International Journal of Engineering Trends and Technology
ISSN: 2231 – 5381 / https://doi.org/10.14445/22315381/IJETT-V70I10P222

Volume 70 Issue 10, 232-246, October 2022

Original Article

Evaluation of CNN based on Hyperparameters to Detect the Quality of Apples

Deepali M Bongulwar[1], Vijay Prakash Singh[2], S. N. Talbar[3]

[1,2]Department of Electronics and Communication Engineering, School of Engineering, Sri Satya Sai University of Technology and Medical Sciences, Sehore, Madhya Pradesh, India
[3]Department of Electronics and Telecommunication Engineering, SGGSIE&T, SRTMU, Nanded, Maharashtra, India

[1]Corresponding Author : ddeepali2006@yahoo.com

Received: 21 July 2022 Revised: 23 September 2022 Accepted: 12 October 2022 Published: 20 October 2022

Abstract - *It is essential for the food industry that fresh goods are automatically categorized. Various kinds of fruits are in the market, making it difficult to classify them according to quality. Manual sorting and evaluation of agricultural goods are possible, but it is not definitive, time-consuming, subjective, costly, and environmentally sensitive. Thus, fast, accurate, effective and automatic methods need to be introduced for inspecting the quality and grading of fruit. Classification of the quality of fruit and, as a result, gradation is critical in the industry for the development of good quality food products and the highest grade fruits that can be offered in the market. This research develops an automatic fruit grading system to grade apples based on their external qualities. The flaws on the fruit's peel have been used to determine if the apple is fresh/good or rotten/bad. It has been demonstrated that convolutional neural networks (CNNs) are efficient in several agricultural applications. Therefore, CNN architecture is utilized to build and train the classification model. This study's objective is to determine the effectiveness of the proposed CNN model considering four hyperparameters like an optimizer, learning rate, number of epochs and batchsize for determining the quality of apples. The two benchmark datasets, 'Fruits Fresh and Rotten' (Dataset1) and 'FruitsGB' (Fruits Good/Bad) (Dataset2), are employed to analyze the performance of the model. Dataset1 consists of 1,693 fresh and 2,342 rotten apples, and dataset2 consists of 1000 good and 1000 bad apple images. The accuracy and computation time are utilized for the evaluation of the classification performance of the proposed CNN model.*

Initially, the model's accuracy is improved by changing batchsize and keeping hyperparameters like epochs constant, and the best results for each optimizer and all learning rates are found. The batchsize that produced the best results is chosen, and the model is reassessed by adjusting the number of epochs, optimizer, and learning rates. Finally, the best outcomes are obtained. The presented model has achieved 100% accuracy on dataset2 with the optimizers SGDM and ADAM and 99.31% and 99.70% for dataset1 with SGDM and ADAM, respectively. The results reveal that the model's accuracy lowers with the increase in the learning rate, and adding more epochs does not improve the accuracy. The performance categorization of the model is assessed using additional metrics like Precision, Recall, F1 and F2 score, MCC, and AUC. Thus, a score of 100% is achieved for all these metrics.

The optimizer SGDM gives more good results than ADAM and optimizes faster. The acquired experimental findings show that the accuracy of the proposed model relies not only on the hyperparameters but also on the dataset used and its size.

Keywords - *Convolutional Neural Network, Deep Learning, Fruit quality, Fruits Fresh and Rotten dataset, Hyperparameters.*

1. Introduction

Agriculture is extremely important in India due to the rapidly expanding population and escalating food needs. As a result, crop yield must be increased. In India, agriculture has emerged as a significant source of economic growth. One of the most popular fresh fruits is the apple. Apples are rich in vitamins. India exports apple fruit to over 75 countries. Fruit quality is important since it is utilized in various applications such as export and fruit juice production. Early diagnosis of fruit abnormalities can assist minimize the spread of infection to neighbouring areas of the fruit and losses in the agricultural industry. As a result, fruit quality evaluation has emerged as a key research topic. The primary purpose of this study is to check the quality of apples. Detecting damaged crops and categorizing decaying fruits are the two greatest issues in agriculture. The decayed fruits can harm other fresh fruits and impair productivity if they are not correctly identified. Men traditionally did this classification, which was a labour-intensive, time-consuming, and inefficient process. In addition, it raises the expense of production. As a result, we require an automated system capable of reducing

human effort, increasing productivity, and reducing production costs and time. Human involvement in recognizing rotten or damaged fruits differs from individual to individual and is solely based on what they see and sense from the outside. When exporting or importing, the transit process is long and time-consuming, making it difficult to inspect the condition of nearly rotten fruits in large amounts.
Additionally, the inspection of rotting fruit and the decline of its quality substantially impact the food processing industry. As a result, sorting and grading high-quality fruits requires a competent and effective approach. In general, fruit quality is determined by exterior parameters such as intensity, colour, size, surface, and shape and inside parameters such as acid and sugar content. However, the most important factor in fruit sorting is size and colour. Different techniques for grading and inspecting the quality of fruits have been improved in recent years. For two reasons, the existence of outward flaws lowers the price of food. First thing, exterior flaws detract from the appearance of food. Second, they're markers of foods that aren't as nutritious or even diseased. Fruits and vegetables must go through several phases before reaching the buyer, starting with harvesting. Harvesting, sorting, classification, grading, and other processes are included. Manual execution of these operations necessitates a lot of expert resources and a significant amount of time. Because of a lack of interest in such a demanding profession, several countries are experiencing a resource deficit for agricultural jobs. As a result, automation is required in every phase of the fruit and vegetable processing industry.

The topic of automatic identification of spoiled vegetables and fruits has received a lot of interest. The form, colour, and texture are usually altered on the surface of decaying fruit and vegetables. The presence of a foul odour is also a sign of decay. There are a variety of reasons why a fruit or vegetable gets spoiled. Temperature, humidity, air, light, and microorganisms play a role. A single rotten fruit or vegetable can harm multiple fresh fruits and vegetables in inventory. In the fruit and vegetable industry, inventory deterioration results in a significant loss of revenue. By smelling, seeing shape deformation, and determining the colour and texture of the surface, one can detect rotten fruits and vegetables using manual resources. With deep learning algorithms, it is not possible to test the smell of rotting fruits and vegetables. The computers can only detect the changes in the surface features compared with fresh fruits and vegetables. This research tackles the challenge of rotten fruit identification using cutting-edge deep learning algorithms. Thus, a CNN architecture to detect fresh fruits is proposed.

Regarding apples, it has been discovered that certain researchers classify apple fruit as fresh or rotten using transfer learning and CNN from scratch. These studies do not, however, explain how to tune the hyperparameters to get favourable outcomes. This article's primary contribution is the identification of the hyperparameters that had the most influence on how a CNN architecture is trained and produces the best results for sorting fresh apples.

