of other berries to gray pixels (42 images)
Create berries mask (42 berres)
Crop using bunch's bounding box
Apply image erosion operation
Compute difference between berries mask and the result from erosion operation

Figure 5.3 AF image generation process.

5.1.3 Removing berry identification model

Although, the classification of AF images is compatible with general image classification models (Gu et al. 2018), the existing hybrid network was made an extension as shown in Figure 5.4 to improve the accuracy. The image classification uses model combining convolution neural network (CNN) (Le Cun Jackel, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard et al. 1990) and Long Short-Term Memory network (LSTM) (Hochreiter and Schmidhuber 1997) was introduced by Islam et al (Islam, Islam, and Asraf 2020). They employed the encoder-decoder approach, the CNN based encoder is used to extract image features while LSTM is used to decode these features and perform the classification. Nevertheless, to train a successful model, a huge amount of labeled data is required. So Islam et al's CNN backbone is replaced by Resnet18 (He et al. 2016) and use transfer learning to fine tuning the model pretrained with ImageNet dataset (Deng et al. 2009), The size of modified hidden layers in LSTM is shown in Table 5.1.

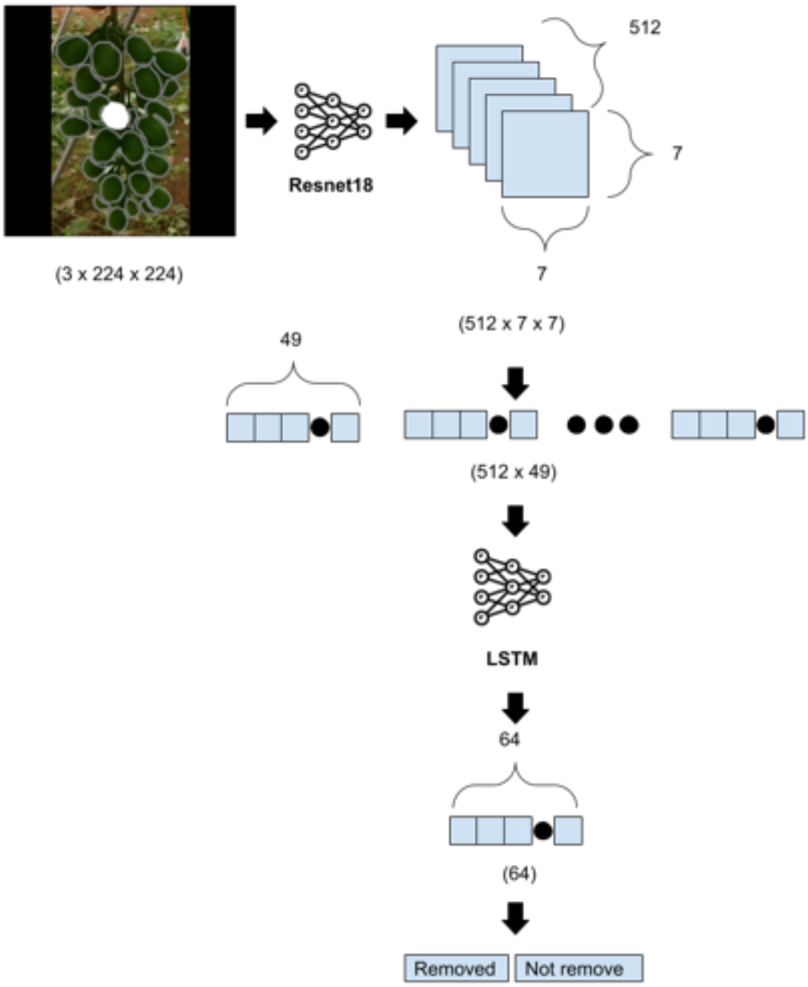


Figure 5.4 Hybrid network structure (Resnet18 + LSTM) for removing berry identification.

Table 5.1 Hybrid network (Resnet18 + LSTM) for removing berry identification.

Component	**Layer**	**Input size**	**Output size**
Resnet18	Input	3, 224, 224	64, 112, 112
	Layer 1	64, 112, 112	64, 56, 56
	Layer 2	64, 56, 56	128, 28, 28
	Layer 3	128, 28, 28	256, 14, 14
	Layer 4	256, 14, 14	512, 7, 7
LSTM	LSTM	512, 7, 7	64
Head	Fully connected	64	2

5.2 Experiment results and discussion

5.2.1 Dataset and implementation details

Two skilled farmers were asked to participate the data collection by capturing the working perspective with Microsoft HoloLens throughout the berry thinning task. Then the 'Removed' was added as an additional attribute when creating instance segmentation ground truth task using CVAT application (Sekachev et al. 2020) as in Figure 5.5. Each image has a resolution of 1,920 × 1,080 pixels, each was rescaled to 1,333 × 800 pixels for instance segmentation task, and to 224 × 224 pixels for removing berry identification task. The 723 removed berries in 723 images from 54 different bunches were manually labeled. Since it's challenging to collect the data, the dataset is small. The imbalance class was tackled by dividing classes' proportion by one 'Removed' berry per eight 'Not remove' berries. Totally 6,507 ground truth berries were obtained. 5,205 berries are used for training, and another 1,302 berries are used for the test. Both instance segmentation and removing berry identification models are trained and evaluated on the single Titan RTX GPU.

Image normalization was performed following the ImageNet dataset (Deng et al. 2009) for both the training and test stage. Throughout the training stage, image augmentation techniques, as shown in Table 5.2, were employed. Figure 5.6. shows the examples of applied augmentations for a particular image. Table 5.3. shows the hyper-parameter applied to the removing berry identification model.

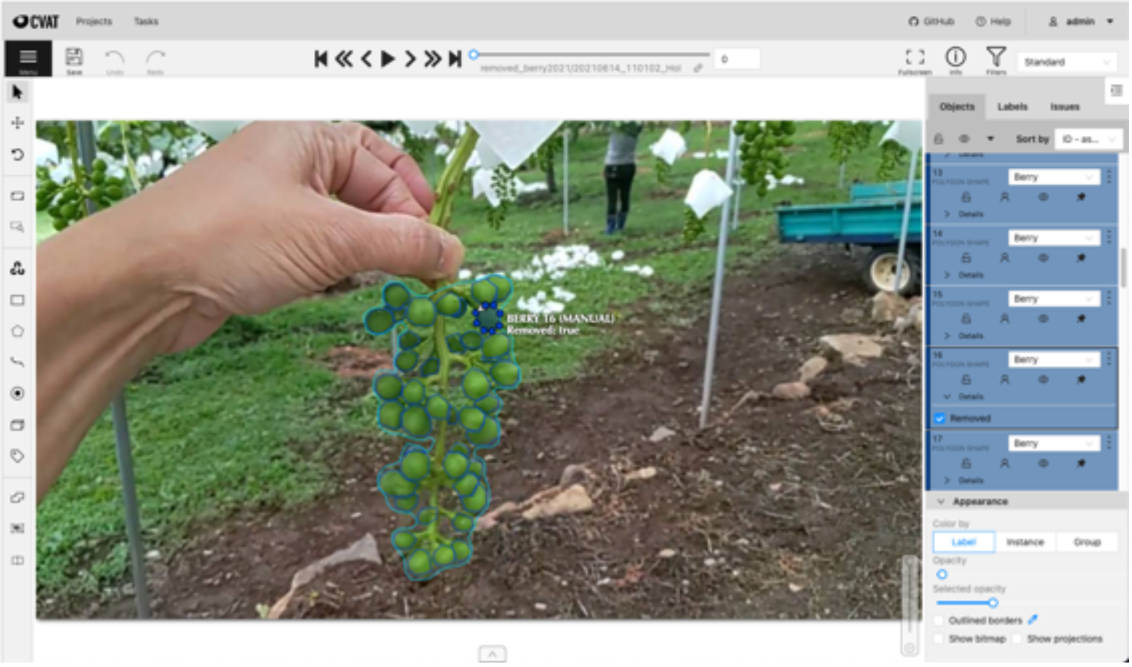


Figure 5.5 Manually label berries with removed attributes using the CVAT application (Sekachev et al. 2020).

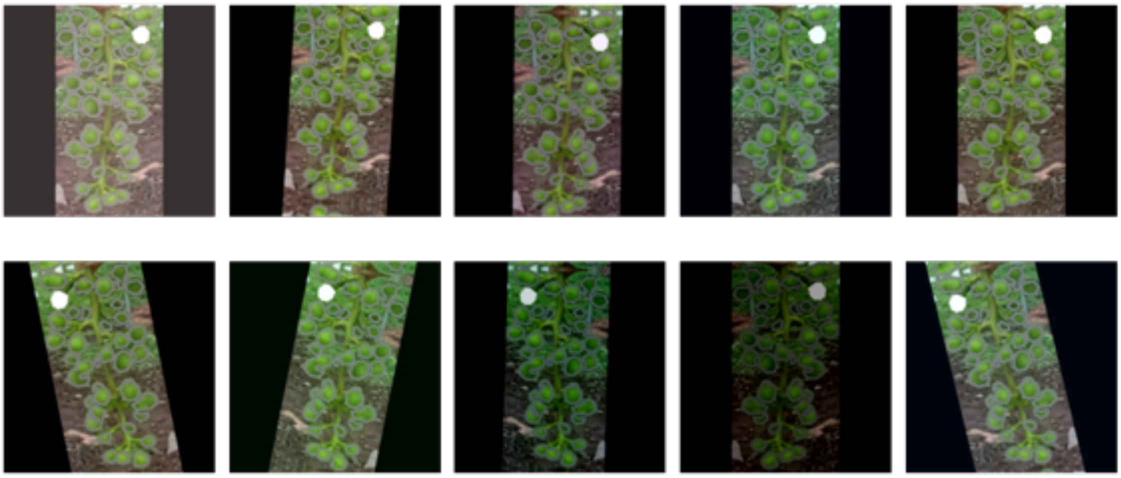


Figure 5.6 An example of image augmentation.

Table 5.2 Image augmentation techniques parameter setting.

Augmentation	**Property**	**Value**
Shift, scale, rotate	Shift limit	0.05
	Scale limit	0.05
	Rotate limit	15
	Probability	0.5
Horizontal flip	Probability	0.5
RGB shift	R shift	15
	G shift	15
	B shift	15
	Probability	0.5
Random brightness contrast	Probability	0.5

Table 5.3 Hyper-parameters applied to the removing berry identification model.

