Executive Report of Grape Project

By Ian Brown, Peiqian Chen, and Danny Demelfi

1. Project Overview

1.1 Background and Objectives

It's no secret that everybody loves grapes. Grapes are one of the earliest cultivated and widely distributed fruit tree species in the world. However, grapes like many produce are prone to contracting diseases, making them unavailable for consumption. Separating "good" grapes from inedible grapes is part of a process called "thinning." Thinning is an important agronomic measure for controlling yield, improving grape commercial properties, and enhancing quality. In recent years, manual grape thinning technology has been widely promoted and applied, which can improve the fruit quality of grapes and increase their economic value. However, this has also resulted in high labor costs, and all too frequently a shortage of labor.

To meet the needs of grape industry thinning, based on deep learning algorithms and Machine Learning models, we are developing a grape thinning algorithm. It can intelligently judge the appropriate length of flowering shoots and the size of grape clusters, intelligently recognize the number and position of preserved fruit, and ultimately achieve automated grape thinning. This has significant implications for high-quality grape production.

1.2 Task Description

The mission of this project is to conduct exploratory work on the development of intelligent devices. The main task of this project is to collect grape datasets and train an algorithm model to recognize good and bad grapes.

1.3 Selection of Algorithm Model and Reasoning

At first, we realized that our algorithm would not only need to identify where the grapes are in an image, but also differentiate between good and bad grapes. So, after some discussion, we decided to work with an image segmentation model using a CNN.

CNN refers to Convolutional Neural Network, which is a commonly used deep learning neural network that is mainly used for tasks such as image recognition, object recognition, and classification. The CNN model can extract high-level abstract features representing image features from the input image through operations such as convolution, pooling, and fully connected layers, thereby achieving tasks such as image classification and recognition.

R-CNN, which is the predecessor of Faster R-CNN, stands for Region-based Convolutional Neural Network and was initially proposed to solve the problem of object detection in images. R-CNN first extracts a series of possible regions in the image, then uses CNN to extract features, and finally uses a support vector machine (SVM) classifier for classification to obtain information about whether there is an object and the object category in each region.

We were initially leaning towards using YOLOv5, a famous image recognition model, for our project. However,we soon realized that YOLO would not be a good fit for our problem. YOLO is an incredibly popular and versatile tool, but it is ultimately used for object detection, and not image segmentation. The difference between these two is subtle but important - Object detection can determine *what* is in a certain image - like detecting cars or people in security camera footage. Image segmentation, however, divides the image into several parts, which we can then perform operations on (for example, telling if an individual grape is good or not).

We inevitably decided on using Mask-RCNN, which is a popular, open-source R-CNN model used for object detection and instance segmentation on Keras and TensorFlow.

2. Data Collection and Processing

2.1 Data Collection Methods

We have constructed two datasets: a training and a validation set - each containing good and bad grapes, all of which are images scraped from the internet. Also, for the purpose of simplicity, all grape images were green grapes.

2.2 Data Preprocessing Techniques

As the number of good and bad grapes was limited, we planned to use augmentation techniques to process the training set images. This would have entailed looping through the training set, and ever so slightly changing specific grapes (located through image segmentation) by applying a filter, rotating them, etc. This would have added hundreds of automatically generated images to our dataset, making data collection much easier. However, due to time constraints, we were unable to implement this function.

3. Model Training and Evaluation

3.1 Training and Validation Dataset Split

The dataset was split into the training and validation sets that we manually labeled. The training and validation sets were made using makesense.ai, a free tool for labeling data. We created bounding boxes for each individual grape in a bunch and uploaded it to the project directory in google collab and we split it into 2 classes for grape configuration to be loaded and split.

3.2 Model Training Process

We trained the model using the model.train method built into the Mask R-CNN and used it with the configuration learning rate, and the epochs and layers that we wanted to show for the training results.

3.3 Model Evaluation Metric Selection

Although not implemented, we planned to use Intersection over Union to measure the evaluation of the object detection. IoU measures the overlap between the predicted and actual bounding boxes of the validation set and the result of what the model predicted. The reason it was not implemented is due to time constraints, but this is an excellent metric for comparing predicted versus actual results.

3.4 Model Evaluation Results

The model cannot train in its entirety and the evaluation is hard to measure, so further implementation is needed for this project.

4. Summary and Outlook

4.1 Project Summary

We created two datasets, and designed an R-CNN model algorithm. Unfortunately, due to time constraints, we were unable to realize the true potential of the project. Based on our experimentation, we created a Github repository and developed a web application to showcase our results.

4.2 Improvement and Optimization Suggestions

The datasets need to be further improved. Firstly, the dataset for good and bad grapes has a severe lack of images and needs to be supplemented. This could be helped with the implementation of image augmentation, as previously mentioned. Secondly, the dataset collected only has images of green grapes, making its usability limited. Also, there is just an insufficient amount of training images - what should be an 80/20 split of training and validation images is more like 50/50. In addition, there is still significant room for improvement in the algorithm model. The biggest problems that the project faced were a lack of data and a lack of time, both of which could easily be fixed in the future.

4.3 Prospects for Future Work

The proposal for this project owes its thanks to a group of friends dedicated to high-quality grape cultivation who provided a lot of help. If this project can ultimately be realized, it will bring about a huge change for the cultivation of fresh eating grapes. We are looking forward to future learning and continuing the development work of this project.