2. Literature Review

An automated fruit grading system for apples is developed in the study [1] to classify them based on their external qualities. Different combinations of numerous features are examined depending on the damage to apple fruits. These characteristics were used as input in this study to train the SVM. The classifier was tested on two separate apple databases: one with 100 colour images of apples, 24 of which were of fruits with varied faults, and another dataset with 112 images of apples; among these, 56 were of apples with varied faults. The highest accuracy for the two datasets was 96.81% and 93.00%, respectively. The authors of the research [2] proposed a model to recognize the fruit in a given image and to determine how much of the fruit is damaged. The Inceptionv3 model was used to improve the detection and identification of fruit disorders. A machine learning-based solution is provided in the paper [3] for detecting the type of apple from a set of 8,554 images. The researchers utilized a widely used deep learning approach for image recognition. The trained model was 100% accurate on a test set that was held out. The researchers in the study [4] first created a dataset of good and diseased apple leaves from several farms located around the Kashmir valley. They built a deep learning model for autonomous apple disease recognition and classification on a prepared dataset that was initialized via transfer learning. They could achieve around 97% accuracy. The authors of the study [5] suggested a framework for categorizing dates using colour, size, and form data retrieved from date images**.** Aseel, Karbalain, and Kupro were the only three different dates chosen for identification and classification. A total of 500 samples were gathered, and 350 were used for the experiments. The proposed framework was used to conduct the experiments in MATLAB. The greatest accuracy of 97.2% was attained. The external fault identification of tomatoes based on deep learning is presented in the study [6]. The research created a dataset of 43,843 images with external flaws. The dataset, which is available online, was strongly skewed toward the healthy group. They found that a fine-tuned ResNet50 model was the most effective model. On the testing dataset, the model had a precision of 94.60%. The best classifier has an 86.6% recall while retaining a 91.70% precision. In the study [7], a multilayer CNN was used to characterize the mango leaves that had been affected by Anthracnose, a fungal disease. A real-time dataset of 1070 images of mango tree leaves collected at the University was used to validate this investigation. The authors of [8] suggested a technique based on D-CNN to identify the guava leaf's diseases automatically. Leaf Spot, Whitefly, Alga and Rust were all classified as prevalent guava leaf diseases using the provided methodology. They also built their dataset, BUGuava Leaf (BUGL2018), which included four different classes and obtained 98.74% accuracy on the test dataset. The study in [9] aimed to detect and recognize a variety of fruits, including apples, sugarcane potatoes, grapes, tomatoes and corn, as well as many plant diseases. The researchers used normal and damaged plant leaves images and trained the models to recognize plant diseases. The system was 100% accurate in detecting and differentiating the kind of plant and the type of plant disease with an accuracy of 96.5%. The various machine-learning approaches for identifying plant diseases were reviewed in [10]. Authors mostly used the SVM classifier to categorize diseases compared to other classifiers. The study showed that the CNN algorithm alone accurately detects a greater number of diseases. With a novel technique, the authors of [11] employed transfer learning models to build an autonomous system to identify and categorize healthy potato leaves and diseased leaves like early blight and late blight with an accuracy of 97.8%. The study in [12] developed a novel approach for identifying plant diseases and pests: Multimodal pre-trained CNN (MLP-CNNs) concatenated with LSTM. The suggested method utilizes several CNN models for deep feature extraction, followed by feature classification using SVM and LSTM classifiers. To examine the effectiveness of the deep models, the CNN models were utilized individually and in concatenated form. When comparing the separate models to the concatenated models, the findings showed that the concatenated models gave better outcomes. Furthermore, the LSTM classifier produced superior results to the SVM classifier. MLP-CNNs and LSTM classifiers achieved the maximum accuracy score of 99.2%. The study in [13] presents a segmentation of the rotting region contained in the apple based on deep learning architecture. For segmentation, UNet and the Enhanced UNet (En-UNet) were used, with encouraging outcomes. The suggested En-UNet model produced 97.46 and 97.54 percent accuracy for training and validation correspondingly, compared to 95.36% for UNet as the basic architecture. En-UNet has an IoU score of 0.866 under a 0.95 threshold, whereas UNet scored 0.66. For the goal of detecting apple defects, frameworks like SSD and YOLOv2 were used in the study [14]. They have created a real-time dataset of 244 defective apple images. Using the dataset, two distinct models SSD and YOLOv2, were developed, trained, and assessed. The SSD-based system was superior to the YOLOv2-based system in terms of performance. To increase feature variety and disease detection accuracy, the study [15] proposes (plant leaf disease detection) PLDD based on DCNN. The suggested method was examined using data from the PlantVillage dataset and tomato plants. For PLDD, the suggested technique offered accuracy in 2-class and 9-class of 98.83% and 96.06%, respectively. The authors of [16] suggested a model for detecting fresh and rotting fruits, focusing on creating transfer learning models. In their project, they used three distinct types of fruits, namely bananas, oranges and apples. The accuracy of MobileNet, VGG16, VGG19 and Xception was compared to the suggested CNN model in this problem. Several hyperparameters, such as batchsize, optimizer, learning rate and the number of epochs, were examined in this research. The outcomes showed that the presented CNN model outperformed transfer learning approaches in properly classifying rotten and fresh fruits with an accuracy of 97.82%. The research [17] looks at an automated detection method using deep learning for apple leaf disease categorization. VGG16, ResNetV2, InceptionV3, and MobileNetV2 were the chosen transfer learning models. Considering hyperparameters such as batchsize, learning rate, and optimizer, ResNetV2 and Adam optimizer showed the highest accuracy of 94%. With a chosen learning rate, the ResNetV2 model made predictions with a 94.7% accuracy. In this study, the influence of several optimizers was investigated, and it was observed that the Adam optimizer is excellent with the ResNetV2 model. The researchers in [18] evaluated the performance of various CNNs while taking into account transfer learning for their training and a few hyperparameters for the recognition and classification of ripe Medjool dates. The CNN architectures evaluated were VGG16/19, ResNet50/101/152, and CNN developed from scratch. The hyperparameters examined were the batch size, optimizer, learning rate and the number of layers and epochs. With an accuracy of 99.32%, they discovered that the VGG19 model achieved the highest performance with 128 batchsize, a learning rate of 0.01 and an Adam optimizer. The ResNet152 model was used in the study [19] to identify the dragon fruit's mellowness. To train the model, TensorFlow and Python were utilized. The developed structure was tested with an additional 100 samples using the ROC and the confusion matrix after being trained using images of the dragon fruit at various stages of mellowness. The testing was done with epoch numbers ranging from 10 to 500. The findings obtained were more accurate than the VGG16/19.

This study created a CNN model to distinguish between fresh and rotten apples. The two different datasets, dataset1 and dataset2, have been utilized for training the model. The model is evaluated by changing the hyperparameters such as learning rate, batchsize, number of epochs and optimizers. The model is implemented using MATLAB R2021a.

The entire paper is divided into sections: The apple quality evaluation requirement is stated in Section 1. Section 2 explores the research on fruit quality detection. Section 3 presents the details of different datasets employed in the system. The methodology and architecture of the suggested CNN model are described in Section 4. The outcomes of the experiments for both datasets with different hyperparameters are presented in Sections 5 and discussed in 6. Section 7 gives the limitation. The conclusion is presented in Section 8.

3. Dataset Information

The proposed model is tested and compared using the two datasets based on different hyperparameters such as batchsize, learning rate, number of epochs and optimizer for checking the quality of fruits. The measurement system (imaging system) was not used. Instead, the dataset was downloaded. The two datasets are Fruits Fresh and Rotten and FruitsGB (Fruits Good/Bad). Table 1 shows the details of both datasets.

Table 1. Dataset Properties

Properties	(Dataset1) Fruits Fresh and Rotten	(Dataset2) Fruits GB (Fruits Good/Bad)
Fresh/Good Apples	1,693	1000
Rotten/Bad Apples	2,342	1000
Total Images	4,035	2000
Training Set Size	3,026	1500
Validation Set Size	1009	500
Image Size	224X224	224X224

3.1. Dataset1

The dataset used was “Fruits fresh and rotten for classification” [20]. This dataset was created by Sriram Reddy Kalluri and updated on Aug 24, 2018. It has six categories of fruits, fresh/ rotten apples, fresh/ rotten bananas, and fresh/ rotten oranges. The dataset is not used as it is; instead, custom augmentations are performed on these images. The testing set is directly taken from the Kaggle fruits dataset, so the assessment is done on a diverse data set. Though it contains three different fruits, only apples were used to test the model. There are 4,035 total images in this dataset. 25% of the images (1009) in the dataset are utilized for validation and 75% (3026) for training, and 996 for testing. The images were resized to 224X224 pixels. Data augmentation helps CNN work better and avoids overfitting. Augmentations like scaling, translation and rotation have been performed on the training images. The training images have performed augments like scaling, translation and rotation. Figure 1 shows nine randomly chosen images.

3.2. Dataset2

The publicly available FruitsGB (Fruits Good/Bad): "Top Indian Fruits with quality" [21], downloaded from IEEE DataPort and last updated in July 2020, was used. The dataset consists of a total of 12000 high-quality images with 12 categories, namely, Good/Bad Guava, Apple, Pomegranate, Orange, Banana, and Lime. Each class in the dataset contains 256x256 pixel 1000 images. The images were obtained from multiple angles, with varied backgrounds, and in various lighting circumstances. Only Good apples and Bad apples were used. The images were resized to 224X224. 1500 images for training and 500 for validation were used. Figure 2 shows 9 images that were chosen at random. The images were rotated, translated in horizontal and vertical directions, and randomly reflected in horizontal directions. The data augmentation has only been applied exclusively to training images.

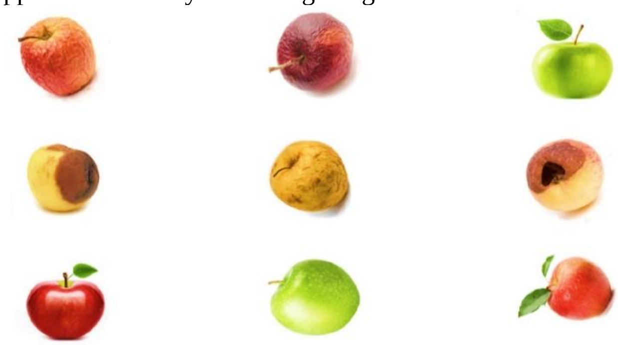


Fig. 1 Fruits Fresh and Rotten (Dataset1)

