MethodsX 13 (2024) 102859

Contents lists available at ScienceDirect

MethodsX

journal homepage: www.elsevier.com/locate/methodsx

TopoGeoFusion: Integrating object topology based feature computation methods into geometrical feature analysis to enhance classification performance ☆

N. Shobha Rani [a], Keshav Shesha Sai [b], B.R. Pushpa [b], Arun Sri Krishna [b], M.A. Sangamesha [c], K.R. Bhavya [d], Raghavendra M. Devadas [e,*], Vani Hiremani [f]

[a] Department of Artificial Intelligence and Data Science, Gitam School of Technology, Bengaluru, GITAM (Deemed to be University), India
[b] Department of Computer Science, School of Computing, Amrita Vishwa Vidyapeetham, Mysuru, India
[c] Department of Chemistry, The National Institute of Engineering, Mysuru, India
[d] Department of Computer Science and Engineering, Gitam School of Technology, Bengaluru, GITAM (Deemed to be University), India
[e] Department of Information Technology, Manipal Institute of Technology Bengaluru, Manipal Academy of Higher Education, Manipal, India
[f] Department of Computer Science and Engineering, Symbiosis Institute of Technology, Pune Campus, Symbiosis International (Deemed University), Pune, India

ARTICLE INFO

Method name:
TopoGeoFusion

Keywords:
Computer vision
Geometrical features
Object detection
Topology based features
Sustainable technology
Quality assessment

ABSTRACT

This study used smartphone captured RGB images of gooseberries to automatically sort into standard, premium, or rejected categories based on topology. Main challenges addressed include, separation of touching or overlapping fruits into individual entities and new method called 'TopoGeoFusion' that combines basic geometrical features with topology aware features computed from the fruits to assess the grade or maturity. Quality assessment helps in grading the fruit to determine market suitability and intelligent camera applications. Computer Vision-based techniques have been applied to automatically grade the quality of gooseberries as standard, premium, or rejected according to fruit maturity. Smartphone-captured images of 1697 Indian Star Gooseberries are contributed to the study. This work acquired images consisting multiple fruits with overlapping and non-overlapping boundaries for concurrent quality assessment. Multiple classifiers such as Random Forest, SVM, Naive Bayes, Decision Tree, and KNN were applied to grade the gooseberry fruit. Random Forest classification with a fusion feature model resulted in an accuracy of 100 % towards reject, standard, and premium classes for test sets with four training strategies. The proposed segmentation model proves reliable in fruit detection & extraction with an average mAP of 0.56, resulting in an acceptable model for grade assessment.

- The study highlights the effectiveness of TopoGeoFusion in automating the grading process of gooseberry fruits using topologically computed features.
- The developed models exhibit high accuracy and reliability, even in challenging scenarios such as overlapping and touching fruits.
- The method provides the technique to detect and extract the occluded objects and compute the features based on the partial object's topology.

☆ **Related research article:** Prabhu, Akshatha, N. Shobha Rani, and Chandra Sekhar Nandi. "Towards importance of comprehensive color features analysis using iterative golden ratio proportions for Alphonso mango ripening stage classification by adapting to natural progressive ripening method." Journal of Food Composition and Analysis 126 (2024): 105873. https://doi.org/10.1016/j.jfca.2023.105873

* Corresponding author.
E-mail addresses: shobarani.kamala@gmail.com (N.S. Rani), raghavendra.devadas@manipal.edu (R.M. Devadas).

https://doi.org/10.1016/j.mex.2024.102859
Received 10 April 2024; Accepted 10 July 2024
Available online 14 July 2024

Specifications table

Subject area:	Computer Science
More specific subject area:	Machine Learning, Computer Vision
Name of your method:	TopoGeoFusion
Name and reference of original method:	Geometrical features [33] Chen, Y. Q., Nixon, M. S., & Thomas, D. W. (1995). Statistical geometrical features for texture classification. Pattern recognition, 28(4), 537–552.
Resource availability:	NA

Background

The proposed method TopoGeoFusion moves beyond traditional geometric feature analysis by incorporating object topology-based methods for feature computation. Object topology refers to the inherent connectivity, shape relationships, and spatial arrangements within an object, independent of its size or deformation. By integration of the topological features with traditional geometric features such as size, area, perimeter etc., a more comprehensive feature set can be created. The comprehensive set of features can significantly improve the discriminative power when classifying objects. Additional information about how different parts of an object is connected and arranged helps in obtaining a more accurate classification compared traditional features set.

In this study, we use the self-collected datasets of Phyllanthus acidus (Gooseberries) to classify into standard, premium and reject based on its topology by adapting to proposed method “TopoGeoFusion”. The proposed method help improve classification performance by weeding out irrelevant details and focusing on the topology-based features. Gooseberries, similar to many fruits, have intricate shapes and unique ways their parts are connected. The proposed method determines the most important geometric features from the datasets of gooseberries by mainly capturing the overall form, roundness, outline, and how different parts of the gooseberry are arranged in space.

Gooseberries are small, spherical fruits that are native to tropical regions of Southeast Asia and other parts of India [1,2]. Indian Star Gooseberries, loaded with essential nutrients, are powerful health and energy boosters supplemented with nutrients such as iron, calcium, phosphorus, vitamin C, and vitamin A with antibacterial, anti-inflammatory, and antioxidant properties [3]. The fruits are used to treat various health issues related to immune systems, digestive problems, and other medical ailments. These fruits serve industries, humankind, and scientists in several ways. The cultivation of gooseberries may bring lots of opportunities to farmers and the agricultural sector to generate revenue and support in rural development present in tropical regions of India [4,5]. Indian star gooseberries are usually harvested manually by shaking the tree branches or using Mechanical harvesters resulting in a combination of grades such as standard, premium, and reject based on size [6,7]. In the continually evolving landscape of agricultural production, growers, distributors, and consumers all place a high priority on fruit quality. Quality evaluation has typically relied on subjective techniques that are frequently erratic and prone to human error. Quality assessment is crucial in determining the market suitability of fruit and has a significant impact on intelligent camera applications Using different grade qualities in packaging would impact the marketability and increase product waste [8]. A crucial post-harvest operation is the grade assessment based on the visual features of the fruit; quality assessment helps in grading the fruit to determine market suitability. Grading involves the measurement of the physical geometry of fruit for a Computer Vision system. Computer vision-based techniques have revolutionized the fruit grading industry by providing a high level of accuracy and efficiency in assessing fruit quality based on its physical geometry. These techniques combined with machine learning attract great benefits for use in industrial food processing.

Method details

The goal of the proposed method “TopoGeoFusion” is to perform the automatic sorting and grading of Indian Star Gooseberries [9,10] and introduce innovations to control food waste due to quality issues in agricultural products [11,12]. Computer vision systems can analyze fruit images at a rapid rate and with precision by using advanced image processing and machine learning-based models. They can extract significant data from the images, including size and volume that was impractical for human assessors to determine with high accuracy and accurate quality assessment leading to increased revenue generation. In the context of current technologies employed for grading and sorting fruit, various challenges exist in detecting and extracting fruit with overlapping and touching contours. The shaking mechanism of fruit in the assembly line may leave the fruit samples with touching and overlapping contours [13], leading to challenges in sorting fruit and other challenges exist due to variability in fruit appearance, occlusion, complex contour extraction issues, lighting conditions, data diversity, and real-time processing requirements. From the perspective of fruit grading, the presence of variability in terms of size, shape, color, texture, and surface defects is natural due to data diversity, varying conveyor belt alignments, and lighting conditions. Developing a robust algorithm that can handle various challenges based on their visual geometric representation is crucial per current advancements. Further, fruit in real-time industrial processing scenarios are often challenging to capture their exclusive visual features due to the partial appearance of fruit samples on conveyor belts and improper lighting conditions. Thus, the challenges pointed out can significantly affect the appearance of the fruit, making it difficult to extract reliable visual features for grading. Therefore, addressing these challenges requires advancements in computer vision algorithms, image processing techniques, and machine learning models, and it is very crucial to deploy these modules as part of computer vision systems for fruit grading.

The proposed study investigates the effectiveness of image processing techniques, fusion feature models, and machine learning classifiers to grade Indian Star Gooseberries as standard, premium, and reject. We address the challenges of non-

overlapping/overlapping/touching fruit contours from a single image view of multiple fruit samples. In this work, the primary objective is to provide an inexpensive, non-destructive, and reliable computer vision system for industrial advantage in quality control & production process. Thus, the proposed grading system detects and extracts multiple fruits of varying grades from a single image view and classifies them into standard, premium, and reject. In the proposed study, the following contributions are made toward the quality assessment of star gooseberries based on the visual geometry of the fruit.

1. A dataset of 1697 samples that are collected in a fixed acquisition setup is contributed for the sake of analysis. The dataset comprises samples of 1697 fruits with single and multiple fruit images comprising of overlapping and touching fruit challenges.
2. An improved method "TopoGeoFusion" utilizing computer vision and machine learning approaches is devised and implemented to assess the quality of gooseberries. The method is based on the feature computation based on visual appearance and maturity classification which are crucial in determining the market suitability. A fusion approach is introduced that combines hand-crafted and conventional geometric features for maturity analysis resulting in more precise grading of gooseberry fruits.
3. To address the challenges in the process of fruit quality assessment when dealing with images of multiple fruits with overlapping and touching boundaries, a method is proposed for the detection, extraction, and classification of fruit quality based on the images.
4. As part of the evaluation, the study evaluates the performance of Random Forest, SVM, Naive Bayes, Decision Tree, and KNN, in grading gooseberry fruits via TopoGeoFusion. The evaluation of the TopoGeoFusion for the grade assessment model is performed on both self-built datasets of single and multiple fruit images with overlapping and touching challenges.

Numerous investigations have been conducted in the past on automated assessment of quality based on the visual characteristics of the fruit in RGB images. The review of the significant works related to the classification of fruits or vegetables based on visual features is briefed as follows.

Zhou et al. [14] proposed a technique for different maturity stages classification of strawberry flowers and fruits using YOLO-v3 based on aerial and ground-level digital images, and a Mean Average Precision (MAP) of 0.88 was obtained. Next, using the improved YOLO-v4 model, grape maturity classification is investigated by Qiu et al. [15] the method adopts Mobilenetv3 to extract features, and experiments are carried out to compare various deep learning models that achieve an average precision of 93.52 % with a 10.82 ms detection speed. Castro et al., [16] compared the performance of four machine-learning classifiers to categories of Cape gooseberries based on their ripeness level. The three-color space models such as RGB, HSV, and Lab used to distinguish the seven different levels of ripeness of fruits, the model based on Lab and SVM provided the best result with an F1 score of 70.14 %. Shahi et al., [17] applied mobileNetV2 to extract fruit information and obtained 95.75 %, 96.14 %, and 96.37 % on three datasets with reduced parameters and improved accuracy. Saranya et al. [18] proposed Convolution Neural Networks to classify banana fruit ripeness at four stages. The model was compared with the state of deep learning models and 96.14 % accuracy was achieved.

Subsequently, Li et al. [19] explored methods to monitor the growth stages of Blueberries using color component analysis that are trained on different machine learning classifiers, with the highest accuracy of 86 % achieved with KNN. Pistachios fruit's color and texture information is utilized to estimate ripeness level by Kheiralipour et al., [20] Further, feature selection is used for choosing the most discriminant features that are trained by LDA, QDA, and ANN with resulting accuracies of 93.75 %, 97.5 %, and 100 %, respectively. In another study, Behera et al., [21] applied a combination of machine-learning approaches to extract features and classify Papaya fruit based on its maturity stage. Also, various deep learning models are trained to estimate the three maturity stages. Later, Ohali et al., [22] designed a computer vision system to perform Date fruit grading and sorting using RGB color space and BPNN classifier with an accuracy of 80 % achieved. Zhang et al. [23] performed fruit classification based on color, texture, and shape features. The PCA is used to reduce the dimensionality and lastly, versions of SVM are utilized for classification. Hossain et al. [24] proposed six convolution neural network lightweight models for fruit classification with average accuracies in the range of 85–90 %.