Property	**Value**
Loss function	Cross entropy
Optimizer	Stochastic gradient descent
Learning rate	0.001
Momentum	0.9
Epoch	1000

5.2.2 Evaluation metric

The following metric is used for evaluating the accuracy of prediction in this study:

$$Acc = \left(\frac{\sum_{i=1}^{N} x_i}{N}\right) \times 100 \tag{5.1}$$

Here, Acc is the average accuracy, x_i is the accuracy of berry i which is 1 when prediction result match the ground truth. Otherwise x_i is 0 and N is the number of images used for the test.

5.2.3 Experiment results

This section presents the evaluation results of the removing berry identification model using (5.1). Table 5.4 compares the various AF image styles when the model is trained with transfer learning. The size of AF images is 224 x 224 pixels. Three different AF image styles are described as following:

1. Without texture

The candidate berry mask is changed to white color and the mask of all other area are changed to another color, e.g. gray in the experiment, as shown in Figure 5.7 (a).

2. With texture

The original image was kept and only the candidate berry mask is changed to white color as shown in Figure 5.8 (a).

3. With texture and edges

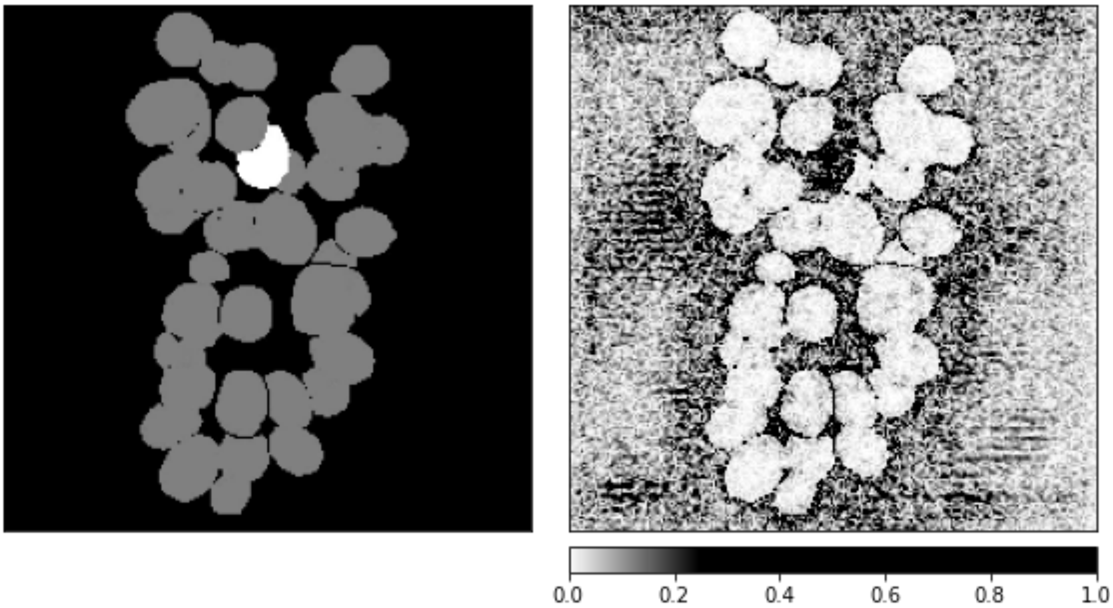
Change the candidate berry mask to white color. For other berries, change the edge to gray and keep the inner area to be the same as original image, as shown in Figure 5.9 (a).

Furthermore, the DNN model is expected to essentially consider the candidate berry based on nearby berry density or relative position among its neighbors. Hence, a model interpretability algorithm called GradientShap (Lundberg and Lee 2017) was employed to find the area that influences the classification result.

Firstly, for the AF image without texture, the berry removing model considers only the masks information to determine whether the candidate berry should be removed or not, because the input image style doesn't have the texture information. Figure 5.7 (b) shows the gradient image from the model interpretability algorithm. Black color means that the input pixel influences removing berry identification result the most. The result shows that the prediction result of the candidate berry was ambiguous and all other berries didn't influence the prediction results. It explains why the model couldn't be trained successfully with this AF image style.

Table 5.4 Comparison of the various input styles when used for transfer learning, input size 224 x 224 pixels.

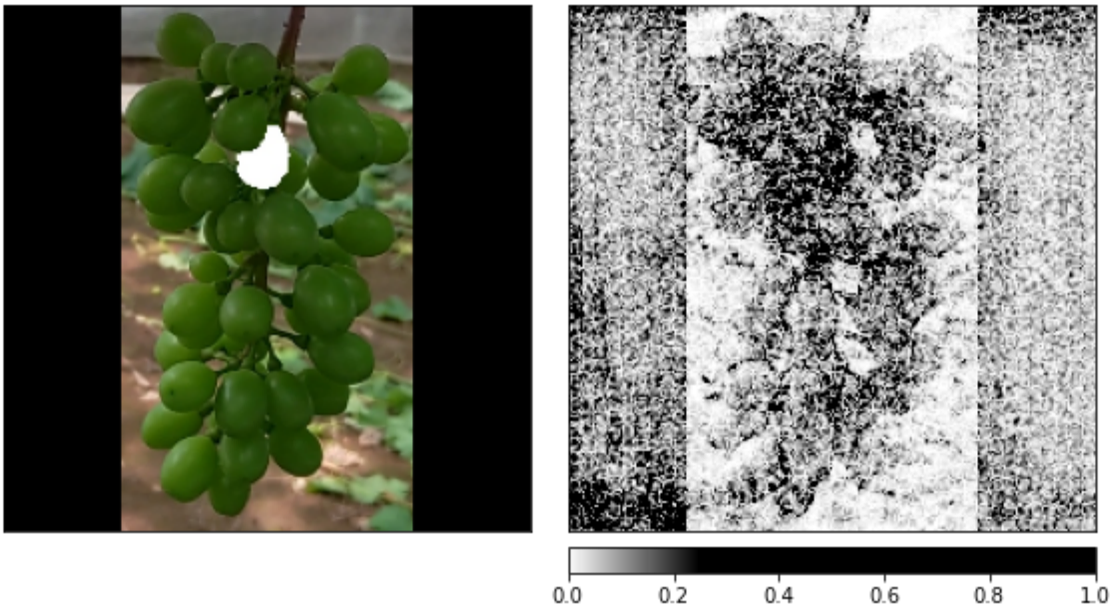
Model	Input style	Transfer learning	Accuracy (%)
Resnet18	Without texture	Yes	Underfitting
Resnet18	With texture	Yes	84.56
Resnet18	With texture and edges	Yes	84.87
Resnet18 + LSTM	**With Texture and edges**	**Yes**	**88.02**



a) AF image without texture **b) Gradient image**

Figure 5.7 Applying model interpretability algorithm using Resnet18. Input image size is 224 x 224 pixels. a) AF image without texture; b) gradient image (black color is the most influence).

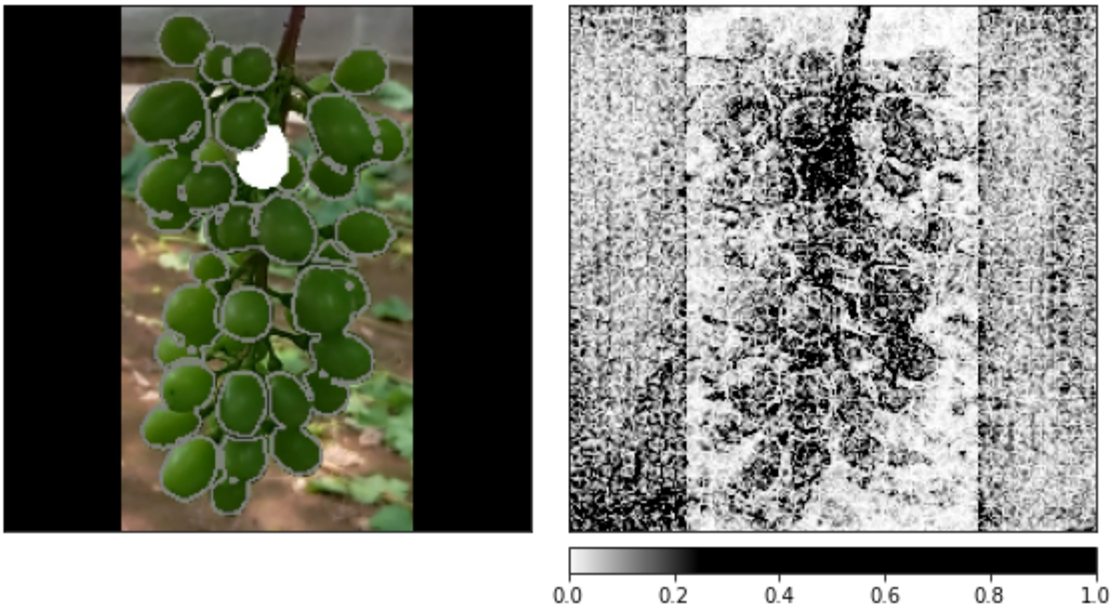
Secondly, for the AF image with texture, it could archive a high accuracy of 84.56%. The gradient image is shown in Figure 5.8 (b). The candidate berry and neighborhood berries are equally influencing identification results, which can ruin the accuracy. The candidate berry is expected to be the most influencing, and neighborhood berries should be the second. Besides, the boundary of the bunch is fuzzy compared to Figure 1.2 b).



a) AF image with texture **b) Gradient image**

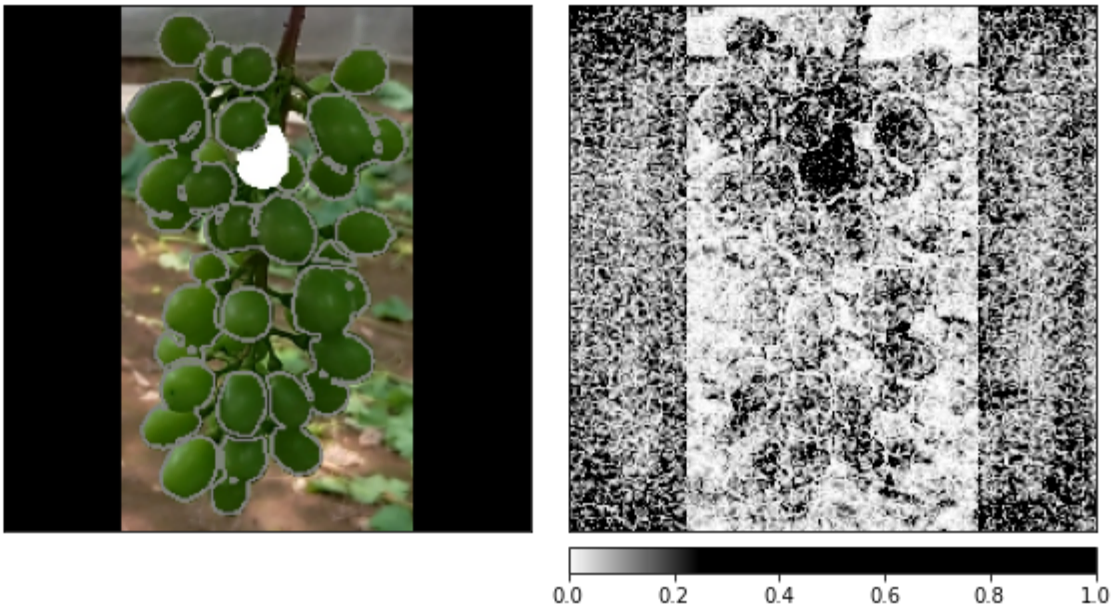
Figure 5.8 Applying model interpretability algorithm using Resnet18. Input image size is 224 x 224 pixels. a) AF image with texture; b) gradient image (black color is the most influence).

