Journal of Food Measurement and Characterization (2026) 20:4463–4477
https://doi.org/10.1007/s11694-025-03936-w

ORIGINAL PAPER

Classification of Turkish hazelnut (Corylus colurna L.) varieties: a comparative study of YOLOv8 and fine-tuned vision transformer

Yavuz Unal[1] · Elham Tahsin Yasin[2] · Talha Alperen Cengel[3] · Murat Koklu[3]

Received: 23 August 2025 / Accepted: 5 December 2025 / Published online: 9 January 2026

Abstract

Hazelnut is a shrubby plant well-suited to the temperate and humid climate of the Black Sea region, particularly known for its mild winters. Turkey plays a critical role in global hazelnut production. Hazelnuts are not only an essential agricultural product for the country's economy but are also highly valued for their nutritional content, being particularly rich in healthy oils and proteins, and their associated health benefits. The classification of different hazelnut species using modern deep learning techniques aimed to automatically distinguish between eight commonly cultivated hazelnut types: *caklidak*, *damat*, *devedisi*, *sivri*, *karafindik*, *palaz*, *tombul,* and *yagli*. This study employed both YOLOv8 classification models and Vision Transformer (ViT) with fine-tuning to classify a dataset comprising 2,722 labeled images of these hazelnut types. Various configurations of YOLOv8 models were tested, specifically YOLOv8n-cls, YOLOv8s-cls, YOLOv8m-cls, YOLOv8l-cls, and YOLOv8x-cls, alongside ViT-Base/16 with systematic fine-tuning optimization. The results demonstrate that the fine-tuned Vision Transformer achieved the highest classification accuracy of 99.75%, establishing a new state-of-the-art performance for this dataset. Among YOLOv8 models, YOLOv8s-cls and YOLOv8l-cls achieved 99.25% accuracy, while the lightest model, YOLOv8n-cls, recorded 97.38% accuracy. The superior performance of ViT-Base/16 (99.75%) represents a significant advancement over previous studies, demonstrating the effectiveness of transformer-based architectures with transfer learning for agricultural image classification. These findings highlight the strong potential for reliable use in real-world agricultural applications, with the Vision Transformer providing near-perfect classification accuracy. This study shows that advanced computer vision, particularly transformer-based models with fine-tuning, can effectively automate the identification and quality control of agricultural products, enhancing efficiency in food processing and supply chains.

Keywords Hazelnut species · Vision transformer · Turkish hazelnut · YOLOv8, fine tuning

Introduction

Hazelnuts are rich in protein, healthy, and a source of fat and fiber that belongs to the *Betulaceae* family. It is also known to be very rich in vitamins and minerals. Owing to the antioxidants it contains, it is a product that is beneficial to health and supports heart health. In recent years, hazelnut shells have become increasingly popular for the production of activated carbon. Hazelnut shells are used in the production of activated carbon due to the high carbon levels they contain [1, 2]. Hazelnuts, which have an economically important place in the hazelnut food industry, are cultivated worldwide, particularly in regions with a Mediterranean climate. The Turkish hazelnut industry accounts for 80% of global hazelnut exports, making Turkey the world's largest hazelnut producer [3]. Product quality has a direct impact on export earnings and is a significant determinant of hazelnut

✉ Yavuz Unal
yunal@sinop.edu.tr

Elham Tahsin Yasin
ilham.tahsen@gmail.com

Talha Alperen Cengel
talhacengel@gmail.com

Murat Koklu
mkoklu@selcuk.edu.tr

[1] Department of Computer Engineering, Sinop University, Sinop, Türkiye

[2] Department of Computer Engineering, Graduate School of Natural and Applied Sciences, Faculty of Technology, Selcuk University, Konya, Türkiye

[3] Department of Computer Engineering, Faculty of Technology, Selcuk University, Konya, Türkiye

exports. One could argue that the commercial significance of hazelnuts raises interest in characterization and variety standards [4].

YOLOv8's high accuracy rate and fast processing capacity offer significant advantages in the classification of hazelnut species [5]. One of these advantages is the ability to process large datasets in a short time. YOLOv8 is also a suitable model for real-time applications [6]. YOLOv8's architecture and algorithms allow different species to be identified accurately. YOLOv8's advanced algorithms minimize the margin of error by enabling more precise processing of data. In this way, classifications made between different hazelnut species become more reliable. In addition, the flexible structure of the model increases its ability to adapt to different environments and conditions. Because of this, YOLOv8 is a great choice for a variety of agricultural applications [7].

Giraudo et al. [8] automatically detected defective hazelnuts in RGB images using multivariate analysis methods and used them to create classification models. They achieved a 97% rate on the test dataset. Menesatti et al. [9] discussed using shaping techniques to differentiate between Italian hazelnut varieties. The study was conducted on 414 hazelnut images of 4 types of hazelnuts. They achieved a success rate of 98.8%. Gencturk et al. [10] proposed using deep learning algorithms to categorize hazelnuts in their research. The dataset used consists of 1324 Ordu, 1165 Giresun, and 1138 Van Hazelnut variety images. They achieved 100% classification accuracy on this dataset. In their study, Unal and Aktas [3] applied a deep learning-based network structure to separate the "full kernel" of hazelnut from other classes, such as "damaged kernel", "shell", and "small", which are found in small amounts. On a four-class dataset, they trained EfficientNetB0-EfficientNetB1-EfficientNetB2-EfficientNetB3 and InceptionV3 network architectures. With 97.85% in the InceptionV3 network design, the test accuracy was highest. In this study, Harmanci et al. [2] achieved 96.99% and 96.18% classification accuracy with InceptionV3 and DenseNet121 on eight types of hazelnuts and 2722 data, respectively. Bayrakdar et al. [11] demanded to identify the kind and grade of hazelnut in shell by image processing employing size and shape characteristics of hazelnuts. With 84% success, they found the highest rate. Keles and Taner [12] presented the classification of hazelnut varieties via artificial neural networks and discriminant analysis. The physical, mechanical, and optical qualities of 11 hazelnut cultivars were assessed along three axes. The findings obtained were 89.1% and 92.7% for the X-axis, 92.7% and 92.7% for the Y-axis, and 86.8% and 88.7% for the Z-axis, respectively.

Within the scope of the study, a total of 2,722 image data from 8 types of hazelnuts, namely *cakildik*, *damat*, *devedisi*, *karafindik*, *palaz*, *sivri*, *tombul*, and *yagli*, were used. Classification operations were performed on this dataset with YOLOv8n-cls, YOLOv8s-cls, YOLOv8m-cls, YOLOv8l-cls, and YOLOv8x-cls models, along with the Vision Transformer (ViT-Base/16) model using a fine-tuning approach. A total of 6 different model configurations were compared, the confusion matrix findings were analyzed, and the resulting performance measures were obtained. Accuracy and train/loss graphs for every YOLOv8 model and training curves for the ViT model were acquired in model training procedures. Visual results of the confusion matrix showed accurate and inaccurate categorization rates of every variety of hazelnut. Comparative analysis determined which model variants performed better or worse across different evaluation metrics. The methodologies applied, the acquired findings, and the discussion sections are presented separately in the next section of the research.

Materials and methods

A classification procedure was executed on the hazelnut dataset obtained within the scope of the research using YOLOv8 models and Vision Transformer (ViT-Base/16) with fine-tuning. The characteristics of the dataset employed in the research, the methodologies selected for the classification process, and the metrics used in the performance evaluation are discussed. The steps in this study began with preparing a dataset developed by Harmanci et al. [2] consisting of images from eight different types of Turkish hazelnuts. Various versions of the YOLOv8-cls model, along with ViT-Base/16 using a transfer learning approach, were then used to carry out the classification process. Systematic fine-tuning experiments were conducted for the Vision Transformer with different hyperparameter configurations to optimize performance. Following this, the models' performance was assessed using widely accepted evaluation metrics. The roadmap of this study is given in Fig. 1.

