Comparative analysis of 2D and 3D vineyard yield prediction system using artificial intelligence

Dhanashree Barbole, Parul M. Jadhav

Dr. Vishwanath Karad MIT World Peace University, Pune, India

Abstract

Traditional techniques for estimating the weight of clusters in a winery, generally consist of manually counting the variety of clusters per vine, and scaling by means of the entire variety of vines. This method can be arduous, and costly, and its accuracy depends on the scale of the sample. To overcome these problems, hybrid approaches of computer vision, deep learning (DL), and machine learning (ML) based vineyard yield prediction systems are proposed. Self-prepared datasets are used for comparative analysis of 2D and 3D yield prediction systems for vineyards. DL-based approach for segmentation operation on an RGB-D image dataset created with the D435I camera is used along with the ML-based weight prediction technique of grape clusters present in the single image using these datasets. A comparative analysis of the DL-based Keras regression model and various ML-based regression models for the weight prediction task is taken into account, and finally, a prediction model is proposed to estimate the yield of the entire vineyard. The analysis shows improved performance with the 3D vineyard yield prediction system compared to the 2D vineyard yield prediction system with grape cluster segmentation pixel accuracy up to 94.81% and yield prediction accuracy up to 99.58%.

Correspondence: Dhanashree Barbole, Research Scholar, Dr. Vishwanath Karad MIT World Peace University, Pune, India.
E-mail: manedhanashree04@gmail.com

Key words: precision agriculture; vineyard; cluster segmentation; yield prediction; deep learning; machine learning.

Contributions: the authors contributed equally.

Conflict of interest: the authors declare no potential conflict of interest

Funding: none.

Availability of data and material: the data that support the findings of proposed studies are openly available in the repository name: GrapesNet: Indian Grape Clusters RGB & RGB-D Image Datasets at the link: https://data.mendeley.com/datasets/mhzmzd5cwx/1 with DOI: 10.17632/mhzmzd5cwx.1

Received: 20 March 2023.
Accepted: 22 July 2023.

Journal of Agricultural Engineering 2024; LV:1545
doi:10.4081/jae.2023.1545

Introduction

The world population is predicted to be 10 billion by the year 2050 which is 35% of today's population (FAO, 2009). The requirement for food will increase by 70% with respect to current food requirements (Ranganathan *et al.,* 2018). Currently, as per the rapid growth of urbanization, there will be huge decrements in land available for farming. As per reports, India will be the most populated country by 2050 (FAO, 2009; Ranganathan *et al.,* 2018) and currently, it is already holding behind in population per food production ratio. There are reasons behind this situation, like lack of knowledge and awareness, uneducated farmers, unpredictable weather conditions, and use of traditional harvesting techniques. The best way to secure the food production ratio of the entire world is precision farming (Abdul Hakkim *et al.,* 2016). The use of advanced tools and techniques for different stages of farming can improve food production rapidly. Many countries are adapting to the precision agriculture culture to prevent soil quality degradation, reduce the use of chemical applications for crop production, improve the quantity and quality of crops, and reduce production costs. One of the excellent natural sources of essential vitamins, minerals and fibers is fruit (Khan *et al.,* 2020). Fruit farming has more economic advantages than vegetable farming. It also provides the essentials to agro-based industries like storage, preservation, packaging, transportation, marking of fresh fruit (Khan *et al.,* 2020) and processing fruit to manufacture various products like cosmetics, eatable products, drinks, *etc*. Therefore, fruit farming is one of the most important and long-standing traditions in most of the countries.

Fruit harvesting is the core of fruit farming, so to make it automated, various researchers have proposed their studies in this domain. In the yield, prediction of any fruit detection and counting is the primary need. Some traditional approaches like thresholding (Fernandez-Maloigne *et al.,* 1993), morphological operations (Baeten *et al.,* 2008), circle Hough Transform (Grasso *et al.,* 1996), filtering (Ceres *et al.,* 1998), edge detection (Ceres *et al.,* 1998), *etc*., were used for fruit detection purpose. There are so many special methods available to extract the region of interest (ROI), which is nothing but fruit from the total image. An easy technique for determining the weight of the proposed fruit is to calculate the area of the fruit in the image and relate it to the real size of the fruit. While this estimation is desired to be automated,

a training and validating platform that demonstrates the applicability with accuracy is necessary. Artificial Intelligence (AI) is a huge domain that includes Machine Learning (ML) field into it. Various ML-based algorithms (Wang *et al.,* 2017; Liu and Whitty, 2015) like Support Vector Machine (SVM), K-Nearest Neighbor (KNN), and K-Mean Clustering (K-mean) are used for the fruit classification task. Advanced image-capturing techniques have been utilized in various research to get information from fruit images. These images make the fruit detection task much easier. Currently, Deep Learning (DL), which is a sub-domain of ML, is very popular for object detection applications. For fruit detection task, various DL models like Convolutional Neural Network (CNN) (Chen *et al*., 2017; Habaragamuwa *et al*., 2018), Field Control Node (Liu *et al*., 2018), Visual Geometry Group-16 (VGG16) (Liu *et al*., 2020; Arad *et al*., 2019; Altaheri *et al.,* 2019; Marani *et al*., 2020), Faster Region-based Convolutional Neural Network (RCNN) (Lee *et al.,* 2020; Stein *et al.,* 2019), Mask RCNN (MRCNN) (Lee *et al.,* 2020; Ni *et al.,* 2020; Santos *et al.,* 2020; Zhang *et al.,* 2022), ResNet (Kang *et al*., 2019), YOLO-versions (Tang *et al.,* 2020; Tang *et al.,* 2023; Zhou *et al.,* 2022) are implemented. Among all available DL-based fruit detection models, MRCNN with ResNet-101 and YOLO versions provide extremely good results (Barbole *et al.,* 2021).

In fruit production businesses, grapes are considered a cash crop. Grapes are used for multiple purposes like fresh eating, for making wines, raisins, jams, jelly, vinegar, *etc*. To determine the sales and profits between merchants and farmers, farmers first need to get an idea about their total production. Fruit detection and segmentation, as well as counting, are the fundamental processes in any automated yield estimating system. In the case of grapes, they are multi-fruit and have high variance in their shapes and sizes. So, counting clusters will not provide the accurate yield of vineyard. This suggests that the prediction of an accurate agricultural yield for vineyards is one of the tough issues in precision agriculture. Yield prediction traditional methods for grapes are dependent on manual approaches which are less efficient, less accurate and time-consuming. To produce an accurate automated yield prediction system, intelligent grape cluster acquisition needs to be performed. Since the crop yield prediction model is based on different variables, which include light conditions, weather, soil, software of fertilizer, and seed range, it necessitates the creation and use of many different datasets.

Few algorithms and techniques are available for grape cluster detection, segmentation and yield estimation but those are not suitable for real-time applications due to shaded regions under a canopy, different illuminance, different color shades of clusters and in-differential occlusions from backgrounds. Liu and Whitty (2015) used a SVM classifier with 88% accuracy supported by color and texture information of grape images for detecting the clusters out of entire images. Nuske *et al.* (2011) applied a berry detection approach using radial symmetry transform for yield prediction of vineyard with a 3-11% error rate. Luo *et al*. (2016) adapted an Ada-Boost-based framework for grape cluster detection with 96.5% accuracy. Along with the main classifier, the authors also used thresholding and morphological operations for noise removal from the outputs to make them more desirable (Luo *et al.,* 2016). Luo *et al.* (2018) proposed a K-mean clustering-based segmentation algorithm which is capable of separating the overlapping grape bunches with 88% accuracy. Badeka *et al.* (2019) utilized KNN classification techniques for the segmentation of red and white grapes with tore local binary patterns related to color and texture properties of images. Badeka *et al.* (2019) achieved segmentation accuracies up to 94% for red grapes and 83% for white grapes. Cecotti *et al.* (2020) experimented transfer learning approach on 11 pre-trained CNN-based models like VGG versions, GoogLeNet, ResNet50, *etc*., for red and white grapes segmentation, and finally concluded that ResNet architecture gives promising results that are up to 99% as compared to others. Santos *et al*. (2020) compared Masked Recurrent CNN (MRCNN), YOLOv2 and YOLOv3 for grape cluster segmentation application on Embrapa Wine Grape Instant Segmentation Dataset with MRCNN having superior F1-score up to 89%. According to Marani *et al*. (2020), VGG16 model gives the best performance of 80.58% accuracy when compared with AlexNet, GoogLeNet, and VGG19. According to Barbole *et al.* (2021), a comparative study of various DL models like MRCNN, Yolov3, and U-Net for grape cluster detection and models have been trained to get segmented images as output. Among all these models, U-Net performs better for grape cluster segmentation tasks. Zhang *et al*. (2022) proposed a real-time red grape cluster detection algorithm with the help of YOLOv5s, which is claimed to be fast and accurate in complex natural scenes.