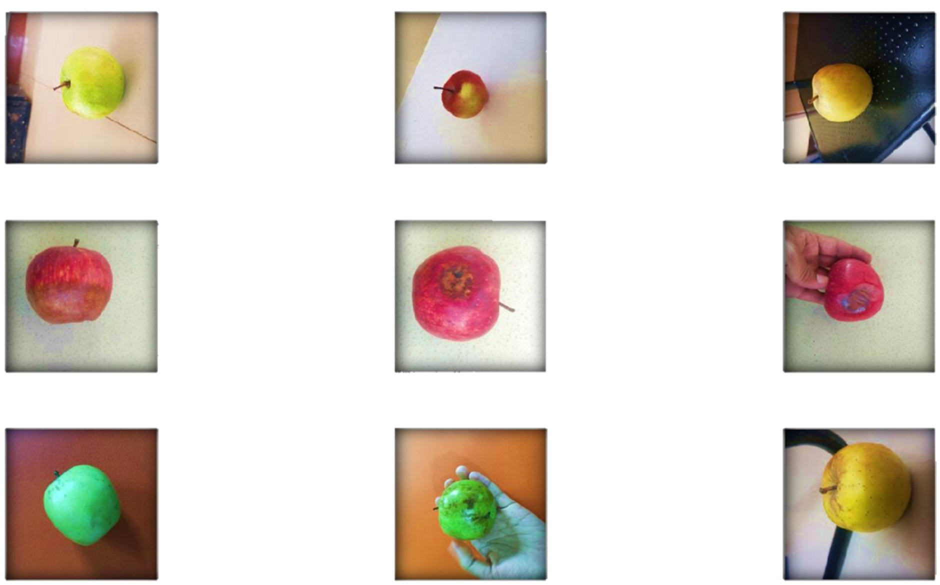


Fig. 2 Fruits GB (Dataset2)

4. Materials and Method

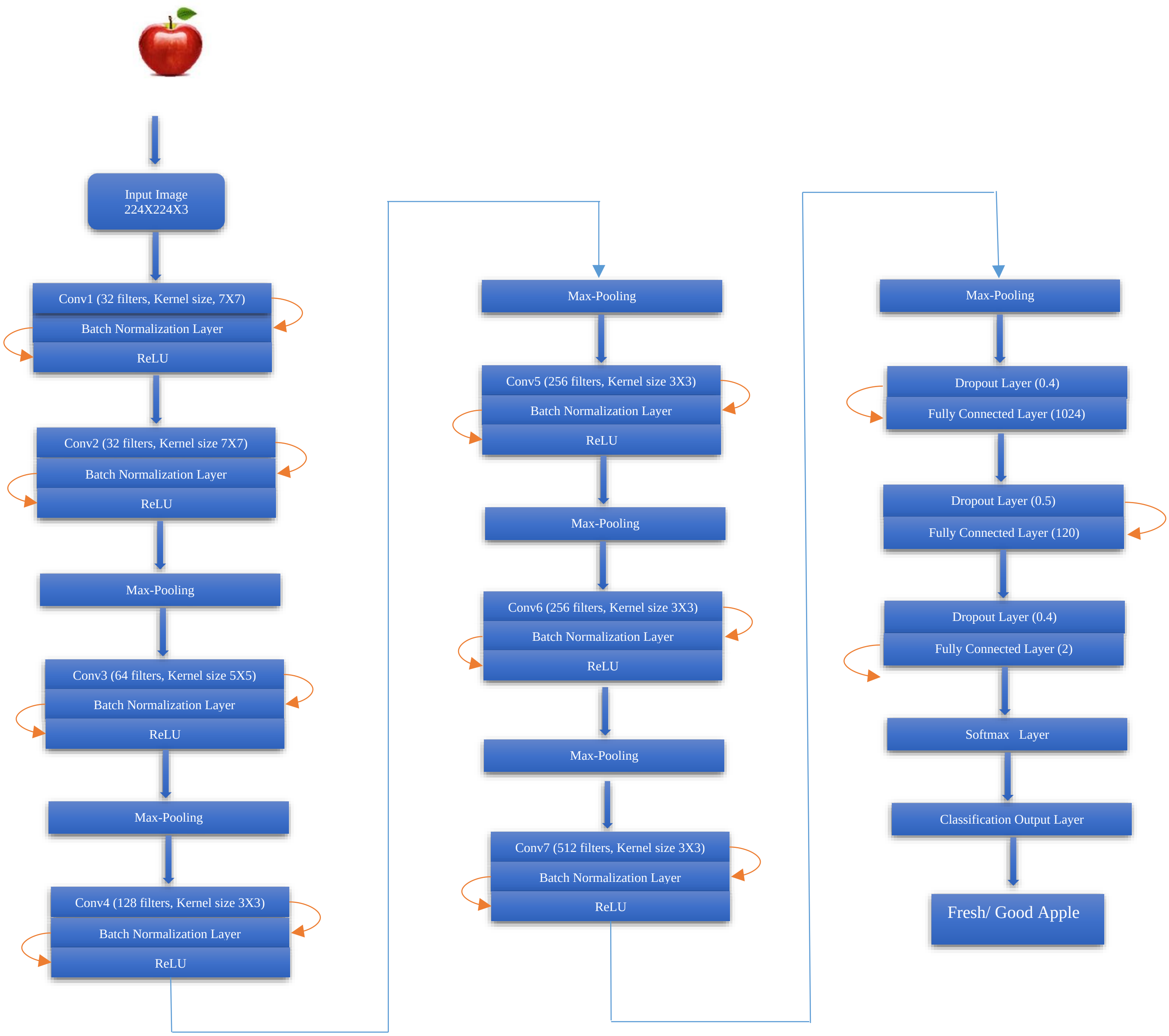


Fig. 3 The Proposed CNN Model

4.1. The proposed CNN Model

The Convolutional neural network architecture is utilized to detect apple quality and classify it using two different datasets. Numerous CNN models exist, including AlexNet, VGG, GoogleNet, ResNet, and others. But, the model is designed from scratch. The proposed model has 35 layers. Figure 3 above shows the proposed model. The input layer is the topmost part where an image is fed. The images were cropped and resized to 224X224X3 pixels. The model comprises several layers stacked on top of one another, including convolution, batch normalization, ReLu, max-pooling, fully connected, and output layer. There are seven convolutional layers and three dropout layers. Every convolutional layer is accompanied by batch normalization and the ReLu layer. The first two convolutional layers have 32 filters of size 7X7 and are succeeded by the max-pooling layer. The convolutional layers 3, 4, 5, 6 and 7 have 64,128,256,256 and 512 filters of size 5X5, 3X3, 3X3,3X3, and 3X3 size kernels, respectively. The maximum convolution depth utilized is 512. As the depth of the feature map increases, more features are extracted, resulting in the desired outputs. The spatial width and height are lowered in the convolution process while the depth is increased. These kernels create feature maps by sliding through the images from left to right and top to bottom. The simplest nonlinear function, rectified linear units (ReLU), is utilized to activate all of the levels in the model. The output of ReLU is 0 if the input is below zero, and it is equal to its input if the input is more than zero. As a result, the network's size is unaffected. The adoption of the ReLU activation function allows for substantially faster training of large networks while boosting nonlinearity at the same time.

$$\text{ReLU}(x) = \begin{cases} x, & x \geq 0 \\ 0, & x<0 \end{cases} \tag{1}$$

The Max-Pooling layer comes next. Max-Pooling reduces the activation function's computational requirements and spatial dimension. Max-pooling is more widely employed due to higher convergence and better performance. Using the max-pooling layer, the images are downsampled. It also lowers the chances of overfitting. It is then subsequently followed by three fully connected layers with 1024,120 neurons and 2 neurons with dropout rates of 40%, 50%, and 40%respectively. Using dropout layers, networks can be normalized and protected against overfitting. The FC layer is capable of determining the image class. The outermost last layer is the softmax layer with two outputs to classify the apple as fresh/good or rotten/bad based on the input image. The number of output neurons is always equivalent to the number of categories in the system. The output of softmax is equal to the number of classes. The highest probability gives the output class. The loss function is calculated using binary cross-entropy. Table 2 below summarizes the proposed CNN's various parameters and configuration details.

4.3. Hyperparameters

Hyperparameters are the parameters that specify how a convolutional network is structured and control the learning process. Their values can be chosen before training begins and will remain the same even when training ends. Therefore, setting the right hyperparameters for a given dataset is very important because it directly impacts the model's performance, resulting from using them during model training. The number of layers, activation function, learning rate, optimizer, epochs, batchsize, and train-validation split ratio are all hyperparameters that can be modified to make CNN more efficient. Here, the parameters of hyperparameters like an optimizer, learning rate, batchsize, and epochs are changed in this experiment. The optimizers, stochastic gradient descent with momentum (SGDM) and adaptive moment estimation (ADAM), are chosen to categorize images in CNN as they are widely used and perform well. The learning rates are 0.0001, 0.0005, and 0.001. The epochs are 20, 30, 40, and 50, while the batchsize is 8, 16, 32, and 64.

Table 2. Hyperparameters

S. No	Hyperparameters	Values
1	**Learning Rate**	0.0001,0.0005,0.001
2	**Batchsize**	8,16,32,64
3	**Epochs**	20,30,40,50
4	**Optimizers**	SGDM, ADAM
5	**Activation Function**	ReLU
6	**Train-Validation Split Ratio**	75% - 25%
7	**L2 Regularization**	0.0005
8	**Drop period for Learning rate**	10
9	**Drop factor for Learning rate**	0.2

4.4. Experimental Framework

NVIDIA GeForce RTX 2060 Super GPU and Intel Core i7-8700K CPU @ 3.70 GHz with 32.0 GB RAM was used. Deep Learning Toolbox was used to implement the model in MATLAB R2021a on Windows 10 Pro.

