Türkiye Teknoloji ve Uygulamalı Bilimler Dergisi

ISSN: 2528-8652 e-ISSN: 2822-2660 2024, 5(2):151-171
Period Biannually Founded 2017 Publisher Siirt University https://dergipark.org.tr/tr/pub/tubid

Deep Learning for Automatic Classification of Fruits and Vegetables: Evaluation from the Perspectives of Efficiency and Accuracy

Demet PARLAK SÖNMEZ
Kayseri University, Department of Computational Sciences, Kayseri, Türkiye
http://orcid.org/0000-0001-5705-2467

Şafak KILIÇ*
Kayseri University, Faculty of Engineering Architecture and Design, Department of Software Engineering, Kayseri, Türkiye
http://orcid.org/0000-0002-2014-7638

Research Article

Received

22/07/2024

Accepted

25/07/2024

DOI

10.70562/tubid.1520357

Abstract

Within the agricultural domain, accurately categorizing the freshness levels of fruits and vegetables holds immense significance, as this classification enables early detection of spoilage and allows for appropriate grouping of products based on their intended export destinations. These processes necessitate a system capable of meticulously classifying fruits and vegetables while minimizing labor expenditures. The current study concentrates on developing an advanced model that can effectively categorize the freshness status of each fruit and vegetable as 'good,' 'medium,' or 'spoiled.' To achieve this goal, when various AI models, including CNN, AlexNet, ResNet50, GoogleNet, VGG16, and EfficientNetB3, were applied at different epoch values, the notable highest success rates were 99.82%, 99.14%, 98.19%, 99.76%, 99.50%, and 99.94%, respectively.

Keywords: Fruit and vegetable freshness, fruit and vegetable rot detection, precision agriculture, image processing, computer vision

1. Introduction

The agriculture and food industry faces a significant challenge in accurately classifying fruits and vegetables. Proper classification is vital for ensuring food safety and enabling traceability. Categorizing and labeling products streamlines the supply chain monitoring process, allowing for the swift recall of any compromised items. Moreover, classifying fruits and vegetables facilitates tailored marketing strategies catered to specific market segments and consumer preferences, thereby enhancing customer satisfaction and augmenting market share. Furthermore, an automatic classification system that incorporates comprehensive details about fruits and vegetables can guide the selection process to ensure proper nutritional intake. Such a system could greatly benefit children, the elderly, and visually impaired individuals by enhancing their ability to shop independently at supermarkets.

*Corresponding Author: safakkilic@outlook.com

Deep learning emerges as a promising solution to address the challenges prevalent in the agriculture and food industry. Deep learning possess the capability to classify fruits with remarkable accuracy by leveraging their ability to learn intricate features from fruit images. This automated approach yields a significant increase in efficiency compared to traditional manual classification methods (1-2).

A study undertaken in 2020 by Naranjo and collaborators (3) focused on developing diverse methodologies aimed at automating the processes of handling, categorizing, and accurately forecasting the quality parameters of various fruits. In a recent study, Liu et al. (4) selected 11 prominent publications from the literature that garnered significant attention on the topic of fruit analysis and classification. Their findings revealed that only four of these publications employed deep learning or other conventional machine learning techniques. This observation highlights the fact that, despite the considerable interest generated by convolutional neural networks (CNNs), their application in the study and analysis of fruits had not been widely adopted until the year 2017. Similarly, in their research, Sa et al. (5) noted that although convolutional neural networks (CNNs) had garnered substantially more attention compared to other machine learning algorithms, their adoption in fruit-related studies remained limited. Remarkably, their findings indicated that until the year 2016, there were no published articles that utilized CNNs for fruit research and analysis.

Sonwani et al. (6) also designed a prototype to monitor the quality of fruits and vegetables and manage the home storage systems. In this study, a Convolutional Neural Network (CNN) model was first applied to the Fruit 360 dataset. Then, the proposed system monitored the gas emission level, humidity level, and temperature of fruits and vegetables using sensors and actuators to check the spoilage level of the food. The accuracy rate of the model used is 95%. Some studies suggest detecting fruit and vegetable freshness based on artificially extracted features. However, these features have adaptability issues leading to low efficiency. To solve this, Yuan and Chen (7) propose using deep features from pre-trained deep learning models. They utilized a 20-class image set from Kaggle with fresh and spoiled fruits and vegetables. First, images are resized and processed through a pre-trained model. Deep features are then extracted. Principal component analysis reduces these features. Three machine learning methods analyze the reduced features. Experimental results show this method achieves 96.98% accuracy in detecting freshness. It combines deep features from GoogLeNet, DenseNet-201, and ResNeXt-101. This approach promises improved efficiency in detecting fruit and vegetable freshness.

Identifying fruit quality is crucial. It matters for farmers during harvest and classification. It matters for food retailers monitoring quality. It matters for consumers assessing freshness. There is a lack of multiple fruit datasets. These would support real-time fruit quality assessment. To address this gap, Abayomi-Alli et al. (8) introduced a new dataset. It contains fruit images for assessing freshness in real-time. The dataset fills the shortage of multiple fruit datasets. It was created from YouTube time-lapse videos of 11 fruits. The dataset includes classes like "good quality," "medium quality," and

"normal." Among tested models, ResNet18 achieved highest accuracy at 99.8%. Other models tested were ShuffleNet, SqueezeNet, EfficientNet, and MobileNet-V2.

In the paper by Amin et al. (9), the classification method developed using CNN and AlexNet to determine fruit freshness achieved accuracy rates of 98.2%, 99.8%, and 99.3% when applied to three separate datasets. In the study by Kumar et al. (10), a ready-made dataset from Kaggle was used. They created a CNN model using TensorFlow and Keras. When the CNN model was applied to the dataset consisting of 8400 images and 6 classes, it achieved an accuracy of 97.14%.

Mukhiddinov et al. (11) curated their dataset from multiple sources including Kaggle, Google, Bing, and a mobile camera. The dataset comprised fresh and spoiled samples across five different fruits and vegetables. This proposed fruit and vegetable dataset contained a total of 12,000 images. Each fruit and vegetable type was divided into fresh and spoiled classes, resulting in 20 classes altogether. To mitigate misclassification risks, each class consisted of approximately 600 photographs captured under varying lighting conditions like backlit, frontlit, diffuse, and sidelit. Through the application of data augmentation techniques, the training accuracy of the developed YOLOv4 model improved from 72.5% to 75.8%, representing a 3.3% increase. Furthermore, the test accuracy witnessed an enhancement from 69.2% to 73.5%, marking a 4.3% improvement.

Kazi and Panda (12) utilized a dataset comprising 13,599 images of three fruit varieties, both fresh and spoiled specimens. The dataset encompassed six classes in total, with two distinct classes (fresh and spoiled) for each fruit type. This dataset underwent evaluation by models such as ResNet50, AlexNet, and VGG-16. When tested on the dataset, the ResNet50 model achieved an accuracy of 99.7%, AlexNet attained 99.3% accuracy, and VGG-16 exhibited an accuracy of 97.74%. Palakodati et al. (13) conducted a similar study where they utilized a pre-existing dataset available on Kaggle. This dataset comprised 5,989 images categorized into six classes, encompassing fresh and spoiled specimens of apples, bananas, and oranges. The researchers investigated the effects of varying batch sizes (8, 16, 32, 64), epochs (15, 25, 50, 75, 150, 225), and different optimizers (SGD, Adam, Adagrad, RMSprop) during the training process. They adjusted these parameters to develop more accurate models through transfer learning. Upon application, the VGG-16 model achieved an accuracy rate of 89.42%, while VGG-19 attained 76.18% accuracy. The MobileNet model exhibited 68.72% accuracy, and Xception achieved 78.68% accuracy. Furthermore, the CNN model demonstrated an impressive accuracy rate of 97.82%.