Most of the references (Liu and Whitty, 2015; Nuske *et al.,* 2011; Luo *et al.,* 2016; Badeka *et al.,* 2019; Cecotti *et al.*, 2020; Zhang *et al.,* 2022; California Historical Society collection, 2012) have considered red grapes for grape cluster detection, segmentation and yield estimation applications. However, these techniques are only suitable for red grape cluster detection and weight prediction. Worldwide red grape production is higher compared to that of white grapes but in countries like India, most of the vineyards have white grapes, where the red grape datasets fail. It can be observed that very few vineyard datasets are available, especially on white grapes, for future research. Hence, there is a need to create more vineyard datasets with white grape clusters. In some studies (Barbole *et al.,* 2021; Santos *et al.,* 2020), the authors presented a grape cluster dataset for the segmentation of grape clusters from complex environments. But only the last rows of vines are considered so that there will be less confusion with the background vines and clusters. Vines with limited grape clusters are taken into consideration, and pruning is also done to remove leaf occlusion on clusters. In the case of Indian vineyards, there are a large number of clusters per vine. So, this approach in all the above-mentioned references is suitable only for vineyards with small and limited clusters, not in the Indian scenario. By considering all the drawbacks of current techniques, the development of an RGB-D grape cluster dataset is performed in this proposed work which consists of RGB images as well as depth images of grape clusters. A whole new approach to grape cluster weight prediction (2D and 3D) and their comparative studies are presented in this paper. The proposed approach is the combination of DL for segmentation tasks and ML for regression-based weight prediction of cluster tasks.

Materials and Methods

Materials

GrapesNet (Barbole *et al.,* 2023) dataset from Mendeley data is used in the proposed work. This dataset consists of a total of 11,000+ images of grape clusters from Indian vineyards. GrapesNet includes four different types of sub-dataset and all of them are considered for the proposed work. The GrapesNet (Barbole *et al.,* 2023) contains the RGB and depth images, as shown in Table 1.

The dataset considered in the proposed work contains grape cluster images with a natural background as well as an artificial

background, which makes it the best choice for the proposed research. The technique of transfer learning has been adapted in the proposed work. In GrapesNet (Barbole *et al*., 2023) dataset, each image includes various objects inside it as a background like leaves, branches, wires, poles and so many others like soil, old leaves, grass, drip irrigation pipes, *etc*. (Figure 1). To increase the number of images in the dataset, Barbole *et al.* (2023) have already performed data augmentation on the original datasets.

When real-time application is the goal, the model has to be trained and tested on several datasets. In the proposed work, GrapesNet dataset has been used to fulfill this need. To create and develop a more generalized model, real background is studied along with that various factors affecting image acquisition have been taken into consideration in GrapesNet dataset. In the used dataset (Figure 2), images are taken at different daytime slots to cover illuminance effects, with different camera angles and with different blockages like leaves, branches and other bunches (Barbole *et al.,* 2023).

Methods

This work consists of two sections, the upper section is a 2D vineyard yield prediction system and the lower section is a 3D vineyard yield prediction (Figure 3). A comparative study of two sections has been considered in this proposed work. For both 2D and 3D vineyard yield prediction systems, there are two main stages which are grape cluster segmentation and weight prediction of cluster. Hybrid approach for yield prediction of vineyard has been proposed with a DL-based model for grape cluster segmentation and DL, ML-based model for weight prediction of clusters. In 2D systems, RGB images and their masks are given as input to the modified U-Net (Barbole *et al.,* 2021) grape cluster segmentation model, while for 3D systems, RGB images undergo the new proposed process of unwanted region removal from images using depth information of corresponding images, called pre-masking process. These pre-masked RGB images along their masks are given as inputs. Segmented outputs are given separately to the two weight prediction approaches: DL-based approach and ML-based approach. In the DL-based approach, keras regression model (Barbole *et al.,* 2022) has been implemented which accepts segmented output from the modified U-Net, weight (kg) per image and average distance (cm) between camera and clusters from the images. With only 3 inputs, keras regression model will predict weight of grape clusters per image. In ML-based approaches, regression models like Linear Regressor (LR), Ridge Regressor (RR), Bayesian Ridge Regressor, Decision Tree Regressor (DTR) and Random Forest Regressor (RFR) are used to predict the weight of clusters. For these ML-based models, some estimated feature vectors from segmented images are used as inputs and, as a result, each model predicts the weight of grape clusters per image. Finally, by combining all the models together, a comparative study of 2D and 3D vineyard yield prediction systems is performed in order to conclude the results.

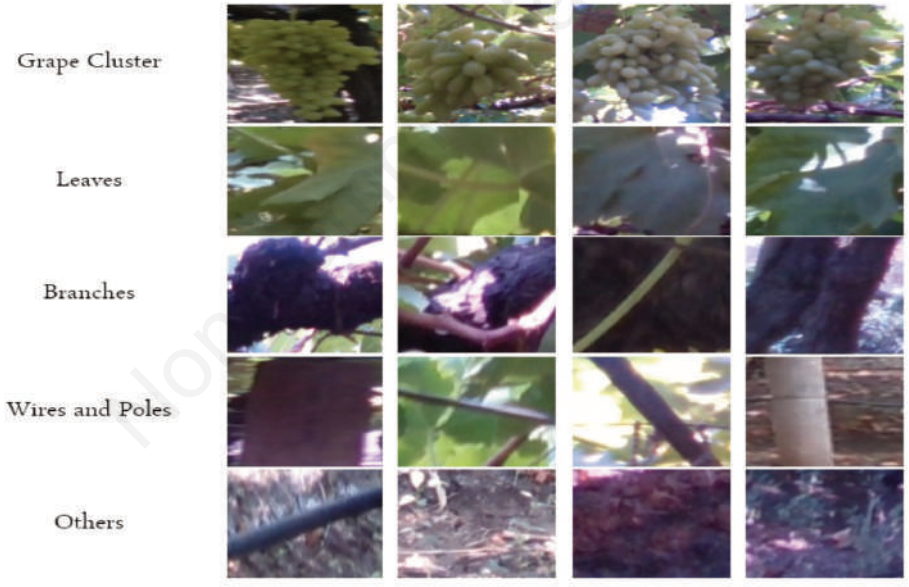

Figure 1. Object samples included in images in GrapesNet (Barbole *et al.,* 2023) dataset. 1st row indicates the grape cluster with color variation, and the remaining rows indicate objects from the background like leaves, branches, wires, poles, and others.

Table 1. GrapesNet (Barbole *et al*., 2023) dataset for proposed model training and testing.

Dataset	Image types	Total images		Image resolution	
		RGB	Depth	RGB	Depth
1	RGB	4305	-	500p×500p	-
2	RGB	2960	-	500p×500p	-
3	RGB & Depth	1696	424	500p×500p	424p×240p
4	RGB & Depth	2100	350	500p×500p	424p×240p

Grape cluster segmentation model

The dataset consists of RGB and RGB-D images of vineyards. The first task in this proposed approach is to separate the grape clusters from the background. Here, the modified U-Net is used to perform the grape cluster segmentation task. In the modified U-Net, the depth of the U shape has been increased by adding two additional layers: one in the encryption section and another in the decryption section. In the encryption section, some locality features are sacrificed to obtain higher-level features that aid in object detection. The output of the first layer is up-sampled using an additional up-sampling layer to preserve locality features in the image. Similarly, in the decryption section, object features are compro-

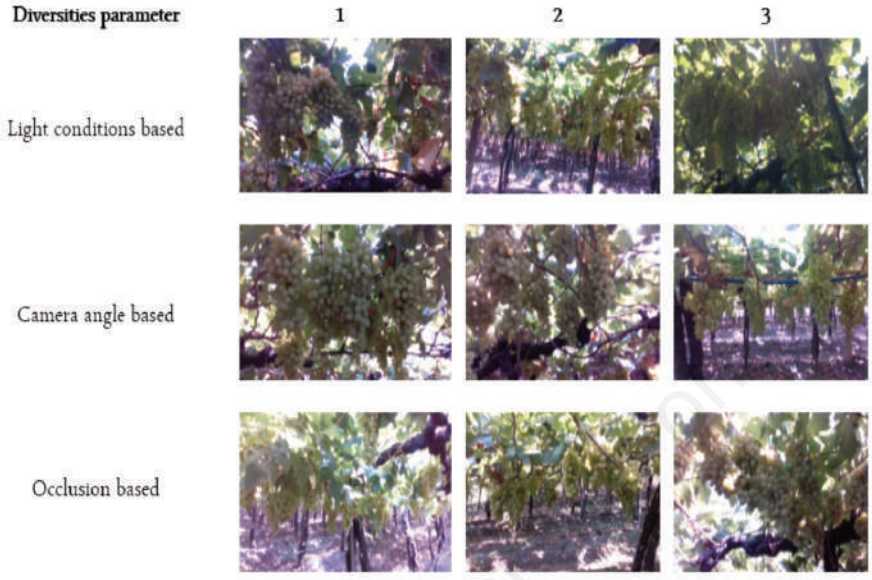

Figure 2. Diversities in GrapesNet (Barbole *et al*., 2023) dataset. 1st, 2nd and 3rd rows indicate diversities due to different light conditions (shelter, sidelight, back-light), different camera angles (front, elevated, dropped) and different occlusions (leaves, branches, other grape clusters) respectively.