Dataset

The images acquired throughout the research are in 3088 × 3088 dimensions and JPG format. The images were taken without flash, at ISO 40 sensitivity and f/1.9 aperture. The images were taken using a Samsung Galaxy A02 digital camera with 13 megapixel resolution from a distance of 10 cm [2]. The dataset used throughout this research comprises a total of 2,722 image data representing eight different varieties of Turkish hazelnuts. Specifically, it contains 354 images of the *Cakildak* type, 399 images each of the *Damat* and *Devedisi* types, and 437 images of *Karafindik*. The *Palaz* variety is represented by 155 images, while the *Sivri* type includes 402 images. Additionally, there are 236 images of

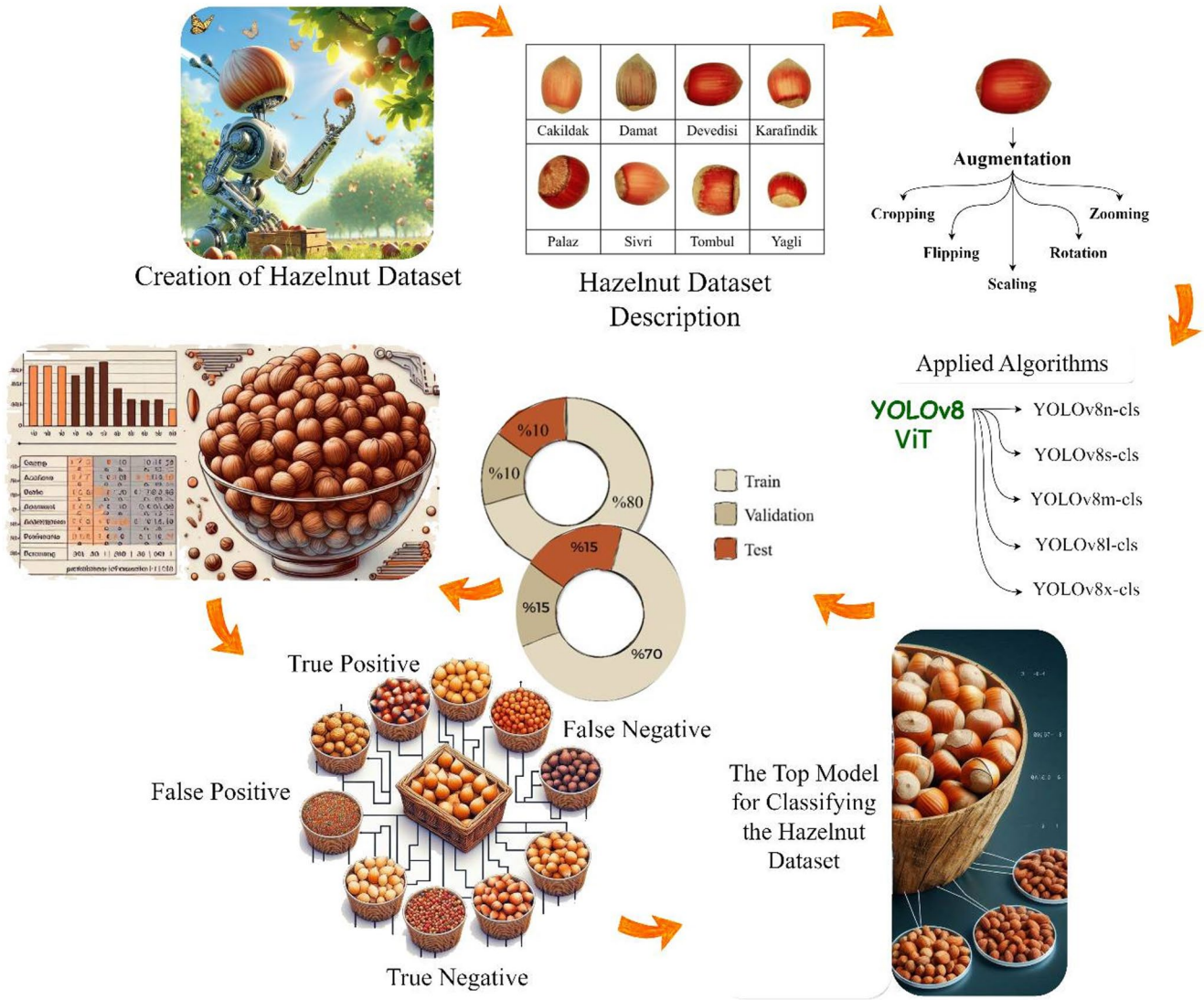

Fig. 1 Turkish hazelnut classification flow diagram

Table 1 Information about the dataset

No	Type of Hazelnut	Total Number of Images
1	Cakildak	354
2	Damat	399
3	Devedisi	399
4	Karafindik	437
5	Palaz	155
6	Sivri	402
7	Tombul	236
8	Yagli	340
∑		**2722**

Tombul and 340 images of *Yagli* hazelnuts. This dataset serves as the foundation for the classification tasks conducted in the research. Information summary about the dataset is shown in Table 1, and sample images are shown in Fig. 2.

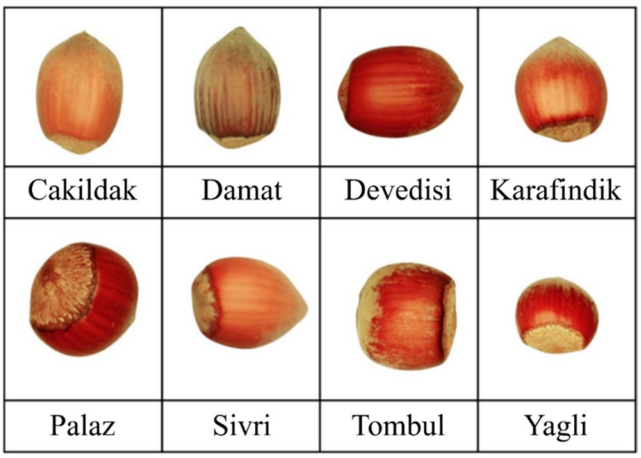

Fig. 2 Sample Images from the Hazelnuts Dataset

Hold-Out method

In machine learning and data mining, the hold-out approach is a widely used data partitioning method to assess a model's performance. This method usually divides the dataset into two main parts, training and testing, but in some cases, it can be divided into three by adding the validation dataset. The hold-out method allocates a certain percentage of the dataset for training the model, while using the rest to evaluate the performance of the model. This approach is used to prevent the model from overfitting during training and to determine how it will perform in real-world data [13, 14].

The study uses a dataset of eight distinct groups of hazelnut data. The dataset was split into three distinct sections: training (80%), validation (10%), and test (10%), thereby assessing the model's performance. Examining the accuracy and improvement capacity of the model depends on this separation of the data [15]. For the Vision Transformer experiments, a different data partitioning strategy was employed. The hazelnut dataset was divided into 70% training, 15% validation, and 15% test sets to optimize the fine-tuning process. This distribution provided sufficient training data for transfer learning while maintaining adequate validation and test sets for robust performance evaluation of the ViT-Base/16 model. The values of the datasets obtained after dividing the dataset into train, validation, and test are shown in Fig. 3.

You only look once (YOLO)

YOLO is a method employing a convolutional neural network (CNN) for real-time object detection. The most important feature of YOLO is that it performs object detection by making a single pass on the image. In this way, it works much faster than other object detection algorithms. Although YOLOv8 is designed for object detection, its output layer can be adapted to the classification task. For this purpose, the last layer of the model has been modified to be suitable only for the classification task [16, 17]. The YOLOv8 models used in the study are described in this section. YOLOv8 architecture [18] shown in Fig. 4.

Data Augmentation techniques increase the robustness and performance of YOLO models by adding diversity to the training data. It also improves the ability of Yolov8 models to generalize to new and previously unseen data. In YOLOv8 models, data augmentation methods are dynamically applied to the original dataset during training. This means that the physical size of the dataset does not change, but different variations are presented to the model at each epoch [19]. A summary of the data augmentation techniques used in this study is shown in Table 2.

YOLOv8n-cls: It is the smallest model. This model is ideal for mobile devices or real-time applications where speed and low hardware requirements are at the forefront. Although its accuracy is slightly lower, its high speed and efficiency provide a significant advantage in environments with resource constraints [5, 20]. *YOLOv8s-cls*: It offers a good balance between speed and accuracy. Its tiny scale qualifies for usage in devices and applications requiring modest resources [6].