Valentino et al. (14) also analyzed and proposed a new design of a computer vision-based method using deep learning with a Convolutional Neural Network (CNN) model to detect the freshness levels of various fruits. In this CNN model, a batch size of 32, MaxPooling2D with pool size (2.2), ReLU as the activation function, and Adam as the optimizer were used. In another study conducted by Nerella et al. (15) to determine the freshness of fruits, features were extracted from fruit images in the CNN layers, and the softmax function was used to classify the images as fresh or rotten. Various state-of-the-art deep learning models, including ResNet50, MobileNetV2, VGG 16, and Inception V3, were

trained and tested on the fruit dataset available on the Kaggle website. The Inception V3 model, which was found to be superior to the other models, achieved an accuracy of 97.1%.

The features of studies conducted to detect the freshness of fruits and vegetables can be seen in the Table 1 below.

Table 1. Data sets, models and accuracy rates used in different studies

Authors	Dataset	Classes	Models	Accuracy
Yuan and Chen (7), 2024	12000 images	20	GoogLeNet, DenseNet-201 and ResNeXt-101	96.98%
Abayomi-Alli et al. (8), 2023	9421 images	32**	ShuffleNet, SqueezeNet, EfficientNet, ResNet18, and MobileNet-V2	99.8%
Amin et al. (9), 2023	Dataset 1: 12000 images	20	CNN, AlexNet	98.2%
	Dataset 2: 13346 images	6	CNN, AlexNet	99.8%
	Dataset 3: 3200 images	16	CNN, AlexNet	99.3%
Kumar et al. (10), 2022	8400 images	6	CNN	97.14%
Mukhiddinov et al. (11), 2022	12000 images	20	YOLOv4	75.8%
Kazi and Panda (12), 2022	13599 images	6	ResNet50, AlexNet, VGG-16	99.7%
Palakodati et al. (13), 2020	5989 images	6	VGG-16, VGG-19, MobileNet, Xception, CNN	97.82%

** Since the "good strawberry" folder is empty in the published dataset, there are 32 classes in this dataset.

In this study, a classification model for fruits and vegetables was developed using a ready-made dataset consisting of fruits and vegetables. Models created using deep learning methods such as CNN, AlexNet, ResNet50, VGG16, GoogleNet and EfficientNetB3 were applied to the data set at different epoch values. It has been observed that the EfficientNetB3 model has higher performance than other models at each epoch value. In this study, it was observed that the model created using EfficientNetB3 had higher accuracy than the models listed in Table 1 in the literature.

2. Materials and Methods

Various functional solutions have been developed to address the issue of overfitting in deep learning models when working with small datasets. These solutions include dropout regularization, batch normalization, transfer learning, and pre-training. Below is a brief description of these overfitting solutions. A study by Kukačka et al. (16) provides a comprehensive review of regularization methods in deep learning.

2.1. Dropout

Dropout is a regularization technique that zeroes the activation values of randomly selected neurons during training (17). This constraint encourages the network to learn more robust features rather than relying on a small subset of neurons for predictions. Tompson et al. (18) have extended this concept to convolutional networks with Spatial Dropout, which drops entire feature maps instead of individual neurons. For example, in Figure 1, in a network model with a threshold of 0.5, applying dropout reduces the number of nodes in the affected hidden layer by half in the subsequent layer (19).

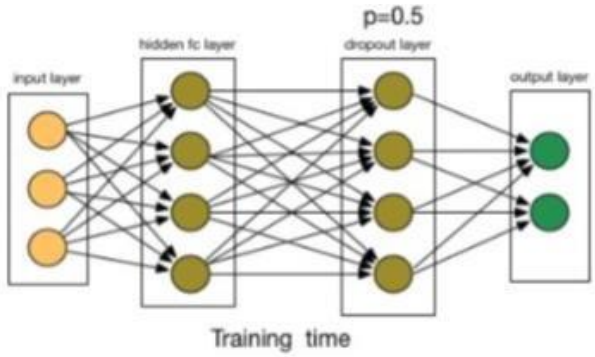

Figure 1. Example of Dropout

2.2. Batch normalization

Batch normalization is another regularization technique that normalizes a set of activations within a layer. It operates by subtracting the batch mean from each activation and then dividing by the batch standard deviation. This normalization technique, often used in conjunction with standardization, is a common method for preprocessing pixel values (20). As seen in the loss graph below, batch normalization significantly reduces losses Figure 2 (21).

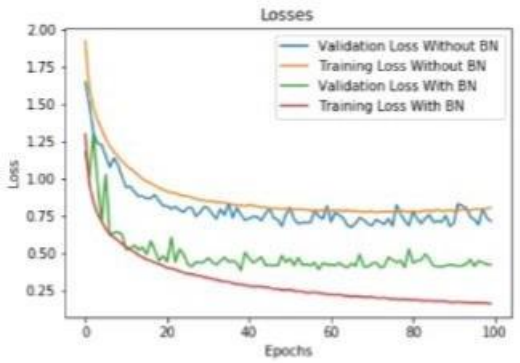

Figure 2. Loss Graphic

2.3. Transfer learning

Transfer learning is an intriguing paradigm used to prevent overfitting. It involves training a network on a large dataset, like ImageNet (22-23-24), and then using those weights as the initial weights for a new classification task. Typically, only the weights from the convolutional layers are transferred, not the entire network, including the fully connected layers. This approach is highly effective because many image datasets share low-level spatial features that are better learned with large amounts of data. Understanding the relationship between transferred data domains remains an ongoing area of research (25). Yosinski et al. (26) found that transferability is primarily hindered by the specialization of upper layer neurons and the difficulties associated with separating co-adapted neurons. Transfer learning leverages prior knowledge to achieve higher performance with less training data and allows models to learn more quickly. We can summarize transfer learning as follows (27) Figure 3.

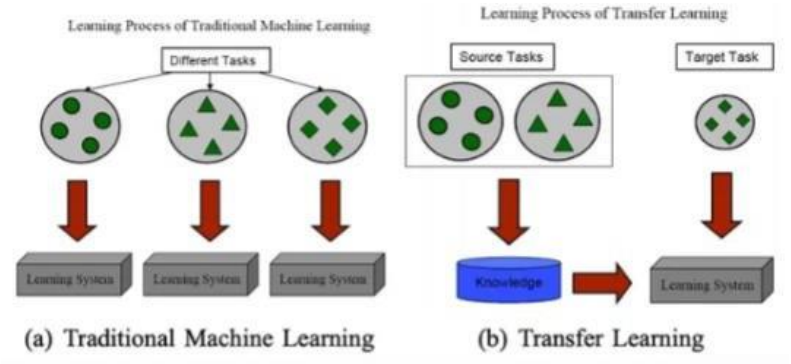

Figure 3. Transfer Learning

2.4. Pre-training

Pre-training is conceptually similar to transfer learning. In pre-training, a network architecture is defined and then trained on a large dataset such as ImageNet (24-28). This differs from transfer learning because, in transfer learning, not only the network architecture like VGG-16 (29) or ResNet (30) needs to be utilized but also the weights must be transferred. Pre-training allows flexibility in designing network architectures while initializing the weights using large datasets.

Data augmentation (31, 32): Data augmentation is a commonly used technique in machine learning and deep learning. It aims to expand and diversify the existing training dataset through various methods, helping the model to perform better and be more generalizable. Data augmentation plays a crucial role, especially in situations where the amount of data is limited, by preventing overfitting and achieving more effective results. In image processing, data augmentation includes modifications like rotation, flipping, scaling, random cropping, and color variations.