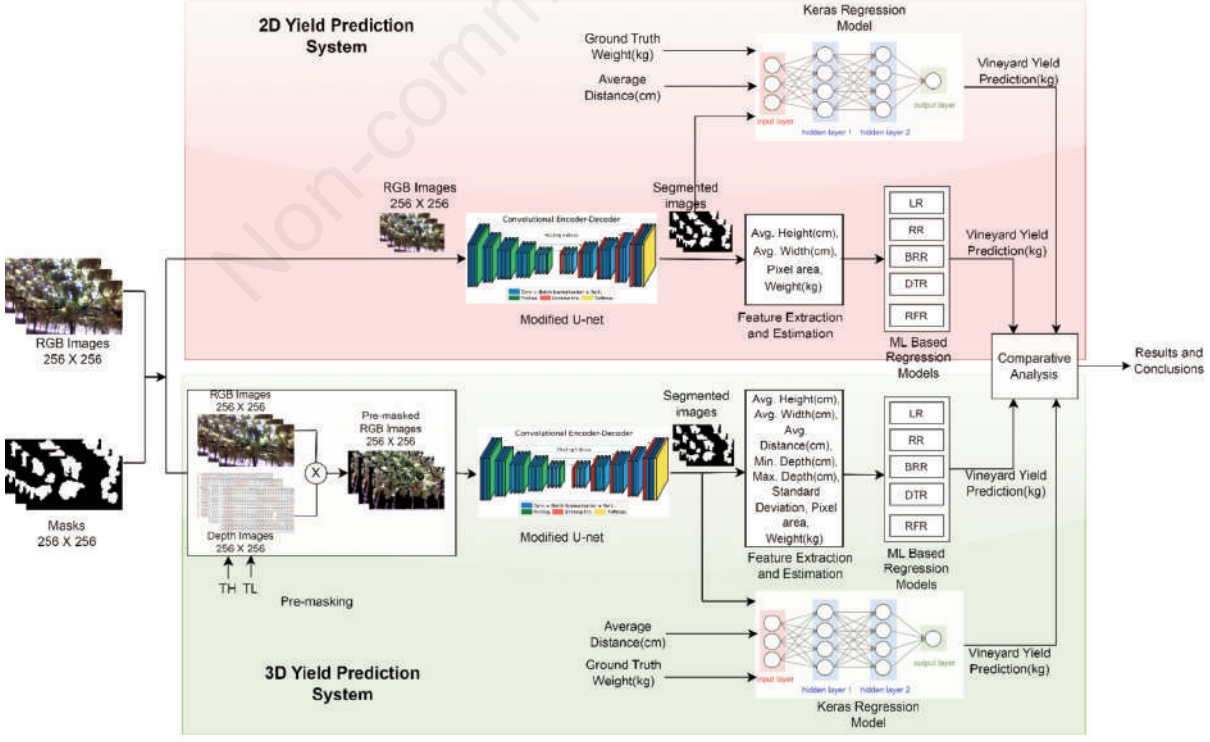

Figure 3. Core diagram of the proposed vineyard yield prediction systems. Upper part: a 2D yield prediction system; bottom part: a 3D yield prediction system.

mised to retain the location information of the same object within the image. An extra down-sampling layer is introduced just before the output layer in the decryption section. This layer takes two inputs: one from the previous layer and another from a skip connection through an additional up-sampling layer. By including this additional down-sampling layer, object features are preserved in the image. These two additional layers enhance the model's performance compared to the original U-net model. Since high-resolution images are provided as input, improved results are achieved (Barbole *et al.,* 2021). For this task, both 2D and 3D models are trained on Dataset 2 with single grape clusters per image, and through transfer learning, the same trained models are again trained on a dataset with multiple grape clusters per image.

2D grape cluster segmentation model

Dataset 2 contains RGB images of a single grape cluster per image. Masks of each image in dataset 2 are generated with the help of masking tools. RGB images and their masks are given as input to the modified U-Net model (Barbole *et al*., 2021). In this model, the depth of U of the original U-Net model has been increased by adding an additional up-sampling layer at the input side and a down-sampling layer at the output end. An increase in the resolution has magnified the features and shown improvement in the output segmentation results. The modified U-Net segmentation model gives the binary images as an output, which are segmented output images with separated grape clusters in white color and background in black color (Figure 4). These segmented images are given as input to the next 2D weight prediction model. The same trained segmentation model is trained on dataset 1 with multiple grape clusters per image and tested on dataset 3, which contains slot-wise images of vineyards.

3D grape cluster segmentation model

As per 2D grape cluster segmentation outputs, it is observed that: i) unwanted grape clusters from the background and other vines are also getting segmented, which is undesirable; ii) the training process of the grape cluster segmentation model is time-consuming as the data size is larger. This may strongly degrade the output results and lead to a complex time system. To solve that, depth information obtained through a depth camera can be used to mask the unwanted pixels that do not satisfy the distance requirement. This facilitates the masking of clusters that give rise to ambiguity during DL-based segmentation. There are two possible methods for unwanted region masking: using a generated depth image or utilizing raw distance information captured during image acquisition. The latter approach, involving raw distance units, is considered the most effective method for masking regions that do not meet the requirements for DL-based segmentation. The process of removing unwanted regions from the images using raw information of the respective images is called as "pre-masking process".

RGB images are multiplied with raw images to generate pre-masked images. For masking unwanted regions from the image, there is a requirement of two thresholds: the low threshold value (TL) and the high threshold value (TH). Pre-masking consists of three main blocks: the TL-TH range decider, the TL value decider, and the TH value decider. Masks of RGB images and raw images are given as input to the TL-TH range decider, which will find out minimum (min.) and maximum (max.) lower/upper threshold values given as TL_{min}, TL_{max}, TH_{min} and TH_{max}. TL_{min} and TL_{max} values are given to the TL value decider block where TH= TH_{max}. Similarly, TH_{min} and TH_{max} values are given to the TH value decider block where TL= TL_{min}. TL value decider and TH value decider will finally find out that TL and TH values are based on some mathematical calculations.

Low threshold-high threshold range decider

The aim of this block is to come up with appropriate TL-TH values for pre-masking without affecting the ROI, which are the masks of those images. So, masks are taken as input along with raw images.

RGB masks will be converted into binary masks, which means they will have values only of 0 or 1 (Figure 5). Raw images consist of depth values of each pixel present in the entire RGB image. To find the depth information for ROIs, all binary masks are multiplied with the corresponding raw images. As an output, new depth/raw images of masks are estimated. From each new raw image, min. TL and TH values as well as max. TL and TH values are extracted in the TL and TH columns. The range of TL-TH is estimated as:

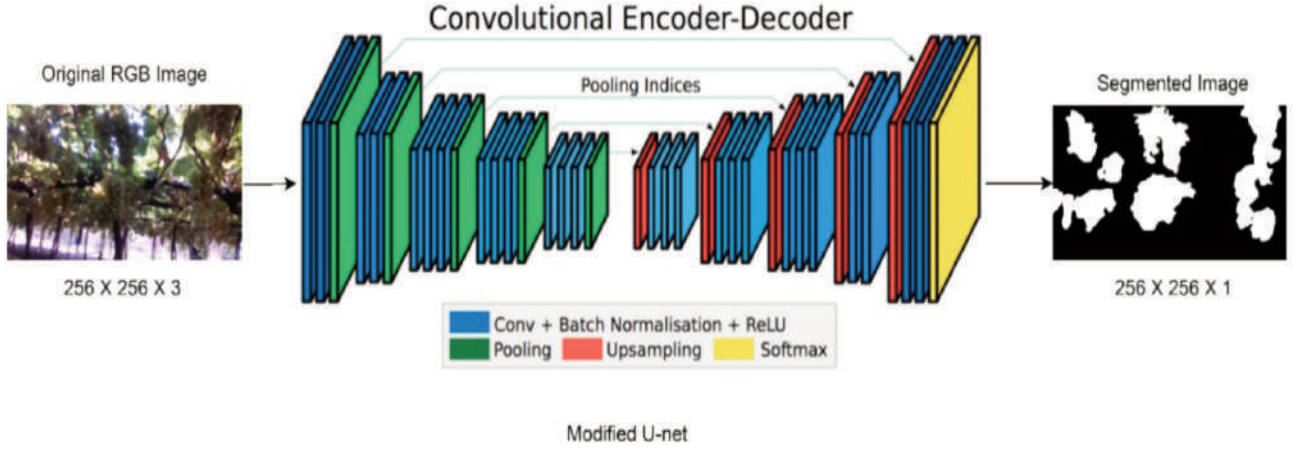

Figure 4. 2D grape cluster segmentation model.