YOLOv8m-cls: This model is a medium-sized model that provides sufficient accuracy and speed for more complex tasks. The YOLOv8m-cls model is designed for use in systems with greater computational capacity, such as desktop computers or powerful mobile devices [18, 21]. *YOLOv8l-cls*: It is suitable for tasks requiring high accuracy

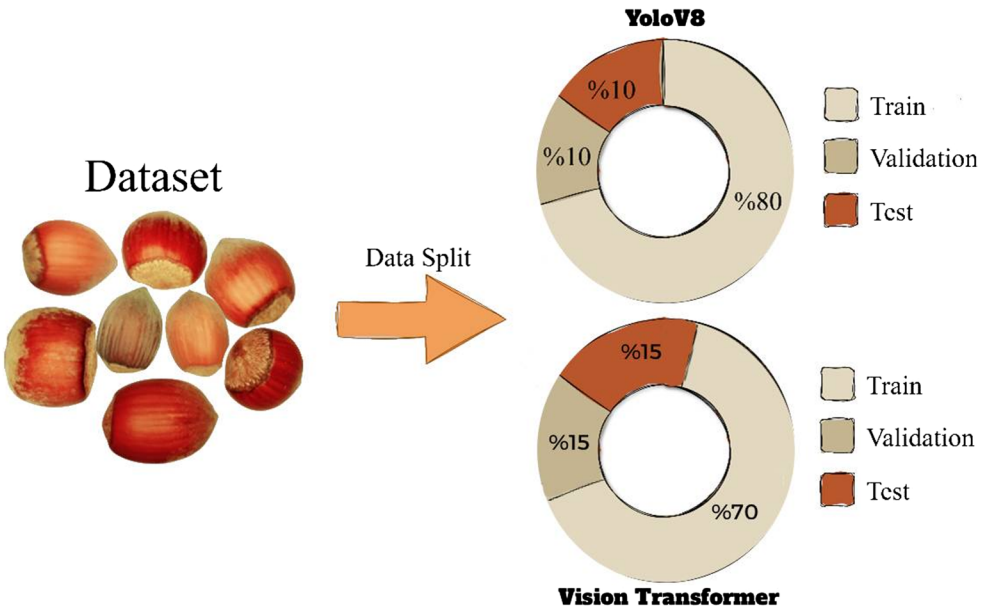

Fig. 3 Hold-out split for the dataset

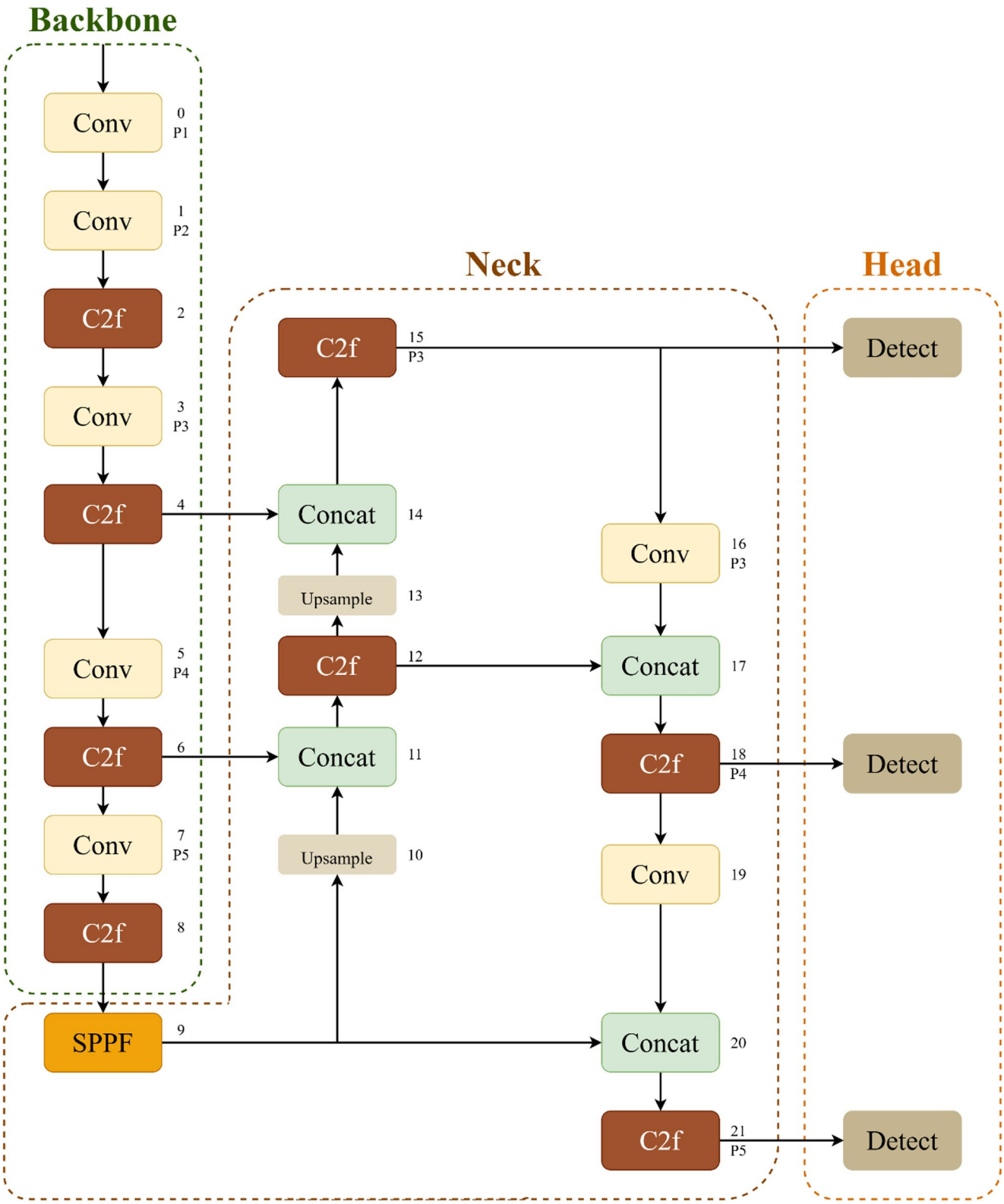

Fig. 4 Architectural design for YOLOv8 model

Table 2 Data augmentation parameters

Parameters	Value	Description
hsv_h	0.015	The hue values of the images will be changed randomly.
hsv_s	0.7	The saturation values of the images will be changed randomly.
hsv_v	0.4	The values of the images will be changed randomly.
translate	0.1	The images will be shifted horizontally and vertically randomly.
scale	0.5	The images will be scaled randomly.
fliplr	0.5	The images will be flipped horizontally with a 50% probability.
erasing	0.4	Some parts of the images will be deleted randomly with a 40% probability.
auto_augment	randaugment	Random augmentation techniques will be applied.
mosaic	1.0	Mosaic augmentation will be used.

due to its large size. It is used in systems with powerful hardware infrastructure. This model has a higher accuracy because it contains more parameters and layers, but it is slightly lower in speed [20, 22]. YOLOv8x-cls It is the most prominent member of the model family and offers the highest level of accuracy. YOLOv8x-cls, which possesses the most parameters and layers, is designed to provide the highest accuracy and also has the highest computational cost and memory requirements [18, 23].

Vision transformer (ViT)

Transformer architecture is used by the deep learning technique Vision Transformer (ViT) for image classification and other visual tasks. Its main characteristic is that it splits an image into patches of a set size, processes them as sequential information, similar to words in a sentence, and assesses them using self-attention mechanisms. Compared to conventional convolution-based methods, this enables it to more successfully capture global relationships between various regions of an image. By altering the output layer, ViT can be tailored to a variety of computer vision tasks, including object detection and segmentation, even though its primary purpose was classification [24]. This section describes the vision transformer model that was employed in this investigation.

To support diverse application needs and computational capacities, the Vision Transformer architecture has been developed in a number of configurations. These variations fall into three primary categories: input resolutions (224 × 224, 384 × 384, 512 × 512), model sizes (Tiny, Small, Base, Large, Huge), and patch sizes (e.g., 16 × 16, 32 × 32). The number of patches that the image is divided into and, consequently, the fineness with which the model can capture local details, depends on the patch size selection. The number of parameters and computational complexity are directly impacted by the size of the model. Conversely, input resolution plays a crucial role in striking a balance between the computational cost and the amount of image detail the model can handle. Vision Transformer can perform at its best in a variety of image classification tasks thanks to its versatility in multi-dimensional configurations [25].

Considering this multidimensional configuration flexibility, the optimal ViT model variant and hyperparameters for hazelnut classification were systematically determined during the fine-tuning process. Although the dataset contains 2,722 images, extensive data augmentation techniques and transfer learning from ImageNet-21k pre-trained weights were used to improve generalization and prevent overfitting.

Fine-tuning strategy

The process of readjusting previously trained deep learning models for a particular task is known as fine-tuning. This method preserves the general feature representations that models trained on large datasets have learned while optimizing the final layers and hyperparameters for the intended task. A subfield of transfer learning called fine-tuning is frequently applied to image classification issues, particularly when datasets are small. Fine-tuning previously trained models on massive datasets like ImageNet lowers computational costs and improves performance in specialized domains like medical image analysis, agricultural product classification, and industrial quality control. This is in contrast to training a model from scratch. In this regard, the Vision Transformer architecture has also shown promising outcomes on a number of image classification tasks using a fine-tuning technique [26].

