Comparative Analysis of 2D and 3D Vineyard Yield Prediction System Using Artificial Intelligence (AI)

Ms. Dhanashree Barbole[1*], Dr. Parul M. Jadhav[2]

[1] *Research Scholar, Dr. Vishwanath Karad MIT World Peace University, Pune*
[2] *Dr. Vishwanath Karad MIT World Peace University, Pune*

Email: manedhanashree04@gmail.com

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, costly, and its accuracy is dependent on the scale of the sample. To overcome these problems, hybrid approaches of Computer Vision (CV), 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. Three different datasets have been created with specific strategies, and are used for different stages of the proposed system. 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 is employed 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 a complete 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 upto 94.81% and yield prediction accuracy upto 99.58%.

Keywords: Precision agriculture, Vineyard, Cluster segmentation, Yield prediction, Deep learning, Machine learning, etc.

1. INTRODUCTION

The world population is predicted to be 10 billion by the year 2050 which is 35% of the today's population [1]. Requirement of food will increase by 70% with respect to current food requirement [2]. Currently, as per rapid growth of urbanization, there will be huge decrements in land available for farming. As per reports, India will be most populated country by 2050 [1, 2] and currently it is already holding behind in population per food production ratio. There are reasons behind this situation are lack of knowledge and awareness, uneducated farmers, unpredictable weather conditions and use of traditional harvesting techniques [3]. Best way to secure food production ratio of entire world is precision farming [4]. Use of advance tools and techniques for different stages of farming can improve the food production rapidly. Many countries are adapting to the precision agriculture culture to prevent soil quality degradation, to reduce use of chemical application for crop production, to improve quantity and quality of crops and to reduce the production cost.

One of the excellent natural source of essential vitamins, minerals and fiber are fruits [5]. Fruit farming has more economic advantages than vegetable farming. It's also provides the essentials to the agro-based industries like storage, preservation, packaging, transportation, marking of the fresh fruits [5] and, processing fruits to manufacture various products from the fruits 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 core of fruit farming, so to make it automated various researchers have proposed their research in this domain. In the yield prediction of any fruit detection and counting is the prime need. Some traditional approaches like thresholding [6, 7, 8, 9, 10, 11], morphological operations [12], circular Hough transform [13, 14, 15], filtering [16], edge detection [16] 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 fruit that is proposed is to calculate the area of the fruit in the image and relate that area 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 huge domain which includes Machine Learning (ML) field into it. Various ML based algorithms

[17, 18, 19, 20, 21] like Support Vector Machine (SVM), K-Nearest Neighbor (KNN), K-Mean Clustering (K-mean) are used for the fruit classification task. Advance image capturing techniques [22, 23, 24] have been utilized in various research to get information from the fruit images. These images makes the fruit detection task much easier. Currently, Deep Learning (DL) which is sub-domain of ML is very popular for object detection applications. For fruit detection task, various deep learning models like CNN [25, 26, 27], FCN [22, 28], VGG16 [29, 30, 31, 32, 33], Faster RCNN [23, 34, 24], MRCNN [34, 35, 36, 37, 38, 39, 40], ResNet [35, 41, 42, 40], YOLO-versions [40, 43, 44, 45] are implemented. Among all available DL based fruit detection models, MRCNN with ResNet-101 and YOLO versions are providing extremely good results [17].

In fruit production businesses, grapes are considered as cash crop. Grapes are used for multi-purposes like fresh eating, for making wines, raisins, jams, jelly, vinegar etc. For determining the sales and profits between merchants and farmers, farmers firstly need to get an idea about their total production. Core steps in automated yield estimation of any fruit yard are: fruit detection/segmentation and counting them. In case of grapes, grapes 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 vineyard is one of the tough issues in the precision agriculture. Yield prediction traditional methods for grapes are dependent on the manual approaches which are less efficient, less accuracy and time consuming. To produce accurate automated yield prediction system, intelligent grape cluster acquisition needed to be perform. Since the crop yield prediction model is based on different variables, which includes light condition, weather, soil, software of fertilizer, and seed range [3], it necessitates the creation and use of many different datasets.

Few algorithms and techniques are available for grape cluster detection, segmentation and its yield estimation but those are not suitable for real-time applications due to shaded region under canopy, different illuminance, different color shades of clusters and in-differential occlusions from background. Author Liu et al. [19] used Support Vector Machine (SVM) classifier with 88% accuracy supported by color and texture information of grape images for detecting the clusters out of entire images. Researcher Nuske et al. [46] applied berry detection approach using radial symmetry transform for yield prediction of vineyard with 3-11% of error rate. Lufeng et al [47] adapted Ada-Boost based framework for grape cluster detection with 96.5% of accuracy. Along with main classifier, authors also used thresholding and morphological operations for noise removal from the outputs to make it more desirable [47]. Author Lue et al. [48] proposed k- mean clustering based segmentation algorithm which is capable of separating the overlapping grape bunches with 88% accuracy. Badeka et al. [49] utilized k-Nearest Neighbor classification techniques for segmentation of red and white grapes with tore local binary patterns (LBP) related to color and texture properties of images. Badeka et al. [49] achieved segmentation accuracies up to 94% for red grapes and 83% for white grapes. Cecotti et al. [50] 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 is up to 99% as compare to others. Santosa et al. [36] compared Masked Recurrent CNN (MRCNN), YOLOv2 and YOLOv3 for grape cluster segmentation application on Embrapa Wine Grape Instant Segmentation Dataset (WGISD) with MRCNN having superior F1-score up to 89%. According to Marani et al. [33], VGG16 model gives best performance that is 80.58% accuracy when compared with AlexNet, GoogLeNet and VGG19. According to Barbole et al. [40], a comparative study of various deep learning 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. [37] proposed 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 [19, 46, 47, 49, 50, 37, 51] have considered red grapes for grape cluster detection, segmentation and yield estimation applications. But these techniques are only suitable for red grape cluster detection and weight prediction. World-wide red grapes production is more as compare to white grapes but in countries like India, most of the vineyards have white grapes, where the red grape datasets fails. It can be observed that very few vineyard datasets are available, especially on white grapes, for the future researchers. Hence, there is a need to create more vineyard datasets with white grape clusters. In some of the papers [40, 36, 33], the authors presented a grape cluster dataset for segmentation of grape clusters

from complex environment. But here 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 above mentioned references are 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 approaches of grape cluster weight prediction (2D and 3D) and their comparative studies are presented in this paper. The proposed approaches are the combination of deep learning for segmentation tasks and machine learning for regression-based weight prediction of cluster tasks.