Thirdly, for AF image with texture and edges, the model should consider how the candidate berry locates with other berries by emphasizing the shape of the bunch. Instead of mask information, the edge information is used to keep the texture information from the input image. Although the accuracy is slightly improved from the second style (0.31%), the gradient image in Figure 5.9 (b) shows that this style enables the model focus on the candidate berries and at the same time, other berries are also considered as a second priority. Moreover, by replacing Resnet18 with the hybrid network (Resnet18 + LSTM) shown in Figure 5.10 (a), rather than directly convert extracted features from the CNN layer to another domain like a fully connected layer, the LSTM layer can consider every 2D feature patch of CNN features (Output features from Resnet18 size of 7x7 in Figure 5.4) and find the important features while discarding insignificant features to improve the identification accuracy. The hybrid network could increase the accuracy by 3.15%. The improvement can also be observed from the gradient image shown in Figure 5.10 (b), which makes the model focus on the candidate berry and its neighborhood more.



a) AF image with texture and edge b) Gradient image

Figure 5.9 Applying model interpretability algorithm using Resnet18. Input image size is 224 x 224 pixels. a) AF image with texture and edge; b) gradient image (black color is the most influence).



a) AF image with texture and edges b) Gradient image

Figure 5.10 Applying model interpretability algorithm using Resnet18 + LSTM. Input image siz 224 x 224 pixels. a) AF image with texture and edges; b) gradient image (black color is the mos influence).

Deep learning models have achieved excellent performance in visual recognition tasks (Gu et al. 2018). Nevertheless, a large amount of training data is required to train a successful model. And it is difficult to create adequate removing berry identification data, which consumes labor, time, and money. Therefore, transfer learning technique (Yosinski et al. 2014) was adopted, which initializes the training parameters with those trained with ImageNet dataset (Deng et al. 2009), containing millions of images. Transfer learning can improve generalization performance for a new task (Yosinski et al. 2014). As show in Table 5, training resnet18 from scratch resulted in an underfitted model.

Table 5.5 Comparison of with or without transfer learning.

Model	Input size	Transfer learning	Acc (%)	Parameters (M)	Inference speed (images/s)
Resnet18	224	No	Underfitting	11.17	3405.92
Resnet18	**224**	**Yes**	**84.26**	**11.17**	**3409.71**

Various image recognition models have been tested for removing berry identification to find the appropriate model size for the task. Since the goal of this study is to support berry thinning task in the actual field environment, the inference processing time should be considered. Therefore, the throughput of the models were measured. Considering the maximum number of berries in a bunch is around 70, the batch size was set to 70 in the throughput experiment. The throughput measurement was repeated over 100 times and then the average number of images that the model was capable of processing in one second (images/second) was computed.

Table 5.6 shows Resnet50 achieve the best accuracy of 85.87%. Using the larger model does not improve the accuracy but ruins it. The result shows the proper model size is around 11 to 23 million parameters. If the number of parameters exceeds the above range, the model's performance drops. For example, VGG16 model (Simonyan and Zisserman 2015), which comprises 134 million parameters, could not be trained successfully.

Table 5.6 Comparison of the various models.

Model	Input size	Transfer learning	Acc (%)	Parameters (M)	Inference speed (images/s)
Resnet18	224	Yes	84.87	**11.17**	**3409.71**
Resnet50	224	Yes	**85.87**	23.51	1043.75
Resnet101	224	Yes	83.79	42.50	628.86
VGG16	224	Yes	Underfitting	134.26	536.36
VGG16	640	Yes	Underfitting	134.26	56.01

The AF image was cropped from an original image using the detected bunch region, which is a small image. Besides, considering inference speed, for an image size of 224 x 224 pixels, the processing speed can be over 500 images per second. Especially for the Resnet18 model, it can achieve an inference speed over 3,400 images per second. While VGG16 model with increased input image size of 640 x 640 pixels can gain an inference speed of only 56.01 images per second, such gap in processing speed is too huge to trade for the image size reduction to 224 x 224 pixels. Thus, it can be said that using Resnet18 model and an image size of 224 x 224 pixels is appropriate for the removing berry identification model.

Figure 5.7 shows the results by varying the size of encoder for the hybrid CNN-LSTM model. The results show that Resnet18 encoder can achieve an accuracy of 88.02%, which is 3.76% higher than Resnet50 encoder. The reason is that the LSTM layer in Resnet50 + LSTM model increases model parameters by around 500,000. These additional parameters make the model too big to be trained with the available dataset consisting only 5,205 berry images. Moreover, hybrid Resnet18+LSTM can achieve an accuracy of 3.15% higher than using Resnet18 only. While the number of model parameters increases by 150,000, the drop of inference speed is only 162.79 images per second. From the above experiment results, hybrid Resnet18+LSTM model is suitable for removing berry prediction in real berry thinning task.

Table 5.7 Comparison of the hybrid CNN-LSTM with varying size of features extraction backbone.

Model	Input size	Transfer learning	Acc (%)	Parameters (M)	Inference speed (images/s)
Resnet18 + LSTM	224	Yes	**88.02**	**11.32**	**3246.92**
Resnet50 + LSTM	224	Yes	84.26	24.05	1035.58

The effect of detection post-processing depicted in Figure 5.2 was evaluated by interviewing the grape farmers who performed thinning tasks with the proposed technology. The interview results show that they are satisfied with the detection post-processing. It can make the system present the removing berry identification result consistently. Without the proposed technique, the removing berry identification result may change rapidly, and it causes fatigue to farmers and reduce their performance. On the contrary, with the proposed technique, the removing berry identification result does not change when farmers hold the bunch still, thus they can easily recognize the removal berry.

5.3 Summary

Considering that berry thinning is a significant task influencing the market value of table grapes, and the berry thinning season has time constraints, effective support technologies are highly desired by table grape farms. The proposed automatic removing berry identification technique can empower beginner farmers to start berry thinning without in-person coaching by expert farmers. It has been invented for practical use in a real grapevine environment. Integrating the detection post-processing improves the user experience by showing consistent results to farmers, and it can prevent farmers from getting fatigued and improve their performance. The image preprocessing technique, AF, compatible with the general DNN models for image classification, can also be used for identifying the target for trimming in the cultivation of other fruits or vegetables.

EXPERIMENTS IN A REAL TABLE GRAPE FIELD DURING THE ENTIRE GROWING SEASON

To verify the practicality of the proposed system in actual viticultural settings, the table grape field in Yamanashi prefecture, Japan, was selected to experiment with the proposed system during the entire annual cultivation process in season 2021. Table 6.1 is the list of the participants including 1 skilled farmer and 6 people who have no experience of grape cultivation at all. The 6 amateur participants, referred as unskilled farmers hereafter, were further divided into 3 groups by their age range. They performed inflorescence trimming and berry thinning tasks with the support of the proposed system while the 1 skilled farmer performed the tasks without using the proposed system. As shown in Figure 6.1, the table grape field for evaluation experiment was divided in to 4 sections, ① is a section by group Unskilled20-30 farmers; ② is by group Unskilled40-50 farmers; ③ is the by group UnskilledAbove60 farmers and ④ is by group Skilled farmers.

Two metrics were used to evaluate the proposed system. First is the operation time of unskilled farmers using the proposed system compared with skilled farmers. Second is the product quality of harvested table grape. This chapter present two main experiments: 1) Evaluation of inflorescence measurement for supporting table grape trimming. And 2) Evaluation of automatic berry-counting and removing berry prediction technique for supporting berry thinning.

Table 6.1 Farmer groups participated in real grape field environment.

Group	**Age range**	**Number of Participants**	**Abbreviation**
Unskilled farmer with proposed system	20-30	2	Unskilled20-30
Unskilled farmer with proposed system	40-50	2	Unskilled40-50
Unskilled farmer with proposed system	Above 60	2	UnskilledAbove60
Skilled farmer without proposes system	-	1	Skilled

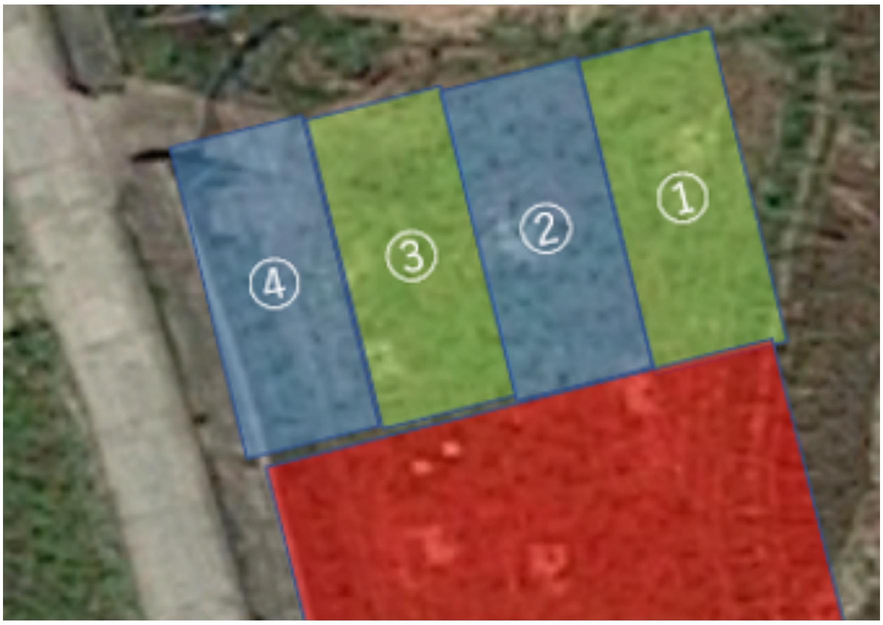


Figure 6.1 Table grape field for evaluating the proposed system in Yamanashi Prefecture, Japan. (1) is a section for group Unskilled20-30 farmers. (2) is a section for group Unskilled40-50 farmers. (3) is the section for group UnskilledAbove60 farmers. (4) is the section for group Skilled farmers.