A systematic approach was adopted during the fine-tuning process of the study, testing all combinations of the configurations shown in Table 3. A total of 27 different model-strategy-augmentation combinations were evaluated, and the configuration that provided the optimal performance balance was determined.

All possible combinations of the three model variants, training strategies, and augmentation approaches resulted in 27 experimental configurations. Through comprehensive fine-tuning experiments, each model-strategy-augmentation combination was systematically evaluated, and performance comparisons were conducted. The experimental results demonstrated that the ViT-Base/16 + Standard training + Light augmentation combination achieved the optimal balance between computational efficiency and classification

Table 3 Vision transformer fine-tuning configuration parameters and experimental settings

Configuration Type	Parameter	Options	Details
Model	ViT-Base/16	google/vit-base-patch16-224-in21k	224 × 224, patch 16 × 16
	ViT-Base/32	google/vit-base-patch32-224-in21k	224 × 224, patch 32 × 32
	ViT-Large/16	google/vit-large-patch16-224-in21k	224 × 224, patch 16 × 16, large model
Training Strategy	Standard	LR: 5×10^{-5}, Epoch: 8, Batch: 16	Weight Decay: 0.01
	Aggressive	LR: 1×10^{-4}, Epoch: 12, Batch: 12	Weight Decay: 0.001
	Conservative	LR: 2×10^{-5}, Epoch: 15, Batch: 20	Weight Decay: 0.02
Data Augmentation	Light	RandomHorizontal-Flip, RandomRotation, ColorJitter	p=0.3, 5°, brightness=0.1
	Heavy	RandomVertical-Flip, RandomAffine, RandomPerspective	p=0.5, 15°, brightness=0.3
	Medical	GaussianBlur, medical image focused	blur kernel=3, 10° rotation

performance, delivering the most successful outcome with 99.75% accuracy. This selection was determined based on the criteria of maintaining reasonable model complexity while achieving superior classification performance.

Confusion matrix

A classification model may be evaluated using a confusion matrix, which is a table. By looking at this table, it is possible to observe the cases where the relevant model classifies correctly and incorrectly. Thus, error types can be analyzed, and the model's performance is a statistic that is simple to measure [27] [28]. A confusion matrix is generally a four-cell table for two-class classification problems, but it can also be expanded for multiclass problems [29]. The sample confusion matrix used in this study is presented in Fig. 5.

Performance metrics

Performance metrics are quantitative values applied to assess the model's performance, used at the end of the process. Performance metrics may vary for classification, regression, clustering, and other machine learning tasks [32, 33].

Accuracy

accuracy measures whether the class that the model predicts with the highest probability is correct. If the category with the greatest likelihood forecasted by the model is the same as the true class, it is accepted as correct [34, 35]. The equation for accuracy is given in Eq. 1. Besides the accuracy evaluation, other metrics, including recall, F1-score, precision, kappa, and Matthew's correlation coefficient, were calculated.

$$Accuracy = \frac{Number\ of\ Correct\ Predictions}{Total\ Number\ of\ Predictions} \times 100 \quad (1)$$

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

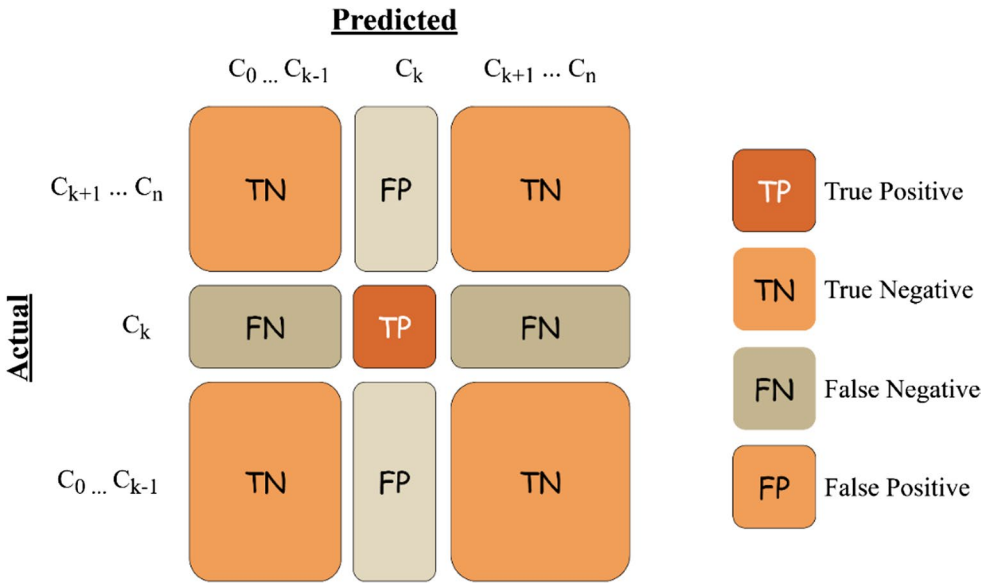

Fig. 5 The confusion matrix used for multiclass dataset [30]. True positive (TP) Cases that the model appropriately foresaw as a given type of hazelnut. From Fig. 5. False positive (FP) Illustrations of incidents that the model falsely forecast as a given type of hazelnut but are not actually that type of hazelnut. True negative (TN) Cases that the model correctly predicted as a given type of hazelnut but are not actually that type of hazelnut. False negative (FN) Cases that the model falsely predicted as a given type of hazelnut, but are not actually that type of hazelnut [31]

$$Cohen's\ Kappa = \frac{P_o - P_e}{1 - P_e} \tag{5}$$

$$Matthews\ Correlation\ Coefficient = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{6}$$

The model's classification performance at various accuracy levels is assessed using the accuracy metric. The *accuracy* of the model's highest probability prediction is measured by accuracy. This statistic is used to comprehend and evaluate the model's performance, particularly in challenges involving multi-class categorization [36, 37].

Train Loss calculates the model's error on the training set. For every training example, the difference between the expected and actual results is computed during model training.

Validation Loss measures the error made by the model in the validation dataset. The validation dataset is used to assess the model's capacity for improvement and includes cases that the model has never encountered during training.

Out of all the times the model said "yes," *precision* tells us how often it was actually right. Of all the actual "yes" cases, *recall* represents how many the model accurately found. The *F1-score* balances recall and precision to give one overall performance score.

Cohen's Kappa is a useful metric for evaluating classification models, as it measures how much agreement exists between predicted and actual labels while factoring in chance agreement. Unlike basic accuracy, which can be misleading in imbalanced datasets or when class overlap occurs, Cohen's Kappa provides a more balanced view of model performance. This makes it especially valuable when assessing classifiers in complex scenarios where standard metrics like precision or recall may fall short [30].

The *Matthew's Correlation Coefficient (MCC)* is a statistic used to assess how well binary classification models perform. Its values vary from − 1 to + 1, where + 1 denotes a perfect prediction, 0 denotes performance that is no better than imagining, and − 1 denotes predictions that are entirely at odds with the labels. What makes MCC especially valuable is its ability to handle situations where the dataset has an uneven distribution between classes. Unlike simpler metrics, MCC takes all parts of the confusion matrix into account, including true positives, true negatives, false positives, and false negatives, giving a more balanced view of a model's accuracy [38].

Experimental results

The results of the classification of hazelnut species with YOLOv8 are presented. The YOLOv8-cls models YOLOv8n-cls, YOLOv8s-cls, YOLOv8m-cls, YOLOv8-cls, YOLOv8l-cls, and YOLOv8x-cls were used to classify the Turkish hazelnut dataset across 50 epochs, respectively. The reason for leaving it at 50 epochs is that the models give the same value after a certain epoch. The results of the performance metrics are presented in Table 4, the confusion matrix findings are illustrated in Fig. 6, and the graphical representation of the train-validation loss is shown in Fig. 7.

According to the data in Table 4, the accuracy rates of the YOLOv8 algorithm among different dimensions are as follows: YOLOv8-n 97.38%, YOLOv8-s and YOLOv8-l 99.25%, YOLOv8-m 98.88%, and YOLOv8-x 98.13%. While YOLOv8-s and YOLOv8-l achieve the highest accuracy rate of 99.25%, the lowest accuracy rate of 97.38% is seen in YOLOv8-n. These results show that the YOLOv8 algorithm has high accuracy rates in general, and especially YOLOv8-s and YOLOv8-l dimensions show the best performance.

