Applied Fruit Science (2025) 67:102
https://doi.org/10.1007/s10341-025-01327-5

ORIGINAL ARTICLE

Webserver-Based Mobile Application for Multi-class Chestnut (*Castanea sativa*) Classification Using Deep Features and Attention Mechanisms

Mustafa Yurdakul[1] · **Kübra Uyar**[2] · **Şakir Taşdemir**[3]

Received: 1 October 2024 / Accepted: 31 March 2025 / Published online: 5 May 2025

Abstract
Chestnut (*Castanea sativa*) is a nutritious food with fiber, vitamins C and B group, minerals such as potassium, magnesium, and iron. In addition to being a nutritious food, chestnuts are used in various fields such as medicine, cosmetics, and energy. All the mentioned characteristics make it a demanded product worldwide. To determine the market price of chestnuts, it is necessary to have a good classification. In traditional approaches, producers classify chestnuts according to their external appearance; however, this is tedious, time-consuming, and prone to errors. There is a need for computer-aided systems to analyze the chestnut varieties. Therefore, a camera system was set up and images of chestnuts belonging to 'Alandız', 'Aydın', 'Simav', and 'Zonguldak' varieties were captured to create a novel dataset. Moreover, a deep-based mobile application was developed to classify chestnut types. After testing the 16 state-of-the-art convolutional neural network (CNN) models, the three most successful models from the 16 CNN models were used as feature extractors, and the extracted features were classified using Decision Tree (DT), Naive Bayes (NB), Support Vector Machine (SVM), Adaboost, and Xtreme Gradient Boosting (XGB) algorithms. Finally, attention modules were integrated to CNN models to enhance accurate classification of chestnut images. The highest result achieved by MobileNet with attention mechanism was accuracy of 99.65%, precision of 99.62%, recall of 99.67%, f1-score of 99.64%, kappa score of 100%, and area under the curve (AUC) of 100%. The chestnut dataset can be used in literature studies for different purposes and the proposed framework can be utilized as a computer-aided decision support system for experts in farming.

Keywords CBAM attention module · Transfer learning in agriculture · Convolutional neural network · Smart crop classification · Machine Learning

✉ Mustafa Yurdakul
mustafayurdakul@kku.edu.tr

Kübra Uyar
kubra.uyar@alanya.edu.tr

Şakir Taşdemir
stasdemir@selcuk.edu.tr

1 Computer Engineering Department, Kırıkkale University, Kırıkkale, Turkey

2 Computer Engineering Department, Alanya Alaaddin Keykubat University, Antalya, Turkey

3 Computer Engineering Department, Selçuk University, Konya, Turkey

Introduction

Chestnut is the seed of a tree from the beech family that grows in temperate climates. It is rich in carbohydrates, protein, fat, fiber, vitamin C, potassium and magnesium. It is consumed in cooking, frying, making flour or flavoring. Apart from food, it is used as a biofuel, as an antioxidant in cosmetics, and in the pharmaceutical industry for its anti-inflammatory and antimicrobial properties. Thanks to this versatility, it is in demand worldwide. In 2021, world chestnut production was approximately 2.2 million tons, 65.5% of which was produced by China, 7.22% by Spain, 3.13% by Bolivia and 2.99% by Turkey. The market value of chestnuts is directly related to quality. According to the United Nations Economic Commission for Europe (UNECE), quality is determined by criteria such as integrity, cleanliness and freedom from harmful organisms. The region where it grows also affects quality. This categorization is usually done by humans, but it is time-consuming, tiring and error-prone. Therefore, computer-aided systems are needed.

Computer vision methods have been widely used in the classification of agricultural products. High accuracies have been achieved for crops such as corn (Avuçlu et al. 2023), raisins (Çınar et al. 2020), pumpkin seeds (Koklu et al. 2021), hazelnuts (Gencturk et al. 2024), wheat (Yasar et al. 2024), chickpeas (Golcuk et al. 2023) and coffee (Aghdamifar et al. 2023).

There are various studies on chestnut classification. Öztekin et al. (2020) measured five chestnut varieties based on geometric features and obtained 99.99% accuracy with an artificial neural network (ANN). Donis-González et al. (2013) extracted color, texture and geometric features from images of chestnut slices at different decay levels and classified the data selected by sequential feature selection (SFS) with quadratic discriminant analysis (QDA) with 89.6% accuracy. Kurtulmuş et al. (2018) analyzed the sounds of falling chestnuts and obtained 88% accuracy with Support Vector Machine (SVM) based on humidity. Donis-González et al. (2014) classified features extracted from computed tomography (CT) images of chestnut slices with QDA and achieved 85.9, 91.2 and 96.1% accuracy.

Although chestnuts are a globally demanded food and classification is a difficult task, there are only few studies in the literature. The studies on chestnut classification in the literature are summarized in Table 1.

According to Table 1, various features were extracted from chestnuts and successful classifications were made with machine learning (ML) methods. However, feature extraction from images requires expertise and it becomes difficult to identify distinctive features as the number of classes increases. Therefore, the methods in the literature are not fully automatic. Deep learning (DL) methods such as CNN can automatically extract meaningful features and have shown successful results in food classification. In this context, it also has high potential in chestnut classification.

The main contributions of this study to the literature are as follows;

- Chestnut classification studies in the literature have been extensively reviewed.
- A new dataset was created with a total of 1156 chestnut images of four varieties in a real environment.
- The images were classified with 16 different advanced CNN models.
- The three most successful models were used as feature extractors and these features were passed to various ML algorithms.
- High performance and robust classification was achieved CNN with attention mechanism.
- A user-friendly mobile web application that classifies chestnuts with the most successful DL models was developed.

This paper is organized as follows: Sect. 2 presents a detailed description of the materials and methods. The experimental setup and experimental results with discussion are explained in Sects. 3 and 4, respectively. Finally, Sect. 5 summarizes the main conclusion of this work and gives details about future works.

Materials and Methods

The details of the image acquisition process, the knowledge about ML and DL methods, the description of the transfer learning used to improve the classification performance of CNN models, and the importance of feature extraction are discussed in this section. In this study, chestnut images are analyzed with pre-trained CNN models and the extracted features are transferred to ML algorithms. Classification performance is improved by transfer learning. Figure 1 shows the block diagram of the proposed method.

Table 1 A summary of chestnut classification studies in the literature using similar datasets

Reference	Methodology	Features	Classifier	Limitation	Highest accuracy (%)
Öztekin et al. (2020)	Physical measurements of chestnuts	Geometric dimensions	ANN	Low applicability (Extracting features using traditional image processing techniques)	99.99
Donis-González et al. (2013)	Image analysis of chestnut slices	Color, texture, geometry	QDA		89.60
Kurtulmuş et al. (2018)	Acoustic analysis based on moisture content	Acoustic signals	SVM		88.00
Donis-González et al. (2014)	X-ray CT image analysis of chestnut slices	Grayscale intensity, texture	QDA		96.10

ANN Artificial neural network, *QDA* Quadratic discriminant analysis, *SVM* Support Vector Machine