Later, Liu et al., [25] worked on measuring the freshness of fruits based on machine learning and deep learning approaches on public datasets consisting of 10 classes. In a couple of investigations by Taner et al., [9] proposed a CNN model to classify seventeen varieties of hazelnuts. The experiments are conducted to evaluate the performance of the proposed model with the existing pre-trained deep learning models. In subsequent investigations, Kaur et al., [26] analyzed the color, size, and texture features to determine the different maturity stages of plum fruits. Sarkar et al., [10] worked on the freshness classification of Indian gooseberry by employing image-processing techniques to extract shape and geometric features. In a study, Sarkar et al., [27] used color histogram analysis of RGB and HSV color space combined with other features to determine the freshness of gooseberry fruit. Next, Gai et al., [28] proposed an improved YOLO V4 model to categorize cherry fruit based on ripeness level; the study proved that improved YOLO V4 gives 94 % accuracy compared to other versions of YOLO models. In work on ripeness and spoilage detection of Green Apples by Cárdenas et al., [29], color features and machine learning classifiers are used. A CNN-based method based on RGB channel processing is proposed by Chen et al., [30] for Citrus fruit ripeness evaluation and reported 95.07 % accuracy. Mukherjee et al., [31] texture features are used to measure surface irregularities on the Amla fruits using thresholding segmentation. Arunachalaeshwaran et al., [32] worked on freshness classification on hog Plum fruit using pre-trained deep-learning models. Mehta et al., [33] proposed a classification model to predict the freshness of multiple fruits. VGG16 model adapted to extract features which are further classified using machine learning methods.

From the survey, the following observations are made from the existing works that contributed towards the development of automated fruit quality based on visual features of the fruit images.

1. The majority of works focus on applying conventional features by approximating the overall shape using traditional image processing techniques, works specifically focusing on feature computation precisely as per the shape of the fruit are not devised.
2. Implications of estimating the shape of the fruit along with the outer area of the fruit shape may lead to lower precision in terms of estimation of size or volume.
3. Challenges related to multiple fruit quality classification containing touching fruits samples and partial views are not considered in any of the work.
4. Though works related to infield detection exist in the literature, the techniques for quality assessment based on infield images may not be apt for industrial post-harvest processing needs. It is highly recommended to consider the constrained approach of acquiring RGB images, especially in the case of classification of fruits based on the fruit size and visual properties.

Description

In this study, Indian Star Gooseberries of varying sizes/maturities were harvested from orchards in Chandravana, Bogadi, India. First, the harvested fruit is graded by experts into standard, premium, and reject. In total, 706 standard, 481 premium, and 510 reject fruit were available from multi-fruit image samples and 832 of single-fruit sample images. The image acquisition system is made with cardboard as a base covered with black tissue paper. A mobile digital camera of 12 megapixels Android phone is fixed onto a stand 15 cm from the ground to capture the images in natural daylight. As grade assessment is mainly based on visual geometry for Indian Star Gooseberries, no additional light sources are considered to emphasize the texture. The image acquisition setup is depicted in Fig. 1. Multiple challenges are considered to capture the multiple fruit samples with/without overlapping/touching contours in a

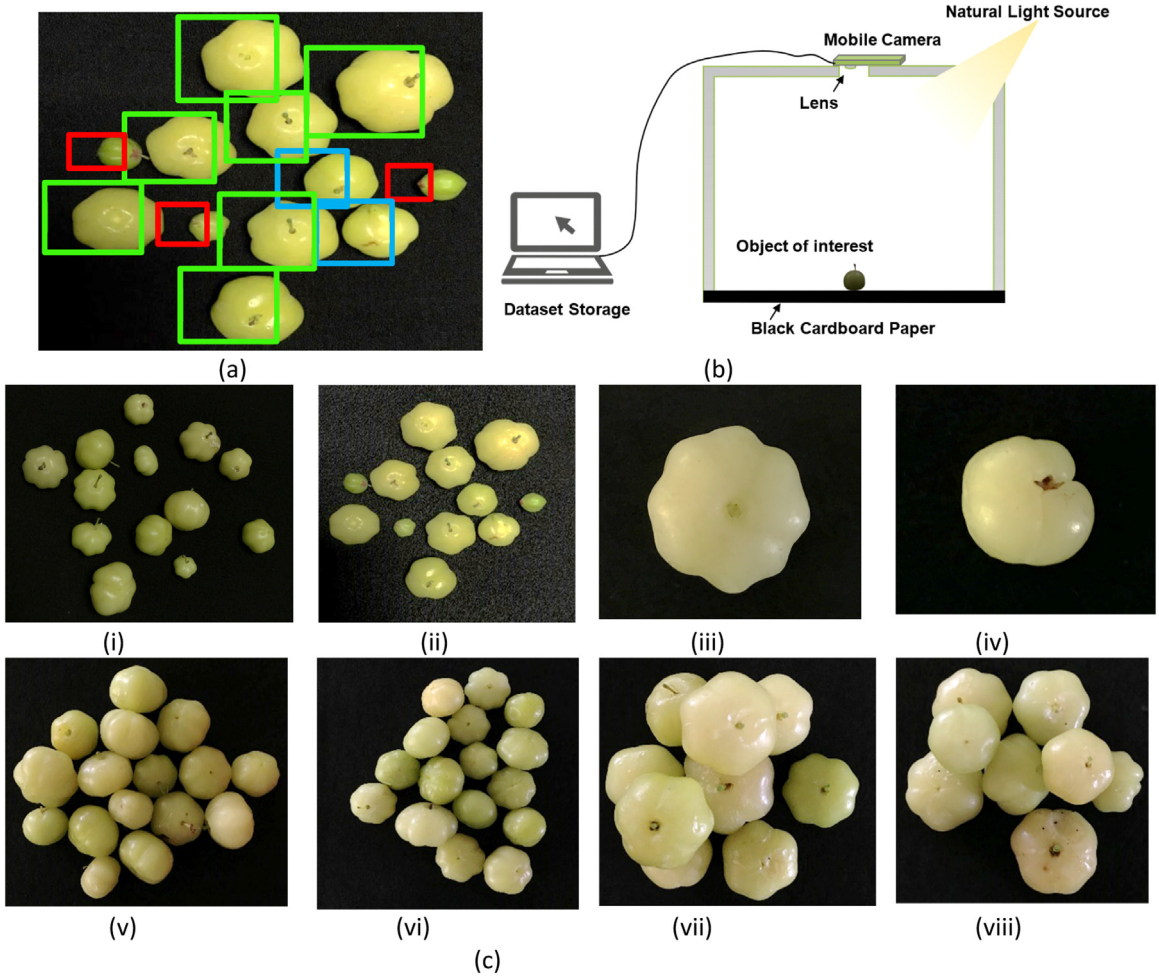

Fig. 1. Samples of Indian Star Gooseberries (A) Multi-fruit samples: Bounding box-Green: Premium, Blue: Standard, Red: Reject (B) Image acquisition setup for Indian Star Gooseberry dataset collection (C) Multiple fruit: isolated boundaries (i) & (ii), single fruit samples (iii) & (iv), multiple fruit touching boundaries (v) & (vi), multiple fruit overlapping boundaries (vii) & (viii).

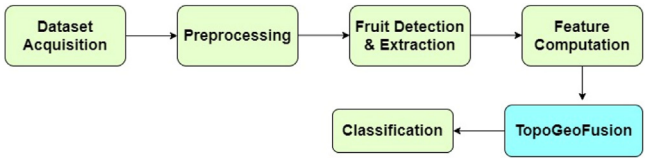

Fig. 2. Block diagram of TopoGeoFusion for grading of gooseberries.

single frame, processing of multiple fruit samples in one iteration, and time spent in sorting and grading processes will be optimized. Challenges in additional labor requirements to monitor the fruit flow in industrial processing operations can be eliminated. Therefore, multiple fruits in a single image frame of varying grades were obtained for analysis.

Flow of method

The steps in extracting fruit samples from single/multiple fruit images by addressing overlapping/touching fruit boundary challenges are described in subsequent sections. Fig. 2 shows the block diagram of the workflow of the proposed algorithm.

The proposed system for grading gooseberries begins with acquiring images using a fixed image acquisition setup to ensure that the images are appropriately captured to facilitate subsequent analysis. Next, preprocessing techniques are applied to remove noise that occurs during image acquisition. Consequently, fruit detection is performed to detect and locate each fruit using computer vision-based methods. Following this, each fruit is subjected to feature analysis, where a set of conventional and hand-crafted features are computed to quantify the visual properties of the fruit. Finally, extracted features are combined and employed for predicting fruit grade using Machine learning classifiers.

Region of interest - Single/multiple fruit extraction with isolated boundaries

The harvested fruit scattered on a dark background is the region of interest. To extract the fruit from RGB image I_c, initially, I_c is converted to grayscale image I_g. As images are captured in natural daylight in an indoor environment, there will be impulse noise that resembles salt and pepper noise in a dark background. Therefore, a linear filtering method is applied using Gaussian filter G_f on grayscale image I_g with 11×11 rectangular kernel and standard deviation $\sigma = 0$ along both X & Y directions. Filter G_f convolves the image I_g, pixel by pixel with the center pixel c of the kernel as the origin. The target pixel p in the image where the response is stored is overlayed by c. As noise resembles maximal gray levels in the dark background the responses obtained by G_f with maximal gray levels are repainted to minimal gray levels resulting in noise removal producing a noise-filtered image with a blurry effect. The Canny edge detector is known for its exceptional precision in edge detection. In the fruit detection task, where precise contour segmentation is necessary for further processing stages, this precision is very vital. By precisely determining the borders of fruits, the Canny technique improves the precision of our detection system. Also, the Gaussian filtering prior to the edge detection helps to reduce the influence of noise on the edge detection process. This is important in real-world situations where there may be different levels of noise in the input images. It's possible that noise cannot be adequately handled by morphological procedures alone, which could result in less precise contour recognition.

The first step in contour-based image processing tasks is edge detection. We generate a binary edge map that emphasises the fruit contours by utilising the Canny edge detector. Applying morphological techniques like dilation to this edge map is an ideal starting point. Accurately identifying and segmenting the fruits may be difficult if morphological techniques are used alone as they may not yield a clear contour map.

A processed pixel p with spatial coordinates $(x,\ y)$ of filtered image I_f is obtained by convolving G_f concerning kernel spatial locations $(x',\ y')$ with $I_f(x,\ y)$ as given by (1).

$$G_f = \left(\frac{1}{2\pi\sigma^2}\right) e^{-\frac{\left(x^2+y^2\right)}{2\sigma^2}} \tag{1}$$

$$I_f(x,y) = \sum_{i=x\in I_g} \sum_{j=y\in I_g} I_g\left(x',\ y'\right) * G_f\left(x-x', y-y'\right) \tag{2}$$

$G_f(x-x', y-y')$ is the value of the filter, G_f relative to the location $(x-x',\ y-y')$. Subsequently, edge detection is performed using the canny edge operator op_{Canny} on filtered image I_f.

Based on the gradient magnitude and directions of pixels, op_{Canny} marks the pixels as background and edge pixels (boundary) by eliminating weak gradient pixels by hysteresis thresholding. Thus op_{Canny} produces edge-detected binary image I_B as given by (3).

$$I_B \leftarrow op_{Canny}\left(I_f\right) \tag{3}$$

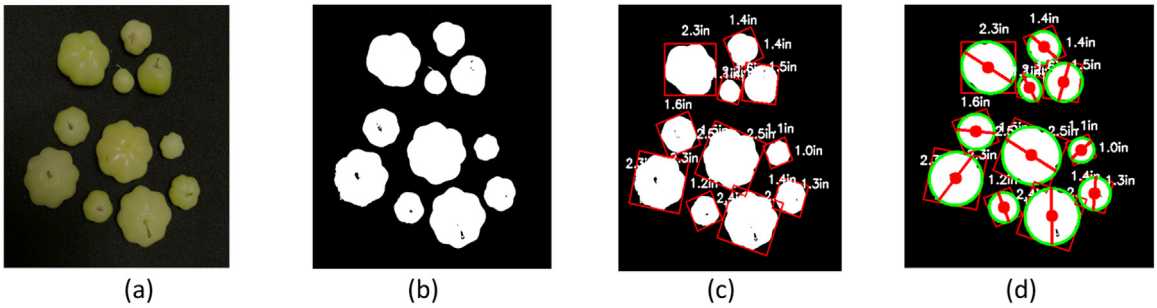

Fig. 3. Contour extraction - Single/multiple fruit samples with isolated boundaries (a) Original image. (b) Thresholded image. (c) object detection. (d) Contour estimation.