When the confusion matrices showing the classification performance of various versions of the YOLOv8 algorithm are examined, the YOLOv8n-cls model makes 260 correct predictions and eight incorrect predictions (Fig. 6a). The YOLOv8s-cls model makes 266 correct predictions and two incorrect predictions (Fig. 6b). The YOLOv8m-cls model makes 264 correct predictions and four incorrect predictions (Fig. 6c). The YOLOv8l-cls model makes 266 correct predictions and two incorrect predictions (Fig. 6d). Finally, the YOLOv8x-cls model makes 263 correct predictions and five incorrect predictions (Fig. 6e). When these results are examined, it is observed that there are generally high accuracy rates and minor errors between some classes.

It is observed that all models generally make correct classifications in the classes "Damat", "Devecisi", "Karafindik", "Sivri", "Tombul", "Yagli", and "Cakildak". It is observed that, especially in the classes "Damat", "Devecisi", "Karafindik", "Sivri", "Yagli", and "Cakildak", there are very few incorrect classifications. However, errors related to the "Karafindik" class are observed in some models. It is observed that the "Karafindik" class is sometimes confused with the "Palaz" or "Cakildak" classes in the YOLOv8n, and YOLOv8m models. This indicates that the features of

Table 4 Performance metric results of YOLOv8-cls models

Algorithm	Accuracy (%)	Precision	Recall	F1-Score	Kappa Score	MCC
YOLOv8n-cls	97.38	0.972	0.960	0.957	0.963	0.962
YOLOv8s-cls	99.25	0.993	0.993	0.993	0.991	0.991
YOLOv8m-cls	98.88	0.987	0.987	0.987	0.984	0.984
YOLOv8l-cls	99.25	0.993	0.994	0.993	0.993	0.993
YOLOv8x-cls	98.13	0.976	0.976	0.976	0.978	0.978

Yolov8n-cls		Predicted							
		Damat	Devedişi	Karafındık	Palaz	Sivri	Tombul	Yağlı	Çakıldak
Actual	Damat	39	0	0	0	0	0	0	0
	Devedişi	0	34	0	0	0	0	0	0
	Karafındık	0	0	43	1	1	0	0	0
	Palaz	0	0	0	13	0	0	0	0
	Sivri	0	0	0	0	39	0	0	0
	Tombul	0	2	0	0	0	23	0	0
	Yağlı	0	0	0	1	0	0	34	0
	Çakıldak	0	3	0	0	0	0	0	35

(a)

Yolov8s-cls		Predicted							
		Damat	Devedişi	Karafındık	Palaz	Sivri	Tombul	Yağlı	Çakıldak
Actual	Damat	39	0	0	0	0	0	0	0
	Devedişi	0	38	0	0	0	0	0	0
	Karafındık	0	0	43	0	1	0	0	0
	Palaz	0	0	0	15	0	0	0	0
	Sivri	0	0	0	0	39	0	0	0
	Tombul	0	0	0	0	0	23	0	0
	Yağlı	0	0	0	0	0	0	34	0
	Çakıldak	0	1	0	0	0	0	0	35

(b)

Yolov8m-cls		Predicted							
		Damat	Devedişi	Karafındık	Palaz	Sivri	Tombul	Yağlı	Çakıldak
Actual	Damat	39	0	0	0	0	0	0	0
	Devedişi	0	37	0	0	0	0	0	0
	Karafındık	0	0	42	0	1	0	0	0
	Palaz	0	0	0	15	0	0	0	0
	Sivri	0	0	1	0	39	0	0	0
	Tombul	0	0	0	0	0	23	0	0
	Yağlı	0	0	0	0	0	0	34	0
	Çakıldak	0	2	0	0	0	0	0	35

(c)

Yolov8l-cls		Predicted							
		Damat	Devedişi	Karafındık	Palaz	Sivri	Tombul	Yağlı	Çakıldak
Actual	Damat	39	0	0	0	0	0	0	0
	Devedişi	0	39	0	0	0	0	0	1
	Karafındık	0	0	43	0	1	0	0	0
	Palaz	0	0	0	15	0	0	0	0
	Sivri	0	0	0	0	39	0	0	0
	Tombul	0	0	0	0	0	23	0	0
	Yağlı	0	0	0	0	0	0	34	0
	Çakıldak	0	0	0	0	0	0	0	34

(d)

Yolov8x-cls		Predicted							
		Damat	Devedişi	Karafındık	Palaz	Sivri	Tombul	Yağlı	Çakıldak
Actual	Damat	39	0	0	0	0	0	0	0
	Devedişi	0	38	0	0	0	0	0	0
	Karafındık	0	0	42	1	1	0	0	0
	Palaz	0	0	0	14	0	0	1	0
	Sivri	0	0	1	0	39	0	0	0
	Tombul	0	0	0	0	0	23	0	0
	Yağlı	0	0	0	0	0	0	33	0
	Çakıldak	0	1	0	0	0	0	0	35

(e)

Fig. 6 Confusion matrix results of YOLOv8-cls models

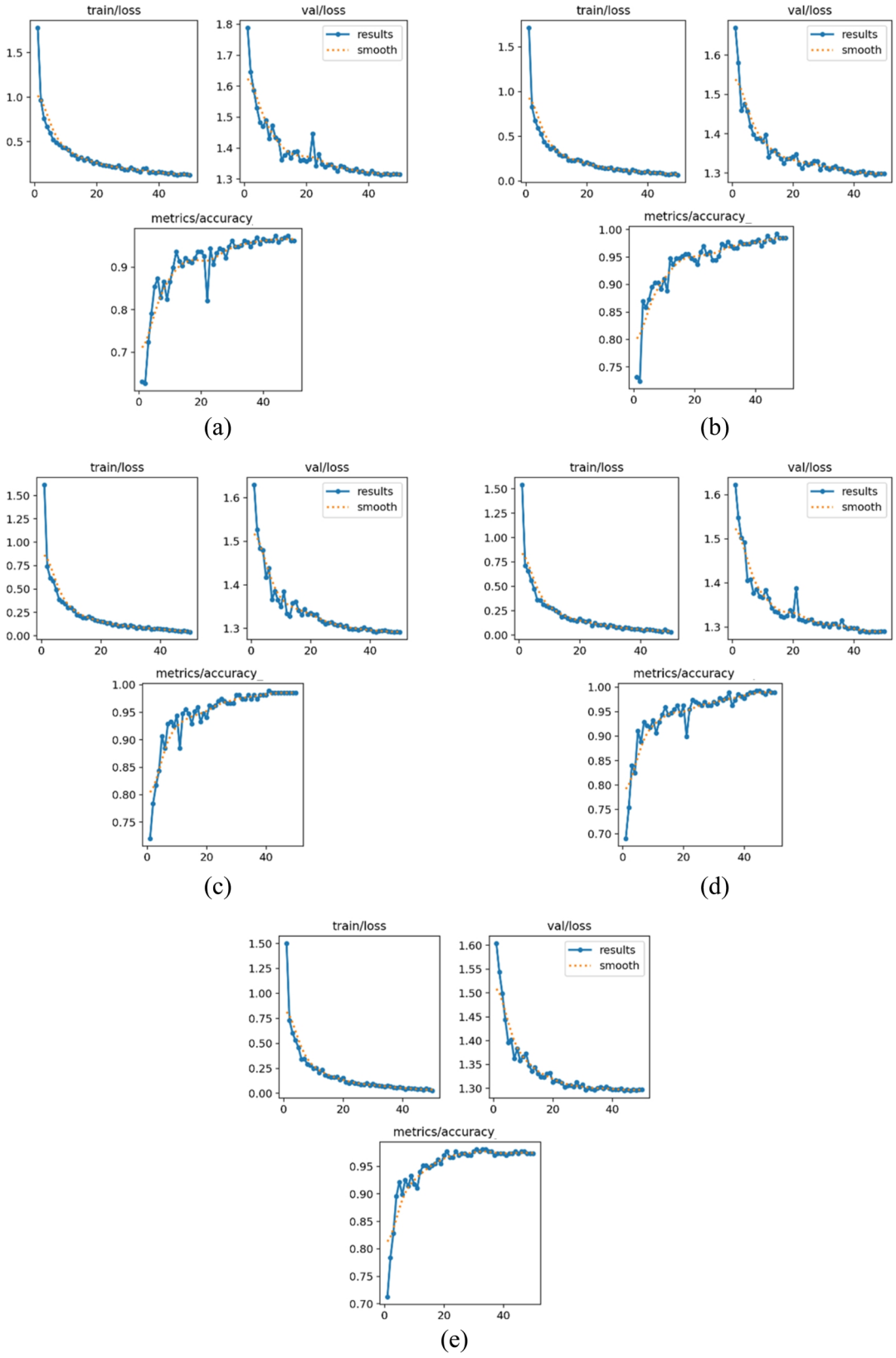

Fig. 7 Graphs obtained from Yolov8 algorithms (**a**) YOLOv8n-cls, (**b**) YOLOv8s-cls, (**c**) YOLOv8m-cls, (**d**) YOLOv8l-cls, (**e**) YOLOv8x-cls

these two classes may overlap in some cases, and the model struggles to make this distinction.