4.5. Performance Evaluation

Metrics including accuracy, precision, recall, F1 and F2 scores, AUC, and MCC are employed to assess the model. The classification effectiveness of the proposed model is evaluated using the accuracy metric [22]. It represents the percentage of samples that were correctly categorized out of all the samples that were evaluated and are calculated as follows:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \tag{2}$$

Where TP stands for true positives or samples belonging to the category and was appropriately categorized therein; TN stands for true negatives or those who should have been categorized in a different category but weren't;

FP stands for false positives or samples which were mistakenly added to a class despite not being there; and at last, False negatives, or samples belonging to the class but incorrectly placed in a different class, are referred to as FNs. With a balanced distribution of test samples for each class, this metric offers perfect judgement for classification problems. The effectiveness of the given approach is determined using the following performance metrics:

The number of assertions made by a model for each class and the classes to which those predictions belong is summarized in a confusion matrix. It aids in comprehending many sorts of model prediction errors.

The terms "true positives" and "true negatives" refer to accurately predicted observations. Precision is defined as the percentage of accurately anticipated positive findings to all predicted positive findings. The recall is the proportion of accurately predicted, favourable findings to all the class findings. Maximizing the precision will reduce false-positive errors, while increasing recall will reduce false-negative errors. The recall and precision's harmonic means are taken, and equal weight is given to each in obtaining the F-measure. The F-measure makes it possible to compare models and evaluate their performance by giving each model a single score that considers both precision and recall. The F-measure is useful when both precision and recall are crucial. Still, one requires significantly more care than the other, such as when false negatives are more important than false positives or vice versa. It has the effect of lowering the value of precision while raising the value of recall. Precision, recall, and F1-score are asymmetric, and accuracy is sensitive to class imbalance. If both classes are of interest, a binary classification problem can be treated as a multiclass problem with two classes, and the associated multiclass metrics can then be calculated. However, it is worth mentioning that the industry places a premium on precision, which ensures accurate prediction because these items must pass food safety and quality inspections.

Another option for binary classification is to treat the true and predicted classes as two (binary) variables and compute their correlation coefficients (similar to computing the correlation coefficient between any two variables). The better the forecast, the stronger the correlation between true and predicted values.

When applied to classifiers, the phi-coefficient *(φ)* is renamed Matthews Correlation Coefficient (MCC). The formula can easily be used to derive the properties of MCC. MCC equals 1 if and only if the classifier is perfect (FP = FN = 0). In practice, the MCC score varies from -1 to 1, with 0 denoting that the classifier is as good as a coin flip. A high score (around 1) suggests that both classes are accurately predicted by MCC, which considers all four values in the confusion matrix.

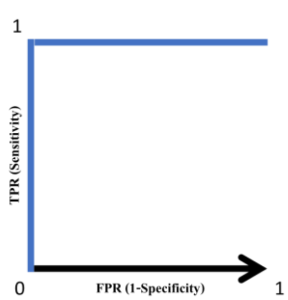
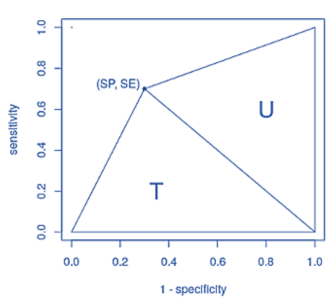


Fig. 4 TPR against FPR



Where SE(Sensitivity) and SP(Specificity)

Fig. 5 AUC with a single threshold point

Table 3. Performance metrics equations

Metric	Equation
Precision	$\frac{TP}{TP + FP}$
Recall	$\frac{TP}{TP + FN}$
F1	$\frac{2 * Precision * Recall}{Precision + Recall}$
F2	$\frac{5 * Precision * Recall}{4 * Precision + Recall}$
MCC	$\frac{TP * TN - FP * FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$
TPR/Recall/ Sensitivity	$\frac{TP}{TP + FN}$
Specificity	$\frac{TN}{TN + FP}$
FPR	1- Specificity

A binary classification assessment can be done using the Receiver Operator Characteristic (ROC) curve. It is a probability curve that contrasts the TPR and FPR at different threshold levels to separate the 'signal' from the 'noise'. The AUC measures how well a classifier can distinguish between the classes, a summary of the ROC curve. The AUC measures the model's capacity to differentiate between positive and negative categories. The AUC is a measure of how well something works. The higher the AUC, the better. When AUC is 1, the classifier successfully differentiates between both Positive and Negative class points.

For the obtained confusion matrix, we can calculate AUC. As shown in figure 5, with a single threshold point, we can consider the AUC as the sum of two triangles, T and U:

$$T= \frac{1*SE}{2} = \frac{SE}{2} = \frac{TP}{2(TP+FP)} \tag{3}$$

$$U = \frac{SP*1}{2} = \frac{SP}{2} = \frac{TN}{2(FN+TN)} \tag{4}$$

$$AUC = T+ U = \frac{TP}{2(TP+FP)} + \frac{TN}{2(FN+TN)} = \frac{SE+SP}{2} \tag{5}$$

5. Results

The efficiency of the suggested model is demonstrated in this section. It was implemented using MATLAB software. The system has a Core i7 with a 3.7 GHz processor and 32 GB memory. Images for training and validation are initially separated from the dataset. It is accomplished by dividing the dataset into the training set, which contains approximately 75% of the images, and the validation set, which contains approximately 25% of the images. In neural network applications, this is the default ratio distribution. A deep convolutional neural network was utilized to categorize and evaluate the quality of apple fruit. The model is evaluated with different hyperparameters like batchsize, epochs, learning rate, and optimizers like SGDM and ADAM on two different datasets as Fresh/Rotten apples or Good/Bad apple datasets. The accuracy of the presented model is assessed for two different datasets with three different learning rates, which differ by 0.0005. Initially, the number of epochs, LR constant, and the batchsize vary. Later, the batchsize with the highest accuracy is kept constant, and the number of epochs is observed for best accuracy. It is explained in the tables given below. Let us consider step by step for each optimizer.

5.1. Adam Optimizer

Keeping the Adam as an optimizer and LR as 0.0001and epochs 30, it can be observed in Table 4 that the highest performance percentage for dataset1 was 99.70% with a batchsize of 16, and for dataset2 the highest accuracy was 100% for the same batchsize 16. So, keeping batchsize constant at 16, the number of epochs is varied from 20, 30, 40, and 50. As in table 5, the highest accuracy for dataset1 was 99.70% and for dataset2 was 100% for 30 epochs. The same is repeated for the other two learning rates. Figure 6 clearly shows the accuracy for each epoch.

Table 6 shows that for a learning rate of 0.0005 and keeping epoch constant at 30, the highest accuracy was obtained for dataset1 as 99.80% with batchsize 16 and for dataset2 as 99.80% with batchsize 32. Therefore, for dataset1 with the same LR now, epochs are varied with batchsize 16 as in table 7 and found the highest accuracy of 99.80% for 30 epochs. Keeping batchsize of 32 for dataset2, the highest accuracy obtained was 100% for 30 epochs. The same is shown in figure 7.

Table 8 shows that keeping LR as 0.001 and epochs 30, dataset1 achieved the highest accuracy of 99.70% for batchsize 16, and dataset2 achieved 96.00% with the same batchsize 16. Now keeping batchsize 16 and LR 0.001 for both datasets as in table 9 and figure 8, the highest accuracy was obtained for 30 epochs for dataset1 at 99.70% and dataset2 at 96.00%.

5.2. SGDM Optimizer

Keeping the SGDM as an optimizer and LR as 0.0001 and epochs 30, it can be observed in Table 4 that the highest performance percentage for dataset1 was 99.31% with a batchsize of 16, and for dataset2 the highest accuracy was 100% for the same batchsize 16. So, keeping batchsize constant at 16, the number of epochs is varied from 20, 30, 40, and 50. As in table 5 and figure 6, the highest accuracy for dataset1 was 99.31% and for dataset2 was 100% for 30 epochs. The same is repeated for the other two learning rates.

Table 6 shows that for a learning rate of 0.0005 and keeping epoch as constant at 30, we found the highest accuracy for dataset1 as 99.50% with batchsize 16 and for dataset2 as 99.80% with the same batchsize 16. Therefore, for dataset1 with the same LR now epochs are varied with batchsize 16 as in table 7 and figure 7 and found the highest accuracy of 99.50% for 30 epochs, and for dataset2, the highest accuracy was 99.80% for the same batchsize 16.