TL_{min} = min (TL) (Eq. 1)

TL_{max} = max (TL) (Eq. 2)

TH_{min} = min (TH) (Eq. 3)

TH_{max} = max (TH) (Eq. 4)

Low threshold value decider

RGB masks and their corresponding raw images are given as inputs to the TL value decider (Figure 6). TH is kept constant with a TH=TH (max) value, and the TL is varied from TL (min) to TL (max), which are nothing but 169 and 548 respectively. X1 is the original RGB mask and X2 is an estimated RGB mask. The parameter estimation block takes the average of the intersection over union (IOU) scores, and the average of the exception scores of the X1 and X2 which are mathematically expressed as:

$$\text{Average IOU score} = \frac{\sum_{i=1}^{N} \frac{(X1 \cap X2)}{(X1 \cup X2)}}{N} \quad \text{(Eq. 5)}$$

$$\text{Average exception score} = \frac{\sum_{i=1}^{N} \frac{(X1 \cap X2)^{C}}{(X1 \cup X2)}}{N} \quad \text{(Eq. 6)}$$

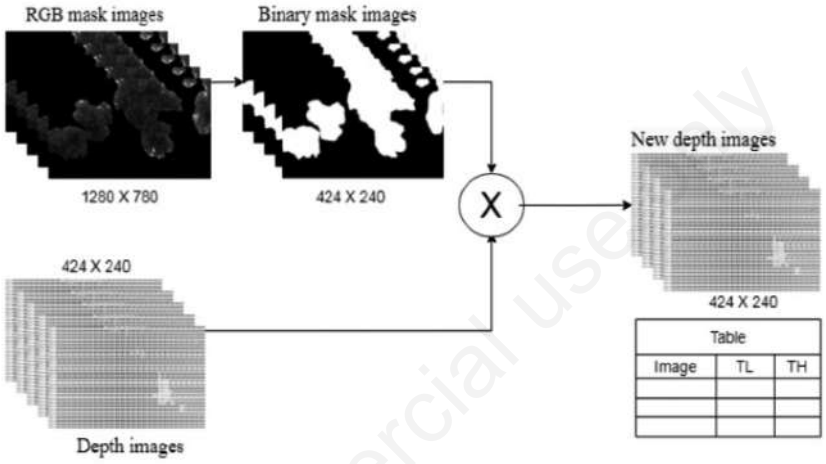

Figure 5. Low threshold-high threshold range decider. TL is the lower value of threshold and TH is the higher value of threshold.

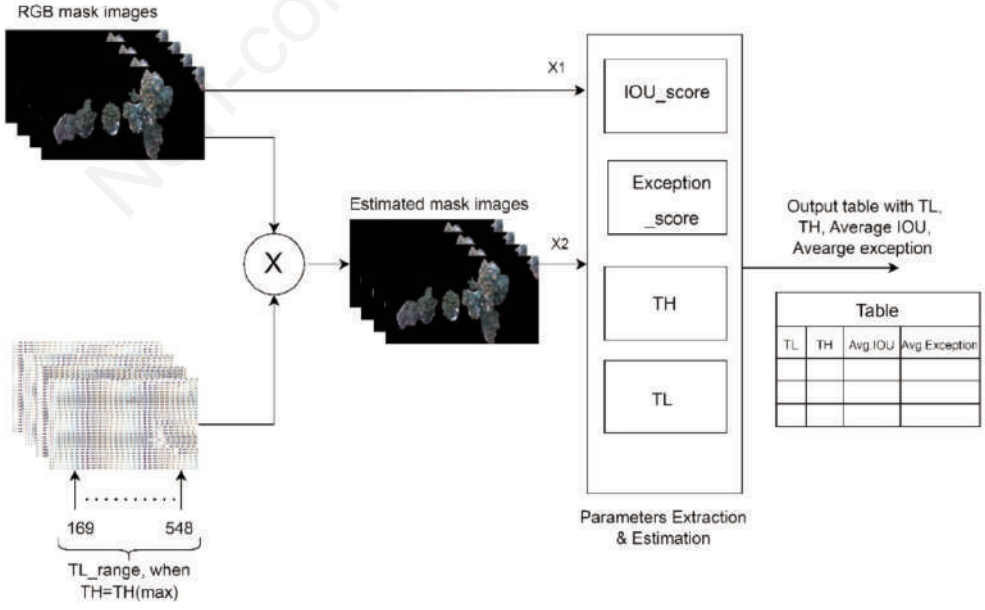

Figure 6. Block diagram of TL value decider where TL varies from 169 to 548 and TH is taken as maximum value of TH that is TH(max). TL is the lower value of threshold and TH is the higher value of threshold. X1 is the RGB mask and X2 is an estimated mask, and both are given to parameter extraction and estimation block which produces a table containing TL, TH, average intersection over union and average exception. TH, high threshold; TL, low threshold; IOU, intersection over union.

For finding the TL value, let us assume that x is the TL value, y1 is the average exception score, and y2 is the average IOU score. So, to find the mathematical relationship of x with y1 and y2, polynomial regression is performed. The degree with the min. mean squared error (MSE) is selected as the final degree of the polynomial equation. For the TL value, degree =11. So, equation for y1 and y2 in terms of x becomes:

$$y1 = \beta_0 + \beta_1.x + \beta_2.x^2 + \beta_3.x^3 + \ldots\ldots\ldots + \beta_{11}.x^{11} + C_1 \quad \text{(Eq. 7)}$$

$$y2 = \beta_0 + \beta_1.x + \beta_2.x^2 + \beta_3.x^3 + \ldots\ldots\ldots + \beta_{11}.x^{11} + C_2 \quad \text{(Eq. 8)}$$

Here $\beta_1, \beta_2 \ldots \beta_{11}$ are the slope coefficients, β_0 is intercept (constant term) and C_1, C_2 are the model's error terms, which are estimated from the polynomial curve of polynomial regressions. So, after putting the values of the slope coefficients, intercept, value of y1=0 in equation (7), and x will be estimated. Similarly, by putting the value of slope coefficients, intercept, and estimated value of x from equation (7) in equation (8), the value of y2 will be estimated. The max. TL value with a 100% average IOU score and a 0% average exception score is the final TL value of TL value decider block.

High threshold value decider

RGB masks and their corresponding raw images are given as inputs to the TH value decider (Figure 7). TL is kept constant with TL=TL (min) value, and TH is reduced from TL (max) to 2000. X1 is the original RGB mask and X2 is an estimated RGB mask. The parameter estimation block takes the average of IOU scores and the average of exception scores of the X1 and X2, which are mathematically expressed in equations (5) and (6) respectively.

As original masks are created manually, there is some acceptable human error in the exception score, which is considered as σ and expressed as:

$$\sigma = \frac{\sum_{i=1}^{N} \frac{(X1 \cap X1')^c}{(X1 \cup X1')}}{N} \quad \text{(Eq. 9)}$$

where X1 = original RGB masks, *X1'* = revised RGB masks, N = total number of images.

After solving equation (9), the estimated value of σ = 3.611. Similar to the TL value decider, for finding the TH value, let us assume that x is the TH-value, y1 is average exception score, and y2 is average IOU score. To find the mathematical relation of the x with the y1 and y2, polynomial regression is performed. The degree with the min. MSE is selected as the final degree of polynomial equations. For the TH value, degree =5. So, the equation for the y1 and y2 in terms of the x becomes:

$$y1 = \beta_0 + \beta_1.x + \beta_2.x^2 + \beta_3.x^3 + \beta_4.x^4 + \beta_5.x^5 + C_1 \quad \text{(Eq. 10)}$$

$$y2 = \beta_0 + \beta_1.x + \beta_2.x^2 + \beta_3.x^3 + \beta_4.x^4 + \beta_5.x^5 + C_2 \quad \text{(Eq. 11)}$$

Here $\beta_1, \beta_2 \ldots \beta_5$ are the slope coefficients, is intercept (constant term) and C_1, C_2 are the model's error terms, which are estimated from the polynomial curve of polynomial regressions. After putting the values of slope coefficients, intercept, and the value of the y1=σ in equation (10), x will be estimated. Similarly, by putting the values of the slope coefficients, intercept, and estimated value of x from equation (10) in equation (11), the value of the y2 will be estimated.