It compares the training and validation losses and accuracies of different YOLOv8 models, namely YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, and YOLOv8x. In general, it is observed that the losses decrease, and the accuracy increases throughout the training process for all models. YOLOv8n and YOLOv8s reach lower losses faster and show early stability, but the larger models YOLOv8l and YOLOv8x train for a longer time and reach higher accuracies. Although the training and validation accuracy graphs show some sudden drops, the general trend shows that accuracy increases, and the models get better.

The results of hazelnut species classification using ViT with fine-tuning are presented. For the hazelnut dataset classification, the ViT-Base/16, ViT-Base/32, and ViT-Large/16 models were fine-tuned for eight epochs using different training strategies and data augmentation techniques. The reason for limiting the number of epochs to eight is that the models achieve the same performance after a certain number of epochs. The optimal hyperparameter values determined as a result of the fine-tuning process are presented in Table 5. This configuration demonstrated the highest performance with 99.75% accuracy among 27 different combinations.

This hyperparameter configuration optimized the balance between model complexity and performance, achieving the highest classification accuracy on the hazelnut dataset. In particular, the combination of the cosine learning rate scheduler and the AdamW optimizer was observed to play a critical role in the model's rapid convergence. Detailed classification performance metrics obtained for each hazelnut type by the ViT model trained with the optimal hyperparameter configuration are shown in Table 6.

As seen in Table 6, the model resulting from the fine-tuning process demonstrated high classification performance across all hazelnut species. Excellent precision and recall values (100%) were achieved in seven of the eight classes, with only a minimal performance decrease observed in the Cakildak class.

The changes in training loss, validation loss, and validation accuracy observed during the fine-tuning process are presented in Fig. 8.

As seen in Fig. 8, the model demonstrated rapid convergence during the training process and reached over 99% validation accuracy by the third epoch. The decrease in the training loss value from 2.0 to 0.06, and the similar trend in the validation loss, demonstrate that the model has successfully learned. The increase in validation accuracy from 80% to 99.8% demonstrates the effectiveness of the fine-tuning strategy.

The classification accuracy of the fine-tuned model for each hazelnut type is shown in Fig. 9 using a confusion matrix.

Table 5 Optimal hyperparameter configuration for vision transformer fine-tuning on Turkish hazelnut classification

Parameter	Value	Description
Model Architecture	ViT-Base/16	google/vit-base-patch16-224-in21k
Input Resolution	224 × 224	Image input size
Patch Size	16 × 16	Vision Transformer patch dimensions
Learning Rate	5×10^{-5}	Initial learning rate
Learning Rate Scheduler	Cosine	Cosine annealing schedule
Optimizer	AdamW	Adaptive moment estimation with weight decay
Batch Size	16	Training batch size
Validation Batch Size	16	Evaluation batch size
Epochs	8	Total training epochs
Weight Decay	0.01	L2 regularization parameter
Warmup Steps	300	Linear warmup for learning rate
Data Augmentation	Light	Random Horizontal Flip (p = 0.3), Random Rotation (5°), Color Jitter (0.1)
Evaluation Strategy	Epoch	Evaluation is performed every epoch
Save Strategy	Epoch	The model checkpoint is saved every epoch
Metric for Best Model	Accuracy	Model selection criterion
Early Stopping	Disabled	Training completed for whole epochs
Seed	43	Random seed for reproducibility
Device	Cuda	Training hardware
Mixed Precision	Disabled	Full precision training

The rapid convergence of the ViT model is attributed to the use of pre-trained ImageNet-21k weights and the AdamW optimizer with a cosine learning-rate schedule, which enabled stable adaptation in fewer epochs.

Examining the confusion matrix in Fig. 9, it is seen that seven of the eight hazelnut species (Yagli, Damat, Devedisi, Karafindik, Palaz, Sivri, and Tombul) exhibit excellent classification performance. Only the Cakildak class exhibits a minimal error rate, with 52 of 53 samples correctly classified. This result demonstrates that the fine-tuning process provides effective learning for all hazelnut species and that inter-class confusion is virtually nonexistent.

According to the data in Table 7, Harmanci et al. achieved 96.18% and 96.99% accuracy values with DenseNet121 and Inceptionv3 models, respectively. In this study, the accuracy rates of different dimensions of the YOLOv8 algorithm are as follows: YOLOv8-n 97.38%, YOLOv8-s and YOLOv8-l 99.25%, YOLOv8-m 98.88%, and YOLOv8-x 98.13%. Additionally, the ViT-Base/16 (Fine-tuned) model achieved the highest accuracy rate of 99.75%, outperforming all other models tested. It is observed that both YOLOv8-cls models

Table 6 Classification performance metrics of the optimal vision transformer model

	Precision	Recall	F1-score	Support
Cakildak	0.9815	1.0000	0.9907	53
Yagli	1.0000	1.0000	1.0000	51
Devedisi	1.0000	0.9833	0.9916	60
Karafindik	1.0000	1.0000	1.0000	65
Palaz	1.0000	1.0000	1.0000	23
Sivri	1.0000	1.0000	1.0000	60
Tombul	1.0000	1.0000	1.0000	36
Accuracy			0.9975	408
Macro avg	0.9977	0.9979	0.9978	408
Weighted avg	0.9976	0.9976	0.9976	408

and the fine-tuned Vision Transformer obtain significantly higher results than the study of Harmanci et al., who worked on the same dataset. While ViT-Base/16 demonstrates the superior performance with 99.75% accuracy, YOLOv8-s and YOLOv8-l follow closely with 99.25%, and the lowest accuracy rate is seen in YOLOv8-n with 97.38%. These results reveal that modern deep learning architectures, notably Vision Transformers with fine-tuning, achieve exceptional accuracy rates in hazelnut classification, with ViT-Base/16 establishing the new state-of-the-art performance on this Turkish hazelnut dataset.

Conclusion and discussion

Upon examining the obtained results, it is evident that the YOLOv8n-cls model achieves the lowest classification accuracy, with 97.38%. This model can be said to have a lower accuracy rate due to its simpler structure compared to other models. The YOLOv8s-cls and YOLOv8l-cls models achieve 99.25% accuracy in 50 epochs. In comparison, the ViT-Base/16 (Fine-tuned) model demonstrates superior performance with 99.75% accuracy in just eight epochs, establishing the new state-of-the-art result for this dataset. The exceptional performance of the Vision Transformer can be attributed to its attention mechanism and effective transfer learning through fine-tuning on pre-trained ImageNet-21k weights.

The obtained results provide valuable insights for model selection in practical applications. While lighter models such as YOLOv8n-cls can be preferred when fast inference is needed, ViT-Base/16 (Fine-tuned) should be considered when the highest accuracy is paramount, despite requiring more computational resources. However, the trade-off between accuracy and computational efficiency should be carefully evaluated based on specific application requirements.