6.1 Evaluation of inflorescence measurement for supporting table grape trimming

The evaluation of inflorescence measurement for supporting table grape trimming proceeded from 25 to 26 May 2021. Table 6.2 and Figure 6.2 shows the average operation time for the inflorescence trimming task. Each farmer was asked to trim the inflorescence, continuing with ten bunches per batch, four batches per farmer. The *target time* is the appropriate trim time per one bunch in terms of economic and cultivation schedule suggested by the Agriculture Department, Yamanashi prefecture, Japan. Almost every farmer can reduce the operation time after familiarizing themselves with the proposed system. All farmers who performed the fourth batch experiment can reach the ideal target time.

Figure 6.3 depicts the average operation time of farmer groups for inflorescence trimming tasks. Unskilled farmers from the age range 20-30 years old gain consistency improvement when they getting familiar with the proposed system. While Unskilled farmers from age range 40-50 years old gain the most improvement when they use to the proposed system. They even perform trimming tasks better than skilled farmers after trimming 30 bunches (third batch). Moreover, unskilled farmers from the age range above 60 years old also achieve good improvement. After using the proposed system to trim the inflorescence for ten bunches, the operation time could be reduced from 62.6 seconds to 44.4 seconds.

Figure 6.4 shows average operation time of all unskilled farmers for inflorescence trimming tasks. The results show that the proposed system can significantly empower inexperienced farmers to trim the inflorescence. The untrained farmers can become familiar with the proposed system very fast. After using the proposed system to trim the inflorescence for 40 bunches, the operation time was reduced from 71.2 seconds to 32.5 seconds. The target time is 35.9 seconds, and the skilled farmers can perform inflorescence trimming in 35.9 seconds. The unskilled farmers perform better than the target time and professional farmers without training beforehand. Such result indicates that it is suitable for the table grape industry that farm owners can hire an untrained farmer to trim the inflorescence immediately using the proposed system.

Table 6.2 Average operation time for inflorescence trimming task. Each farmer was asked to trim four batches and trimming continuing ten bunches per batch.

Group	**Farmer**	**Batch operation and average times per bunch (seconds)**			
		1st Batch	**2nd Batch**	**3rd Batch**	**4th Batch**
Unskilled20-30	A	107.7	75.6	46.7	39.5
	B	38.1	35.4	37	37
Unskilled40-50	C	84.6	102.8	24.7	30.9
	D	71.8	36.1	25.1	22.7
UnskilledAbove60	E	70.7	41.9	33.5	—
	F	54.4	46.9	49	—
Skilled	G	35.9			
Target time	H	39.3			

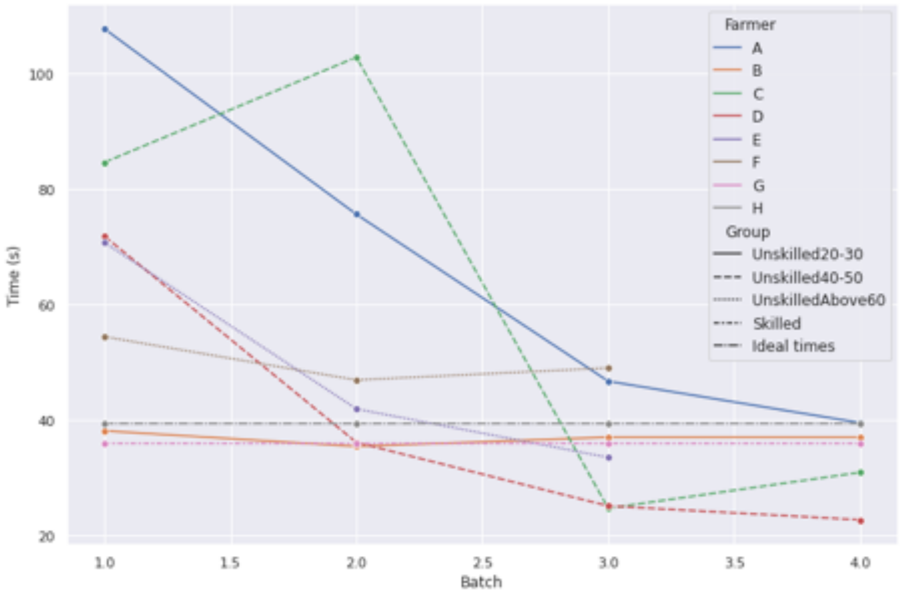


Figure 6.2 Individual farmers' average operation time for inflorescence trimming tasks. Each farmer was asked to trim the inflorescence, continuing with ten bunches per batch, four batches per farmer.

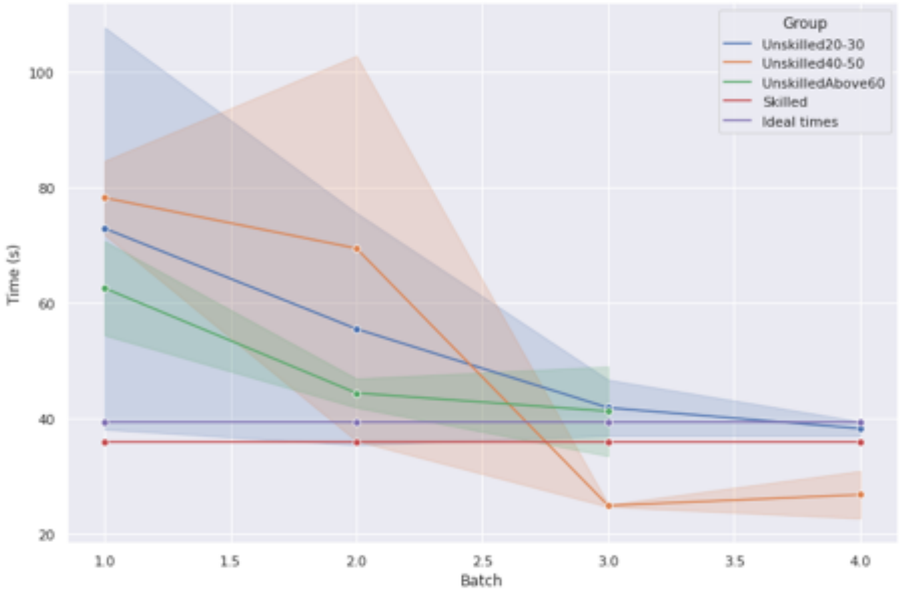


Figure 6.3 Group farmers' average operation time for inflorescence trimming tasks. Each farmer was asked to trim the inflorescence, continuing with ten bunches per batch, four batches per farmer.

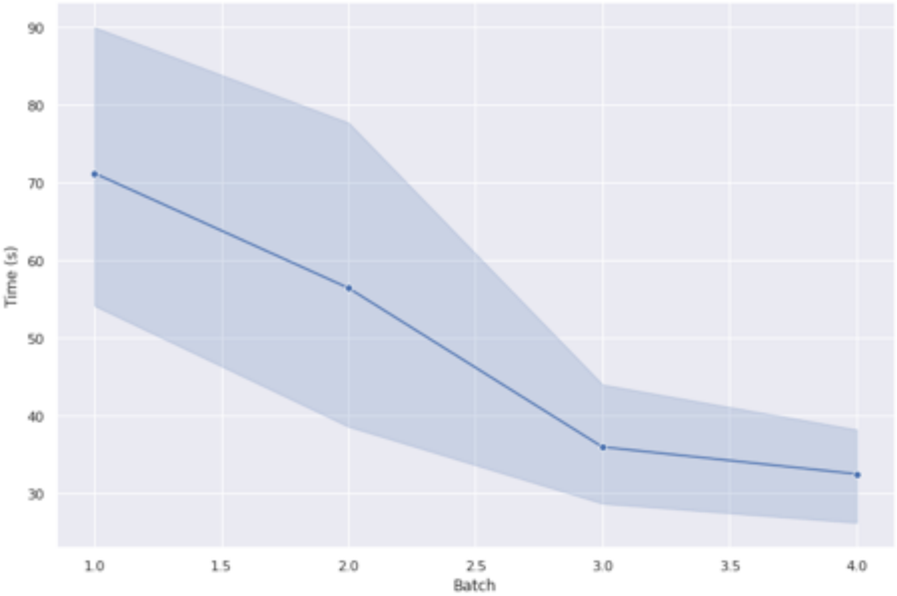


Figure 6.4 Unskilled farmers' average operation time for inflorescence trimming tasks. Each farmer was asked to trim the inflorescence, continuing with ten bunches per batch, four batches per farmer.

6.2 Evaluation of automatic berry-counting and removing berry identification techniques in berry thinning

The evaluation of automatic berry-counting and removing berry identification techniques were conducted from 15 to 17 June 2021.

Table 6.3 shows the average operation time for the berry thinning task. Each farmer was asked to thin the berries, continuing with ten bunches per bunch, four batches per farmer. The *target time* is the appropriate thinning time per one bunch in terms of economic and cultivation schedule suggested by the Agriculture Department, Yamanashi prefecture, Japan.

Figure 6.5 shows the individual farmers' average operation time for berry thinning tasks. Almost every farmer can reduce the operation time after familiarizing themselves with the proposed system. Nevertheless, even with the fifth batch, all farmers could not reach the ideal target time. The reason is that berry thinning is a very difficult and challenging task. Even though the proposed system can support two challenge factors: counting berries in the bunch and determining which berry should be removed, it is still difficult for unskilled farmers as they need to remove berries carefully to avoid causing damage to the neighborhood berries. Another reason is that it took some time for the participants to identify the berry indicated by the system. However, some of the unskilled farmers (E and D) almost reach the target time. After they thin the berry around 50 bunches, the berry thinning time are only 5.5 seconds and 7.5 seconds, respectively, longer than the skilled farmer.

Figure 6.6 depicts the average operation time of farmer groups for berry thinning tasks. Unskilled farmers from the age range 40-50 years old and above 60 years old have the learning rate faster than unskilled farmers from the age range 20-30 years old. It is likely because the berry thinning requires patients and concentration.