Table 8 shows that keeping LR as 0.001 and epochs 30, dataset1 achieved the highest accuracy of 98.41% for batchsize 16, and dataset2 achieved 99.80% with the same batchsize 16. Now keeping batchsize 16 and LR 0.001 for both datasets as in table 9 and figure 8, the highest accuracy was obtained for 30 epochs for dataset1 at 98.41% and dataset2 at 99.80%.

All the results shown in the above tables and figures are very good. But it is also worth noting that dataset2 produced the two outcomes with the highest accuracy rates of 100% with batchsize 16, learning rate 0.0001 and epochs 30 for both SGDM and ADAM optimizers.

Regarding the time parameter in all the tables, whatever may be the accuracy, the corresponding processing time was always less for the SGDM optimizer compared to the ADAM optimizer for every result obtained for both datasets. The processing time required for the best result of 100% accuracy for the dataset2 with SGDM optimizer was 14 min 01 seconds and with ADAM optimizer was 14 min 42 seconds.

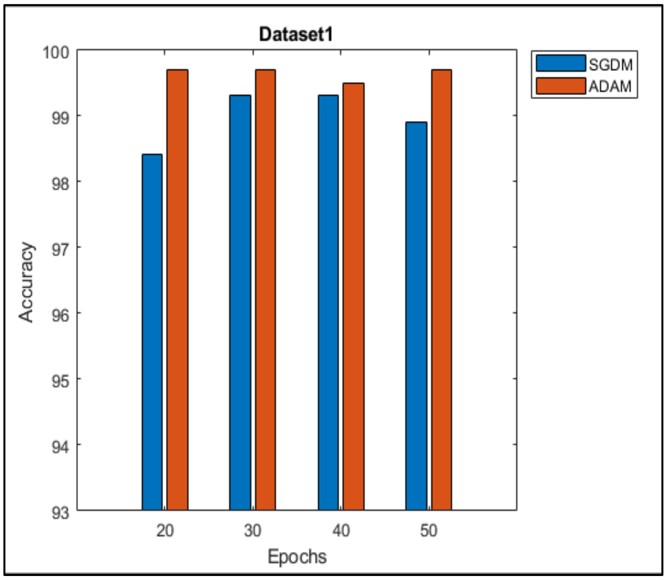
The training of the model for the two best results is depicted in Figures 9 and 10. The resulting classified images are depicted in Figures 11 and 12. Similarly, the confusion matrix for both the optimizers shows 100% accuracy in classifying the apples based on their quality, as shown in Figures 13 and 14, respectively.

Table 4. L.R. = 0.0001 Epoch = 30

		Dataset1				Dataset2			
		SGDM		ADAM		SGDM		ADAM	
S. No.	Batch size	Accuracy	Time	Accuracy	Time	Accuracy	Time	Accuracy	Time
1	8	98.61	64 m 14 s	99.70	68 m 38 s	99.60	20 m 29 s	99.80	22 m 9 s
2	16	**99.31**	38 m 42 s	**99.7 0**	40 m 38 s	**100**	14 m 01 s	**100**	14 m 42 s
3	32	99.21	23 m 0 s	99.60	23 m 32 s	99.80	9 m 21 s	99.80	9 m 25 s
4	64	98.61	21 m 45 s	98.71	22 m 16 s	99.60	9 m 8 s	100	9 m 22 s

Table 5. L.R. = 0.0001 Batchsize = 16

Dataset1						Dataset2			
		SGDM		ADAM		SGDM		ADAM	
S. No	Epoch	Accuracy	Time	Accuracy	Time	Accuracy	Time	Accuracy	Time
1	20	98.41	26 m 2 s	99.70	27 m 6 s	99.60	9 m 54 s	100	9 m 56 s
2	30	**99.31**	38 m 42 s	**99.70**	40 m 38 s	**100**	14 m 01 s	**100**	14 m 42 s
3	40	99.31	51 m 31 s	99.50	54 m 10 s	99.80	19 m 41 s	99.80	19 m 33 s
4	50	98.91	64 m 11 s	99.70	66 m 28 s	99.40	24 m 20 s	100	24 m 42 s



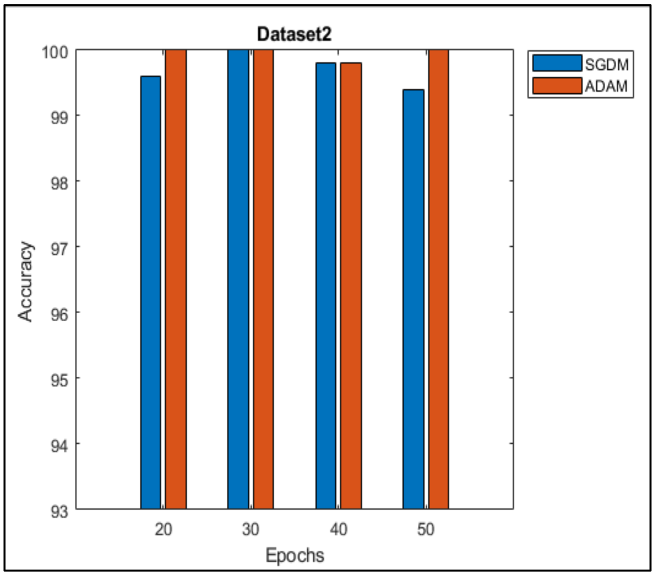


Fig. 6 Graphical representation of Table 5

Table 6. L. R. = 0.0005 Epoch =30

Dataset1						Dataset2			
		SGDM		ADAM		SGDM		ADAM	
S. No.	Batch size	Accuracy	Time	Accuracy	Time	Accuracy	Time	Accuracy	Time
1	8	96.83	54 m 51 s	99.70	61 m 54 s	98.80	20 m 23 s	93.80	21 m 40 s
2	16	**99.50**	39 m 9 s	**99.80**	40 m 22 s	**99.80**	14 m 46 s	96.20	14 m 51 s
3	32	96.92	22 m 55 s	98.91	23 m 39 s	99.20	9 m 17 s	**99.80**	10 m 15 s
4	64	98.61	21 m 34 s	99.40	22 m 11 s	99.40	9 m 16 s	99.60	9 m 45 s

Table 7. L.R. = 0.0005

Dataset1						Dataset2			
		SGDM		ADAM		SGDM		ADAM	
		Batchsize= 16				Batchsize=16		Batchsize= 32	
S. No	Epoch	Accuracy	Time	Accuracy	Time	Accuracy	Time	Accuracy	Time
1	20	98.41	26 m 0 s	98.31	28 m 32 s	99.20	9 m 56 s	98.20	6 m 34 s
2	30	**99.50**	39 m 9 s	**99.80**	40 m 22 s	**99.80**	14 m 46 s	**100**	9 m 40 s
3	40	98.71	51 m 25 s	99.31	53 m 10 s	99.60	19 m 33 s	99.20	13 m 10 s
4	50	98.12	59 m 58 s	99.70	62 m 54 s	99.60	24 m 31 s	99.60	16 m 11 s

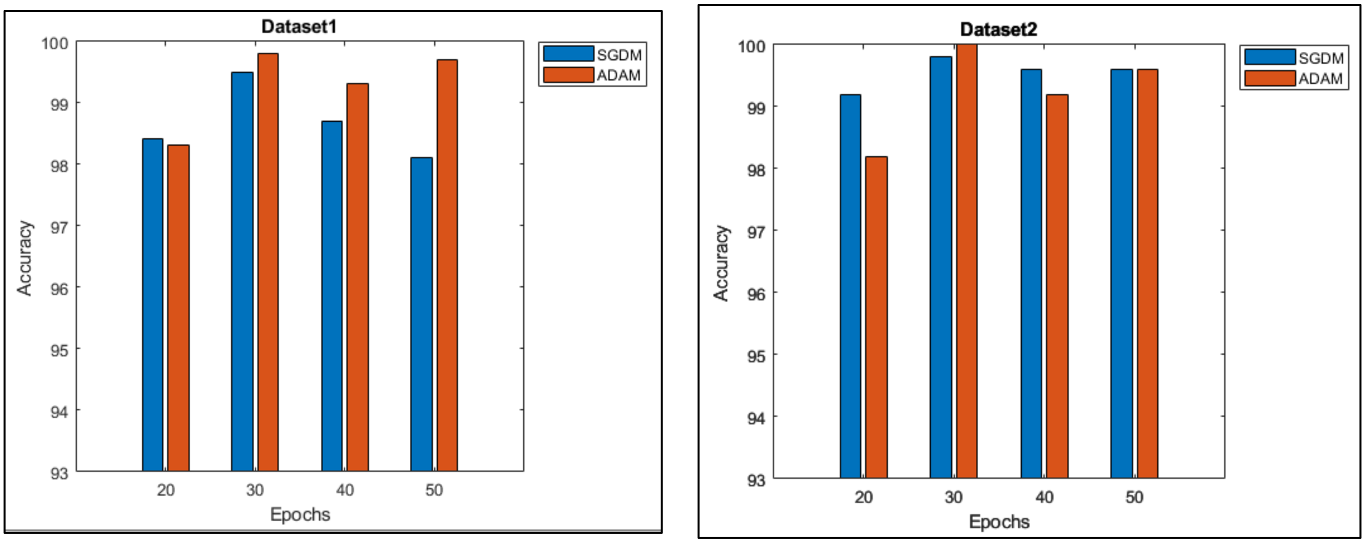


Fig. 7 Graphical representation of Table 7

Table 8. L.R. = 0.001 Epoch = 30

		Dataset1				Dataset2			
		SGDM		ADAM		SGDM		ADAM	
S. No	**Batch size**	**Accuracy**	**Time**	**Accuracy**	**Time**	**Accuracy**	**Time**	**Accuracy**	**Time**
1	8	98.12	54 m 29 s	99.50	61 m 5 s	50	20 m 9 s	91.00	19 m 7 s
2	16	**98.41**	40 m 27 s	**99.70**	38 m 57 s	**99.80**	14 m 31 s	**96.00**	14 m 58 s
3	32	98.31	21 m 50 s	98.71	23 m 57 s	99.20	9 m 10 s	94.60	9 m 12 s
4	64	97.02	20 m 29 s	98.71	22 m 16 s	99.00	9 m 9 s	95.40	9 m 41 s