Among all tested models, ViT-Base/16 (Fine-tuned) achieved the most consistent and highest performance with

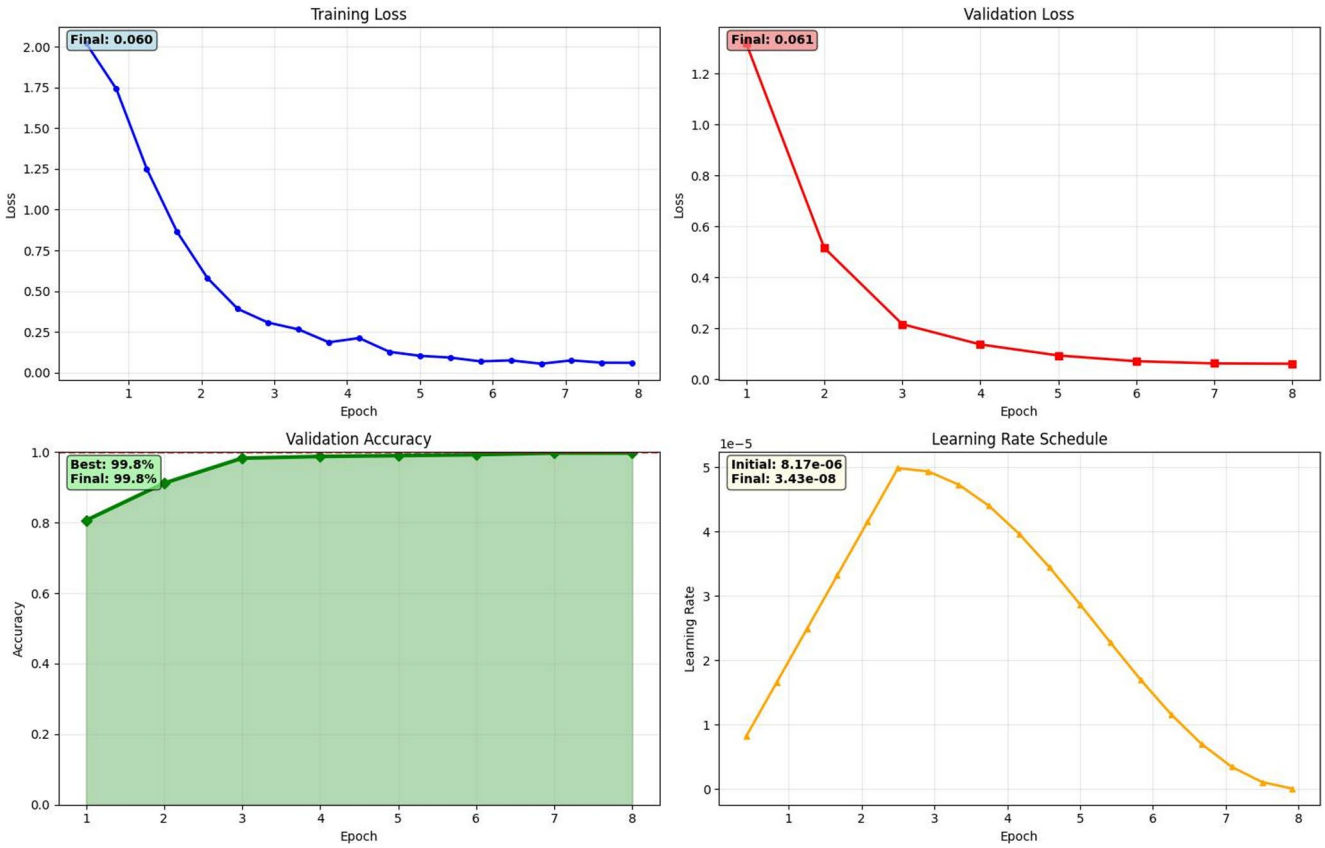

Fig. 8 Training curves showing the convergence of the Vision Transformer model during fine-tuning on the Turkish hazelnut dataset

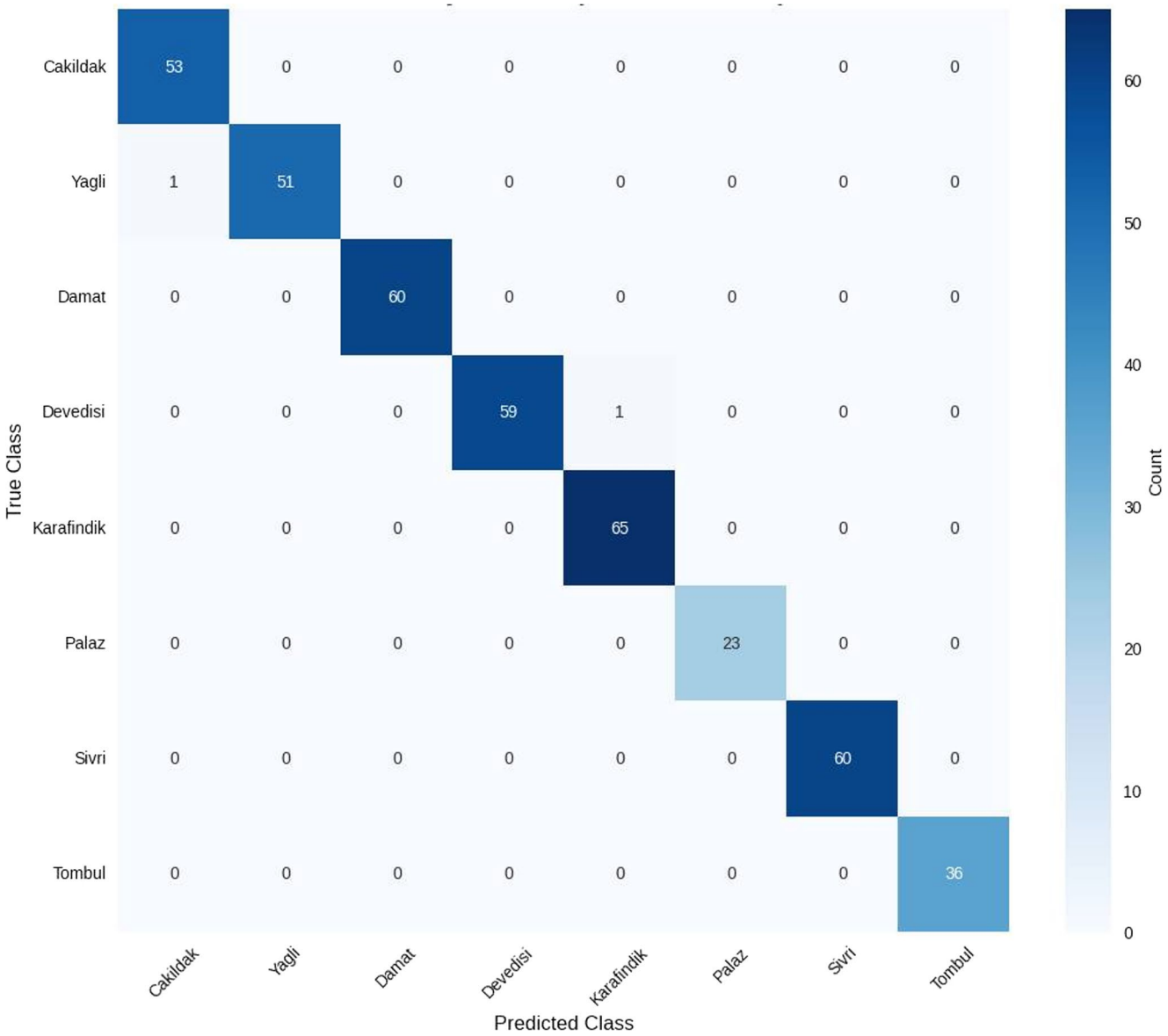

Fig. 9 Confusion matrix of the fine-tuned ViT-Base/16 model

Table 7 Performance metric results of studies obtained in this study and studies conducted with the same dataset

Models	Accuracy (%)	References
DenseNet121	96.18	(Harmanci et al., 2023)
Inceptionv3	96.99	(Harmanci et al., 2023)
YOLOv8n-cls	97.38	This Study
YOLOv8s-cls	99.25	
YOLOv8m-cls	98.88	
YOLOv8l-cls	99.25	
YOLOv8x-cls	98.13	
ViT-Base/16 (Fine-tuned)	99.75	This Study

minimal classification errors across all hazelnut varieties. The confusion matrix analysis revealed that the Vision Transformer successfully distinguished between all eight hazelnut types with only one misclassification out of 408 test samples. This superior discrimination capability demonstrates the effectiveness of transformer-based architectures for fine-grained image classification tasks.

When comparing with the study of Harmanci et al. using the same dataset [2], their most successful result with Inception v3 was 96.99%. In our study, both YOLOv8s-cls/YOLOv8l-cls (99.25%) and especially ViT-Base/16 (Fine-tuned) (99.75%) achieved significantly more successful results, with the Vision Transformer representing a 2.76% point improvement over the previous best result.

The implementation of ViT-based classification systems can bring significant improvements to many areas, from agricultural production to the food industry. Farmers can

quickly and accurately identify hazelnut species with near-perfect accuracy and develop maintenance and harvesting strategies suitable for different varieties. Food processing facilities can classify hazelnut species with 99.75% reliability and ensure that high-quality products are used in production processes. The exceptional accuracy of ViT-Base/16 enables hazelnut traders to optimize marketing strategies and determine market values more precisely. R&D units can investigate genetic differences of hazelnut varieties using these highly accurate classification models for new product development studies. Future work should focus on expanding the dataset and exploring ensemble methods combining Vision Transformers with other high-performing architectures to enhance classification accuracy further.

The main contribution of this study is the systematic fine-tuning design that explored multiple ViT configurations and established a new state-of-the-art accuracy of 99.75% for Turkish hazelnut classification.