Figure 6.7 shows the average operation time of unskilled farmer groups for berry thinning tasks. The results show that the proposed system can significantly support inexperienced farmers to thin the berry. Their operation time can be improved after they get used to the system. After using the proposed system to thin the berry for 50 bunches, the operation time could be reduced from 100.5 seconds to 76.8 seconds. Since berry thinning is a highly skill demand task, it's impressive that unskilled farmers can perform the task without any training. Thus, it demonstrates that the proposed method is effective for supporting table grape cultivation. Besides, unskilled farmers tend to reach the target time after using the proposed system for around 60 or 70 bunches.

Table 6.3 Average operation time for berry thinning task. Each farmer was asked to thin fifth batches and thinning continuing ten bunches per batch.

Group	Farmer	Batch operation and average times per bunch (seconds)				
		1st Batch	2nd Batch	3rd Batch	4th Batch	5th Batch
Unskilled20-30	A	130.7	110.4	100.8	96.7	—
	B	95.8	87.5	88.7	84.3	85.3
Unskilled40-50	C	91.4	85	74.4	—	—
	D	92.2	84.5	89.2	79.8	73.5
UnskilledAbove60	E	98.9	82.7	88.7	82.4	71.5
	F	93.7	83.7	81.7	—	—
Skilled	G	53				
Target time	H	66				

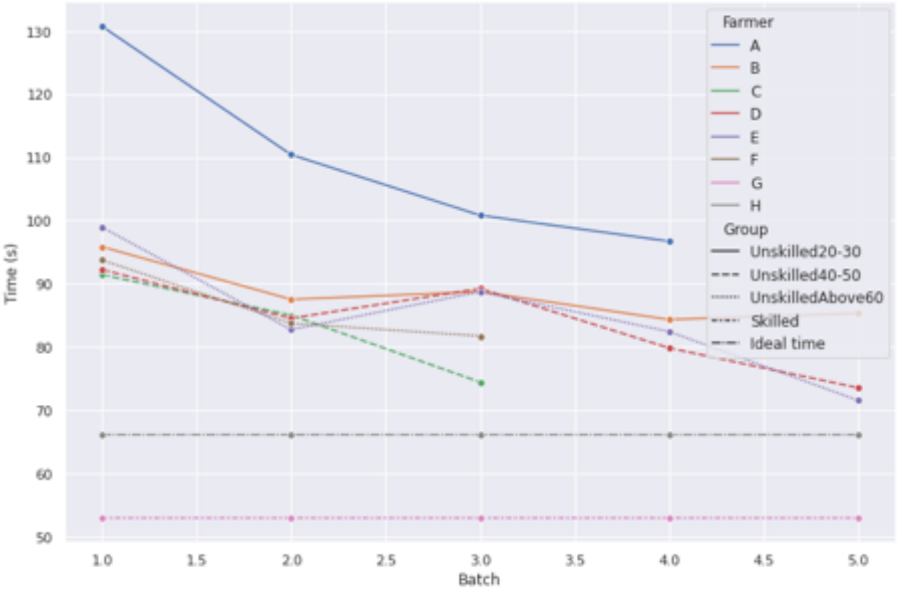


Figure 6.5 Individual farmers' average operation time for berry thinning tasks. Each farmer was asked to thin the berry, continuing with ten bunches per batch, five batches per farmer.

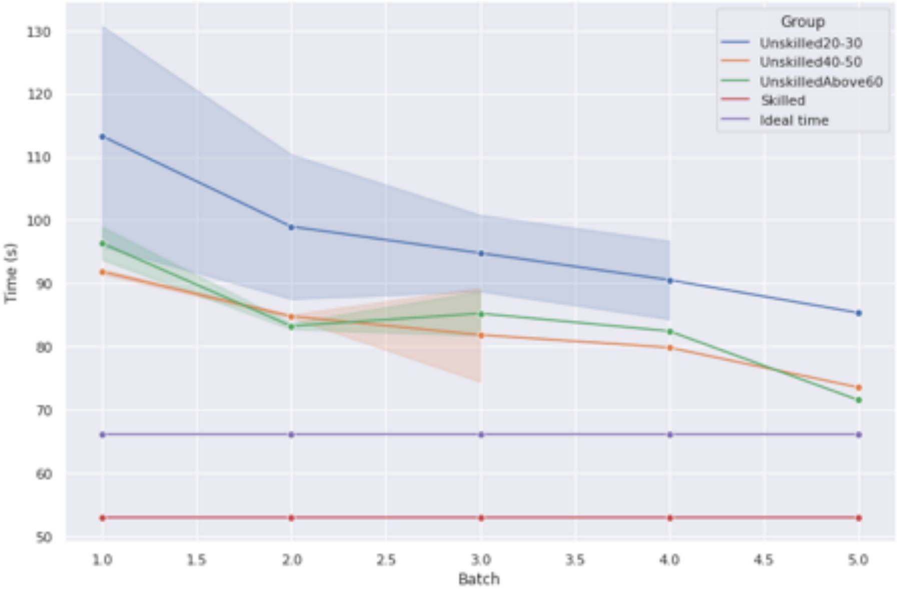


Figure 6.6 Group farmers' average operation time for berry thinning tasks. Each farmer was asked to thin the berry, continuing with ten bunches per batch, five batches per farmer.

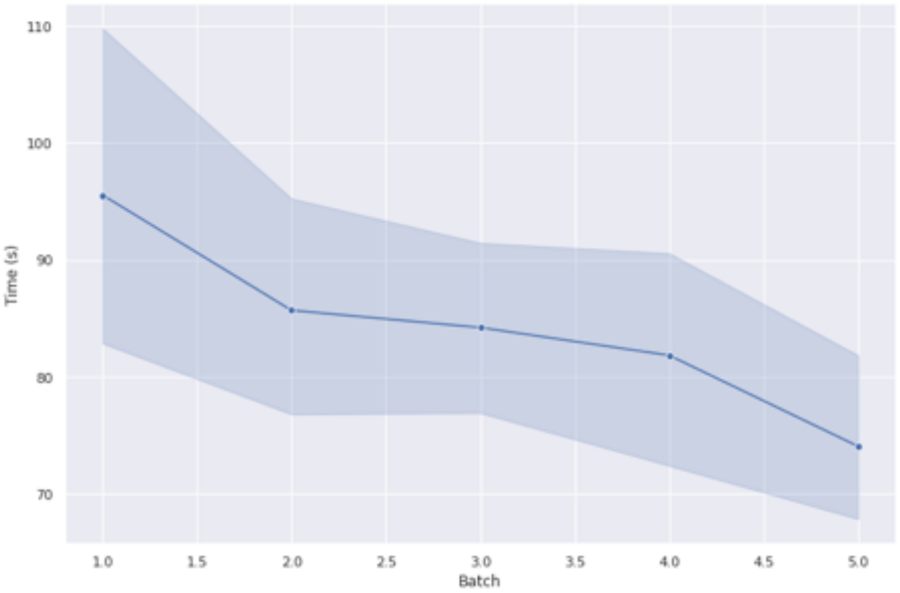
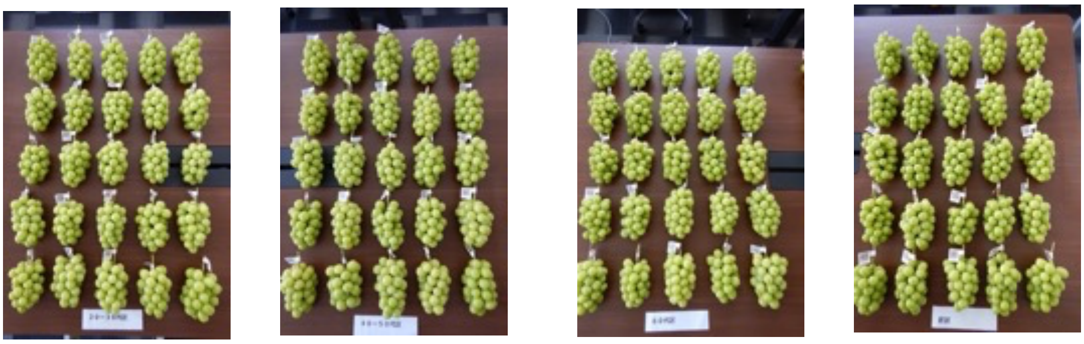


Figure 6.7 Unskilled farmers' average operation time for berry thinning tasks. Each farmer was asked to thin the berry, continuing with ten bunches per batch, five batches per farmer.

6.3 Quality evaluation at harvest time

The berries were harvested on 19 August, 6 September, 14 September, 27 September 2021. Figure 6.8 shows the example of harvested table grape. Table 6.4 shows attributes of harvested table grape grown by skilled farmers without using the proposed system and unskilled farmers using the proposed system. The values are average of 20 bunches for each farmer group. The average bunch weight of the bunches grown by the unskilled farmers using the proposed method is 13.8 grams higher than that of the skilled farmers. Also, the average berry weight by unskilled farmers using the proposed method is 1.99 grams more elevated than that by professional farmers. These two factors directly affect the market value of the table grape.

Table 6.5 compares the harvested table grape quality between skilled farmers without using the proposed system and unskilled farmers using the proposed system. The data is the average of 100 bunches grown by each farmer group. The grape qualities were judged by experts. The average quality score of the grapes grown by unskilled farmers using the proposed system was 8.18 % higher than that of the grapes grown by skilled farmers. Figure 6.9 shows the examples of harvested bunches. It can be confirmed that the grape grown by unskilled farmers using the proposed system have better bunch compactness than the grape by skilled farmer. The bunch by skilled farmer, as shown in Figure 6.9 (d) has a big gap at the top part of the grape which ruins the quality. The proposed removing berry identification fit the good training data while discarding the noise, and it enables farmers to thin the berries more consistently by preventing human error.



a) Unskilled20-30 b) Unskilled40-50 c) UnskilledAbove60 d) Skilled farmers

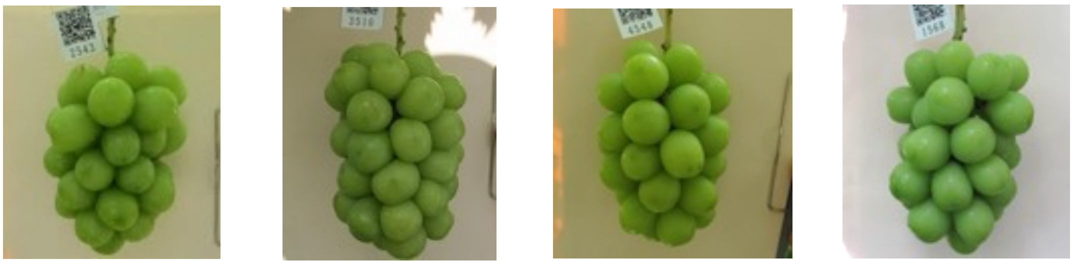
Figure 6.8 Examples of harvested table grapes grown by unskilled farmers using the proposed system and the skilled farmers without using the proposed system.