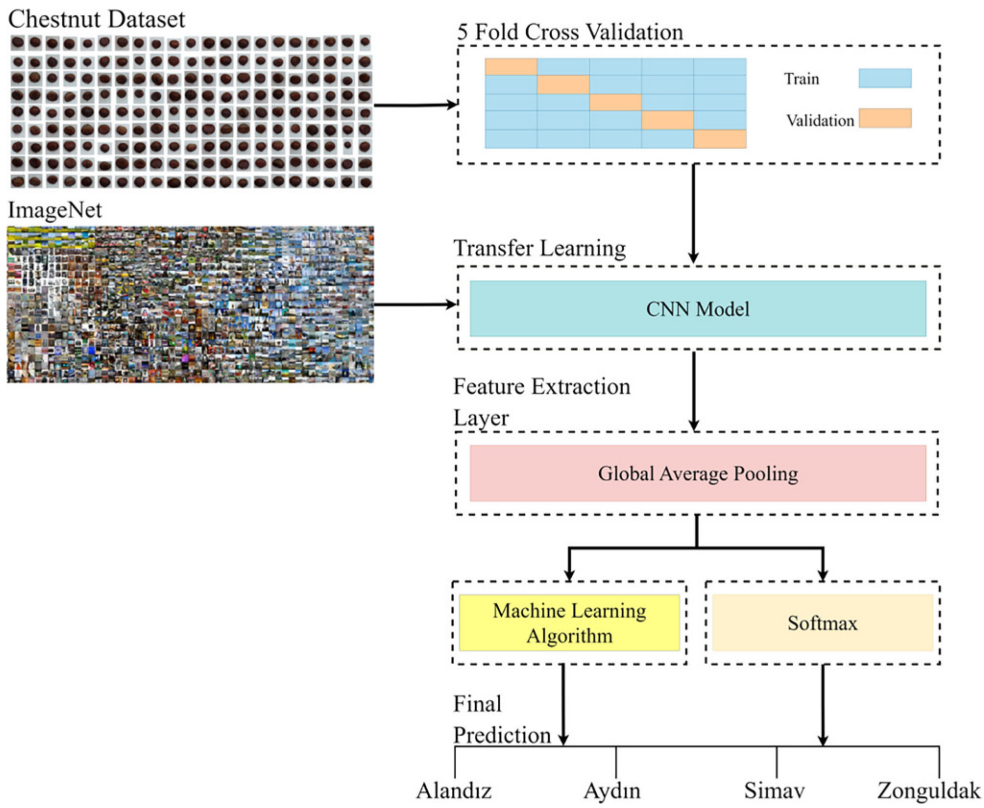

Fig. 1 Block diagram of the proposed deep approach. *CNN* Convolutional neural network

Table 2 The data distribution of the chestnut dataset

Chestnut variety	'Alandız'	'Aydın'	'Simav'	'Zonguldak'	Total
Number of images	272	228	304	352	1156

Data Collection Setup

Image Acquisition

A special box with homogeneous illumination was designed to capture the chestnut images. A Samsung NX300 camera with a 20.3 MP CMOS sensor was used in the box illuminated by 5 V powered LEDs. The lens was fixed at 45 mm, ISO 100 and 1/400 shutter speed. Each chestnut was recorded in JPG format from two angles, front and back. The data collection box is shown in Fig. 2.

Dataset

The researchers created a dataset by obtaining images of four different chestnut varieties ('Alandız', 'Aydın', 'Simav', and 'Zonguldak'), which are widely grown and marketed in Turkey. The raw images consist of four types with 1156 images. To enhance the generalization ability of the model, online data augmentation techniques is utilized. By increasing data diversity, they contribute to obtaining more robust and accurate predictions. The sample images of chestnut classes used in the study and the data distribution of the images belonging to these varieties are listed in Table 2. In addition, sample images of chestnut varieties from the dataset are depicted in Fig. 3. The dataset has been made publicly available at https://github.com/ymyurdakul/datasets/tree/main/chestnuts.

Fig. 2 Data collection setup

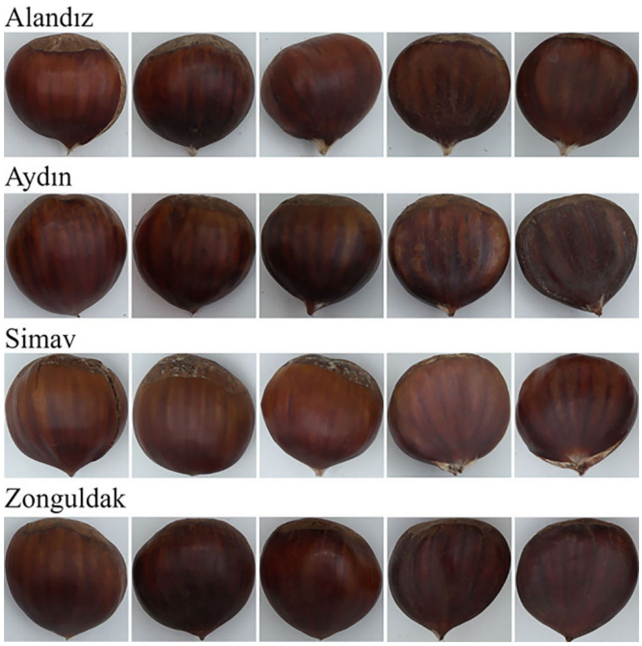

Fig. 3 Sample images from the chestnut dataset

Machine Learning Models

Successful implementation of feature extraction directly affects the classification result; therefore, features should summarize most of the information of the original dataset. For the classification task of the chestnuts, CNN features are used as input vectors of the different ML methods. Decision Tree (DT) classifies the data with a tree structure (Quinlan 2014). Naive Bayes (NB) performs probabilistic classification assuming independence of features (Quinlan 2014). SVM determines the optimal hyperplane to separate the data (Cortes and Vapnik 1995). AdaBoost improves the performance of the model by combining weak classifiers (Freund and Schapire 1997). Xtreme Gradient Boosting (XGB) is an improved gradient boosting method that offers high performance and speed (Chen and Guestrin 2016).

After obtaining extracted features depending on the Global Average Pooling (GAP) layer of CNN models, features are classified into different categories using DT, NB, SVM, Adaboost, and XBG classifiers.

Deep Learning Models

CNN is a successful DL algorithm used in visual analysis such as image classification, object detection and segmentation. It provides effective results in many fields such as medicine (Islam et al. 2020), agriculture (Sharma et al. 2020), defense (Yoo and Cho 2022), industry (Yurdakul et al. 2025) and education (Wang et al. 2023). It basically consists of convolution, pooling and fully connected layers. Convolution extracts distinctive features, pooling reduces the computational burden, and fully connected layer performs the classification.

Different CNN models have been developed with blocks such as dense, inception and residual. DenseNet (Huang et al. 2017) establishes full connectivity between layers. Inception (Szegedy et al. 2015) uses multiscale filters. VGG (Simonyan and Zisserman 2014) creates deep structures with small 3×3 filters. ResNet (He et al. 2016) uses residual connections to facilitate learning. Xception (Chollet 2017) provides high performance with fewer parameters with depth-discrete convolutions. MobileNet (Howard et al. 2017) uses depth-discrete convolutions that require low computational power. NASNet (Zoph et al. 2018) structurally optimizes the optimal network with evolutionary algorithms.

Transfer Learning

Transfer learning is a regularization technique that improves the classification performance of CNN models. The model first learns the basic features on a large dataset and then is retrained with new data. In this way, it learns dataset-specific features faster. The CNN models used in this study were pre-trained on ImageNet data with 1000 classes. To classify the four chestnut species, the final layers were modified and a dense layer with GAP and softmax was added.

Feature Extraction

CNN model training consists of two stages: feature extraction and classification. In the first stage, meaningful features are extracted from the images by convolution and pooling. In this study, classification is performed for four classes and after feature extraction, GAP layer and dense layer with softmax are added. GAP reduces the number of parameters by converting feature maps into vectors and aims to improve performance while reducing training time.

CNN Models with Attention Modules

The combination of CNNs with attention modules significantly improved DL model performance. Although CNNs capture local patterns well, they may be limited in identifying distant dependencies and regions of importance. Attention modules overcome this problem by dynamically assessing the importance of different regions, allowing the model to focus on the most informative areas.

The convolutional block attention module (CBAM), developed by Woo et al. (2018), is characterized by channel (CAM) and spatial (SAM) attention modules. CAM focuses on important features with inter-channel relationships, while SAM identifies the most informative regions in the image. The CBAM structure is shown in Fig. 4.