Table 9. L.R. = 0.001 Batchsize = 16

		Dataset1				Dataset2			
		SGDM		ADAM		SGDM		ADAM	
S. No.	**Epoch**	**Accuracy**	**Time**	**Accuracy**	**Time**	**Accuracy**	**Time**	**Accuracy**	**Time**
1	20	95.93	26 m 11 s	99.11	27 m 9 s	99.00	9 m 58 s	94.60	10 m 15 s
2	30	**98.41**	40 m 27 s	**99.70**	38 m 57 s	**99.80**	14 m 47 s	**96.00**	14 m 58 s
3	40	96.03	55 m 56 s	99.31	58 m 44 s	99.00	18 m 56 s	93.60	20 m 23 s
4	50	95.83	60 m 11 s	99.60	70 m 53 s	99.40	24m 27 s	93.80	23 m 53 s

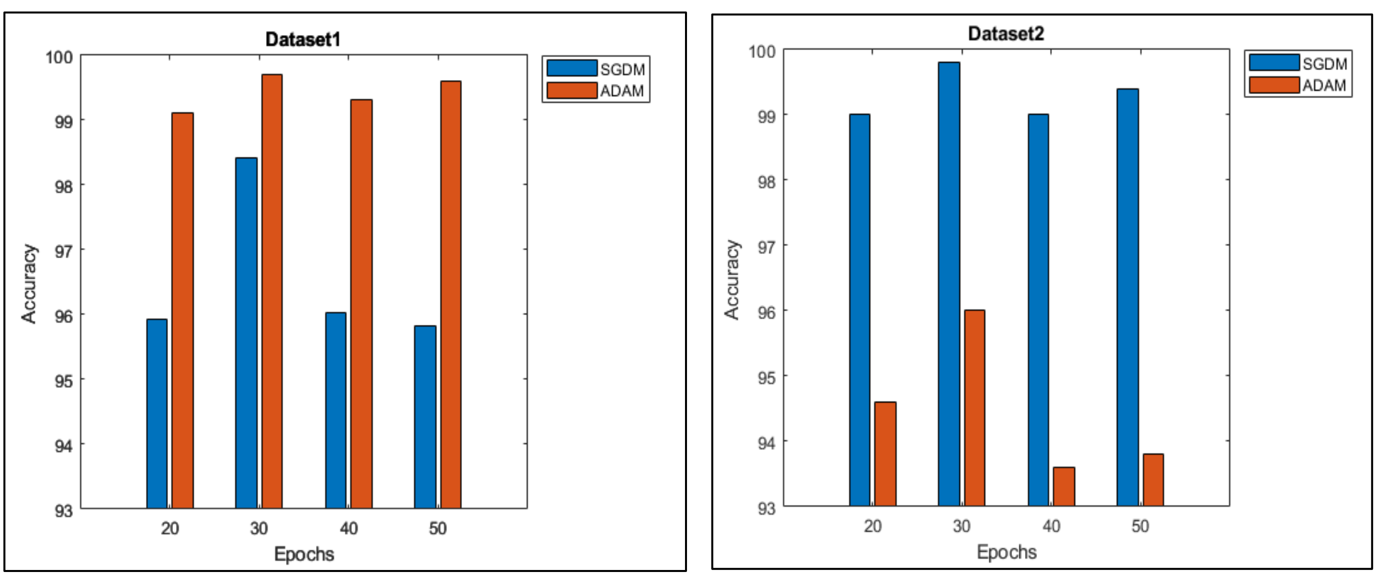


Fig. 8 Graphical representation of Table 9

5.3. Best Results

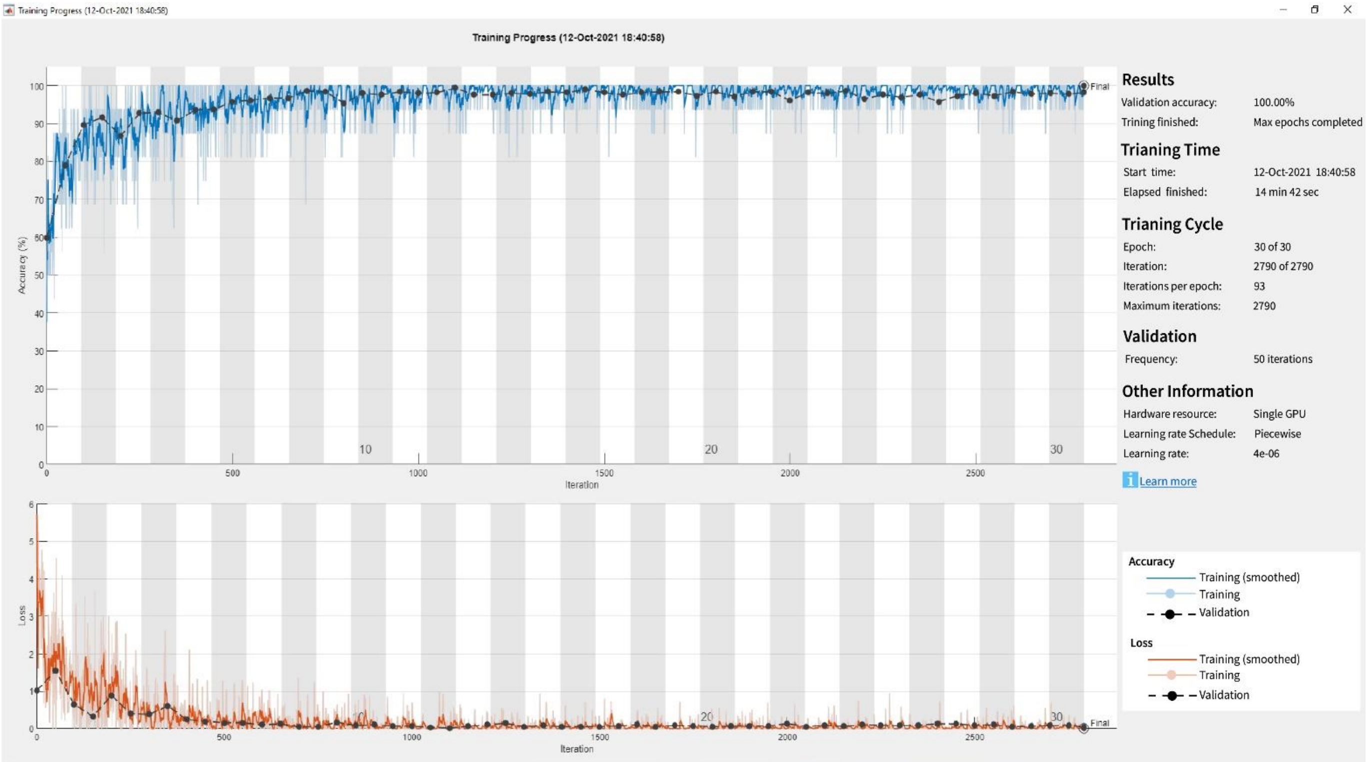


Fig. 9 Training with Optimizer = ADAM, Batchsize=16 L.R=0.0001

Fig. 10 Training with Optimizer=SGDM, Batchsize=16 L.R=0.0001

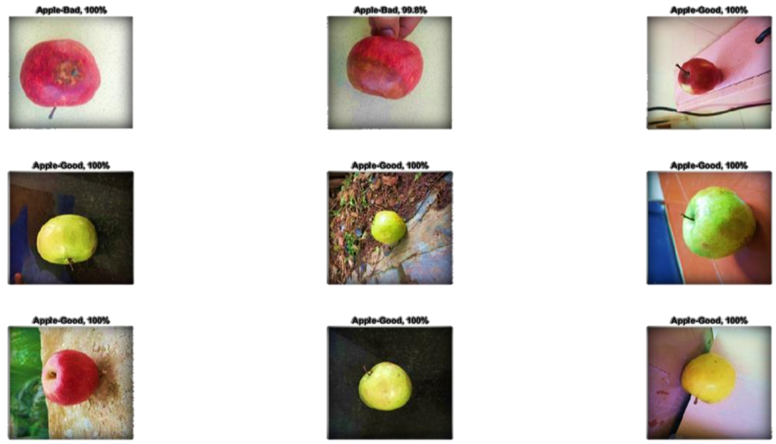


Fig. 11 Results with SGDM Optimizer

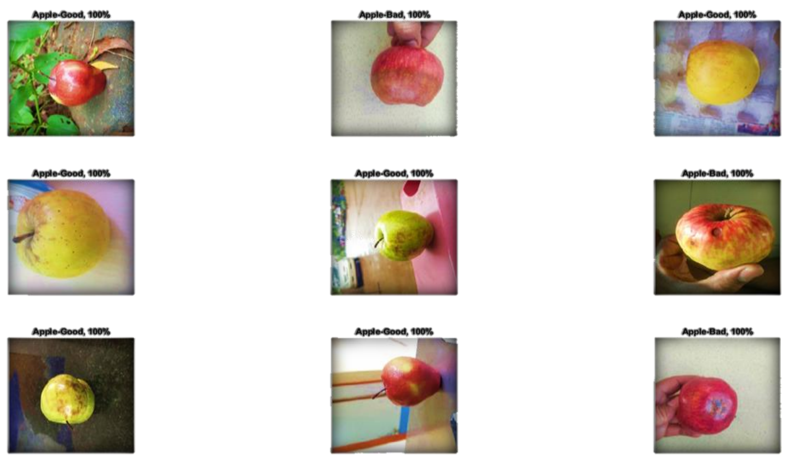


Fig. 12 Results with ADAM Optimizer

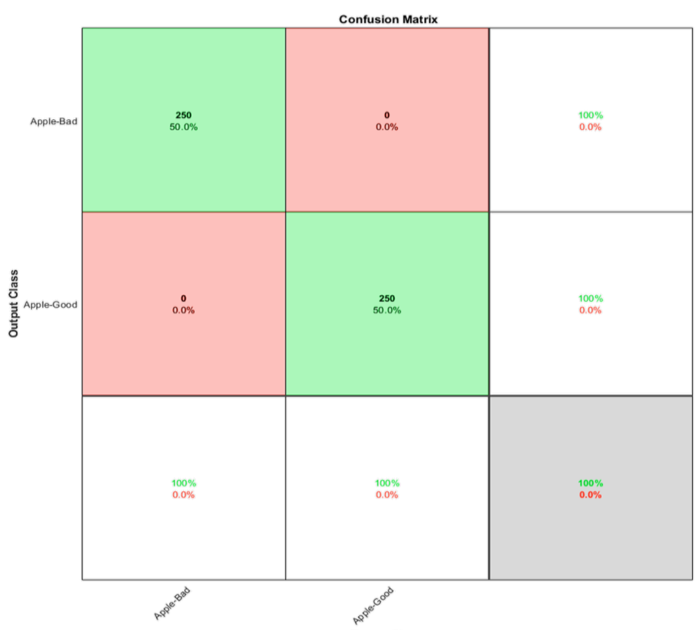


Fig. 13 Confusion Matrix for SGDM Optimizer

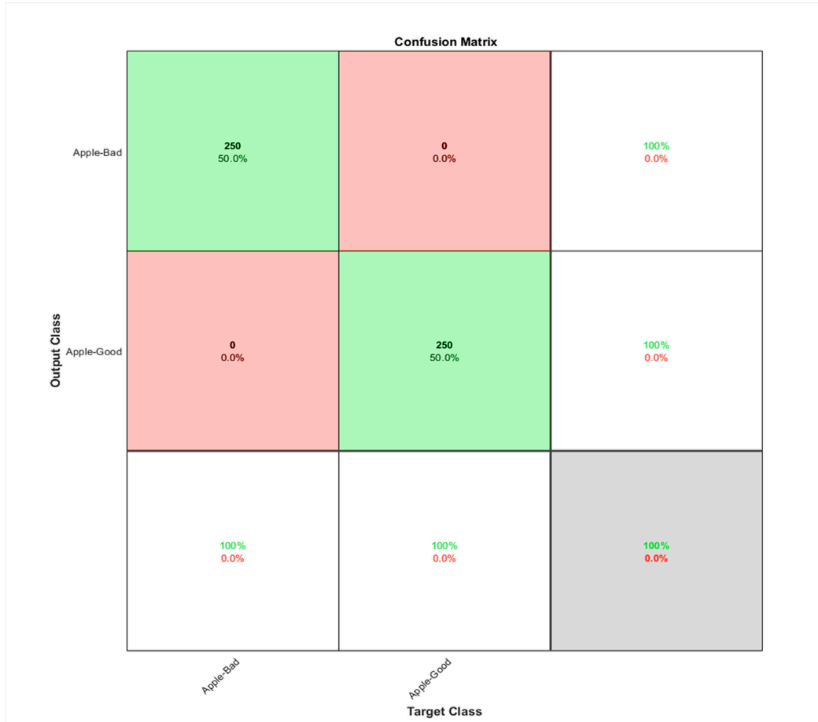


Fig. 14 Confusion Matrix for ADAM Optimizer

Here, for both the optimizers, TP= 250, TN = 250, FP=0, FN =0

Table 10. Performance results of classification

Optimizer	Accuracy	Precision	Recall	F1 Score	F2 Score	MCC	AUC
SGDM	1	1	1	1	1	1	1
ADAM	1	1	1	1	1	1	1

Table 11. Comparison performance for classifying apples as rotten or fresh

S. No.	Author and Year	Model	Dataset	Number of Images	Accuracy
1	[23] Wang, HongJun, et al (2020)	ResNet-50	Own dataset	4035	98.67%
2	[24] Karakaya et el. (2019)	CNN Model	Fruits Fresh and Rotten dataset	3200	99.17%
3	[25] Alharbi et el. (2020)	Improved Mask R-CNN	Own dataset	2000	96.86%
4	Proposed Method	Proposed CNN Model	Fruits GB (Good /Bad)	2000	100%

6. Discussion

The suggested CNN model is evaluated with different hyperparameters like batchsize, learning rate, epochs and optimizers on two different datasets with different sizes to sort the apples based on their quality.

The results in table 4 show that for dataset1, the best accuracy was 99.31% and 99.70% and for dataset2 was 100% for SGDM and ADAM optimizer, respectively. Similar is the case for table 5 and is graphically represented in figure 6. However, it is observed that the corresponding accuracy and processing time for SGDM were always less than the ADAM optimizer for both the datasets in Tables 4 and 5.