The edge detected image I_B is subject to morphological dilation M_d twice using structuring elements s size 1×1 pixel, indicating no expansion or shrink of edge pixels to occur.

$$I_m = M_d(I_B,\ s,\ i = 2) \tag{4}$$

Where $i = 2$ is the number of iterations and I_m is the morphologically processed image. The dilation is mainly applied to abridge the slightly disconnected edge pixels in the fruit boundary.

Contour extraction of fruit

To extract features from each fruit, it is required to detect the fruit boundary/contour and identify the shape. A contour is a group of pixels that defines the shape of an object. In this study, two strategies are adopted to extract the fruit. Strategy one works on the image samples with multiple fruit/single fruit with non-overlapping boundaries, while strategy two is specifically designed to deal with overlapping fruit image samples.

Contour extraction-non-overlapping boundaries

The edge-detected image I_B is subject to contour extraction to retrieve the external contour. Each fruit contour consists of adjoining pixels connected via 4/8/m adjacency relationships defining a connected component. Each connected component is indexed, and its spatial positions are stored without redundant points by the contour detection methods. A circular bounding box is estimated for each fruit f_i where $i = 1, 2, 3 \ldots n$ using a convex hull approach. The enclosed circle ensures all boundary pixels lie inside, producing the center $(c_x,\ c_y)$ and radius R for each fruit f_i. Fig. 3 depicts the processing pipeline of contour extraction for image samples with multiple fruit and non-overlapping boundaries.

Contour extraction-overlapping/touching boundaries

The proposed dataset for overlapping fruit contour extraction needs to be more structured & organized. Therefore, contour extraction of fruit is done using Mask Region Convolutional Neural Networks (Mask RCNN). In this study, 72 images are acquired with overlapping/touching boundary positions of fruit out of 281 harvested fruits. From the samples 72 images, 42 samples are subjected to makesense.ai tool to generate the mask images for overlapping/touching fruit boundaries. These annotated mask images were then used to train a faster R-CNN model and to optimize model parameters. Fig. 4 shows the visuals of annotated images of samples with overlapping/touching challenges.

Faster R-CNN is a state-of-the-art object detection method for instance, segmentation of challenging target proposal regions. In the proposed study, Faster R-CNN is used only to perform instance segmentation and not for classification/grading. Faster R-CNN performs object detection in two phases, backbone network for feature-maps computation from input images and Region Proposal Network (RPN) to detect and identify contours for fruit enclosed with rectangular bounding boxes. Finally, the detected objects are individually extracted for proposed fusion feature analysis, then classification into grades. Fig. 4 depicts the workflow of detecting fruit with overlapping contours.

The Faster R-CNN (Region-based Convolutional Neural Network) model is often used for object detection and has proven to perform well in a range of computer vision tasks. In the proposed study, the authors down-sample the input feature maps of dimension 512×512 with three channels and length strides using a ResNet50 backbone network. To find possible object locations, the Faster R-CNN model's region proposal network (RPN) makes use of a collection of preconfigured anchor boxes. The anchor boxes are described by the authors as having sizes of 8×8, 16×16, 32×32, 64×64, and 128×128 and ratios of [0.5, 1.2]. Previous studies on visual attention mechanisms and the efficacy of composite backbone networks offer acceptance to the use of these anchor box specifications. The present study use ROI pooling to construct 14×14 pixel feature maps for additional processing after the region proposals are generated. Table 1 lists the hyperparameter specifications for the Faster R-CNN model, including information on the number of

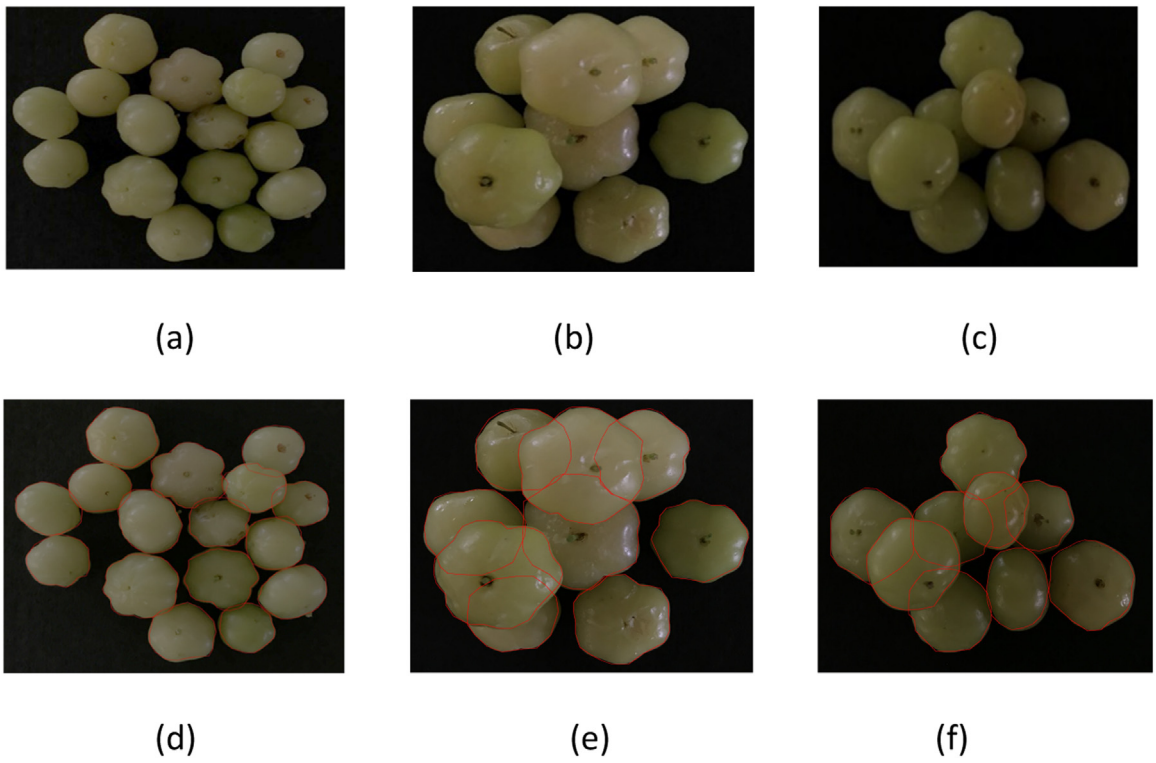

Fig. 4. Samples of annotated images used for training faster R-CNN model (a)–(c) Original images (d)–(f) Annotated samples.

Table 1
Hyperparameter specifications of Faster R-CNN model.

Hyperparameter	Description
Backbone	ResNet50 BACKBONE_STRIDES: [4, 8, 16, 32, 64]
Input Size	512 × 512 with three channels
RoI Pooling	RoI align with bilinear interpolation, FC=MLP
Region proposal method	RPN RPN_ANCHOR_RATIOS: [0.5, 1, 2] RPN_ANCHOR_SCALES: (8, 16, 32, 64, 128)
RoI Sampling	Balanced sampling strategy
RoI Pooling Size	14 × 14
Learning Rate & Learning Momentum	0.0001, 0.9
Batch Size	1
Number of Epochs & steps per epoch	Epoch=50, steps=500
Optimizer	RMSprop
Weight Initialization	Random
Dropout Rate	0.1
Loss Function	cross-entropy

training iterations, weight decay, and learning rate. The method employed makes use of the Faster R-CNN model's advantages, such as its rapid generation of region proposals and its effectiveness in object identification and localization. The successful use of composite backbone networks and the references to earlier research on visual attention indicate that the hyperparameter settings are probably a result of extensive testing and fine-tuning.

Resnet50 backbone

Because of its architectural advantages, ResNet50, or Residual Network with 50 layers, is an often utilised backbone in object detection applications, including fruit detection. ResNet50′s depth helps it to efficiently extract high-level characteristics from images, and its residual blocks help to prevent vanishing gradients—a major problem in deep networks—from occurring. The key components of Resnet50 are residual blocks that are basically defined assuming input as X and a mapping function M to learn the features from

X associating with weight vector W resulting into a output y as given by (5).

$$y = M(X, W) + X \tag{5}$$

And the mapping function M is expressed as $M(X) = W_2\sigma(W_1(X)$, where W_1 and W_2 are the elements of weight vector W and σ is the ReLU activation function.

Region proposal network

Region proposal network is used to generate region proposals from the convolution feature maps of a given input image. The object detection network refines the identified region proposals and predicts the categories of the detected objects. The convolutional feature maps are generated via backbone network through a set of convolution filters with specified kernel size and taken through sliding window process with the given anchor box specifications. The attributes associated with anchor box includes the varying scales and aspect ratios assisting in detecting the object location coordinates precisely. The region proposal network produces the probability score for each object location detected which implies the likelihood of accuracy indicating a foreground or background. Finally, the regression layer outputs the coordinate positions of each bounding box localizing the object. The efficiency of RPN is measured using binary cross entropy loss to determine the classification loss indicating how well the anchor box specifications are determining the goodness of fit with respect to various object sizes in image. Additionally, regression loss is calculated to interpret the object localization prediction. At this stage, region of interest pooling is carried out subsequently on the generated region proposals that transform them into fixed size bins of specified dimensions. Later, the max pooling is applied on each bin estimated from region proposal of objects irrespective of its size. The classification layer uses the softmax function to convert the raw scores generated by region proposal network into probability score associated with each region proposal concerning a particular class. Finally, it predicts the detected object regions along the localization scores of detection associated with each object.

Anchor box specifications

Let F be the feature map generated by backbone network with dimensions as $M \times N \times D$, indicating M as the height, N as the width and D as the number of channels. If S and A representing various scales and A as the corresponding aspect ratios for the scales for $1 \ldots n$. Then the location (i, j) in the feature map be computed as given by (6) based on the scale and aspect ratio.

$$\text{The anchor width } N = S \times \sqrt{r} \text{ and anchor height } M = S/\sqrt{r} \tag{6}$$

Where $s \in S$ *and* $r \in A$. Thus, generating a set of anchor boxes as per each s *and* r for each spatial location. Each location is assigned with a confidence score associated with anchor box signifying the strength of object presence within that corresponding location.

TopoGeoFusion – feature computation

After the contour extraction process, each fruit was interpreted to extract the required features for analysis (see Fig. 4). The bounding boxes enclosing each fruit are used to identify the spatial locations of the fruit. The spatial locations identified from binary mask images of each fruit are utilized for feature computation based on the binary image. This study employs a set of hand-crafted features and four geometrical features for analysis.

Conventional features

Geometrical features such as area, perimeter, major, and minor axis are computed to define the feature representation. The area is the crucial feature to assess the fruit's quality as it quantifies its size, which is considered by the outer boundary of the fruit. The other feature, such as perimeter, gives the circumference of the fruit based on the summing of the length of line segments along the fruit boundary. Further, the perimeter captures the measure of the fruit boundary. The other geometrical feature, the major axis, is the longest line connecting the two contour positions passing through the principal axis center. Finally, the minor axis is the shortest line connecting the center of the fruit to a contour position perpendicular to the major axis. The minor axis helps in capturing the shape and orientation of the fruit.