Acknowledgements We express our thanks to the Scientific Project Coordinator at Selcuk University for their substantial assistance and support during this project.

Authors' contributions E. T. Y.: Conceptualization, Methodology, Formal Analysis, Visualization, Software, Writing – Original Draft.
T. A. C.: Data Preprocessing, Model Training and Evaluation, Writing – Original Draft.
Y. U.: Data Collection, Dataset Curation, Image Labeling, Model Implementation, Data Preprocessing, Software, Writing-Original Draft.
M. K.: Supervision, Project Administration, Validation, Writing – Review & Editing.

Funding The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Data availability The datasets produced and analyzed during this study can be obtained from the corresponding author and this link: http://citedata.com/Turkish_Hazelnut_Dataset.zip.

Declarations

Competing interest The authors declare no financial conflicts of interest related to this manuscript.

References

1. J.S. Amaral, S. Casal, I. Citová, A. Santos, R.M. Seabra, B.P.P. Oliveira, Eur. Food Res. Technol. **222**, 274 (2006). https://doi.org/10.1007/s00217-005-0068-0
2. S. Harmanci, Y. Ünal, B. Ateş, Intell. Methods Eng. Sci. **2**, 58 (2023). https://doi.org/10.58190/imiens.2023.14
3. Z. Ünal, H. Aktaş, Postharvest Biol. Technol. **197**, 112225 (2023). https://doi.org/10.1016/j.postharvbio.2022.112225
4. A. Taner, Y.B. Öztekin, H. Duran, Sustainability. **13**, 6527 (2021). https://doi.org/10.3390/su13126527
5. Q. Liu, W. Huang, X. Duan, J. Wei, T. Hu, J. Yu, J. Huang, Electronics. **12**, 3892 (2023). https://doi.org/10.3390/electronics12183892
6. M. Ma, H. Pang, Appl. Sci. **13**, 8161 (2023). https://doi.org/10.3390/app13148161
7. H. Duong-Trung, N. Duong-Trung, E.A.I. Endorsed Trans, Ind. Net Intel Syst. **11** (2024). https://doi.org/10.4108/eetinis.v11i1.4618
8. A. Giraudo, R. Calvini, G. Orlandi, A. Ulrici, F. Geobaldo, F. Savorani, Food Control. **94**, 233 (2018). https://doi.org/10.1016/j.foodcont.2018.07.018
9. P. Menesatti, C. Costa, G. Paglia, F. Pallottino, S. D'Andrea, V. Rimatori, J. Aguzzi, Biosyst. Eng. **101**, 417 (2008). https://doi.org/10.1016/j.biosystemseng.2008.09.013
10. B. Gencturk, S. Arsoy, Y.S. Taspinar, I. Cinar, R. Kursun, E.T. Yasin, M. Koklu, Eur. Food Res. Technol. **250**, 97 (2024). https://doi.org/10.1007/s00217-023-04369-9
11. S. Bayrakdar, B. Comak, D. Basol, I. Yucedag, in 2015 23nd Signal Processing and Communications Applications Conference (SIU) (IEEE, Malatya, Turkey, 2015), pp. 616–619. https://doi.org/10.1109/SIU.2015.7129899
12. O. Keles, A. Taner, Span. J. Agric. Res. **19**, e0211 (2021). https://doi.org/10.5424/sjar/2021193-17068
13. S. Yadav, S. Shukla, I. Computing, 2016)in, pp. 78–83. https://doi.org/10.1109/IACC.2016.25
14. Y.W. Nam, Y. Arai, T. Kunizane, A. Koizumi, Water Supply. **21**, 3477 (2021). https://doi.org/10.2166/ws.2021.109
15. T.A. Cengel, B. Gencturk, E.T. Yasin, M.B. Yildiz, I. Cinar, M. Koklu, J. Food Sci. **90**, e17553 (2025). https://doi.org/10.1111/1750-3841.17553
16. N. Al Mudawi, A.M. Qureshi, M. Abdelhaq, A. Alshahrani, A. Alazeb, M. Alonazi, A. Algarni, Sustainability. **15**, 14597 (2023). https://doi.org/10.3390/su151914597
17. J. Terven, D.-M. Córdova-Esparza, J.-A. Romero-González, MAKE **5**, 1680 (2023). https://doi.org/10.3390/make5040083
18. O. Kılcı, M. Koklu, Food Anal. Methods. **18**, 815 (2025). https://doi.org/10.1007/s12161-025-02755-5
19. F. Chen, M. Deng, H. Gao, X. Yang, D. Zhang, IET Image Process. **18**, 1915 (2024). https://doi.org/10.1049/ipr2.13073
20. A. Çınar, Joda. **0**, 1 (2024). https://doi.org/10.1177/08953996241308770
21. Z. Wang, G. Yuan, H. Zhou, Y. Ma, Y. Ma, Appl. Sci. **13**, 12775 (2023). https://doi.org/10.3390/app132312775
22. E. Kahya, F.F. Özdüven, B.C. Ceylan, (2023). https://doi.org/10.5281/ZENODO.8320097
23. H. Jiang, B. Gilbert. Murengami, L. Jiang, C. Chen, C. Johnson, F. Auat. Cheein, S. Fountas, R. Li, L. Fu, Comput. Electron. Agric. **219**, 108795 (2024). https://doi.org/10.1016/j.compag.2024.108795
24. D. Ghosh, S. Ghosh, N. Dulal, S. Biswas, R. Mallipeddi, Comput. Electron. Agric. **238** (2025). https://doi.org/10.1016/j.compag.2025.110824
25. M. Srivastava, V. Sisaudia, J. Meena, Expert Syst. Appl. **294**, 128793 (2025). https://doi.org/10.1016/j.eswa.2025.128793
26. P. Mhala, A. Bilandani, S. Sharma, Expert Syst. Appl. **267**, 126066 (2025). https://doi.org/10.1016/j.eswa.2024.126066
27. S.K.S. Al-doori, Y.S. Taspınar, M. Koklu, Int. J. Appl. Math. Electron. Computers. **9**, 116 (2021). https://doi.org/10.18100/ijamec.1035749
28. K. Tutuncu, I. Cinar, R. Kursun, M. Koklu, M. 11 Computing, 2022) in, pp. 1–4. https://doi.org/10.1109/MECO55406.2022.9797212
29. I. Markoulidakis, G. Kopsiaftis, I. Rallis, I. Georgoulas, in Proceedings of the 14th PErvasive Technologies Related to Assistive Environments ConferenceACM, Corfu Greece, (2021), pp. 412–419. https://doi.org/10.1145/3453892.3461323

30. E.T. Yasin, M. Erturk, M. Tassoker, M. Koklu, Clin. Oral Invest. **29**, 203 (2025). https://doi.org/10.1007/s00784-025-06285-6
31. T.A. Cengel, B. Gencturk, E.T. Yasin, M.B. Yildiz, I. Cinar, M. Koklu, Appl. Fruit Sci. **66**, 2123 (2024). https://doi.org/10.1111/1750-3841.17553
32. M. Koklu, M.F. Unlersen, I.A. Ozkan, M.F. Aslan, K. Sabanci, Measurement. **188**, 110425 (2022). https://doi.org/10.1016/j.measurement.2021.110425
33. H. Isik, S. Tasdemir, Y.S. Taspinar, R. Kursun, I. Cinar, A. Yasar, E.T. Yasin, M. Koklu, Food Sci. Nutr. **12**, 786 (2024). https://doi.org/10.1002/fsn3.3783
34. A. Wongpanich, H. Pham, J. Demmel, M. Tan, Q. Le, Y. You, S. Kumar, in 2021 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)IEEE, Portland, OR, USA, (2021), pp. 947–950. https://doi.org/10.1109/IPDPSW52791.2021.00146
35. S. Gerz, T.A. Cengel, T. Ozturk, B. Gencturk, E. Boz, E. Yasin, M. Koklu, Imiens. **2** (2024). https://doi.org/10.58190/imiens.2024.98
36. Z. He, B. Gong, D. Fan, in 2019 IEEE Winter Conference on Applications of Computer Vision (WACV)IEEE, Waikoloa Village, HI, USA, (2019), pp. 913–921. https://doi.org/10.1109/WACV.2019.00102
37. M.N. Örnek, H. Kahramanlı Örnek, Food Measure. **15**, 3471 (2021). https://doi.org/10.1007/s11694-021-00923-9
38. B. Yılmaz, Int. J. Food Sci. **2025**, 9677985 (2025). https://doi.org/10.1155/ijfo/9677985