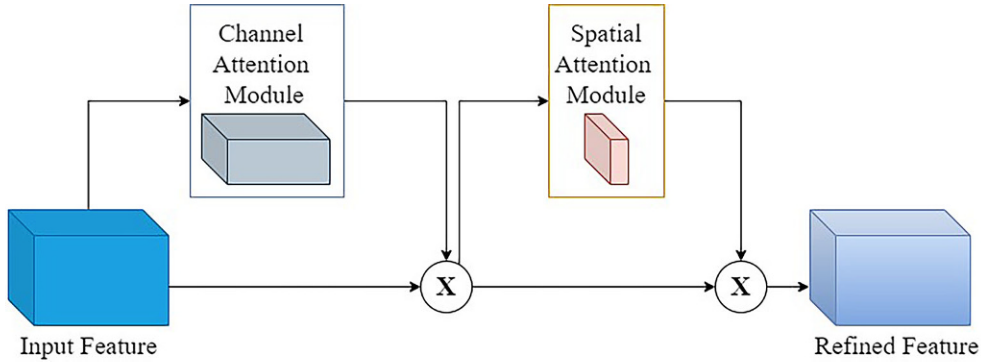

Fig. 4 Baseline structure for convolutional block attention module (CBAM) architecture

Table 3 The range of the hyperparameters

Hyperparameter	Range
Optimizer	Stochastic Gradient Descent (SGD), RMSprop, Adam, Adadelta, Adagrad, Adamax, Nadam
Minibatch size	16, 32, 64, 128
Initial learning rate	0.1, 0.01, 0.001

Experimental Setup

Training of the CNN Models

For the classification of chestnut varieties, pre-trained CNN models were modified and trained. In order to carry out the training step, some parameters need to be tuned. Therefore, the hyperparameters of the CNN models were determined using the grid search algorithm. This algorithm works by exhaustively searching through a predefined set of hyperparameters and their respective values to identify the combination that yields the best performance for a given model. The range of the hyperparameter values is listed in Table 3.

The best settings for the models' classification success were found by experimenting with various batch sizes, optimizers, and learning parameter values. At the end of the optimization of hyperparameters, a minibatch size of 32 was used with a learning rate of 0.01. In addition, Stochastic Gradient Descent (SGD) was used as the optimizer and the training was terminated at 50 epochs to prevent overfitting. Finally, the categorical cross-entropy loss was utilized in the classification layer of the CNN models.

Execution Environment

The experimental studies are conducted on a computer equipped with an Nvidia GTX 3060Ti GPU, 128 GB RAM, and an Intel (R) Core (TM) i9-10920X CPU. The Keras 2.10 API with Python 3.11 is used to develop CNN models and CUDA V12.2 is used for GPU acceleration.

Statistical Analysis

The evaluation of the performance of CNN and ML models was carried out by calculating the performance metrics which are accuracy, precision, recall, F1-score, and Cohen's kappa values. The mathematical formulas of measurement metrics are listed in Eqs. 1–5.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

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

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

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

$$\text{Cohen's Kappa} = \frac{P_{\text{agree}} - P_{\text{chance}}}{1 - P_{\text{chance}}} \quad (5)$$

TP refers to correct instances that the model correctly classifies "correctly"; FN refers to correct instances that the model incorrectly classifies "incorrectly"; TN refers to incorrect instances that the model correctly classifies "incorrectly"; FP refers to incorrect instances that the model incorrectly classifies "correctly". In Eq. 5, P_{agree} denotes the rate at which the judges agree and P_{chance} denotes the rate of random agreement. Model performance was evaluated with five-fold cross-validation. The data was divided into five parts and each part was used as test data in turn. This method is effective for evaluating overall performance on different subsets of data.

Table 4 Performance results of CNN models for the classification task of chestnut

CNN Models	Performance metrics (%)							
	Accuracy	Precision	Recall	F1-score	Cohen's Kappa	Area under the curve	Lowest accuracy	Highest accuracy
DenseNet121	**98.97 (±0.0075)**	**98.98 (±0.0073)**	**98.97 (±0.0075)**	**98.96 (±0.0075)**	**98.61 (±0.0101)**	**99.98 (±0.0002)**	**97.85**	**100**
DenseNet169	**98.88 (±0.0058)**	**98.90 (±0.0057)**	**98.88 (±0.0058)**	**98.88 (±0.0058)**	**98.49 (±0.0078)**	**99.98 (±0.0002)**	**98.27**	**99.57**
DenseNet201	98.80 (±0.0106)	98.82 (±0.0105)	98.80 (±0.0106)	98.79 (±0.0108)	98.38 (±0.0143)	99.98 (±0.0002)	97.00	100
InceptionV3	97.15 (±0.0109)	97.21 (±0.0110)	97.15 (±0.0109)	97.15 (±0.0110)	96.17 (±0.0147)	99.89 (±0.0008)	95.28	98.27
MobileNet	**99.05 (±0.0050)**	**99.07 (±0.0049)**	**99.05 (±0.0050)**	**99.05 (±0.0050)**	**98.72 (±0.0068)**	**99.99 (±0.0001)**	**98.69**	**100**
MobileNetV2	82.15 (±0.0592)	78.41 (±0.1218)	82.15 (±0.0592)	77.58 (±0.0870)	75.62 (±0.0819)	98.33 (±0.0109)	76.52	93.56
NASNetMobile	63.39 (±0.1382)	81.03 (±0.0300)	63.39 (±0.1382)	61.44 (±0.1510)	51.61 (±0.1819)	93.47 (±0.0346)	39.13	77.29
ResNet101	98.18 (±0.0108)	98.25 (±0.0103)	98.18 (±0.0108)	98.17 (±0.0110)	97.56 (±0.0145)	99.97 (±0.0003)	96.52	99.56
ResNet101V2	97.24 (±0.0103)	97.33 (±0.0096)	97.24 (±0.0103)	97.24 (±0.0103)	96.29 (±0.0138)	99.85 (±0.0010)	95.71	98.69
ResNet152	97.58 (±0.0074)	97.68 (±0.0066)	97.58 (±0.0074)	97.57 (±0.0078)	96.75 (±0.0100)	99.95 (±0.0003)	96.14	98.26
ResNet152V2	97.67 (±0.0100)	97.70 (±0.0099)	97.67 (±0.0100)	97.67 (±0.0099)	96.86 (±0.0134)	99.93 (±0.0004)	95.71	98.28
ResNet50	97.50 (±0.0128)	97.58 (±0.0125)	97.50 (±0.0128)	97.47 (±0.0130)	96.63 (±0.0173)	99.94 (±0.0006)	95.71	99.13
ResNet50V2	97.32 (±0.0125)	97.39 (±0.0119)	97.32 (±0.0125)	97.32 (±0.0124)	96.41 (±0.0167)	99.83 (±0.0013)	95.28	98.70
VGG16	96.72 (±0.0129)	96.75 (±0.0128)	96.72 (±0.0129)	96.71 (±0.0129)	95.59 (±0.0173)	99.80 (±0.0013)	95.67	99.13
VGG19	96.72 (±0.0142)	96.75 (±0.0143)	96.72 (±0.0142)	96.72 (±0.0143)	95.59 (±0.0191)	99.81 (±0.0012)	95.28	99.13
Xception	97.58 (±0.0126)	97.66 (±0.0119)	97.58 (±0.0126)	97.58 (±0.0126)	96.75 (±0.0169)	99.91 (±0.0007)	96.14	99.13

Experimental Results and Discussion

The effectiveness of the proposed framework was assessed through the running of numerous experiments using DL and ML approaches. In the experimental studies, 16 CNN models were modified and re-trained to classify multi-class chestnut varieties as illustrated in Fig. 3 and given details in Table 2. Moreover, a webserver-based mobile application was developed to analyze the chestnut images in real-time and instantly. The best performing results in all tables are highlighted in bold.