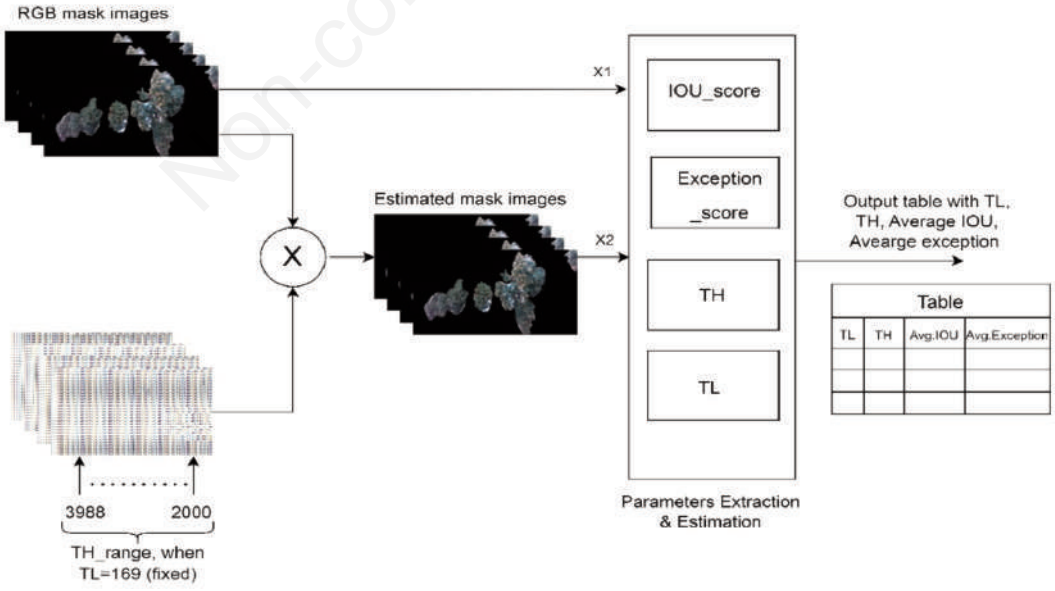

Figure 7. Block diagram of the TH value decider where TH varies from range 3988 to 2000 and TL is a fixed value that is 169. TL is the lower value of threshold and TH is the higher value of threshold. X1 is the RGB mask and X2 is an estimated mask, and both are given to parameter extraction and estimation block which produces a table containing TL, TH, average intersection over union and average exception. TH, high threshold; TL, low threshold; IOU, intersection over union.

Figure 8 shows the original RGB image and its pre-masked image with TL and TH thresholding ranges. In the 3D grape cluster segmentation, instead of giving the original RGB image to the proposed segmentation model, pre-masked images with unwanted regions of the image removal are given to the proposed segmentation model (Figure 9). The addition of a pre-masking block to the proposed model has shown an interesting improvement in the final segmentation results.

Weight prediction model

As mentioned above, Dataset 2 is created with a single grape cluster per image and with fixed distances. The image of the same grape cluster is taken from seven different distances, and meanwhile, height, width, and weight of the same cluster were noted. So here, segmented output images of dataset 2 are given to the 2D weight prediction model. The pixel area of the ROI, which is the grape cluster area of that image is calculated. As mentioned, the segmented image contains only black and white pixels, where white pixels indicate the grape clusters and black pixels indicate the background. Here, the white pixel area is the ROI. Using the appropriate command in Python, the ROI of all images has been estimated. Both 2D and 3D weight prediction models are trained with trainable parameters extracted from the images. For comparison purposes, the segmented images from the above segmentation model are given to the ML and DL-based weight prediction model (Barbole *et al.,* 2022). Keras regression model (Barbole *et al.,* 2022) is used as a DL-based weight prediction of vineyard. Keras regression model (Barbole *et al.,* 2022) is trained and tested with segmented output from grape cluster segmentation model with additional input of weight (kg) for each image. For ML-based systems, the pixel area of each segmented image is estimated, which is also ROI. This pixel area of each cluster will be added to a .csv file, which also contains the actual height and width of corresponding clusters. After that, the entire dataset is divided into four parts: i) X_train: training features; ii) y_train: training labels; iii) X_test: testing features; iv) y_test: testing labels. In our case, 90% of the total dataset is used for training purposes and the remaining 10% is used for testing purposes. The ML regression model is trained with the single grape cluster per image dataset, and predictions for the test dataset are made. Finally, on the segmented output of multiple grape clusters per image dataset, the same models are retrained and tested to get the weight of grape clusters per image. The major difference between prediction and classification is that prediction gives any numeric value as output, whereas classification gives the class to which an object belongs. In our case, the area of interest is predicting the weight of the grape clusters by consid-

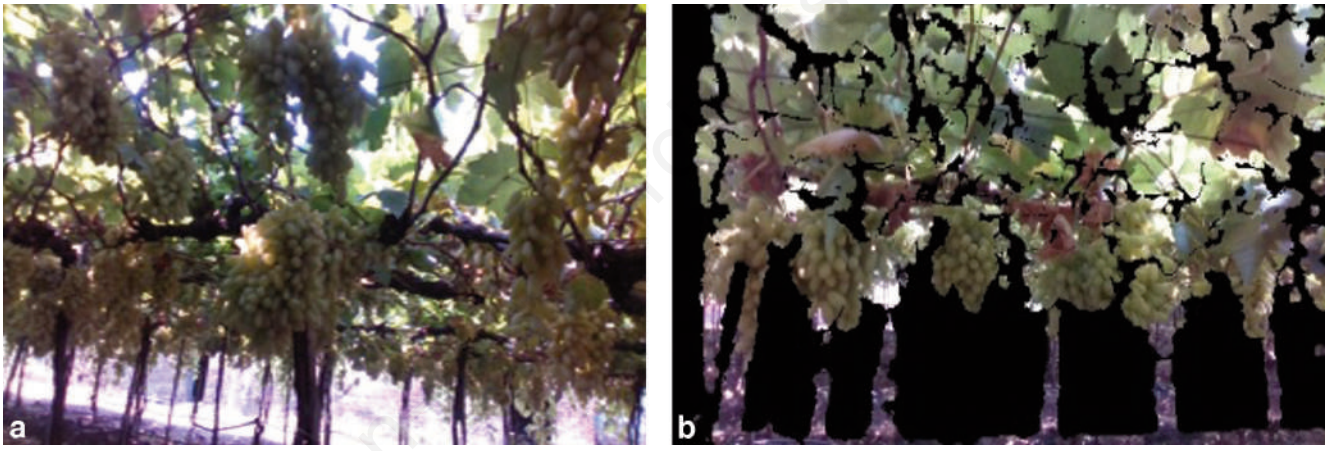

Figure 8. Pre-masking output. **a**) Original RGB image; **b**) pre-masked RGB image.

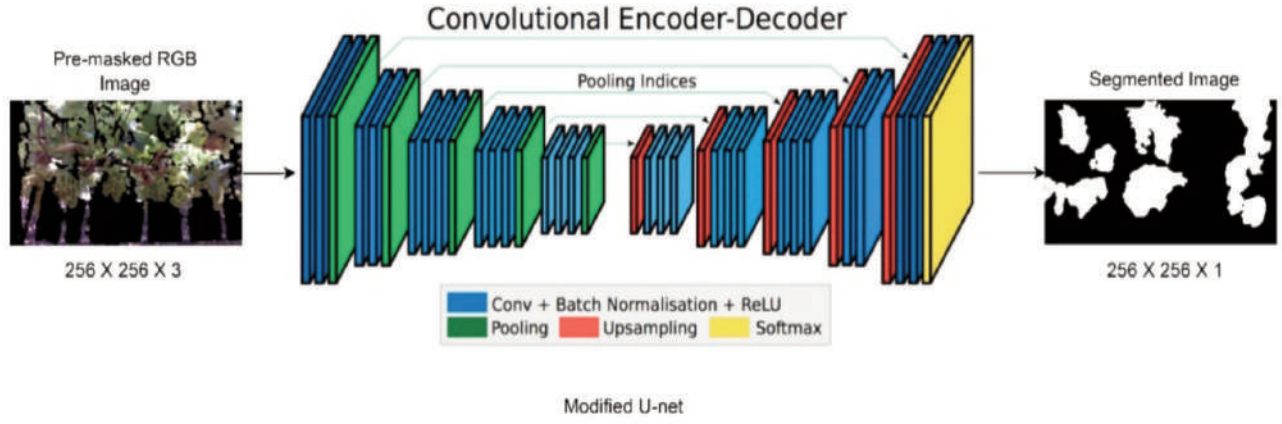

Figure 9. 3D grape cluster segmentation model.

ering other parameters. So, ML regression models are best suited for performing the desired task. Based on the literature survey, five ML models and their comparative studies are considered for the analysis. These considered techniques are as follows: i) LR; ii) RR; iii) BRR; iv) DTR; v) RFR. To design this model, Python language is preferred. In Python, there is a Sci-kit Learn library which contains all the ML models.