2.5. Activation function

In CNN applications, sigmoid, tanh, and rectified linear unit (ReLU) are among the available activation function options. Compared to other functions, the ReLU function is much faster to compute and still delivers good results. The definition of the ReLU function is as follows: it outputs the input directly if it is positive; otherwise, it outputs zero. The graph of the ReLU function is typically a line that passes through the origin with a slope of 1 for positive inputs and remains at zero for negative inputs. The results of the experiment are demonstrated in Figure 4.

$$f(x) = \begin{cases} x, & x > 0 \\ 0, & else \end{cases} \quad (1)$$

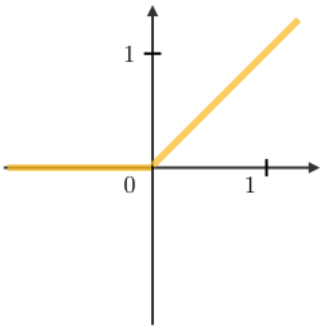

Figure 4. ReLU Activation Function

2.6. Confusion matrix

In machine learning, the confusion matrix (33) is used to evaluate the performance of a created model. The confusion matrix provides a deeper understanding of the model's performance, errors, and weaknesses. Consequently, the confusion matrix allows for further analysis and fine-tuning of the model. The basic structure of the confusion matrix is as described in the Figure 5.

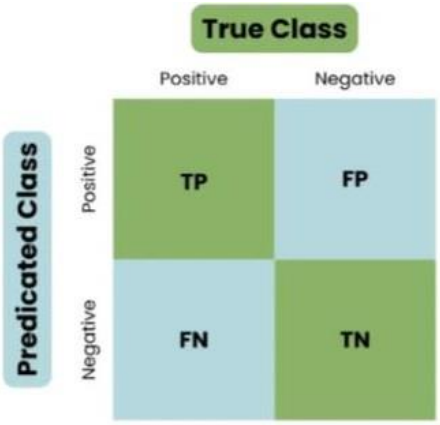

Figure 5. Confusion Matrix

The important metrics used to measure a model's performance using the confusion matrix are as follows:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (2)$$

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

$$Precision = \frac{TP}{TP+FP} \quad (4)$$

$$Specificity = \frac{TN}{TN+FP} \quad (5)$$

$$F1\ Score = \frac{2\times Recall\ \times Precision}{Recall\ +Precision} \quad (6)$$

2.7. Dataset

In a study conducted by Alli et al. (8) in 2023, an original dataset was created using images from time-lapse videos found on YouTube, featuring eleven different types of fruits and vegetables, excluding fresh strawberry photos. The dataset includes images of bananas, cucumbers, grapes, persimmons, papayas, peaches, pears, bell peppers, strawberries, tomatoes, and watermelons in various states: rotten, medium quality, and fresh. The original dataset can be accessed at "https://doi.org/10.5281/zenodo.7224690" address.

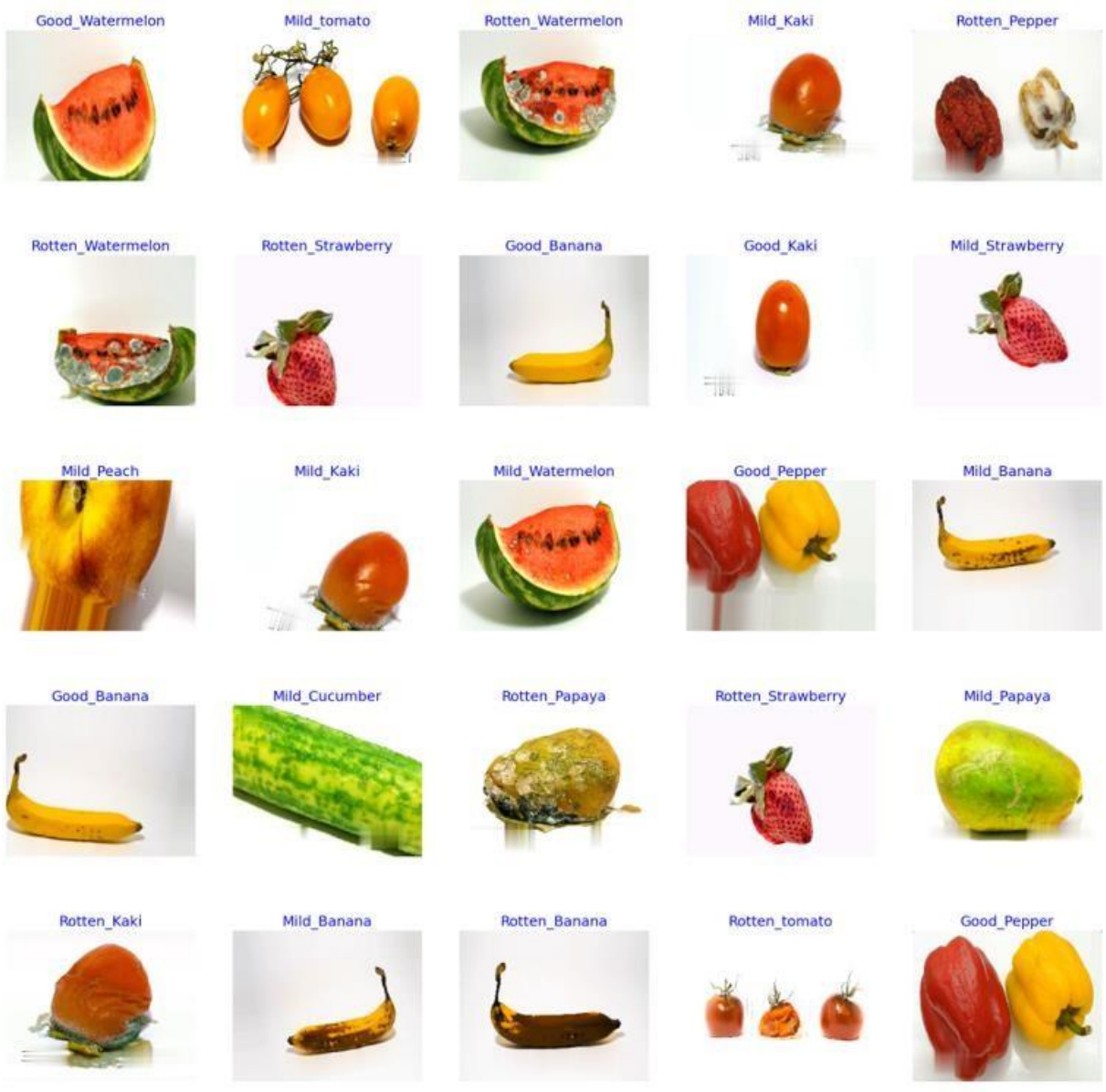

Figure 6. Some images in the dataset

This comprehensive dataset can be used by researchers and developers for various purposes, such as the quality classification of fruits and vegetables. The images of fruits and vegetables in the dataset, with their diversity and wide class distribution, provide a rich resource for machine learning and artificial intelligence applications. This dataset allows for the development of models that can

distinguish between different maturity levels of fruits and vegetables, detect signs of decay, and assist in the selection of fresh produce. An example of the data set related to air tresimerlin is shown in Figure 6. In this study, for the fresh strawberry images, 122 images were taken from the Fruit 360 dataset (34) available on the Kaggle platform, resulting in a total of 9492 images used in the dataset. The images in the Good Strawberry folder are 100x100 pixels, while all other images are 1280x720 pixels. Below, images of some fruits and vegetables from the dataset are shown.

2.8. Models

2.8.1. CNN architecture

CNN, also known as convolutional neural networks, occupy a special place among deep learning models. These networks form a subclass of multi-layered perceptrons. Its main purpose is to extract important features from high-dimensional data such as visual and auditory data, and to perform tasks by learning from these features. Convolutional neural networks have a wide range of applications such as visual recognition, audio processing, natural language processing, and biomedical image analysis. However, their most striking successes have been in the field of visual data processing. The basis of this success lies in the network's ability to perceive and classify complex visual patterns. The convolution kernel is not only a fundamental element of Convolutional Neural Networks (CNNs), but also of many other computer vision algorithms. In this process, a small number matrix (called a kernel or filter) is taken, scanned over our image, and transformed according to the values in the filter. In general, operations for standard convolution are as shown in Figure 7.