Hand-crafted features

In this study, we propose five hand-crafted features, fruit density, T- Extent, B-Extent, L-Extent, and R-Extent, as described subsequently. Fruit density is the area occupied by the fruit pixels, which is the sum of white pixels within the fruit's contour as given by (7).

$$D_f = \sum_{w \in f_i} I(x,\ y) \tag{7}$$

Where D_f is the fruit density, $I(x,\ y)$ represents the white pixel with spatial coordinate position $(x,\ y)$ of fruit f_i with pixel w. Other features are computed based on distance measurements from the center of the fruit. If C is the centroid of the fruit with spatial

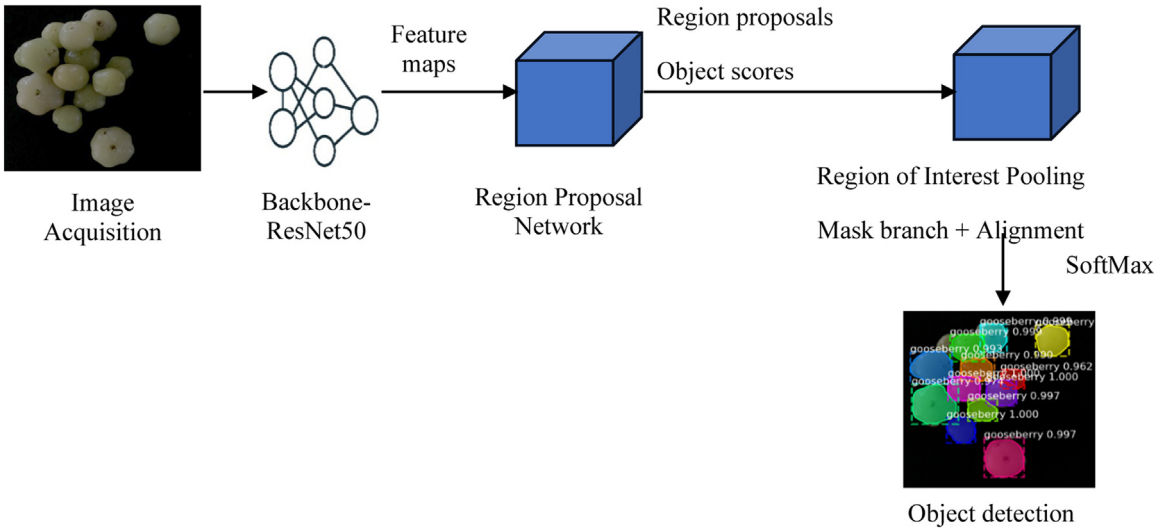

Fig. 5. Architecture of contour extraction-multiple fruit with overlapping contours.

coordinate $(C_x,\ C_y)$ and B_t, B_b, B_l, and B_r represent the boundary points lying to each other in perpendicular directions from center C, then the features T- Extent, B-Extent, L-Extent, and R-Extent are defined as the distances measured from center C to all boundary positions B_t, B_b, B_l, and B_r as given by (8)–(11).

$$T - Extent = \left[\left(x_t - C_x\right)^2 + \left(y_t - C_y\right)^2\right]^{\frac{1}{2}} \tag{8}$$

$$B - Extent = \left[\left(x_b - C_x\right)^2 + \left(y_b - C_y\right)^2\right]^{\frac{1}{2}} \tag{9}$$

$$L - Extent = \left[\left(x_l - C_x\right)^2 + \left(y_l - C_y\right)^2\right]^{\frac{1}{2}} \tag{10}$$

$$R - Extent = \left[\left(x_r - C_x\right)^2 + \left(y_r - C_y\right)^2\right]^{\frac{1}{2}} \tag{11}$$

Where $(x_t,\ y_t)$, $(x_b,\ y_b)$, $(x_l,\ y_l)$, and $(x_r,\ y_r)$ are the spatial coordinates designating boundary points B_t, B_b, B_l, and B_r, respectively. Fig. 6 gives the visual representation of features quantified in this study, and Fig. 6 depicts the visual representation of hand-crafted features and Table 2 presents the statistical descriptions of computed features.

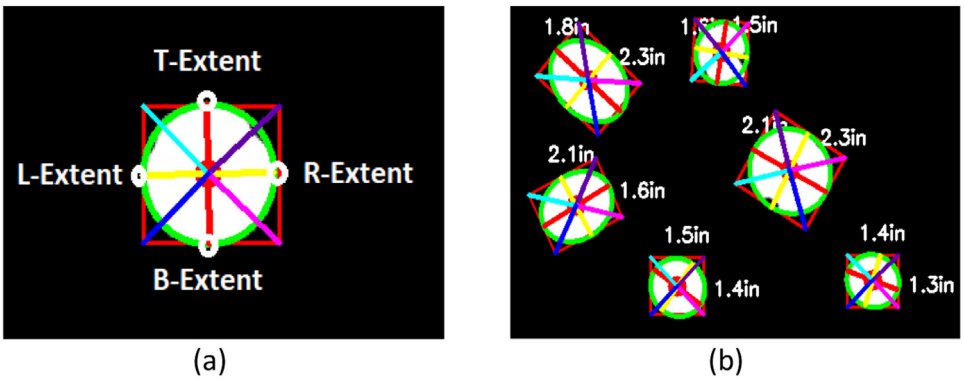

Fig. 6. Visual representation of hand-crafted features in TopoGeoFusion (a) Feature measurements from the center of the fruit to four directions named T-Extent, B-Extent, L-Extent, and R-Extent (b) Fruit detection and feature quantification representation with T-Extent, B-Extent, L-Extent and R-Extent specifications.

Table 2
Statistical description of fusion of computed features - TopoGeoFusion.

Feature Description	Features									Grade
	Fruit Density	Perimeter	Area	Major axis	Minor axis	T-Extent	L-Extent	R-Extent	B-Extent	
Min-Max	53,066–165,557	164–244	0.311–0.578	50.45–73.12	44.3–65.99	33.66–48.54	34.60–48.63	33.70–48.86	34.69–49.20	Standard
Mean	97,160	194	0.417	59.78	54.53	39.99	40.57	39.93	40.53	
Median	88,763	193.18	0.404	59.89	53.75	39.6	40.08	39.56	40.22	
Min-Max	70,316–170,827	232.41–315.04	0.610–1.111	70.94–95.14	62.31–91.62	46.98–64.77	48.43–65.92	47.35–66.13	48.45–67.30	Premium
Mean	106,170	265.78	0.786	80.77	75.66	54.71	55.27	55.14	55.74	
Median	97,769	262.79	0.753	80.34	74.57	53.99	53.9	54.18	54.55	
Min-Max	38,769–163,359	93.95–195.25	0.094–0.297	29.76–56.59	25.14–48.55	18.63–35.20	19.04–38.81	19.40–35.40	19.77–40.36	Reject
Mean	79,451	138.32	0.205	42.76	38.08	28.08	28.73	28	28.7	
Median	74,624	137.19	0.203	42.93	38.41	28.19	28.77	28.1	28.61	

Pseudocode:

1. Import necessary image processing libraries
2. Set the algorithm's input path to $Path$ to the folder containing images
3. Set the thresholding parameter to $thresh = 15$ for extracting region of interest(roi).
4. For each image I from $Path$
5. *Apply image resizing, $I = imresize(I, m \times n)$*
6. *Convert RGB to gray scale image, $I_g = RGBtogray(I)$*
7. *Remove noise in I_g with kernel size k and obtain blurred image, $I_b = Gaussianblur(I_g,\ k)$*
8. *Perform edge detection, $I_e = Canny\_op(I_b,\ lower\_thresh,\ upper\_thresh)$*
9. *Bridge gaps using dilation, $I_d = Dilate(I_e,\ iterations = 1)$*
10. *Detect contours, $I_c = Find\_contours(I_d,\ type = external,\ method = Simple\_approximation)$*
11. *Detect markers around roi, $I_m = Detect\_markers\_roi(I_c,\ method = Aruco\_markers)$*
12. *Perform contour analysis and feature extraction,*
13. *a. $white_{freq} = count\_white\_pixels(I_m(roi_n)),\ for\ n = 1 \ldots number\_of\_roi$*
14. *b. $perimeter = count\_pixels\_arclength(I_m(roi_n))$*
15. *c. $area = count\_pixels\_roi(I_m(roi_n))$*
16. *d. $major\_axis = I_m(roi_n(\max\_distance(p_1,\ p_2))\ where,\ p_1,\ p_2\ are\ boundary\ points\ on\ roi$*
17. *e. $minor\_axis = I_m(roi_n(\min\_distance(p_1,\ p_2))$*
18. *f. $topr = I_m(roi_n(distance(centroid,\ p_{tr})),\ where,\ p_{tr}\ is\ top\ right\ bounding\ box\ position$ centroid is origin of the bounding box of roi_n*
19. *g. $topl = I_m(roi_n(distance(centroid,\ p_{tl})),\ where,\ p_{tl}\ is\ top\ left\ bounding\ box\ position$*
20. *h. $botr = I_m(roi_n(distance(centroid,\ p_{br})),where,\ p_{br}\ is\ bottom\ right\ bounding\ box\ position$*
21. *i. $botl = I_m(roi_n(distance(centroid,\ p_{bl})),\ where,\ p_{bl}\ is\ bottom\ left\ bounding\ box\ position$*
22. *Store the features a to i to a feature vector*
23. Repeat until all images are processed
24. Stop

The above pseudocode depicts the procedure of feature extraction by using image processing techniques. The feature extraction is performed by the detection and analysis of contours within a given input image. The workflow initiates by importing OpenCV, NumPy, SciPy, Pandas libraries. In pseudocode, the variables I represent original RGB image, I_g is gray scale image obtained by processing I with $RGBtogray$ function. Then, I_g is subject to intermediate processing steps via Gaussian, blur, canny edge operator, morphological dilation, finding contours producing the outcomes as blurred image I_b, I_d, and I_c respectively. For each detected region of interest $roi_n,\ where\ n = 1 \ldots number\_of\_roi$ from markers detected image I_m, contour analysis carried out for extraction of features and storage in feature vector. For each roi_n of I_m, the features such as, $white_{freq}$, $perimeter$, $area$, $major_{axis}$, $minor_{axis}$, $topr$, $topl$, $botr$ and $botl$ are computed using the specified functions in pseudocode. A method named $Aruco\_markers$, is employed with $Detect\_markers\_roi$ for calibrating the image for distance-based feature measurements. Then, the image is subject to undergo resizing, conversion to grayscale, and image blurring to reduce noise. Canny edge detection is performed using $Canny\_op$, followed by morphological dilation and erosion to bridge gaps within the contours using $Dilate$. The visualization of detected contours positions are demonstrated by plotting of marker positions and feature measurements in different colors on an RGB image for visualization, and then sorted from left to right. The $Aruco\_markers$ in the pseudocode calculates the pixels-per-metric ratio for each region of interest in markers detected image and then converted to pixel measurements. The detected contour positions are analyzed to calculate attributes in terms of bounding box, centroid, width, height, along with perimeter, area, and dimensions of fitted ellipses. Finally, the computed features are for each detected fruit is stored into a data frame and then saved as CSV file.

Generalization of control parameters for Gaussian filtering, hysteresis thresholding and morphological operations

The control parameters employed for detecting the contours of Gooseberry fruit are determined via trivial experimental procedures with trial-and-error approach. To ensure reproducibility of proposed TopoGeoFusion method, the control parameters are analyzed further to determine its goodness of fit towards different circular fruit commodities. The generalization of various control parameters that are required to be fine-tuned according to fruit type is discussed subsequently.

One of the crucial parameters that is adapted in the proposed workflow is kernel size and standard deviation in Gaussian filtering process. The Gaussian filtering is used to perform image smoothing as a result of which the noise reduction will takes place while preserving edges. The increase in kernel dimensions would result in increasing smoothing effect on the fruit image. The decrease in kernel size would result in reduced smoothing effect on the fruit which is suitable for fruits with minimal noise. With regard to kernel sizes, optimal specifications are experimented between 3×3 and 7×7. For instance, the commodities such as an apple with minimal noise can work with a 3×3 kernel dimensions as that imposes slight smoothing effect. In the case of a noisy image such as oranges which has slightly jagged texture that is captured under inconsistent lighting conditions, a kernel size of 7×7 can possibly address the noise issues while maintaining the circular contour of the fruits.