Table 6.4 Comparison of the attributes of harvested table grapes grown by skilled farmers without using proposed system and unskilled farmers using the proposed system. The results are the averaging of 20 bunches for each farmer group.

Farmer group	**Bunch length (cm)**	**Bunch weight (g)**	**Berry weight (g)**	**No. Berries in the bunch**	**Sugar concentration (%)**
Unskilled20-30	16.40	640.40	18.37	34.40	15.72
Unskilled40-50	17.20	633.00	19.92	31.40	15.92
UnskilledAbove60	16.70	617.80	18.60	32.60	16.44
Average of all unskilled farmers using the proposed system	16.77	**630.40**	**18.96**	32.80	16.03
Skilled farmers without using proposed system	16.90	616.60	16.97	35.60	16.78

Table 6.5 Comparison of harvested table grape quality between skilled farmers without using the proposed system and unskilled farmers using the proposed system. The results are the average of 100 bunches for each farmer group. The grape qualities were judged by experts.

Farmer group	Average quality score (%)
Unskilled20-30	67.21
Unskilled40-50	70.31
UnskilledAbove60	71.64
Average from all ages unskilled farmers with proposed system	69.72
Skilled farmers without proposed system	61.54



a) Unskilled20-30 **b) Unskilled40-50** **c) UnskilledAbove60** **d) Skilled farmers**

Figure 6.9 Comparison of harvested grape between unskilled farmers using the proposed system and skilled farmers without using the proposed system.

6.4 Summary

The proposed system has been designed to support two critical tasks in table grape cultivation, which affect the market value of table grapes. Experiments have been conducted in an actual table grape field during the entire growing season of 2021. The proposed system enables unskilled farmers to perform inflorescence trimming and berry thinning efficiently and accurately. The unskilled farmers can become to be familiar with the proposed system very fast. For inflorescence trimming tasks, they can perform better than the target time and even faster than professional farmers. The berry thinning task is more challenging than the inflorescence trimming task. Unskilled farmers can thin the berry faster when they get used to the system. The results show it tends to reach the target time. Nevertheless, it is impressive that unskilled farmers can perform the tasks without training.

Moreover, the average quality score of harvested table grapes grown by unskilled farmers using the proposed system was 8.18 % higher than that by skilled farmers. The proposed AI model can prevent human error to determine which berry should be removed.

CONCLUSION AND FUTURE WORK

Background: Inflorescence trimming and berry thinning are crucial processes in table grape cultivation. The reason is that bunch compactness, bunch form, and berry size are important factors affecting the market value of table grape production. During inflorescence trimming and berry thinning cultivation stages, the grape rapidly grows, and these tasks overwhelm the skilled farmers. Thus, the farm owners need unskilled farmers to alleviate the task load of skilled farmers. The inflorescence should be of a suitable length for the task of inflorescence trimming. In most cases, just 20–30 percent of an inflorescence is needed to produce a whole bunch of grapes, and the grape variety determines the perfect ideal length empirically. Trimming inflorescences efficiently requires a farmer to properly assess the length of the inflorescences using only their eyes, which is difficult for inexperienced farmers. While the optimal time for inflorescence trimming is one to two weeks, grape growers can considerably benefit from automated inflorescence measuring equipment that operates on a wearable device. Berry thinning is the most important step in table grape cultivation, as it directly impacts the final quality and market value of grapes. Berry thinning is a required procedure for removing unwanted berries and provide sufficient space for remaining berries to grow into desired size and quality. It benefits the production of both table grapes and wine grapes. Karoglan et al. discovered that combining bunch and berry thinning lowered grape yield while increasing mean cluster weight, total phenols, flavan-3-ols, and anthocyanins, as well as a variety of other phenolic chemicals. Consequently, the grape bunch opens up and becomes less susceptible to disease development. However, given the desirable overall shape of the bunch and the full size of matured berries, berry thinning requires professional grape farmers to accomplish such requirements. The table grape varieties have their ideal berry number range. Counting berries during berry thinning, on the other hand, takes time and is particularly challenging for new farmers. Furthermore, determining which berry should be removed is difficult for inexperienced farmers. The standard criteria must be used to consider the amount of berries per layer, the position of neighboring berries, and the overall shape of the bunch by visualizing how it will appear when completely grown. As a consequence, training inexperienced farmers to become professional berry thinning farmers is tough and time-consuming.

Proposal: This dissertation addresses challenging issues on using state-of-the-art AI and AR technology to support the inflorescence trimming and berry thinning tasks in grape cultivation. This dissertation successively proposed solution in table grape cultivation, building a functional application for the actual table grape farm environment to accomplish this goal. The novel end-to-end inflorescence measurement technology allows farmers to perform table grape trimming efficiently. The proposed approach uses 2D images of the trimming scene without requiring extra calibrators or high complexity preprocess. The experiment results demonstrate that proposed approach could achieve an outstanding result in inflorescence measurement. The measurement accuracy and the inference time are sufficient for use in the real table grape environment. The OSTHMD was employed to capture images and guide farmers without interrupting their trimming tasks. Then the novel end-to-end berry number prediction technology enables farmers to perform berry thinning efficiently. By integrating the location feature into the state-of-the-art instance segmentation DNN model, it was succeeded in focusing the berry detection on the working bunch only. The proposed location-sensitive HTC model can also be used for other object detection problems that require detecting a particular object from an image consisting of multiple objects of similar features. Using the originally designed features, berry number prediction can also be applied to the image-based counting of other kinds of fruits or vegetables. Finally, the automatic removing berry identification using a deep neural network with attention forcing technique was proposed for supporting berry thinning. The proposed method empowers beginner farmers to start berry thinning without in-person coaching by expert farmers. It has been invented for practical use in a real grapevine environment. Integrating the detection post-processing improves the user experience by showing consistent results to farmers, and it can prevent farmers from getting eye fatigued and improve working performance. Furthermore, the image preprocessing technique, 'attention forcing,' compatible with the general DNN models for image classification, succeeded in training the image classification models to predict the berry removal. Moreover, the proposed system was validated through the entire growing season in a real table grape field at Yamanashi prefecture, Japan. The unskilled farmers can execute the tasks immediately without training. Furthermore, they can be familiar with the proposed system quickly. The grape products from unskilled farmers who use the proposed system also have a 8.18% higher average quality score than skilled farmers.

Limitation and Future Direction: Even though the techniques proposed in this dissertation can successfully support table grape cultivation by building a functional application for the actual table grape farm environment, there are still some issues to be improved.

1. The proposed system are server-based applications. The advantage is that it can afford various devices, such as mobile applications, smart glasses, or an augmented reality headset, without concern about computation capacity. Nevertheless, the system requires the internet to access the AI server, and some grape farms may encounter difficulty in accessing the internet. I plan to implement the AI models to edge computing devices such as Nvidia jetson or mobile phones so that the system can be used in various environment even without internet connection.

2. The fact that automatic berry number prediction is based on the 2D image. When the farmer rotates the grape bunch, some berries hide at some angle. Thus the 3D counting prediction is changing and causing inconsistent results. And the best 3D counting prediction result could not reach MAE under two berries. Some approaches can tackle this issue. The first approach is tracking individual berries and counting the hidden berries when they appear. The second approach is to build the 3D model using the information from the instance segmentation model when farmer rotates the grape bunch. The third approach is counting only front berries via depth information (requiring a device supporting depth capturing, such as Microsoft HoloLens^{TM}). It is expected that all berries will be counted accurately when the bunch is fully turned 360 degrees.

3. The proposed technique for identifying the berry to be removed during the berry thinning has the

limitations with current visualization. It is difficult for the farmers to identify the berry indicated by the system in the real bunch. It will significantly improve farmer's operating time if the system can overlay the berry indicated by the system on the actual berry directly.

ACKNOWLEDGMENT

I am heartfully grateful to professor Xiaoyang Mao for her guidance in developing a methodology proposal, designing an experiment, and writing a paper. She counseled me on scientific research and how to become a researcher. Additionally, she carefully taught me the value of academic presentation and communication with colleagues. I respect her wisdom and dedication to education and study. She became my role model for both my career and personality. Moreover, she enthusiastically to find a scholarship to support my study. And she provides the best facility for research, such as the AI server and all smart gadget devices. I appreciate her allowing me to be her student, and I appreciate professor Kanda Runapongsa Saikaew for introducing me to professor Xiaoyang Mao when she visited Khon Kaen University. I would also like to thank professor Hiromitsu Nishizaki for his generous support and valuable comments on my dissertation.

I thank Mr. Koji Mitsui, Dream Farm Co., Ltd., for inspiring my research topic by proposing challenges and the room space that my research can fulfill in the actual table grape industry and cooperation with the project from the beginning. I thank my friends, Daisuke Inoue, Kabin Yok-In, Panitan Muangkammuen, Zhuguang Li, Kaishi Naito, Haoyan Jiang, and unnamed friends for their cooperation and discussion with my research. I would like to thank Mr. Naofumi Takayama for providing a great share house with a cheap rental fee during my study. I thank my friend Ikuhiro Nishikawa for giving me his car, making my daily life in Japan very convenient. I also thank my friends Bassamtiano Renaufalgi Irnawan and Meng Wang, who support me in getting my car driver's license in Japan. I thank Zhenyang Zhu, a friend who always supports Japanese translation.

I praise my parents, Sukon Buayai, Pattana Poolngam, who were my life mentors, giving me unconditional love and encouraging me in everything I dream of. I thank my little brother Sonthaya Buayai for his significant contribution to my research, and his data annotation task is mainly part of making my research successful. I want to thank the love of my life Jutarat Thongsirirat for her love, encouraging, and supporting me in everything I was meant to be.

This research was supported by a research grant from the Ministry of Agriculture, Forestry and Fisheries of Japan, "Feasibility Study Project on Smart Agriculture Using Local 5G". In addition, I received support from the Consortium for Analysis and Visualization of Professional Skill for High-Quality Shine-Muscat Grape Production. In particular, I would like to thank YSK e-com Co. Ltd. and the Yamanashi Prefectural Government for their generous cooperation.

REFERENCES

Aquino, Arturo et al. 2015. "VitisFlower®: Development and Testing of a Novel Android-Smartphone Application for Assessing the Number of Grapevine Flowers per Inflorescence Using Artificial Vision Techniques." *Sensors* 15(9): 21204–21218.

Aquino, Arturo, Ignacio Barrio, et al. 2018. "VitisBerry: An Android-Smartphone Application to Early Evaluate the Number of Grapevine Berries by Means of Image Analysis." *Computers and Electronics in Agriculture* 148: 19–28.