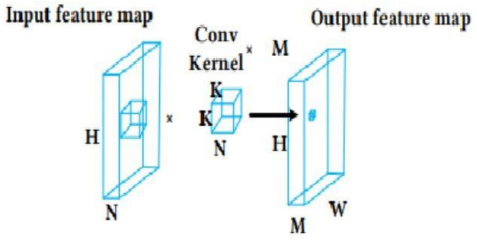

Figure 7. Standard Convolution Operations (35)

The feature map values of an image are calculated according to the following formula. In this formula, the input image is denoted by f and the kernel is denoted by H. The row and column indices of the resulting matrix are indicated by M and N, respectively.

$$G[M,N] = (f * H)[M,N] = \sum_j \sum_k H[j,k] f[M-j, N-k] \quad (7)$$

During the training of convolutional neural networks, various learning algorithms are used to increase the accuracy of the model and minimize error rates. The most well-known and widely used of these algorithms is the backpropagation algorithm. Backpropagation aims to reduce the difference between the predictions made by the model and the actual values, i.e., the error, by propagating it backward from the output layer to the input layer of the model. In this process, the weights in each layer of the network are updated in a way that contributes to the reduction of the error. These updates are

performed by calculating the derivatives of the error and propagating these derivatives backward through the network layers. Thanks to this method, convolutional neural networks can successfully perform challenging tasks such as recognizing, classifying complex visual patterns, and even detecting the locations of objects. In general, the CNN model architecture is as shown in Figure 8 (36).

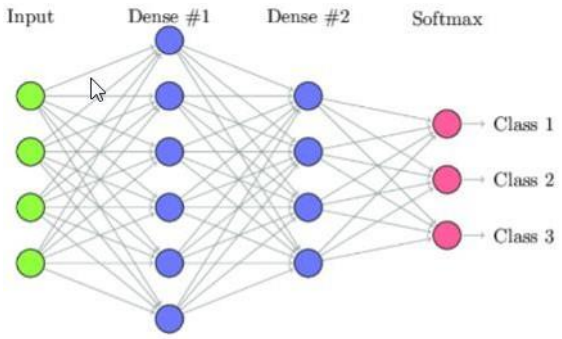

Figure 8. CNN model

In this study all images were prepared in 64x64 pixel size and RGB format, considering the important features and including too many features would extend the training time. To evaluate the performance of the model, trials were conducted with batch size values varying as 16, 32, 64, 128, and 512, and the best accuracy rate was achieved with a batch size of 32. The relationship between these values and the accuracy of the training set is shown in a bar graph in Figure 9. From this analysis, it was understood that there are 7,584 images in the training set, and they are divided into 33 different classes in total. When the same arrangements were applied to the validation set, it was determined that it contained 937 images and again 33 different classes.

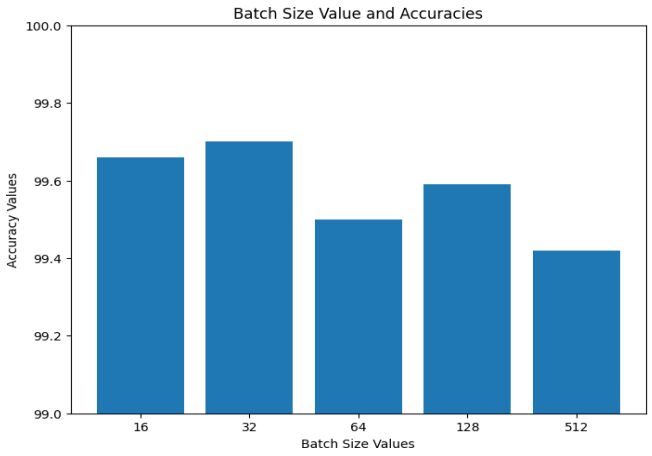

Figure 9. Batch size value and accuracies

In this CNN model, 64 filters of 3x3 type were used and ReLU was used as the activation function. This process was applied twice. 2x2 type MaxPooling has been applied. These operations were applied twice to reduce the image size. Flatten was used to flatten matrix data. Then a hidden layer with 512 neurons and 256 neurons was added. Dropout was used to prevent overlearning. Since there are 33 classes in the data set, a layer with 33 neurons with softmax activation function was created in the result layer. In this way, a model with 5790689 parameters was obtained. Operations such as maxpooling,

flatten, dropout, softmax, including the ReLU activation function in this CNN model, were also used in other models created in this study.

The created CNN model architecture is as in Table 2 below. When the model was tested at different epoch values, the highest success rate was achieved as 99.82%.

Table 2. CNN model

Layers	Output Size	Number of Parameters
Conv2d-20	(None, 64, 64, 64)	1792
Conv2d-21	(None, 60, 60, 64)	36928
Max-pooling2d-10	(None, 30, 30, 64)	0
Conv2d-22	(None, 28, 28, 64)	36928
Conv2d-23	(None, 26, 26, 64)	36928
Max-pooling2d-11	(None, 13, 13, 64)	0
Flatten-5	(None, 10816)	0
Dense-15	(None, 512)	5538304
Dense-16	(None, 256)	131328
Dropout-5	(None, 256)	0
Dense-17	(None, 33)	8481
		Total Parameter: 5790689

2.8.2. ResNet architecture

Among the CNN architectures, ResNet occupies an important position with the innovation it brought. Until ResNet, the main approach in the developed architectures was to increase the depth by increasing the number of layers. However, increasing the depth of the network does not work after a point due to the vanishing gradient problem. In fact, many studies have observed that increasing the depth of the network too much decreases the success rate. For example, Figure 10 shows the error rates of two networks with 20 and 56 layers.

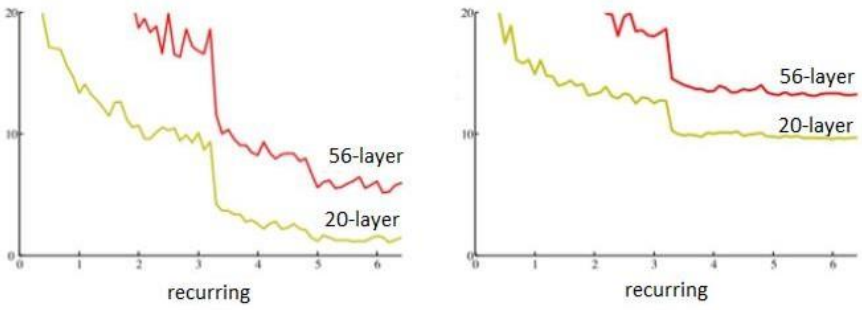

Figure 10. Number of layers error relationship in ResNet

The main idea of the ResNet architecture, as shown in Figure 11 (35) is to enable the residual block to transfer the residual values to the subsequent layers by skipping one or more layers. In this way, ResNet does not proceed in a flat flow like previous deep learning architectures, and the vanishing gradient problem is solved in the ResNet architecture by transferring the residual values. In this study, we will use ResNet50.

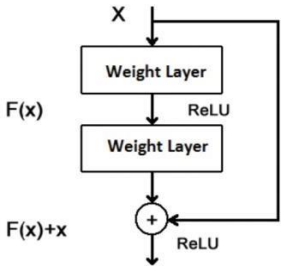

Figure 11. The ResNet architecture

ResNet50 is a 50-layer deep convolutional neural network. This network consists of 5 stages, each comprising a convolutional and an identity block. Each convolutional and identity block consists of three layers. The ResNet50 architecture contains more than 23 million trainable parameters. The upper layers of ResNet50 are not frozen, meaning they learn through backpropagation (37). However, the remaining layers of ResNet50 remain frozen (38). The cross-entropy loss function for the ResNet50 model is expressed as follows:

$$Loss = -(1/n) \sum_{i=0}^{n} logP(N) \quad (8)$$

Here n represents the output size of the classification network.