Fig. 5 The confusion matrices of the CNN models (*0* ‘Alandız’, *1* ‘Aydın’, *2* ‘Simav’, *3* ‘Zonguldak’)

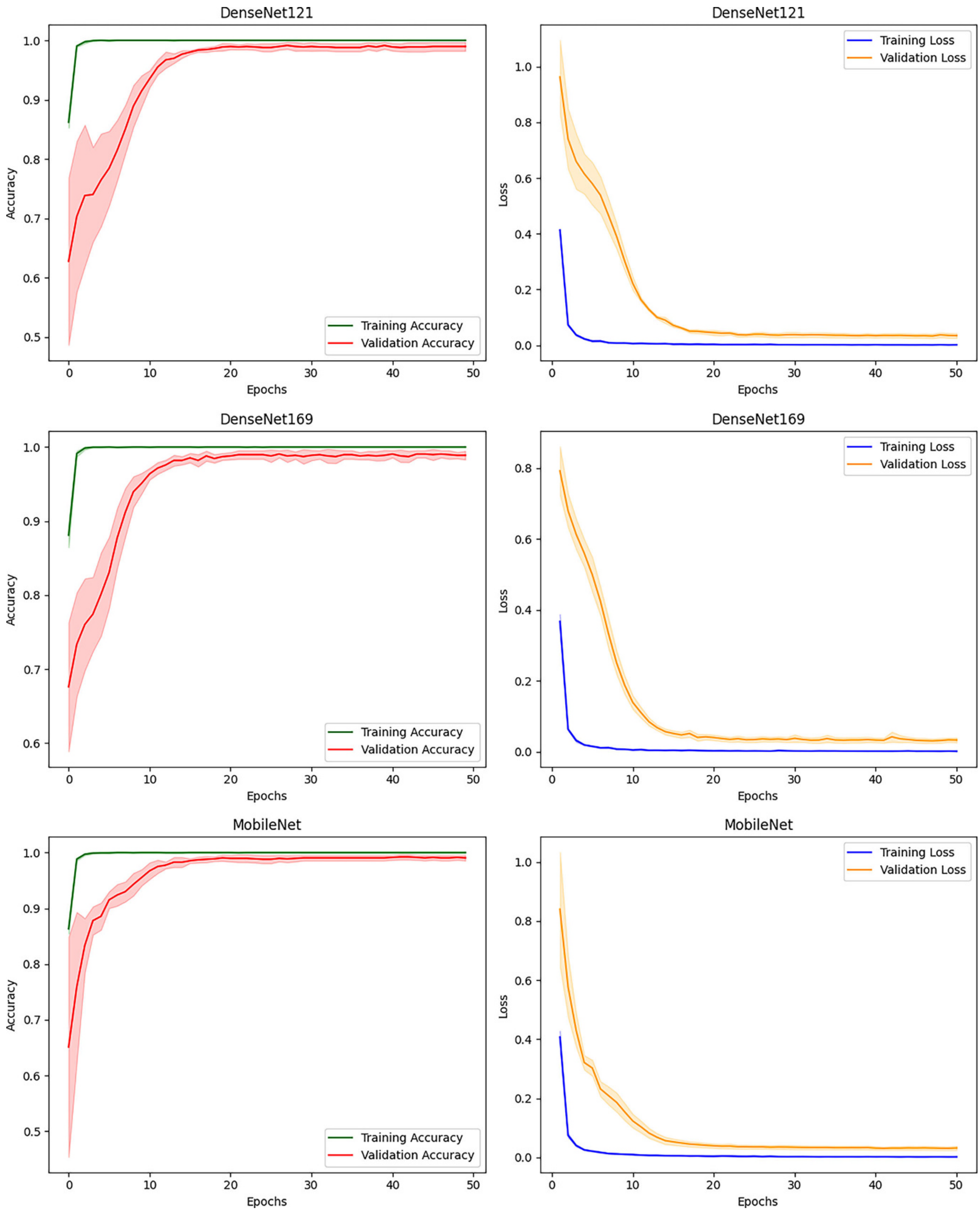

Fig. 6 Minibatch accuracy and loss values of the best three CNN models during training

Results of CNN Models

The classification performances of pre-trained CNN models that contain different topologies were tested to observe and find the deep models that have better feature extraction ability. The classification performances of the pre-trained CNN models are given in Table 4. In Table 4, the bold type numbers indicate the performance metrics of the best-performing CNN models.

Table 4 shows that the pre-trained CNN models achieved high accuracy rates between approximately 63 and 99%. In addition, the standard deviation values show that the results obtained are consistent. The highest result was achieved with MobileNet with an accuracy of 99.05%, precision of 99.07%, recall of 99.05%, F1-score of 99.05, kappa score of 98.72%, and area under the curve (AUC) of 99.99%. MobileNet achieved a high classification performance by using the advantage of depth-wise separable convolutions in its structure.

MobileNet is the most successful model, while MobileNetV2 achieved only an accuracy of 82.15%. MobileNetV2 includes inverted residual blocks in addition to the structure of the MobileNet model. The reason why MobileNetV2 has a lower accuracy rate is that the inverted residual blocks in the structure of the model increase the complexity of the model and negatively affect the learning process. The lowest accuracy rate was obtained with the NASNetMobile model with 63.39%. NASNetMobile learns noise or details in the data due to its complex structure and therefore cannot generalize to the test data. The confusion matrices that visually represent the classification performances of the models are illustrated in Fig. 5.

When Fig. 5 is analyzed, it is seen that the models confused 'Alandız', 'Aydın', and 'Simav' varieties. The main reason for this confusion is that the textural characteristics of 'Alandız', 'Aydın', and 'Simav' chestnut varieties are similar to each other. In addition, the color tones of these types are also very close to each other. These similarities caused the models to make mistakes in distinguishing between these three varieties.

Figure 6 shows the change in accuracy and loss values of the models on training and validation data belonging to the best three CNN models. It is observed that the generalization capability of the models improved after 20 epochs and the value ranges on both training and test data narrowed. In addition, it was determined that there was no significant improvement in the models after 45 epochs and the training was completed in 50 epochs.

Results of ML Algorithms

The three most successful models, DensNet121, DenseNet169, and MobileNet models, were used as feature extractors and 1024, 1664, and 1024 features were used, respectively. As mentioned previous section, GAP layer provided fewer learnable parameters and the extracted features were classified with ML algorithms. The classification performances of ML methods using the extracted features based on DenseNet121 is listed in Table 5 and confusion matrices are presented in Fig. 7. The best performing result in Table 5 is highlighted in bold.

The features extracted with DenseNet121 and ML algorithms achieved accuracy rates between 90 and 99%. The most successful classifier was SVM using the radial basis function (RBF) kernel with an accuracy of 99.40%, pre-

Table 5 The classification performances of ML models using DenseNet121-based extracted features

Model	Performance metrics (%)							
	Accuracy	Precision	Recall	F1-score	Cohen's Kappa	Area under the curve	Lowest accuracy	Highest accuracy
DT	90.32 (±0.0146)	90.44 (±0.0155)	90.32 (±0.0146)	90.31 (±0.0148)	86.97 (±0.0197)	94.25 (±0.0062)	88.41	92.58
NB	96.89 (±0.0126)	97.11 (±0.0107)	96.89 (±0.0126)	96.93 (±0.0121)	95.83 (±0.0168)	99.55 (±0.0028)	95.28	98.28
SVM (Linear)	99.40 (±0.0035)	99.40 (±0.0034)	99.40 (±0.0035)	99.39 (±0.0035)	99.19 (±0.0047)	**99.98 (±0.0002)**	99.13	100
SVM (RBF)	**99.40 (±0.0044)**	**99.40 (±0.0043)**	**99.40 (±0.0044)**	**99.40 (±0.0044)**	**99.19 (±0.0059)**	**99.96 (±0.0003)**	**98.71**	**100**
SVM (Poly)	96.98 (±0.0257)	97.50 (±0.0189)	96.98 (±0.0257)	97.08 (±0.0243)	95.96 (±0.0343)	99.96 (±0.0004)	92.70	100
SVM (Sigmoid)	99.22 (±0.0050)	99.24 (±0.0050)	99.22 (±0.0050)	99.22 (±0.0051)	98.96 (±0.0068)	99.98 (±0.0002)	98.70	100
Adaboost	92.41 (±0.0372)	93.00 (±0.0333)	92.41 (±0.0372)	92.50 (±0.0365)	89.82 (±0.0497)	97.38 (±0.0316)	86.27	96.52
XGB	97.93 (±0.0099)	97.97 (±0.0095)	97.93 (±0.0099)	97.93 (±0.0099)	97.22 (±0.0133)	99.91 (±0.0007)	96.57	99.13