2D weight prediction model

As mentioned above, pre-trained weight prediction models for single grape clusters per image are trained on segmented output images of dataset 1. All ML-based 2D weight prediction models are trained with images taken from an average distance of 75cm, with height (cm), width (cm), and pixel area as trainable parameters. These trained models are again trained for segmented output from the modified U-Net for multiple grape clusters per image. A list of feature vectors inside the .csv file is mentioned in Table 2. Here average height (cm) and average width (cm) are estimated by taking the average of the heights and widths of all images in dataset 2. Finally, these trained 2D weight prediction models are tested on dataset 3, where the weights of grape clusters present in each image were noted as a ground truth.

3D weight prediction model

As mentioned above, pre-trained weight prediction models for single grape clusters per image are trained on segmented output images of dataset 1. All ML-based 3D weight prediction models are trained first on dataset 2, which has a single grape cluster per image and a variety of distances. For training the weight prediction models, a .csv file containing the various features has been considered related to the cluster images (Table 3).

Along with height (cm), width (cm), average distance (cm), and pixel area, for each image, some more estimated features like min. depth, max. depth, and standard deviation are also examined as trainable parameters. By analyzing dataset 2, the average height and width of each cluster in the image have been estimated. For min. and max. depth estimation, first multiplying RGB image with the raw image is taken, and then by using max. and min. functions in the Numpy library, min. depth and max. depth are estimated. In a similar way, with .std () and .mean () functions in Numpy, standard deviation and average distance (cm) are estimated. Finally, all trained 3D weight prediction models are tested on dataset 3, where weights of the grape clusters present in each image were noted as a ground truth. As mentioned earlier, dataset 3 is created by selecting specific areas of vineyard that are 10.219 m^2. Once all trained ML and DL-based weight prediction models are tested on dataset 3, the weights of grape clusters in each image are estimated. Finally, it is given to a yield prediction model, which predicts the yield from the yields of the specified areas.

Evaluation parameters

Model evaluation is the main task to determine how reliably any model performs. By providing some important performance parameters, it makes the model more presentable to the audience. In this section, performance evaluation parameters of all models of yield prediction systems are mentioned.

Grape cluster segmentation model

According to some literature surveys (Marani *et al.*, 2020; Santos *et al.*, 2020; Tang *et al.*, 2020; Tang *et al.*, 2023; Wang *et al.*, 2017; Zhang *et al.*, 2022), the best performance evaluation parameters for segmentation task are Pixel Accuracy (PA) and mean IOU (mIOU). Details of these parameters are given below.

Pixel accuracy

For the segmentation task, the accuracy of correctly classified pixels will be the performance evaluation parameter. Mathematically, it is expressed as:

$$Accuracy = \frac{Correctly\ classified\ Pixels}{Total\ number\ of\ pixels} \times 100 \quad \text{(Eq. 12)}$$

Table 2. 2D trainable feature parameters estimated for each image in the dataset.

Parameter	Information	Type
Average height (cm)	Estimated height of the grape cluster, measured in centimeters	Numeric
Average width (cm)	Estimated width of the grape cluster, measured in centimeters	Numeric
Pixel area	Pixel count obtained from segmented images, which is the region of interest	Numeric
Average distance (cm)	Actual distance of the grape cluster from the camera, measured in centimeters.	Numeric
Weight (kg)	The actual weight of the grape cluster, in kilograms.	Numeric

Table 3. 3D trainable feature parameters estimated for each image in the dataset

Parameter	Information	Type
Average height (cm)	Estimated height of the grape cluster, measured in centimeters	Numeric
Average width(cm)	Estimated width of the grape cluster, measured in centimeters	Numeric
Pixel count (area)	Pixel count obtained from segmented images, which is the region of interest	Numeric
Min. depth	The minimum value of depth obtained from depth information from the D435I Camera.	Numeric
Max. depth	The maximum value of depth obtained from depth information from the D435I Camera.	Numeric
Standard deviation	Standard deviation of the .raw files of the corresponding images	Numeric
Average distance (cm)	Actual distance of the grape cluster from the camera, measured in centimeters	Numeric
Weight (kg)	The actual weight of the grape cluster, in kilograms	Numeric

Higher PA leads to better model performance and makes a more suitable model for real-time applications.

Mean intersection over union

IOU is estimated by dividing the overlapping pixel area between the actual mask and the predicted mask by the combined pixel area of the actual and predicted mask. An average of IOU for each image is nothing but mIOU, which is mathematically expressed as:

$$mIOU = \frac{\sum_{1}^{N}\left(\frac{Area_{actual} \cap Area_{predicted}}{Area_{actual} \cup Area_{predicted}}\right)}{N} \quad \text{(Eq. 13)}$$

where N is the total number of images tested. This value ranges from 0 to 1 and the model that provides a value closer to 1 is considered the best model for segmentation task.

Weight prediction model

There are so many performance evaluation metrics present, but very few are suitable to be used for regression problems. Three performance metrics in this study are given below.

R-squared score

In regression models, the R-square value, also known as the coefficient of determination, is a statistical measure that represents the proportion of the total variation in the dependent variable that can be explained by the independent variables in the model.

The R-square value ranges between 0 and 1, where:

- An R-square value of 0 indicates that the independent variables in the model cannot explain any of the variations in the dependent variable.
- An R-square value of 1 indicates that the independent variables in the model can perfectly explain all of the variations in the dependent variable.

Mean squared error

MSE is a common metric used to evaluate the performance of regression models. MSE measures the average squared difference between the predicted values and the actual values of the dependent variable in a regression model. By squaring the differences, it penalizes larger errors more heavily, providing a measure of the overall model accuracy. A lower MSE value indicates better performance, as it means the model's predicted values are closer to the actual values.

Root mean squared error

Root MSE (RMSE) is a popular metric used to evaluate the performance of regression models. RMSE is derived from the MSE and provides a measure of the average magnitude of the errors between the predicted and actual values of the dependent variable in a regression model. Similar to MSE, a lower RMSE value indicates better model performance, as it signifies smaller errors between the predicted and actual values.

Vineyard yield error

To evaluate the performance of the yield prediction model, the error between actual yield and predicted yield is considered. The lower the error value, the better will be the model performance. It is simply the difference between actual vineyard weight (kg) and predicted vineyard weight (kg).

Results and Discussion

The result analysis of all the 2D and 3D models included in yield prediction systems is discussed in this section. The results of the pre-masking on dataset include total images, image resolution and dataset size (Table 4). Here, the original dataset contains 424 images, with each image having a resolution of 424p×240p. The size of the original dataset was 86.063 MB, and after the pre-masking operation, it was reduced to 71.38 MB.

All the 2D and 3D grape cluster segmentation models are trained on a GPU system with 403 RGB images as the training dataset and 412,845 trainable parameters (Table 5). Specifically, for 500 epochs, the 3D model achieved an accuracy of up to 92.02% and a c of up to 81.69%. The time complexity for this model was approximately 558 minutes. Similarly, for 1000 epochs, the 3D model demonstrated even better results, with an accuracy of up to 94.81% and an excellent mIOU of up to 86.13%. The time complexity for this model was 1293 minutes. As per comparative results of all the 2D DL- and ML-based grape cluster weight pre-

Table 4. Results of the pre-masking on dataset which includes total images, image resolution and dataset size.

Model	Total images	Image resolution	Dataset size
Original dataset	424	424p×240p	86.063 MB
Pre-masked dataset	424	424p×240p	71.38 MB

Table 5. Comparative results of grape cluster segmentation models by hyper parameter (number of epochs) tuning and in terms of performance evaluation parameters like time complexity, pixel accuracy and mean intersection over union.