In this papaer, the images in the dataset were standardized to have dimensions of 100x100 pixels. They could be standardized according to any size. Since the smallest images in our dataset were 100x100 in size, it was preferred to convert all images to 100x100 dimensions. At the same time, a label set was created for each image based on the class numbers. Thanks to this methodology, as can be seen in Figure 12 below, the class number of the data was determined as 3.

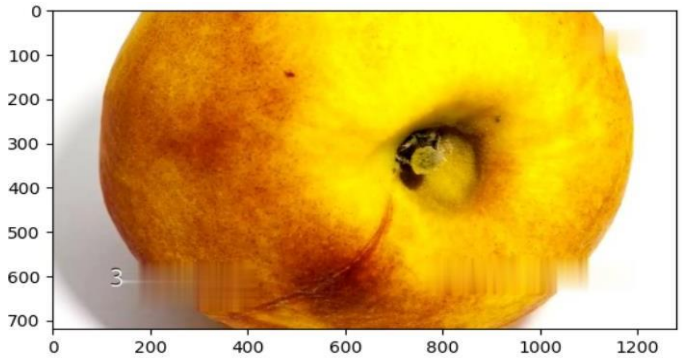

Figure 12. Determining data class with ResNet50

While the ReLU function was preferred as the activation function in the model, the softmax function was used in the output layer. It was observed that the developed model had a total of 24,669,089 parameters. In order to increase the efficiency of the model, an optimization technique called 'adam optimizer' was used. Within the scope of data augmentation, the images were rotated at 20-degree angles, zooming was possible with the zoom-range feature, and horizontal-flip and vertical-flip features were applied to half of the images for horizontal and vertical flipping operations. Additionally, the images were converted from RGB format to BGR format. As a result of these operations, a medium-quality tomato image is presented in Figure 13.

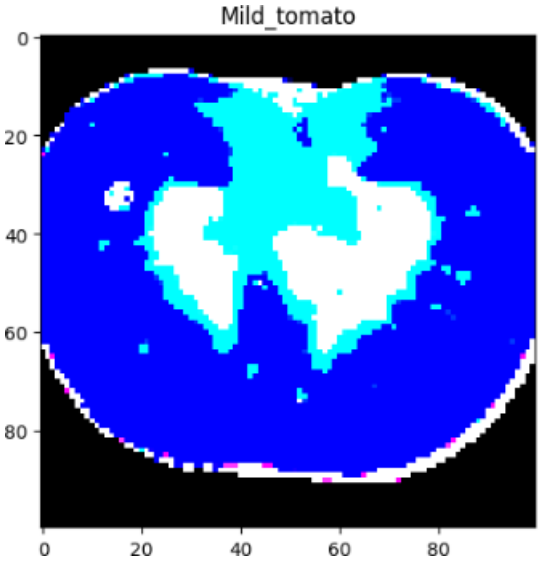

Figure 13. Mild tomato

On the other hand, the complexity matrices of the unnormalized training dataset is shown in Figure 14.

```
[[136   0   0   0   0   0   0   0   0   0   7   0   0   0   0   0   0
    0   0]
 [  0 200   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0]
 [  0   0 181   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0]
 [  0   0   0 436   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0]
 [  0   0   0   0  82   0   0   0   0   0   0   0   0   0   0  22   0
    0   0]
 [  0   0   0   0   0 340   0   0   0   0   0   0   0   0   0   0   0
    0   0]
 [  0   0   0   0   0   0 403   0   0   0   0   0   0   0   0   0   0
    0   0]
 [  0   0   0   0   0   0   0  38   0   0   0   0   0   0   0   0   0
    0   0]
 [  0   0   0   0   0   0   0   0  97   0   0   0   0   0   0   0   0
    0   0]
 [  0   0   0   0   0   0   0   0   0  20   0   0   0   0   0   0  20
    0   0]
 [  0   0   0   0   0   0   0   0   0   0 460   0   0   0   0   0   0  20
    0   0]
 [  0   0   0   0   0   0   0   0   0   0   0  76   0   0   0   0   0
    0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0 276   0   0   0   0
    0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0 113   0   0   0
   42   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0 173   0   0
    0   7]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0 179   0
    0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0]
 [  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
    0   0]]
```

Figure 14. Confusion matrix of training dataset with ResNet50

The applied ResNet50 features are as shown in Table 3. When the applied model was trained on the dataset, it was found that a maximum accuracy rate of 98.19% was achieved.

Table 3. Applied ResNet50 model

Image Shape	100x100
Activation Function	ReLU
Number of Parameter	24669089
Optimizer	Adam
Data Augmantation	Rotate-Zoom-Flip

2.8.3. VGG16 architecture

The VGG16 architecture proposed by Simonyan and Zisserman (29) was among the top 5 most successful models in the 2014 ILSVRC competition [URL: www.image-net.org/challenges/LSVRC\]. VGG16 was designed to reduce the training time by decreasing the number of parameters. Unlike the previous successful CNN architecture AlexNet, it used 3x3 filters and the VGG16 architecture contains consecutive 2 or 3 convolutional layers. The VGG16 architecture has a total of 13 convolutional layers and 5 pooling layers. At the final stage of the VGG16 architecture, there are two fully connected layers consisting of 4096 neurons, followed by a softmax layer with the target class dimension (1000 neurons) (39). The VGG16 model is shown in Figure 15.

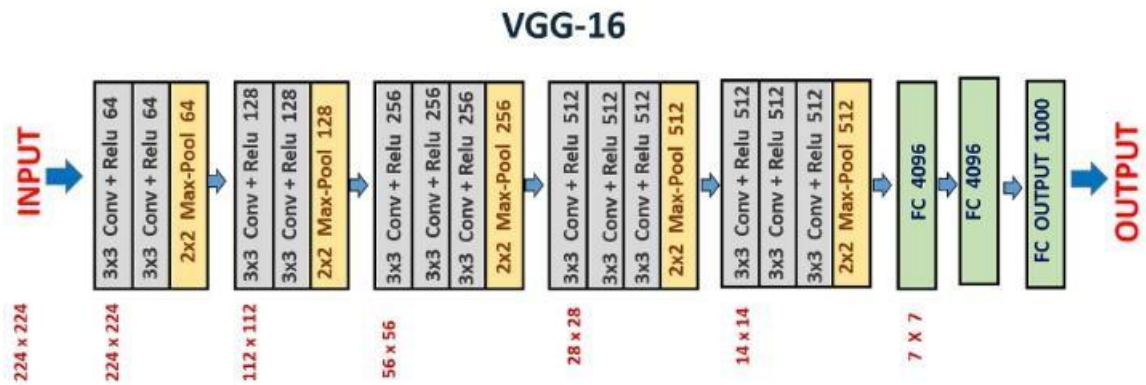

Figure 15. VGG16 Model

In this study, to apply the VGG16 model, each image in the dataset was resized to 224x224 dimensions, and a model with 135,201 parameters was created. RMSprop was used as the optimizer. The features of the used VGG16 model are as shown in Table 4. With the created VGG16 model, an accuracy value of 99.50% was achieved.

Table 4. VGG16 model

Image Shape	224x224
Optimizer	rmsprop
BatchSize	64

2.8.4. EfficientNet architecture

Unlike the architectures developed before it, the EfficientNet architecture did not solely focus on increasing the success rate; it also aimed to provide a more efficient architecture. Indeed, it has been observed that a more efficient architecture also raises the model's accuracy percentage. In this context, during the design of the EfficientNet architecture, a scaling study was carried out on the dimensions of depth, width, and resolution, which affect the efficiency of the architecture. The scaling in the EfficientNet architecture is depicted in Figure 16 (40). For example, if we want to use 2^N times more computational resources, we can increase the network depth by α^N, width by β^N, and image dimension by γ^n over the original small model, with fixed coefficients determined by a small grid search on the original dataset (40).