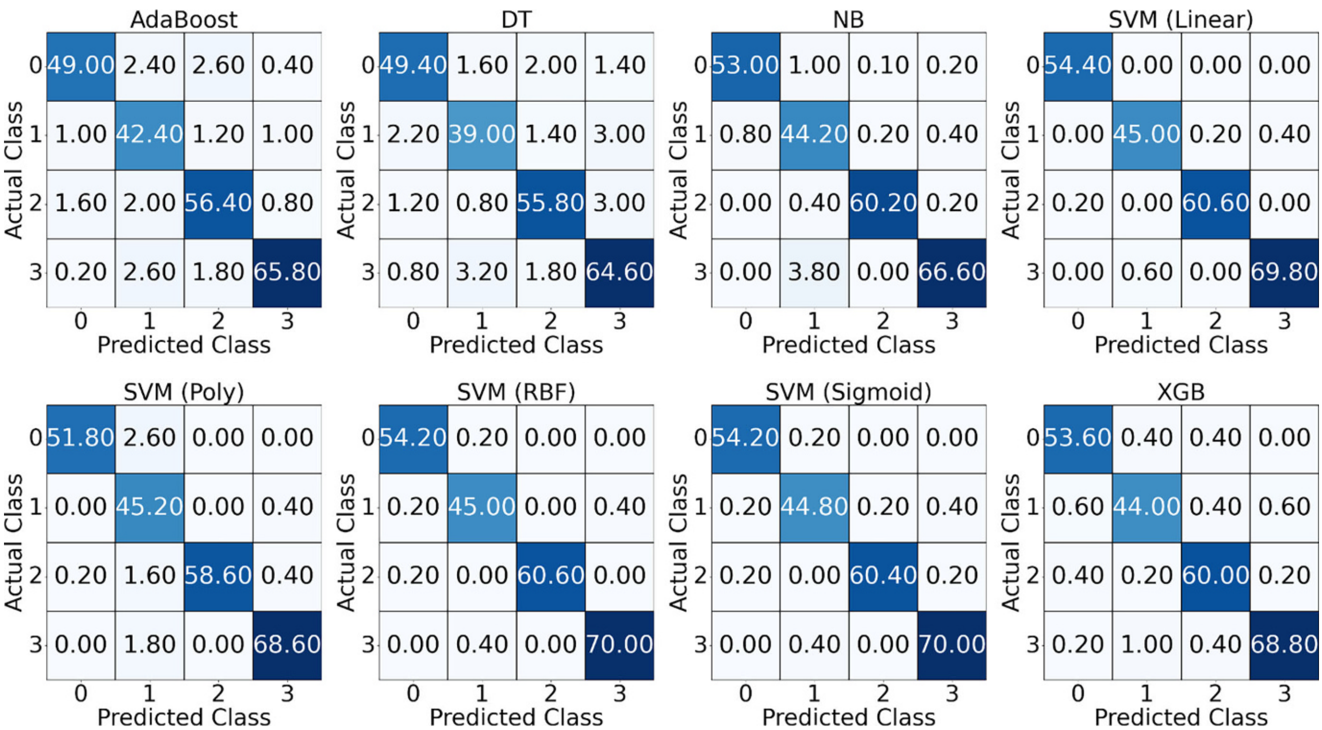

Fig. 7 Confusion matrices of ML algorithms using DenseNet121 features

cision of 99.40%, recall of 99.40%, F1-score of 99.40%, kappa score of 99.19%, and AUC of 99.96%. With the DenseNet121 and softmax classifier, the accuracy rate achieved as 98.97% was increased to 99.40%. DT had the lowest accuracy rate with a 90.32% accuracy rate. When the standard deviation values of the performance results are examined, it is seen that the values are quite minimal, which shows that the results are consistent.

The features extracted with DenseNet169 and ML algorithms achieved accuracy rates between 87 and 99%. The most successful classifier was SVM using Linear kernel with an accuracy rate of 99.05%, precision of 99.07%, recall of 99.05%, F1-score of 99.05%, kappa score of 98.73% and AUC of 99.97%. The accuracy rate achieved was 98.88% with DenseNet169 and the softmax classifier increased to 99.05%. Adaboost had the lowest accuracy rate with 87.45% accuracy rate. The confusion matrices are also depicted in Fig. 8.

The classification performances ML methods using the features extracted with MobileNet and ML algorithms achieved accuracy rates between 85 and 98%. The most successful classifier was SVM using Linear kernel with an accuracy of 98.36%, precision of 98.38%, recall of 98.36%, F1-score of 98.36%, kappa score of 97.80%, and AUC of 99.96%. The accuracy rate of 99.05% with MobileNet and softmax classifiers was achieved as 99.05% and the maximum accuracy value of 98.36% was obtained with ML methods. DT had the lowest accuracy rate with 85.04% accuracy rate. The confusion matrices are also illustrated in Fig. 9.

Experimental studies were completed by analyzing the feature extraction capabilities of CNN models with different topological structures and the classification performances of different ML classifiers. DenseNet121 is an effective feature extractor in classifying chestnut species and the SVM classifier produced a more successful result by outperforming other ML classifiers.

Considering all the confusion matrices shown in Figs. 7, 8 and 9, for the classification task of chestnut species using mentioned deep approaches, we can conclude that while the most accurately classified category among chestnut varieties is 'Zonguldak', 'Aydın' varieties are a little more difficult to distinguish than the others.

Results of CNN Models with Attention Module

The use of CBAM helps the model prioritize important spatial and channel information, leading to more accurate predictions without significantly increasing computational complexity. Therefore, integrating into the attention mechanisms to CNN architectures effectively enhances the ability of models to focus on the most relevant features in an image.

The best-performing three CNN models and their attention mechanism-enhanced variants stand out for improving performance without increasing model complexity. Com-