Model	Number of epochs	Training dataset	Trainable parameters	Time complexity	PA (%)	mIOU (%)
2D grape cluster segmentation model	500	403	412,845	692m	88.50	80.21
	1000	403	412,845	1476m	90.23	81.95
3D grape cluster segmentation model	500	403	412,845	558m	92.02	83.69
	1000	403	412,845	1293m	94.81	86.13

PA, pixel accuracy; mIOU, mean intersection over union.

diction regression models (Table 6), the decision tree regression model and the random forest regression model are performing much better, as they give 100.0 and 99.9311 R2-scores respectively, for the train dataset, and 68.6723 and 70.2908 R2-scores respectively, for the test dataset. LR model has the highest MSE and the RMSE which is up to 0.0298 and 0.0298 respectively, for the train dataset, and with the test dataset, 0.2078 for both the models. DTR and RFR models are performing better, with the lowest MSE values up to 0.0 and 0.00027 respectively, for the train dataset, and for the test dataset, they are 0.1768 and 0.1677 respectively. Similarly, the RMSE values of the DTR and RFR models are lower, up to 0.0 and 0.01645 respectively, for the train dataset, and 0.4205 and 0.4045 respectively, for the test dataset.

Similar to the results of 2D weight prediction models, from comparative results of all the 3D DL- and ML-based grape cluster weight prediction models (Table 7), it can be stated that the decision tree regression model and the random forest regression model are performing better, as it gives 100.0 and 99.9311 R2-scores, respectively for the train dataset, and 68.0075 and 71.6766 R2-scores respectively for the test dataset. DTR and RFR models are performing better, with lower MSE values up to 0.0 and 0.00061 respectively, for the train dataset, and for the test dataset, it is 0.1806 and 0.1599 respectively. Similarly, the RMSE values of the DTR and the RFR models are lower, up to 0.0 and 0.02477 respectively, for the train dataset, and 0.4250 and 0.3999 respectively, for test dataset.

In the average weight of three slots using all the 2D, 3D DL- and ML-based weight prediction models (Table 8), it can be observed that LR and the RR models perform better, with accuracy values up to 97.1654%, for both the models with a 2D dataset, and 99.3356% and 99.3350% respectively, for a 3D dataset. From this table, it can be said that the average weight of three slots is estimated very well with 3D weight prediction models rather than 2D weight prediction models. From a table of comparative results of

Table 6. Comparative results of all the 2D deep learning- and machine learning-based grape cluster weight prediction regression models and in terms of performance evaluation parameters like R2_score, mean squared error and root mean squared error.

Approach	Model	R2_score (%)		MSE		RMSE	
		Train	Test	Train	Test	Train	Test
DL-based	2D Keras regression model (Barbole *et al*., 2022)	98.67	48.17	0.050	0.9811	0.0704	0.9905
ML-based	LR	92.42	63.20	0.02982	0.2078	0.1726	0.4558
	RR	92.42	63.20	0.02982	0.2078	0.1726	0.4558
	BRR	92.42	63.20	0.02982	0.2078	0.1726	0.4558
	DTR	100.0	68.67	0.0	0.1768	0.0	0.4205
	RFR	99.93	70.29	0.00027	0.1677	0.01645	0.4095

MSE, mean squared error; RMSE, root mean squared error; DL, deep learning; ML, machine learning; LR, Linear Regressor; RR, Ridge Regressor; BRR, Bayesian Ridge Regressor; DTR, Decision Tree Regressor; RFR, Random Forest Regressor.

Table 7. Comparative results of all the 3D deep learning- and machine learning-based grape cluster weight prediction regression models and in terms of performance evaluation parameters like R2_score, mean squared error and root mean squared error.

Approach	Model	R2_score (%)		MSE		RMSE	
		Train	Test	Train	Test	Train	Test
DL-based	3D Keras regression model (Barbole et al., 2022)	99.22	51.13	0.0031	0.8420	0.0555	0.9176
ML-based	LR	92.48	66.59	0.02958	0.1886	0.1719	0.4343
	RR	92.84	66.59	0.02813	0.1886	0.1677	0.4343
	BRR	93.01	63.95	0.02778	0.2035	0.1666	0.4512
	DTR	100.0	68.01	0.0	0.1806	0.0	0.4250
	RFR	99.84	71.68	0.00061	0.1599	0.02477	0.3999

MSE, mean squared error; RMSE, root mean squared error; DL, deep learning; ML, machine learning; LR, Linear Regressor; RR, Ridge Regressor; BRR, Bayesian Ridge Regressor; DTR, Decision Tree Regressor; RFR, Random Forest Regressor.

Table 8. Average weight of three slots using all the 2D, 3D deep learning- and machine learning-based weight prediction models by relating actual weight (kg) for grape clusters in each slot with predicted weight (kg) for the same respective slots with performance evaluation factors like accuracy and error.

Approach	Weight prediction model	Actual weight (kg)	Predicted weight (kg)		Error		Accuracy (%)	
			2D	3D	2D	3D	2D	3D
DL-based	Keras regression model (Barbole *et al.*, 2022)	33.8824	29.6363	30.2967	4.2461	3.5857	87.4681	89.4172
ML-based	LR	33.8824	32.9220	33.6573	0.9604	0.2251	97.1654	99.3356
	RR	33.8824	32.9220	33.6571	0.9604	0.2253	97.1654	99.3350
	BRR	33.8824	32.9220	33.0522	0.9604	0.8302	97.1654	97.5497
	DTR	33.8824	32.4046	32.4333	1.4778	1.4491	95.6384	95.7231
	RFR	33.8824	32.1783	32.2354	1.7041	1.6470	94.9705	95.1390

DL, deep learning; ML, machine learning; LR, Linear Regressor; RR, Ridge Regressor; BRR, Bayesian Ridge Regressor; DTR, Decision Tree Regressor; RFR, Random Forest Regressor.

Table 9. Comparative results of deep learning- and all machine learning-based entire yield prediction models by relating actual weight (kg) for entire vineyard with predicted weight (kg) for the same vineyard with respect to performance evaluation parameters like accuracy and error.

Approach	Yield prediction model	Actual weight (kg)	Predicted weight (kg)		Error		Accuracy (%)	
			2D	3D	2D	3D	2D	3D
Ground truth estimated		13384.4279	13417.8944	13417.8944	-33.4665	-33.4665	-	-
DL-based	Keras regression model (Barbole *et al.*, 2022)	13384.4279	11736.3612	11997.8845	1648.066	1386.543	87.68	89.64
ML-based	LR	13384.4279	13037.5385	13328.7374	346.8894	55.6905	97.41	99.58
	RR	13384.4279	13037.5385	13328.6853	346.8894	55.7426	97.41	99.58
	BRR	13384.4279	13037.5385	13089.1007	346.8894	295.3272	97.41	97.79
	DTR	13384.4279	12834.6466	12844.0189	549.7813	540.4090	95.89	95.96
	RFR	13384.4279	12743.0422	12765.6347	641.3857	618.7931	95.20	95.38

DL, deep learning; ML, machine learning; LR, Linear Regressor; RR, Ridge Regressor; BRR, Bayesian Ridge Regressor; DTR, Decision Tree Regressor; RFR, Random Forest Regressor.

DL-based and all ML-based yield prediction models (Table 9), one can say that the 3D LR and RR models are giving the best results with the highest accuracy value, compared to other models, which are up to 99.58% for both the models.

Conclusions

The correct weight prediction of grape clusters using automation is the need of the time. The image processing-based approach with least complexity is a challenging task. The important factor that affects the prediction performance is the distance variation during the capture of images using the camera. The image-to-image distance variation and keeping track of these changes using a manual approach are not practical. The depth information obtained from the use of a depth camera is the best possible solution for general applications. Depth information from a depth camera is used in this paper to predict the weight of the grape clusters. The regression task is performed by using a calibration approach with single cluster images taken at different distances. The known distance, their respective depth information of ROI, standard deviation, and pixel count from segmented images have relationships, which are regulated by considering L1 and L2 parameters in regression models. R2_score greater than 0.5 is considered a good score, which indicates that 50% of the dependent variable variance is explained by the model. From this, it can be stated that all models considered in the proposed work are performing well, and all of them have an R2_score greater than 60% for train and test datasets. The weight prediction with the 3D DTR and the 3D RFR gives better output compared to the other 2D and 3D ML-based weight prediction models, but when slot-wise average weight prediction is considered, the 3D LR and the 3D RR models perform better. Some parameter tuning also affects the results of ML models in positive ways. The maximum error of ±1% is seen while predicting the weight of the clusters. At the final task of vineyard yield prediction, again the 3D LR and the 3D RR models give the best results with minimum error values compared to other models which are up to 55.6905 kg and 55.7426 kg. The accuracies of 3D LR and RR models are up to 99.58% for both, which is remarkable. From all the comparative analyses of 2D and 3D yield prediction systems, it can be concluded that 3D yield prediction gives superior results with additional parameters estimated from the depth information.