2. MATERIALS AND METHODS

2.1 Materials

GrapesNet [52, 53] dataset from Mendeley data is used in the proposed work. This dataset consists of total 11000+ 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 [52] contains the RGB and depth images as shown in table 1. Dataset considered in the proposed work contains grape cluster images with natural background as well as artificial background, which makes it best choice for proposed research. Technique of transfer learning has been adapted in the proposed work.

TABLE 1. GRAPESNET [52] DATASET

Data-set	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

Figure 1. Object samples included in images of GrapesNet [53] dataset.

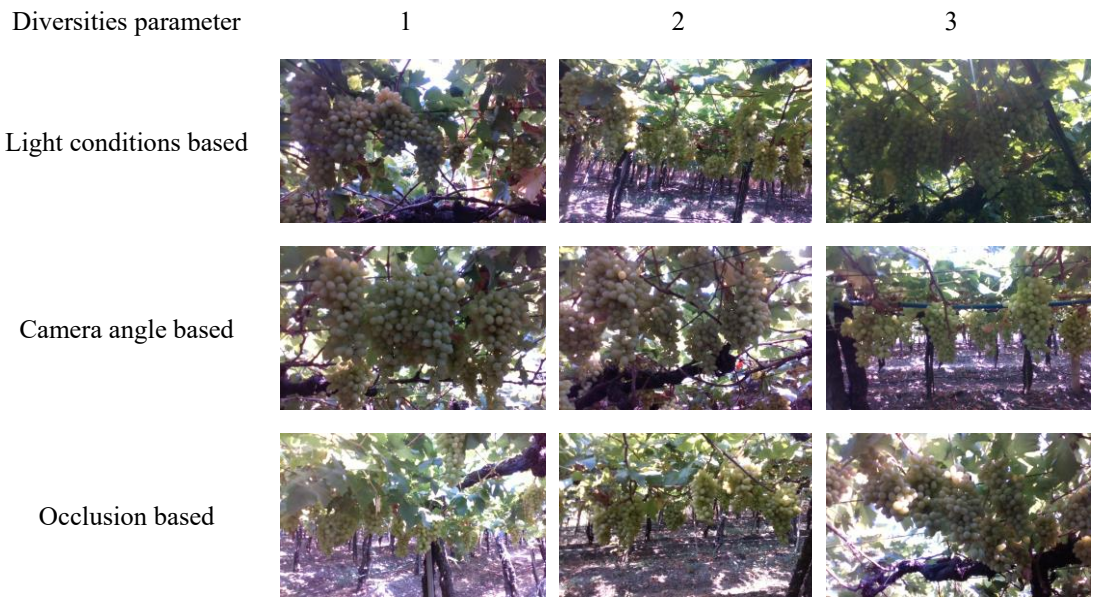

Figure 2. Diversities in GrapesNet [53] Dataset.

As shown in Fig. 1, in GrapesNet [53] dataset, each image includes various object inside it as a background like leaves, branches, wires, poles and so many others(soil, old leaves, grass, drip irrigation pipes etc). To increase number of images in dataset, Barbole et al. [52, 53] have already performed data augmentation on the original datasets.

When real-time application is the goal, model has to be trained and tested on versatile datasets. In proposed work, GrapesNet dataset has been used which fulfill this need. To create and develop model more generalized, real background is studied along with that various factors affecting images acquisition have been taken into consideration in GrapesNet dataset. As shown in Fig. 2, images are taken at different day time slots to cover illuminance effects, with different camera angles and with different blockages like leaves, branches and other bunches [53].

2.2 Methods

As shown in Fig. 3, proposed work consists of two sections, upper section is 2D vineyard yield prediction system and lower section is 3D vineyard yield prediction. Comparative study of two sections is been considered in this proposed work. For both 2D and 3D vineyard yield prediction system, there are two main stages which are grape cluster segmentation and weight prediction of cluster. For 2D system, the RGB images along with its masks and for 3D system, RGB-D images which contains RGB as well as depth information along with its masks are given as input to deep learning model. In both the systems second stage contains machine learning based regression models like Linear Regressor (LR), Ridge Regressor (RR), Bayesian Ridge Regressor (BRR), Decision Tree Regressor (DTR) and Random Forest Regressor (RFR) are used to predict weight of clusters.

For 2D weights prediction, features are extracted from RGB images and some are noted at the time of dataset creation. For 3D weight prediction, noted features are used along with some estimated features of the images from the depth information of those same images. Finally, comparative study of 2D and 3D vineyard yield prediction systems is performed in order to conclude the results.

2.2.1 Grape cluster segmentation model

The data set consists of RGB and RGB-D images of vineyards which are created with Intel Real-Sense D435I camera. The first task in this proposed approach is to separate the grape clusters from the background. The modified U-net is used for performing the grape cluster segmentation task. In modified U-net, depth of U shape has been increased by adding two layers one at the