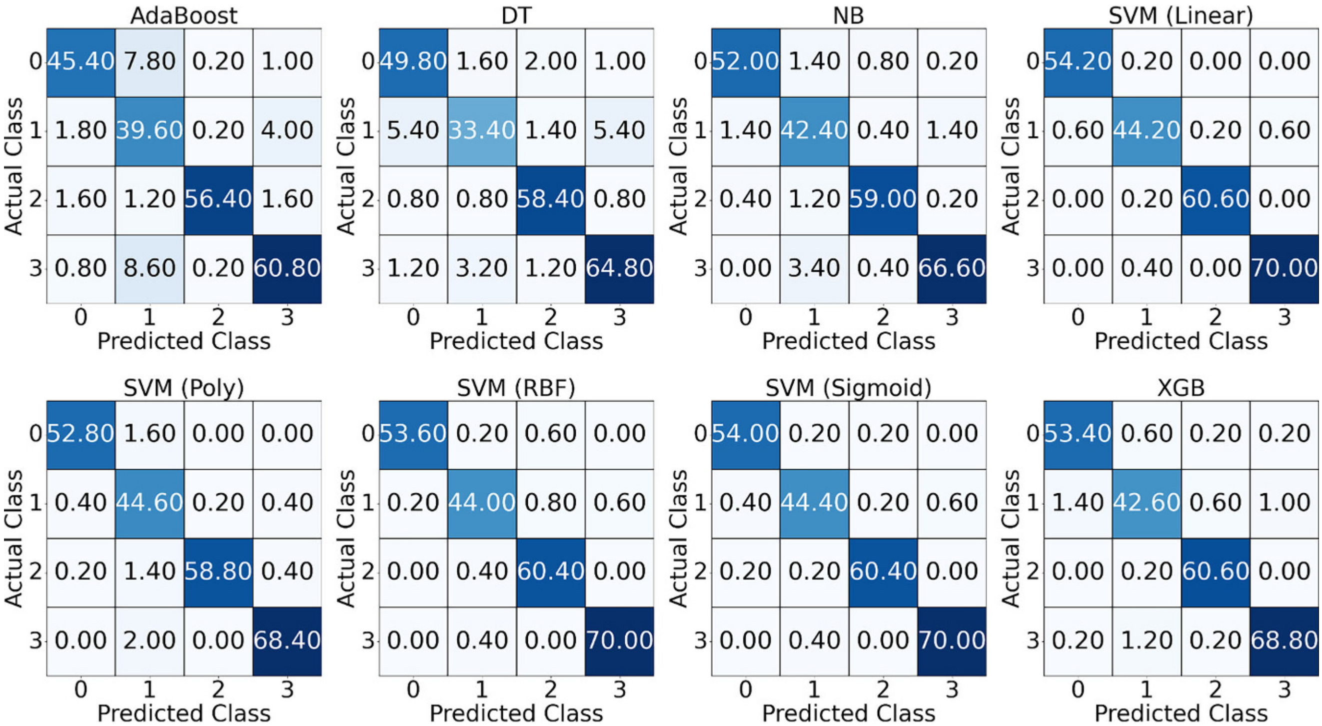

Fig. 8 Confusion matrices of ML algorithms using DenseNet169 features

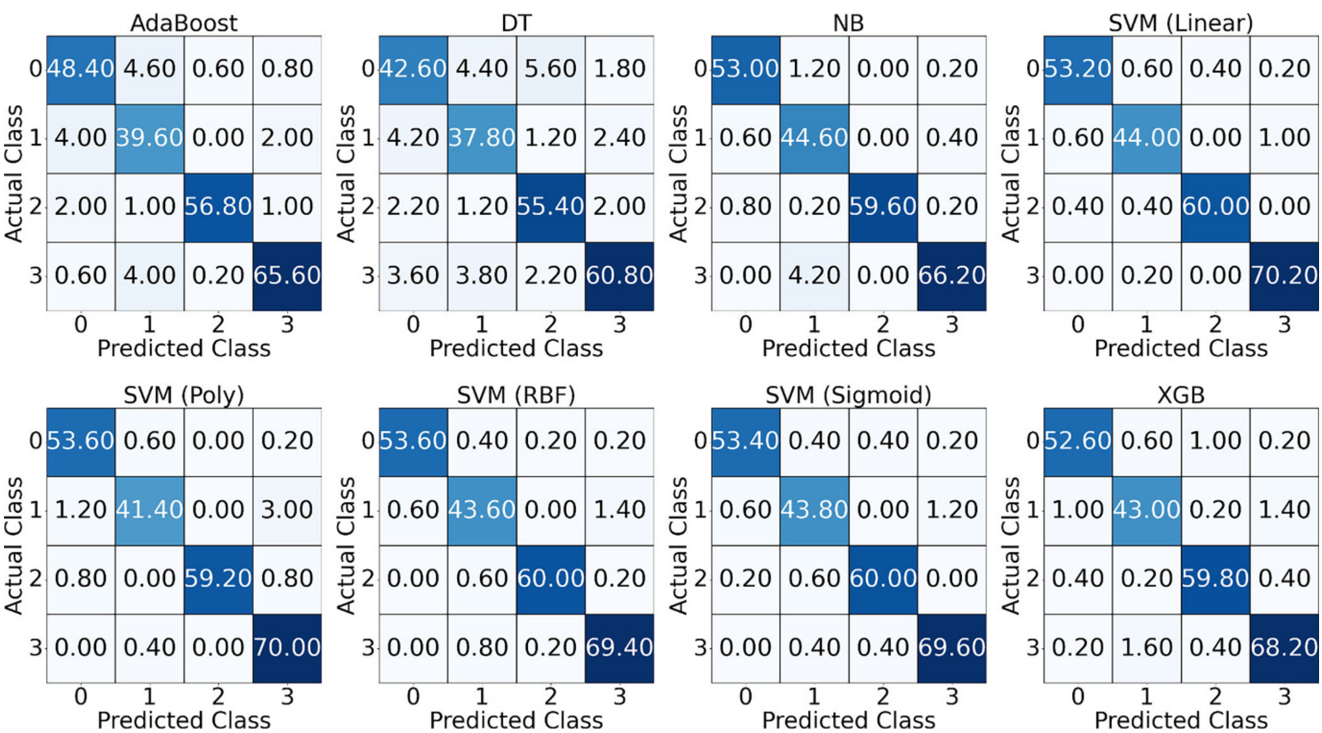

Fig. 9 Confusion matrices of ML algorithms using MobileNet features

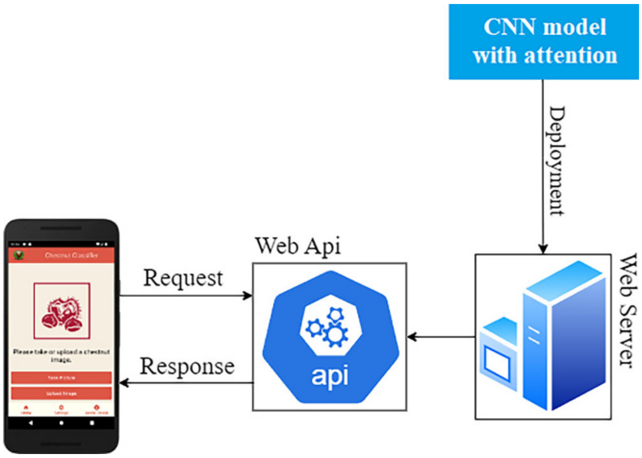

Fig. 10 The flowchart of the chestnut detection using computer vision techniques with webserver-based mobile application. *CNN* Convolutional neural network

pared to the classification result using the top three CNN models with SVM classifier (MobileNet, DenseNet121, and DenseNet169), a significant increase in success was achieved by adding the CBAM attention mechanism to the mentioned three CNN models.

Table 8 shows that the highest result achieved by MobileNet with attention mechanism (MobileNet-CBAM) is accuracy of 99.65%, precision of 99.62%, recall of 99.67%, F1-score of 99.64%, kappa score of 100%, and AUC of 100%. The most important feature of MobileNet for web applications is its lightweight architecture, which enables efficient performance on devices with limited computational resources, such as smartphones and tablets. MobileNet achieves this by using depthwise separable convolutions, significantly reducing the number of parameters and computational cost.

Webserver-Based Mobile Application for Chestnut Classification

A webserver-based mobile application was developed for real-time chestnut varieties classification. With this user-friendly application, instant image analysis can be performed in a short time. The MobileNet-CBAM model used in the mobile application is hosted on the webserver to keep it up-to-date and accessed through the web service provided with ASP.NET Core. The application developed with Kotlin is compatible with different platforms. Figure 10 shows the flowchart of this system.

Mobile applications can serve different purposes and the application developed for this study facilitates users to identify chestnut species. Users can take a photo of the chestnut at any time and learn its class. This artificial intelligence based system works with DL and ML algorithms. According to the experimental results, the most successful deep model was integrated into the application. Figure 11 shows the mobile application interface.