Concerning the control parameter such as hysteresis thresholding, that is applied on the result of the Canny edge detection process for determination of strong and weak edges, the experimented thresholds are set in between 50 and 100 as the lowest and the high threshold is set between 150 and 200. For fruits such as lemons with adequate lighting conditions in image with clearly defined edges, low threshold range between 50 and 100 are employed. In case of uneven illumination conditions, a high threshold range between 150 and 200 works effectively.

Yet another crucial parameter in the proposed work includes size of structuring elements and number of iterations used for morphological dilation in the process of segmentation of fruits from its background. The morphological operations are highly subjective and depends on the amount of noise that is caused due to external lighting conditions during image acquisition process. However, to

maintain the integrity in the process of segmentation, a set of experiments are conducted by adapting the size of structuring elements varying from range from 3×3 to 5×5, with 1 to 3 iterations. In all experiments, it is noticed that the proposed sizes implies that morphological dilation helps in filling the small gaps in detected contours. It is recommended to increase the size of structuring element in case of images with more external noise occurred due to lighting effect. Furthermore, the number of iterations can also be increased in addition to the increase in the number of broken edgelike structures in the contours of fruits. For instance, in case of strawberries, the number of broken edges will be high due to its texture and which requires more iterations and increased in structuring element size compared to gooseberries. As per the context of image acquisition setup in the current scenarios of proposing method, the standard ranges of structuring element size and number of iterations it is restricted to the range of 3×3–5×5 and 3.

For commodities such as apples, using a 5×5 Gaussian filter with low hysteresis thresholds in the range of 50 and 150, and a 3×3 structuring element for dilation with 1 iteration, the proposed technique can handle minor noise producing the circular edges of apples. Similarly, in case of oranges, a 7×7 Gaussian filter with higher hysteresis thresholds ranging between 100 and 200 and a 5×5 structuring element with 3 iterations ensures precise extraction of fruit contours.

Table 2 emphasizes the efficiency of features computed concerning the discriminating properties from one grade to another, corresponding to hand-crafted and conventional features. The statistical descriptions such as mean, median, and min-max help in highlighting the distinctiveness of proposed hand-crafted over conventional features. The mean and median measures indicate the central feature positions concerning the distribution of analyzed features. The min and max represent the range of features distributed to each feature; a more extensive range signifies the high reliability of the features. As per Table 2, hand-crafted features project a wide range of distribution as well as the central feature positions are aligned to mean and median values. In the case of conventional features, though the min to the max range is competent, the central values of the distribution are not aligned with the mean and median for the features area and major axis.

Fig. 7 depicts the feature distribution using pair plots. The pairwise feature analysis uses pair plots with a grid of scatterplots and histogram representations. The comprehensive overview and relationship between a pair of other hand-crafted features are expressed using scatter plots by allowing simultaneous examination of the multiple features. However, in the context of one vs. the rest of the classes, the data distribution exhibits a satisfactory relationship indicating a fit for linear classifiers such as Decision Trees (DT), Support Vector Machines (SVM), and Random Forests (RF). Therefore, it is proposed to employ the Machine Learning classifiers with the linear kernel to perform predictions in the proposed model.

Classifier training and evaluation

Computer vision-based techniques, in combination with Machine Learning classifiers, have been extensively applied to automatically grade the quality of gooseberries as standard, premium, or rejected according to fruit maturity. A total of 1697 Indian Star Gooseberries samples graded by eight to ten human experts are employed for analysis. The study also considered images extracted from multiple fruit samples with overlapping and non-overlapping boundaries. The classifier training is conducted using four training strategies using stratified sampling with K-fold cross-validation for $K = 11$; the details of sample distribution for each training strategy are presented in Table 3. For classification, models such as RF, DT, SVM, Naïve Bayesian (NB), and K-Nearest Neighbor (KNN) are used for predictions. The training is conducted without any data augmentation methods to supplement with original datasets, and no potential samples influence the bias within datasets. All the samples are sorted out to ensure no spoilage in the samples considered for the study. Further, the classifier is evaluated using Accuracy, Precision, Recall, and F1-score metrics. The class-wise sample distribution statistics are depicted in Table 4, which provides a comprehensive understanding of the number of samples considered for analysis with/without challenges.

Table 3
Training strategies and dataset statistics.

Train: Test Strategies	No. of Training Samples	No. of Testing Samples	Total Number of Samples
80:20	1357	340	1697
70:30	1187	510	1697
60:40	1018	679	1697
50:50	848	849	1697

Table 4
Class-wise sample distribution statistics with various dataset artifacts.

Premium	No. of Single-fruit Samples	No. of Multi-fruit (touching contours) Samples	No. of Multi-fruit (overlapping contours) Samples	Total Number of Samples
Grade	138	62	281	481
Standard	269	183	254	706
Reject	304	183	23	510

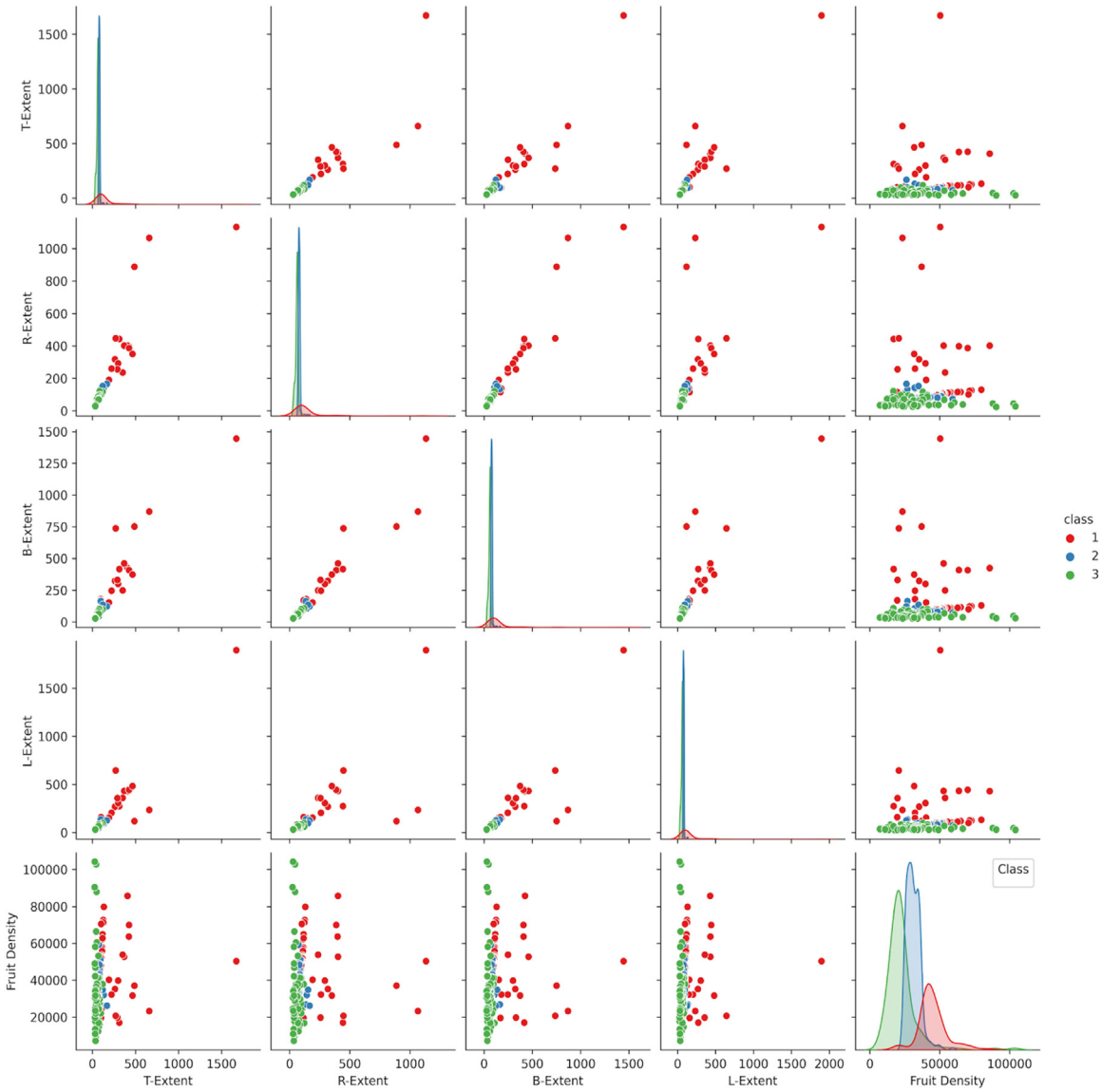

Fig. 7. Pair-plot representation of hand-crafted features for three grades 1: Reject, 2: Standard, 3: Premium.

KNN

KNN's are often used for classification tasks which share the data distribution characteristics that are linearly separable in nature. The classifier though addresses the problem of noisy data issues, it is not that effective on datasets with large number of dimensions. The KNN can perform pretty well on the context of limited number of features and reduced noise. As the principle of KNN classifier is often oriented to work based on the hyper parameter k indicating the number of neighbors considered to determine target class label of a unknown test sample. Given an unkown test sample associated with a set of features, which needs to be compared with features of training samples to identify k neighbors with closer distances. The k neighbors are the samples that exhibit high proximity in matching of features between test and k neighbors in training dataset. The target class label determined is the one with the set of neighbors that share maximum voting among the designated number of classes. In the proposed method, geometrical features are combinedly used with the distance-based features computed upon the fruit topology. The features computed exhibit a linear relationship and thus exhibiting a better efficacy making KNN an apt tool to address the present topic of research.

SVM

SVM classifier is also proved to be effective at addressing the issues associated with solving various complex classification tasks even in the presence of noisy data with non-linear as well linear data distributions. The SVM classifies the unknown test sample

by finding the optimal hyperplane position in an n-dimensional feature space with (n-1) features. The hyperplane separates the datapoints belonging to different classes by maximizing the margin between the closest points of different classes. Though SVM is designed to handle the larger dimensional data, to works pretty well with linearly separable feature space. Especially, linear kernel is for linearly separable data, whereas in case of non-linear data distribution SVM can handle using polynomial kernel, Radial Basis Function(RBF), quadratic, cubic polynomial kernel etc. Once the optimal position of hyperplane is determined, the classifier outputs the prediction. The hyper parameter C representing the learning speed plays a crucial role in classification process, for smaller values, high misclassification rate is observed and larger values a stricter separation will be achieved.

Naïve Bayes

Naïve Bayes is a probabilistic classifier that makes the predictions based on the probability of test samples towards a particular class. Given a set of features associated with training dataset, the classifier builds a learning model using the concept of posterior probability among the feature sets that hold class conditional independence property among the features. Those features would contribute towards the predictions without any dependency on other features. The Naïve Bayes classifier is known for its computational efficacy in terms of speed and accuracy when compared with other classifiers. However, the classifier offers assurance of accurate results only in the presence of dataset without any noise. The performance of Naïve Bayes is highly subjective to the linear distribution of data.

Random Forests (RF)

In the context of fruit maturity assessment or classification, RF model plays a superior role compared to any other existing machine learning models. Basically, RF performs classification by generating multiple decision trees during the learning process through an ensemble learning approach. Each decision tree that is generated as part of ensemble learning comprises a subset of samples as part of training dataset. By extracting features from each image and sending them to the Random Forest model for prediction, we can utilize the model to classify new images once it has been trained. For every image, the model will provide a class label that represents the expected class of the image based on the features that have been extracted from it. The performance efficiency of the model depends the number of decision trees that are present as part of random forest, usually the accuracy of the model increases along with the increase in number of decision trees that are present as part of RF model. The potential of the Random Forest Algorithm to handle data sets with both continuous variables—as in regression—and categorical variables—as in classification—is one of its most important properties. In tasks involving regression and classification, it performs better.