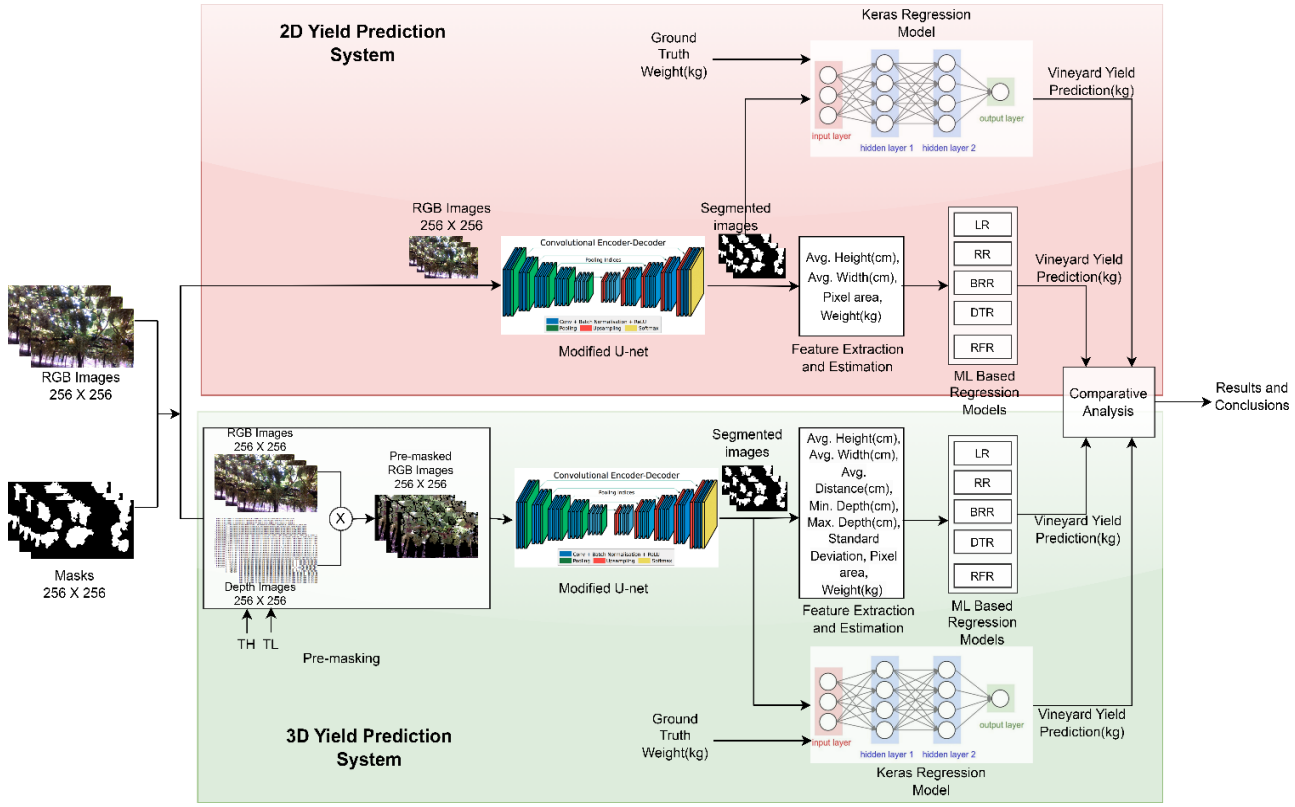

Figure 3. Core diagram of the proposed vineyard yield prediction system

encryption section and one at the decryption section. At the encryption section, basically some locality features are compromised in exchange for higher level features responsible for object detection. Output of the 1st layer is up-sampled with additional up-sampling layer, hence locality features can be preserved in to the image. Similarly at the decryption section, object features are compromised to preserve location information of same object inside image.

Additional down-sampling layer is added just before output layer in decryption section. This layer accepts two inputs, one from its previous layer and another from skip connection through additional up-sampling layer. Due to additional down-sampling layer, object features are preserved in to the image. These additional two layers upgrades the performance of model compared to the original U-net model. As the high resolution images are received at the input side, better results are achieved [40]. 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.

2.2.1.1 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 [40] model. 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. As shown in Fig. 4 below, the modified U-Net segmentation model gives the binary images as an output, which are segmented output images

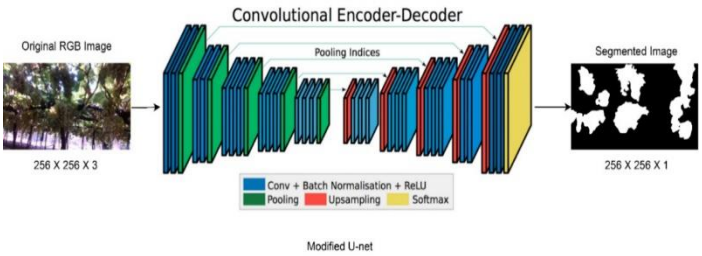

Figure 4. 2D grape cluster segmentation model

With separated grape clusters in white color and background in black color. 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.

2.2.1.2 3D grape cluster segmentation model

As per 2D grape cluster segmentation outputs, it is observed that: 1) unwanted grape clusters from the background and other vines are also getting segmented, which is undesirable 2) 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. So solve that the 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.

The unwanted region masking can be done with the use of a generated depth image or using raw information recorded at the time of taking the image. Thus raw distance units are the best possible way which can be used to mask the regions that do not satisfy the requirement for deep learning based segmentation.

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 (T_L) and the high threshold value (T_H).

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 and maximum 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 TL and TH value based on some mathematical calculations.

(A) TL-TH Range Decider

Here, the aim is to come up with appropriate TL-TH values for pre-masking without affecting the region of interest (ROI), which are the masks of those images. So here, masks are taken as input along with raw images.

As shown in Fig. 5, RGB masks will be converted into binary masks, which means they will have values only of 0 or 1. 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, minimum TL and TH values as well as maximum TL and TH values are extracted in the TL and TH columns. The range of TL-TH is estimated as:

TL_{min} = min (TL) ……. (Eq. 1)
TL_{max} = max (TL) …… (Eq. 2)
TH_{min} = min (TH) …… (Eq. 3)
TH_{max} = max (TH) …… (Eq. 4)

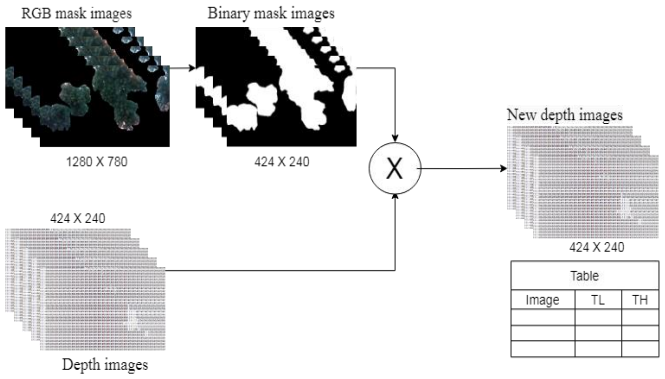

Figure 5. TL-TH Range Decider

(B) TL value Decider

As shown is the Fig. 6, RGB masks and their corresponding raw images are given as an input to the TL value decider. 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:

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

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

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 minimum mean squared error (MSE) is selected as the final degree of the polynomial equation. For the TL value, degree = 11.