The results in table 6 show that for dataset1, the best accuracy was 99.50% and 99.80% and for dataset2 was 99.80% and 99.80% for SGDM and ADAM optimizer, respectively. Table 7 and figure 7 show that for dataset1, the best accuracy was 99.50% and 99.80% and for dataset2 was 99.80% and 100% for SGDM and ADAM optimizer, respectively. However, it is observed that the corresponding accuracy and processing time for SGDM was always less compared to the ADAM optimizer for dataset1. However, this was not the case with dataset2 in table 7 as the batchsize was different.

The results in table 8 show that for dataset1, the best accuracy was 98.41% and 99.70% and for dataset2 was 99.80% and 96.00% for SGDM and ADAM optimizer, respectively. Similar was the case in table 9 and figure 8.

However, it is observed that the corresponding accuracy and processing time for SGDM was always less compared to the ADAM optimizer for dataset1; this was not the case with dataset2 in tables 8 and 9.

It is noticed that increasing the number of epochs for all the datasets resulted in a longer processing time, but the accuracy did not increase. Thus, most of the time, the highest accuracy was achieved with a batchsize of 16. Later, keeping the batchsize as 16, the epoch size is increased to 20,30,40,50. All these observations are done for every learning rate of 0.0001, 0.0005 and 0.001 for both the optimizers on each dataset. Tables 4, 5, 6, 7, 8, and 9 and figures 6, 7 8 show the results of varying batchsize, epochs and learning rates for both optimizers. We found that for dataset1, the best accuracy achieved is 99.50% and 99.70% for batchsize 16 and epochs 30 and 0.0001 LR for SGDM and ADAM optimizers, respectively. The accuracy obtained for dataset2 is 100% for both the optimizers for batchsize 16, 30 epochs and 0.0001 LR, as shown in Figures 9 and 10. It is also observed that the time required to execute the model is more for ADAM optimizers than SGDM. Again, the best accuracy is obtained with a small learning rate and decreases with an increase in LR.