Decision trees (DT)

DT is a non-parametric supervised learning approach that is used for both regression and classification applications. It has a hierarchical tree structure comprising of a root node, branches, internal nodes, and leaf nodes. Root node stands for both the initial choice to be made and the complete dataset. While internal nodes indicate choices or evaluations based on features. Branches indicate the conclusion of a choice or examination, leading to a new node and leaf nodes stand for the final judgment or forecast. Root node stands for both the initial choice to be made and the complete dataset. While internal nodes indicate choices or evaluations based on features. Branches indicate the conclusion of a choice or examination, leading to a new node and leaf nodes to indicate the class labels. To create a decision tree, it is important to choose the Ideal quality, the optimum attribute to separate the data is chosen based on a metric such as information gain, entropy, or Gini impurity. Dataset splitting technique is used for splitting the dataset into subsets. A decision tree algorithm operates by recursively choosing the feature that yields the maximum information gain or gain ratio for each internal node. The process will be continued until the tree impacts a stopping criterion, such as a maximum depth or a minimum amount of samples in a leaf node.

Method validation

In the proposed study, the outcome of the classifier predictions is divided into three grades Premium, Standard, and Reject. Performance evaluation of proposed contributions is conducted in two phases.

1. In phase one, the Faster-R-CNN model's efficiency towards extracting overlapping/touching fruit samples is analyzed.
2. In phase two, the robustness of the classifier's predictions using the fusion feature model is studied.

Performance evaluation of faster R-CNN model

Though the main objective of the proposed study is to perform grading of Indian star Gooseberries, the challenge of the algorithm lies in extracting fruit samples with overlapping and touching contours. To extract the fruit samples with the challenges mentioned above, a faster R-CNN model is used to extract the fruit with challenging contours. We adopt a single-shot training process in one batch carried out in 10 epochs with 500 steps per epoch with a learning rate of 0.0001 and momentum of 0.9. The training uses an RMSprop optimizer with random weight initialization and cross-entropy loss function with a dropout of 0.1. Fig. 8 presents the outcomes achieved on object detection using the Faster R-CNN model.

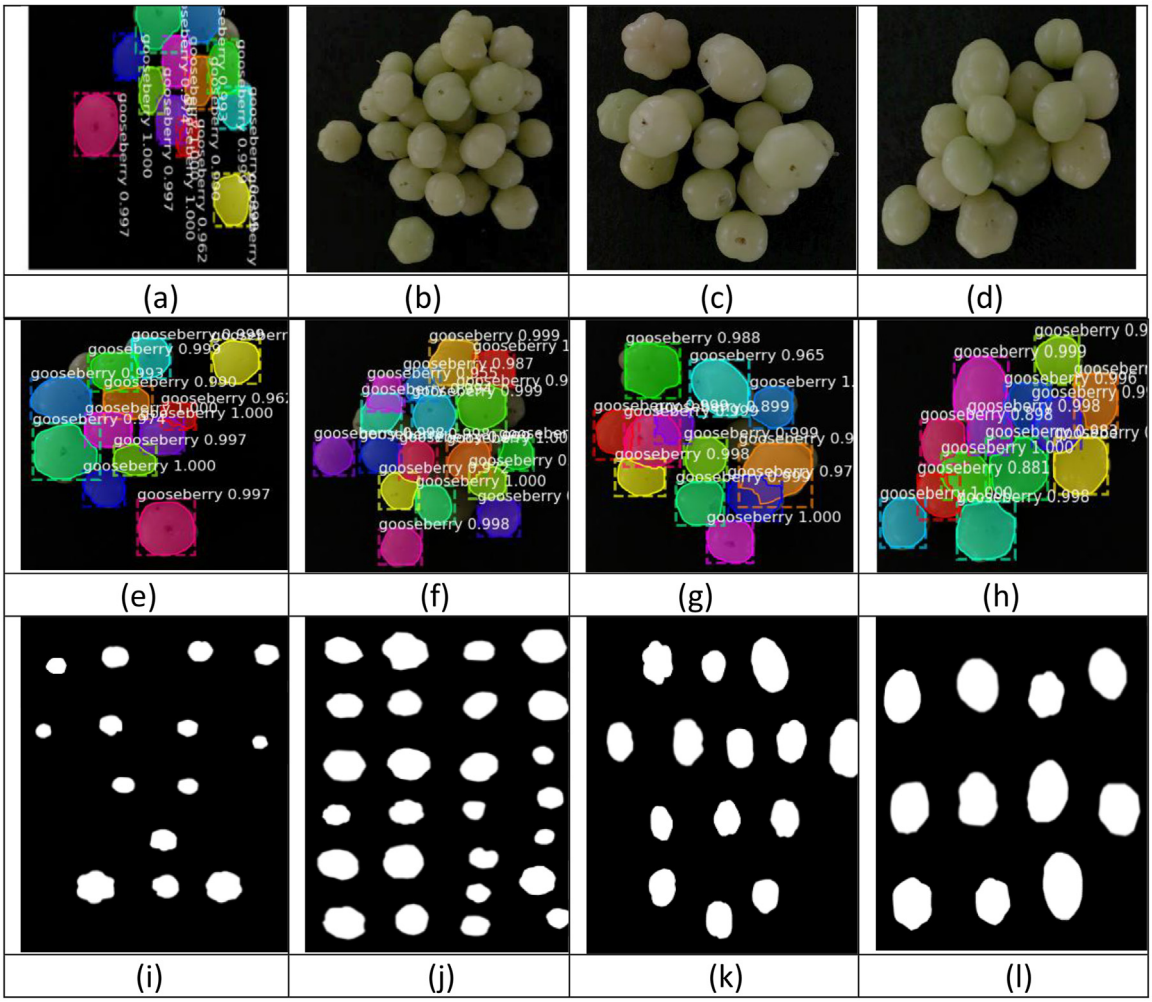

Fig. 8. Contour extraction of fruit with overlapping contours using mask-R-CNN (a)–(d) Sample images with overlapping fruit contours (e)–(h) Fruit detected by mask-R-CNN (i)–(l) Mask images of extracted fruit from (e)–(h).

As per Fig. 8, for the original images with challenges of overlapping and touching contours, the fruit detection outcomes using the region proposal network, followed by mask branch outcomes, are shown. The outcomes of individual fruit extracted from the mask image are presented by visually separating one fruit from another. The performance of the Faster R-CNN model for ten epochs and the obtained mean Average Precision (mAP), more significant than the Intersection over Union (IoU) of 0.5, is presented in Fig. 9. The learning curve indicates a consistent improvement from one to another epoch, and it observed that performance convergence was achieved from epoch 7. The performance from the seventh epoch also closely matches visual outcomes and the fruit extraction desired to perform subsequent analysis.

The mean Average Precision (mAP) is used as an evaluation metric for object detection in the proposed work for contour estimation of fruit samples with overlapping contours. A mAP of 0.56 at the 10th epoch and remains to be plateau with small increments. Furthermore, visually the outcomes achieved in terms of object detection is sufficient to carryout analysis, with an Intersection over the Union (IoU) threshold of 0.5. The outcomes indicates the performance of the Mask R-CNN model in terms of accurately localizing and delineating the contours of partially visible fruit in annotated images. Also, a mAP of 0.56 suggests that the model can detect and segment partially visible fruit with desired precision. The model correctly identifies approximately 56 % of the fruit instances with a reasonable localization accuracy. The training analyses provides empirical evidence that the validation accuracy of the model plateaued after 10 epochs. Increasing the training duration past this point did not result in appreciable accuracy gains.

The IOU threshold of 0.5 determines the minimum overlap required between predicted and ground truth contours to be considered a correct detection. In this case, if the overlapping area between the predicted and ground truth contours is at least 50 % of the union area, it is considered a successful detection. Further, the number of epochs indicates the number of iterations the model has undergone during training. In this case, the model was trained for ten epochs. Concerning training progress, the mapped value at the 10th epoch indicates the model's progress throughout the training process. It shows the average performance of the model after training for the

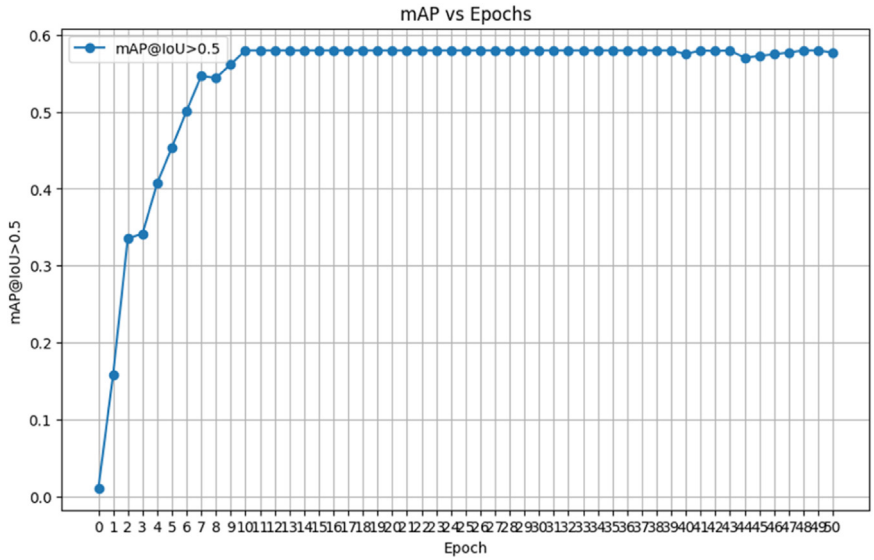

Fig. 9. Performance of Mask-R-CNN- Contour extraction of overlapping/touching fruit boundaries.

specified number of epochs. As the model trains, the map tends to increase, indicating improved performance over time. Overall, the mean Average Precision (mAP) of 0.56 at the 10th epoch suggests that the visual results of contour extraction using the Mask R-CNN model have shown promising performance in contour extraction from overlapping fruit. As this research focuses mainly on the grade assessment of fruit, we thus aim at the fine-tuning of the model's performance towards the fusion feature model to classify the fruit based on grade.

Performance evaluation of classifiers with TopoGeoFusion

This study evaluates various machine learning classifiers using four training strategies, as discussed in classifier training and evaluation section. The evaluation uses K-fold cross-validation for $K = 11$ by analyzing the trade-off between the classifier's performance vs. fusion feature model, hand-crafted features, and Geometric features. A thorough grid search and cross-validation procedure led to the selection of $K = 11$. According to our evaluations, $K = 11$ consistently provided the best resilience and accuracy at various training-testing proportions. In particular, we used $K = 11$, SVM classifiers and Decision Trees to obtain 100 % accuracy. These findings reveal how well $K = 11$ captures the underlying patterns in our dataset. Table 5 depicts the performance of various machine learning classifiers concerning hand-crafted features, fusion features, and Geometric features with precision, recall, and accuracy.

Table 5 shows that the classifiers RF and DT exhibit 100 % precision, recall, and accuracy with fusion features regardless of the training strategy employed. While the classifier KNN produces an average accuracy of 97 %, precision, and recall are close to 98 %. On the other hand, the NB produces 86 % of average accuracy and precision. In comparison, the SVM is the poor-performing classifier with minimum to maximum accuracy ranging from 62 to 68 % across various training strategies. It is also important to note that the performance of classifiers such as KNN, NB, and SVM varies and degrades across four training strategies. The performance of classifiers is observed to be reliable with fusion features compared to hand-crafted and geometric features. It is inferred that the classifier's performance degrades across training strategies by independently evaluating geometric and hand-crafted features.

DT and RF classifiers perform exceptionally well on our dataset as shown in Figs. 10 and 11. They consistently achieve 100 % accuracy, precision, and recall [25] across different training/testing proportions (80:20, 70:30, 60:40, and 50:50). K-fold cross-validation indicates that they can correctly classify all three data classes in your study. This high level of performance suggests that the classifiers effectively capture the underlying patterns and relationships within your dataset, leading to accurate predictions. It implies that the features or variables used for classification are highly informative and distinct, allowing the models to make precise decisions.