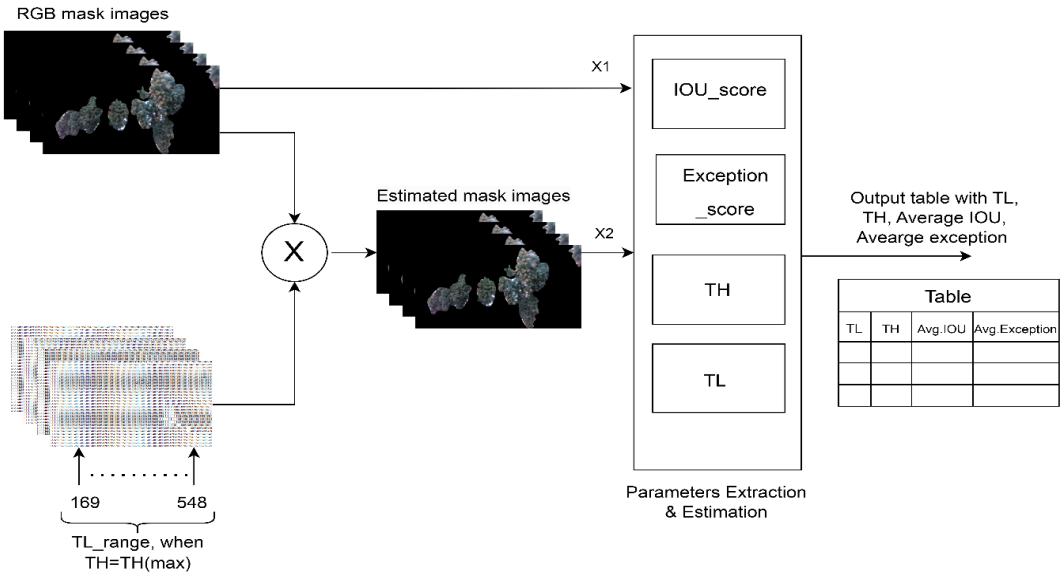

Figure 6. Block diagram of TL value decider

So, equation for y1 and y2 in terms of x becomes:

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

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

Here $\beta_1, \beta_2 \ldots \beta_{11}$ are the slope coefficients, β_0 is intercept (constant term) and C_1, C_2 are 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, estimated value of x from equation (7) in equation (8), the value of y2 will be estimated.

The maximum TL value with a 100% average IOU score and a 0% average exception score is final TL value of TL value decider block.

(C) TH value Decider

Here, as shown in fig. 7, RGB masks and their corresponding raw images are given as an input to the TH value decider. TL is kept constant with TL=TL(min) value, and TH is reducing 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 intersection over union (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:

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

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 minimum mean squared error (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:

$$y1 = \beta_0+\beta_1.x +\beta_2.x^2+\ldots..+\beta_5.x^5 + C_1 \text{ ..... (Eq. 10)}$$

$$y2 = \beta_0 + \beta_1.x +\beta_2.x^2+ \ldots.. +\beta_5.x^5 + C_2 \text{ ….. (Eq. 11)}$$

Here $\beta_1, \beta_2 \ldots \beta_5$ are the slope coefficients, β_0 is intercept (constant term) and C_1, C_2 are model's error terms, which are estimated from the polynomial curve of polynomial regressions. So, 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 the value of x estimated from equation (10) in equation (11),

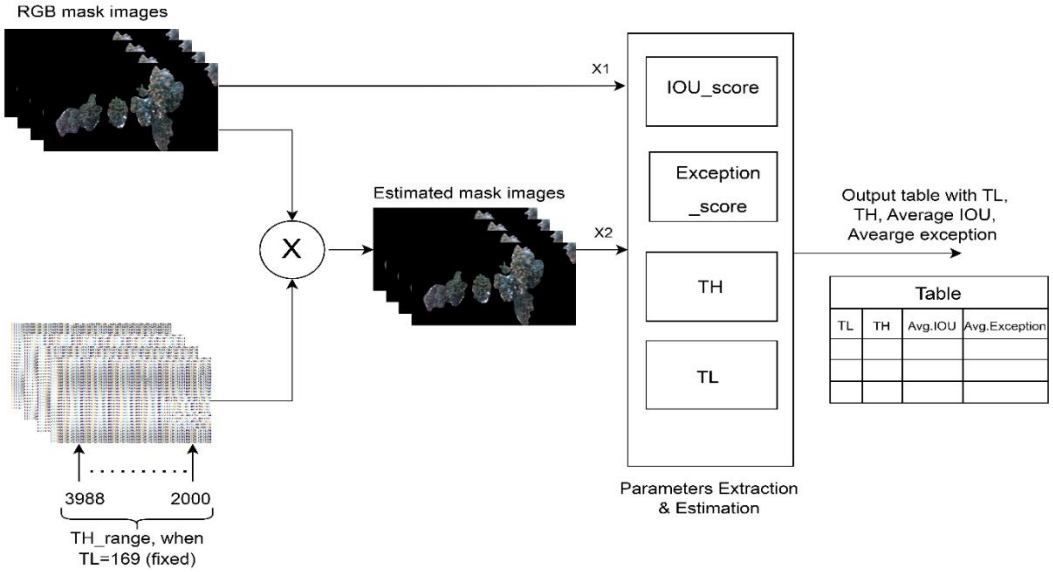

Figure 7. Block diagram of the TH value decider

(a) (b)

Figure 8. Pre-masking output (a) Original RGB image (b) Pre-masked RGB image

value of the y2 will be estimated. Fig. 8 shows the original RGB image and its pre-masked image with TL and TH thresholding ranges. As shown in Fig. 8b, in pre-masked image, black color background indicates unwanted pixel from the image. Hence, segmentation model will get less confused with the surroundings and model results will be automatically improved.

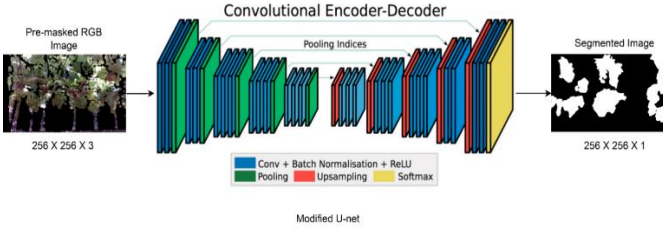

Figure 9. 3D grape cluster segmentation model

As per Fig. 9, in 3D grape cluster segmentation, instead of giving the original RGB image to the proposed modified U-net segmentation model, pre-masked images with an unwanted regions of the image removal are given to the proposed segmentation model. The addition of pre-masking block to proposed model have shown an interesting improvement in the final segmentation results.

2.2.2 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 the dataset 2 are given to the 2D weight

prediction model. The pixel area of region of interest (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, white pixel area is the region of interest (ROI). Using the *white_area_pixel = (img==255)* 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.