The average size of the images was determined, and the classes containing fewer than 350 images were augmented so that the number of images in each class was completed to 350. Max pooling operation was applied to deepen the feature learning of the model, and fully connected layers were added to model more complex relationships. Batch Normalization techniques were utilized to accelerate the training process, minimize losses, and improve learning performance. The Dropout method was employed to prevent overfitting. When the model was applied to the dataset for 25 epochs, the total time took 9 hours 32 minutes 40.50 seconds, and an extraordinary accuracy value of 99.94% was achieved. The Loss, accuracy, F1-score, and AUC graphs obtained with this model are shown in Figure 17.

Additionally, when the model was tested, the tested images and the corresponding values are shown in Table 5.

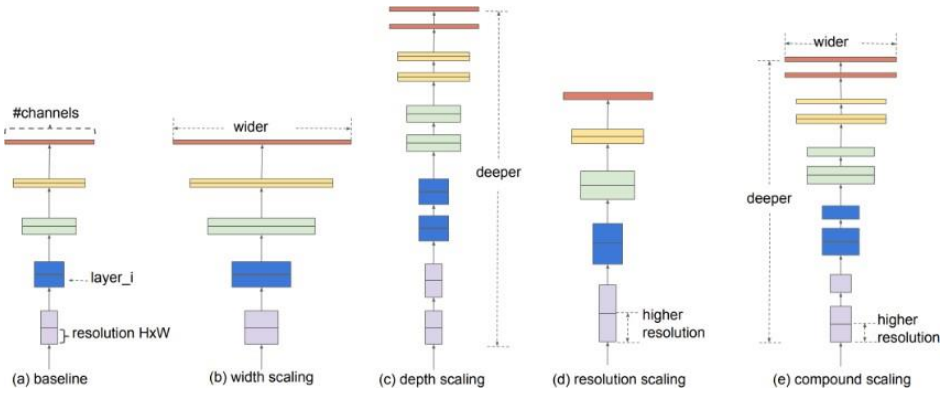

Figure 16. EfficientNet Model Scaling. (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of network width, depth, or resolution. (e) is our proposed compound scaling method that uniformly scales all three dimensions with a fixed ratio

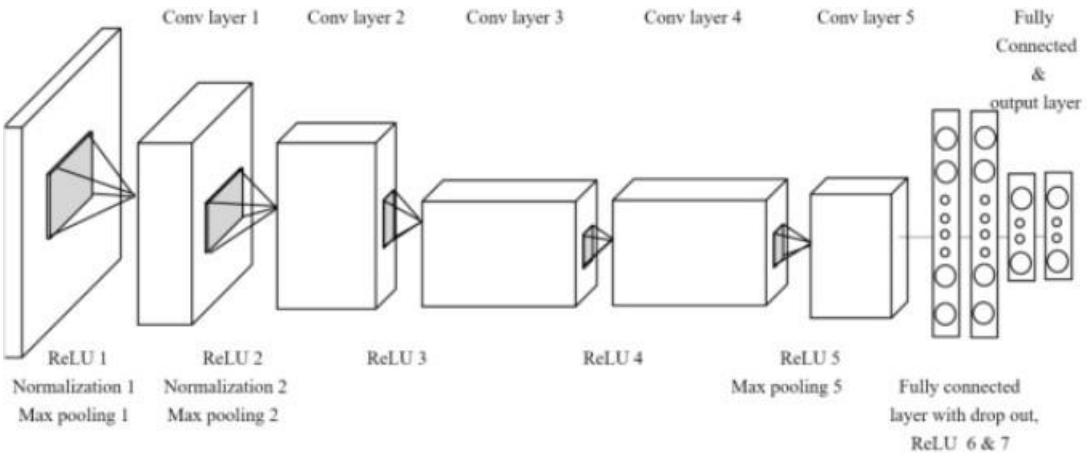

Figure 18. AlexNet architecture

Table 5. Testing in EfficientNet model

	precision	recall	f1-score	support
Good_Banana	1.0000	1.0000	1.0000	19
Good_Cucumber	1.0000	1.0000	1.0000	25
Good_Grape	1.0000	1.0000	1.0000	24
Good_Kaki	1.0000	1.0000	1.0000	55
Good_Papaya	1.0000	0.9231	0.9600	13
Good_Peach	1.0000	1.0000	1.0000	43
Good_Pear	1.0000	1.0000	1.0000	51
Good_Pepper	1.0000	1.0000	1.0000	6
Good_Strawberry	1.0000	1.0000	1.0000	13
Good_Watermelon	1.0000	1.0000	1.0000	6
Good_tomato	1.0000	1.0000	1.0000	60
Mild_Banana	1.0000	1.0000	1.0000	11
Mild_Cucumber	1.0000	1.0000	1.0000	35
Mild_Grape	1.0000	1.0000	1.0000	20
Mild_Kaki	1.0000	1.0000	1.0000	24
Mild_Papaya	0.9615	1.0000	0.9804	25
Mild_Peach	1.0000	1.0000	1.0000	15
Mild_Pear	1.0000	1.0000	1.0000	50
Mild_Pepper	1.0000	1.0000	1.0000	3
Mild_Strawberry	1.0000	1.0000	1.0000	13
Mild_Watermelon	1.0000	1.0000	1.0000	6
Mild_tomato	1.0000	1.0000	1.0000	44
Rotten_Banana	1.0000	1.0000	1.0000	35
Rotten_Cucumber	1.0000	1.0000	1.0000	13
Rotten_Grape	1.0000	1.0000	1.0000	30
Rotten_Kaki	1.0000	1.0000	1.0000	34
Rotten_Papaya	1.0000	1.0000	1.0000	42
Rotten_Peach	1.0000	1.0000	1.0000	59
Rotten_Pear	1.0000	1.0000	1.0000	10
Rotten_Pepper	1.0000	1.0000	1.0000	66
Rotten_Strawberry	1.0000	1.0000	1.0000	11
Rotten_Watermelon	1.0000	1.0000	1.0000	15
Rotten_tomato	1.0000	1.0000	1.0000	95
accuracy			0.9990	971
macro avg	0.9988	0.9977	0.9982	971
weighted avg	0.9990	0.9990	0.9990	971

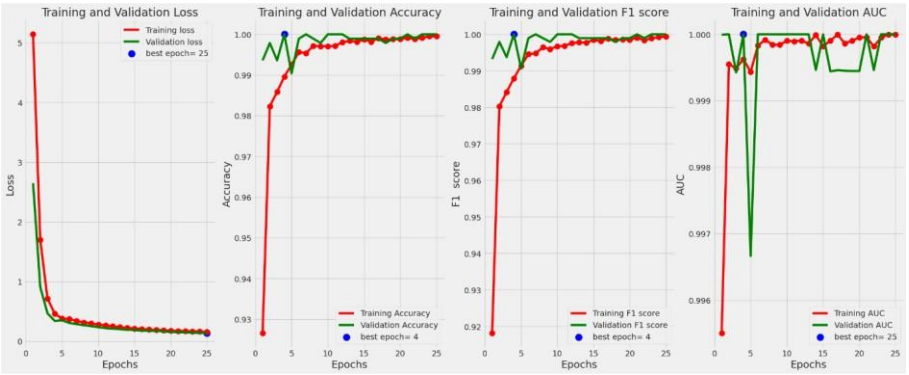

Figure 17. Loss, accuracy, F1 and AUC graphs at epoch 25 in the EfficientNetB3 model

2.8.5. AlexNet architecture

In the large-scale visual recognition competition held in 2012, the AlexNet deep learning architecture ranked first by reducing the error rate in pattern recognition from around 26% to around 15%. This performance naturally made AlexNet one of the most well-known deep learning architectures. Activation functions (ReLU) are used for non-linear functions. This activation function is used to shorten the training time as it is faster than the classic tanh function. A dropout layer is used during the training process to prevent overfitting and stagnation. The gradient descent model is used for weight delays and momentum values. As can be seen in Figure 18, the AlexNet, consisting of 25 layers, includes

5 main convolutional layers within the deep learning network, and typically a ReLU layer is used as an activation layer after each convolutional layer. In addition, there are input layer, normalization layer, pooling layer, dropout layer, fully connected layer, SoftMax layer, and output layer. (41-42-43).