Aquino, Arturo, Maria P. Diago, Borja Millán, and Javier Tardáguila. 2017. "A New Methodology for Estimating the Grapevine-Berry Number per Cluster Using Image Analysis." *Biosystems Engineering* 156: 80–95.

Aquino, Arturo, Borja Millan, Maria-Paz Diago, and Javier Tardaguila. 2018. "Automated Early Yield Prediction in Vineyards from On-the-Go Image Acquisition." *Computers and Electronics in Agriculture* 144: 26–36.

Arad, Boaz et al. 2020. "Development of a Sweet Pepper Harvesting Robot." *Journal of Field Robotics* 37(6): 1027–1039.

Azuma, Ronald T. 1997. "A Survey of Augmented Reality." *Presence: Teleoperators and Virtual Environments* 6(4): 355–385.

Barbedo, Jayme Garcia Arnal. 2019. "Plant Disease Identification from Individual Lesions and Spots Using Deep Learning." *Biosystems Engineering* 180: 96–107.

Bertalanffy, Ludwig Von. 1938. "A Quantitative Theory of Organic Growth (Inquiries on Growth Laws. II)." *Human Biology* 10(2): 181–213.

Bimber, Oliver, and Ramesh Raskar. 2006. "Modern Approaches to Augmented Reality." In *ACM SIGGRAPH 2006 Courses*, SIGGRAPH '06, New York, NY, USA: Association for Computing Machinery.

Bishop, Christopher M. 2006. 4 Pattern Recognition *Pattern Recognition and Machine Learning*. eds. M Jordan, J Kleinberg, and B Schölkopf. Springer.

Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao. 2020. "YOLOv4: Optimal Speed and Accuracy of Object Detection." *arXiv preprint*.

Bolya, Daniel, Chong Zhou, Fanyi Xiao, and Yong-Jae Lee. 2019. "YOLACT: Real-Time Instance Segmentation." In *IEEE International Conference on Computer Vision*, Institute of Electrical and Electronics Engineers Inc., 9156–9165.

Bolya, Daniel, Chong Zhou, Fanyi Xiao, and Yong-Jae Lee. 2022. "YOLACT++ Better Real-Time Instance Segmentation." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44: 1108–1121.

Botterill, Tom et al. 2017. "A Robot System for Pruning Grape Vines." *Journal of Field Robotics* 34(6): 1100–1122.

Bottou, Léon. 2010. "Large-Scale Machine Learning with Stochastic Gradient Descent." In *Proceedings of COMPSTAT*, Springer, 177–186.

Bottou, Léon, Frank E Curtis, and Jorge Nocedal. 2018. "Optimization Methods for Large-Scale Machine Learning." *SIAM Review* 60(2): 223–311.

Breiman, Leo. 2001. "Random Forests." *Machine Learning* 45(1): 5–32.

Brooks, Justin. 2019. “COCO Annotator.”

Buayai, Prawit, Kanda Runapongsa Saikaew, and Xiaoyang Mao. 2021. “End-to-End Automatic Berry Counting for Table Grape Thinning.” *IEEE Access* 9: 4829–4842.

Burger, Wilhelm, and Mark James Burge. 2009. Interactive Image Processing for Machine Vision *Principles of Digital Image Processing*. London: Springer London.

Cai, Zhaowei, and Nuno Vasconcelos. 2019. “Cascade R-CNN: High Quality Object Detection and Instance Segmentation.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Chen, Kai, Jiangmiao Pang, et al. 2019. “Hybrid Task Cascade for Instance Segmentation.” In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, , 4969–4978.

Chen, Kai, Wanli Ouyang, et al. 2019. “Hybrid Task Cascade for Instance Segmentation.” *Proceedings of the IEEE International Conference on Computer Vision*: 4969–4978.

Creasy, Glen L, and Leroy L Creasy. 2018. 27 *Grapes*. 2nd ed. CABI.

Le Cun Jackel, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D., Bb Le Cun, Js Denker, and D. Henderson. 1990. “Handwritten Digit Recognition with a Back-Propagation Network.” In *Advances in Neural Information Processing Systems*, , 396–404.

Danielsson, Oscar, Magnus Holm, and Anna Syberfeldt. 2020. “Augmented Reality Smart Glasses for Operators in Production: Survey of Relevant Categories for Supporting Operators.” *Procedia CIRP* 93: 1298–1303.

Dassot, Mathieu, Meriem Fournier, and Christine Deleuze. 2019. “Assessing the Scaling of the Tree Branch Diameters Frequency Distribution with Terrestrial Laser Scanning: Methodological Framework and Issues.” *Annals of Forest Science* 76(3).

Deng, J et al. 2009. “ImageNet: A Large-Scale Hierarchical Image Database.” In *IEEE Conference on Computer Vision and Pattern Recognition*,.

Duan, Yiping et al. 2017. “SAR Image Segmentation Based on Convolutional-Wavelet Neural Network and Markov Random Field.” *Pattern Recognition* 64: 255–267.

Dyrmann, Mads, Henrik Karstoft, and Henrik Skov Midtiby. 2016. “Plant Species Classification Using Deep Convolutional Neural Network.” *Biosystems Engineering* 151: 72–80.

FAO. 2009. “How to Feed the World in 2050, High-Level Expert Forum.” *Food and Agriculture Organization of the United Nations*: 35.

Font, Davinia et al. 2014. “A Proposal for Automatic Fruit Harvesting by Combining a Low Cost Stereovision Camera and a Robotic Arm.” *Sensors* 14(7): 11557–11579.

Forsyth, David A, and Jean Ponce. 2002. *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference.

Friedman, Jerome H. 2001. “Greedy Function Approximation: A Gradient Boosting Machine.” *Annals of statistics*: 1189–1232.

Friel, John J. 2000. *Practical Guide to Image Analysis*.

Gebbers, Robin, and Viacheslav I. Adamchuk. 2010. “Precision Agriculture and Food Security.” *Science* 327(5967): 828–831.

Géron, Aurélien. 2019. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. O’Reilly Media.

Girshick, Ross. 2015. “Fast R-CNN.” In *Proceedings of the IEEE International Conference on Computer Vision*, , 1440–1448.

Goyal, Priya et al. 2017. *Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour*.

Gu, Jiuxiang et al. 2018. “Recent Advances in Convolutional Neural Networks.” *Pattern Recognition* 77: 354–377.

Hawkins, Douglas M. 2004. “The Problem of Overfitting.” *Journal of Chemical Information and Computer Sciences* 44(1): 1–12.

He, Kaiming, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. 2017. “Mask R-CNN.” In *IEEE International Conference on Computer Vision*,.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. “Deep Residual Learning for Image Recognition.” In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, IEEE Computer Society, 770–778.

Hochreiter, Sepp, and Jürgen Schmidhuber. 1997. “Long Short-Term Memory.” *Neural Computation* 9(8): 1735–1780.

Huang, Xia, Shunyi Zheng, and Li Gui. 2021. “Automatic Measurement of Morphological Traits of Typical Leaf Samples.” *Sensors* 21(6): 2247.

Huuskonen, Janna, and Timo Oksanen. 2018. “Soil Sampling with Drones and Augmented Reality in Precision Agriculture.” *Computers and Electronics in Agriculture* 154: 25–35.

Huuskonen, Janna, and Timo Oksanen. 2019. “Augmented Reality for Supervising Multirobot System in Agricultural Field Operation.” In *IFAC-PapersOnLine*, Elsevier, 367–372.

Iizuka, Satoshi, Edgar Simo-Serra, and Hiroshi Ishikawa. 2017. “Globally and Locally Consistent Image Completion.” *ACM Transactions on Graphics* 36(4): 107.

Islam, Md Zabirul, Md Milon Islam, and Amanullah Asraf. 2020. “A Combined Deep CNN-LSTM Network for the Detection of Novel Coronavirus (COVID-19) Using X-Ray Images.” *Informatics in Medicine Unlocked* 20: 100412.

Ivorra, E. et al. 2015. “Assessment of Grape Cluster Yield Components Based on 3D Descriptors Using Stereo Vision.” *Food Control* 50: 273–282.

Jackson, Ron S. 2000. “Grapevine Structure and Function.” In *Wine Science*, , 45–95.

Ji, Wei et al. 2012. “Automatic Recognition Vision System Guided for Apple Harvesting Robot.” *Computers and Electrical Engineering* 38(5): 1186–1195.

Jocher, Glenn et al. 2021. “Yolov5.” *Zenodo*.

Kaack, K, and H. Lindhard Pedersen. 2010. “Prediction of Diameter, Weight and Quality of Apple Fruit (Malus Domestica Borkh.) Cv. ‘Elstar’ Using Climatic Variables and Their Interactions.” *European Journal of Horticultural Science* 75(2): 60–70.

Kaizu, Yutaka, and Jongmin Choi. 2012. “Development of a Tractor Navigation System Using Augmented Reality.” *Engineering in Agriculture, Environment and Food* 5(3): 96–101.

Kamilaris, Andreas, Feng Gao, Francesc X. Prenafeta-Boldu, and Muhammad Intizar Ali. 2017. “Agri-IoT: A Semantic Framework for Internet of Things-Enabled Smart Farming Applications.” In *IEEE 3rd World Forum on Internet of Things*, Institute of Electrical and Electronics Engineers Inc., 442–447.

Kamilaris, Andreas, Andreas Kartakoullis, and Francesc X. Prenafeta-Boldú. 2017. “A Review on the Practice of Big Data Analysis in Agriculture.” *Computers and Electronics in Agriculture* 143: 23–37.

Kamilaris, Andreas, and Francesc X. Prenafeta-Boldú. 2018. “Deep Learning in Agriculture: A Survey.” *Computers and Electronics in Agriculture* 147: 70–90.

Karkee, Manoj, Bikram Adhikari, Suraj Amatya, and Qin Zhang. 2014. “Identification of Pruning Branches in Tall Spindle Apple Trees for Automated Pruning.” *Computers and Electronics in Agriculture* 103: 127–135.

Karoglan, M, M Osrečak, L Maslov, and B Kozina. 2014. “Effect of Cluster and Berry Thinning on Merlot and Cabernet Sauvignon Wines Composition." *Czech Journal of Food Sciences* 32(No. 5): 470–476.

Katsaros, Alexander, and Euclid Keramopoulos. 2017. "FarmAR, a Farmer's Augmented Reality Application Based on Semantic Web." In *South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference*, Institute of Electrical and Electronics Engineers Inc.