As shown in Fig. 1, for comparison purpose, the segmented images from the above segmentation model is given to the ML and DL based weight prediction model [54]. Keras regression model [54] is used as DL based weight prediction of vineyard. Keras regression model [54] 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. List of feature vectors inside the .csv file mentioned in table 2.

After that, in *sklearn.model_selection,* a train_test_split *( )* model is present that divides the complete dataset into 4 parts:

1. X_train: training features,
2. y_train: training labels,
3. X_test: testing features,
4. y_test: testing labels.

Depending on the *test_size* parameter, splitting will be performed. In our case, *test_size = 0.1,* which means 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 *model.fit (X_train, y_train)* command, and predictions for the test dataset are made with the *model.predict (X_test)* command. And finally, on the segmented output of dataset 3 is also tested on each weight prediction model 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, area of interest is predicting the weight of the grape clusters by considering other parameters. So machine learning regression models are best suited for performing the desired task. Based on the literature survey, five machine learning models and their comparative studies are considered for the analysis. These considered techniques are as follows:

A. Linear Regressor
B. Ridge Regressor
C. Bayesian Ridge Regressor
D. Decision Tree Regressor
E. Random Forest Regressor

To design this model, Python language is preferred. In python, there is a Sci-kit Learn library which contains all the ML models.

2.2.2.1 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 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. Trainable parameters are 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.

TABLE 2. 2D TRAINABLE FEATURE PARAMETERS

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 ROI	Numeric
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

2.2.2.2 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. As shown in table 3, for training the weight prediction models, .csv file containing the various features has been considered related to the cluster images. With height (cm), width (cm), distance (cm), and pixel area, for each image, some more estimated features like

TABLE 3. 3D TRAINABLE FEATURE PARAMETERS

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 ROI	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 (SD)	Standard deviation of the .raw files of the corresponding images	Numeric
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

minimum depth, maximum depth, and standard deviation are also examined as trainable parameters. By analyzing the dataset 2, the average height and width of each cluster in the image have been estimated. For minimum (min.) and maximum (max.) depth estimation, first multiplying RGB image with the raw image is taken, and then using .max and .min functions in the Numpy library, minimum depth and maximum depth are estimated. And in a similar way, with .std() and .mean() functions in Numpy, standard deviation and average distance (cm) are estimated. And 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.

2.3 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. So in this section, performance evaluation parameters of all models of yield prediction systems are mentioned.

2.3.1 Grape cluster segmentation model

According to literature survey [33, 36, 37, 43, 44, 45, 55, 56], best performance evaluation parameters for segmentation task are Pixel Accuracy (PA) and Mean Intersection-over-union (MIoU). Details of these parameters is given below:

2.3.1.1 Pixel Accuracy (PA)

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

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

2.3.1.2 Mean Intersection over Union (MIoU)

Intersection over union (IOU) is estimated by dividing overlapping pixel area between the actual mask and the predicted mask to the combined pixel area of actual and predicted mask. Average of IOU for each image is nothing but Mean IOU (MIoU), which is mathematically expressed as:

$$MIoU = \frac{\sum_1^N \left(\frac{Area_{actual} \cap Area_{predicted}}{Area_{actual} \cup Area_{predicted}}\right)}{N} \ldots\ldots\ldots \text{(Eq. 13)}$$

Where, N is total number of images tested. This value range from 0 to 1 and model providing value closer to 1 is considered as best model for segmentation task.

2.3.2 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:

2.3.2.1 R-Squared Score

R Square measures goodness of best fit line. Its expression is given as:

$$R^2 = \frac{\Sigma(y_{predicted} - y_{mean})^2}{\Sigma(y_{actual} - y_{mean})^2} \ldots\ldots\ldots\ldots\ldots \text{(Eq. 14)}$$

The R-squared value is between 0 and 1, and the highest value of it indicates the best fit line is perfect.

2.3.2.2 Mean Squared Error (MSE)

The mean square error is an average of the squares of the errors. It is given as:

$$\text{MSE} = \frac{1}{\text{N}}\sum_{\text{i=0}}^{\text{N}}(\text{y}_{\text{actual}} - \text{y}_{\text{predicted}})^2 \quad \ldots\ldots \text{(Eq. 15)}$$

Minimum the value of MSE better will be the model performance.

2.3.2.3 Root Mean Squared Error (RMSE)

Root Mean Square Error (RMSE) is the square root of MSE. It is used more commonly than MSE because, in some cases MSE value becomes too large to compare easily. Secondly, MSE is calculated by squaring the error, and when the square root is considered for same, it becomes equal to prediction error. It is given as:

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{\text{N}}\sum_{\text{i=0}}^{\text{N}}(\text{y}_{\text{actual}} - \text{y}_{\text{predicted}})^2} \quad \ldots\ldots\ldots\ldots \text{(Eq. 16)}$$

2.3.2.4 Vineyard yield error

To evaluate the performance of the yield prediction model, error between actual yield and predicted yield is being considered. Lesser the error value, better will be the model performance. Mathematically, it is given as:

$$Error = Yield_{(actual)}(kg) - Yield_{(predicted)}(kg) \quad \ldots\ldots\ldots \text{(Eq. 17)}$$

3. Results and Discussions

The result analysis of all the 2D and 3D models included in yield prediction systems is discussed in this section. As show in table 4, the results of the pre-masking on dataset includes total images, image resolution and dataset size. 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 pre-masking operation, it was reduced to 71.38 MB.

TABLE 4. RESULTS OF PRE-MASKING ON GRAPESNET DATASET

Model	Total Images	Image Resolution	Dataset Size
Original dataset	424	424p × 240p	86.063 MB
Pre-masked dataset	424	424p × 240p	71.38 MB

As shown in table 5, all the 2D and 3D grape cluster segmentation models are trained on a GPU system with 403 RGB images as the training dataset and 412845 trainable parameters. When 500 epochs are given, the time complexity of 3D grape cluster segmentation model is less, and the accuracy is higher as compared to the 2D grape cluster segmentation model with time complexity of 50m 36s, accuracy up to 92.02 % and MIoU upto 81.69%. Similarly for 1000 epochs as well the 3D model is performing better with lesser time complexity of 92m 11s, better pixel accuracy up to 94.81% and excellent MIoU up to 86.13 %.

As per table 6, comparative results of all the 2D DL and ML based grape cluster weight prediction regression models, the decision tree regression model and the random forest regression model are performing much better, as they gives 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 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.