Finally, our study evaluated the proposed CNN model's performance by modifying hyperparameters producing the model with the best performance, with 100% accuracy for dataset2. The model's performance was also evaluated using other metrics, as shown in Table 10. Table 10 shows accuracy, precision, recall, F1 score, F2 score, MCC and AUC as 1. It illustrates the perfect classification between apples that are fresh and rotten.

The performance of the suggested approach is compared with different techniques used for apples. The comparative study of the suggested CNN model is depicted in Table 11. The analysis demonstrates that the suggested architecture produces better classification outcomes for fresh and rotting apples. The study [23] suggests a model based on Mask R-CNN that precisely detects defects on the fruits' surfaces, like apples, peaches, oranges, and pears. The results show that the surface lesions detection accuracy for apples is 96.86%. The study [24] uses comparative analysis to separate fresh samples from those that are rotting from a dataset containing three different types of fruits. The suggested vision-based approach employs convolutional neural networks, histograms, grey-level co-occurrence matrices, a collection of features, and a bag of features to extract features. The convolutional neural networks consistently produced the best success rates with accuracy for fresh and rotten apples were 98.7% and 96.67%, respectively. The classification of good and diseased apples is the subject of research [25]. The experiment was carried out using different CNN models with a varied percentage of training and testing images. When 90% of the training and 10% of the testing datasets were used, model 5 had the highest accuracy of 99.17%.

7. Limitations

The model is computationally more expensive as we have focused more on accuracy.

8. Conclusion

The accuracy and processing time of the proposed CNN model were evaluated with different hyperparameters like batchsize, epochs, learning rate and optimizers on two different datasets as these parameters have been proven to enhance the performance of convolutional networks. The outcomes showed that the suggested CNN model obtained state-of-the-art accuracy of 100% for dataset2, performing the best for sorting apples with 16 batchsize, 0.0001 LR, and 30 epochs for both SGDM and ADAM optimizers and processing time was 14 min 01 seconds and 14 min 42 seconds respectively.

Furthermore, the model was evaluated with other metrics and achieved a 100% score for precision, recall, F1 and F2 scores, MCC and AUC. Thus, it is concluded that the proposed model can be useful to the producers of apples to improve their sorting process to detect fresh/good and rotten/bad apples.

This research demonstrates that the accuracy of the CNN model relies on the hyperparameters, the dataset used and its size.

As a future aspect, the model can also be evaluated to get the best accuracy and the least processing time by modifying other hyperparameters like the number of hidden layers, different optimizers, different activation functions and so on.

References

[1] Bhargava, Anuja, and Atul Bansal, "Machine Learning-Based Quality Evaluation of Mono-Coloured Apples," *Multimedia Tools and Applications* vol. 79, no. 31, pp. 22989-23006, 2020.

[2] Nikhitha, M., S. Roopa Sri, and B. Uma Maheswari, "Fruit Recognition and Grade of Disease Detection Using Inception V3 Model", *3rd International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, IEEE, pp. 1040-1043, 2019.

[3] Al-Shawwa, Mohammed O, "Classification of Apple Fruits By Deep Learning", *International Journal of Academic Engineering Research(IJAER)*, vol. 3, no. 12, 2020.

[4] Khan, Asif Iqbal, S. M. K. Quadri, and Saba Banday, “Deep Learning for Apple Diseases: Classification and Identification,” *International Journal of Computational Intelligence Studies* vol. 10, no. 1, pp. 1-12, 2021

[5] Magsi, Aurangzeb, J. Ahmed Mahar, and Shahid Hussain Danwar, “Date Fruit Recognition Using Feature Extraction Techniques and Deep Convolutional Neural Network,” *Indian Journal of Science and Technology* vol.12, no. 32, pp. 1-12, 2019.

[6] Da Costa, Arthur Z., Hugo EH Figueroa, and Juliana A. Fracarolli, “Computer Vision-Based Detection of External Defects on Tomatoes Using Deep Learning,” *Biosystems Engineering* vol. 190, pp. 131-144, 2020.

[7] Singh, Uday Pratap, Siddharth Singh Chouhan, Sukirty Jain, and Sanjeev Jain, “Multilayer Convolution Neural Network for the Classification of Mango Leaves Infected By Anthracnose Disease,” *IEEE Access* vol. 7, pp. 43721- 43729, 2019.

[8] Howlader, Md Rasel, Umme Habiba, Rahat Hossain Faisal, and Md Mostafijur Rahman, “Automatic Recognition of Guava Leaf Diseases Using Deep Convolution Neural Network", *International Conference on Electrical, Computer and Communication Engineering (ECCE)*, pp. 1-5. IEEE, 2019.

[9] Militante, Sammy V., Bobby D. Gerardo, and Nanette V. Dionisio, “Plant Leaf Detection and Disease Recognition Using Deep Learning", *IEEE Eurasia Conference on Iot, Communication and Engineering (ECICE)*, pp. 579-582. IEEE, 2019.

[10] Shruthi, U., V. Nagaveni, and B. K. Raghavendra, “A Review on Machine Learning Classification Techniques for Plant Disease Detection", *International Conference on Advanced Computing & Communication Systems (ICACCS)*, pp. 281-284. IEEE, 2019.

[11] Tiwari, Divyansh, Mritunjay Ashish, Nitish Gangwar, Abhishek Sharma, Suhanshu Patel, and Suyash Bhardwaj, “Potato Leaf Diseases Detection Using Deep Learning", *4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, *IEEE*, pp. 461-466, 2020.

[12] Turkoglu, Muammer, Davut Hanbay, and Abdulkadir Sengur, “Multi-Model LSTM-Based Convolutional Neural Networks for Detection of Apple Diseases and Pests,” *Journal of Ambient Intelligence and Humanized Computing,* pp.1-11, 2019.

[13] Roy, Kyamelia, Sheli Sinha Chaudhuri, and Sayan Pramanik, “Deep Learning Based Real-Time Industrial Framework for Rotten and Fresh Fruit Detection Using Semantic Segmentation,” *Microsystem Technologies,* vol.27, no. 9, pp. 3365-3375, 2021.

[14] Siddiqi, Raheel, “Automated Apple Defect Detection Using State-of-the-Art Object Detection Techniques,” *SN Applied Sciences*, vol. 1, no. 11, pp.1 -12, 2019.

[15] Bade Ashwini Vivekanand, M. Suresh Kumar, "Deep Learning Based Tomato PLDD,” *International Journal of Engineering Trends and Technology*, vol.70. no. 7 pp. 414- 421, 2022. Crossref, https://doi.org/10.14445/22315381/IJETT-V70I7P243.

[16] Palakodati, S.S.S., Chirra, V.R.R., Yakobu, D. and Bulla, S., “Fresh and Rotten Fruits Classification Using CNN and Transfer Learning”, *Revue d'Intelligence Artificielle (RIA)*, vol.*34,* no. 5, pp. 617-622, 2020.

[17] Alsayed, Ashwaq, Amani Alsabei, and Muhammad Arif, "Classification of Apple Tree Leaves Diseases Using Deep Learning Methods,” *International Journal of Computer Science & Network Security* vol. 21, no. 7, pp. 324-330, 2021.

[18] Pérez-Pérez, Blanca Dalila, Juan Pablo Garcia Vazquez, and Ricardo Salomón-Torres, "Evaluation of Convolutional Neural Networks' Hyperparameters with Transfer Learning to Determine Sorting of Ripe Medjool Dates", *Agriculture*, vol. 11, no .2, pp. 115, 2021.

[19] Vijayakumar, T., and Mr R. Vinothkanna, “Mellowness Detection of Dragon Fruit Using Deep Learning Strategy,” *Journal of Innovative Image Processing (JIIP)* vol. 2, no. 01, pp. 35-43, 2020.

[20] Kaggle Dataset Link, [Online]. Available: https://www.kaggle.com/Datasets/Sriramr/Fruits-Fresh-and-Rotten-for-Classification

[21] IEEE Dataport Dataset Link, [Online]. Available: https://IEEE-Dataport.org/Open-Access/Fruitsgb-Top-Indian-Fruits-Quality

[22] Challa, Sravan Kumar, Akhilesh Kumar, and Vijay Bhaskar Semwal, “A Multi-Branch CNN-Bilstm Model for Human Activity Recognition Using Wearable Sensor Data", *The Visual Computer,* pp. 1-15, 2021.

[23] Wang, Hongjun, Et Al, “Research on Detection Technology of Various Fruit Disease Spots Based on Maskr-CNN", *International Conference on Mechatronics and Automation (ICMA)*, IEEE, 2020.

[24] Karakaya, Diclehan, Oguzhan Ulucan, and Mehmet Turkan, “A Comparative Analysis on Fruit Freshness Classification,” *Innovations in Intelligent Systems and Applications Conference (ASYU)*, IEEE, 2019.

[25] Alharbi, Asmaa Ghazi, and Muhammad Arif, “Detection and Classification of Apple Diseases Using Convolutional Neural Networks", *2nd International Conference on Computer and Information Sciences (ICCIS)*, IEEE, 2020.