Limitations and future scopes

In the proposed research, we have created a vineyard dataset for only one type of grape (sonaka) due to lack of time. By following the steps and methodology used in this proposal for creating a vineyard dataset, future researchers can create more datasets on a variety of grape types in India.

Moreover, while training a DL-based model, it needs images and their masks as inputs. This masking is done manually, which is a very hectic and time-consuming process. So future researchers should work on automated masking techniques for vineyard images.

Finally, the currently trained segmentation model is trained only for a single type of grape (sonaka), so by training the same model using the concept of transfer learning for multiple grape varieties, it can become more versatile and suitable for real-time scenarios.

References

Abdul Hakkim V.M., Abhilash Joseph E., Ajay Gokul A.J., Mufeedha K. 2016. Precision farming: The future of Indian agriculture. J. Appl. Biol. Biotechnol. 4:68-72.

Altaheri H., Alsulaiman M., Muhammad G. 2019. Date fruit classification for robotic harvesting in a natural environment using deep learning. IEEE Access. 7:117115-33.

Arad B., Kurtser P., Barnea E., Ha B., Edan Y., Ben-Shahar O. 2019. Controlled Lighting and Illumination-Independent Target Detection for Real-Time Cost-Efficient Applications. The Case Study of Sweet Pepper Robotic Harvesting. Sensors. 19:1390.

Badeka E., Kalabokas T., Tziridis K., Nicolaou A., Vrochidou E., Mavridou E., Papakostas G.A., Pachidis T. 2019. Grapes Visual Segmentation for Harvesting Robots Using Local Texture Descriptors. Springer Link Computer Vision Systems. 98-109.

Baeten J., Donn K., Boedrij S., Beckers W. 2008. Autonomous fruit picking machine: A robotic apple harvester. Field and Service Robotics. 42:531-9.

Barbole D.K., Jadhav P.M. 2021. Comparative Analysis of Deep Learning Architectures for Grape Cluster Instance Segmentation. Inf. Technol. Industry. 9.

Barbole D.K., Jadhav P.M. 2022. Grape Yield Prediction using Deep Learning Regression Model. 2022 International Conference for Advancement in Technology (ICONAT), Goa,

India, 1-6.
Barbole D.K., Jadhav P.M. 2023a. GrapesNet: Indian Grape Clusters RGB & RGB-D Image Datasets. Mendeley Data.
Barbole D.K., Jadhav P.M. 2023b. GrapesNet: Indian RGB & RGB-D vineyard image datasets for deep learning applications. Data Brief. 48:109100.
Barbole D., Jadhav P., Patil S. 2021. A Review on Fruit Detection and Segmentation Techniques in Agricultural Field. International Conference on Image Processing and Capsule Networks. 300:269-88.
California Historical Society Collection. 2012. Close-up of a grape cluster on a vine. University of Southern California (USC) Digital Library.
Cecotti H., Rivera A., Farhadloo M., Pedroza M.A. 2020. Grape detection with convolutional neural networks. Expert Syst. Appl. 159.
Ceres R., Pons J.L., Jiménez A.R., Martín J.M. 1998. Agribot: A robot for aided fruit harvesting. Industrial Robot. 5.
Chen S., Shivakumar S., Dcunha S., Das J., Okon E., Qu C., Kumar V. 2017. Counting apples and oranges with deep learning: A data-driven approach. IEEE Robot. Autom. Lett. 2:781-8.
Fernandez-Maloigne C., Laugier D., Boscolo C. 1993. Detection of apples with texture analyses for an apple picker robot. Proceedings of the Intelligent Vehicles '93 Symposium, IEEE Xplore. pp. 323-8.
FAO. 2009. Global agriculture towards 2050. High-level Expert Forum. Available at the link: https://www.fao.org/wsfs/forum 2050/wsfs-forum/en/
Grasso G., Recce M. 1996. Scene analysis for an orange picking robot. Proceeding in International Conference of Computer Technology in Agriculture (ICCTA '96).
Habaragamuwa H., Ogawa Y., Suzuki T., Shiigi T., Ono M., Kondo N. 2018. Detecting greenhouse strawberries (mature and immature), using deep convolutional neural network. Eng. Agric. Environ. Food. 11:127-38.
Kang H., Chen C. 2019a. Fruit Detection, Segmentation and 3D Visualization of Environment in Apple Orchards. Comput. Electron. Agric. 171:105302.
Kang H., Chen C. 2019b. Fruit Detection and Segmentation for Apple Harvesting Using Visual Sensor in Orchards. Sensors. 19:4599.
Khan N., Fahad S., Naushad M., Faisal S. 2020. Grape Production Critical Review in the World. SSRN Electron. J.
Lee J., Nazki H., Baek J., Hong Y., Lee, M. 2020. Artificial Intelligence Approach for Tomato Detection and Mass estimation in Precision Agriculture. Sustainability 12:9138.
Liu S., Whitty M. 2015. Automatic grape bunch detection in vineyards with an SVM classifier. J. Appl. Log. 13:643-53.
Liu X., Chen S.W., Aditya S., Sivakumar N., Dcunha S., Qu C., Taylor C.J., Das J., Kumar V. 2018. Robust Fruit Counting: Combining Deep Learning, Tracking, and Structure from Motion. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).
Liu Z., Wu J., Fu L., Maje Y., Feng Y., Li R., Cui Y. 2020. Improved Kiwifruit Detection Using Pre-Trained VGG16 With RGB and NIR Information Fusion. IEEE Access. 8.
Luo L., Tang Y., Lu Q., Chen X., Zhang P., Zou X. 2018. A vision methodology for harvesting robot to detect cutting points on peduncles of double overlapping grape clusters in a vineyard. Comput. Ind. 99:130-9.
Luo L., Tang Y., Zou X., Wang C., Zhang P., Feng W. 2016. Robust Grape Cluster Detection in a Vineyard by Combining the AdaBoost Framework and Multiple Color Components. Sensors. 16.
Marani R., Milella A., Petitti A., Reina G. 2020. Deep neural networks for grape bunch segmentation in natural images from a consumer-grade camera. J. Prec. Agric. 22:387-413.
Ni X., Li C., Jiang H., Takeda F. 2020. Deep learning image segmentation and extraction of blueberry fruit traits associated with harvest ability and yield. Horticul. Res. 7.
Nuske S., Achar S., Bates T., Narasimhan S. 2011. Yield estimation in vineyards by visual grape detection. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems.
Ranganathan J., Waite R., Searchinger T., Hanson C. 2018. How to Sustainably Feed 10 Billion People by 2050. World Resources Institute. Available from: https://www.wri.org/insights/how-sustainably-feed-10-billion-people-2050-21-charts
Santos T.T., De-Souza L.L., Dos-Santos A., Avila S. 2020. Grape detection, segmentation and tracking using deep neural networks and three-dimensional association. Comput. Electron. Agric. 170.
Stein M., Bargoti S., Underwood J. 2019. Image Based Mango Fruit Detection, Localization and Yield Estimation using Multiple View Geometry. Sensors. 16:1915
Tang Y., Chen M., Wang C., Luo L., Li J., Lian G., Zou X. 2020. Recognition and Localization Methods for Vision-Based Fruit Picking Robots: A Review. Frontiers Plant Sci. 11.
Tang Y., Zhou H., Wang H., Zhang Y. 2023. Fruit detection and positioning technology for a Camellia oleifera C. Abel orchard based on improved YOLOv4-tiny model and binocular stereo vision. Exp. Syst. Appl. 211.
Wang C., Tang Y., Zou X., SiTu W., Feng W. 2017. A Robust Fruit Image Segmentation Algorithm against Varying Illumination for Vision System of Fruit Harvesting Robot. Optik - Int. J. Light Electron. Opt. 131:626-31.
Zhang C., Ding H., Shi Q., Wang Y. 2022. Grape Cluster Real-Time Detection in Complex Natural Scenes Based on YOLOv5s Deep Learning Network. Agriculture. 12.
Zhou Y., Tang Y., Zou X., Wu M., Tang W., Meng F., Zhang Y., Kang H. 2022. Adaptive Active Positioning of Camellia oleifera Fruit Picking Points: Classical Image Processing and YOLOv7 Fusion Algorithm. Appl. Sci. 12:12959.