TABLE 5. COMPARTIVE RESULTS OF GRAPE CLUSTER SEGMENTATION MODELS

Model	Number of Epochs	Training dataset	Trainable Parameters	Time Complexity	PA (%)	MIoU (%)
2D grape cluster segmentation model	500	403	412845	56m 41s	88.50	80.21
	1000	403	412845	101m 01s	90.23	81.95
3D grape cluster segmentation model	500	403	412845	50m 36s	92.02	83.69
	1000	403	412845	92m 11s	94.81	86.13

TABLE 6. COMAPARATIVE RESULTS OF 2D DL AND ML BASED GRAPE CLUSTER WEIGHT PREDICTION REGRESSION MODELS

Approach	Model	R2_score (%)		MSE		RMSE	
		Train	Test	Train	Test	Train	Test
DL Based	2D Keras Regression model (Barbole *et al.,* 2022)	98.6723	48.1713	0.050	0.9811	0.0704	0.9905
ML Based	LR	92.4172	63.2012	0.02982	0.2078	0.1726	0.4558
	RR	92.4172	63.2012	0.02982	0.2078	0.1726	0.4558
	BRR	92.4172	63.2012	0.02982	0.2078	0.1726	0.4558
	DTR	100.0	68.6723	0.0	0.1768	0.0	0.4205
	RFR	99.9311	70.2908	0.00027	0.1677	0.01645	0.4095

Similarly, the RMSE values of the DTR and RFR models are less, which are 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 from table 7, it can be stated that, the decision tree regression model and the random forest regression model are performing much 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 lowest 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 less, which are up to 0.0 and 0.02477 respectively, for the train dataset, and 0.4250 and 0.3999 respectively, for test dataset.

From the table 8, it can be observed that LR and the RR models perform better, with an accuracy values up to 97.1654%, for both the models with 2D dataset, and 99.3356% and 99.3350% respectively, for 3D dataset.

TABLE 7. COMAPARATIVE RESULTS OF 3D DL AND ML BASED GRAPE CLUSTER WEIGHT PREDICTION REGRESSION MODELS

Approach	Model	R2_score (%)		MSE		RMSE	
		Train	Test	Train	Test	Train	Test
DL Based	3D Keras Regression Model (Barbole *et al.,* 2022)	99.2201	51.1289	0.0031	0.8420	0.0555	0.9176
ML Based	LR	92.4773	66.5864	0.02958	0.1886	0.1719	0.4343
	RR	92.8396	66.5864	0.02813	0.1886	0.1677	0.4343
	BRR	93.0195	63.9485	0.02778	0.2035	0.1666	0.4512
	DTR	100.0	68.0075	0.0	0.1806	0.0	0.4250
	RFR	99.8439	71.6766	0.00061	0.1599	0.02477	0.3999

TABLE 8. COMPARTIVE RESULTS OF 2D/3D DL AND ML BASED AVERAGE SLOTWISE WEIGHT PREDICTION MODELS

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	37.6363	30.2967	-3.7539	3.5857	-	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

TABLE 9. COMPARATIVE RESULTS OF DL AND ML BASED ENTIRE 2D AND 3D VINEYRAD YIELD PREDICTION SYSTEMS

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	14904.4609	11997.8845	-1520.03	1386.543	-	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

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 9 which has comparative results of DL based and all ML based yield prediction models, 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.

4. CONCLUSION

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 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 with 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, 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 proposed work are performing well, and all of them have 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 are performing 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 are giving the best results with minimum error values compared to other models which are up to 55.6905 kg and 55.7426 kg. An accuracies of 3D LR and RR models are up to 99.58% for both, which is remarkable. From all the comparative analysis of 2D and 3D yield prediction system, it can be concluded that 3D yield prediction gives superior results with additional parameters estimated from the depth information.

LIMITATIONS & FUTURE SCOPES

1. In this 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 such datasets on a variety of grape types in India.
2. While training a DL-based model, it needs images and their masks as an input. 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.
3. The current 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.

DATA AVAILABILITY STATEMENT

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

REFERENCES

[1] "Global agriculture towards 2050," High-level Expert Forum, 2009.

[2] J. Ranganathan, R. Waite, T. Searchinger and C. Hanson, "How to Sustainably Feed 10 Billion People by 2050," World Resources Institute, 2018.

[3] A. Sharma, A. Jain, P. Gupta and V. C, "Agriculture: A Comprehensive Review," *IEEE Access,* 2020.

[4] V. Hakkim, E. Joseph, A. Gokul and K. Mufeedha, "Precision farming: The future of Indian agriculture," *Journal of Applied Biology & Biotechnology,* vol. 4, pp. no. 6, pp. 68_72, 2016.

[5] F. Shah, F. Shah and N. Mahnoor , "Grape Production Critical Review in the World," *SSRN Electronic Journal,* 2020.

[6] C. Fernandez-Maloigne, D. Laugier and C. Boscolo, "Detection of apples with texture analyses for an apple picker robot," *Proceedings of the Intelligent Vehicles '93 Symposium,* pp. pp. 323-328, 1993.

[7] E. Parrish and A. K. Goksel , "Pictorial pattern recognition applied to fruit harvesting," *Transactions of the ASAE,* vol. 20, p. 822–827, 1977.

[8] Whittaker, Miles, Mitchell and Gaultney, "Fruit location in a partially occluded image," *Transactions of the ASAE,* vol. 30, p. 591–597, 1987.

[9] D. Slaughter and R. C. Harrel, "Color vision in robotic fruit harvesting," *Transactions of the ASAE,* vol. 4, no. 30, p. 1144–1148, 1987.

[10] A. Sites and M. J. Delwiche, "Computer vision to locate fruit on a tree," *Transactions of the ASAE,* pp. 2285-3039, 1988.

[11] M. Cardenas, A. Hetzroni and G. E. Miles, "Machine vision to locate melons and guide robotic harvesting," *Transactions of the ASAE,* pp. 91-7006.

[12] J. Baeten, K. Donn, S. Boedrij and . W. Beckers, "Autonomous fruit picking machine: A robotic apple harvester," *Field and Service Robotics,* vol. 42, p. 531–539, 2008.

[13] R. O. Duda and . P. E. Hart, "Use of the Hough transformation to detect lines and curves in pictures," *Comunications of the ACM,* vol. 15(1), pp. 11-15, 1972.