While accuracy, precision, and recall are commonly used evaluation metrics, examining the F1 score to gain a more comprehensive understanding would be beneficial. Fig. 11 shows that the F1 score is also consistently 100 % across all training strategies, especially with the fusion feature model. Additionally, the outcomes support the inference that RF and DT classifiers are performing exceptionally well compared with other datasets. Further, the F1 score metric combines precision and recall and exhibits appreciable and balanced performance by RF and DT. The F1 score of 100 % indicates that the proposed fusion feature model achieves both high precision (low false positive rate) and high recall (low false negative rate) simultaneously. It is worth noting that the fusion of geometric and hand-crafted distance features proves robust in capturing comprehensive features compared to Geometric and hand-crafted features independently for classification. Thus, the fusion feature model can produce accurate and precise predictions across three classes, regardless of the training strategy.

Table 5
Performance of classifiers vs. training strategies with K-fold Cross-validation for $K = 11$.

Classifier	Hand-crafted Features			TopoGeoFusion			Geometric Features		
	Precision (%)	Recall (%)	Accuracy (%)	Precision (%)	Recall (%)	Accuracy (%)	Precision (%)	Recall (%)	Accuracy (%)
Training Strategy S1									
SVM	80.45	82.4	81.11	62.43	59.79	68.53	84.67	82.55	83.42
DT	96.02	95.85	95.8	**100**	**100**	**100**	94.17	94.09	94
RF	99.46	99.39	99.3	**100**	**100**	**100**	94.2	94.48	94.19
KNN	86.5	88.76	87.41	98.85	98.21	98.6	86.46	86.55	85.91
NB	88.07	82.23	87.41	88.5	82.82	88.81	84.55	77.23	83.44
Training Strategy S2									
SVM	81.95	83.24	82.71	72.08	68	77.57	84.07	82.52	83.07
DT	97.31	96.94	96.72	**100**	**100**	**100**	92.52	92.17	92.35
RF	97.57	96.78	97.66	**100**	**100**	**100**	94.2	94.4	93.95
KNN	86.29	88.48	87.85	98.41	98.37	98.59	85.39	85.73	84.69
NB	86.14	82.06	86.91	88.13	82.72	88.31	85.84	76.92	82.9
Training Strategy S3									
SVM	84.31	84.33	84.91	52.78	49.9	59.64	81.79	80.4	80.74
DT	95.09	95.24	94.73	**100**	**100**	**100**	94.04	93.47	93.65
RF	96.64	96.32	96.84	**100**	**100**	**100**	93.77	93.84	93.41
KNN	87.14	89.31	88.42	97.23	97.62	97.54	82.73	83.19	82.17
NB	87.01	78.69	86.31	87.72	79.9	87.36	86.91	77.09	83.09
Training Strategy S4									
SVM	82.22	82.24	82.58	56.29	53.82	62.92	80.78	77.02	79.13
DT	94.94	93.55	94.1	**100**	**100**	**100**	91.73	91.07	91.55
RF	96.9	96.49	96.91	**100**	**100**	**100**	91.84	92.44	91.83
KNN	85.44	87.25	86.23	97.86	98.12	98.03	83.95	84.38	83.63
NB	83.8	76.18	83.42	87.87	80.57	87.07	84.78	75.41	81.7

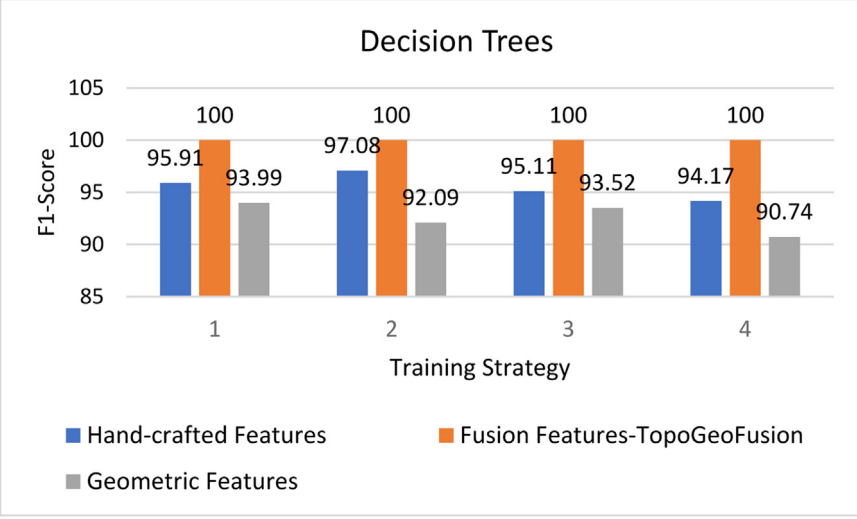

Fig. 10. Performance of DT classifier using F1 Score vs. Training strategies.

On the other hand, the lower F1 scores for geometric and hand-crafted distance features may indicate that more than these individual feature sets are needed to capture the discriminative features across all three classes. The performance is observed to be degraded as the training data size decreases by training the classifier with independent feature sets. The consistent F1 score of 100 % for the fusion feature model is inferred even with a 50 % training and 50 % testing data split, suggesting the strengths of integrating geometric and hand-crafted distance features.

Statistical analysis on performance of classifiers across different feature sets

A thorough statistical analysis is carried out on the performance outcomes reported in Table 5 in terms of precision, recall and accuracy. The statistical analysis is carried out using measures such as mean, standard deviation and analysis of variance and the outcome of the analysis is presented in Table 6.

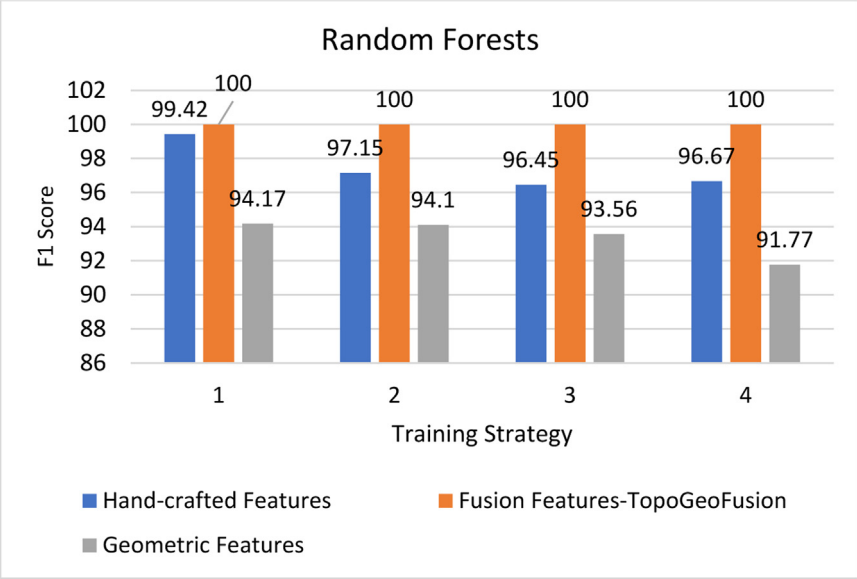

Fig. 11. Performance of Random Forests classifier using F1 Score vs. Training strategies.

Table 6
Statistical analysis of classifiers across different feature sets.

Precision			
Classifier	Hand-crafted Features (Mean ± SD)	TopoGeoFusion (Mean ± SD)	Geometric Features (Mean ± SD)
SVM	82.73 ± 1.73	60.40 ± 8.38	82.33 ± 1.58
DT	95.84 ± 0.95	100 ± 0	93.62 ± 1.06
RF	97.64 ± 1.38	100 ± 0	93.50 ± 0.52
KNN	86.34 ± 0.84	98.34 ± 0.75	84.13 ± 1.49
NB	86.26 ± 1.84	88.31 ± 0.38	85.52 ± 1.12
Recall			
Classifier	Hand-crafted Features (Mean ± SD)	TopoGeoFusion (Mean ± SD)	Geometric Features (Mean ± SD)
SVM	83.05 ± 1.58	57.88 ± 7.94	80.12 ± 2.13
DT	95.89 ± 1.20	100 ± 0	92.70 ± 1.41
RF	97.25 ± 1.34	100 ± 0	93.79 ± 0.35
KNN	88.45 ± 0.93	98.08 ± 0.49	84.96 ± 1.22
NB	79.29 ± 3.13	81.00 ± 1.21	76.14 ± 0.67
Accuracy			
Classifier	Hand-crafted Features (Mean ± SD)	TopoGeoFusion (Mean ± SD)	Geometric Features (Mean ± SD)
SVM	82.83 ± 1.41	67.17 ± 7.65	81.09 ± 1.65
DT	95.83 ± 1.04	100 ± 0	92.99 ± 1.13
RF	97.68 ± 1.19	100 ± 0	93.85 ± 0.36
KNN	87.48 ± 0.94	98.19 ± 0.32	84.10 ± 0.94
NB	86.51 ± 0.95	87.64 ± 0.78	82.33 ± 0.57

Based on Table 6, in the point view of precision, it is noticed that a significant difference (for range 82.73 %–97.64 %) is observed concerning the handcrafted feature set. The proposed TopoGeoFusion exhibits notably high precision of 100 % for classifiers DT and RF. While, it is reported as below 60.40 % for SVM and 88.31 % by Naïve Bayes classifier. On the other hand, precision is reported in the range of 82.33 %–93.30 % for geometric features. Furthermore, recall and accuracy is reported high as 100 % for the proposed TopoGeoFusion technique compared other method which lie below 76.14 % in case of other classifiers.

Based on the analysis of variance with p-value < 0.05 upon various feature sets considered, it is noticed that atleast one feature set significantly differs in terms of precision, recall and accuracy from other classifiers. Finally, notable difference is reported in terms of precision with TopoGeoFusion resulting into a high precision for RF and DT classifiers. Similarly, a perfect recall of 100 % is evident with DT and RF classifiers. Thus, the proposed method shows a significant impact on classifier performances.

Table 7
Comparative study on state-of-the-art methods versus performance achieved.

State-of-the-art work	Quality assessment of fruit/ vegetable	Features used	Classifier	Accuracy
Behera et al. (2021) [21]	Papaya	Shape + Texture	KNN Naive Bayes SVM	KNN+HOG – 100 % VGG19 – 100 %
Sarkar et al. (2022) [10]	Amla	Hue & RGB	Correlation Coefficient	96 %
Wajid et al. (2018) [29]	Orange	RGB color space Borders	DT ANN Naive Bayes	Ripe - 95.24 % Scaled 94.96 % Unripe - 90.15 %
Mazen et al. (2019) [28]	Banana	Color Statistical Texture	ANN	Green and over-ripen classes - 100 % Yellowish green + mid-ripen - 97.75 %
Faisal et al. (2020) [27]	Date fruit	Convolution	VGG 19 NASNet	Seven stages- 98.5 %
Zhou et al. (2021) [14]	Seven maturity stages of Strawberry	Convolution	You Only Look Once (YOLOv3),	Flower - 0.93 Flower fruit - 0.84 Green fruit - 0.89 Green, white fruit - 0.93 White, red fruit - 0.92 Red fruit - 0.94 Rotted fruit - 0.80
TopoGeoFusion (proposed method)	Gooseberries	Handcrafted topologically aware features + conventional features	SVM DT RF KNN NB	Standard – 100 % Premium – 100 % Reject – 100 % (with DT and RF)

Comparative analysis of TopoGeoFusion with state-of-the-art methods

The performance of proposed contributions in terms of maturity-based grade assessment is also compared with the effective state-of-the-art methods in the literature in this section. To compare, the contributions to grade assessment provide comprehensive insights on the type of fruit, features employed, and classifiers used to perform a quality assessment. Through this, the robustness of features employed concerning the geometrical structure of fruit becomes more apparent and assists in validating the results achieved. Table 7 presents the details of state-of-the-art methods related to classified fruit.

Table 7 shows that works related to maturity assessment of circular fruit types such as Amla and Orange include Bediako et al. [31] and Wajid et al. [32] using multi-spectral color features. Accuracy levels range from 90 % to 96 % for fruit Amla and Orange using color features. While the fruit types do entirely not rely on the color and statistical features from multi-spectral feature space, it is vital to consider some specific features related to geometrical structure that would result in robust and precise outcomes. Further, the other works employed deep learning models to perform the quality assessment of non-circular fruit structures such as bananas, papaya, and apples. Though the state-of-the-art works employ micro-level features through deep convolutional models, the performance is at most 90 % towards most fruit types. Further, the grade assessment of fruit utilizing the integration of geometrical and hand-crafted features would result in the proposed method achieving high accuracies.