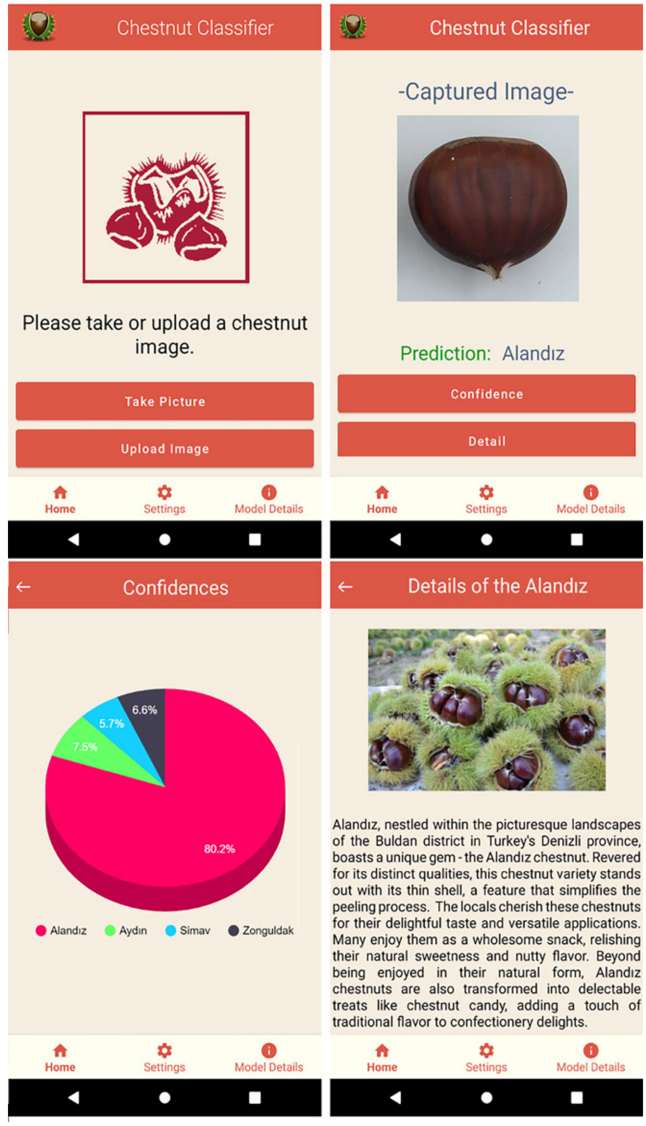

Fig. 11 The interface of the mobile application

As shown in Fig. 11, firstly the main page of the application is opened. For starting the chestnut category prediction, it displays the prediction ways, either the user can take a photo of the chestnut and see the prediction result or press the “Upload Image” button to upload an image from the mobile gallery. After uploading the image, the result of the category of chestnut is shown to the user as a prediction. As a result of prediction of deep model, confidence values of the chestnut types are illustrated with pie chart. Moreover, there is an option to see the details of the predicted chestnut class.

Table 6 The classification performances of ML models using DenseNet169-based extracted features

Model	Performance metrics (%)							
	Accuracy	Precision	Recall	F1-score	Cohen's Kappa	Area under the curve	Lowest accuracy	Highest accuracy
DT	89.28 (±0.0116)	89.45 (±0.0113)	89.28 (±0.0116)	89.11 (±0.0126)	85.55 (±0.0155)	94.61 (±0.0080)	87.88	90.83
NB	95.16 (±0.0137)	95.37 (±0.0127)	95.16 (±0.0137)	95.20 (±0.0133)	93.50 (±0.0183)	97.98 (±0.0084)	93.51	96.57
SVM (Linear)	**99.05 (±0.0069)**	**99.07 (±0.0067)**	**99.05 (±0.0069)**	**99.05 (±0.0068)**	**98.73 (±0.0092)**	**99.97 (±0.0004)**	**98.28**	**100**
SVM (RBF)	98.62 (±0.0110)	98.66 (±0.0104)	98.62 (±0.0110)	98.62 (±0.0109)	98.15 (±0.0147)	99.96 (±0.0004)	96.57	99.57
SVM (Poly)	97.15 (±0.0125)	97.39 (±0.0100)	97.15 (±0.0125)	97.20 (±0.0119)	96.18 (±0.0168)	99.86 (±0.0022)	94.85	98.27
SVM (Sigmoid)	98.97 (±0.0064)	98.98 (±0.0063)	9897 (±0.0064)	98.97 (±0.0064)	98.61 (±0.0086)	99.98 (±0.0002)	97.85	99.57
Adaboost	87.45 (±0.0232)	89.05 (±0.0224)	87.45 (±0.0232)	87.76 (±0.0225)	83.20 (±0.0310)	93.74 (±0.0331)	83.55	90.13
XGB	97.50 (±0.0077)	97.51 (±0.0077)	97.50 (±0.0077)	97.49 (±0.0077)	96.63 (±0.0104)	99.87 (±0.0009)	96.57	98.26

Discussion

The analysis of chestnuts is important to accurately determine their market value and increase agricultural productivity. Therefore, there is a need to create an original dataset. Conventional approaches to chestnut variety detection are based on visual analyses by laborers that are tedious, imprecise, subjective, and prone to error. Therefore computer-aided systems are needed to obtain more effective automatic systems. In the study, a novel dataset that contains 1156 images belonging to chestnuts varieties and made publicly available (Figs. 2 and 3). In the first experiment, the chestnut images are classified with 16 pre-trained CNN models. As a result of the first experiment, MobileNet, DenseNet121, and DenseNet169 models are the three best models achieving accuracies of 99.05%, 98.97%, and 98.88%, respectively (Table 4). In the second experiment, the best three models are used as feature extractors and the extracted features are classified with various ML algorithms. As a result of the second experiment, DenseNet121 and SVM (RBF kernel) achieved an accuracy of 99.40% (Table 5), DenseNet169 and SVM (Linear) achieved 99.05% (Table 6), MobileNet and SVM (Linear) achieved 98.36% (Table 7). Classification results of the SVM algorithm using features of DenseNet121 and DenseNet169 deep models obtained better results

Table 7 The classification performances of ML models using MobileNet-based extracted features

Model	Performance metrics (%)							
	Accuracy	Precision	Recall	F1-score	Cohen's Kappa	Area under the curve	Lowest accuracy	Highest accuracy
DT	85.04 (±0.0113)	85.61 (±0.0129)	85.04 (±0.0113)	84.99 (±0.0121)	79.91 (±0.0154)	91.55 (±0.0106)	84.12	87.01
NB	96.63 (±0.0136)	96.95 (±0.0105)	96.63 (±0.0136)	96.67 (±0.0131)	95.48 (±0.0182)	99.46 (±0.0031)	94.76	98.70
SVM (Linear)	**98.36 (±0.0116)**	**98.38 (±0.0116)**	**98.36 (±0.0116)**	**98.36 (±0.0116)**	**97.80 (±0.0156)**	**99.96 (±0.0004)**	**96.14**	**99.13**
SVM (RBF)	98.01 (±0.0130)	98.05 (±0.0127)	98.01 (±0.0130)	98.01 (±0.0129)	97.33 (±0.0174)	99.95 (±0.0004)	96.14	99.57
SVM (Poly)	96.98 (±0.0075)	97.13 (±0.0073)	96.98 (±0.0075)	96.96 (±0.0076)	95.93 (±0.0102)	99.95 (±0.0005)	95.71	97.83
SVM (Sigmoid)	98.10 (±0.0135)	98.13 (±0.0133)	98.10 (±0.0135)	98.10 (±0.0134)	97.45 (±0.0181)	99.95 (±0.0004)	95.71	99.57
Adaboost	91.00 (±0.0163)	91.60 (±0.0155)	91.00 (±0.0163)	91.14 (±0.0161)	87.92 (±0.0221)	97.55 (±0.0093)	88.26	92.70
XGB	96.72 (±0.0096)	96.76 (±0.0097)	96.72 (±0.0096)	96.72 (±0.0097)	95.58 (±0.0130)	99.76 (±0.0013)	95.65	97.84

compared to the softmax classifier. However, when used as a feature extractor, MobileNet performed better with softmax than ML algorithms. It can be seen that using ML algorithms instead of the softmax classifier generally improves the classification performance of deep models. Moreover, the use of CBAM helps the MobileNet (MobileNet-CBAM) prioritize important spatial and channel information, leading to more accurate predictions (Table 8) without significantly increasing computational complexity.