[14] J. Illingworth and J. Kittler, "A survey of the Hough transform," *Computer Vision, Graphics and Image Processing,* no. 44, p. 87–116, 1988.

[15] G. Grasso and M. Recce, "Scene analysis for an orange picking robot," *Proceeding in International Conference of Computer Technology in Agriculture(ICCTA'96),* 1996.

[16] R. Ceres, J. L. Pons, A. R. Jim´enez and Mart´ın , "Agribot: A robot for aided fruit harvesting," *Industrial Robot,* vol. 5, no. 25, 1998.

[17] D. Barbole, P. Jadhav and S. B. Patil, "A Review on Fruit Detection and Segmentation Techniques in Agricultural Field," *International Conference on Image Processing and Capsule Networks,* vol. 300, p. 269–288, 2021.

[18] C. Wang, Y. Tang, X. Zou, W. SiTu and W. Feng, "A Robust Fruit Image Segmentation Algorithm against Varying Illumination for Vision System of Fruit Harvesting Robot," *Optik - International Journal for Light and Electron Optics,* vol. 131, pp. 626-631, 2017.

[19] S. Liu and M. Whitty, "Automatic grape bunch detection in vineyards with an SVM classifier," *Journal of Applied Logic,* vol. 13, p. 643–653, 2015.

[20] F. Liu, L. Snetkov and D. Lima, "Summary on fruit identification methods: A literature review," *Advances in Social Science, Education and Humanities Research,* vol. 119, 2017.

[21] I. Goodfellow, Y. Bengio and A. Courville, Deep Learning, Cambridge: MIT Press, 2016.

[22] G. Lin, Y. Tang, X. Zou and J. Xi, "Guava detection and pose estimation using a low-cost RGB-D sensor in the field," *Article in sensor,* 2019.

[23] S. Bargoti and J. Underwood, "Image segmentation for fruit detection and yield estimation in apple orchards," *Journal of Field Robotics,* vol. 34, p. 1039–1060, 2017.

[24] M. Stein, S. Bargoti and J. Underwood, "Image Based Mango Fruit Detection, Localization and

Yield Estimation using Multiple View Geometry," *Article in sensors,* 2019.

[25] J. Naranjo-Torres, M. Mora, R. Hernánd, R. Hernández-García and . R. J. Barrientos , "A Review of Convolutional Neural Network Applied to Fruit Image Processing," *Journal of applied science,* 2020.

[26] S. W. Chen, S. S. Shivakumar, S. Dcunha, J. Das, E. Okon, C. Qu, C. J. Taylor and V. Kumar, "Counting apples and oranges with deep learning: A data-driven approach," *IEEE Robotics and Automation Letters,* vol. 2, p. 781–788, 2017.

[27] H. Habaragamuwa, Y. Ogawa, T. Suzuki, T. Shiigi, M. Ono and N. Kondo, "Detecting greenhouse strawberries (mature and immature), using deep convolutional neural network," *Engineering in Agriculture Environment and Food,* vol. 11, no. 3, pp. 127-138, 2018.

[28] X. Liu, S. W. Chen, S. Aditya, N. Sivakumar, S. Dcunha, C. Qu, C. T. Taylor , J. Das and V. Kumar, "Robust Fruit Counting: Combining Deep Learning, Tracking, and Structure from Motion," *International Conference on Intelligent Robots and Systems(IROS),* 2018.

[29] Z. Liu, J. Wu, L. Fu, Y. Maje, Y. Feng, R. Li and Y. Cui, "Improved Kiwifruit Detection Using Pre-Trained VGG16 With RGB and NIR Information Fusion," *IEEE access,* vol. 8, 2020.

[30] I. Sa, , Z. Ge, F. Dayoub, B. Upcroft, T. Perez and C. McCool, "DeepFruits: A Fruit Detection System Using Deep Neural Networks," *Article in sensors,* 2016.

[31] B. Arad, P. Kurtser, E. Barnea, B. Ha, Y. Edan and O. Ben-Shahar, "Controlled Lighting and Illumination-Independent Target Detection for Real-Time Cost-Efficient Applications. The Case Study of Sweet Pepper Robotic Harvesting," *Article in sensors,* 2019.

[32] H. Altaheri, M. Alsulaiman and G. Muhammad, "Date fruit classification for robotic harvesting in a natural environment using deep learning," *IEEE Access,* vol. 7, p. 117115–117133, 2019.

[33] R. Marani, A. Milella, A. Petitti and G. Reina, "Deep neural networks for grape bunch segmentation in natural images from a consumer-grade camera," *Journal of Precision Agriculture,* 2020.

[34] J. Lee, H. Nazki, J. Baek, Y. Hong and M. Lee, "Artificial Intelligence Approach for Tomato Detection and Mass estimation in Precision Agriculture," *Article in sustainability,* 2020.

[35] X. Ni, C. Li, H. Jiang and F. Takeda, "Deep learning image segmentation and extraction of blueberry fruit traits associated with harvest ability and yield," *Article in Horticulture Research,* 2020.

[36] T. T. Santos, L. L. De-Souza, . A. A. Dos-Santos and S. Avila, "Grape detection, segmentation and tracking using deep neural networks and three-dimensional association," *Computer Vision and Pattern Recognition,* 2020.

[37] C. Zhang, H. Ding, Q. Shi and Y. Wang, "Grape Cluster Real-Time Detection in Complex Natural Scenes Based on YOLOv5s Deep Learning Network," *Agriculture,* vol. 12, no. 1242, 2022.

[38] Y. Yu, K. Zhang, L. Yang and D. Zhang, "Fruit detection for strawberry harvesting robot in non-structural environment based on Mask-RCNN," *Computers and Electronics in Agriculture,* vol. 163, no. 104846, 2019.

[39] P. Ganesh, K. Volle, T. Burks and S. Mehta, "Deep Orange: Mask R-CNN based Orange Detection and Segmentation," *6th IFAC Conference on Sensing, Control and Automation Technologies for Agriculture AGRICONTROL 201,* vol. 52, p. 70–75, 2019.

[40] D. K. Barbole and P. M. Jadhav, "Comparative Analysis of Deep Learning Architectures for Grape Cluster Instance Segmentation," *IT in Industy,* 2021.

[41] H. Kang and C. Chen, "Fruit Detection, Segmentation and 3D Visualization of Environment in Apple Orchards," *Computer and Electronics in Agriculture,* 2019.