King, Gary R, Wayne Piekarski, and Bruce H Thomas. 2005. "ARVino — Outdoor Augmented Reality Visualisation of Viticulture GIS Data." In *Proceedings of the 4th IEEE/ACM International Symposium on Mixed and Augmented Reality*, ISMAR '05, USA: IEEE Computer Society, 52–55.

Kitzes, Justin et al. 2008. "Shrink and Share: Humanity's Present and Future Ecological Footprint." *Philosophical Transactions of the Royal Society B: Biological Sciences* 363(1491): 467–475.

Kussul, Nataliia, Mykola Lavreniuk, Sergii Skakun, and Andrii Shelestov. 2017. "Deep Learning Classification of Land Cover and Crop Types Using Remote Sensing Data." *IEEE Geoscience and Remote Sensing Letters* 14(5): 778–782.

Law, Hei, and Jia Deng. 2020. "CornerNet: Detecting Objects as Paired Keypoints." *International Journal of Computer Vision* 128(3): 642–656.

Lecun, Yann, and Yoshua Bengio. 1995. "Convolutional Networks for Images, Speech, and Time Series." *The handbook of brain theory and neural networks* 3361(10): 1995.

Lecun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. "Deep Learning." *Nature* 521(7553): 436–444.

Li, Ming et al. 2015. "Apple Fruit Diameter and Length Estimation by Using the Thermal and Sunshine Hours Approach and Its Application to the Digital Orchard Management Information System" ed. Changjie Xu. *PLoS ONE* 10(4).

Liaghat, S., and S. K. Balasundram. 2010. "A Review: The Role of Remote Sensing in Precision Agriculture." *American Journal of Agricultural and Biological Science* 5(1): 50–55.

Lin, TsungYi et al. 2017. "Focal Loss for Dense Object Detection." In *Proceedings of the IEEE International Conference on Computer Vision*, Institute of Electrical and Electronics Engineers Inc., 2999–3007.

Liu, Scarlett, Mark Whitty, and Steve Cossell. 2015. "A Lightweight Method for Grape Berry Counting Based on Automated 3D Bunch Reconstruction from a Single Image." In *ICRA, International Conference on Robotics and Automation (IEEE), Workshop on Robotics in Agriculture*, , 4.

Lu, Xingtong et al. 2020. "Reconstruction Method and Optimum Range of Camera-Shooting Angle for 3D Plant Modeling Using a Multi-Camera Photography System." *Plant Methods* 16(1).

Lundberg, Scott M., and Su In Lee. 2017. "A Unified Approach to Interpreting Model Predictions." In *Advances in Neural Information Processing Systems*, Neural information processing systems foundation, 4766–4775.

Luo, Lufeng et al. 2016. "Robust Grape Cluster Detection in a Vineyard by Combining the AdaBoost Framework and Multiple Color Components." *Sensors* 16(12): 2098.

Mitsui, Koji. 2019. *Smart Agri - Basic Information about the Management Work of Grapes*. Kofu City, Yamanashi Prefecture.

Mottaghi, Roozbeh et al. 2014. "The Role of Context for Object Detection and Semantic Segmentation in the Wild." In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, , 891–898.

Murphy, Kevin P. 2012. *Machine Learning: A Probabilistic Perspective*. MIT press.

Nellithimaru, Anjana K., and George A. Kantor. 2019. "ROLS : Robust Object-Level SLAM for Grape Counting." In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, , 2648–2656.

Neto, M., and P. Cardoso. 2013. "Augmented Reality Greenhouse." In *EFITA- WCCA-CIGR Conference "Sustainable Agriculture through ICT Innovation,"* , 1–8.

Nigam, Apurv, Priyanka Kabra, and Pankaj Doke. 2011. "Augmented Reality in Agriculture." In *International Conference on Wireless and Mobile Computing, Networking and Communications*, , 445–448.

Nuske, Stephen et al. 2014. "Automated Visual Yield Estimation in Vineyards." *Journal of Field Robotics* 31(5): 837–860.

Okamoto, Goro. 2007. "Effect of Shoot and Cluster Nutrition on Grape Berry Set." *Journal of ASEV JAPAN* 18(1): 36–45.

Okayama, Tsuyoshi, and Kazuya Miyawaki. 2013. "The 'Smart Garden' System Using Augmented Reality." In *IFAC Proceedings Volumes*, IFAC Secretariat, 307–310.

Ozdogan, Mutlu, Yang Yang, George Allez, and Chelsea Cervantes. 2010. "Remote Sensing of Irrigated Agriculture: Opportunities and Challenges." *Remote Sensing* 2(9): 2274–2304.

Pérez-Zavala, Rodrigo, Miguel Torres-Torriti, Fernando Auat Cheein, and Giancarlo Troni. 2018. "A Pattern Recognition Strategy for Visual Grape Bunch Detection in Vineyards." *Computers and Electronics in Agriculture* 151: 136–149.

Qiao, Siyuan, Liang Chieh Chen, and Alan Yuille. 2020. "DetectoRS: Detecting Objects with Recursive Feature Pyramid and Switchable Atrous Convolution." *arXiv*.

Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. "You Only Look Once: Unified, Real-Time Object Detection." In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, , 779–788.

Redmon, Joseph, and Ali Farhadi. 2017. "YOLO9000: Better, Faster, Stronger." In *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition*, , 6517–6525.

Redmon, Joseph, and Ali Farhadi. 2018. "YOLOv3: An Incremental Improvement." *Arxiv*.

Reis, M. J.C.S. et al. 2012. "Automatic Detection of Bunches of Grapes in Natural Environment from Color Images." *Journal of Applied Logic* 10(4): 285–290.

Ren, Shaoqing, Kaiming He, Ross Girshick, and Jian Sun. 2017. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39(6): 1137–1149.

Roscher, Ribana et al. 2014. "Automated Image Analysis Framework for High-Throughput Determination of Grapevine Berry Sizes Using Conditional Random Fields." *Computers and Electronics in Agriculture* 100: 148–158.

Rudolph, Robert, Katja Herzog, Reinhard Töpfer, and Volker Steinhage. 2019. "Efficient Identification, Localization and Quantification of Grapevine Inflorescences and Flowers in Unprepared Field Images Using Fully Convolutional Networks." *Vitis - Journal of Grapevine Research* 58(3): 95–104.

Sa, Inkyu et al. 2016. "Deepfruits: A Fruit Detection System Using Deep Neural Networks." *Sensors (Switzerland)* 16(8).

Santana-Fernández, Javier, Jaime Gómez-Gil, and Laura del-Pozo-San-Cirilo. 2010. "Design and Implementation of a GPS Guidance System for Agricultural Tractors Using Augmented Reality Technology." *Sensors (Basel, Switzerland)* 10(11): 10435–10447.

Santos, Thiago T., Leonardo L. de Souza, Andreza A. dos Santos, and Sandra Avila. 2020. "Grape Detection, Segmentation, and Tracking Using Deep Neural Networks and Three-Dimensional Association." *Computers and Electronics in Agriculture* 170.

Saxena, Lalit, and Leisa Armstrong. 2014. "A Survey of Image Processing Techniques for Agriculture." In *Proceedings of Asian Federation for Information Technology in Agriculture.*, , 401–413.

Schöler, Florian, and Volker Steinhage. 2015. "Automated 3D Reconstruction of Grape Cluster Architecture from Sensor Data for Efficient Phenotyping." *Computers and Electronics in Agriculture* 114: 163–177.

Sekachev, Boris et al. 2020. "Computer Vision Annotation Tool (CVAT)."

Silwal, Abhisesh et al. 2017. "Design, Integration, and Field Evaluation of a Robotic Apple Harvester." *Journal of Field Robotics* 34(6): 1140–1159.

Simonyan, Karen, and Andrew Zisserman. 2015. "Very Deep Convolutional Networks for Large-Scale Image Recognition." In *International Conference on Learning Representations*, International Conference on Learning Representations, ICLR.

Singh, Arti, Baskar Ganapathysubramanian, Asheesh Kumar Singh, and Soumik Sarkar. 2016. "Machine Learning for High-Throughput Stress Phenotyping in Plants." *Trends in Plant Science* 21(2): 110–124.

Stajnko, D., M. Lakota, and M. Hočevar. 2004. "Estimation of Number and Diameter of Apple Fruits in an Orchard during the Growing Season by Thermal Imaging." *Computers and Electronics in Agriculture* 42(1): 31–42.

Sun, Ke, Bin Xiao, Dong Liu, and Jingdong Wang. 2019. "Deep High-Resolution Representation Learning for Human Pose Estimation." In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Los Alamitos, CA, USA: IEEE Computer Society, 5686–5696.

Tan, Mingxing, and Quoc V. Le. 2019. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks." In *International Conference on Machine Learning*, , 10691–10700.

Teke, Mustafa et al. 2013. "A Short Survey of Hyperspectral Remote Sensing Applications in Agriculture." In *Proceedings of 6th International Conference on Recent Advances in Space Technologies*, , 171–176.

Tijskens, L. M.M., S. van Mourik, J A Dieleman, and R E Schouten. 2020. "Size Development of Tomatoes Growing in Trusses: Linking Time of Fruit Set to Diameter." *Journal of the Science of Food and Agriculture* 100(10): 4020–4028.

Vidal, N. R., and R. A. Vidal. 2010. "Augmented Reality Systems for Weed Economic Thresholds Applications." *Planta Daninha* 28(2): 449–454.

Wang, Zhou, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P. Simoncelli. 2004. "Image Quality Assessment: From Error Visibility to Structural Similarity." *IEEE Transactions on Image Processing* 13(4): 600–612.

Xi, Mingze, Matt Adcock, and John McCulloch. 2018. "Future Agriculture Farm Management Using Augmented Reality." In *IEEE Workshop on Augmented and Virtual Realities for Good*, Institute of Electrical and Electronics Engineers Inc.

Xiong, Ya, Yuanyue Ge, Lars Grimstad, and Pål J. From. 2020. "An Autonomous Strawberry-Harvesting Robot: Design, Development, Integration, and Field Evaluation." *Journal of Field Robotics* 37(2): 202–224.

Yang, Zishang, and Yuxing Han. 2020. "A Low-Cost 3D Phenotype Measurement Method of Leafy Vegetables Using Video Recordings from Smartphones." *Sensors* 20(21): 1–15.

Yosinski, Jason, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. "How Transferable Are Features in Deep Neural Networks?" In *Advances in Neural Information Processing Systems*, Neural information processing systems foundation, 3320–3328.

Zabawa, Laura et al. 2019. "Detection of Single Grapevine Berries in Images Using Fully Convolutional Neural Networks." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, , 2571–2579.