Table 8 The classification performances of best-performing three CNN models with attention mechanisms

Model	Performance metrics (%)							
	Accuracy	Precision	Recall	F1-score	Cohen's Kappa	Area under the curve	Lowest accuracy	Highest accuracy
MobileNet-CBAM	**99.65 (±0.0173)**	**99.62 (±0.0101)**	**99.67 (±0.013)**	**99.64 (±0.013)**	**99.9 (±0.1)**	**99.9 (±0.01)**	**98.45**	**100**
DenseNet121-CBAM	99.26 (±0.089)	99.15 (±0.078)	99.25 (±0.069)	99.20 (±0.059)	99.9 (±0.01)	99.9 (±0.01)	97.95	100
DenseNet169-CBAM	99.22 (±0.046)	99.11 (±0.56)	99.21 (±0.065)	99.16 (±0.075)	99.9 (±0.01)	99.9 (±0.01)	97.20	100

Strengths and Weaknesses

The study provides highly accurate classification with a unique and balanced dataset consisting of four different chestnut varieties, and can be integrated into mobile applications thanks to lightweight models supported by attention mechanisms. However, the images were acquired in a controlled environment. In future studies, it is aimed to detect chestnuts in a natural environment and the model is aimed to show high performance in real field conditions.

Conclusion

Chestnuts, with their high nutritional value, are widely used in areas such as medicine, energy and cosmetics. Therefore, the correct classification of its varieties is important for its market value. Traditional methods rely on the human eye and are time-consuming and error-prone. In this study, an open dataset of 1156 images of four different chestnut varieties was created. Two different approaches were used for classification: In the first one, 16 convolutional neural network (CNN) models were tested using transfer learning, with the highest success rate of 99.05% in MobileNet. In the second approach, the three most successful CNN models were used as feature extractors and classified with machine learning algorithms; the best result was obtained in the combination of DenseNet121 + SVM with 99.40%. The integration of attention mechanisms (e.g. CBAM) further improved the classification performance, with MobileNet-CBAM achieving 99.65%. In future work, the proposed approach is aimed to work in real-time on embedded systems.

Author Contribution Mustafa Yurdakul: Conceptualization, Data collection, Methodology, Review and editing, Software, Validation, Visualization, Writing-original draft; Kübra Uyar: Conceptualization, Methodology, Review and editing, Software, Validation, Visualization, Writing-original draft; Şakir Taşdemir: Conceptualization, Methodology, Supervision.

Data Availability The dataset used in the study can be accessed from the link https://github.com/ymyurdakul/datasets/tree/main/chestnuts

Declarations

Conflict of interest M. Yurdakul, K. Uyar and Ş. Taşdemir declare that they have no competing interests.

Ethical standards This study does not contain any studies with human participants or animals performed by any of the authors.

References

Aghdamifar E, Rasooli Sharabiani V, Taghinezhad E, Rezvanivand Fanaei A, Szymanek M (2023) Non-destructive method for identification and classification of varieties and quality of coffee beans based on soft computing models using VIS/NIR spectroscopy. Eur Food Res Technol 249(6):1599–1612. https://doi.org/10.1007/s00217-023-04240-x

Avuçlu E, Taşdemir Ş, Köklü M (2023) A new hybrid model for classification of corn using morphological properties. Eur Food Res Technol 249(3):835–847. https://doi.org/10.1007/s00217-022-04181-x

Chen T, Guestrin C (2016) Xgboost: a scalable tree boosting system. In: Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, pp 785–794 https://doi.org/10.1145/2939672.2939785

Chollet F (2017) Xception: deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1251–1258 https://doi.org/10.48550/arXiv.1610.02357

Çınar İ, Koklu M, Taşdemir Ş (2020) Classification of raisin grains using machine vision and artificial intelligence methods. Gazi J Eng Sci 6(3):200–209. https://doi.org/10.30855/gmbd.2020.03.03

Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20:273–297

Donis-González IR, Guyer DE, Leiva-Valenzuela GA, Burns J (2013) Assessment of chestnut (Castanea spp.) slice quality using color images. J Food Eng 115(3):407–414. https://doi.org/10.1016/j.jfoodeng.2012.09.017

Donis-González IR, Guyer DE, Fulbright DW, Pease A (2014) Postharvest noninvasive assessment of fresh chestnut (Castanea spp.) internal decay using computer tomography images. Postharvest Biol Technol 94:14–25. https://doi.org/10.1016/j.postharvbio.2014.02.016

Freund Y, Schapire RE (1997) A decision-theoretic generalization of on-line learning and an application to boosting. J Comput Syst Sci 55(1):119–139. https://doi.org/10.1006/jcss.1997.1504

Gencturk B, Arsoy S, Taspinar YS, Cinar I, Kursun R, Yasin ET, Koklu M (2024) Detection of hazelnut varieties and development of mobile application with CNN data fusion feature reduction-based models. Eur Food Res Technol 250(1):97–110. https://doi.org/10.1007/s00217-023-04369-9

Golcuk A, Yasar A, Saritas MM, Erharman A (2023) Classification of Cicer arietinum varieties using MobileNetV2 and LSTM. Eur Food Res Technol 249(5):1343–1350. https://doi.org/10.1007/s00217-023-04217-w

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 770–778 https://doi.org/10.48550/arXiv.1512.03385

Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, Andreetto M, Adam H (2017) Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861. https://doi.org/10.48550/arXiv.1704.04861

Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4700–4708 https://doi.org/10.1109/CVPR.2017.243

Islam MZ, Islam MM, Asraf A (2020) A combined deep CNN-LSTM network for the detection of novel coronavirus (COVID-19) using X-ray images. Informatics Med Unlocked 20:100412. https://doi.org/10.1016/j.imu.2020.100412

Koklu M, Sarigil S, Ozbek O (2021) The use of machine learning methods in classification of pumpkin seeds (Cucurbita pepo L.). Genet Resour Crop Evol 68(7):2713–2726. https://doi.org/10.1007/s10722-021-01226-0

Kurtulmuş F, Öztüfekçi S, Kavdır İ (2018) Classification of chestnuts according to moisture levels using impact sound analysis and machine learning. J Food Meas Charact 12(4):2819–2834. https://doi.org/10.1007/s11694-018-9897-y

Öztekin YB, Alper T, Duran H (2020) Chestnut (Castanea sativa Mill.) cultivar classification: an artificial neural network approach. Notulae Bot Horti Agrobot Cluj Napoca 48(1):366–377. https://doi.org/10.15835/nbha48111752

Quinlan JR (2014) C4. 5: programs for machine learning. Elsevier

Sharma R, Das S, Gourisaria MK, Rautaray SS, Pandey M (2020) A model for prediction of paddy crop disease using CNN. In: Progress in computing, Analytics and networking: proceedings of ICCAN 2019. Springer, pp 533–543

Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556. https://doi.org/10.48550/arXiv.1409.1556

Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1–9 https://doi.org/10.48550/arXiv.1409.4842

Wang J, Satapathy SC, Wang S, Zhang Y (2023) LCCNN: a lightweight customized CNN-based distance education app for COVID-19 recognition. Mob Netw Appl. https://doi.org/10.1007/s11036-023-02185-9

Woo S, Park J, Lee J-Y, Kweon IS (2018) CBAM: Convolutional Block Attention Module. Computer Vision—ECCV 2018, Cham, pp 3–19 https://doi.org/10.1007/978-3-030-01234-2_1

Yasar A, Golcuk A, Sari OF (2024) Classification of bread wheat varieties with a combination of deep learning approach. Eur Food Res Technol 250(1):181–189. https://doi.org/10.1007/s00217-023-04375-x

Yoo J, Cho Y (2022) ICSA: Intelligent chatbot security assistant using Text-CNN and multi-phase real-time defense against SNS phishing attacks. Expert Syst Appl 207:117893. https://doi.org/10.1016/j.eswa.2022.117893

Yurdakul M, Uyar K, Taşdemir Ş (2025) Enhanced ore classification through optimized CNN ensembles and feature fusion. Iran J Comput Sci: 1–19

Zoph B, Vasudevan V, Shlens J, Le QV (2018) Learning transferable architectures for scalable image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 8697–8710 https://doi.org/10.48550/arXiv.1707.07012

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.