[42] H. Kang and C. Chen, "Fruit Detection and Segmentation for Apple Harvesting Using Visual Sensor in Orchards," *Article in Sensors,* 2019.

[43] K. Bresilla, G. D. Perulli,, A. Boini and B. Morandi, "Single-Shot Convolution Neural Networks for Real-Time Fruit Detection Within the Tree," *Frontiers in Plant Science,* 2019.

[44] Y. Tang, M. Chen, C. Wang, L. Luo, J. Li, G. Lian and X. Zou, "Recognition and Localization Methods for Vision-Based Fruit Picking Robots: A Review," *Frontiers in Plant Science,* 2020.

[45] Y. Tang, H. Zhou, H. Wang and Y. Zhang, "Fruit detection and positioning technology for a Camellia oleifera C. Abel orchard based on improved

YOLOv4-tiny model and binocular stereo vision," *Expert Systems with Applications,* vol. 211, 2023.

[46] S. Nuske, S. Achar, T. Bates and S. Narasimhan, "Yield estimation in vineyards by visual grape detection," *Proceedings of the IEEE International Conference on Intelligent Robots and Systems,* pp. 25-30, 2011.

[47] L. Luo, Y. Tang, X. Zou, C. Wang, P. Zhang and W. Feng, "Robust Grape Cluster Detection in a Vineyard by Combining the Ada-Boost Framework and Multiple Color Components," *Article in sensors,* vol. 16, 2016.

[48] L. Luo, Y. Tang, Q. Lu, X. Chen, P. Zhang and X. Zou, "A vision methodology for harvesting robot to detect cutting points on peduncles of double overlapping grape clusters in a vineyard," *Computers in Industry,* Vols. 130-139, p. 99, 2018.

[49] E. Badeka, T. Kalabokas, K. Tziridis, A. Nicolaou, E. Vrochidou, E. Mavridou, G. A. Papakostas and T. Pachidis, "Grapes Visual Segmentation for Harvesting Robots Using Local Texture Descriptors," *Computer Vision Systems,* p. 98–109, 2019.

[50] H. Cecotti, A. Rivera, M. Farhadloo and M. A. Pedroza, "Grape detection with convolutional neural networks," *Expert Systems with Applications,* vol. 159, 2020.

[51] California Historical Society collection , "Close-up of a grape cluster on a vine," *University of Southern California Digital Library ,* 2012.

[52] D. K. Barbole and P. M. Jadhav, "GrapesNet: Indian Grape Clusters RGB & RGB-D Image Datasets," *Mendeley Data,* 2023.

[53] D. K. Barbole and P. M. Jadhav, "GrapesNet: Indian RGB & RGB-D vineyard image datasets for deep learning applications," *Journal of Data in Brief,* 2023.

[54] D. K. Barbole and P. M. Jadhav, "Grape Yield Prediction using Deep Learning Regression Model," *2022 International Conference for Advancement in Technology (ICONAT), Goa, India,* pp. 1-6, 2022.

[55] M. J., C. S. and M. N. Dailey, "Fruit detection, tracking, and 3D reconstruction for crop mapping and yield estimation," *11th International Conference on Control Automation Robotics & Vision (ICARCV, 2010),* p. 11, 2010.

[56] A. B. Payne, K. Walsh, P. Subedi and D. Jarvi, "Estimation of mango crop yield using image analysis – Segmentation method," *Computers and Electronics in Agriculture,* vol. 91, pp. 57-64, 2013.

[57] K. A. Forbes and G. M. Tattersfield, "Estimating fruit volume from digital images," *IEEE AFRICON Conf. 1999,* vol. 1, pp. 107-112, 1999.

[58] G. Lin, Y. Tang, X. Zou, J. Xiong and J. Li, "Guava detection and pose estimation using a low-cost RGB-D sensor in the field," *Article in sensors, ResearchGate,* 2019.

[59] D. Wang, C. Li, H. Song, H. Xiong, C. Liu and D. He, "Deep Learning Approach for Apple Edge Detection to Remotely Monitor Apple Growth in Orchards," *IEEE access,* vol. 8, 2020.

[60] S. Nuske, K. Wilshusen, S. Achar, L. Yoder, S. Narasimhan and S. Singh, "Automated visual yield estimation in vineyards," *Journal of Field Robotics,* vol. 31, p. 837–860, 2014.

[61] Y. Ge, Y. Xiong and P. From, "Instance Segmentation and Localization of Strawberries in Farm Conditions for Automatic Fruit Harvesting," *6th IFAC Conference on Sensing, Control and Automation Technologies for Agriculture AGRICONTROL,* vol. 52, p. 294–29, 2019.

[62] M. Stein, S. Bargoti and J. Underwood, " Image based mango fruit detection, localization and yield estimation using multiple view geometry," *Articles in sensors,* vol. 16, no. 1915.

[63] S. Filippo, D. Gennaro, P. Toscano, P. Cinat, A. Berton and A. Matese, "Low-cost and Unsupervised Image Recognition Methodology for Yield Estimation in Vineyard," *Frontiers in plant science,* vol. 10, no. 559, 2019.

[64] B. Millan, S. Velasco-Forero, A. Aquino and J. Tardaguila, "On-the-Go Grapevine Yield Estimation using Image Analysis and Boolean Model," *Hindawi-Journal of Sensors,* 2018.

[65] X. Wei, K. Jia , J. Lan , Y. Li , Y. Zeng and C. Wang , "Automatic method of fruit object extraction under complex agricultural background for vision system of fruit picking robot," *Optik - International Journal for Light and Electron Optics,* vol. 125, pp. 5684-5689, 2014.

[66] Y. Tang, J. Qiu and Y. Zhang, "Optimization strategies of fruit detection to overcome the challenge of unstructured background in field orchard environment: a review," *Precision Agricuture,* 2023.

[67] H. Williams, M. Jones, M. Nejati, M. Seabright, J. Bell, N. Penhall, J. Barnett, M. Duke, A. Scarfe and

H. Ahn, "Robotic kiwifruit harvesting using machine vision, convolutional neural networks, and robotic arms," *Biosystems Engineering,* vol. 181, pp. 140-156, 2019.

Ms. Dhanashree K. Barbole, Research Scholar at School of Electronics and Communication Engineering, Dr. Vishwanath Karad MIT World Peace University, Pune, India.

Dr. Parul M. Jadhav, Associate Professor at School of Electronics and Communication Engineering, Dr. Vishwanath Karad MIT World Peace University, Pune, India