In this model, the following formula is used to calculate the output at each layer.

$$Output = \frac{W-K+2(P)}{S} + 1 \quad (9)$$

Here W is the input size, K is the size of the filter kernel, P is the number of padding, and S is the number of steps (44).

To apply the AlexNet model in this study, all images were set to 227x227 pixel size, and 20% of the dataset was separated for testing, and 80% for training. The model developed through Keras contains a total of 28,896,145 parameters. During the development process of the model, the Batch Normalization technique was applied, and the Adam optimization algorithm was used, and the model was tested on the training dataset for 10 epochs. As a result, it was observed that the model was successfully trained with an accuracy rate of 99.14%. The features of the applied AlexNet model are as shown in the following Table 6.

Table 6. AlexNet model

Image Shape	227x227
Normalisation	Batch
Number of Parameter	28896145
Optimizer	Adam

2.8.6. GoogleNet architecture

It has a complex architecture. With a depth of 22 layers, GoogleNet has a architecture consisting of a total of 144 layers. By using filters of different sizes with the Inception module, it presents a formation that differs from previous deep learning architectures. These filtering operations are filters used for dimensionality reduction. It contains 12 times fewer parameters compared to AlexNet, and the number of layers may vary depending on the independent building blocks used. The filter elements in the Inception module are 1x1, 3x3, and 5x5. Departing from the consecutive layered architecture found in other deep learning architectures, a deep architecture has been established. A modular filtering logic has been introduced to create this depth (45). The layers of GoogleNet are as follows:

Table 7. The layers of GoogleNet

Layer	Number
Convolution	57
ReLU	57
Pooling	14
LRN	2
Concat	9
Dropout	1
Inner Product	1
SoftMax	1
Total	142

In this paper, the images in the dataset were resized to 224x224 dimensions. When the model obtained with 22,335,809 parameters, using the Keras and TensorFlow libraries, was applied to the dataset, an accuracy value of 99.76% was achieved. The features of the GoogleNet model used in this study are as shown in the following Table 8.

3. Results

Table 8. GoogleNet model

Layers	Output Shape	Number of Parameter
Input-I	(None, 224, 224, 3)	0
Inception-v3	(None, 5, 5, 2048)	21802784
Global-average-pooling2d	(None, 2048)	0
Dense-4	(None, 256)	524544
Dense-5	(None, 33)	8481
		Total parameter: 22335809

Information including the comparison of CNN, ResNet 50, AlexNet, GoogleNet, VGG16 and EfficientNetB3 models according to the accuracy values obtained at different epoch values on the data set is shown in the Table. 9.

Table 9. Models and features and accuracy in different epochs

Models	Number of Parameters	10 Epoch	15 Epoch	20 Epoch	25 Epoch
CNN	5.790.689	0.9920	0.9975	0.9982	0.9962
ResNet-50	24.669.089	0.9671	0.9781	0.9792	0.9819
AlexNet	28.896.145	0.9797	0.9899	0.9914	0.9903
GoogleNet	22.335.809	0.9949	0.9957	0.9976	0.9974
VGG16	138.357.544	0.9875	0.9850	0.9925	0.9950
EfficientNetB3	11.101.257	0.9972	0.9981	0.9993	0.9994

From the Table 9 containing the number of parameters used in the CNN, ResNet-50, AlexNet, GoogleNet, VGG16 and EfficientNetB3 models on the data set, it can be seen that the model with the highest number of parameters is AlexNet, and the model with the fewest parameters is VGG16. Additionally, the EfficientNet model is the second model with the most parameters.

Comparison of the models according to their highest accuracy values can be seen in the bar graph in Figure 19 below. As can be seen from the table, the highest accuracy value was obtained as 99.94% with the EfficientNetB3 model at epoch 25.

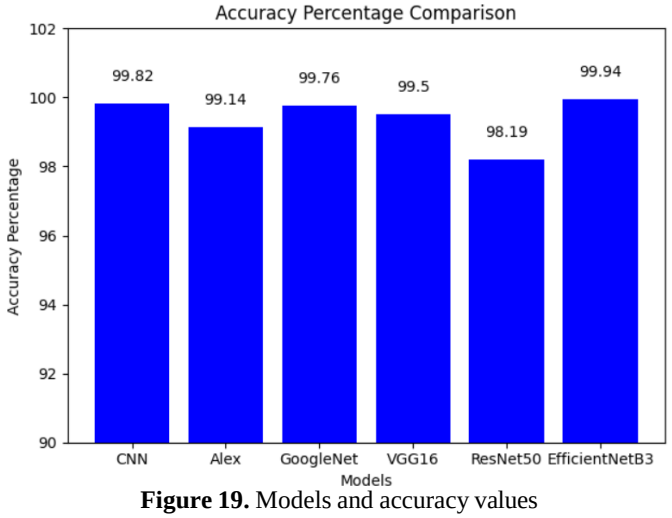

Figure 19. Models and accuracy values

4. Conclusion and Discussion

The classification of fruits and vegetables, which is greatly needed in the agricultural sector, is among the popular studies in the field of image recognition in deep learning recently. In this study, the classification of fruits and vegetables was performed with six deep learning models, and, the EfficientNetB3 model was by far the best performer, with an impressive success rate of 99.94%. In the study by Abayomi-Alli et al. (8), which constituted the data set used in this study, the highest accuracy value was obtained as 99.8% with ResNet-18. In this respect, the accuracy value of the model obtained in this study is higher than the accuracy value obtained by Abayomi-Alli et al. (8). However, the working time of the models is a bit long. In future studies, the working time can be shortened by making some adjustments in the models. Additionally, in future studies, classification can be performed using video images.

Acknowledgment

This study was supported by the Scientific Research Projects Coordination Unit of Kayseri University, Turkey within the scope of project FKB-2024-1134.

References

1. Ozdemir C, Dogan Y. Advancing brain tumor classification through MTAP model: an innovative approach in medical diagnostics. Medical & Biological Engineering & Computing 2024;.1(12): 2165–2176. Available from: https://doi.org/10.1007/s11517-024-03064-5

2. Ozdemir C, Dogan Y. Advancing early diagnosis of Alzheimer's disease with next-generation deep learning methods. Biomedical Signal Processing and Control 2024; 96, 106614.3. Arens AA. Auditing in Australia : an integrated approach. 5th ed. Frenchs Forest: Pearson Education Australia; 2002.

3. Naranjo-Torres J, Mora M, Hernández-García R, Barrientos RJ, Fredes C, Valenzuela A. A review of Convolutional Neural Network applied to fruit image processing. Applied Science 2020.

4. Liu F, Snetkov L, Lima D. Summary on Fruit identification methods: a literature review. In: Proceedings of the 2017 3rd International Conference on Economics, Social Science, Arts, Education and Management Engineering (ESSAEME 2017). Atlantis Press, July 2017; Huhhot, China.

5. Sa I, Ge Z, Dayoub F, Upcroft B, Perez T, McCoo C. Deep Fruits: A fruit detection system using deep neural networks. Sensors 2016; 16(1222).

6. Sonwani E, Bansal U, Alroobaea R, Baqasah A.M, Hedabou M. An artificial intelligence approach toward food spoilage detection and analysis. Frontiers in Public Health 2021; 9.