Model's generalizability

In the context of the model's generalizability, the proposed TopoGeoFusion model is extensible, specifically for classifying circular fruit structures that vary in size in terms of maturity for grading rather than color. Specifically, the sorting problem is mainly based on size for packaging purposes in industries. The small fruits or vegetables such as all berries, circular fruit including *round purple fruit, round green fruit, red and yellow small fruit, red and brown fruit, small apples, cherries,* and *tomatoes* can be easily recognized by the proposed model with high precision. Further, the fusion feature model is also scalable to other fruit and pulses with non-circular structures by consideration of datasets in the context of maturity assessment. By supplementing the learning models specific to fruit, pulses, or berries, the proposed model can be generalized to produce highly reliable outcomes.

In this work, the selection of amla was influenced by specific characteristics and challenges to identifying contours of it, including its size, texture, and color fluctuations. It is required to thoroughly examine and optimize our detection and grading process for these specific features by focusing on a single fruit variety and then rescale the algorithm to other fruits with varying and rough surface textures.

Objective derivative

In the context of accomplishing research outcomes concerning the proposed objectives, the challenges concerning extracting objects with overlapping and fruit contours are successful. The model for fruit detection proved reliable for dataset creation for the

image of multiple fruit or object samples where one is overlapping and touching each other and also isolated form. Consequently, the proposed fusion feature model has also proved to be robust enough to classify fruit based on size as the primary aspect. Thus, the proposed contribution and evaluation help accomplish the objectives derived from the research.

Limitations

- As the method TopoGeoFusion mainly depends on the topology and geometry of the object which is targeted for detection, computing the features related to complex spatial relationships mainly on texture is not possible.
- In order to employ the proposed method in diverse scenarios with varied object shapes, it is required to devise a model that detects and interprets the object's topology.
- In scenarios where objects are present in complex backgrounds with noise and occlusions, it is required to extract the location of the object before applying the proposed method.

Ethics statements

Human subjects do not appear in our research work nor animal experiments or even social media data for that matter.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

N. Shobha Rani: Conceptualization, Methodology, Data curation, Supervision, Formal analysis. **Keshav Shesha Sai:** Conceptualization, Methodology, Data curation, Supervision, Formal analysis. **B.R. Pushpa:** Conceptualization, Methodology, Data curation, Supervision, Formal analysis. **Arun Sri Krishna:** Methodology, Software, Formal analysis, Validation. **M.A. Sangamesha:** Methodology, Software, Formal analysis, Validation. **K.R. Bhavya:** Methodology, Software, Formal analysis, Validation. **Raghavendra M. Devadas:** Data curation, Visualization, Investigation, Data curation, Writing – original draft, Formal analysis. **Vani Hiremani:** Data curation, Visualization, Investigation, Data curation, Writing – original draft, Formal analysis.

Data availability

Data will be made available on request.

Acknowledgments

We are thankful to Amrita Vishwa Vidyapeetham Mysuru campus for providing an opportunity to use lab resources.

References

[1] S. Malviya, N. Malviya, V. Johariya, R. Saxena, R. Gupta, M. Dhere, A. Singh, Medicinal plants having anti-cancer activity, in: Medicinal Plants and Cancer Chemoprevention, CRC Press, 2024, pp. 55–162, doi:10.1201/9781003251712.
[2] S. Kalina, R. Kapilan, I. Wickramasinghe, S.B. Navaratne, Potential use of plant leaves and sheath as food packaging materials in tackling plastic pollution: a review, Ceylon J. Sci. 53 (1) (2024) 21–37, doi:10.4038/cjs.v53i1.8145.
[3] M.T. Kamble, S. Chaiyapechara, K.R. Salin, P. Bunphimpapha, B.R. Chavan, R.C. Bhujel, N. Pirarat, Guava and Star gooseberry leaf extracts improve growth performance, innate immunity, intestinal microbial community, and disease resistance in Nile tilapia (Oreochromis niloticus) against Aeromonas hydrophila, Aquac. Rep. 35 (2024) 101947, doi:10.1016/j.aqrep.2024.101947.
[4] G.A. Ataguba, M.T. Kamble, Food industry by-products as protein replacement in aquaculture diets of tilapia and catfish, in: A.K. Anal (Ed.), Food Processing By-Products and Their Utilization, 2017, pp. 471–507, doi:10.1002/9781118432921.ch20.
[5] Y. Leeya, M.J. Mulvany, E.F. Queiroz, A. Marston, K. Hostettmann, C. Jansakul, Hypotensive activity of an n-butanol extract and their purified compounds from leaves of Phyllanthus acidus (L.) Skeels in rats, Eur. J. Pharmacol. 649 (2010) 301–313, doi:10.1016/j.ejphar.2010.09.038.
[6] A. Nath, S. Mangaraj, T.K. Goswami, J. Chauhan, Post Harvest Management and Production of Important Horticultural Crops, Scientific Publishers, 2016, doi:10.22271/phyto.2020.v9.i4ab.12054.
[7] D.D. Heaton, A Produce Reference Guide to Fruits and Vegetables from Around the World: Nature's Harvest, CRC Press, 1997, doi:10.1201/9780429246616.
[8] G. Davis, J.H. Song, Biodegradable packaging based on raw materials from crops and their impact on waste management, Ind. Crops Prod. 23 (2) (2006) 147–161, doi:10.1016/j.indcrop.2005.05.004.
[9] A. Taner, Y.B. Öztekin, H. Duran, Performance analysis of deep learning CNN models for variety classification in hazelnut, Sustainability 13 (12) (2021) 6527, doi:10.3390/su13126527.
[10] T. Sarkar, A. Mukherjee, K. Chatterjee, Correlation-aided 3d vector distance estimation-based quality assessment of Indian gooseberry, J. Inst. Eng. India Ser. A 103 (2) (2022) 397–407, doi:10.1007/s40030-022-00616-6.
[11] A. Mukherjee, T. Sarkar, K. Chatterjee, D. Lahiri, M. Nag, M. Rebezov, J.M. Lorenzo, Development of artificial vision system for quality assessment of oyster mushrooms, Food Anal. Methods 15 (6) (2022) 1663–1676, doi:10.1007/s12161-022-02241-2.
[12] K. Koyama, M. Tanaka, B.H. Cho, Y. Yoshikawa, S. Koseki, Predicting sensory evaluation of spinach freshness using machine learning model and digital images, PLoS One 16 (3) (2021) e0248769, doi:10.1371/journal.pone.0248769.
[13] S. Pareek, A. N. Shikov, O. N. Pozharitskaya, V. G. Makarov, G. A. González-Aguilar, S. A. Ramalho, N. Narain, in: Phytochemicals: Chemistry and Human Health, 2nd Edition, 2017, pp. 1077–1106, doi:10.1002/9781119158042.ch54Citations.

[14] X. Zhou, W.S. Lee, Y. Ampatzidis, Y. Chen, N. Peres, C. Fraisse, Strawberry maturity classification from UAV and. near-ground imaging using deep learning, Smart Agric. Technol. 1 (2021) 100001, doi:10.1016/j.atech.2021.100001.
[15] C. Qiu, G. Tian, J. Zhao, Q. Liu, S. Xie, K. Zheng, Grape maturity detection and visual pre-positioning based on improved YOLOv4, Electronics 11 (17) (2022) 2677, doi:10.3390/electronics11172677.
[16] W. Castro, J. Oblitas, M. De-La-Torre, C. Cotrina, K. Bazán, H. Avila-George, Classification of cape Gooseberry fruit. according to its level of ripeness using machine learning techniques and different color spaces, IEEE Access 7 (2019) 27389–27400, doi:10.1109/ACCESS.2019.2898223.
[17] T.B. Shahi, C. Sitaula, A. Neupane, W. Guo, Fruit classification using attention-based MobileNetV2 for industrial. applications, PLoS One 17 (2) (2022) e0264586, doi:10.1371/journal.pone.0264586.
[18] N. Saranya, K. Srinivasan, S.K. Kumar, Banana ripeness stage identification: a deep learning approach, J. Ambient Intell. Humaniz. Comput. 13 (8) (2022) 4033–4039, doi:10.1371/journal.pone.0264586.
[19] H. Li, W.S. Lee, K. Wang, Identifying blueberry fruit of different growth stages using natural outdoor color images, Comput. Electron. Agric. 106 (2014) 91–101, doi:10.1371/journal.pone.0264586.
[20] K. Kheiralipour, M. Nadimi, J. Paliwal, Development of an intelligent imaging system for ripeness determination of wild pistachios, Sensors 22 (19) (2022) 7134, doi:10.3390/s22197134.
[21] S.K. Behera, A.K. Rath, P.K. Sethy, Maturity status classification of papaya fruit based on machine learning and transfer learning approach, Inf. Process. Agric. 8 (2) (2021) 244–250, doi:10.1016/j.inpa.2020.05.003.
[22] Y. Al Ohali, Computer vision-based date fruit grading system: design and implementation, J. King Saud Univ. Comput. Inf. Sci. 23 (1) (2011) 29–36, doi:10.1016/j.jksuci.2010.03.003.
[23] Y. Zhang, L. Wu, Classification of fruit using computer vision and a multiclass support vector machine, Sensors 12 (9) (2012) 12489–12505, doi:10.3390/s120912489.
[24] M.S. Hossain, M. Al-Hammadi, G. Muhammad, Automatic fruit classification using deep learning for industrial. applications, IEEE Trans. Ind. Inform. 15 (2) (2018) 1027–1034, doi:10.1109/TII.2018.2875149.
[25] H Liu, C Li, Y Li, YJ Lee, Improved baselines with visual instruction tuning, in: In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 26296–26306.
[26] H. Kaur, B.K. Sawhney, S.K. Jawandha, Evaluation of plum fruit maturity by image processing techniques, J. Food Sci. Technol. 55 (8) (2018) 3008–3015, doi:10.1007/s13197-018-3220-0.
[27] T. Sarkar, A. Mukherjee, K. Chatterjee, Supervised learning aided multiple feature analysis for freshness class detection of Indian gooseberry (Phyllanthus emblica), J. Inst. Eng. India Ser. A 103 (1) (2022) 247–261.
[28] R. Gai, N. Chen, H. Yuan, A detection algorithm for cherry fruit based on the improved YOLO-v4 model, Neural Comput. Appl. (2021) 1–12, doi:10.1007/s00521-021-06029-z.
[29] S. Cárdenas-Pérez, J. Chanona-Pérez, J.V. Méndez-Méndez, G. Calderón-Domínguez, R. López-Santiago, M.J. Perea-Flores, I. Arzate-Vázquez, Evaluation of the ripening stages of apple (Golden Delicious) by means of computer vision system, Biosyst. Eng. 159 (2017) 46–58, doi:10.1016/j.biosystemseng.2017.04.009.
[30] S. Chen, J. Xiong, J. Jiao, Z. Xie, Z. Huo, W. Hu, Citrus fruit maturity detection in natural environments based on. convolutional neural networks and visual saliency map, Precis. Agric. 1-17 (2022) (2022), doi:10.1007/s11119-022-09895-2.
[31] J.K. Bediako, S. Lin, A.K. Sarkar, Y. Zhao, J.W. Choi, M.H. Song, Y.S. Yun, Evaluation of orange peel-derived activated carbons for treatment of dye-contaminated wastewater tailings, Environ. Sci. Pollut. Res. 27 (2020) 1053–1068.
[32] A. Wajid, N.K. Singh, P. Junjun, M.A. Mughal, Recognition of ripe, unripe and scaled condition of orange citrus based on decision tree classification, in: Proceedings of the 2018 International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), IEEE, 2018, pp. 1–4.
[33] Y.Q. Chen, M.S. Nixon, D.W. Thomas, Statistical geometrical features for texture classification, Pattern Recognit. 28 (4) (1995) 537–552.