7. Yuan Y, Chen X. Vegetable and fruit freshness detection based on deep features and principal component analysis. Current Research in Food Science 2024; 8. Available from: https://doi.org/10.1016/j.crfs.2023.100656

8. Abayomi-Alli O.O, Damaševičius R, Misra S, Abayomi-Alli A. FruitQ: a new dataset of multiple fruit images for freshness evaluation. 2023. Available from: https://doi.org/10.1007/s11042-023- 16058-6

9. Amin U, Shahzad M.I, Shahzad A, Shahzad M, Khan U, Mahmood Z. Automatic fruits freshness classification using CNN and transfer learning. Applied Sciences 2023; 13(8087). Available from: https://doi.org/10.3390/app13148087

10. Kumar T.B, Prashar D, Vaidya G, Kumar V, S. D. Kumar S.D, Sammy F. A novel model to

detect and classify fresh and damaged fruits to reduce food waste using a deep learning technique. Hindawi Journal of Food Quality 2022. Available from: https://doi.org/10.1155/2022/4661108

11. Mukhiddinov M, Muminov A, Cho J. Improved classification approach for fruits and vegetables freshness based on deep learning. Sensors 2022. 22. Available from: https://doi.org/10.3390/s22218192

12. Kazi A, Panda S.P. Determining the freshness of fruits in the food industry by image classification using transfer learning. Multimedia Tools and Applications 2022. 81; p. 7611-7624. Available from: https://doi.org/10.1007/s11042-022-12150-5

13. Palakodati SSS, Chirra VRR, Dasari Y, Bulla S. Fresh and rotten fruits classification using CNN and transfer learning. Revue d'Intelligence Artificielle 2020. 34(5); p. 617-622. Available from: https://doi.org/10.18280/ria.340512

14. Valentino F, Cenggoro TW, Pardamean B. A design of deep learning experimentation for fruit freshness detection. IOP Conference Series: Earth and Environmental Science 2021. 794. doi= 10.1088/1755-1315/794/1/012110

15. Tanuia Nerella JNVD, Nippulapalli VK, Nancharla S, Vellanki LP, Suhasini PS. Performance comparison of deep learning techniques for classification of fruits as fresh and rotten. In: International Conference on Recent Advances in Electrical, Electronics, Ubiquitous Communication, and Computational Intelligence (RAEEUCCI), 2023. doi= 10.1109/RAEEUCCI57140.2023.10134242

16. Kukačka J, Golkov V, Cremers D. Regularization for deep learning: a taxonomy. ArXiv 2017. 1710.10686v1.

17. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 2014. 15.

18. Tompson J, Goroshin R, Jain A, LeCun Y, Bregler C. Efficient object localization using convolutional networks. ArXiv 2015. 1411.4280v3

19. Necmettin Ç. Derin öğrenme uygulamalarında başarım iyileştirme yöntemleri (Regularization). Linkedin 2017. Available from: https://www.linkedin.com/pulse/derin-%C3%B6%C4%9Frenme-uygulamlar%C4%B1nda-ba%C5%9Far%C4%B1m-iyile%C5%9Ftirme-necmettin-%C3%A7arkac%C4%B1/

20. Ioffe S, Szegedy C. Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv 2015. 1502.03167v3.

21. Zehra. Batch normalization. Medium 2021. Available from: https://zerzavot.medium.com/batch- normalization-b7d73c9cc6df

22. Weiss K, Khoshgoftaar TM, Wang D. A survey of transfer learning. Journal of Big Data 2016. (9). doi: 10.1186/s40537-016-0043-6.

23. Shao L, Zhu F, Li X. Transfer learning for visual categorization: a survey. IEEE Trans Neural Netw Learn Syst. 26(5), doi: 10.1109/TNNLS.2014.2330900

24. Deng J, Dong W, Socher R, Li-Jia Li, Li K, Fei-Fei L. ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), Miami, Florida, USA, 2009. doi: 10.1109/CVPR.2009.5206848

25. Zamir A, Sax A, Shen W, Guibas L, Malik J, Savarese S. Taskonomy: disentangling task transfer learning. arXiv 2018. 1804.08328v1.

26. Yosinski J, Clune J, Bengio Y, Lipson H. How transferable are features in deep neural networks?". ArXiv 2014, 1411.1792v1.

27. Yiğit G, Yeğin MN. Öğrenme aktarımı/transfer learning. Nova Research Lab. 2020. Available from: https://medium.com/novaresearchlab/%C3%B6%C4%9Frenme-aktar%C4%B1m%C4%B1-transfer-learning-c0b8126965c4

28. Erhan D, Bengio Y, Courville A, Manzagol P-A, Vincent P, Bengio S. Why does unsupervised pre- training help deep learning?. Journal of Machine Learning Research 2010. (11); p. 625—660.

29. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv 2014. 1409.1556v6.

30. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. arXiv 2015. 1512.03385v1, 2015.

31. Yılmaz AO. Data augmentation: veri artırma yöntemleri ve uygulamaları. Medium 2023. Available from: https://aoyilmaz.medium.com/data-augmentation-veri-art%C4%B1rma-y%C3%B6ntemleri- ve-uygulamalar%C4%B1-4dd33e12bf1d

32. _Ozdemir C, Dogan Y, Kaya Y. RGB-Angle-Wheel: A new data augmentation method for deep learning models. Knowledge-Based Systems 2024. 291(111615).

33. Özden S. Confusion matrix (Karışıklık matrisi). Medium 2024. Available from: https://medium.com/@serapozden922/confusion-matrix kar%C4%B1%C5%9F%C4%B1kl%C4%B1k-matrisi-62c43b8ad2b0#:~:text=Kar%C4%B1%C5%9F%C4%B1kl%C4%B1k%20matrisi%2C%20bir%20modelin%20performans%C4%B1n%C4%B1,daha%20derin%20bir%20anlay%C4%B1%C5%9F%20sa%C4%9Flar.

34. Horea M, Mihai O. Fruit recognition from images using deep learning. Acta Universitatis Sapientiae, Informatica 2018. 10(1).

35. Kılıç Ş, Askerzade İ, Kaya Y. Using ResNet transfer deep learning methods in person identification according to physical actions. IEEE 2020. 8. doi=10.1109/ACCESS.2020.3040649

36. Tüfekçi M, Karpat F. Derin öğrenme mimarilerinden konvolüsyonel sinir ağları (CNN) üzerinde görüntü işleme-sınıflandırma kabiliyetininin arttırılmasına yönelik yapılan çalışmaların incelenmesi. In: International Conference on Human-Computer Interaction. Optimization and Robotic Applications, 2019.

37. Joseph JL, Kumar VA, Mathew SP. Fruit classification using deep learning. Springer 2021.

38. Arrabelly SBR, S. Juliet S. Transfer learning with ResNet-50 for malaria cell-image classification. In: Proceedings of the International Conference on Communication and Signal Processing (ICCSP), 2019; Melmaruvathur, India,

39. Ulusoy O, Akgül CB, Anarım E. Improving image captioning with language modeling regularizations. In: 2019 Innovations in Intelligent Systems and Applications Conference (ASYU), 2019; İzmir.

40. Tan M, Le QV. Efficientnet: rethinking model scaling for Convolutional Neural Networks. Proceedings of the 36th International Conference on Machine Learning, Long Beach, 2019; United States of America.

41. Mahadeokar J, Pesavento G. Open sourcing a deep learning solution for detecting NSFW images. Yahoo Engineering Blog, 2016. Available from: https://yahooeng.tumblr.com/post/151148689421/open-sourcing-a-deep-learning-solution-for

42. You Y, Zhang Z, Cho-Jui Hsieh, Demmel J, Keutzer K. ImageNet training in minutes. arXiv 2018. 1709.05011, 2018.

43. A. Krizhevsky A, Sutskever I, G. E. Hinton GE. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems 2012; 25.

44. Barua A, AlexNet. 2019. Available from: https://arnabfly.github.io/arnab_blog/alexnet/

45. Doğan F, Türkoğlu İ. Derin öğrenme modelleri ve uygulama alanlarına ilişkin bir derleme. DÜMF Mühendislik Dergisi, 2019. doi= https://doi.org/10.24012/dumf.411130